Aus dem Institut für Mathematik
der Universität zu Lübeck

Direktor:
Prof. Dr. Jürgen Prestin

Nonequispaced FFT

Generalisation and Inversion

Inauguraldissertation
zur Erlangung der Doktorwürde
der Universität zu Lübeck
- Aus der Technisch-Naturwissenschaftlichen Fakultät -

Vorgelegt von
Stefan Kunis
aus Oelsnitz, Vogtland

Lübeck, im Juni 2006

Vorsitzender	Prof. Dr. Thomas Martinetz, Universität zu Lübeck
Gutachter	Prof. Dr. Daniel Potts, Technische Universität Chemnitz
	Prof. Dr. Jürgen Prestin, Universität zu Lübeck
	Prof. Dr. Wolfgang Hackbusch, Max-Planck-Institut Leipzig
23.08.2006	Tag der mündlichen Prüfung

Contents

1 Introduction 5

2 Polynomials and discrete Fourier transforms 9
2.1 Trigonometric polynomials . . . 9
  2.1.1 Sampling and the discrete Fourier transform . . . 11
  2.1.2 Trigonometric kernels . . . 14
2.2 Trigonometric polynomials on the hyperbolic cross . . . 19
  2.2.1 Sampling and the sparse discrete Fourier transform . . . 20
  2.2.2 Kernels on the hyperbolic cross . . . 21
2.3 Polynomials on the sphere . . . 22
  2.3.1 Sampling and the discrete spherical Fourier transform . . . 24
  2.3.2 Kernels on the sphere . . . 25

3 Nonequispaced FFT and generalisations 31
3.1 Nonequispaced fast Fourier transform . . . 31
  3.1.1 Unified approach . . . 32
  3.1.2 Computational requirements . . . 38
  3.1.3 Numerical experiments . . . 41
  3.1.4 Concluding remarks and comments . . . 47
3.2 Nonequispaced FFT on the hyperbolic cross . . . 48
  3.2.1 Bivariate case . . . 49
  3.2.2 Trivariate case . . . 50
  3.2.3 Numerical experiments . . . 53
  3.2.4 Concluding remarks . . . 53
3.3 Miscellaneous transforms and implementation . . . 55
  3.3.1 Fast spherical Fourier transform . . . 55
  3.3.2 Real trigonometric transforms . . . 55
  3.3.3 Software library . . . 56

4 Fast Gauss transform 59
4.1 Univariate Gaussian . . . 59
  4.1.1 Derivation of Algorithm 4.1 . . . 60
  4.1.2 Error estimates . . . 61
  4.1.3 Numerical experiments . . . 63
4.2 Spherical Gaussian . . . 66
  4.2.1 Derivation of Algorithm 4.2 . . . 66
  4.2.2 Error estimates . . . 67
  4.2.3 Numerical experiments . . . 68
4.3 Notes and comments . . . 70

5 Inverse NFFT 71
5.1 Least squares approximation . . . 71
  5.1.1 Derivation of Algorithm 5.1 . . . 72
  5.1.2 Convergence results . . . 72
  5.1.3 Numerical experiments . . . 74
  5.1.4 Concluding remarks . . . 75
5.2 Optimal interpolation . . . 77
  5.2.1 Derivation of Algorithm 5.2 . . . 77
  5.2.2 Convergence results . . . 78
  5.2.3 Numerical experiments . . . 87
  5.2.4 Concluding remarks . . . 88
5.3 Generalisation and application . . . 90
  5.3.1 Sparse reconstruction . . . 90
  5.3.2 Gridding in magnetic resonance imaging . . . 92
5.4 Notes and comments . . . 96

1

Introduction

Fast Fourier transforms (FFTs) belong to the "10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century", cf. [DS00]. The classic algorithm in [CT65] computes the discrete Fourier transform

$$f_j = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{-2\pi\mathrm{i}\frac{kj}{N}}$$

for $j = -\frac{N}{2}, \ldots, \frac{N}{2} - 1$ and given complex coefficients $\hat{f}_k \in \mathbb{C}$. Using a divide and conquer approach, the number of floating point operations is reduced from $\mathcal{O}(N^2)$ for a straightforward computation to only $\mathcal{O}(N \log N)$. In conjunction with publicly available efficient implementations including multivariate versions in [FJ05], the fast Fourier transform has become of great importance in scientific computing. Its applications include for example digital signal and image processing as well as the numerical solution of differential and integral equations. The author of [Loa92, p. ix] summarises "life as we know it would be very different without the FFT".

However, two shortcomings of traditional schemes are the need for equispaced sampling and the restriction to the system of complex exponential functions. During the last two decades, both problems have attracted much attention. We refer to [MR95] for an introduction to discrete Fourier analysis with orthogonal polynomials, spherical harmonics, and more general functions. In particular, Fourier transforms on the sphere [DH94, PST98b, Moh99, ST02, HKMR03, RT06] and on the hyperbolic cross [BD89, Hal92] have been suggested recently. The second branch of generalisations dispenses with the need for equispaced sampling and establishes so-called nonequispaced FFTs. An excellent review of these algorithms is given in [PST01], its references include the most prominent approaches in this area [DR93, Bey95, AD96, Ste98]. The common concept in most generalised fast Fourier transforms - and other algorithms for the computation with dense matrices including the fast multipole method, hierarchical matrices, and mosaic-skeleton approaches [GR87, HN89, Tyr96, Hac99] - is the use of approximation schemes. The functions involved, e.g. the complex exponentials, are replaced by approximations with prescribed accuracy that allow for the design of a fast algorithm. We trade exactness for efficiency; instead of precise computations, i.e., up to machine precision for actual implementations, the proposed methods guarantee a given target accuracy.

The principal contribution of this thesis is the further development of computational tools in Fourier analysis that generalise the FFT:

1. We start with a unified introduction to *discrete* Fourier transforms on the d-dimensional torus, the hyperbolic cross, and the sphere as evaluations of corresponding polynomials at a set of sampling nodes. Furthermore, we follow the "smoothness-and-decay" principle and construct for their subsequent usage sharply localised trigonometric and spherical polynomials from "smooth" Fourier coefficients.

2. Subsequently, *fast* Fourier transforms are considered. In contrast to [BD89, DH94, PST98b], our FFTs on the sphere and on the hyperbolic cross allow for arbitrary sampling schemes. Moreover, we focus on the substantially different computational costs that arise in various implementations of the nonequispaced fast Fourier transform and improve [DR93, Bey95, AD96, Ste98, PST01] to trade precomputation storage as well as target accuracy for computation time in a manageable way.

3. We contribute new versions of the fast Gauss transform. Compared to multipole approximations in previously suggested schemes [Str91, GS91, GS98, BR02], our method relies solely on nonequispaced FFTs. Uniform error estimates are proven and allow for the adjustment of involved parameters. We easily obtain a matrix formulation and show that the proposed algorithm performs as accurately as the corresponding truncated singular value decomposition but is orders of magnitude faster.

4. The discrete Fourier transform is a unitary operation up to a constant and hence, the inverse is simply given by its scaled adjoint. However, ambiguity arises for nonequispaced sampling nodes where even the number of Fourier coefficients and the number of samples need not coincide. Early approaches for an inversion of the nonequispaced fast Fourier transform suggested to use simply a weighted adjoint NFFT as an approximate inverse, see for example [JMNM91]. In contrast, we provide least squares and interpolation formulations, where the latter are generalised to the sphere and the hyperbolic cross. The convergence of corresponding algorithms is analysed in detail, in particular, we prove rigorous eigenvalue estimates for the involved matrices. Thus, the proposed methods achieve a given target accuracy in a certain number of iterations and hence with proven computational costs.

In summary, we suggest new algorithms that allow for fast matrix vector arithmetic, i.e., the number of floating point operations needed is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log^q N)$ with $0 \leq q \leq 2$ for arbitrary input vectors. All schemes are approximate in the sense that the constant involved in the $\mathcal{O}$-notation depends on the prescribed target accuracy. This dependence is worked out clearly for the proposed algorithms. We contributed to the development of efficient and reliable public software [KP06b, KR06a] which is the commonly accepted basis of reproducible research. Various numerical experiments are presented in order to demonstrate the performance and reliability of our algorithms.

Outline of the thesis

Chapter 2 introduces the spaces of polynomials and various discrete Fourier transforms. Furthermore, we construct localised polynomials, referred to as *kernels*, that prove useful within the subsequent analysis. Localised univariate trigonometric polynomials rely on "smooth" Fourier coefficients, i.e., we prove for compactly supported functions g with derivatives of a given order how the trigonometric polynomial with Fourier coefficients $g(\frac{k}{N})$ decays, see Theorem 2.14 for our main result, [MP00, Thm. 2.2] for a closely related approach, and Corollary

2.16 for an application to sampled B-splines. Generalisations are obtained in Corollary 2.19 for multivariate kernels, in Lemma 2.24 for kernels on the hyperbolic cross, and in Theorem 2.35 for kernels on the sphere.

The following chapter is devoted to the further development and generalisation of the multivariate *nonequispaced fast Fourier transform* (NFFT) [DR93, Bey95, AD96, Ste98, PST01], see Algorithm 3.1. For the univariate case, its aim is the fast computation of

$$f_j = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{f}_k \mathrm{e}^{-2\pi\mathrm{i}kx_j}$$

for given Fourier coefficients $\hat{f}_k \in \mathbb{C}$ and nodes $x_j \in \mathbb{T}$, $j = 0, \ldots, M-1$. The well established NFFT scheme is shown to approximate each individual entry of the nonequispaced Fourier matrix up to a certain accuracy and we prove spectral norm estimates, see Lemma 3.5 and Corollary 3.6. We focus on the actual requirements in terms of computational time and memory usage with respect to the problem size but also with respect to the achieved accuracy. Moreover, we propose the *nonequispaced FFT on the hyperbolic cross*, see Algorithm 3.3, and the *nonequispaced FFT on the sphere*, see Algorithm 3.5. Both algorithms allow for arbitrary sampling geometries, whereas previous approaches in [BD89, Hal92] and [DH94, PST98b, Moh99, ST02, HKMR03, RT06] were restricted to specific evaluation nodes. The chapter is summarised by an overview of the NFFT software library, which also includes a classification of algorithms from the subsequent chapters.

In Chapter 4, we consider the *fast Gauss transform*, see Algorithm 4.1, i.e., the fast computation of

$$g\left(x_j\right) = \sum_{l=0}^{L-1} \alpha_l \mathrm{e}^{-\sigma\|x_j - y_l\|_2^2}$$

for given complex coefficients $\alpha_l \in \mathbb{C}$, source nodes $y_l \in \mathbb{R}$, target nodes $x_j \in \mathbb{R}$, $j = 0, \ldots, M-1$, and a complex parameter $\sigma \in \mathbb{C}$ with positive real part $\mathrm{Re}(\sigma) > 0$. Previous approaches [Str91, GS91, GS98, BR02] relied on the fast multipole method and hence on a specific multipole expansion of the Gaussian kernel and an algorithm with tree-like organisation. In contrast, we prefer an expansion into trigonometric polynomials. Thus, the problem of multiplying with a "generalised convolution matrix" is reduced to the application of two nonequispaced FFTs and the multiplication with a diagonal matrix. The approximation is shown to obey a pointwise and a spectral norm estimate in Theorem 4.7 and Corollary 4.8, respectively. Analogously, we obtain a fast spherical Gauss transform in Algorithm 4.2, where we expand the Gaussian kernel into spherical harmonics and apply the nonequispaced fast Fourier transform on the sphere. The corresponding error estimates are given in Theorem 4.15.

Finally, iterative algorithms for an *inverse NFFT* are proposed in Chapter 5. We construct a trigonometric polynomial f, such that for given samples $(x_j, y_j) \in \mathbb{T} \times \mathbb{C}$, $j = 0, \ldots, M-1$, the approximate identity

$$f\left(x_j\right) \approx y_j$$

is fulfilled. In contrast to the ordinary Fourier matrix, its nonequispaced analogue is in general neither unitary nor square. Hence, we propose Algorithm 5.1 for the solution of a least squares approach and Algorithm 5.2 for computing an optimal interpolation from the given data. Quantitative convergence results within the least squares setting are obtained in Theorem 5.2, cf. [Grö92], and Corollary 5.3. A stable least squares fit with trigonometric polynomials is ensured for sufficiently dense sampling nodes. Convergence of Algorithm 5.2 for the optimal interpolation is proven in Section 5.2.2. The main result is given in Theorem 5.16

for the interpolation with multivariate trigonometric polynomials. Corollary 5.20 reveals the precise condition on the degree of the interpolating trigonometric polynomial with respect to the spatial dimension and the separation distance of the sampling set. Subsequently, Lemma 5.21 and Theorem 5.22 generalise these stability results to the hyperbolic cross and to the sphere, respectively. Stable interpolation by polynomials is possible for sufficiently separated sampling nodes. In summary, we guarantee a stable inverse NFFT under simple conditions on the density of the sampling set and on the degree of the involved polynomial. The chapter closes with an application in magnetic resonance imaging and a recent generalisation for trigonometric polynomials with few non-zero Fourier coefficients.

Parts of this thesis are submitted or accepted for publication. All algorithms were implemented in C or MATLAB and tested on a personal computer (AMD Athlon XP 2700+, 2GB main memory, SuSE-Linux 2.4.20, gcc 3.3, double precision arithmetic) using FFTW3 [FJ] and other software libraries. In particular, a pre-release of the upcoming C subroutine library NFFT3 [KP06b] that generalises and improves on previous versions and a MATLAB toolbox for sparse reconstruction [KR06a] are available.

Acknowledgements

First of all, my sincere thanks go to my family for their constant support and to Susanne for all the love and patience I receive from her. I thank my doctoral adviser Prof. Daniel Potts for his excellent guidance throughout the years. His wonderful ideas and his constant encouragement kept me doing research. Moreover, I want to thank Prof. Jürgen Prestin for his generous support and many discussions that improved my work substantially. In addition, I highly appreciate the participation of Prof. Wolfgang Hackbusch and Prof. Thomas Martinetz in the examination of this thesis.

Large parts of the research presented here was done in cooperations. I want to thank my co-authors: Dr. Markus Fenn [FKP06, FKPar], Jens Keiner [KKP06], Tobias Knopp [KKP05], Dr. Holger Rauhut [KR06b], and Prof. Gabriele Steidl [KPS06]. Following my diploma, I enjoyed my doctoral studies at the Institute of Mathematics, University Lübeck, from September 2003 until April 2005. Subsequently, I spent almost one year at the Numerical Harmonic Analysis Group, University of Vienna. I want to thank Prof. Hans Georg Feichtinger, Prof. Karlheinz Gröchenig, and their nice group for sharing their insight with me. Finally, thanks for code contributions go to Martin Böhme, Bjoern Eskofier, Matt Fulkerson, Steffen Klatt, and Sseziwa Mukasa.

The support of my doctoral studies by the Graduate School 357: Efficient Algorithms and Multiscale Methods and by the German Academic Exchange Service is appreciated.

2

Polynomials and discrete Fourier transforms

In this chapter, we introduce the fundamentals, define the spaces of trigonometric and spherical polynomials and review some basic concepts in Fourier analysis, sampling theory, and discrete Fourier transforms. Moreover, we construct a family of univariate trigonometric polynomials, referred to as kernels, which possess a prescribed decay away from the origin. Theorem 2.14 gives a precise meaning to the rule of thumb that “smooth” Fourier coefficients lead to a localised polynomial. In conjunction with our multivariate generalisations in Corollary 2.19, Lemma 2.24, and Theorem 2.35, these kernels are an essential ingredient to stability estimates in Chapter 5.

2.1 Trigonometric polynomials

For a dimension $d \in \mathbb{N}$ let $\mathbb{T}^d := [-\frac{1}{2}, \frac{1}{2})^d$ be the standard representation of the torus, where opposing sides are identified with each other. Typically, we denote by the vector $\boldsymbol{x} \in \mathbb{T}^d$ a time or spatial node, use the multi index $\boldsymbol{k} \in \mathbb{Z}^d$ to address a frequency location, and abbreviate their inner product $\boldsymbol{k}\boldsymbol{x} := \boldsymbol{k}^\top \boldsymbol{x}$.

Definition 2.1. *We define the classical Fourier transform on Euclidean space $\mathbb{R}^d$ and the semi-discrete Fourier transform as follows.*

1. *For an integrable function, i.e. $f \in L^1(\mathbb{R}^d)$, the Fourier transform is given for $\boldsymbol{k} \in \mathbb{R}^d$ by*
$$\hat{f}\,(\boldsymbol{k}) := \int_{\mathbb{R}^d} f\,(\boldsymbol{x})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\,.$$

2. *For an integrable function on the torus, i.e. $f \in L^1(\mathbb{T}^d)$, the Fourier coefficients are given for $\boldsymbol{k} \in \mathbb{Z}^d$ by*
$$\hat{f}_{\boldsymbol{k}} := \int_{\mathbb{T}^d} f\,(\boldsymbol{x})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} \mathrm{d}\boldsymbol{x}\,.$$

Their classical relation is given by the Poisson summation formula, Theorem 2.3 below, cf. [Zyg93], where we need in addition the notion of bounded variation and focus for simplicity on the univariate case $d = 1$ only.

Definition 2.2. *A function* $g : \mathbb{R} \to \mathbb{R}$ *is said to be of* bounded variation, *denoted by* $g \in BV(\mathbb{R})$, *if*

$$|g|_V := \int_{\mathbb{R}} |\,dg\,(z)| := \sup \sum_{j=0}^{n-1} |g\,(z_{j+1}) - g\,(z_j)| < \infty,$$

where the supremum is taken over all strictly increasing real sequences $\{z_j\}_{j \in \mathbb{N}_0}$. *Functions* $g : \mathbb{T} \to \mathbb{R}$ *are said to be of bounded variation, denoted by* $g \in BV(\mathbb{T})$, *if they have finite variation within the periodic unit interval.*

Theorem 2.3. *[Zyg93, Vol. I, pp. 68] Let* $g \in L^1(\mathbb{R}) \cap BV(\mathbb{R})$ *be given and satisfy* $2g(z) = g(z+0) + g(z-0)$ *for all* $z \in \mathbb{R}$, *then the identity*

$$\sum_{j \in \mathbb{Z}} g\,(j+z) = \sum_{k \in \mathbb{Z}} \hat{g}\,(k)\,\mathrm{e}^{-2\pi \mathrm{i} k z}$$

is fulfilled for all $z \in \mathbb{R}$.

Loosely speaking, we say that the sampling of the Fourier transform $\hat{g}$ at the integers corresponds to the periodisation of g in time domain.

Furthermore, the space of square integrable functions on the torus $L^2(\mathbb{T}^d)$ is a Hilbert space with the usual inner product

$$\langle f, g \rangle_{L^2} := \int_{\mathbb{T}^d} f\,(\boldsymbol{x})\,\overline{g\,(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}\,.$$

This space possesses the orthonormal basis $\{\boldsymbol{x} \mapsto \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} : \boldsymbol{k} \in \mathbb{Z}^d\}$ and thus, every $f \in L^2(\mathbb{T}^d)$ can be expanded into its Fourier series

$$f\,(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}$$

converging with respect to the induced norm $\|\cdot\|_{L^2}$. Furthermore, the square summable Fourier coefficients $(\hat{f}_{\boldsymbol{k}})_{\boldsymbol{k} \in \mathbb{Z}^d} =: \hat{\boldsymbol{f}} \in \ell^2(\mathbb{Z}^d)$, cf. Definition 2.1, satisfy the Parseval identity

$$\|f\|_{L^2}^2 = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \left|\hat{f}_{\boldsymbol{k}}\right|^2 =: \left\|\hat{\boldsymbol{f}}\right\|_2^2\,. \tag{2.1}$$

We obtain the class of trigonometric polynomials by restricting to finitely supported expansions. Let the multi degree $\boldsymbol{N} = (N_0, N_1, \ldots, N_{d-1})^\top \in \mathbb{N}^d$ and the index set for possible frequencies

$$I_{\boldsymbol{N}} := \mathbb{Z}^d \cap \left(\left[-\frac{N_0}{2}, \frac{N_0}{2}\right) \times \ldots \times \left[-\frac{N_{d-1}}{2}, \frac{N_{d-1}}{2}\right)\right)$$

be given. We define the space of d-variate trigonometric polynomials by

$$T_{\boldsymbol{N}} := \operatorname{span}\left\{\boldsymbol{x} \mapsto \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} : \boldsymbol{k} \in I_{\boldsymbol{N}}\right\}\,,$$

and hence, every $f \in T_{\boldsymbol{N}}$ has the unique expansion

$$f\,(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}\,. \tag{2.2}$$

The dimension of this space and hence the total number of Fourier coefficients is $|I_{\boldsymbol{N}}| = N_0 \cdot \ldots \cdot N_{d-1}$ for $\boldsymbol{N} \in \mathbb{N}^d$, where $|\cdot|$ denotes for finite sets their cardinality. Differentiating a trigonometric polynomial term by term and using the Parseval identity, we obtain a simple version of the Bernstein inequality.

Lemma 2.4. *[Zyg93, Vol. II, pp. 11] Let $d = 1$, $N \in \mathbb{N}$, and $f \in T_N$ then*

$$\left\|f'\right\|_{L^2} \le N\pi \left\|f\right\|_{L^2} .$$

One of the key concepts when approximating with trigonometric polynomials is the fact, that the smoother a function is the more rapidly its Fourier coefficients decay. Thus, we obtain the following pointwise estimate for the approximation of reasonable smooth functions by trigonometric polynomials.

Theorem 2.5. *[Zyg93, Vol. I, p. 48] Let $f \in L^1(\mathbb{T}) \cap BV(\mathbb{T})$ be given, then its Fourier coefficients obey $\hat{f}_k = \mathcal{O}(k^{-1})$. In particular, if $f \in L^1(\mathbb{T})$ and $f' \in L^1(\mathbb{T}) \cap BV(\mathbb{T})$, then its truncated Fourier series obeys the uniform estimate*

$$\left| f(x) - \sum_{k \in I_N} \hat{f}_k \mathrm{e}^{-2\pi \mathrm{i} k x} \right| \le \sum_{|k| > \frac{N}{2}} \left|\hat{f}_k\right| \le C_f N^{-1} .$$

We use similar estimates to approximate Gaussian kernels by trigonometric polynomials in Chapter 4. Note however, that weaker conditions on a function f already yield absolute convergence of its Fourier series, cf. [Zyg93, Vol. I, Chapter VI].

2.1.1 Sampling and the discrete Fourier transform

In practical applications we are often confronted with the situation that a function has to be evaluated (or its values are only known) at a finite sampling set

$$\mathcal{X} := \left\{ \boldsymbol{x}_j \in \mathbb{T}^d \,:\, j = 0, \ldots, M-1 \right\} ,$$

whose cardinality is typically denoted by $M \in \mathbb{N}$. The sampling set is called

1. nonequispaced or arbitrary, if the set $\mathcal{X}$ obeys no additional structure and
2. equispaced, regular, or a lattice, if there exists a vector $\boldsymbol{M} \in \mathbb{N}^d$ such that $\mathcal{X} = \boldsymbol{M}^{-1} \odot I_{\boldsymbol{M}}$, where two vectors are linked by the pointwise product $\boldsymbol{a} \odot \boldsymbol{b} := (a_0 b_0, \ldots, a_{d-1} b_{d-1})^\top$ with the inverse $\boldsymbol{a}^{-1} := (\frac{1}{a_0}, \frac{1}{a_1}, \ldots, \frac{1}{a_{d-1}})^\top$.

In the multivariate case $d > 1$ it is possible to construct semi-regular sampling sets like so called line settings, cf. [GS01]. Furthermore, some applications yield particular nonequispaced sampling sets with additional properties within nonstandard coordinates, e.g., polar, spiral, or linogram grids in tomography, see e.g. [PS01a, PS02, KKP05] and their references.

Here, we are highly interested in nonequispaced sampling sets and describe these by simple geometric properties.

Definition 2.6. *Taking periodicity into account, the distance of two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{T}^d$ is given by*

$$\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y}) := \min_{\boldsymbol{j} \in \mathbb{Z}^d} \left\|(\boldsymbol{x} + \boldsymbol{j}) - \boldsymbol{y}\right\|_\infty .$$

The mesh norm *and the* separation distance *of a sampling set $\mathcal{X} \subset \mathbb{T}^d$ are defined by*

$$\begin{aligned} \delta_{\mathcal{X}} &:= \min \left\{ \delta > 0 : \mathbb{T}^d \subset \bigcup_{j=0}^{M-1} \left(\boldsymbol{x}_j + \left[-\frac{\delta}{2}, \frac{\delta}{2} \right]^d \right) \right\} \\ &= 2 \max_{\boldsymbol{x} \in \mathbb{T}^d} \min_{j=0,\ldots,M-1} \operatorname{dist}(\boldsymbol{x}_j, \boldsymbol{x}), \\ q_{\mathcal{X}} &:= \min_{j,l=0,\ldots,M-1; j \ne l} \operatorname{dist}(\boldsymbol{x}_j, \boldsymbol{x}_l) , \end{aligned}$$

respectively. The sampling set $\mathcal{X}$ is called

1. *δ-dense for some $0 < \delta \le 1$, if $\delta_{\mathcal{X}} \le \delta$,*
2. *q-separated for some $0 < q \le \frac{1}{2}$, if $q_{\mathcal{X}} \ge q$.*
3. *A sequence $\{\mathcal{X}_M \subset \mathbb{T}^d : |\mathcal{X}_M| = M\}_{M\in\mathbb{N}}$ is called quasi uniform, if $\delta_{\mathcal{X}_M} \le C q_{\mathcal{X}_M}$ with a constant C independent of M.*

We might interpret the mesh norm and the separation distance as the largest and the smallest gap between neighbouring nodes, respectively. Of course, every δ-dense sampling set is δ'-dense for $\delta' \ge \delta$ and vice versa for the separation distance. Their relation to the cardinality is given in the following lemma.

Lemma 2.7. *Let $d \in \mathbb{N}$, $\mathcal{X} \subset \mathbb{T}^d$ with $M = |\mathcal{X}| \ge 2$ be given, then the relation $q_{\mathcal{X}} \le M^{-1/d} \le \delta_{\mathcal{X}}$ holds true.*

Proof. *Note first, that for $d, M_0 \in \mathbb{N}$ and $\boldsymbol{M} = (M_0, \ldots, M_0)^\top \in \mathbb{N}^d$ the equispaced lattice $\mathcal{X} = \boldsymbol{M}^{-1} \odot I_{\boldsymbol{M}}$ has $M = M_0^d$ nodes and fulfils $q_{\mathcal{X}} = \delta_{\mathcal{X}} = M_0^{-1}$. Now assume that $\delta_{\mathcal{X}} < M^{-1/d}$ and assign to each node its surrounding cube with side length $\delta_{\mathcal{X}}$, then the sum of these volumes is $\sum_{j=0}^{M-1} \delta_{\mathcal{X}}^d < 1$, a contradiction to the covering of $\mathbb{T}^d$ by these boxes. Analogously, we assign boxes of side length $q_{\mathcal{X}} > M^{-1/d}$ to the nodes and thus, $\sum_{j=0}^{M-1} q_{\mathcal{X}}^d > 1$, a contradiction to the fact that these boxes are contained in $\mathbb{T}^d$ and have no common interior points.* □

Fast Fourier transform

For the moment, we summarise the basic knowledge on the sampling of a trigonometric polynomial on a regular grid. Analogously to the Poisson summation formula, one easily obtains an alias formula when sampling a periodic function on a lattice in the following sense.

Theorem 2.8. *Let $f : \mathbb{T}^d \to \mathbb{C}$ have an absolutely convergent Fourier series and let $\boldsymbol{M} \in \mathbb{N}^d$ be given, then the Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ of f obey*

$$|I_{\boldsymbol{M}}|^{-1} \sum_{\boldsymbol{j} \in \boldsymbol{M}^{-1} \odot I_{\boldsymbol{M}}} f(\boldsymbol{j})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{j}} = \hat{f}_{\boldsymbol{k}} + \sum_{\boldsymbol{r} \in \mathbb{Z}^d \setminus \{\boldsymbol{0}\}} \hat{f}_{\boldsymbol{k}+\boldsymbol{M}\odot\boldsymbol{r}} \,. \tag{2.3}$$

In particular, the second term on the right hand side vanishes whenever $f \in T_{\boldsymbol{N}}$, $I_{\boldsymbol{N}} \subset I_{\boldsymbol{M}}$, in which case for $\boldsymbol{f} := (f(\boldsymbol{j}))_{\boldsymbol{j} \in \boldsymbol{M}^{-1}\odot I_{\boldsymbol{M}}}$ the discrete Parseval relation $\|\boldsymbol{f}\|_2 = |I_{\boldsymbol{M}}|\, \|\hat{\boldsymbol{f}}\|_2$ holds.

Proof. *Using the identity*

$$\sum_{\boldsymbol{j} \in \boldsymbol{M}^{-1} \odot I_{\boldsymbol{M}}} \mathrm{e}^{2\pi \mathrm{i}(\boldsymbol{k}-\boldsymbol{l})\boldsymbol{j}} = \begin{cases} |I_{\boldsymbol{M}}|^{-1} & \text{for } \boldsymbol{M}^{-1} \odot (\boldsymbol{k}-\boldsymbol{l}) \in \mathbb{Z}^d\,, \\ 0 & \text{otherwise,} \end{cases}$$

the assertion follows from the substitution of the Fourier series of f into the left hand side of (2.3). □

The left hand side of (2.3) is often referred to as discrete Fourier transform (DFT), mapping samples of a function to its Fourier coefficients, up to the alias. Being a linear map, this transform has a matrix representation, where we use the multi index $\boldsymbol{k}$ to address the elements of vectors and matrices. For clarity of presentation and since the particular order for indices is of minor interest only, we omit plain indices like $k = \sum_{t=0}^{d-1} (k_t + \frac{N_t}{2}) \prod_{t'=t+1}^{d-1} N_{t'}$ whenever possible.

The overwhelming success of this transform relies on the following two facts.

Theorem 2.9. *[Loa92, pp. 3] Let* $\boldsymbol{N} \in \mathbb{N}^d$ *and the* Fourier matrix *be defined by*

$$\boldsymbol{F}_{\boldsymbol{N}} := \left(\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{j}}\right)_{\boldsymbol{j}\in\boldsymbol{N}^{-1}\odot I_{\boldsymbol{N}};\boldsymbol{k}\in I_{\boldsymbol{N}}} .$$

Then the inverse satisfies $\boldsymbol{F}_{\boldsymbol{N}}^{-1} = |I_{\boldsymbol{N}}|^{-1}\boldsymbol{F}_{\boldsymbol{N}}^{\mathsf{H}}$ *and both matrices can be applied to a vector by means of the* fast Fourier transform *(FFT) in* $\mathcal{O}(|I_{\boldsymbol{N}}|\log|I_{\boldsymbol{N}}|)$ *arithmetic operations.*

Hence, the discrete Fourier transform (DFT) and its inverse differ only by a normalisation factor and a sign in the exponent which led to ambiguity in the naming of the DFT and its inverse in literature. Analogous to the documentation of the FFTW library, cf. [FJ], we believe that it is more convenient to call the multiplication with $\boldsymbol{F}_{\boldsymbol{N}}$ the forward transform while having a plus in the exponent is called backward and not inverse.

Nonequispaced DFT

In the more general setting of an arbitrary sampling set, the orthogonality of the Fourier matrix is lost anyway and hence, we subsequently use the term discrete Fourier transform for the evaluation of a trigonometric polynomial at given nodes and reserve the notion of the inverse for solving a linear system to calculate Fourier coefficients from sampled values, cf. Chapter 5.

Given a finite number of Fourier coefficients $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, we want to evaluate the trigonometric polynomial $f \in T_{\boldsymbol{N}}$, i.e.,

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \hat{f}_{\boldsymbol{k}}\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}$$

at given nonequispaced nodes $\boldsymbol{x}_j \in \mathbb{T}^d$, $j = 0,\dots,M-1$. Expressed in terms of a matrix vector product, this reads as

$$\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}} \tag{2.4}$$

where

$$\boldsymbol{A} = \boldsymbol{A}_{\mathcal{X}} := \left(\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}\right)_{j=0,\dots,M-1;\,\boldsymbol{k}\in I_{\boldsymbol{N}}} \tag{2.5}$$

is called the nonequispaced Fourier matrix and the vector of samples and the vector of Fourier coefficients are denoted as before by $\boldsymbol{f} = (f(\boldsymbol{x}_j))_{j=0,\dots,M-1}$ and $\hat{\boldsymbol{f}} = (\hat{f}_{\boldsymbol{k}})_{\boldsymbol{k}\in I_{\boldsymbol{N}}}$, respectively. The straight forward algorithm for this matrix vector product, subsequently denoted by *discrete Fourier transform at nonequispaced nodes* (NDFT) requires $\mathcal{O}(|I_{\boldsymbol{N}}|M)$ operations for $|I_{\boldsymbol{N}}|$ equispaced frequencies $\boldsymbol{k} \in I_{\boldsymbol{N}}$ and M nonequispaced sampling nodes $\boldsymbol{x}_j \in \mathbb{T}^d$. Neglecting the time for the evaluation of the complex exponential, we store no matrix elements at all and obtain the simple Algorithm 2.1.

A related matrix vector product is the *adjoint NDFT*

$$\hat{\boldsymbol{h}} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{f}, \qquad \hat{h}_{\boldsymbol{k}} = \sum_{j=0}^{M-1} f_j\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}\,, \tag{2.6}$$

where the update step in Algorithm 2.1 is simply changed to $\hat{h}_{\boldsymbol{k}}{+}= f_j\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$. Further related transforms are the conjugated NDFT, multiplying with $\overline{\boldsymbol{A}}$, and the transposed NDFT, multiplying with $\boldsymbol{A}^{\top}$, where $\boldsymbol{A}^{\mathsf{H}} = \overline{\boldsymbol{A}}^{\top}$. Chapter 3 provides a thorough introduction to the fast computation of nonequispaced DFTs as well as our recent improvements and generalisations. Note again, that the inversion formula in Theorem 2.9 for the (equispaced and normalised) Fourier matrix $\boldsymbol{F}$ does not hold in the general situation of arbitrary sampling nodes for the matrix $\boldsymbol{A}$. Generalised inverse NDFTs are the topic of Chapter 5, where the following construction of localised trigonometric kernels is used.

Algorithm 2.1 NDFT

```
Input:  $d, M \in \mathbb{N}$, $\boldsymbol{N} \in \mathbb{N}^d$,
        $\boldsymbol{x}_j \in \mathbb{T}^d$, $j = 0, \dots, M-1$, and $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$.

for $j = 0, \dots, M-1$ do
    $f_j = 0$
    for $\boldsymbol{k} \in I_{\boldsymbol{N}}$ do
        $f_j += \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
    end for
end for

Output:      values $f_j = f(\boldsymbol{x}_j)$, $j = 0, \dots, M-1$.
Complexity:  $\mathcal{O}(|I_{\boldsymbol{N}}|M)$.
```

2.1.2 Trigonometric kernels

In the sequel, we define the particular class of trigonometric polynomials with two arguments and positive coefficients, denoted as kernels.

Definition 2.10. *Let $d \in \mathbb{N}$, $\boldsymbol{N} \in \mathbb{N}^d$, and positive weights $\hat{w}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, referred to as damping factors, be given. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{T}^d$, we define the following* kernel

$$K_{\boldsymbol{N}}(\boldsymbol{x},\, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}} \hat{w}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}(\boldsymbol{x}-\boldsymbol{y})}\,.$$

For notational convenience, we abbreviate $K_{\boldsymbol{N}}(\boldsymbol{x}) = K_{\boldsymbol{N}}(\boldsymbol{x},\, \boldsymbol{0})$ and call the kernel normalised *if $K_{\boldsymbol{N}}(\boldsymbol{0}) = \max_{\boldsymbol{x} \in \mathbb{T}^d} K_{\boldsymbol{N}}(\boldsymbol{x}) = 1$. The particular class of* tensor product kernels *is given by $K_{\boldsymbol{N}}(\boldsymbol{x},\, \boldsymbol{y}) = \prod_{t=0}^{d-1} \tilde{K}_{N_t}(x_t,\, y_t)$ where $\tilde{K}_{N_t}$ denote univariate kernels and $\boldsymbol{x} = (x_0, \dots, x_{d-1})^\top$, $\boldsymbol{y} = (y_0, \dots, y_{d-1})^\top$.*

Furthermore, given two sampling sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{T}^d$, we define the matrix

$$\boldsymbol{K}_{\boldsymbol{N}} := \left(K_{\boldsymbol{N}}(\boldsymbol{x}_j, \boldsymbol{y}_l)\right)_{j=0,\dots,M-1;l=0,\dots,L-1}$$

where the special case $\mathcal{X} = \mathcal{Y}$ is simply denoted as kernel matrix.

Lemma 2.11. *The matrix $\boldsymbol{K}_{\boldsymbol{N}}$, cf. Definition 2.10, obeys the factorisation*

$$\boldsymbol{K}_{\boldsymbol{N}} = \boldsymbol{A}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{A}_{\mathcal{Y}}^{\mathsf{H}}$$

with the diagonal matrix $\hat{\boldsymbol{W}} := \operatorname{diag}(\hat{\boldsymbol{w}})$, $\hat{\boldsymbol{w}} = (\hat{w}_{\boldsymbol{k}})_{\boldsymbol{k} \in I_{\boldsymbol{N}}}$. Moreover, the kernel matrix is positive semidefinite and has constant diagonal entries $(\boldsymbol{K}_{\boldsymbol{N}})_{j,j} = 1$, $j = 0, \dots, M-1$, if the kernel $K_{\boldsymbol{N}}$ is normalised.

Proof. The assertions are due to $(\boldsymbol{A}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{A}_{\mathcal{Y}}^{\mathsf{H}})_{j,l} = \sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j} \hat{w}_{\boldsymbol{k}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{y}_l}$. □

These kernels are used for

1. the degenerate approximation of non polynomial kernels, e.g. Gaussians, to obtain fast summation schemes, cf. Chapter 4, and
2. the construction of well conditioned schemes for the polynomial interpolation of scattered data, cf. Chapter 5.

For the latter task, we start from a class of admissible weight functions in Definition 2.12 and construct localised univariate trigonometric kernels in Theorem 2.14, whereas Lemma 2.13 serves as an intermediate step. Similar to the approach in [MP00, Thm. 2.2], we relate the smoothness of the weight function to the decay of the kernel K_N built upon the sampled weights. We would like to highlight that in comparison to Theorem 2.5 the roles of localisation and smoothness are interchanged with respect to the Fourier coefficients and the corresponding Fourier series. Subsequently, Corollary 2.16 specifies the result for kernels constructed from sampled splines.

Definition 2.12. *For $\beta \in \mathbb{N}$, $\beta \geq 2$, a continuous function $g : \mathbb{R} \to \mathbb{R}$ is an* admissible weight function *if it is non-negative and possesses a $(\beta-1)$-fold derivative $g^{(\beta-1)} := g_{(0)}$ of bounded variation, i.e.,*

$$g(z) := g_{(\beta-1)}(z), \quad g_{(\gamma)}(z) := \int_{-\frac{1}{2}}^{x} g_{(\gamma-1)}(x)\,\mathrm{d}x, \quad \gamma = 1, \ldots, \beta-1,$$

and $g_{(0)} \in BV(\mathbb{R})$ with the additional properties $\operatorname{supp} g_{(0)} = [-\frac{1}{2}, \frac{1}{2}]$, *$g_{(\gamma)}(\pm\frac{1}{2}) = 0$ for $\gamma = 0, \ldots, \beta-1$, $g(z) > 0$, $|z| < \frac{1}{2}$, and the normalisation $\|g\|_{L^1} = 1$. We denote by $BV_0^{\beta-1}$ the set of admissible weight functions.*

Furthermore, we define for notational convenience the zeta function $\zeta(\beta) := \sum_{r=1}^{\infty} r^{-\beta}$, *$\beta > 1$, and for $g \in BV_0^{\beta-1}$ the* norm of the samples

$$\|g\|_{1,N} := \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} g\left(\frac{k}{N}\right).$$

Lemma 2.13. *For $\beta \in \mathbb{N}$, $\beta \geq 2$, let a function $g \in BV_0^{\beta-1}$ be given. Then for $N \in 2\mathbb{N}$, $N \geq 2\beta$, and $x \in \mathbb{T} \setminus \{0\}$ the following estimates hold true*

$$\begin{aligned} \left| \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} g\left(\frac{k}{N}\right) \mathrm{e}^{-2\pi \mathrm{i} k x} \right| &\leq \frac{\left(2^{\beta}-1\right) \zeta(\beta) \left|g^{(\beta-1)}\right|_{V}}{(2N)^{\beta-1} |2\pi x|^{\beta}}, \\ \|g\|_{1,N} &\geq N\left(1 - 2\zeta(\beta)(4\pi\beta)^{-\beta} \left|g^{(\beta-1)}\right|_{V}\right). \end{aligned}$$

Proof. First, we define for $x, z \in \mathbb{T}$ the function $h_x(z) := g(z)\,\mathrm{e}^{-2\pi \mathrm{i} N x z}$. Thus, the Poisson summation formula, cf. Theorem 2.3, yields

$$\frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} g\left(\frac{k}{N}\right) \mathrm{e}^{-2\pi \mathrm{i} k x} = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} h_x\left(\frac{k}{N}\right) = \sum_{r \in \mathbb{Z}} \int_{\mathbb{T}} h_x(z)\,\mathrm{e}^{2\pi \mathrm{i} N r z}\,dz$$

and by applying integration by parts and the fact that $g^{(\tilde{\beta})}\left(\pm\frac{1}{2}\right) = 0$ for $\tilde{\beta} = 0, \ldots, \beta-1$

further

$$\begin{aligned}\left|\sum_{k=-\frac{N}{2}}^{\frac{N}{2}} h_x\left(\frac{k}{N}\right)\right| &= \left|N\sum_{r\in\mathbb{Z}}\int_{\mathbb{T}} g\left(z\right)\mathrm{e}^{-2\pi\mathrm{i}Nxz}\mathrm{e}^{2\pi\mathrm{i}Nrz}\,dz\right| \\ &= \left|N\sum_{r\in\mathbb{Z}}\left(2\pi\mathrm{i}N\left(x-r\right)\right)^{-(\beta-1)}\int_{\mathbb{T}} g^{(\beta-1)}\left(z\right)\mathrm{e}^{-2\pi\mathrm{i}Nz(x-r)}\,dz\right| \\ &\leq \frac{1+|x|^{\beta}\sum\limits_{r\in\mathbb{Z}\setminus\{0\}}|r-x|^{-\beta}}{\left(2\pi\right)^{\beta}N^{\beta-1}|x|^{\beta}}\sup_{r_0\in\mathbb{Z}}\left|\int_{\mathbb{T}} g^{(\beta-1)}\left(z\right)\left(\frac{\mathrm{d}}{\mathrm{d}z}\mathrm{e}^{-2\pi\mathrm{i}Nz(x-r_0)}\right)\,dz\right|.\end{aligned}$$

Using $1+|x|^{\beta}\sum_{r\in\mathbb{Z}\setminus\{0\}}|r-x|^{-\beta}\leq(2^{\beta}-1)2^{1-\beta}\zeta(\beta)$ *for* $|x|\leq\frac{1}{2}$ *and*

$$\left|\int_{\mathbb{T}} g^{(\beta-1)}\left(z\right)\left(\frac{\mathrm{d}}{\mathrm{d}z}\mathrm{e}^{-2\pi\mathrm{i}Nz(x-r)}\right)\,dz\right|\leq\left|g^{(\beta-1)}\right|_{V}$$

yields the assertion.

By the Poisson summation formula, we note furthermore that

$$\frac{1}{N}\left\|g\right\|_{1,N}\geq 1-\left|\sum_{r\in\mathbb{Z}\setminus\{0\}}\int_{\mathbb{T}} g\left(z\right)\mathrm{e}^{2\pi\mathrm{i}Nrz}\,dz\right|$$

and proceed analogously in order to prove the second assertion where we use $N\geq 2\beta$ *to obtain an estimate of the right hand side which is independent of* N. □

Hence, we have collected the necessary tools to prove our main result on localised trigonometric kernels.

Theorem 2.14. *For* $\beta\in\mathbb{N}$, $\beta\geq 2$, *let a function* $g\in BV_0^{\beta-1}$ *be given. Furthermore, let* $N\in 2\mathbb{N}$, $N\geq 2\beta$, *and the damping factors*

$$\hat{w}_k=\frac{g\left(\frac{k}{N}\right)+g\left(\frac{k+1}{N}\right)}{2\left\|g\right\|_{1,N}},\qquad k=-\frac{N}{2},\ldots,\frac{N}{2}-1,$$

be given. Then the kernel K_N, *cf. Definition 2.10, is normalised and fulfils*

$$\left|K_N\left(x\right)\right|\leq\frac{\left(2^{\beta}-1\right)\zeta\left(\beta\right)\left|g^{(\beta-1)}\right|_{V}}{2^{\beta-1}\left(2\pi\right)^{\beta}-\zeta\left(\beta\right)\beta^{-\beta}\left|g^{(\beta-1)}\right|_{V}}\frac{1}{N^{\beta}|x|^{\beta}}$$

for $x\in\mathbb{T}\setminus\{0\}$ *whenever the denominator is positive, i.e.* $|g^{(\beta-1)}|_V<\frac{4\pi\beta)^{\beta}}{2\zeta(\beta)}$.

Proof. Note first, that

$$K_N\left(x\right)=\frac{1+\mathrm{e}^{2\pi\mathrm{i}x}}{2\left\|g\right\|_{1,N}}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}}g\left(\frac{k}{N}\right)\mathrm{e}^{-2\pi\mathrm{i}kx}.$$

Thus, we obtain $K_N(0)=1$ *and by applying Lemma 2.13 also the decay property.* □

We apply Theorem 2.14 in the following to the particular class of B-spline kernels. Note that the order β of the B-spline and the degree $N \in 2\mathbb{N}$ of the kernel, we built upon the samples of the B-spline, are independent of each other, whereas [NSW98] considers the special case $\beta = N$. Moreover, $\beta = 2$, i.e., the hat function $g_2(z) = 2 - 4|z|$ for $|z| \leq \frac{1}{2}$ and $g_2(z) = 0$ elsewhere, leads to the well known *Fejér kernel.* Under additional conditions our localisation result is slightly improved in Remark 2.17.

Definition 2.15. *Let $\beta \in \mathbb{N}$ be given. The* normalised B-spline *is defined by*

$$g_\beta\left(z\right) := \beta N_\beta \left(\beta z + \frac{\beta}{2}\right) ,$$

where N_β denotes the cardinal B-spline of order β. The cardinal B-splines are given by $N_1(z) = 1$ for $z \in (0,1)$, $N_1(z) = 0$ elsewhere, and $N_{\beta+1}(z) = \int_{z-1}^{z} N_\beta(\tau)\,d\tau$ for $\beta \geq 1$, see e.g. [Chu88, Chapter 1]. Furthermore, we define for $\beta \geq 2$ and $N \in 2\mathbb{N}$ the B-spline kernel *by*

$$B_{\beta,N}\left(x\right) := \frac{1 + \mathrm{e}^{2\pi\mathrm{i}x}}{2\left\|g_\beta\right\|_{1,N}} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} g_\beta\left(\frac{k}{N}\right) \mathrm{e}^{-2\pi\mathrm{i}kx} .$$

Corollary 2.16. Let $\beta \in \mathbb{N}$, $\beta \geq 2$, and $N \in 2\mathbb{N}$, $N \geq 2\beta$, be given. Then the B-spline kernel $B_{\beta,N}(x)$, cf. Definition 2.15, fulfils

$$\left|B_{\beta,N}\left(x\right)\right| \leq \frac{\left(2^\beta - 1\right)\zeta\left(\beta\right)\beta^\beta}{2^{\beta-1}\pi^\beta - \zeta\left(\beta\right)} \left|Nx\right|^{-\beta}$$

for $x \in \mathbb{T} \setminus \{0\}$ where $B_{\beta,N}(0) = 1$.

Proof. *Note that $g_\beta \in BV_0^{\beta-1}$. Using $N'_\beta(z) = N_{\beta-1}(z) - N_{\beta-1}(z-1)$, we conclude*

$$\left|g_\beta^{(\beta-1)}\right|_V = \beta^\beta \left|N_\beta^{(\beta-1)}\right|_V = \beta^\beta \left|\sum_{\tau=0}^{\beta-1} (-1)^\tau \binom{\beta-1}{\tau} N_1\left(\cdot - \tau\right)\right|_V = (2\beta)^\beta$$

and apply Theorem 2.14. ☐

Remark 2.17. *If we assume in Corollary 2.16 furthermore, that $N = \beta\sigma$, $\sigma \in \mathbb{N}$, then the stronger estimate $|B_{\beta,N}(x)| \leq 2\zeta(\beta)(1 - 2^{-\beta})(\frac{\beta}{\pi})^\beta |Nx|^{-\beta}$ holds true. This improvement is due to $\|g_\beta\|_{1,N} = N$ in Lemma 2.13 and follows from the partition of unity of the cardinal B-spline N_β and the refinement equation $N_\beta(z) = \sum_{\tau\in\mathbb{Z}} a_\tau^{(\beta,\sigma)} N_\beta(\sigma z - \tau)$ for some finitely supported coefficients $a_\tau^{(\beta,\sigma)} > 0$, see e.g. [Chu88, pp. 8].*

In particular, the Fejér kernel fulfils $|B_{2,N}(x)| \leq |Nx|^{-2}$, $N \in 2\mathbb{N}$, which also follows from the representation

$$B_{2,N}\left(x\right) = \frac{2\left(1 + \mathrm{e}^{2\pi\mathrm{i}x}\right)}{N^2} \left(\frac{\sin\left(\frac{N}{2}\pi x\right)}{\sin\left(\pi x\right)}\right)^2$$

and $\sin(\pi x) \geq 2x$ *for* $0 \leq x \leq \frac{1}{2}$.

In addition, we present some classical kernels and their decay properties in the following example.

Example 2.18. The *Dirichlet kernel* is given for $N \in 2\mathbb{N}$ by

$$D_N(x) := \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \mathrm{e}^{-2\pi\mathrm{i}kx} = \frac{\mathrm{e}^{\pi\mathrm{i}x} \sin(N\pi x)}{N \sin(\pi x)},$$

it fulfils $|D_N(x)| \leq |Nx|^{-1}$ for $x \in \mathbb{T} \setminus \{0\}$ and $D_N(0) = 1$.

Besides the B-spline kernel, the β-th *Jackson kernel* is another generalisation of the Fejér kernel. We define for $\beta \in 2\mathbb{N}$, $\sigma \in \mathbb{N}$, and $N = \beta(\sigma - 1) + 2$ the Jackson kernel by

$$J_{\beta,N}(x) := \frac{1 + \mathrm{e}^{2\pi\mathrm{i}x}}{2\sigma^\beta} \left(\frac{\sin(\sigma\pi x)}{\sin(\pi x)} \right)^\beta .$$

Being a normalised power of the Fejér kernel $J_{2,2\sigma} = B_{2,2\sigma}$, the coefficients $\hat{w}_k$ to the Jackson kernel can be obtained by an iterated discrete convolution of the coefficients of this Fejér kernel, see [Ale93] for details. The kernel is localised as $|J_{\beta,N}(x)| \leq (\frac{\beta}{2})^\beta |Nx|^{-\beta}$ for $x \in \mathbb{T}\setminus\{0\}$ where $J_{\beta,N}(0) = 1$.

It is well known that the Sobolev norm is just a weighted norm on the Fourier coefficients, i.e., $\|f\|_{L^2}^2 + \|f^{(\alpha)}\|_{L^2}^2 = \sum_{k \in I_N} (1 + |2\pi k|^{2\alpha}) |\hat{f}_k|^2$. For $\beta \in \mathbb{N}$ and $\alpha, \gamma > 0$, a regularised and slightly generalised weight is given by

$$g_{\alpha,\beta,\gamma}(z) := c_{\alpha,\beta,\gamma} \frac{\left(\frac{1}{4} - z^2\right)^\beta}{\gamma + |z|^{2\alpha}}$$

for $|z| \leq \frac{1}{2}$ and $g_{\alpha,\beta,\gamma}(z) = 0$ elsewhere, whereas the constant $c_{\alpha,\beta,\gamma}$ is chosen such that $\|g_{\alpha,\beta,\gamma}\|_{L^1} = 1$. Here, the denominator generalises the weight $1 + (2\pi k)^{2\alpha}$ and the nominator ensures $g_{\alpha,\beta,\gamma} \in BV_0^{\beta-1}$. We define for $N \in 2\mathbb{N}$ the Sobolev kernel by

$$S_{\alpha,\beta,\gamma,N}(x) := \frac{1 + \mathrm{e}^{2\pi\mathrm{i}x}}{2\|g_{\alpha,\beta,\gamma}\|_{1,N}} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} g_{\alpha,\beta,\gamma}\left(\frac{k}{N}\right) \mathrm{e}^{-2\pi\mathrm{i}kx} .$$

The kernel is localised as $|S_{\alpha,\beta,\gamma,N}(x)| \leq \tilde{c}_{\alpha,\beta,\gamma} |Nx|^{-\beta}$ for $x \in \mathbb{T} \setminus \{0\}$ and some constant $\tilde{c}_{\alpha,\beta,\gamma} > 0$ where $S_{\alpha,\beta,\gamma,N}(0) = 1$.

The kernels in Example 2.18 are shown in Figure 2.1. Furthermore, we illustrate the estimates made for the B-spline kernel and some comparison in Figure 2.2. Note, that the B-spline kernel $B_{4,80}$ is slightly better localised than its counterpart Jackson kernel $J_{4,78}$ (right).

Multivariate kernels

We complete this section by an extension of our result to the multivariate case $d > 1$. Indeed, tensor products of the kernels constructed in Theorem 2.14 yield also localised multivariate kernels as shown in the following corollary.

Corollary 2.19. Let the normalised univariate kernel $\tilde{K}_N$, cf. Definition 2.10, fulfil for some $\beta \in \mathbb{N}$, some constant $C_\beta > 1$, and all $x \in \mathbb{T} \setminus \{0\}$ the decay condition $|\tilde{K}_N(x)| \leq C_\beta |Nx|^{-\beta}$. Then the product kernel $K_{\boldsymbol{N}}(\boldsymbol{x}) = \prod_{t=0}^{d-1} \tilde{K}_N(x_t)$, $\boldsymbol{N} = (N, \ldots, N)^\top$, $\boldsymbol{x} = (x_0, \ldots, x_{d-1})^\top$, fulfils

$$|K_{\boldsymbol{N}}(\boldsymbol{x})| \leq \frac{C_\beta}{N^\beta \|\boldsymbol{x}\|_\infty^\beta}$$

for $\boldsymbol{x} \in \mathbb{T}^d \setminus \{\boldsymbol{0}\}$, with the same order of decay β and constant C_β.

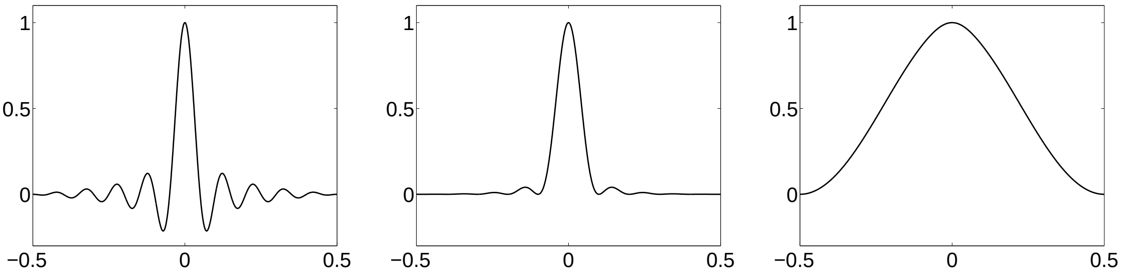


Figure 2.1: From left to right: Real part of the Dirichlet kernel D_{20}, the Fejér kernel $B_{2,20}$, and the Sobolev kernel $S_{1,2,10^{-4},20}$.

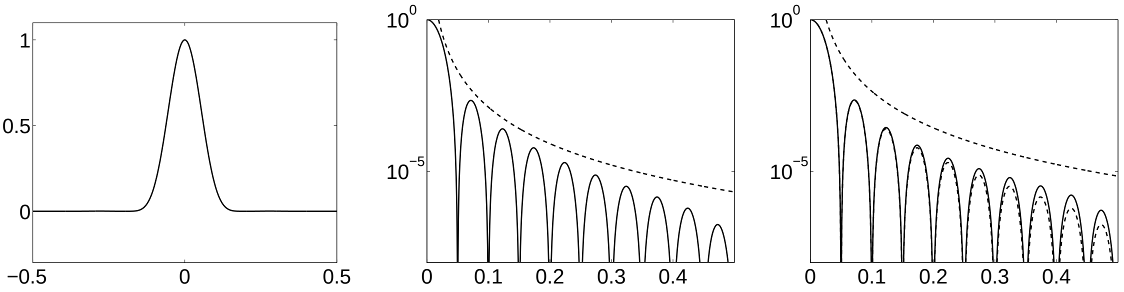


Figure 2.2: Left: Real part of the B-spline kernel $B_{4,20}$. Middle: Log plot of the modulus $|B_{4,80}|$ (solid) and the estimate by Corollary 2.16 (dashed). Right: Log plot of the modulus Jackson kernel $|J_{4,78}|$ (solid) , its decay estimate, cf. Example 2.18 (dashed), and the slightly better localised B-spline kernel $|B_{4,80}|$ (dashed).

Proof. Let $t \in \{0,\dots,d-1\}$ be such that $|x_t| = \|\boldsymbol{x}\|_\infty$. We estimate all factors except the t-th of the tensor product kernel uniformly by one. Hence, the assertion follows from $|K_N(\boldsymbol{x})| \le |\tilde{K}_N(|x_t|)|$ and the localisation property of the univariate kernel. ☐

2.2 Trigonometric polynomials on the hyperbolic cross

In multivariate approximation one has to deal with the so called curse of dimensionality, i.e., the number of degrees of freedom for representing an approximation with a prescribed accuracy depends exponentially on the space dimension of the considered problem. This obstacle can be circumvented to some extent by the interpolation on sparse grids and the related approximation on hyperbolic cross points in the Fourier domain, see e.g. [Zen91, Spr00, BG04b]. For $d \in \mathbb{N}$, $J \in \mathbb{N}_0$, and $N = 2^J$, let the hyperbolic cross be given by

$$H_N^d := \bigcup_{\boldsymbol{N}\in\mathbb{N}^d,\, |I_{\boldsymbol{N}}|=N} I_{\boldsymbol{N}}\,.$$

Obviously, $H_N^d \subset I_{(N,\dots,N)^\top}$ and we define the space of d-variate trigonometric polynomials on the hyperbolic cross $T_{N,d}^{\mathcal{H}}$ by

$$T_{N,d}^{\mathcal{H}} := \operatorname{span}\left\{\boldsymbol{x} \mapsto \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}} \,:\, \boldsymbol{k} \in H_N^d\right\}\,.$$

The dimension of this space and hence the total number of Fourier coefficients is of order $\mathcal{O}(N \log^{d-1} N)$ since there exist at most J^{d-1} possibilities to build a set $I_{\boldsymbol{N}}$, $|I_{\boldsymbol{N}}| = 2^J$.

2.2.1 Sampling and the sparse discrete Fourier transform

Starting from the observation, that the degrees of freedom of a polynomial $f \in T_{N,d}^{\mathcal{H}}$ increase only slightly with the spatial dimension d, the geometry of an arbitrary sampling might mimic an univariate setting and be described in the following way.

Definition 2.20. *A sampling set $\mathcal{X} \subset \mathbb{T}^d$ is called* q^*-separated*, if for every coordinate $t = 0, \dots, d-1$ the projection $\mathcal{X}_t = \{(\boldsymbol{x}_j)_t : j = 0, \dots, M-1\} \subset \mathbb{T}$ is q-separated and fulfils $|\mathcal{X}_t| = |\mathcal{X}|$.*

The second condition ensures that two nodes in the original sampling set do not coincide in a projection. Note furthermore, that the definition of a density of the sampling set is less obvious. However, a natural choice for a sampling lattice with respect to the hyperbolic cross is given by the sparse grid

$$\mathcal{H}_N^d := \bigcup_{\boldsymbol{N}\in\mathbb{N}^d,\, |I_{\boldsymbol{N}}|=N} \boldsymbol{N}^{-1} \odot I_{\boldsymbol{N}}$$

which is of course a subset of the regular lattice $N^{-1} I_{(N,\dots,N)^\top}$. While using only a small fraction of the nodes on the full grid, the interpolation order deteriorates only by a factor $\log^{d-1} N$ for functions with higher order mixed smoothness. More explicitely, we cite for the bivariate case $d = 2$ a result for the interpolation error. For details concerning the interpolation operators as well as the shown estimates, we refer the interested reader to [Spr97, Spr00].

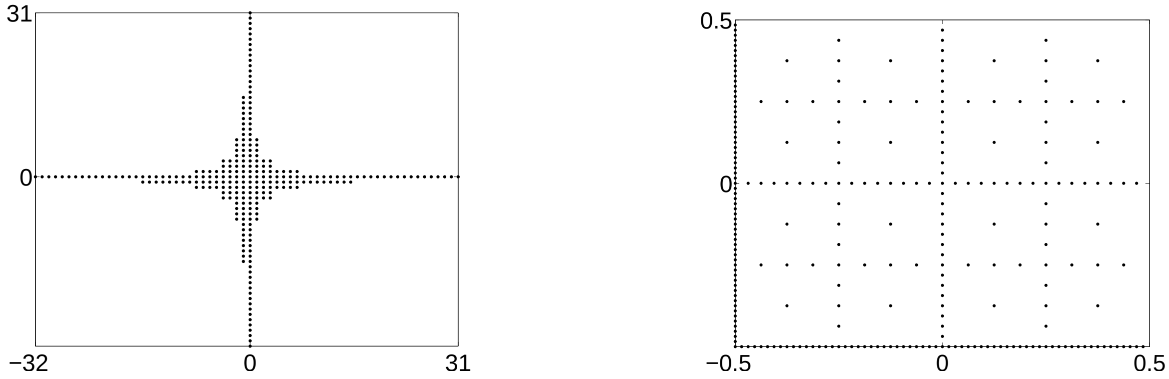


Figure 2.3: Hyperbolic cross H_{64}^2 (left) and the corresponding sparse grid $\mathcal{H}_{64}^2$ (right).

Definition 2.21. *Let $d \in \mathbb{N}$, $\boldsymbol{N} \in \mathbb{N}^d$, and a continuous $f : \mathbb{T}^d \to \mathbb{C}$ be given. We define the interpolation operator $\mathcal{L}_{\boldsymbol{N}}$ on the full grid by*

$$\mathcal{L}_{\boldsymbol{N}} f := |I_N|^{-1} \sum_{\boldsymbol{j}\in\boldsymbol{N}^{-1}\odot I_{\boldsymbol{N}}} f(\boldsymbol{j}) \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}(\cdot-\boldsymbol{j})} .$$

Furthermore, we define for $d = 2$, $N = 2^J$, $J \in \mathbb{N}_0$, the bivariate interpolation operator on the sparse grid

$$\mathcal{B}_N := \sum_{r=0}^{J} \mathcal{L}_{(2^r, 2^{J-r})} - \sum_{r=0}^{J-1} \mathcal{L}_{(2^r, 2^{J-1-r})} .$$

Theorem 2.22. *[Spr97, Cor. 2.23, Thm. 2.24] Let $\alpha > 1$, $N \in \mathbb{N}$, $\boldsymbol{N} = (N, N)^\top$, and $f \in H^\alpha(\mathbb{T}^2)$ with finite* Sobolev norm $\|f\|_{H^\alpha}^2 := \sum_{\boldsymbol{k}\in\mathbb{Z}^2}(1 + \|\boldsymbol{k}\|_2^2\|)^{\alpha/2}|\hat{f}_{\boldsymbol{k}}|^2$ *be given, then the trigonometric polynomial interpolating at the full grid obeys*

$$\|f - \mathcal{L}_{\boldsymbol{N}} f\|_{L^2} \le C_{f,\alpha} N^{-\alpha} \|f\|_{H^\alpha} .$$

Let $\alpha > 1$, $J \in \mathbb{N}$, $N = 2^J$, and $f \in S^\alpha(\mathbb{T}^2)$ with finite mixed Sobolev norm $\|f\|_{S^\alpha}^2 := \sum_{\boldsymbol{k}\in\mathbb{Z}^2}(1 + |k_0|^2)^{\alpha/2}(1 + |k_1|^2)^{\alpha/2}|\hat{f}_{\boldsymbol{k}}|^2$ *be given, then the trigonometric polynomial interpolating at the sparse grid obeys*

$$\|f - \mathcal{B}_N f\|_{L^2} \le \tilde{C}_{f,\alpha} N^{-\alpha} \log N \|f\|_{S^\alpha} .$$

Here, we are interested in the actual computation of such operators. Being more general, our aim is an algorithm for the *nonequispaced sparse discrete Fourier transform* (NSDFT) and its adjoint. We want to evaluate trigonometric polynomials on the hyperbolic cross $f \in T_{N,d}^{\mathcal{H}}$, i.e.,

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in H_N^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}$$

at the nodes of a general sampling set $\mathcal{X} \subset \mathbb{T}^d$. Given a vector of Fourier coefficients $\hat{\boldsymbol{f}} \in \mathbb{C}^{|H_N^d|}$ and the nonequispaced sparse Fourier matrix

$$\boldsymbol{Z} = \boldsymbol{Z}_{\mathcal{X}} := \left(\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}\right)_{j=0,\dots,M;\boldsymbol{k}\in H_N^d}$$

we are interested in the computation of

$$\boldsymbol{f} = \boldsymbol{Z}\hat{\boldsymbol{f}} . \tag{2.7}$$

Obviously, the straightforward method takes $\mathcal{O}(|H_N^d|M)$ operations. Fast algorithms for such a transform are developed in Chapter 3. Chapter 5 comments on the convergence of an inverse NSDFT reconstructing $f \in T_{N,d}^{\mathcal{H}}$ from its sampled values.

2.2.2 Kernels on the hyperbolic cross

For a reconstruction of $f \in T_{N,d}^{\mathcal{H}}$ from its samples, more precisely, for the interpolation of the samples, we construct localised kernels that allow for a stable interpolation scheme.

Definition 2.23. *Let $d \in \mathbb{N}$, $J \in \mathbb{N}_0$, $N = 2^J$, and damping factors $\hat{w}_{\boldsymbol{k}}^{\mathcal{H}} > 0$, $\boldsymbol{k} \in H_N^d$, be given. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{T}^d$, we define the* kernel on the hyperbolic cross

$$K_N^{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}) = K_N^{\mathcal{H}}(\boldsymbol{x} - \boldsymbol{y}) := \sum_{\boldsymbol{k}\in H_N^d} \hat{w}_{\boldsymbol{k}}^{\mathcal{H}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}(\boldsymbol{x}-\boldsymbol{y})} .$$

Similar to Lemma 2.11, the matrix with entries $(\boldsymbol{K}_N^{\mathcal{H}})_{j,l} := K_N^{\mathcal{H}}(\boldsymbol{x}_j, \boldsymbol{y}_l)$ obeys the factorisation $\boldsymbol{K}_N^{\mathcal{H}} = \boldsymbol{Z}_{\mathcal{X}}\hat{\boldsymbol{W}}\boldsymbol{Z}_{\mathcal{Y}}^{\mathsf{H}}$ for the diagonal matrix $\hat{\boldsymbol{W}}$ with entries $(\hat{\boldsymbol{W}})_{\boldsymbol{k},\boldsymbol{k}} = \hat{w}_{\boldsymbol{k}}^{\mathcal{H}}$.

Lemma 2.24. *Let $J \in \mathbb{N}_0$ and the family of normalised univariate kernels K_N, $N = 2^r$, $r = 0, \dots, J$, cf. Definition 2.10, obey the localisation $|K_N(x)| \le C_\beta |Nx|^{-\beta}$ for $x \in \mathbb{T} \setminus \{0\}$ and some $\beta > 0$. Then, for $d = 2$, $\boldsymbol{x} = (x_0, x_1)^\top$, $N = 2^J \ge 4$, the corresponding bivariate kernel on the hyperbolic cross*

$$K_N^{\mathcal{H}}(\boldsymbol{x}) = \frac{K_N(x_0) + N^{-\beta}C_\beta^{-1}(J-1)^{-1}\sum_{r=1}^{J-1} K_{2^r}(x_0) K_{2^{J-r}}(x_1) + K_N(x_1)}{2 + N^{-\beta}C_\beta^{-1}}$$

is normalised and obeys for $\boldsymbol{x} \in (\mathbb{T} \setminus \{0\})^2$ the localisation property

$$\left|K_N^{\mathcal{H}}(\boldsymbol{x})\right| \le \frac{C_\beta}{2N^\beta}\left(|x_0|^{-\beta} + |Nx_0x_1|^{-\beta} + |x_1|^{-\beta}\right) .$$

Proof. *Of course, the normalisation* $K_N^{\mathcal{H}}(\mathbf{0}) = 1$ *holds true and since* $N^{-\beta}C_\beta^{-1} \geq 0$*, the estimate for the first and for the last term easily follows from the localisation of* K_N*. We refine Corollary 2.19 to obtain for each individual summand* $r = 1, \ldots, J-1$ *and* $x_0, x_1 \in \mathbb{T} \setminus \{0\}$ *the inequality*

$$|K_{2^r}(x_0)\, K_{2^{J-r}}(x_1)| \leq C_\beta \left|2^r x_0\right|^{-\beta} \cdot C_\beta \left|2^{J-r} x_1\right|^{-\beta}$$

what finally proves the assertion. □

Hence, the Fourier coefficients of $K_N^{\mathcal{H}}$ are obtained by summing the coefficients of the univariate kernel K_N and its bivariate version on the hyperbolic cross, in particular, these coefficients are positive. In Chapter 5, the stability of an interpolation scheme relies on the constructed kernel in conjunction with an q^*-separated sampling set.

2.3 Polynomials on the sphere

Let $\mathbb{S}^2 := \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 = 1\}$ define the two dimensional unit *sphere*. A point $\boldsymbol{\xi} \in \mathbb{S}^2$ is identified with spherical coordinates $(\vartheta, \varphi)^\top \in [0, \pi] \times [-\pi, \pi)$ by $\boldsymbol{\xi} = (\sin\vartheta\cos\varphi, \sin\vartheta\sin\varphi, \cos\vartheta)^\top$, where the angles ϑ, φ are the longitude and latitude of that point. We abbreviate for notational convenience $\boldsymbol{\xi} \sim (\vartheta, \varphi)$.

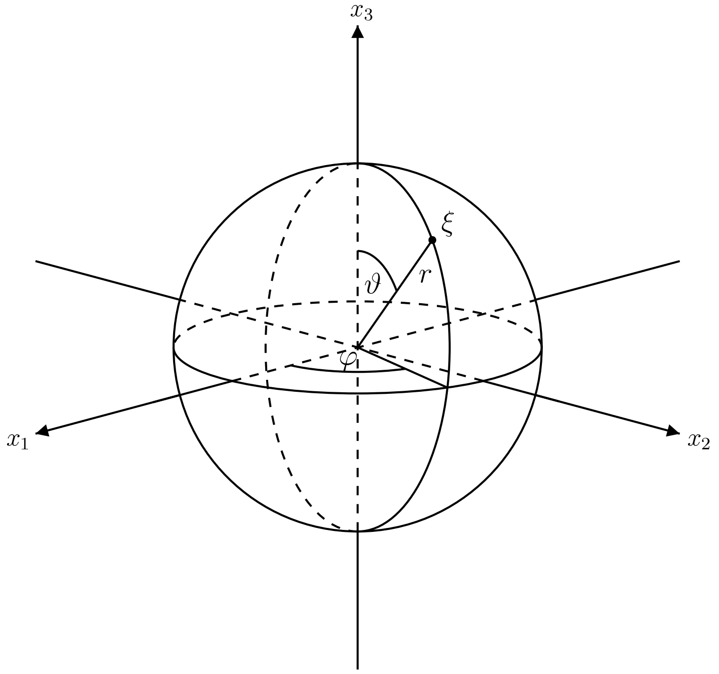


Figure 2.4: Two dimensional unit sphere.

Analogous to the complex exponentials on the torus, the spherical harmonics are the key to the Fourier analysis on the sphere. Moreover, we define the Legendre polynomials, their associated functions, cite a couple of useful properties of these functions, and define the Fourier transform on the sphere.

Definition 2.25. *The* Legendre polynomials $P_k : [-1, 1] \to \mathbb{R}$, $k \in \mathbb{N}_0$, *are given by the Rodrigues formula*

$$P_k(x) := \frac{1}{2^k k!} \frac{\mathrm{d}^k}{\mathrm{d}x^k} \left(x^2 - 1\right)^k.$$

Their associated Legendre functions *$P_k^n : [-1,1] \to \mathbb{R}$, k, $n \in \mathbb{N}_0$, $n \leq k$, are defined by*

$$P_k^n(x) := \sqrt{\frac{(k-n)!}{(k+n)!}} \left(1-x^2\right)^{\frac{n}{2}} \frac{\mathrm{d}^n}{\mathrm{d}x^n} P_k(x) \,.$$

Each integrable $f \in L^1([-1,1])$ possesses Fourier-Legendre coefficients, given for $k \in \mathbb{N}_0$ by

$$\hat{f}_k := \int_{-1}^{1} f(x)\, P_k(x)\, \mathrm{d}x \,.$$

Lemma 2.26. *[AS72, pp. 331, pp. 773] Let $k \in \mathbb{N}_0$ and for notational convenience $P_{-1}(x) := 0$. The Legendre polynomials obey*

$$P_k(\pm 1) = (\pm 1)^k \,, \qquad \max_{x \in [-1,1]} |P_k(x)| = 1 \,.$$

Furthermore, two recurrence relations are given by

$$(k+1)\, P_{k+1}(x) = (2k+1)\, x P_k(x) - k P_{k-1}(x)$$

and

$$(2k+1)\, P_k(x) = P_{k+1}'(x) - P_{k-1}'(x) \,,$$

respectively. The associated Legendre functions form for fixed order $n \in \mathbb{N}_0$ and degrees $k \geq n$ a complete set of orthogonal functions with respect to

$$\langle f, g \rangle_{L^2([-1,1])} := \int_{-1}^{1} f(x)\, g(x)\, \mathrm{d}x \,.$$

Definition 2.27. *The* spherical harmonics *$Y_k^n : \mathbb{S}^2 \to \mathbb{C}$, $\boldsymbol{\xi} \in \mathbb{S}^2$, $\boldsymbol{\xi} \sim (\vartheta,\, \varphi)^\top$, of degree $k \in \mathbb{N}_0$ and order $n \in \mathbb{Z}$, $|n| \leq k$ are given by*

$$Y_k^n(\vartheta,\, \varphi) := \sqrt{\frac{2k+1}{4\pi}} P_k^{|n|}(\cos\vartheta) \mathrm{e}^{\mathrm{i}n\varphi} \,.$$

We abbreviate the pair degree/order by using a multi index $\boldsymbol{k}$ from the triangular index set $J_\infty := \{(k,n)^\top \in \mathbb{N}_0 \times \mathbb{Z} : |n| \leq k\}$ and readily define the Fourier coefficients of an integrable function $f \in L^1(\mathbb{S}^2)$ for $\boldsymbol{k} \in J_\infty$ by

$$\hat{f}_{\boldsymbol{k}} := \int_{\mathbb{S}^2} f(\boldsymbol{\xi})\, \overline{Y_{\boldsymbol{k}}(\boldsymbol{\xi})} \mathrm{d}\mu(\boldsymbol{\xi}) \,,$$

where $\mathrm{d}\mu(\boldsymbol{\xi}) = \sin\vartheta \mathrm{d}\vartheta \mathrm{d}\varphi$ denotes the surface element and we abbreviate $Y_{\boldsymbol{k}}(\boldsymbol{\xi}) = Y_k^n(\vartheta,\, \varphi)$. Thus, the Fourier series of f is given by $\sum_{\boldsymbol{k} \in J_\infty} \hat{f}_{\boldsymbol{k}} Y_{\boldsymbol{k}}$.

Moreover, the space of square integrable functions on the sphere $L^2(\mathbb{S}^2)$ is a Hilbert space with the inner product

$$\langle f, g \rangle_{L^2(\mathbb{S}^2)} := \int_{\mathbb{S}^2} f(\boldsymbol{\xi})\, \overline{g(\boldsymbol{\xi})} \mathrm{d}\mu(\boldsymbol{\xi}) \,,$$

possessing an orthonormal basis of the spherical harmonics, and for each function $f \in L^2(\mathbb{S}^2)$ the Parseval relation

$$\|f\|_{L^2(\mathbb{S}^2)}^2 = \sum_{\boldsymbol{k} \in J_\infty} \left|\hat{f}_{\boldsymbol{k}}\right|^2 = \left\|\hat{\boldsymbol{f}}\right\|_2^2$$

holds true.

Again, we restrict for $N \in \mathbb{N}_0$ to the finite index set

$$J_N := \bigcup_{k=0}^{N} \{k\} \times I_{2k+1}$$

and define the space of spherical polynomials by

$$T_N^{\mathcal{S}} := \operatorname{span}\{Y_{\boldsymbol{k}} : \boldsymbol{k} \in J_N\} .$$

The dimension of this space is $(N+1)^2$ and of course, $f \in T_N^{\mathcal{S}}$ has the unique expansion

$$f(\boldsymbol{\xi}) = \sum_{\boldsymbol{k} \in J_N} \hat{f}_{\boldsymbol{k}} Y_{\boldsymbol{k}}(\boldsymbol{\xi}) . \tag{2.8}$$

In addition, note that every spherical polynomial can be identified with a dilated bivariate trigonometric polynomial, i.e., for $f \in T_N^{\mathcal{S}}$ we obtain $f((2\pi)^{-1} \cdot\,) \in T_{(2N+1,2N+1)^\top}$, see [GM06] for a slightly refined version.

2.3.1 Sampling and the discrete spherical Fourier transform

Starting from two finite sets $\mathcal{X}_1 \subset [0, \pi]$ and $\mathcal{X}_2 \subset [-\pi, \pi)$ their Cartesian product $\mathcal{X}_1 \times \mathcal{X}_2$ is an obvious candidate for a sampling lattice on the sphere. Its nice properties include the easy construction of quadrature rules on the sphere, e.g., having a Gauss-Legendre grid $\mathcal{X}_1$ and an equispaced grid $\mathcal{X}_2$. However, the main drawback of such constructions is a much higher spatial resolution near the poles compared to the regions around the equator. Hence, we are interested in more general sampling sets $\mathcal{X} \subset \mathbb{S}^2$ and denote their cardinality by $M = |\mathcal{X}|$.

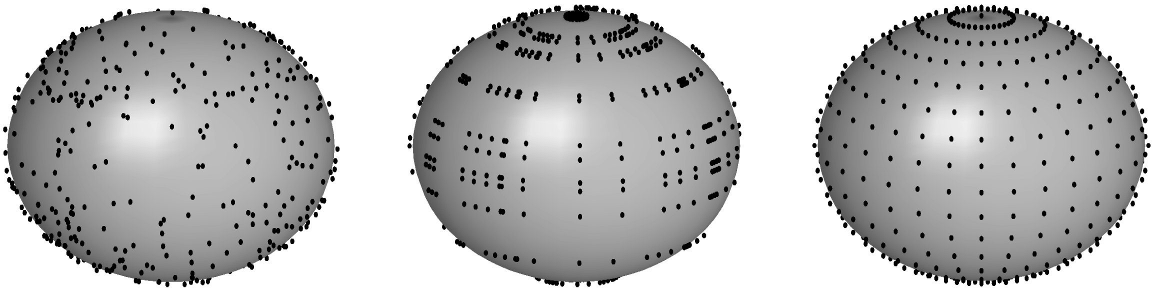


Figure 2.5: Sampling sets on the sphere, nonuniform sampling set (left), nonuniform lattice (middle), and uniform lattice (right).

Definition 2.28. *The* geodetic distance *of* $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathbb{S}^2$ *is given by*

$$\operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}, \boldsymbol{\eta}) := \arccos(\boldsymbol{\eta} \cdot \boldsymbol{\xi}) .$$

We measure the "nonuniformity" of the sampling set $\mathcal{X} \subset \mathbb{S}^2$ *by the mesh norm and the separation distance, defined by*

$$\delta_{\mathcal{X}} := 2 \max_{\boldsymbol{\xi} \in \mathbb{S}^2} \min_{j=0,\ldots,M-1} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}_j, \boldsymbol{\xi}),$$

$$q_{\mathcal{X}} := \min_{j,l=0,\ldots,M-1; j \neq l} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}_j, \boldsymbol{\xi}_l) ,$$

respectively. Analogous to the torus, cf. Definition 2.6, the sampling set $\mathcal{X}$ *is called*

1. *δ-dense for some $0 < \delta \leq 2\pi$, if $\delta_{\mathcal{X}} \leq \delta$,*
2. *q-separated for some $0 < q \leq 2\pi$, if $q_{\mathcal{X}} \geq q$.*
3. *A sequence $\{\mathcal{X}_M \subset \mathbb{S}^2 : |\mathcal{X}_M| = M\}_{M\in\mathbb{N}}$ is called quasi uniform, if $\delta_{\mathcal{X}_M} \leq C q_{\mathcal{X}_M}$ with a constant C independent of M.*

Obviously, for $\boldsymbol{\eta}$, $\boldsymbol{\xi} \in \mathbb{S}^2$ the Euclidean distance fulfils $\|\boldsymbol{\eta} - \boldsymbol{\xi}\|_2^2 = 2 - 2\boldsymbol{\eta} \cdot \boldsymbol{\xi}$ and a node $\boldsymbol{\xi} \sim (\vartheta, \varphi)$ has geodetic distance $\arccos(\boldsymbol{e}_3 \cdot \boldsymbol{\xi}) = \vartheta$ to the north pole $\boldsymbol{e}_3 = (0,0,1)^\top$. The relation between mesh-norm and separation distance is slightly more complicated on the sphere, we obtain the following lemma.

Lemma 2.29. *Let $\mathcal{X} \subset \mathbb{S}^2$ be a sampling set on the sphere, then the inequality $q_{\mathcal{X}} \leq \delta_{\mathcal{X}}$ with equality if and only if $M = 1$, or $M = 2$ and $\boldsymbol{\xi}_0 = -\boldsymbol{\xi}_1$, is fulfilled. Moreover, for a cardinality $M \geq 2$ the relations $\delta_{\mathcal{X}} \geq 4M^{-\frac{1}{2}}$ and $q_{\mathcal{X}} \leq 4.5M^{-\frac{1}{2}}$ are fulfilled.*

Proof. *The first assertion is due to the fact, that the union of spherical caps of colatitude $\delta_{\mathcal{X}}/2$ around $\boldsymbol{\xi}_j$, i.e., points $\boldsymbol{\xi}$ with $\mathrm{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}_j, \boldsymbol{\xi}) \leq \delta_{\mathcal{X}}/2$, have to cover the whole sphere, whereas the set of caps with size $q_{\mathcal{X}}$ are not allowed to share interior points. Thus, equality holds if and only if the sphere is partitioned into such spherical caps.*

Applying the technique of Lemma 2.7 and the fact that the sphere has surface area 4π yields the remaining assertions as follows. We assume $\delta_{\mathcal{X}} < 4M^{-\frac{1}{2}}$ and assign to each node $\boldsymbol{\xi}_j$ its surrounding spherical cap with colatitude $\delta_{\mathcal{X}}/2$, having an area of $2\pi(1 - \cos(\delta_{\mathcal{X}}/2))$ each. Hence, the sum of these areas can be estimated for $M \geq 2$ by $\sum_{j=0}^{M-1} 2\pi(1 - \cos(\delta_{\mathcal{X}}/2)) < 2\pi M(1 - \cos(2M^{-\frac{1}{2}})) \leq 4\pi$, i.e., these caps do not cover the whole sphere - a contradiction. Furthermore, caps with colatitude $q_{\mathcal{X}}/2 > 2.25M^{-\frac{1}{2}}$ sum up to a total area greater than 4π - another contradiction. □

Following our general notation, we define the *nonequispaced discrete spherical Fourier transform* (NDSFT) as the evaluation of a spherical polynomial (2.8), given by its vector of Fourier coefficients $\hat{\boldsymbol{f}} \in \mathbb{C}^{(N+1)^2}$, at the nodes of the sampling set $\mathcal{X} \subset \mathbb{S}^2$. As a linear mapping from the Fourier coefficients to the sample values, we write this as the matrix vector product

$$\boldsymbol{f} = \boldsymbol{Y}\hat{\boldsymbol{f}}, \tag{2.9}$$

where

$$\boldsymbol{Y} = \boldsymbol{Y}_{\mathcal{X}} := \left(Y_{\boldsymbol{k}}\left(\boldsymbol{\xi}_j\right)\right)_{j=0,\ldots,M-1;\boldsymbol{k}\in J_N} \in \mathbb{C}^{M\times(N+1)^2} \tag{2.10}$$

is called the nonequispaced spherical Fourier matrix and $\boldsymbol{f}$ denotes the vector of the samples. Of course, we are also interested in the related transforms which come with the name of the adjoint NDSFT and the inverse NDSFT. The adjoint transform, multiplying a vector with the matrix $\boldsymbol{Y}^{\mathsf{H}}$ is used subsequently within the fast Gauss transform on the sphere in Chapter 4 and as a building block for iterative schemes which realise the inverse transform, i.e., the reconstruction of Fourier coefficients from samples, cf. Chapter 5. Again, we introduce kernels on the sphere to serve within the analysis of both applications.

2.3.2 Kernels on the sphere

As proves useful here, we restrict ourselves to the spherical counterpart of radial symmetric functions for the definition of kernels on the sphere. We cite the addition theorem for spherical harmonics and hence, the expansion coefficients of such a zonal function only depend on the degree $k \in \mathbb{N}_0$ and will be computed as Fourier-Legendre coefficients of the underlying univariate function.

Definition 2.30. *A function* $K : \mathbb{S}^2 \times \mathbb{S}^2 \to \mathbb{C}$ *is called* zonal *if it depends on the geodetic distance of its two arguments only, i.e.,* $K(\boldsymbol{\xi},\boldsymbol{\eta}) = K_0(\boldsymbol{\xi}\cdot\boldsymbol{\eta})$ *for* $\boldsymbol{\xi},\boldsymbol{\eta} \in \mathbb{S}^2$ *and some function* $K_0 : [-1,1] \to \mathbb{C}$.

Theorem 2.31. *[FGS98b, p. 37] Let* $k \in \mathbb{N}_0$ *and* $\boldsymbol{\eta}$, $\boldsymbol{\xi} \in \mathbb{S}^2$. *The spherical harmonics fulfil*

$$\sum_{n=-k}^{k} \overline{Y_k^n(\boldsymbol{\eta})} Y_k^n(\boldsymbol{\xi}) = \frac{2k+1}{4\pi} P_k(\boldsymbol{\eta}\cdot\boldsymbol{\xi}) \, .$$

We assume the univariate function $K_0 : [-1,1] \to \mathbb{C}$ has an absolutely convergent Fourier-Legendre series in the sense that

$$K_0(x) = \sum_{k\in\mathbb{N}_0} \frac{2k+1}{4\pi} \hat{w}_k^{\mathcal{S}} P_k(x) \, , \qquad \sum_{k\in\mathbb{N}_0} (2k+1) \left|\hat{w}_k^{\mathcal{S}}\right| < \infty$$

with coefficients

$$\hat{w}_k^{\mathcal{S}} = 2\pi \int_{-1}^{1} K_0(x) P_k(x) \, \mathrm{d}x. \tag{2.11}$$

Thus, the associated zonal function $K : \mathbb{S}^2 \times \mathbb{S}^2 \to \mathbb{C}$ obeys the absolutely convergent expansion

$$K(\boldsymbol{\xi},\boldsymbol{\eta}) = \sum_{k\in\mathbb{N}_0} \frac{2k+1}{4\pi} \hat{w}_k^{\mathcal{S}} P_k(\boldsymbol{\xi}\cdot\boldsymbol{\eta}) \, .$$

Finite expansions of such type are spherical polynomials, denoted subsequently as kernels on the sphere, and we give a simple uniform estimate on the approximation of a zonal function when neglecting its higher frequency content in Lemma 2.33.

Definition 2.32. *For* $N \in \mathbb{N}_0$, $\hat{w}_k^{\mathcal{S}} > 0$, $k = 0,\ldots,N$, *and* $\boldsymbol{\xi}$, $\boldsymbol{\eta} \in \mathbb{S}^2$, *we define the* kernel on the sphere

$$K_N^{\mathcal{S}}(\boldsymbol{\xi},\, \boldsymbol{\eta}) := \sum_{k\in\mathbb{N}_0} \frac{2k+1}{4\pi} \hat{w}_k^{\mathcal{S}} P_k(\boldsymbol{\xi}\cdot\boldsymbol{\eta})$$

and associate for sampling sets $\mathcal{X} = \{\boldsymbol{\xi}_j\}_{j=0,\ldots,M-1}$, $\mathcal{Y} = \{\boldsymbol{\eta}_l\}_{l=0,\ldots,L-1} \subset \mathbb{S}^2$ *the matrix*

$$\boldsymbol{K}_N^{\mathcal{S}} := \left(K_N^{\mathcal{S}}(\boldsymbol{\xi}_j,\, \boldsymbol{\eta}_l)\right)_{j=0,\ldots,M-1;l=0,\ldots,L-1} ,$$

where again, the term kernel matrix *is used if* $\mathcal{X} = \mathcal{Y}$.

Again, the factorisation $\boldsymbol{K}_N^{\mathcal{S}} = \boldsymbol{Y}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{Y}_{\mathcal{X}}^{\mathsf{H}}$ is valid with $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{\boldsymbol{w}})$, $\hat{\boldsymbol{w}} = (\hat{w}_{\boldsymbol{k}})_{\boldsymbol{k}\in J_N}$.

Lemma 2.33. *Let the univariate function* $K_0 : [-1,1] \to \mathbb{C}$ *have an absolutely convergent Fourier-Legendre series and let* $K : \mathbb{S}^2 \times \mathbb{S}^2 \to \mathbb{C}$ *be its associated zonal function on the sphere. Taking only the first* $N \in \mathbb{N}_0$ *Fourier-Legendre coefficients* $\hat{w}_k^{\mathcal{S}}$ *into account yields for* $\boldsymbol{\xi},\boldsymbol{\eta} \in \mathbb{S}^2$ *the uniform error estimate*

$$\left| K(\boldsymbol{\xi},\, \boldsymbol{\eta}) - \sum_{k=0}^{N} \frac{2k+1}{4\pi} \hat{w}_k^{\mathcal{S}} P_k(\boldsymbol{\xi}\cdot\boldsymbol{\eta}) \right| \leq \sum_{k>N} \frac{2k+1}{4\pi} \left|\hat{w}_k^{\mathcal{S}}\right| \, .$$

Proof. The assertion is due to the triangle inequality and the fact $|P_k(x)| \leq 1$ *for* $|x| \leq 1$. □

The relation of a kernel on the sphere to its associated univariate algebraic polynomial gives rise to the following construction principle for localised kernels on the sphere by means of localised univariate trigonometric polynomials. We need the following lemma on the connection between the expansion with respect to Chebyshev and Legendre polynomials.

Lemma 2.34. *For $k, l \in \mathbb{N}_0$ and the Chebyshev polynomials $T_l : [-1,1] \to \mathbb{R}$, $z \mapsto T_l(z) := \cos(l \arccos z)$ the matrix $\boldsymbol{C} = (C_{k,l})_{k,l\in\mathbb{N}_0}$ with entries*

$$C_{k,l} := \int_{-1}^{1} P_k(z)\, T_l(z)\, \mathrm{d}z$$

fulfils

$$\begin{array}{ll} C_{k,l} = 2 & \textit{if } k = l = 0, \\ C_{k,l} > 0 & \textit{if } k = l, \\ C_{k,l} < 0 & \textit{if } l > k \textit{ and } (-1)^{k+l} = 1, \\ C_{k,l} = 0 & \textit{otherwise.} \end{array}$$

Moreover, the identity

$$\sum_{l=k}^{\infty} C_{k,l} = \begin{cases} 1 & k = 0, \\ 0 & \textit{otherwise,} \end{cases}$$

is satisfied.

Proof. The matrix entries obey the explicit form, cf. [Sze75, p. 99],

$$C_{2k,2l} = \frac{-2\prod_{s=0}^{k-1}\left((2l)^2 - (2s)^2\right)}{\prod_{s=0}^{k}\left((2l)^2 - (2s+1)^2\right)}, \tag{2.12}$$

$$C_{2k+1,2l+1} = \frac{-2\prod_{s=0}^{k-1}\left((2l+1)^2 - (2s+1)^2\right)}{\prod_{s=0}^{k}\left((2l+1)^2 - (2s+2)^2\right)}$$

for $l \geq k \geq 0$. We comment on the zero entries of the matrix $\boldsymbol{C}$. For the lower triangular part, i.e. $l < k$, the k-th Legendre polynomial is orthogonal to all polynomials of degree at most l; furthermore, T_{2k}, P_{2k} and T_{2k+1}, P_{2k+1} are even and odd, respectively. Moreover, the only negative factor in the proposed identity appears for $s = k = l$ and thus the diagonal entries of $\boldsymbol{C}$ are positive, whereas the non-zeros of the upper triangular part are negative. The last assertion, i.e., the diagonal dominance of $\boldsymbol{C}$ and thus the strict diagonal dominance of its finite sections is due to

$$\sum_{l=0}^{\infty} C_{0,2l} = \sum_{l=0}^{\infty} \int_{-1}^{1} T_{2l}(x)\, \mathrm{d}x = \sum_{l=0}^{\infty} \frac{2}{1 - (2l)^2} = 1$$

and the following calculations, where we restrict to the assertion (2.12). For notational convenience, let $c_{k,l} := C_{2k,2l}$, $k, l \in \mathbb{N}$. We prove for $k \in \mathbb{N}$ and by induction over $N \geq k$ that

$$\sum_{l=k}^{N-1+k} c_{k,l} = \frac{1}{2N-1} \prod_{r=0}^{2k-1} \frac{N+r}{N+r+\frac{1}{2}}. \tag{2.13}$$

For $N = 1$ and $k = 1$ equation (2.13) is fulfilled and due to

$$\begin{aligned} \frac{c_{k+1,k+1}}{c_{k,k}} &= \frac{\prod_{s=0}^{k}(k+1-s)(k+1+s) \cdot \prod_{s=0}^{k}\left(k-s-\frac{1}{2}\right)\left(k+s+\frac{1}{2}\right)}{\prod_{s=0}^{k+1}\left(k+1-s-\frac{1}{2}\right)\left(k+1+s+\frac{1}{2}\right) \cdot \prod_{s=0}^{k-1}(k-s)(k+s)} \\ &= \frac{(2k+1)(2k+2)}{\left(2k+\frac{3}{2}\right)\left(2k+\frac{5}{2}\right)} \end{aligned}$$

this extends to all $N = k$. We proceed for fixed k by

$$\begin{aligned}\sum_{l=k}^{N+k} c_{k,l} &= c_{k,N+k} + \sum_{l=k}^{N-1+k} c_{k,l}\\ &= -\frac{2\prod_{s=0}^{k-1}\left(4\left(N+k\right)^2 - 4s^2\right)}{\prod_{s=0}^{k}\left(4\left(N+k\right)^2-\left(2s+1\right)^2\right)} + \sum_{l=k}^{N-1+k} c_{k,l}\\ &= -\frac{\prod_{s=0}^{k-1}\left(N+k-s\right)\left(N+k+s\right)}{2\prod_{s=0}^{k}\left(N+k-s-\frac{1}{2}\right)\left(N+k+s+\frac{1}{2}\right)} + \sum_{l=k}^{N-1+k} c_{k,l}\end{aligned}$$

and apply the induction hypothesis (2.13), i.e.,

$$\begin{aligned}\sum_{l=k}^{N+k} c_{k,l} &= \frac{1}{2N-1}\prod_{r=0}^{2k-1}\frac{N+r}{N+r+\frac{1}{2}}\left(1-\frac{N+k}{N\left(N+2k+\frac{1}{2}\right)}\right)\\ &= \frac{1}{2N-1}\frac{N}{N+\frac{1}{2}}\left(\prod_{r=0}^{2k-2}\frac{N+1+r}{N+1+r+\frac{1}{2}}\right)\frac{N\left(N+2k+\frac{1}{2}\right)-\left(N+k\right)}{N\left(N+2k+\frac{1}{2}\right)}\\ &= \frac{1}{2N+1}\prod_{r=0}^{2k-1}\frac{N+1+r}{N+1+r+\frac{1}{2}}.\end{aligned}$$

Hence, from (2.13) follows that the N-th partial sum of each even numbered row of $\boldsymbol{C}$ decays like $\mathcal{O}\left(N^{-1}\right)$ what finally concludes our proof for these rows. The same technique yields the assertion for all odd numbered rows. ☐

Theorem 2.35. *Let the univariate, even, real, and normalised kernel*

$$K_{2N+1}\left(x\right) = \sum_{l=-N}^{N}\hat{w}_l \mathrm{e}^{-2\pi\mathrm{i}lx}, \qquad N\in\mathbb{N}_0,$$

cf. Definition 2.10, obey the localisation $|K_{2N+1}(x)| \le \tilde{C}_\beta |(N+1)x|^{-\beta}$ for all $x \in \mathbb{T}\setminus\{0\}$ and some $\beta > 1$. Moreover, let the Fourier coefficients $\hat{w}_l$ be positive and nonincreasing, i.e., $0 < \hat{w}_{|l|} \le \hat{w}_{|l-1|}$ for $l = 1,\ldots,N$.

Then, the kernel on the sphere $K_N^{\mathcal{S}} : \mathbb{S}^2\times\mathbb{S}^2 \to \mathbb{R}$,

$$K_N^{\mathcal{S}}\left(\boldsymbol{\eta},\, \boldsymbol{\xi}\right) = K_{2N+1}\left(\frac{\arccos\left(\boldsymbol{\eta}\cdot\boldsymbol{\xi}\right)}{2\pi}\right),$$

is

1. *a zonal function, a polynomial on the sphere with respect to both variables, i.e., $K_N^{\mathcal{S}}(\boldsymbol{\eta},\cdot), K_N^{\mathcal{S}}(\cdot,\boldsymbol{\xi}) \in T_N^{\mathcal{S}}$, is normalised by $K_N^{\mathcal{S}}(\boldsymbol{\xi},\,\boldsymbol{\xi}) = 1$, and obeys the localisation property*

$$K_N^{\mathcal{S}}(\boldsymbol{\eta},\,\boldsymbol{\xi}) \le \tilde{C}_\beta\left|(N+1)\frac{\arccos(\boldsymbol{\eta}\cdot\boldsymbol{\xi})}{2\pi}\right|^{-\beta}.$$

2. *Moreover, its Fourier-Legendre coefficients in (2.11) are given by*

$$\hat{w}_k^{\mathcal{S}} = 2\pi\sum_{l=k}^{N} C_{k,l}\left(2-\delta_{l,0}\right)\hat{w}_l,$$

where the connection coefficients $C_{k,l}$ are defined in Lemma 2.34. In particular, these Fourier-Legendre coefficients are positive for $k = 0,\ldots,N$.

Proof. *The assertions in 1. follow straightforward from the construction, in particular note that* $K_{2N+1}(\arccos(\boldsymbol{\eta}\cdot\boldsymbol{\xi})/(2\pi)) = \sum_{l=0}^{N}(2-\delta_{l,0})\hat{w}_l T_l(\boldsymbol{\eta}\cdot\boldsymbol{\xi})$*. The sum representation in 2. is due to the orthogonality of the Legendre polynomials. The positivity of these Fourier-Legendre coefficients is slightly more involved and follows from* $0 < \hat{w}_{|l|} \le \hat{w}_{|l-1|}$*, Lemma 2.34, and the estimate* $\hat{w}_k^{\mathcal{S}} \ge 2\pi\hat{w}_0 \sum_{l=k}^{N} C_{k,l}(2-\delta_{l,0})$ *for* $k = 0, \ldots, N$*.* ☐

3

Nonequispaced FFT and generalisations

This chapter develops the framework for the *fast* computation of the nonequispaced discrete Fourier transforms as introduced in Chapter 2, i.e., the computation of

$$f\left(\boldsymbol{x}_j\right) = \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}, \qquad \text{for given } \hat{f}_{\boldsymbol{k}} \in \mathbb{C},\ \boldsymbol{x}_j \in \mathbb{T}^d,\ j = 0, \ldots, M-1, \tag{3.1}$$

cf. formula (2.4), and its generalisations on the hyperbolic cross (2.7) and on the sphere (2.9). In Section 3.1, we recapitulate the unified approach of [Ste98, PST01] on the approximation of the nonequispaced Fourier matrix which allows for the usage of the highly efficient ordinary FFT [FJ]. Particularly, we focus on the actual requirements in terms of computational time and usage of memory with respect to the achieved accuracy, where we compare a wide range of used window functions and their possible precomputation schemes in theory as well as numerically. Section 3.2 generalises the fast evaluation techniques to the class of trigonometric polynomials on the hyperbolic cross, where we obtain the *nonequispaced sparse fast Fourier transform* (NSFFT) computing

$$f\left(\boldsymbol{x}_j\right) = \sum_{\boldsymbol{k}\in H_N^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}, \qquad \text{for given } \hat{f}_{\boldsymbol{k}} \in \mathbb{C},\ \boldsymbol{x}_j \in \mathbb{T}^d,\ j = 0, \ldots, M-1.$$

Further generalisations are presented in Section 3.3, introducing in particular the *nonequispaced fast spherical Fourier transform* (NFSFT), i.e., the computation of

$$f\left(\boldsymbol{\xi}_j\right) = \sum_{\boldsymbol{k}\in J_N} \hat{f}_{\boldsymbol{k}} Y_{\boldsymbol{k}}\left(\boldsymbol{\xi}_j\right), \qquad \text{for given } \hat{f}_{\boldsymbol{k}} \in \mathbb{C},\ \boldsymbol{\xi}_j \in \mathbb{S}^2,\ j = 0, \ldots, M-1.$$

We summarise our efforts to provide a well written software package for nonequispaced FFTs in the last section, where we also comment on the history of these discrete transforms and their applications. Most material of this chapter is based on our research papers [KP03, FKP06, KP06c, FKPar].

3.1 Nonequispaced fast Fourier transform

As we already know, the computational time, needed to perform a nonequispaced DFT is $\mathcal{O}(|I_{\boldsymbol{N}}|M)$, mapping $|I_{\boldsymbol{N}}|$ Fourier coefficients to M sample values. In contrast, the conventional FFT takes only $\mathcal{O}(|I_{\boldsymbol{N}}|\log|I_{\boldsymbol{N}}|)$ arithmetic operations for this task, whereas the sampling set has to be equispaced, i.e., $\mathcal{X} = \boldsymbol{N}^{-1} \odot I_{\boldsymbol{N}}$, cf. Theorem 2.9. Here, we summarise the developments leading to the nonequispaced fast Fourier transform, which performs the general task in time $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$. The constant of proportionality depends solely on the prescribed target accuracy and on the space dimension d.

3.1.1 Unified approach

The most successful approach for the fast computation of (3.1), cf. [DR93, Bey95, Ste98, PST01, Fou03, FS03, GL04], is based on the usage of an oversampled FFT and a window function φ which is simultaneously localised in time/space and frequency. Basically, the scheme utilises the convolution theorem in the following three informal steps, where the precise meaning of the used notation is discussed in the rest of this section:

1. deconvolve the polynomial f in (3.1) with the window function $\hat{g} \leftarrow \hat{f}/\hat{\varphi}$,
2. compute an oversampled fast Fourier transform $g \leftarrow \operatorname{fft}(\hat{g})$,
3. convolve with the window function and evaluate $f(\boldsymbol{x}_j) \leftarrow (g \star \tilde{\varphi})(\boldsymbol{x}_j)$.

Subsequently, $\sigma > 1$ and $n = \sigma N \in \mathbb{N}$ denote the *oversampling factor* and the *FFT size*. Furthermore, for $d > 1$ let $\boldsymbol{\sigma} \in \mathbb{R}^d$, $\sigma_0, \ldots, \sigma_{d-1} > 1$, $\boldsymbol{n} = \boldsymbol{\sigma} \odot \boldsymbol{N}$, and $|I_{\boldsymbol{n}}| = n_0 \cdot \ldots \cdot n_{d-1}$ denote the oversampling factor, the FFT size, and the total FFT size, respectively.

The window function

We start with a window function well localised in the time/spatial domain $\mathbb{R}$ and in the frequency domain $\mathbb{R}$, respectively.

Definition 3.1. *For some $C, \alpha, \beta > 0$, and $N \in \mathbb{N}$, let a continuous and even function $\omega : \mathbb{R} \to \mathbb{R}$ with $\omega(z) > 0$, $|z| \leq \frac{N}{2}$, and $|\omega(z)| \leq C|z|^{-1-\alpha}$, $z \in \mathbb{R}$, be given. We define $\varphi : \mathbb{R} \to \mathbb{C}$ by*

$$\varphi(x) := \int_{\mathbb{R}} \omega(z) \, \mathrm{e}^{-2\pi \mathrm{i} x z} \mathrm{d}z$$

and call it window function *if $|\varphi(x)| \leq C|x|^{-1-\beta}$, $x \in \mathbb{R}$.*

Lemma 3.2. *The window function φ is even, real-valued, continuous, and is the Fourier transform of the auxiliary function ω. Moreover, the Fourier series*

$$\tilde{\varphi}(x) := \sum_{k \in \mathbb{Z}} \omega(k) \, \mathrm{e}^{-2\pi \mathrm{i} k x}$$

obeys for $x \in \mathbb{R}$ the identity $\tilde{\varphi}(x) = \sum_{r \in \mathbb{Z}} \varphi(x + r)$ and both series converge absolutely.

Proof. The window function φ is even and real-valued since we assume the same symmetry for ω. Due to its decay, the auxiliary function ω is integrable and thus possesses a continuous Fourier transform. In conjunction with the continuity of ω, we obtain the absolute convergence of the Fourier series. Moreover, the window function φ obeys all assumptions of the Poisson summation formula, cf. Theorem 2.3 and hence, the last identity follows. □

Remark 3.3. *The decay condition on the window function φ is necessary for the absolute convergence of the periodisation $\tilde{\varphi}$. Neglecting this condition gives still a continuous and square integrable window function φ with continuous and square integrable Fourier transform $\hat{\varphi} := \omega$. The merit of relaxing the decay condition is shown by the Kaiser-Bessel function, which asks for a more involved analysis but proves to be well suited as window function.*

We wish the periodic window function $\tilde{\varphi}$ to be well localised in time/spatial domain $\mathbb{T}$ and in the frequency domain $\mathbb{Z}$, respectively. Using the latter, we truncate the Fourier series at the FFT length n which causes a hopefully small *aliasing error.* If $\tilde{\varphi}$ is furthermore well localised in time/spatial domain $\mathbb{T}$, it can be truncated with a *cut-off parameter* $m \in \mathbb{N}$, $m \ll n$ and approximated by the function $\varphi \cdot \chi_{[-\frac{m}{n},\frac{m}{n}]}$ with $\chi_{[-\frac{m}{n},\frac{m}{n}]}(x) = 1$ for $|x| \le \frac{m}{n}$ and $\chi_{[-\frac{m}{n},\frac{m}{n}]}(x) = 0$ otherwise. This causes a hopefully small *truncation error.*

To keep the aliasing error and the truncation error small, several univariate functions φ with good localisation in time and frequency domain were proposed. For an oversampling factor $\sigma > 1$, a degree $N \in 2\mathbb{N}$, the FFT length $n = \sigma N$, and a cut-off parameter $m \in \mathbb{N}$, we consider the following window functions:

1. for a shape parameter $b = \frac{2\sigma}{2\sigma-1}\frac{m}{\pi}$ the dilated *Gaussian window* [DR93, Ste98, DS99]

$$\varphi(x) = (\pi b)^{-1/2}\, \mathrm{e}^{-\frac{(nx)^2}{b}}, \tag{3.2}$$

2. for N_{2m} denoting the cardinal B-Spline of order $2m$, cf. Definition 2.15, the dilated *B-Spline window* [Bey95, Ste98]

$$\varphi(x) = N_{2m}(nx - m), \tag{3.3}$$

3. the dilated *Sinc window* [Pot03]

$$\varphi(x) = \operatorname{sinc}^{2m}\left(\frac{(2\sigma-1)\,N}{2m}\pi x\right) \tag{3.4}$$

   with $\operatorname{sinc}(x) := \sin(x)\,/\,x$ for $x \ne 0$ and $\operatorname{sinc}(0) := 1$,

4. for a shape parameter $b = \pi(2 - \frac{1}{\sigma})$ the dilated *Kaiser-Bessel window* [PS03]

$$\varphi(x) = \begin{cases} \frac{\sinh\left(b\sqrt{m^2-n^2x^2}\right)}{\pi\sqrt{m^2-n^2x^2}} & \text{for } |x| \le \frac{m}{n}, \\ \\ \frac{\sin\left(b\sqrt{n^2x^2-m^2}\right)}{\pi\sqrt{n^2x^2-m^2}} & \text{otherwise.} \end{cases} \tag{3.5}$$

Note that the latter two have compact support in frequency domain while the second one has compact support in time domain. Further references on the usage of (generalised) Kaiser-Bessel window functions include [JMNM91, Fou03, MFK04], where some authors prefer to interchange the role of time and frequency domain.

The following Figure 3.1 illustrates the usage of such a window function. The Gaussian window function and its 1-periodic version visibly coincide. This is also covered by the following lemma and hence, it is convenient to replace the periodic window function $\tilde{\varphi}$ again by the original one φ within the actual computation.

Lemma 3.4. *The Gaussian window function* (3.2) *obeys for an FFT-length* $n \ge \max\{4m, 12\}$ *and* $x \in [-\frac{m}{n}, \frac{m}{n}]$ *the estimate*

$$\frac{|\tilde{\varphi}(x) - \varphi(x)|}{|\varphi(x)|} < 10^{-16}.$$

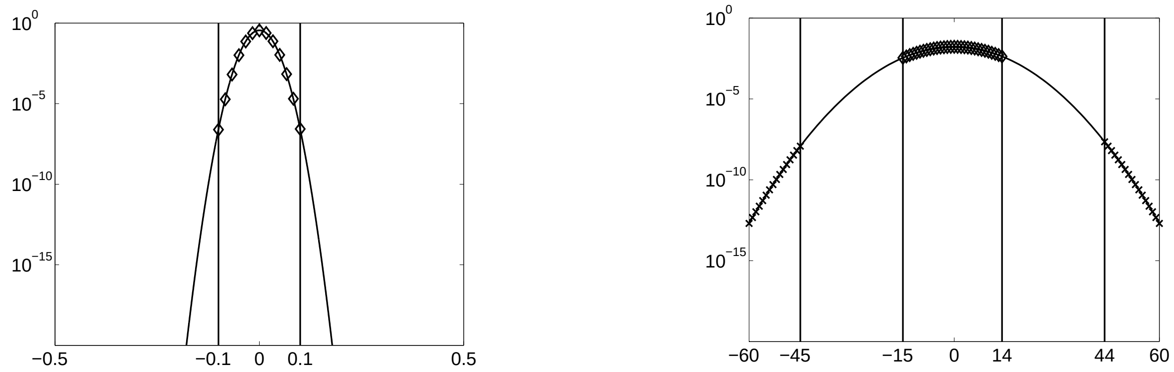


Figure 3.1: Left: Gaussian window function $\varphi(x) = c\,\mathrm{e}^{-\alpha x^2}$ (solid), cf. (3.2), and its 1-periodic version $\tilde{\varphi}$ sampled on $2m+1$ nodes $x_0 - \frac{m}{n}, x_0 - \frac{m-1}{n}, \ldots, x_0 + \frac{m}{n}$ denoted by $\diamond$ where $x_0 = 0$; Right: integral Fourier transform $\hat{\varphi}$ with "pass" ($\diamond$), "transition", and "stop" band ($\times$); for the parameters $N = 30$, $\sigma = 2$, $n = 60$, $m = 6$.

Proof. Note first the validity of

$$\int_v^{\infty} \mathrm{e}^{-u\,x^2}\,\mathrm{d}x = \int_0^{\infty} \mathrm{e}^{-u(x+v)^2}\,\mathrm{d}x \le \mathrm{e}^{-uv^2}\int_0^{\infty} \mathrm{e}^{-2uvx}\,\mathrm{d}x = \frac{\mathrm{e}^{-uv^2}}{2uv} \tag{3.6}$$

for $u, v > 0$. *Using the estimate* $|r - 2x| \ge \frac{|r|}{2}$ *in*

$$\frac{|\tilde{\varphi}(x) - \varphi(x)|}{|\varphi(x)|} = \sum_{r\in\mathbb{Z}\setminus\{0\}} \mathrm{e}^{-\frac{n^2}{b}r(r-2x)} \le 2\mathrm{e}^{-\frac{n^2}{2b}} + \int_1^{\infty} \mathrm{e}^{-\frac{n^2}{2b}r^2}\mathrm{d}r$$

together with (3.6), $b \le \frac{n}{2\pi}$, *and* $n \ge 12$ *yields the assertion.* ☐

For $d > 1$, univariate window functions $\varphi_0, \ldots, \varphi_{d-1}$, and $\boldsymbol{x} = (x_0, \ldots, x_{d-1})^\top \in \mathbb{T}^d$, we consider multivariate window functions

$$\varphi(\boldsymbol{x}) := \varphi_0(x_0)\,\varphi_1(x_1)\ldots\varphi_{d-1}(x_{d-1}) \tag{3.7}$$

with periodisation $\tilde{\varphi}(\boldsymbol{x}) = \sum_{\boldsymbol{r}\in\mathbb{Z}^d} \varphi(\boldsymbol{x} + \boldsymbol{r})$. An immediate observation is

$$\hat{\varphi}(\boldsymbol{k}) = \int_{\mathbb{R}^d} \varphi(\boldsymbol{x})\,\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}\,\mathrm{d}\boldsymbol{x} = \hat{\varphi}_0(k_0)\,\hat{\varphi}_1(k_1)\ldots\hat{\varphi}_{d-1}(k_{d-1}) \,.$$

In contrast to other parameters, e.g., the oversampling factor $\boldsymbol{\sigma} \in \mathbb{R}^d$, we use a single cut-off parameter $m \in \mathbb{N}$ to truncate the window function to the cube $\boldsymbol{n}^{-1} \odot [-m, m]^d$.

The approximation

Following the general approach of [Ste98, PST01], let φ be a window function and approximate the complex exponentials in (3.1) by

$$\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}} \approx \frac{1}{|I_{\boldsymbol{n}}|\,\hat{\varphi}(\boldsymbol{k})} \sum_{\boldsymbol{l}\in I_{\boldsymbol{n},m}(\boldsymbol{x})} \tilde{\varphi}\left(\boldsymbol{x} - \boldsymbol{n}^{-1}\odot \boldsymbol{l}\right)\mathrm{e}^{-2\pi\mathrm{i}\left(\boldsymbol{n}^{-1}\odot\boldsymbol{l}\right)\boldsymbol{k}}, \tag{3.8}$$

where the set

$$I_{\boldsymbol{n},m}(\boldsymbol{x}) := \{\boldsymbol{l} \in I_{\boldsymbol{n}} : \boldsymbol{n}\odot\boldsymbol{x} - m\boldsymbol{1} \le \boldsymbol{l} \le \boldsymbol{n}\odot\boldsymbol{x} + m\boldsymbol{1}\}$$

collects these indices where the window function is mostly concentrated (the inequalities have to be fulfilled modulo $\boldsymbol{n}$ and for each component, $\boldsymbol{1}$ denotes the vector with all entries one). This approach causes the following total error, cf. [DR93, Bey95, Ste98, PST01, Pot03].

Lemma 3.5. *For the univariate case $d = 1$ and the presented window functions φ, cf.* (3.2)-(3.5)*, the introduced error obeys*

$$\left| \mathrm{e}^{-2\pi\mathrm{i}kx} - \frac{1}{n\hat{\varphi}(k)} \sum_{l \in I_{n,m}(x)} \tilde{\varphi}\left(x - \frac{l}{n}\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{lk}{n}} \right| \le C_{\sigma,m} \tag{3.9}$$

or as approximation to our original problem (3.1) *for $j = 0, \ldots, M-1$ equivalently*

$$\left| f(x_j) - \sum_{k \in I_N} \frac{\hat{f}_k}{n\hat{\varphi}(k)} \sum_{l \in I_{n,m}(x_j)} \tilde{\varphi}\left(x_j - \frac{l}{n}\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{lk}{n}} \right| \le C_{\sigma,m} \sum_{k \in I_N} \left|\hat{f}_k\right|. \tag{3.10}$$

The involved constant is bounded for the presented window functions by

$$C_{\sigma,m} := \begin{cases} 4\,\mathrm{e}^{-m\pi(1-1/(2\sigma-1))} & \text{for (3.2), cf. [Ste98, PST01],} \\ 4\left(\frac{1}{2\sigma-1}\right)^{2m} & \text{for (3.3), cf. [Ste98],} \\ \frac{1}{m-1}\left(\frac{2}{\sigma^{2m}} + \left(\frac{\sigma}{2\sigma-1}\right)^{2m}\right) & \text{for (3.4), cf. [Pot03, Thm.1.8],} \\ 4\pi\left(\sqrt{m}+m\right)\sqrt[4]{1-\frac{1}{\sigma}}\,\mathrm{e}^{-2\pi m\sqrt{1-1/\sigma}} & \text{for (3.5), cf. [Fou03], [Pot03, Thm.1.10],} \end{cases}$$

respectively.

Proof. *Beyond previous results, we only note that the equivalence of* (3.9) *and* (3.10) *is due to the identity* $\max_{\boldsymbol{x}\in\mathbb{C}^N\setminus\{\boldsymbol{0}\}} \|\boldsymbol{P}\boldsymbol{x}\|_\infty / \|\boldsymbol{x}\|_1 = \max_{j,k} |p_{j,k}|$ *for general matrices* $\boldsymbol{P} = (p_{j,k}) \in \mathbb{C}^{M\times N}$. ☐

Thus, for fixed $\sigma > 1$, the approximation error introduced by the NFFT decays exponentially with the cut-off parameter m. Using the tensor product approach, the above error estimates have been generalised for the multivariate setting in [ES98, DS99].

The algorithm and its matrix notation

After changing the order of summation in the proposed approximation, i.e., using (3.8) in (3.1), we obtain

$$f(\boldsymbol{x}_j) \approx s_j := \sum_{\boldsymbol{l} \in I_{\boldsymbol{n},m}(\boldsymbol{x}_j)} \left(\sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}} \frac{\hat{f}_{\boldsymbol{k}}}{|I_{\boldsymbol{n}}|\,\hat{\varphi}(\boldsymbol{k})} \mathrm{e}^{-2\pi\mathrm{i}(\boldsymbol{n}^{-1}\odot\boldsymbol{l})\boldsymbol{k}} \right) \tilde{\varphi}\left(\boldsymbol{x}_j - \boldsymbol{n}^{-1}\odot\boldsymbol{l}\right).$$

As can be seen, after an initial deconvolution step in the frequency domain, the expression in brackets can be computed via a d-variate FFT of total size $|I_{\boldsymbol{n}}|$. The final step consists of a convolution, i.e., the evaluation of sums having at most $(2m+1)^d$ summands where the window function is sampled only in the neighbourhood of the node $\boldsymbol{x}_j$. More precisely, we propose Algorithm 3.1 for the fast computation of (3.1) with computational cost $\mathcal{O}(|I_{\boldsymbol{N}}|\log|I_{\boldsymbol{N}}| + m^d M)$, where a target accuracy ε is achieved for $m = \mathcal{O}(|\log\varepsilon|)$.

The proposed scheme reads in matrix vector notation, cf. equation (2.4), as

$$\boldsymbol{A}\hat{\boldsymbol{f}} \approx \boldsymbol{B}\boldsymbol{F}\boldsymbol{D}\hat{\boldsymbol{f}}, \tag{3.11}$$

where $\boldsymbol{B}$ denotes the real $M \times |I_{\boldsymbol{n}}|$ sparse matrix

$$\boldsymbol{B} := \left(\tilde{\varphi}\left(\boldsymbol{x}_j - \boldsymbol{n}^{-1}\odot\boldsymbol{l}\right) \cdot \chi_{I_{\boldsymbol{n},m}(\boldsymbol{x}_j)}(\boldsymbol{l})\right)_{j=0,\ldots,M-1;\,\boldsymbol{l}\in I_{\boldsymbol{n}}}, \tag{3.12}$$

Algorithm 3.1 NFFT

```
Input: $d, M \in \mathbb{N}$, $\boldsymbol{N} \in 2\mathbb{N}^d$,
       $\boldsymbol{x}_j \in \mathbb{T}^d$, $j = 0, \dots, M-1$, and $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$.

  For $\boldsymbol{k} \in I_{\boldsymbol{N}}$ compute
      $\hat{g}_{\boldsymbol{k}} = \frac{\hat{f}_{\boldsymbol{k}}}{|I_{\boldsymbol{n}}| c_{\boldsymbol{k}}(\tilde{\varphi})}.$
  For $\boldsymbol{l} \in I_{\boldsymbol{n}}$ compute by $d$-variate FFT
      $g_{\boldsymbol{l}} = \sum_{\boldsymbol{k} \in I_{\boldsymbol{N}}} \hat{g}_{\boldsymbol{k}} \, \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}(\boldsymbol{n}^{-1} \odot \boldsymbol{l})}.$
  For $j = 0, \dots, M-1$ compute
      $s_j = \sum_{\boldsymbol{l} \in I_{\boldsymbol{n},m}(\boldsymbol{x}_j)} g_{\boldsymbol{l}} \, \tilde{\varphi}\left(\boldsymbol{x}_j - \boldsymbol{n}^{-1} \odot \boldsymbol{l}\right).$

Output:     approximate values $s_j \approx f_j$, $j = 0, \dots, M-1$.
Complexity: $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$.
```

where $\boldsymbol{F}$ is the d-variate Fourier matrix of size $|I_{\boldsymbol{n}}| \times |I_{\boldsymbol{n}}|$, and where $\boldsymbol{D}$ is the real $|I_{\boldsymbol{n}}| \times |I_{\boldsymbol{N}}|$ "diagonal" matrix

$$\boldsymbol{D} := \bigotimes_{t=0}^{d-1} \left(\boldsymbol{O}_t \,|\, \operatorname{diag}\left(1/\hat{\varphi}_t\left(k_t\right)\right)_{k_t \in I_{N_t}} \,|\, \boldsymbol{O}_t\right)^{\top}$$

with zero matrices $\boldsymbol{O}_t$ of size $N_t \times \frac{n_t - N_t}{2}$.

The approximate matrix splitting applies to the adjoint matrix as $\boldsymbol{A}^{\mathsf{H}} \approx \boldsymbol{D}^{\top} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{B}^{\top}$, where the multiplication with the sparse matrix $\boldsymbol{B}^{\top}$ is implemented in a transposed way, summation as outer loop and using the index sets $I_{\boldsymbol{n},m}(\boldsymbol{x}_j)$ only. More precisely, the non zero elements of $\boldsymbol{B}^{\top}$ are characterised by the index set

$$I^{\top}_{\boldsymbol{n},m}(\boldsymbol{l}) := \left\{j = 0, \dots, M-1 : \boldsymbol{l} - m\boldsymbol{1} \le \boldsymbol{n} \odot \boldsymbol{x}_j \le \boldsymbol{l} + m\boldsymbol{1}\right\},$$

which fulfils

$$\bigcup_{j=0}^{M-1} \{j\} \times I_{\boldsymbol{n},m}(\boldsymbol{x}_j) = \bigcup_{\boldsymbol{l} \in I_{\boldsymbol{n}}} I^{\top}_{\boldsymbol{n},m}(\boldsymbol{l}) \times \{\boldsymbol{l}\}.$$

Thus, we obtain Algorithm 3.2, where we use matrix vector notation for convenience, to multiply with the adjoint matrix $\boldsymbol{A}^{\mathsf{H}}$ in a fast way.

Note finally, that both schemes fulfil the following error estimate with respect to the spectral norm and thus, the computational cost of Algorithm 3.1 and 3.2 can be estimated in the univariate case by $\mathcal{O}(N \log N + M \log(NM))$ when we wish a certain accuracy in the spectral norm.

Corollary 3.6. Let for $d = 1$ and $N, M \in \mathbb{N}$ the nonequispaced Fourier matrix $\boldsymbol{A} \in \mathbb{C}^{M \times N}$, cf. equation (2.5), be given. The proposed approximation obeys

$$\left\|\boldsymbol{A}^{\mathsf{H}} - \boldsymbol{B}^{\top} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{D}^{\top}\right\|_2 = \|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{F}\boldsymbol{D}\|_2 \le \sqrt{MN} C_{\sigma,m}.$$

Algorithm 3.2 adjoint NFFT

```
Input:  $d, M \in \mathbb{N}$, $\boldsymbol{N} \in 2\mathbb{N}^d$,
        $\boldsymbol{x}_j \in \mathbb{T}^d$ and $f_j \in \mathbb{C},\ j = 0, \ldots, M-1$.

  Compute the sparse matrix vector product
        $\boldsymbol{g} = \boldsymbol{B}^\top \boldsymbol{f}$.
  Apply the $d$-variate IFFT as
        $\hat{\boldsymbol{g}} = \boldsymbol{F}^{\mathsf{H}} \boldsymbol{g}$.
  Multiply by the "diagonal" matrix, i.e.,
        $\hat{\boldsymbol{s}} = \boldsymbol{D}^\top \hat{\boldsymbol{g}}$.

Output:      approximate values $\hat{s}_{\boldsymbol{k}} \approx \hat{h}_{\boldsymbol{k}},\ \boldsymbol{k} \in I_{\boldsymbol{N}}$ in (2.6).
Complexity:  $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$.
```

Proof. A matrix and its adjoint have the same spectral norm. We obtain the estimate by applying Hölder's inequality

$$\|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{F}\boldsymbol{D}\|_2 = \max_{\hat{\boldsymbol{f}} \in \mathbb{C}^N \setminus \{\boldsymbol{0}\}} \frac{\|\boldsymbol{f} - \boldsymbol{s}\|_2}{\left\|\hat{\boldsymbol{f}}\right\|_2} \le \max_{\hat{\boldsymbol{f}} \in \mathbb{C}^N \setminus \{\boldsymbol{0}\}} \frac{\sqrt{MN}\, \|\boldsymbol{f} - \boldsymbol{s}\|_\infty}{\left\|\hat{\boldsymbol{f}}\right\|_1}$$

to the second statement in Lemma 3.5. □

NDFT and Taylor expansion based approaches

Of course, the nonequispaced FFT breaks even with the NDFT only for reasonable large problem sizes. Furthermore, the straightforward computation can be sped up to some extent by avoiding the evaluations of $M|I_{\boldsymbol{N}}|$ complex exponentials in Algorithm 2.1 and the usage of a Horner-like scheme, i.e., we change the update step to $f_j = f_j \mathrm{e}^{2\pi\mathrm{i}x_j} + \hat{f}_k$, $(d = 1)$ and hence, $2dM$ direct calls of the complex exponential function suffice, see also [BM01, pp. 29]. Trading even more memory for the acceleration of the computation, one might precompute all entries of the matrix $\boldsymbol{A}$, which is only feasible for small $|I_{\boldsymbol{N}}|$ and M, see Example 3.9 in Section 3.1.3.

The case when only a few N_t are small needs more care and we exemplify our solution for $d = 2$. Splitting up the sum in equation (3.1) into both dimensions yields

$$f(\boldsymbol{x}_j) = f\left((\boldsymbol{x}_j)_0, (\boldsymbol{x}_j)_1\right) = \sum_{k_1 \in I_{N_1}} \left(\sum_{k_0 \in I_{N_0}} \hat{f}_{k_0,k_1} \mathrm{e}^{-2\pi\mathrm{i}k_0(\boldsymbol{x}_j)_0} \right) \mathrm{e}^{-2\pi\mathrm{i}k_1(\boldsymbol{x}_j)_1}. \tag{3.13}$$

Now the computation can be done by N_1 one-dimensional NFFTs for the inner bracket, followed by a direct computation of the outer sum.

A simple but nevertheless fast alternative scheme for the computation of (3.1) in the univariate case $d = 1$ is presented in [AD96]. This approach uses for each node $x_j \in \mathbb{T}$ a m-th order Taylor expansion of the trigonometric polynomial in (3.1) about the nearest neighbouring point on the oversampled equispaced lattice $\frac{1}{n}I_n$ where again $n = \sigma N$ with a somewhat larger oversampling σ. Besides its simple structure, this algorithm utilises m FFTs of size n compared to only one in the NFFT approach, uses a medium amount of extra memory, and is not suited for highly accurate computations, see Example 3.10. Due to these drawbacks, we do not consider a Taylor expansion based method in higher spatial dimensions.

3.1.2 Computational requirements

The number of floating point operations (flops) used by Algorithm 3.1 to compute (3.1) up to a fixed target accuracy ε and for a fixed spatial dimension d, and hence also by its adjoint Algorithm 3.2, is of order $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$.

More detailed, the deconvolution step of the NFFT takes $|I_{\boldsymbol{N}}|$ flops and we distinct between two schemes regarding the precomputation of the needed values of the Fourier-transformed window function $\hat{\varphi}$. Precomputing the factors $\hat{\varphi}_t(k_t)$, $k_t \in I_{N_t}$, $t = 0, \ldots, d-1$ is denoted by its associated flag `PRE_PHI_HUT` within the software library, stores $N_0 + \ldots + N_{d-1}$ real numbers, and saves $|I_{\boldsymbol{N}}|$ direct calls to $\hat{\varphi}$ during the actual NFFT. This is followed by one FFT, taking $\mathcal{O}(|I_{\boldsymbol{n}}| \log |I_{\boldsymbol{n}}|)$ flops, and hence, having higher costs also for a larger oversampling factor $\boldsymbol{\sigma}$.

This section focuses within the NFFT on the computational most involved part which turned out to be the convolution and evaluation step. As already noted, we have to choose the cut-off parameter $m = \mathcal{O}(|\log \varepsilon|)$ to achieve an element-wise approximation of the nonequispaced Fourier matrix of accuracy ε and hence the computational cost for this last step of the NFFT is $\mathcal{O}(|\log \varepsilon|^d M)$.

We compare a wide range of precomputation schemes which lead to substantially different computation times and memory requirements. We suggest different methods for the compressed storage and application of the matrix $\boldsymbol{B}$ in (3.12) which are all available within our NFFT library by choosing particular flags in a simple way during the initialisation phase. These methods do not yield a different asymptotic performance but rather yield a lower constant in the amount of computation.

Fully precomputed window function

One possibility is to precompute all values $\varphi(\boldsymbol{x}_j - \boldsymbol{n}^{-1} \odot \boldsymbol{l})$ for $j = 0, \ldots, M-1$ and $\boldsymbol{l} \in I_{\boldsymbol{n},m}(\boldsymbol{x}_j)$ explicitly. Thus, one has to store the large amount of $(2m+1)^d M$ real numbers but uses no extra floating point operations during the matrix vector multiplication beside the necessary $(2m+1)^d M$ flops. Furthermore, we store for this method explicitly the row and column index for each nonzero entry of the matrix $\boldsymbol{B}$. This method, included by the flag `PRE_FULL_PSI`, is the fastest procedure but can only be used if enough main memory is available.

Tensor product based precomputation

Using the fact that the window functions are built as tensor products one can store $\varphi_t((\boldsymbol{x}_j)_t - \frac{l_t}{n_t})$ for $j = 0, \ldots, M-1$, $t = 0, \ldots, d-1$, and $l_t \in I_{n_t,m}((\boldsymbol{x}_j)_t)$ where $(\boldsymbol{x}_j)_t$ denotes the t-th component of the j-th node. This method uses a medium amount of memory to store $d(2m+1)M$ real numbers in total. However, one has to carry out for each node at most $2(2m+1)^d$ extra multiplications to compute from the factors the multivariate window function $\varphi(\boldsymbol{x}_j - \boldsymbol{n}^{-1} \odot \boldsymbol{l})$ for $\boldsymbol{l} \in I_{\boldsymbol{n},m}(\boldsymbol{x}_j)$. Note, that this technique is available for every window function discussed here and can be used by means of the flag `PRE_PSI` which is also the default method within our software library.

Linear interpolation from a lookup table

For a large number of nodes M, the amount of memory can by further reduced by the use of lookup table techniques. For a recent example within the framework of gridding see [BNP05].

We suggest to precompute from the even window function the equidistant samples $\varphi_t(\frac{rm}{Kn_t})$ for $t = 0, \dots, d-1$ and $r = 0, \dots, K$, $K \in \mathbb{N}$, and then compute for the actual node $\boldsymbol{x}_j$ during the NFFT the values $\varphi_t((\boldsymbol{x}_j)_t - \frac{l_t}{n_t})$ for $t = 0, \dots, d-1$ and $l_t \in I_{n_t,m}((\boldsymbol{x}_j)_t)$ by means of the linear interpolation from its two neighbouring precomputed samples.

Lemma 3.7. *Let* φ *be one of the univariate window functions* (3.2) - (3.5) *and* $m \geq 2$. *Then the linear interpolated window function*

$$\varphi_K(x) := \frac{1}{2} \sum_{l=0}^{2K} \varphi(y_l)\, g_2\left(\frac{x - y_l}{2h}\right),$$

where $K \in \mathbb{N}$, $h := \frac{m}{Kn}$, $y_l := hl - \frac{m}{n}$, *and the hat function* g_2 *is defined in Definition 2.15, fulfils*

$$\max_{|x| \leq \frac{m}{n}} |\varphi(x) - \varphi_K(x)| \leq \begin{cases} \left(\dfrac{2\sigma - 1}{\sigma}\right)^{3/2} \dfrac{\pi\sqrt{2m}}{16K^2} & \text{for } (3.2)\,, \\ \dfrac{m^2}{4K^2} & \text{for } (3.3)\,, \\ \dfrac{m\,(2\sigma - 1)^2\,\pi^2}{48\sigma^2 K^2} & \text{for } (3.4)\,, \\ \dfrac{\mathrm{e}^{2\pi m}}{8K^2} & \text{for } (3.5)\,. \end{cases}$$

Proof. *Since the window function* φ *is two times continuously differentiable, the interpolation error is bounded by*

$$\max_{|x| \leq \frac{m}{n}} |\varphi(x) - \varphi_K(x)| \leq \frac{m^2}{8K^2 n^2} \max_{|\xi| \leq \frac{m}{n}} \left|\varphi^{(2)}(\xi)\right|.$$

The maximum of this second derivative is met for the window functions (3.2) - (3.5) *at* $\xi = 0$. *Thus, the assertion follows by*

$$|\varphi^{(2)}(0)| = \begin{cases} \left(\dfrac{2\sigma - 1}{\sigma m}\right)^{3/2} \dfrac{\pi n^2}{\sqrt{2}} & \text{for } (3.2)\,, \\ 2n^2\left(N_{2m-2}(m-1) - N_{2m-2}(m)\right) & \text{for } (3.3)\,, \\ \dfrac{(2\sigma - 1)^2\,\pi^2 n^2}{6m\sigma^2} & \text{for } (3.4)\,, \\ \dfrac{n^2}{2m^3\pi}\left(bm\cosh(bm) - \sinh(bm)\right) & \text{for } (3.5)\,, \end{cases}$$

and the estimates $N_{2m-2}(m-1) - N_{2m-2}(m) \leq 1$ *and* $bm\cosh(bm) - \sinh(bm) \leq 2\pi m\, \mathrm{e}^{2\pi m}$.

□

Using the symmetry of the window function, this method needs only the storage of $d(K+1)$ real numbers in total where K depends solely on the target accuracy but neither on the number of nodes M nor on the multi degree $\boldsymbol{N}$. Choosing K to be a multiple of m, we further reduce the computational costs during the interpolation since the distance from $(\boldsymbol{x}_j)_t - \frac{l_t}{n_t}$ to the two neighbouring interpolation nodes and hence the interpolation weights remain the same for all $l_t \in I_{n_t,m}((\boldsymbol{x}_j)_t)$. This method requires $2(2m+1)^d$ extra multiplications per node and is used within the NFFT by the flag `PRE_LIN_PSI`.

Fast Gaussian gridding

Two useful properties of the Gaussian window function (3.2) within the present framework were recently reviewed in [GL04]. Beside its tensor product structure for $d > 1$, which also holds for all other window functions, it is remarkable that the number of evaluations of the exponential function can be greatly decreased. More precisely, for $d = 1$ and a fixed node x_j the evaluations of $\varphi(x_j - \frac{l'}{n})$, $l' \in I_{n,m}(x_j)$, can be reduced by the splitting

$$\sqrt{\pi b}\varphi\left(x_j - \frac{l'}{n}\right) = \mathrm{e}^{-\frac{(nx_j-l')^2}{b}} = \mathrm{e}^{-\frac{(nx_j-u)^2}{b}} \left(\mathrm{e}^{-\frac{2(nx_j-u)}{b}}\right)^l \mathrm{e}^{-\frac{l^2}{b}},$$

where $u = \lfloor nx_j - m \rfloor$ and $l = 0, \ldots, 2m$. Note, that the first factor and the exponential within the brackets are constant for each fixed node x_j. Once, we evaluate the second exponential, its l-th power can be computed consecutively by multiplications only. Furthermore, the last exponential is independent of x_j and these $2m + 1$ values are computed only once within the NFFT and their amount is negligible. Thus, it is sufficient to store or evaluate $2M$ exponentials for $d = 1$. The case $d > 1$ uses $2dM$ storages or evaluations by using the general tensor product structure. This method is employed by the flags `FG_PSI` and `PRE_FG_PSI` for the evaluation or storage of $2d$ exponentials per node, respectively.

No precomputation of the window function

The last considered method uses no precomputation at all, but rather evaluates the univariate window function $(2m + 1)^d M$ times. Thus, the computational time depends on how fast we can evaluate the particular window function. However, no additional storage is necessary which suits this approach whenever the problem size reaches the memory limits of the used computer.

Summary on the different precomputation schemes

As pointed out already, the NFFT takes order $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$ floating point operations to achieve a fixed target accuracy. In the actual usage of Algorithm 3.1 or 3.2, the most time consuming part is the convolution and evaluation step. This multiplication with the sparse matrix $\boldsymbol{B}$, cf. (3.11), clearly takes $\mathcal{O}(m^d M)$ operations achieving a target accuracy c^{-m}, see Lemma 3.5. Table 3.1 summarises the memory requirements and the extra costs it takes to use "compressed" versions of the matrix $\boldsymbol{B}$, i.e., to trade precomputation storage for computation time.

Method	Memory	Flops	Evaluations
no flag	0	m^d	m^d
`FG_PSI`	0	$m^d + dm$	d
`PRE_LIN_PSI`	dK	$m^d + dm$	0
`PRE_FG_PSI`	dM	$m^d + dm$	0
`PRE_PSI`	dmM	m^d	0
`PRE_FULL_PSI`	$m^d M$	0	0

Table 3.1: Theoretical memory requirements to store the whole matrix $\boldsymbol{B}$ (memory) and the number of extra floating point operations (flops) and evaluations of the univariate window function φ (evaluations) it takes to compute the nonzero entries in one row of $\boldsymbol{B}$ during the multiplication phase.

3.1.3 Numerical experiments

This section gives detailed information on the computation time, the memory requirements, and the accuracy of Algorithm 3.1. We start by an introductory comparison of the computation time needed by the FFT, NDFT, and the NFFT with respect to an increasing problem size $N, M \in \mathbb{N}$ in Example 3.8. This is followed by testing particular univariate transforms - accelerated NDFTs and the Taylor expansion based NFFT in Example 3.9 and Example 3.10, respectively.

Example 3.8. In this introductory example, we compare the computation time of the FFT ([FJ], `FFTW_MEASURE`), the NDFT (Algorithm 2.1), and the NFFT (Algorithm 3.1) for increasing total problem sizes $|I_{\boldsymbol{N}}|$ and space dimensions $d = 1, 2, 3$, where $\boldsymbol{N} = (N, \ldots, N)^{\top}$, $N \in \mathbb{N}$. While the nodes for the FFT are restricted to the lattice $\boldsymbol{N}^{-1} \odot I_{\boldsymbol{N}}$, we choose $M = N^d$ random nodes for the NDFT and the NFFT. Within the latter, we use the oversampling factor $\sigma = 2$, the cut-off $m = 4$, and the Kaiser-Bessel window function (`PRE_PSI`, `PRE_PHI_HUT`). This results in a fixed accuracy of $E_\infty := \|\boldsymbol{f} - \boldsymbol{s}\|_\infty / \|\hat{\boldsymbol{f}}\|_1 \approx 10^{-8}$ for $d = 1, 2, 3$, see also Lemma 3.5.

We conclude the following: The FFT and the NFFT show the expected $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}|)$ time complexity, i.e., doubling the total size $|I_{\boldsymbol{N}}|$ results in only slightly more than twice the computation time, whereas the NDFT behaves as $\mathcal{O}(|I_{\boldsymbol{N}}|^2)$. Note furthermore, that the constant in the $\mathcal{O}$-notation is independent of the space dimension d for the FFT and the NDFT, whereas the computation time of the NFFT increases considerably for larger d.

l_N	FFT	NDFT	NFFT	l_N	FFT	NDFT	NFFT
$d = 1$				$d = 2$			
3	$1.3e-07$	$8.7e-06$	$4.6e-06$	6	$9.9e-07$	$5.7e-04$	$3.2e-04$
4	$2.0e-07$	$3.5e-05$	$8.7e-06$	8	$4.4e-06$	$9.2e-03$	$1.3e-03$
5	$4.0e-07$	$1.4e-04$	$1.7e-05$	10	$2.1e-05$	$1.5e-01$	$5.2e-03$
6	$8.9e-07$	$5.6e-04$	$3.6e-05$	12	$1.2e-04$	$2.4e+00$	$2.3e-02$
7	$2.2e-06$	$2.2e-03$	$7.2e-05$	14	$1.7e-03$	$3.8e+01$	$1.5e-01$
8	$4.8e-06$	$9.0e-03$	$1.4e-04$	16	$2.1e-02$	*	$6.8e-01$
9	$1.1e-05$	$3.6e-02$	$2.9e-04$	18	$8.4e-02$	*	$2.8e+00$
10	$2.4e-05$	$1.4e-01$	$6.0e-04$	20	$3.2e-01$	*	$1.2e+01$
11	$5.7e-05$	$5.8e-01$	$1.4e-03$	22	$1.4e+00$	*	$5.3e+01$
12	$1.5e-04$	$2.3e+00$	$3.2e-03$	$d = 3$			
13	$5.5e-04$	$9.4e+00$	$8.2e-03$	9	$1.0e-05$	$3.7e-02$	$2.5e-02$
14	$1.7e-03$	$3.8e+01$	$2.0e-02$	12	$1.1e-04$	$2.4e+00$	$2.5e-01$
15	$3.8e-03$	$1.5e+02$	$4.9e-02$	15	$3.4e-03$	$1.5e+02$	$2.4e+00$
16	$8.2e-03$	*	$1.2e-01$	18	$5.2e-02$	*	$2.1e+01$
17	$1.9e-02$	*	$2.4e-01$	21	$9.0e-01$	*	$1.8e+02$
18	$4.5e-02$	*	$3.6e-01$				
19	$9.2e-02$	*	$9.8e-01$				
20	$1.9e-01$	*	$2.1e+00$				
21	$4.2e-01$	*	$4.2e+00$				
22	$1.0e-00$	*	$9.5e+00$				

Table 3.2: Computation time in seconds with respect to $l_N = \log_2 |I_{\boldsymbol{N}}|$. Note that we used accumulated measurements in case of small times and the times (*) are not displayed due to the large response time in comparison to the FFT time.

Example 3.9. Now, we examine the computation time of accelerated univariate NDFTs with respect to increasing $N, M \in \mathbb{N}$. The three proposed possibilities to compute the matrix vector product differ by their access to the nonequispaced Fourier matrix $\boldsymbol{A}$, we use either MN direct calls of the complex exponential function, a Horner-like scheme, or a fully precomputed matrix.

Figure 3.2 (left) reveals, that the complete precomputation of the matrix $\boldsymbol{A}$ does not pay off. The Horner-like NDFT uses no extra memory and is considerably faster than the NDFT. Furthermore, this version is faster than the default NFFT until a break even of $N = 128$.

Example 3.10. This example considers the computation time and the accuracy of the Taylor expansion based NFFT, again only for $d = 1$. We note that this scheme actually provides a competitive univariate NFFT with respect to the computation time relative to the problem size $N, M \in \mathbb{N}$ - at least within a factor, cf. Figure 3.2 (right).

Moreover, we compare the accuracy of this transform with the accuracy of Algorithm 3.1 (Gaussian window function). Figure 3.3 (left) reveals, that it is not possible to obtain high accurate results by increasing the order m of the Taylor expansion and hence the number of used FFTs. This fact remains even true for a very large oversampling factor $\sigma = 16$. Furthermore, even when the allowed error $\tilde{E}_\infty := \|\boldsymbol{f} - \boldsymbol{s}\|_\infty \,/\, \|\boldsymbol{f}\|_\infty$, used, e.g., in [AD96], is somewhat larger, Algorithm 3.1 needs considerable fewer arithmetic operations to reach it, cf. Figure 3.3 (right).

More detailed, we consider the computation time of the deconvolution step, the oversampled FFT, and the convolution/evaluation step of Algorithm 3.1 separately in Example 3.11 and Example 3.12. Afterwards, we fix the problem size and focus in Example 3.13 on the convolution/evaluation step, where we compare the computation time for different window functions and precomputation schemes with respect to the achieved accuracy. Finally, Example 3.14 verifies the quadratic growth of the accuracy for an increasing size K of the lookup table when using the linear interpolated window function.

Example 3.11. We now compare the computation time for the three tasks within Algorithm 3.1, i.e., the deconvolution step, the oversampled FFT, and the convolution/evaluation step for space dimension $d = 1$. Figure 3.4 shows the timings for increasing degree N, $M = N$ nodes, and a fixed cut-off $m = 4$. The linear dependence of the computation time with respect to the problem size can be seen for the matrix-vector multiplication with the "diagonal" matrix $\boldsymbol{D}$ and the sparse matrix $\boldsymbol{B}$ whereas the FFT takes $\mathcal{O}(N \log N)$ operations.

For the deconvolution step we obtain a speed up of more than 3 by avoiding direct calls of the Fourier-transformed window function $\hat{\varphi}$, this method is default and turned on by the precomputation flag `PRE_PHI_HUT`, cf. Figure 3.4 (top-left). Ways to speed up the FFT by a more exhaustive search of an optimal FFT-plan are discussed in [FJ], Figure 3.4 (top-right) shows for larger degree N a speed up of around 2 when we use the planner `FFTW_MEASURE`, which is also default within the NFFT software. Note furthermore, that the $\log N$-term in the computational complexity of the FFT is of minor interest over large scales of the problem size.

The time to compute the last step of Algorithm 3.1 differs from no precomputation of the matrix entries of $\boldsymbol{B}$ to explicitly precomputed entries with `PRE_FULL_PSI` by a factor of 20 to 100 for small degrees $N \leq 2048$ and by a factor of 5 to 20 for larger degrees. Note however, that the use of this flag with maximal precomputation is limited by the available memory, e.g. for $m = 4$, and $M = 2^{20}$ we already need 144 MByte just for storing the matrix entries and its indices.

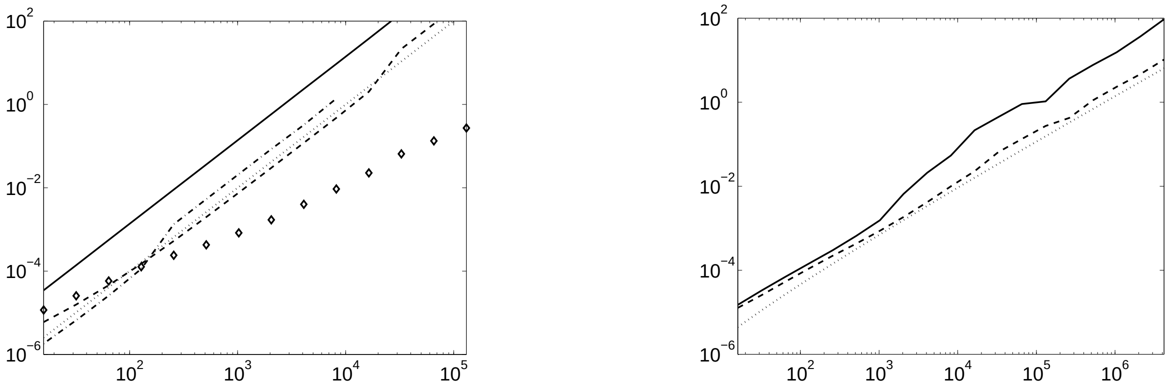


Figure 3.2: Comparison of different NDFTs and the Taylor expansion based NFFT with Algorithm 3.1 for the univariate case $d = 1$. Computation time in seconds with respect to increasing degree $N = 2^4, \ldots, 2^{22}$ and $M = N$. Left: NDFT (solid), Horner-like NDFT (dashed), Multiplication with fully precomputed Matrix $\boldsymbol{A}$ (dash-dot), the curve $10^{-8}N^2$ (dotted), and default version of Algorithm 3.1, i.e. Kaiser-Bessel window, $\sigma = 2$, $m = 6$, and precomputation methods `PRE_PHI_HUT` and `PRE_PSI` ($\diamond$). Right: Taylor expansion based NFFT with $\sigma = 4$, $m = 6$ (solid), Algorithm 3.1 with $\sigma = 2$, $m = 6$, and precomputed fast Gaussian gridding `PRE_FG_PSI` (dashed), which uses the same amount of memory, and the curve $10^{-7}N \log N$ (dotted).

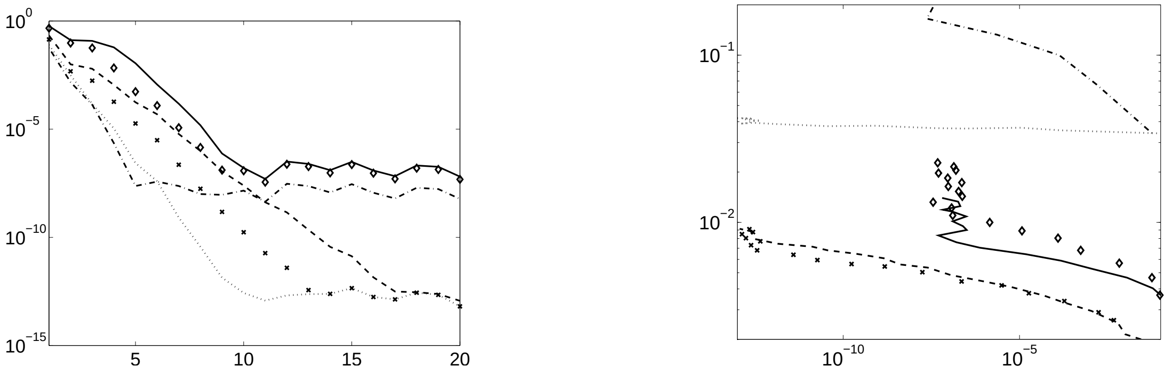


Figure 3.3: Comparison of the Taylor expansion based NFFT with the Algorithm 3.1 for the univariate case $d = 1$. Accuracy of the Taylor expansion based NFFT and Algorithm 3.1 with respect to increasing Taylor-order/cut-off $m = 1, \ldots, 20$, fixed degree $N = 4096$ and $M = N$ nodes. Different oversampling factors are denoted for the Taylor expansion based NFFT as $\sigma = 1.5$ (solid), $\sigma = 2$ ($\diamond$), $\sigma = 16$ (dash-dot) and for Algorithm 3.1 as $\sigma = 1.5$ (dashed), $\sigma = 2$ ($\times$), and $\sigma = 16$ (dotted). Algorithm 3.1 is used with precomputed fast Gaussian gridding `PRE_FG_PSI`. Left: Accuracy of the Taylor expansion based NFFT with respect to increasing order m of the Taylor expansion and accuracy of Algorithm 3.1 with respect to increasing cut-off m. Right: Computation time in seconds with respect to achieved accuracy $\tilde{E}_\infty$.

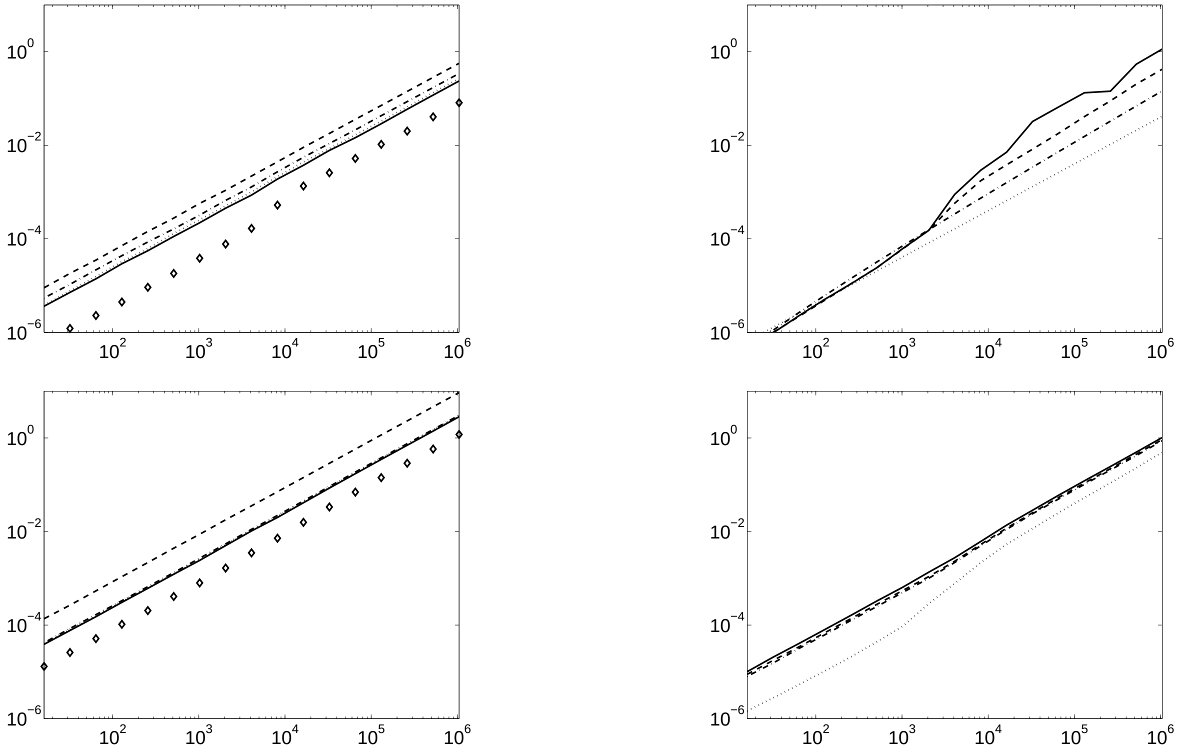


Figure 3.4: Computation time in seconds with respect to increasing degree $N = 2^4, \dots, 2^{20}$, $M = N$ nodes, $d = 1$, cut-off $m = 4$, and oversampling factor $\sigma = 2$. Window functions are denoted (top-left) and (bottom-left) by: Gaussian (solid), Kaiser-Bessel (dashed), Sinc (dash-dot), and B-Spline (dotted). **Top:** Left: Deconvolution step, i.e., multiplication with the "diagonal" matrix $\boldsymbol{D}$, where the method with precomputation `PRE_PHI_HUT` is denoted by $\diamond$. Right: Oversampled FFT of length $n = \sigma N$, planner flags are `FFTW_ESTIMATE` (solid) and `FFTW_MEASURE` (dashed). Furthermore, the curves $10^{-8} N \log N$ (dash-dot) and $4 \cdot 10^{-8} N$ (dotted) are shown. **Bottom:** Convolution/evaluation step, i.e., multiplication with the sparse matrix $\boldsymbol{B}$. Left: Comparing the different window functions without any precomputation, denoted as above and the fast Gaussian gridding `FG_PSI` ($\diamond$). Right: Precomputed Gaussian window function with all proposed methods, i.e., `PRE_LIN_PSI` (solid), `PRE_FG_PSI` (dashed), `PRE_PSI` (dash-dot), and `PRE_FULL_PSI` (dotted).

Example 3.12. Furthermore, we show the timings of the convolution/evaluation step for increasing N, the multi degree $\boldsymbol{N} = (N, \dots, N)^\top$, $M = N^d$ nodes, a fixed cut-off $m = 4$, and space dimension $d = 2, 3$ in Figure 3.5. Note, that for $d = 2$ and $m = 4$ the matrix $\boldsymbol{B}$ has already 81 nonzero entries per row.

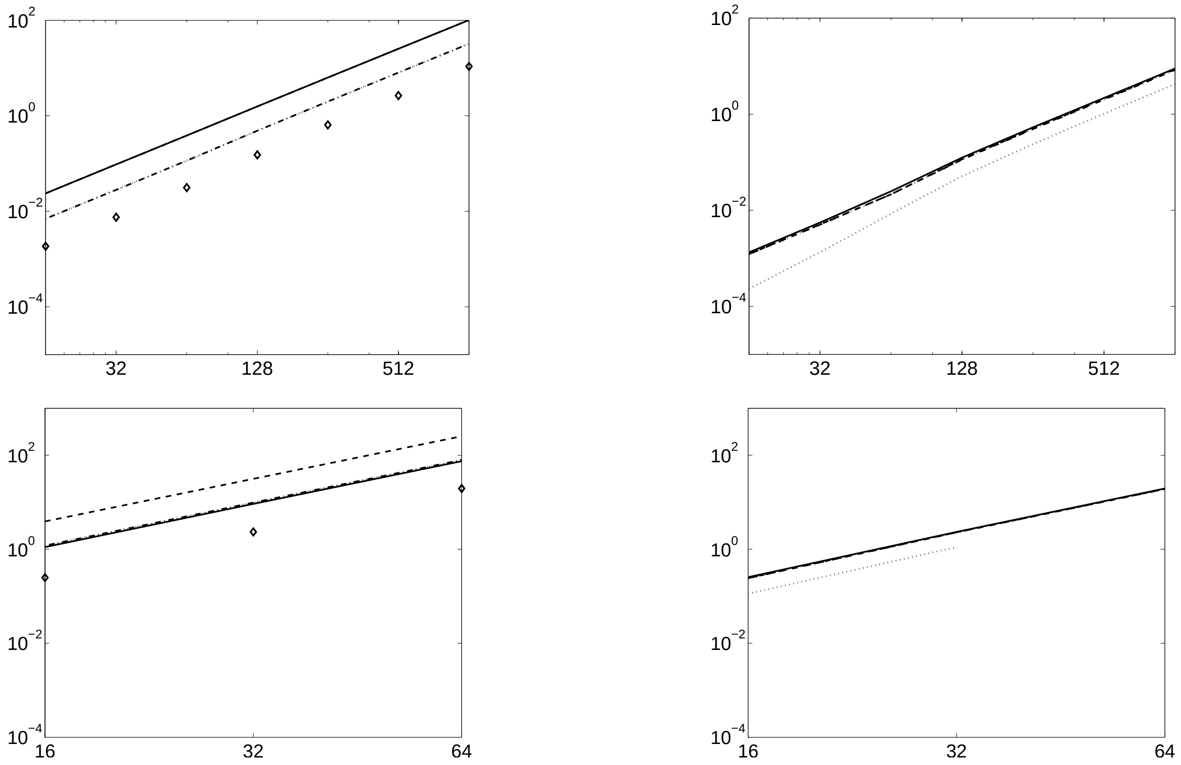


Figure 3.5: Computation time of the multivariate convolution/evaluation step in seconds with respect to increasing multi degree $\boldsymbol{N} = (N, \dots, N)^\top$, cut-off $m = 4$, and oversampling factor $\sigma = 2$. **Top:** Space dimension $d = 2$, degree $N = 2^4, \dots, 2^{10}$ and $M = N^2$ nodes. **Bottom:** Space dimension $d = 3$, degree $N = 2^4, 2^5, 2^6$ and $M = N^3$ nodes. Window functions and precomputations are shown as in Figure 3.4 (bottom).

Example 3.13. More detailed, we focus on the convolution/evaluation step for space dimension $d = 1$. Figure 3.6 shows the computation time with respect to achieved accuracy $\tilde{E}_2 := \|\boldsymbol{f} - \boldsymbol{s}\|_2 \,/\, \|\boldsymbol{f}\|_2$, used, e.g., in [PST01], by increasing the cut-off m for fixed degree and number of nodes.

We conclude, that if no additional memory is used for precomputing the entries of the matrix $\boldsymbol{B}$, the Gaussian window function in conjunction with the flag `FG_PSI` performs best, cf. Figure 3.6 (top-left). If no precomputation is used, the non desirable behaviour of the B-Spline window function is due to the fact that evaluating this window function once already takes $\mathcal{O}(m)$ operations. When only a small amount of memory is used for precomputations, the decision between the linear interpolated Kaiser-Bessel window function and the fast Gaussian gridding with precomputation `PRE_FG_PSI` depends on the accuracy one would like to achieve - here, the linear interpolated Kaiser-Bessel window performs better up to single precision (top-right). Whenever at least $2mM$ values can be precomputed, the Kaiser-Bessel window performs always best, i.e., needs the least time to achieve a given target accuracy, cf. Figure 3.6 (bottom).

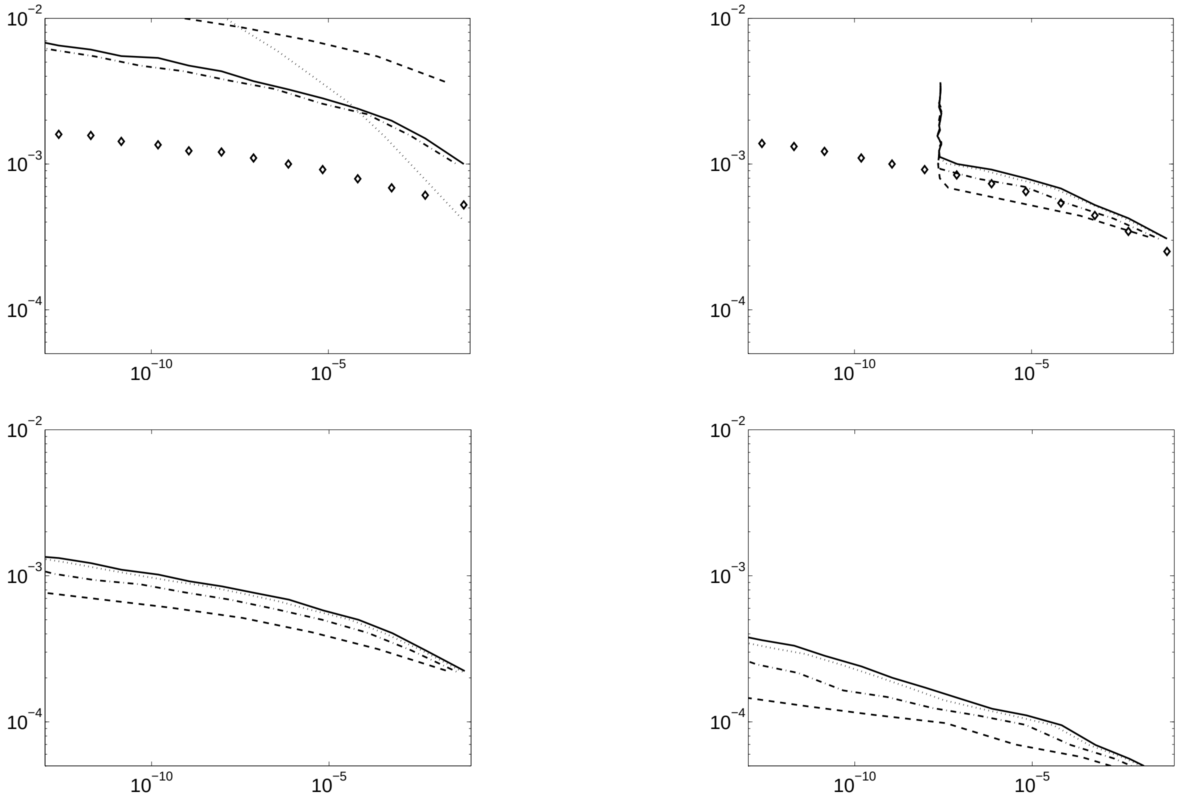


Figure 3.6: Computation time in seconds with respect to achieved accuracy $\tilde{E}_2$. We use a fixed degree $N = 1024$, $M = N$ nodes, and set $d = 1$. Increasing the cut-off $m = 1, \ldots, 20$ increases both, the accuracy and the needed computation time. Window functions: Gaussian (solid), Kaiser-Bessel (dashed), Sinc (dash-dot), and B-Spline (dotted). **Top:** Left: No precomputation, fast Gaussian gridding without precomputation `FG_PSI` is denoted by $\diamond$. Right: Linear interpolated window function `PRE_LIN_PSI` from lookup table with $K = 2^{11}m$ precomputed values achieving single precision 10^{-8}; and fast Gaussian gridding with precomputation `PRE_FG_PSI` ($\diamond$). **Bottom:** Left: Tensor product based precomputation `PRE_PSI`. Right: Fully precomputed matrix $\boldsymbol{B}$, i.e. `PRE_FULL_PSI`.

Example 3.14. Finally, Table 3.3 shows the quadratic decay of the error introduced by the linear interpolation of the window function if the method `PRE_LIN_PSI` is used, i.e., the error $\tilde{E}_2$ decreases to a quarter when doubling the size of the lookup table.

l_K	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\tilde{E}_2$	$2.4e-02$	$9.0e-03$	$1.5e-03$	$3.9e-04$	$8.9e-05$	$2.4e-05$	$1.1e-05$	$1.6e-06$	$3.1e-07$	$7.2e-08$	$1.7e-08$	$1.1e-08$	$1.5e-09$	$2.7e-10$

Table 3.3: Accuracy of Algorithm 3.1 with linear interpolated window function. We use the Kaiser-Bessel window and set $m = 10$, $d = 1$, $N = M = 1024$ and increase the size of the lookup table K, $l_K := \log_2(K/(m + 1))$.

3.1.4 Concluding remarks and comments

Algorithm 3.1 takes $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$ arithmetic operations for $|I_{\boldsymbol{N}}|$ Fourier coefficients, M nonequispaced sampling nodes, and a fixed target accuracy. However, various precomputation strategies result in substantially different computation time and usage of memory for interesting problem sizes, we summarise:

If the problem size is really small, e.g. $N = M < 32$ for $d = 1$, just use the NDFT or its Horner-like derivative. The simplest fast method is the Taylor expansion based NFFT, it achieves not even single precision, needs a somewhat larger oversampling factor, and is slower than window function based methods. If the problem barely fits into the computer, one should use the fast Gaussian gridding NFFT, i.e., the Gaussian window function in conjunction with the flag `FG_PSI` which uses no extra memory.

Using only a small amount of memory for precomputation and asking for high accuracy, the fast Gaussian gridding NFFT with precomputation performs best while storing $2d$ real numbers per node. However, the Kaiser-Bessel window in conjunction with the lookup table method `PRE_LIN_PSI` with 2^{12} precomputed values suffices for single precision 10^{-8}, regardless of the problem size, and outperforms the fast Gaussian gridding. Furthermore, the lookup table is the only precomputation method which is independent of the actual sampling set.

If a medium amount of memory can be used for precomputation, the Kaiser-Bessel window function performs best. The tensor product based precomputation scheme `PRE_PSI` yields a faster NFFT than the lookup table method or the fast Gaussian gridding with precomputation, but stores for each node dm real numbers. For small to medium size problems, one can gain another factor 2 to 5 by means of a fully precomputed window function `PRE_FULL_PSI`. However, this causes a storage cost of m^d real numbers per sampling node.

Default precomputation methods, selected by the simple initialisation routine of the NFFT, are: `PRE_PHI_HUT` for the deconvolution step, the flag `FFTW_MEASURE` for planning the FFT, and the tensor product based precomputation scheme `PRE_PSI` for the convolution/evaluation step. Furthermore, the Kaiser-Bessel window function is selected as default at compilation.

History of the NFFT

We stress again that the nonequispaced FFT and its adjoint are algorithms that realise the matrix vector multiplications with $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{H}}$, cf. equation (2.5), in a fast and approximate way, respectively. The origin of such schemes dates back to the eighties, when *(re-)gridding*, denoting the adjoint NFFT, was invented in astrophysics, tomography, and engineering, cf. [O'S85, JMNM91, ST95, Pel97]. Gridding is often done in three steps: "approximation to an oversampled Cartesian grid", inverse fast Fourier transform, and "roll-off correction", see e.g. [JMNM91]. In conjunction with a preceding sampling density compensation, its main purpose has been the computation of an *inverse* NFFT, whereas Chapter 5 presents far more accurate algorithms for this task.

A second body of literature refers to these generalised fast Fourier transforms as non-uniform [FS03], generalised [DR93], unequally-spaced [Bey95], or non-equispaced [Fou03]. Here, also the first rigorous derivation of the NFFT, including the relation between speed and accuracy of the algorithm appears [DR93, Bey95, AD96]. In [Ste98, PST01] a unified approach was suggested and the error estimates from [DR93] were improved, which led to criteria for the choice of the parameters of the algorithm. In particular, various papers on the nonequispaced FFT differ only by the chosen window function φ, e.g. a Gaussian pulse tapered with a Hanning window in [DS99], Gaussian kernels combined with sinc kernels in [Pel97], and special optimised windows in [JMNM91, DS99]. Moreover, an approach for the univariate case $d = 1$ is considered in [DR95] and based on a Lagrange interpolation technique. After taking an FFT of length N of the vector $\hat{\boldsymbol{f}}$ in (3.1) one uses an exact polynomial interpolation scheme to obtain the values of the trigonometric polynomial f at the nonequispaced nodes x_j. Here, the time consuming part is the exact polynomial interpolation scheme which can however be realised fast in an approximate way by means of the fast multipole method, see e.g. [GR87, Gre88, BG97, SP01]. Nevertheless, numerical experiments in [DR95, Dro96] showed that this approach is far more time consuming than Algorithm 3.1. Similar multilevel algorithms are also used in [Bra91] for a more general class of matrices. Using a fully discrete approach, one might fix the entries of the "diagonal" matrix $\boldsymbol{D}$ in (3.11) first and precompute optimised entries for the sparse matrix $\boldsymbol{B}$ in (3.12) to achieve higher accuracy. Conceptually slightly more complicated and more expensive within the precomputation step, these schemes gain some accuracy for the Gaussian or B-Spline windows, whereas no reasonable improvement is obtained for the Kaiser-Bessel window function, cf. [NL99, NS03, FS03, Fou03].

We conclude, that Algorithm 3.1 and its adjoint Algorithm 3.2 can be realised with a broad class of suggested window functions yielding the same accuracy by changing the cut-off parameter m. In comparison of several algorithms, cf. [War98, PST01, GL04, KP06c], the NFFT with Kaiser-Bessel window performs best whenever one can afford some memory for precomputation and the Gaussian window if this is not the case, cf. [GL04]. Furthermore, the NFFT has been shown to be as effective as the FFT for the univariate setting $d = 1$ and still competitive for $d = 2, 3$. Tutorials on the NFFT include [PST01, Pot03, Fen06],[HL05, Section 3.2] for an introduction and [KPS02, KP04a] for a detailed discussion of version 1.0 and 2.0 of the NFFT software package [KP06b]. Alternatively, implementations of the NFFT are available in the MATLAB toolboxes [FS02b, KR06a].

There is a variety of important applications which utilise the NDFT, e.g. in computerised tomography, for fast summation algorithms [PS03], as fast Fourier transform on the sphere [KP03], or as part of the ridgelet and curvelet transforms [MF06, CDDY06, Fen06]. Furthermore, the reconstruction from nonequispaced samples is stated in [FGS95, AD96, KP04b] as inversion of the NDFT and used for example in magnetic resonance imaging [FS03]. In each of these applications, the actual computation of the NDFT is the computationally dominant task and one has to deal with different requirements on the NFFT with respect to the target accuracy, the usage of memory, and the actual computation time.

3.2 Nonequispaced FFT on the hyperbolic cross

As we have seen already, the discrete Fourier transform in d dimensions with Fourier coefficients supported on the set $\boldsymbol{I_N}$, $\boldsymbol{N} = (N, \dots, N)^\top$, and nodes on the lattice $\boldsymbol{N}^{-1} \odot \boldsymbol{I_N}$ in spatial domain can be done by the fast Fourier transform in $\mathcal{O}(N^d \log N)$ arithmetic operations and these methods have been generalised for nonequispaced nodes in the previous section. Furthermore, if the frequencies are chosen from a hyperbolic cross and the nodes are located on a particular sparse grid, there exist fast methods utilising a representation as in Definition 2.21 that need only $\mathcal{O}(N \log^d N)$ arithmetic operations for a *sparse FFT*, cf. [BD89, Hal92].

In contrast, we propose an algorithm for the *nonequispaced fast Fourier transform on the hyperbolic cross* (NSFFT for nonequispaced sparse FFT) in two and three dimensions based on the NFFT and an appropriate partitioning of the hyperbolic cross. Thus, we aim to compute for $N = 2^J$, $J \in \mathbb{N}$, $J \geq 2$, in a fast way

$$f(\boldsymbol{x}_j) = \sum_{\boldsymbol{k} \in H_N^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}, \qquad \text{for given } \hat{f}_{\boldsymbol{k}} \in \mathbb{C},\ \boldsymbol{x}_j \in \mathbb{T}^d,\ j = 0, \dots, M-1, \tag{3.14}$$

cf. formula (2.7) in Section 2.2.

Given a partition (disjoint union) of the hyperbolic cross $H_N^d = \bigcup_r (I_{\boldsymbol{N}_r} + \boldsymbol{\rho}_r)$ into blocks with frequency-shifts $\boldsymbol{\rho}_r \in \mathbb{Z}^d$, we use ordinary NFFTs of size $|I_{\boldsymbol{N}_r}|$ to compute (3.14), i.e.,

$$f(\boldsymbol{x}_j) = \sum_r \omega_{j,r} \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}_r}} \hat{f}_{\boldsymbol{k}+\boldsymbol{\rho}_r} \, \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j} \,,$$

where $\omega_{j,r} = \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{\rho}_r\boldsymbol{x}_j}$ denote "nonuniform twiddle factors". Note furthermore, that the multivariate version of Lemma 3.5 on the accuracy of approximating the complex exponentials remains true in the present setting and of course, the partitioning immediately yields the adjoint transform as well. The arithmetic complexity for one set of indices $I_{\boldsymbol{N}_r}$ out of the partition is $\mathcal{O}(|I_{\boldsymbol{N}_r}| \log |I_{\boldsymbol{N}_r}| + M)$, see also observation (3.13), and hence, our main task consists in the construction of a partition with a small number of blocks.

3.2.1 Bivariate case

The hyperbolic cross H_N^2, $N = 2^J$, $J \in \mathbb{N}$, $J \geq 2$, is partitioned with the help of the short hand notation $\tilde{I}_r := I_{2^r}$ for $r = 0, \ldots, \lceil \frac{J}{2} \rceil - 1$ into the blocks

$$\begin{aligned}
H_{J,r}^{\text{left}} &:= \tilde{I}_r \times (\tilde{I}_{J-r-2} - \lceil 3 \cdot 2^{J-r-3} \rceil), \\
H_{J,r}^{\text{right}} &:= \tilde{I}_r \times (\tilde{I}_{J-r-2} + \lfloor 3 \cdot 2^{J-r-3} \rfloor), \\
H_{J,r}^{\text{top}} &:= (\tilde{I}_{J-r-2} + \lfloor 3 \cdot 2^{J-r-3} \rfloor) \times \tilde{I}_r, \\
H_{J,r}^{\text{bottom}} &:= (\tilde{I}_{J-r-2} - \lceil 3 \cdot 2^{J-r-3} \rceil) \times \tilde{I}_r, \\
H_J^{\text{center}} &:= \tilde{I}_{\lfloor \frac{J}{2} \rfloor} \times \tilde{I}_{\lfloor \frac{J}{2} \rfloor}, \\
H_{J,r} &:= H_{J,r}^{\text{left}} \cup H_{J,r}^{\text{right}} \cup H_{J,r}^{\text{top}} \cup H_{J,r}^{\text{bottom}}, \\
H_N^2 &= H_J^{\text{center}} \cup \bigcup_{r=0}^{\lceil \frac{J}{2} \rceil - 1} H_{J,r}.
\end{aligned}$$

We readily obtain

$$|H_{J,r}^{\text{left}}| = \ldots = |H_{J,r}^{\text{bottom}}| = 2^{J-2}, \qquad |H_J^{\text{center}}| = 2^{2\lfloor \frac{J}{2} \rfloor},$$

and thus, the total number of indices in the hyperbolic cross is $|H_N^2| = (J+2)2^{J-1} = \mathcal{O}(N \log N)$, whereas $|I_{(N,N)^\top}| = 4^J = N^2$.

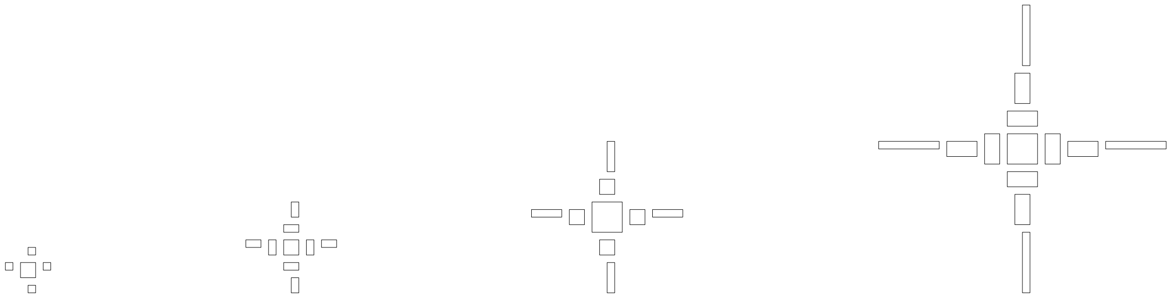


Figure 3.7: Hyperbolic cross H_N^2 for $N = 2^J$, $J = 2, \ldots, 5$.

Bringing our particular partition of H_N^2 into action, we are interested in the computation of

$$f(\boldsymbol{x}_j) = \sum_{\boldsymbol{k}\in H_J^{\text{centre}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j} + \sum_{r=0}^{\lceil \frac{J}{2} \rceil - 1} \sum_{\boldsymbol{k}\in H_{J,r}} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j} \tag{3.15}$$

at arbitrary nodes $\boldsymbol{x}_j \in \mathbb{T}^2$, $j = 0, \dots, M-1$. The first term in (3.15) is simply an NFFT of total size $|H_J^{\text{centre}}| = 2^{2\lfloor \frac{J}{2} \rfloor}$ and is computed with arithmetic complexity of $\mathcal{O}(J2^J + M)$. All other blocks introduce a "nonuniform twiddle factor" but need nevertheless only one bivariate NFFT of total size 2^{J-2} each. Finally, the overall complexity for computing $f(\boldsymbol{x}_j)$ for $j = 0, \dots, M-1$ by means of the bivariate nonequispaced sparse FFT, cf. Algorithm 3.3, is $\mathcal{O}(J^2 2^J + JM)$.

Algorithm 3.3 Bivariate nonequispaced sparse FFT

```
Input:  $J, M \in \mathbb{N}$, $J \geq 2$,
        $\boldsymbol{x}_j \in \mathbb{T}^2$, $j = 0, \dots, M-1$, and $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in H_N^2$, $N = 2^J$.

  Compute by a bivariate NFFT
      $s_j = \sum_{\boldsymbol{k} \in H_J^{\text{centre}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$.

  for $j = r, \dots, \lceil \frac{J}{2} \rceil - 1$ do
    Compute by four bivariate NFFTs
      $s_j = s_j$
      $+ \mathrm{e}^{-2\pi\mathrm{i}\lfloor 3\cdot 2^{J-r-3} \rfloor (\boldsymbol{x}_j)_1} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_{J-r-2}} \hat{f}_{k_0, k_1 + \lfloor 3\cdot 2^{J-r-3} \rfloor} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
      $+ \mathrm{e}^{-2\pi\mathrm{i}\lfloor 3\cdot 2^{J-r-3} \rfloor (\boldsymbol{x}_j)_0} \sum_{\boldsymbol{k} \in \tilde{I}_{J-r-2} \times \tilde{I}_r} \hat{f}_{k_0 + \lfloor 3\cdot 2^{J-r-2} \rfloor, k_1} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
      $+ \mathrm{e}^{+2\pi\mathrm{i}\lceil 3\cdot 2^{J-r-3} \rceil (\boldsymbol{x}_j)_1} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_{J-r-2}} \hat{f}_{k_0, k_1 - \lceil 3\cdot 2^{J-r-3} \rceil} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
      $+ \mathrm{e}^{+2\pi\mathrm{i}\lceil 3\cdot 2^{J-r-3} \rceil (\boldsymbol{x}_j)_0} \sum_{\boldsymbol{k} \in \tilde{I}_{J-r-2} \times \tilde{I}_r} \hat{f}_{k_0 - \lceil 3\cdot 2^{J-r-3} \rceil, k_1} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$.

  end for

Output:      values $s_j \approx f(\boldsymbol{x}_j)$, $j = 0, \dots, M-1$.
Complexity:  $\mathcal{O}(N \log^2 N + M \log N)$.
```

3.2.2 Trivariate case

Next, we consider the three dimensional hyperbolic cross H_N^3, $N = 2^J$, $J \geq 2$. With the help of the auxiliary index sets

$$\begin{aligned} H_{J,0}^{\text{front}} &:= H_{2^J}^2 \times \{0\}\,, & H_{J,r}^{\text{front}} &:= H_{2^{J-r-1}}^2 \times \left(\tilde{I}_{r-1} + \left\lfloor 3 \cdot 2^{r-2} \right\rfloor\right)\,, \\ H_{J,0}^{\text{rear}} &:= H_{2^{J-1}}^2 \times \{-1\}\,, & H_{J,r}^{\text{rear}} &:= H_{2^{J-r-1}}^2 \times \left(\tilde{I}_{r-1} - \left\lceil 3 \cdot 2^{r-2} \right\rceil\right) \end{aligned}$$

for $r = 1, \dots, J-1$, let the hyperbolic cross be partitioned into

$$H_N^3 = \bigcup_{r=0}^{J-1} H_{J,r}^{\text{front}} \cup \bigcup_{r=0}^{J-1} H_{J,r}^{\text{rear}}\,.$$

This partition is illustrated in Figure 3.8 and we obtain furthermore its exact total size $|H_N^3| = 2^{J-3}(J^2 + 7J + 8) = \mathcal{O}(N \log^2 N)$.

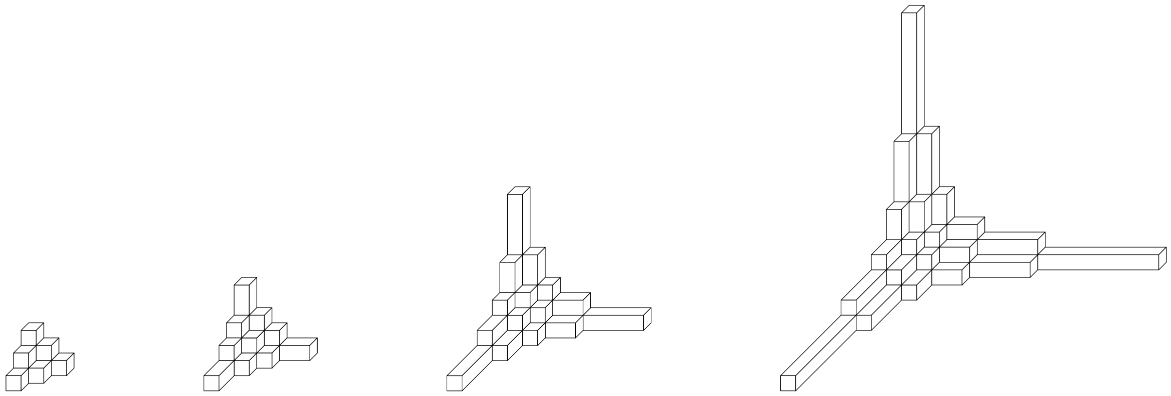


Figure 3.8: Hyperbolic cross H_N^3 for $N = 2^J$, $J = 2, \dots, 5$, only the part k_1, k_2, $k_3 \le 0$ is shown.

The arithmetic complexity of the resulting algorithm is $\mathcal{O}(J^3 2^J + J^2 M)$, where we combine for $r = 1, \dots, J-1$ a bivariate NSFFT for $H^2_{2^{J-r-1}}$ with an univariate NFFT for $\tilde{I}_{r-1}$. Totally, this yields for $r = 1, \dots, J-1$ order $\mathcal{O}(J-r)$ trivariate NFFTs with complexity $\mathcal{O}(J 2^J + M)$ each. Unfortunately, this algorithm has drawbacks: We have to compute NFFTs for $\mathcal{O}(J^2)$ blocks and thus, the (asymptotic) arithmetic complexity for an equal number of nodes and Fourier coefficients $M = |H_N^3|$ is $\mathcal{O}(N \log^4 N)$, i.e., larger than the optimal $\mathcal{O}(N \log^3 N)$. Furthermore, the convolution and evaluation step of the NFFTs for the blocks is the most time consuming part as has been shown in Section 3.1.2.

Hence, we suggest to use a simplified hyperbolic cross $\tilde{H}_N^3$ with $H_N^3 \subset \tilde{H}_N^3 \subset I_{(N,N,N)^\top}$ which is easily partitioned into only $\mathcal{O}(J)$ blocks but has a total cardinality of $|\tilde{H}_N^3| = \mathcal{O}(N^{\frac{3}{2}})$. Analogously to the two dimensional case, we define now for the directions *top, left, front, bottom, right, rear,* and for $r = 0, \dots, \lceil \frac{J}{2} \rceil - 1$ auxiliary index sets of the form

$$\begin{aligned} \tilde{H}_{J,r}^{\mathrm{top}} &:= \left(\tilde{I}_{J-r-2} + \left\lfloor 3 \cdot 2^{J-r-3} \right\rfloor \right) \times \tilde{I}_r \times \tilde{I}_r, \; \dots \,, \\ \tilde{H}_{J,r} &:= \tilde{H}_{J,r}^{\mathrm{left}} \cup \tilde{H}_{J,r}^{\mathrm{right}} \cup \tilde{H}_{J,r}^{\mathrm{top}} \cup \tilde{H}_{J,r}^{\mathrm{bottom}} \cup \tilde{H}_{J,r}^{\mathrm{front}} \cup \tilde{H}_{J,r}^{\mathrm{rear}} \,, \end{aligned}$$

a centre block $\tilde{H}_J^{\mathrm{centre}}$ and thus, finally the modified three dimensional hyperbolic index sets $\tilde{H}_N^3$, cf. Figure 3.9,

$$\tilde{H}_J^{\mathrm{centre}} := \tilde{I}_{\lfloor \frac{J}{2} \rfloor + 1} \times \tilde{I}_{\lfloor \frac{J}{2} \rfloor + 1} \times \tilde{I}_{\lfloor \frac{J}{2} \rfloor + 1} \,, \qquad \tilde{H}_N^3 := \tilde{H}_J^{\mathrm{centre}} \cup \bigcup_{r=0}^{\lceil \frac{J}{2} \rceil - 1} \tilde{H}_{J,r} \,.$$

The cardinalities of these index sets are given by

$$\begin{gathered} \left| \tilde{H}_{J,r}^{\mathrm{top}} \right| = \dots = 2^{J+r} \,, \quad \left| \tilde{H}_J^{\mathrm{centre}} \right| = 2^{3(\lfloor \frac{J}{2} \rfloor + 1)} \,, \\ |\tilde{H}_N^3| = 2^J 6 (2^{\lceil \frac{J}{2} \rceil + 1} - 1) + 2^{3(\lfloor \frac{J}{2} \rfloor + 1)} = \mathcal{O}\left(N^{\frac{3}{2}} \right) , \end{gathered}$$

whereas $|I_{(N,N,N)^\top}| = N^3$.

Processing each of the blocks by one trivariate NFFT, we obtain the trivariate nonequispaced sparse FFT, cf. Algorithm 3.4, with arithmetic complexity $\mathcal{O}(J 2^{\frac{3}{2} J} + JM)$.

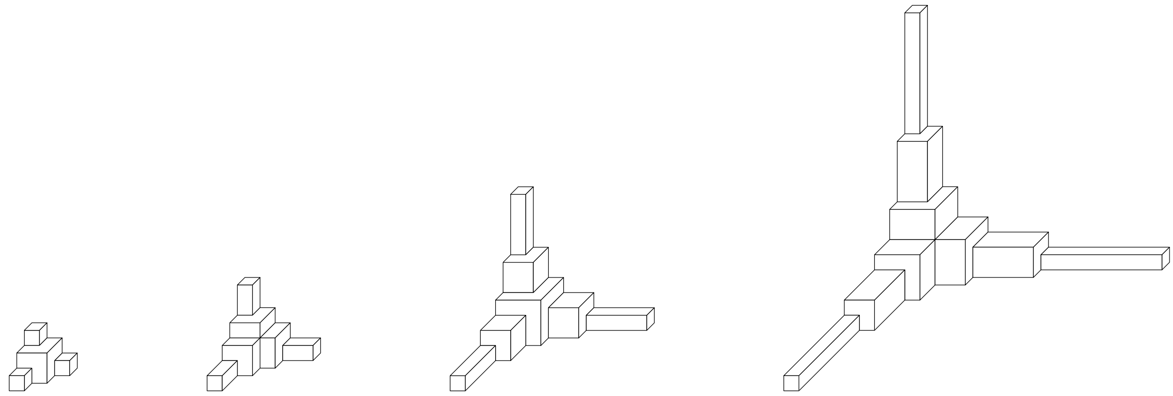


Figure 3.9: Simplified hyperbolic cross $\tilde{H}_N^3$ for $N = 2^J$, $J = 2, \dots, 5$, only the part k_1, k_2, $k_3 \le 0$ is shown.

Algorithm 3.4 Trivariate nonequispaced sparse FFT

```
Input:  $J, M \in \mathbb{N}$, $J \ge 2$,
        $\boldsymbol{x}_j \in \mathbb{T}^3$, $j = 0, \dots, M-1$, and $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in H_N^3$, $N = 2^J$.

Compute by a trivariate NFFT
    $s_j = \sum_{\boldsymbol{k} \in \tilde{H}_J^{\mathrm{centre}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}.$

for $j = r, \dots, \lceil \frac{J}{2} \rceil - 1$ do
    Compute by six trivariate NFFTs
        $s_j = s_j$
        $+ \mathrm{e}^{+2\pi\mathrm{i}\lceil 3\cdot 2^{J-r-3} \rceil (\boldsymbol{x}_j)_0} \sum_{\boldsymbol{k} \in \tilde{I}_{J-r-2} \times \tilde{I}_r \times \tilde{I}_r} \hat{f}_{k_0 - \lceil 3\cdot 2^{J-r-3} \rceil, k_1, k_2} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
        $+ \mathrm{e}^{+2\pi\mathrm{i}\lceil 3\cdot 2^{J-r-3} \rceil (\boldsymbol{x}_j)_1} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_{J-r-2} \times \tilde{I}_r} \hat{f}_{k_0, k_1 - \lceil 3\cdot 2^{J-r-3} \rceil, k_2} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
        $+ \mathrm{e}^{+2\pi\mathrm{i}\lceil 3\cdot 2^{J-r-3} \rceil (\boldsymbol{x}_j)_2} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_r \times \tilde{I}_{J-r-2}} \hat{f}_{k_0, k_1, k_2 - \lceil 3\cdot 2^{J-r-3} \rceil} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
        $+ \mathrm{e}^{-2\pi\mathrm{i}\lfloor 3\cdot 2^{J-r-3} \rfloor (\boldsymbol{x}_j)_0} \sum_{\boldsymbol{k} \in \tilde{I}_{J-r-2} \times \tilde{I}_r \times \tilde{I}_r} \hat{f}_{k_0 + \lfloor 3\cdot 2^{J-r-3} \rfloor, k_1, k_2} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
        $+ \mathrm{e}^{-2\pi\mathrm{i}\lfloor 3\cdot 2^{J-r-3} \rfloor (\boldsymbol{x}_j)_1} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_{J-r-2} \times \tilde{I}_r} \hat{f}_{k_0, k_1 + \lfloor 3\cdot 2^{J-r-3} \rfloor, k_2} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}$
        $+ \mathrm{e}^{-2\pi\mathrm{i}\lfloor 3\cdot 2^{J-r-3} \rfloor (\boldsymbol{x}_j)_2} \sum_{\boldsymbol{k} \in \tilde{I}_r \times \tilde{I}_r \times \tilde{I}_{J-r-2}} \hat{f}_{k_0, k_1, k_2 + \lfloor 3\cdot 2^{J-r-3} \rfloor} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}.$
end for

Output:      values $s_j \approx f(\boldsymbol{x}_j)$, $j = 0, \dots, M-1$.
Complexity:  $\mathcal{O}(N^{\frac{3}{2}} \log N + M \log N)$.
```

3.2.3 Numerical experiments

In the following, we compare Algorithm 3.3 and Algorithm 3.4 with the straightforward summation of (3.14), denoted by NSDFT (nonequispaced sparse discrete Fourier transform) and with the ordinary NFFT on the full index set, where all Fourier coefficients not supported on the set $H_N^2 \subset I_{(N,N)^\top}$ and $\tilde{H}_N^3 \subset I_{(N,N,N)^\top}$ are set to zero, respectively. All tests use the Gaussian window function in conjunction with its fast computation scheme `FG_PSI` and a fixed oversampling factor $\sigma = 2$.

Example 3.15. We will first examine the total error that is caused by the usage of the NFFT on the blocks. The achieved accuracy $E_\infty := \|\boldsymbol{f} - \boldsymbol{s}\|_\infty / \|\hat{\boldsymbol{f}}\|_1$, see also Lemma 3.5, is shown in Figure 3.10 to decay exponentially fast as the cut-off m increases.

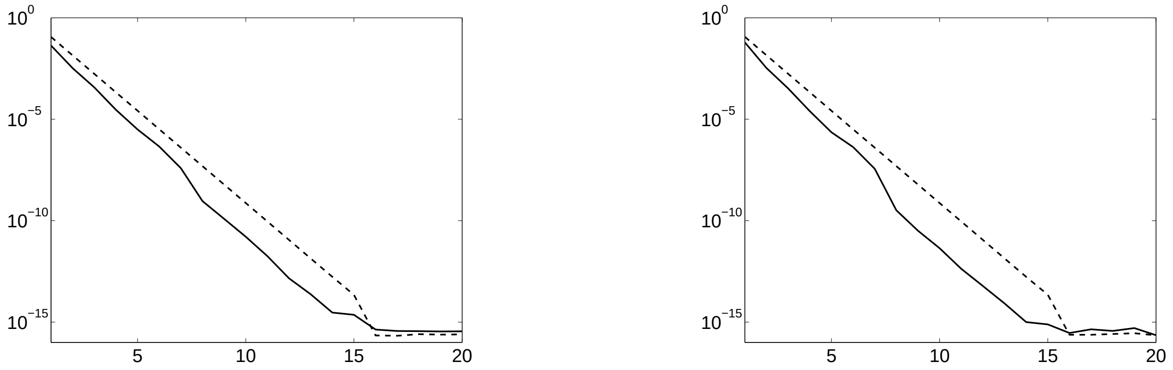


Figure 3.10: The error E_∞ for $J = 6$ and increasing cut-off $m = 1, \dots, 20$; NSFFT (solid), adjoint NSFFT (dashed); bivariate case $d = 2$ (left), trivariate case $d = 3$ (right).

Example 3.16. Next, we are interested in computation times and memory usage, where we choose the number of nodes equal to the number of Fourier coefficients, i.e., $M = |H_N^2|$ and $M = |\tilde{H}_N^3|$, respectively. We compare the computation time and the memory usage of the NSFFT, the NSDFT, and the ordinary NFFT (both NFFT schemes with a cut-off parameter $m = 4$). The computation time of all three algorithms is shown in Figure 3.11 (top) and the memory usage of all three algorithms is shown in the bottom row. As expected the NSFFT outperforms the other algorithms, e.g., for $d = 2$, $J = 12$, and $M = |H_{4096}^2| = 28672$ nodes, the computation time is less than 3 seconds for the NSFFT compared to 117 seconds for the NSDFT.

3.2.4 Concluding remarks

Fast algorithms for the evaluation of trigonometric polynomials on the hyperbolic cross at arbitrary nodes and for the corresponding adjoint transforms have been constructed. The numerical results, cf. Example 3.15 and Example 3.16, show the reliability and superiority of the proposed algorithms with respect to computation time and the total used memory. In summary, Table 3.4 shows a comparison of the computational requirements of the proposed schemes.

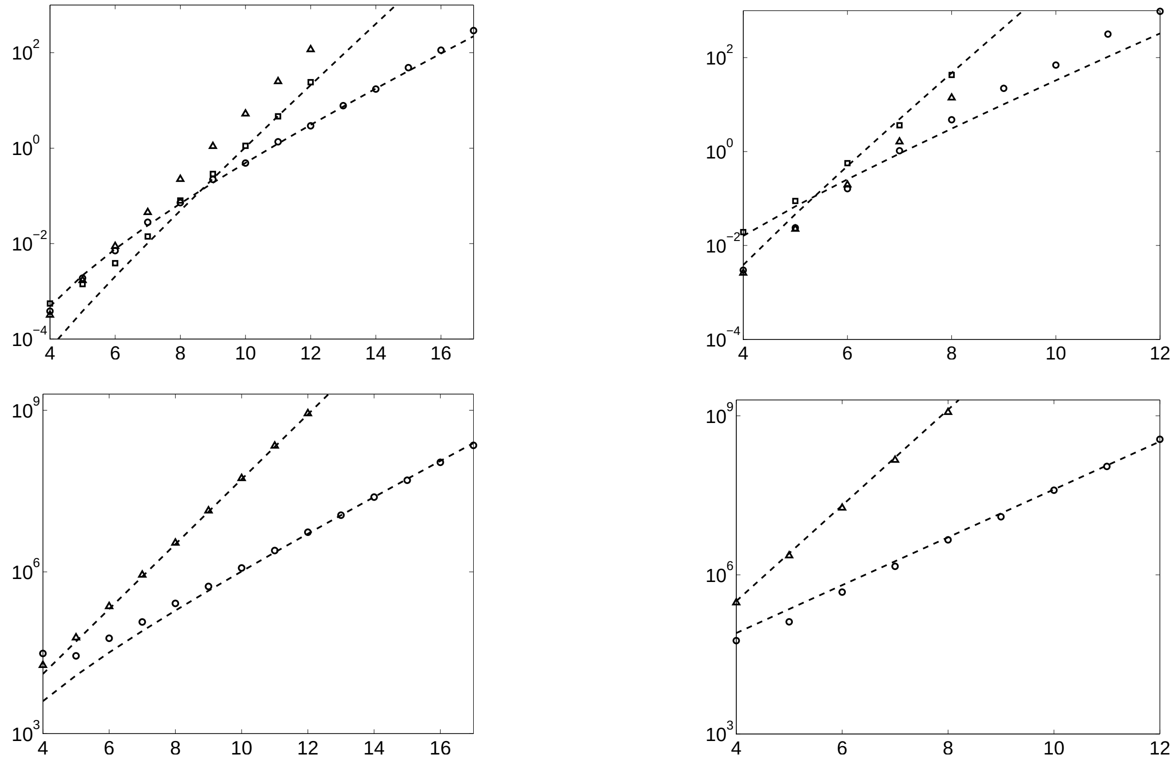


Figure 3.11: Computation time in seconds (top) and total used memory in bytes (bottom) for $J = 4, \ldots, 17$, $d = 2$ (left) and for $J = 4, \ldots, 12$, $d = 3$ (right); NSFFT (circle), NSDFT (triangle), and NFFT (square). Expected orders of magnitude, discussed subsequently, cf. Table 3.4, are shown dashed.

	$d=2$		$d=3$	
Algorithm	Time	Memory	Time	Memory
NFFT	$N^2 \log N$	N^2	$N^3 \log N$	N^3
NSDFT	$N^2 \log^2 N$	$N \log N$	N^3	$N^{\frac{3}{2}}$
NSFFT	$N \log^2 N$	$N \log N$	$N^{\frac{3}{2}} \log N$	$N^{\frac{3}{2}}$

Table 3.4: Order of magnitude for computation time and total used memory.

3.3 Miscellaneous transforms and implementation

Before discussing the NFFT software package [KP06b], we would like to comment on further variations and generalisations of the nonequispaced fast Fourier transform. More detailed, we give subsequently the basic ideas for the nonequispaced FFT on the sphere and for processing real valued data, respectively.

3.3.1 Fast spherical Fourier transform

The nonequispaced discrete spherical Fourier transform (2.9) computes

$$f\left(\boldsymbol{\xi}_j\right) = \sum_{\boldsymbol{k}\in J_N} \hat{f}_{\boldsymbol{k}} Y_{\boldsymbol{k}}\left(\boldsymbol{\xi}_j\right) = \sum_{n=-N}^{N}\sum_{k=|n|}^{N} \hat{f}_k^n P_k^{|n|}\left(\cos\vartheta_j\right) \mathrm{e}^{\mathrm{i}n\varphi_j} \tag{3.16}$$

for given $\hat{f}_{\boldsymbol{k}} = \hat{f}_k^n \in \mathbb{C}$ and $(\vartheta_j, \varphi_j) \sim \boldsymbol{\xi}_j \in \mathbb{S}^2$, $j = 0, \ldots, M-1$. A fundamental difference, when computing with spherical harmonics, is the appearance of the associated Legendre functions and hence, the need for their efficient treatment. More precisely, we are interested in the change of basis, transforming the associated Fourier-Legendre coefficients $\hat{\boldsymbol{f}}^n \in \mathbb{C}^{N-|n|+1}$ in

$$h_n\left(\cos\vartheta\right) := \sum_{k=|n|}^{N} \hat{f}_k^n P_k^{|n|}\left(\cos\vartheta\right)$$

to ordinary Fourier coefficients $\hat{\boldsymbol{c}}^n \in \mathbb{C}^{2N+1}$ in

$$h_n\left(\cos\vartheta\right) = \sum_{k=-N}^{N} \hat{c}_k^n \mathrm{e}^{\mathrm{i}k\vartheta}.$$

Fast schemes for this task have been constructed in [DHR96, PST98a]. Based on the fact that h_n is a polynomial of degree N for even n and $(1-\cdot^2)^{-1}h_n$ is a polynomial of degree $N-1$ for odd n, we use a generalised three term recurrence relation in conjunction with fast polynomial multiplication schemes and a tree-like organisation to obtain a fast Legendre function transform, see [KP03, Lemma 4.1, Algorithm 4.1] for details.

Since a spherical polynomial is almost separable in the sense that equation (3.16) can be written as

$$f\left(\boldsymbol{\xi}_j\right) = f\left(\vartheta_j, \varphi_j\right) = \sum_{n=-N}^{N} h_n\left(\cos\vartheta_j\right) \mathrm{e}^{\mathrm{i}n\varphi_j},$$

one uses this fast Legendre function transform to obtain Algorithm 3.5 for the fast computation of the discrete spherical Fourier transform, see also [KP03, Algorithm 5.3]. Recently, the corresponding adjoint NFSFT has been obtained in [Kei05, KP06a], and we note that all preceding fast spherical Fourier transforms in [DH94, PST98b, Moh99, HKMR03, ST02, RT06] were restricted to particular lattices on the sphere.

3.3.2 Real trigonometric transforms

In case we deal with real-valued functions only, their expansion into Fourier-sine and Fourier-cosine series with real-valued coefficients becomes possible. For a review of discrete (co-)sine transforms (DCT/DST) and fast algorithms thereof see, e.g., [Pot98, pp. 13], their generalisation to nonequispaced nodes appears first in [FP05, Kla05]. The *nonequispaced fast cosine*

```
Algorithm 3.5 Nonequispaced fast spherical Fourier transform
Input:  $N, M \in \mathbb{N}$
        $(\vartheta_j, \varphi_j) \sim \boldsymbol{\xi}_j \in \mathbb{S}^2,\ j = 0, \dots, M-1$, and $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in J_N$.

  for $n = -N, \dots, N$ do
    Compute by the fast Legendre function transform the coefficients $\hat{c}_k^n \in \mathbb{C}$ in
        $h_n(\cos\vartheta) = \sum_{k=-N}^{N} \hat{c}_k^n \mathrm{e}^{\mathrm{i}k\vartheta}$
    from the given coefficients $\hat{f}_k^n$ in
        $h_n(\cos\vartheta) = \sum_{k=|n|}^{N} \hat{f}_k^n P_k^{|n|}(\cos\vartheta).$
  end for
  Compute by Algorithm 3.1 the sum
        $s_j = \sum_{\boldsymbol{k} \in I_{(2N+1,2N+1)^\top}} \hat{c}_{\boldsymbol{k}} \mathrm{e}^{\mathrm{i}(k_0\vartheta_j + k_1\varphi_j)}.$

Output:      values $s_j \approx f(\boldsymbol{\xi}_j),\ j = 0, \dots, M-1$.
Complexity:  $\mathcal{O}(N^2 \log^2 N + M)$.
```

transform (NFCT) and *nonequispaced fast sine transform* (NFST) compute approximations of

$$f(x_j) = \sum_{k=0}^{N-1} \hat{f}_k \cos(2\pi k x_j)$$

and

$$f(x_j) = \sum_{k=1}^{N-1} \hat{f}_k \sin(2\pi k x_j)$$

for given coefficients $\hat{\boldsymbol{f}} \in \mathbb{R}^N$ and $x_j \in [0, \frac{1}{2}]$, $j = 0, \dots, M-1$, respectively. Multivariate transforms for real-valued data are built upon products of univariate (co-)sine polynomials, their application include real versions of the Fourier transform on the sphere and the Fourier (co-)sine transform on the hyperbolic cross [FKP06, Section 4].

3.3.3 Software library

Further discrete transforms included in our current implementation [KP06b] are the time and frequency nonequispaced FFT [Ste98, PST01, LG05], specialised transforms in magnetic resonance imaging [EKPar], and the polar and pseudo polar FFT [PS01a, ACD$^+$06, FKPar]. Another recent generalisation of the FFT on the rotation group [MR95] for nonequispaced nodes has been obtained in [Vol06] and will follow soon.

The fast Gauss transforms of Chapter 4 are also part of our subroutine library (“fastgauss” and “fastsumS2”). Moreover, the inverse NFFTs of Chapter 5 are referred to as “solver”, and its application in magnetic resonance imaging, cf. Section 5.3.2, is included in “mri_appl”. In summary, the following tree structure in Figure 3.12 gives an overview of the most recent version our software package.

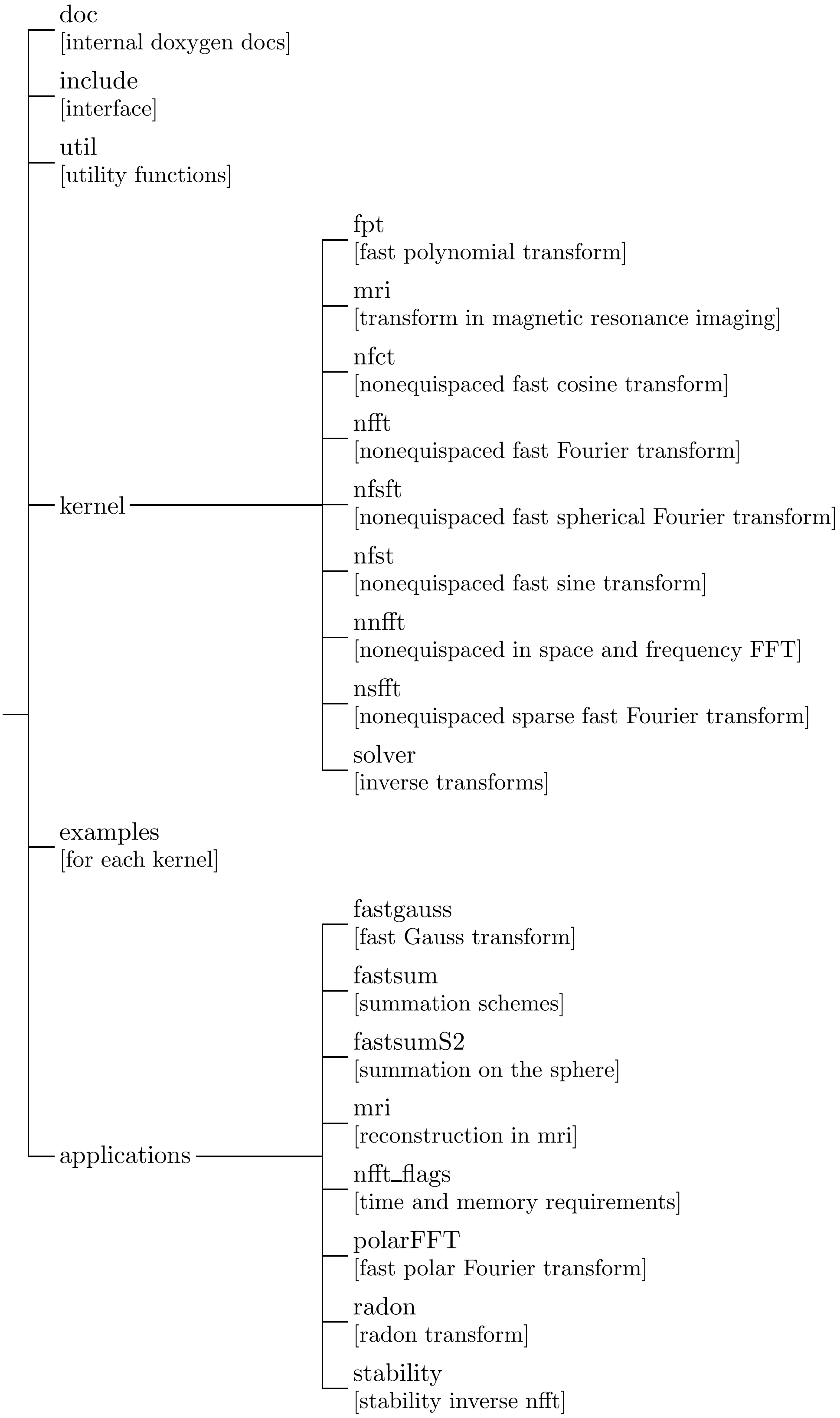

doc
[internal doxygen docs]
include
[interface]
util
[utility functions]
kernel
fpt
[fast polynomial transform]
mri
[transform in magnetic resonance imaging]
nfct
[nonequispaced fast cosine transform]
nfft
[nonequispaced fast Fourier transform]
nfsft
[nonequispaced fast spherical Fourier transform]
nfst
[nonequispaced fast sine transform]
nnfft
[nonequispaced in space and frequency FFT]
nsfft
[nonequispaced sparse fast Fourier transform]
solver
[inverse transforms]
examples
[for each kernel]
applications
fastgauss
[fast Gauss transform]
fastsum
[summation schemes]
fastsumS2
[summation on the sphere]
mri
[reconstruction in mri]
nfft_flags
[time and memory requirements]
polarFFT
[fast polar Fourier transform]
radon
[radon transform]
stability
[stability inverse nfft]

Figure 3.12: Directory structure of the NFFT software library [KP06b].

4

Fast Gauss transform

The objective of this chapter is the discrete Gauss transform and the construction of fast algorithms for its efficient computation in various settings. In its most natural formulation, the given task reads as follows: Given complex coefficients $\alpha_l \in \mathbb{C}$ and source nodes $\boldsymbol{y}_l \in \mathbb{R}^d$, $l = 0, \ldots, L-1$, our goal consists in the fast evaluation of the sum

$$g\left(\boldsymbol{x}\right) = \sum_{l=0}^{L-1} \alpha_l \mathrm{e}^{-\sigma \|\boldsymbol{x}-\boldsymbol{y}_l\|_2^2} \tag{4.1}$$

at the target nodes $\boldsymbol{x}_j \in \mathbb{R}^d$, $j = 0, \ldots, M-1$, where $\sigma > 0$ denotes a real parameter.

Owing to the separation of the nodes, the expansion into trigonometric polynomials allows us the construction of fast algorithms. In Section 4.1 we deal with the univariate case only. We construct an approximate *fast Gauss transform* (FGT) capable of dealing with nonequispaced nodes and a complex parameter σ. Our algorithm is based on the nonequispaced FFT and requires only $\mathcal{O}(L+M)$ arithmetic operations. We prove error bounds to justify this arithmetic complexity for bounded source and target nodes in dependence on the parameter σ. Section 4.2 is devoted to the more general fast spherical Gauss transforms, i.e., the source and target nodes are elements of the two-dimensional unit sphere. The error estimates and the computational requirements of the proposed schemes are validated in numerical experiments. Finally, Section 4.3 discusses a few generalisations and gives reference to previous work on the fast Gauss transform and related approaches. Essential parts of this chapter are published in [KPS06, KKP06].

4.1 Univariate Gaussian

For $p > 0$ and $M, L \in \mathbb{N}$, let $\mathcal{X} = \{x_j \in [-\frac{p}{4}, \frac{p}{4}],\ j = 0, \ldots, M-1\}$, $\mathcal{Y} = \{y_l \in [-\frac{p}{4}, \frac{p}{4}],\ l = 0, \ldots, L-1\}$, and $\alpha_l \in \mathbb{C}$, $l = 0, \ldots, L-1$, be given. An algorithm for the fast computation of

$$g\left(x_j\right) = \sum_{l=0}^{L-1} \alpha_l K\left(x_j - y_l\right), \qquad j = 0, \ldots, M-1, \tag{4.2}$$

where K are special real–valued kernels has been proposed for one dimension in [PS03]. We want to apply this method to the following kernel.

Definition 4.1. *For*

$$\sigma = a + b\mathrm{i} = |\sigma|\mathrm{e}^{\mathrm{i}\varphi}\,, \qquad a > 0,\, b \in \mathbb{R},\, \varphi := \arctan\frac{b}{a} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

the univariate Gaussian kernel $K_\sigma : \mathbb{R} \to \mathbb{C}$ *is given by*

$$K_\sigma\left(x\right) := \mathrm{e}^{-\sigma x^2}.$$

We approximate the kernel K_σ in a first step by a periodic function and in a second step by a trigonometric polynomial as follows.

Lemma 4.2. *Let $p > 0$ and the univariate Gaussian K_σ be given, then its periodisation*

$$\tilde{K}_\sigma\left(x\right) := \sum_{r\in\mathbb{Z}} K_\sigma(x - pr) \tag{4.3}$$

possesses the Fourier series $\tilde{K}_\sigma(x) = \sum_{k\in\mathbb{Z}} \hat{w}_k \mathrm{e}^{-2\pi\mathrm{i}kx/p}$ with Fourier coefficients

$$\hat{w}_k = \frac{\sqrt{\pi}}{p\sqrt{\sigma}}\,\mathrm{e}^{-k^2\pi^2/(\sigma p^2)} \tag{4.4}$$

where $\sqrt{\sigma} = |\sigma|^{\frac{1}{2}}\mathrm{e}^{\mathrm{i}\frac{\varphi}{2}}$.

Proof. *Using Theorem 2.3, the Fourier coefficients are calculated by*

$$\hat{w}_k = \frac{1}{p}\int_{-\frac{p}{2}}^{\frac{p}{2}} \tilde{K}_\sigma\left(x\right)\mathrm{e}^{-2\pi\mathrm{i}kx/p}\mathrm{d}x = \frac{1}{p}\int_{-\infty}^{\infty} \mathrm{e}^{-\sigma x^2}\mathrm{e}^{-2\pi\mathrm{i}kx/p}\mathrm{d}x = \frac{\sqrt{\pi}}{p\sqrt{\sigma}}\,\mathrm{e}^{-k^2\pi^2/(\sigma p^2)},$$

see also [AS72, equation 7.4.6]. □

Definition 4.3. *For $p > 0$, $N \in 2\mathbb{N}$, and the coefficients $\hat{w}_k$ in* (4.4), *we define the kernel $K_{\sigma,N}$, see also Definition 2.10, by*

$$K_{\sigma,N}\left(x\right) := \sum_{k\in I_N} \hat{w}_k \mathrm{e}^{-2\pi\mathrm{i}kx/p}. \tag{4.5}$$

Moreover, let the function $g_N : [-\frac{p}{4}, \frac{p}{4}] \to \mathbb{C}$ be given by

$$g_N\left(x\right) := \sum_{l=0}^{L-1} \alpha_l K_{\sigma,N}\left(x - y_l\right). \tag{4.6}$$

4.1.1 Derivation of Algorithm 4.1

Thus, we approximate g in (4.2) by the degenerate expansion g_N in (4.6). Substituting (4.5) into (4.6) and changing the order of summation yields

$$g_N\left(x_j\right) = \sum_{k\in I_N} \hat{w}_k \left(\sum_{l=0}^{L-1} \alpha_l \mathrm{e}^{2\pi\mathrm{i}ky_l/p}\right) \mathrm{e}^{-2\pi\mathrm{i}kx_j/p}\,.$$

The expression in the inner brackets can be computed by an adjoint NFFT in $\mathcal{O}(L + N\log N)$ arithmetic operations. This is followed by N multiplications with the coefficients $\hat{w}_k$ and completed by an NFFT to compute the outer sum in $\mathcal{O}(M + N\log N)$ arithmetic operations. We will prove that the polynomial degree N depends solely on the desired accuracy of our algorithm and on the complex parameter σ, but not on the numbers L and M. Thus, the overall arithmetic complexity of our algorithm is $\mathcal{O}(L + M)$. In particular, this performance does not depend on the distribution of the nodes x_j and y_l.

Remark 4.4. *In matrix-vector notation the original problem* (4.2) *reads* $\boldsymbol{g} = \boldsymbol{K}_\sigma \boldsymbol{\alpha}$, *where*

$$\begin{aligned} \boldsymbol{g} &:= \left(g\left(x_j\right)\right)_{j=0,\ldots,M-1} \in \mathbb{C}^M, \\ \boldsymbol{K}_\sigma &:= \left(K_\sigma\left(x_j - y_l\right)\right)_{j=0,\ldots,M-1;l=0,\ldots,L-1} \in \mathbb{C}^{M\times L}, \\ \boldsymbol{\alpha} &:= \left(\alpha_l\right)_{l=0,\ldots,L-1} \in \mathbb{C}^L. \end{aligned}$$

Our approach is a particular approximation to the matrix $\boldsymbol{K}_\sigma$ *of at most rank* N *and takes the form* $\boldsymbol{g}_N = \boldsymbol{A}_\mathcal{X} \hat{\boldsymbol{W}} \boldsymbol{A}_\mathcal{Y}^\mathsf{H} \boldsymbol{\alpha}$ *with*

$$\begin{aligned} \boldsymbol{g}_N &:= \left(g_N\left(x_j\right)\right)_{j=0,\ldots,M-1} \in \mathbb{C}^M, \\ \boldsymbol{A}_\mathcal{X} &:= \left(\mathrm{e}^{-2\pi\mathrm{i}kx_j/p}\right)_{j=0,\ldots,M-1,k\in I_N} \in \mathbb{C}^{M\times N}, \\ \hat{\boldsymbol{W}} &:= \operatorname{diag}\hat{\boldsymbol{w}},\ \hat{\boldsymbol{w}} := \left(\hat{w}_k\right)_{k\in I_N} \in \mathbb{C}^N, \\ \boldsymbol{A}_\mathcal{Y} &:= \left(\mathrm{e}^{-2\pi\mathrm{i}ky_l/p}\right)_{l=0,\ldots,L-1,k\in I_N} \in \mathbb{C}^{L\times N}. \end{aligned}$$

In summary, we propose Algorithm 4.1.

Algorithm 4.1 *Fast Gauss transform*

```
Input:  $L,\ M \in \mathbb{N},\ p > 0,\ N \in 2\mathbb{N}$,
        $y_l \in [-\frac{p}{4}, \frac{p}{4}],\ \alpha_l \in \mathbb{C},\ l = 0,\ldots,L-1$,
        $x_j \in [-\frac{p}{4}, \frac{p}{4}],\ j = 0,\ldots,M-1$.

  Compute $\tilde{\boldsymbol{\alpha}} = \boldsymbol{A}_\mathcal{Y}^\mathsf{H}\boldsymbol{\alpha}$ by Algorithm 3.2.

  Evaluate $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{W}}\tilde{\boldsymbol{\alpha}}$.

  Compute $\boldsymbol{g}_N = \boldsymbol{A}_\mathcal{X}\hat{\boldsymbol{\alpha}}$ by Algorithm 3.1.

Output:      $\boldsymbol{g}_N$ approximating $\boldsymbol{g} = \boldsymbol{K}_\sigma\boldsymbol{\alpha}$.
Complexity:  $\mathcal{O}(M + L + N \log N)$.
```

Remark 4.5. *Replacing the NFFT algorithms by their slow versions, i.e., by the nonequispaced discrete Fourier transform (NDFT) and its adjoint which need* $\mathcal{O}(LN)$ *and* $\mathcal{O}(MN)$ *arithmetic operations for the multiplications with the matrices* $\boldsymbol{A}_\mathcal{Y}^\mathsf{H}$ *and* $\boldsymbol{A}_\mathcal{X}$, *respectively, yields an* $\mathcal{O}(M + L)$ *algorithm, too. Nevertheless, the fast algorithms are the key for applications to large data sets since they decouple the degree* N *of the expansion from the number of nodes* L *and* M.

4.1.2 Error estimates

Beyond the well-known errors appearing in the NFFT computations, our algorithm produces the following error.

Lemma 4.6. *The proposed approximations obey the following properties. The periodisation* $\tilde{K}_\sigma$ *in* (4.3) *raises for* $x \in [-\frac{p}{2}, \frac{p}{2}]$ *the error*

$$\left|K_\sigma(x) - \tilde{K}_\sigma(x)\right| \le 2\mathrm{e}^{-\frac{ap^2}{4}} \left(1 + \frac{1}{ap^2}\right).$$

The approximation by the trigonometric kernel $K_{\sigma,N}$ *in* (4.5) *raises for* $x \in [-\frac{p}{2}, \frac{p}{2}]$ *the error*

$$\left|\tilde{K}_\sigma(x) - K_{\sigma,N}(x)\right| \le \frac{\sqrt{\pi}}{p\sqrt{|\sigma|}} \mathrm{e}^{-\frac{N^2\pi^2\cos^2\varphi}{4p^2a}} \left(1 + \frac{2p^2a}{N\pi^2\cos^2\varphi}\right).$$

Proof. Using Lemma 4.2, Definition 4.3, and the fact $|\mathrm{e}^{a+\mathrm{i}b}| = \mathrm{e}^a$ yields the following. The periodisation obeys

$$\begin{aligned}\left|K_\sigma(x) - \tilde{K}_\sigma(x)\right| &\le \sum_{r\in\mathbb{Z}\setminus\{0\}} \left|\mathrm{e}^{-\sigma(x-pr)^2}\right| \\ &\le 2\sum_{r=1}^{\infty} \mathrm{e}^{-\frac{ap^2(2r-1)^2}{4}} \\ &\le 2\mathrm{e}^{-\frac{ap^2}{4}} + \int_1^\infty \mathrm{e}^{-\frac{ap^2}{4}r^2}\,\mathrm{d}r\,.\end{aligned}$$

Using the first inequality in Theorem 2.5, we estimate similarly

$$\begin{aligned}\frac{p\sqrt{|\sigma|}}{\sqrt{\pi}}\left|\tilde{K}_\sigma(x) - K_{\sigma,N}(x)\right| &\le \left|\mathrm{e}^{-\frac{N^2\pi^2}{4p^2\sigma}}\right| + 2\sum_{k=\frac{N}{2}+1}^{\infty} \left|\mathrm{e}^{-\frac{k^2\pi^2}{p^2\sigma}}\right| \\ &\le \mathrm{e}^{-\frac{N^2\pi^2\cos^2\varphi}{4p^2a}} + 2\sum_{k=\frac{N}{2}+1}^{\infty} \mathrm{e}^{-\frac{k^2\pi^2\cos^2\varphi}{p^2a}} \\ &\le \mathrm{e}^{-\frac{N^2\pi^2\cos^2\varphi}{4p^2a}} + 2\int_{\frac{N}{2}}^{\infty} \mathrm{e}^{-\frac{k^2\pi^2\cos^2\varphi}{p^2a}}\,\mathrm{d}k.\end{aligned}$$

In both cases, applying (3.6) from the proof of Lemma 3.4 yields the assertion. □

Theorem 4.7. *For* $N \in 2\mathbb{N}$, *the approximation* g_N *in* (4.6) *obeys the uniform error estimate*

$$\frac{\|g - g_N\|_\infty}{\|\boldsymbol{\alpha}\|_1} \le 2\mathrm{e}^{-\frac{ap^2}{4}}\left(1 + \frac{1}{ap^2}\right) + \frac{\sqrt{\pi}}{p\sqrt{|\sigma|}}\mathrm{e}^{-\frac{N^2\pi^2\cos^2\varphi}{4p^2a}}\left(1 + \frac{2p^2a}{N\pi^2\cos^2\varphi}\right)$$

within the interval $[-\frac{p}{4}, \frac{p}{4}]$. *In particular, the matrix approximation obeys*

$$\left\|\boldsymbol{K}_\sigma - \boldsymbol{A}_\mathcal{X}\hat{\boldsymbol{W}}\boldsymbol{A}_\mathcal{Y}\right\|_2 \le \sqrt{LM}\left(c_a^{-p^2} + c_\sigma^{-\frac{N^2}{p^2}}\right)$$

with constants $c_a, c_\sigma > 1$ *depending solely on the Gaussian parameter* $\sigma \in \mathbb{C}$.

Proof. Using Definition 4.3, the estimates

$$\begin{aligned}\|g - g_N\|_\infty &\le \|\boldsymbol{\alpha}\|_1 \|K_\sigma - K_{\sigma,N}\|_\infty \\ &\le \|\boldsymbol{\alpha}\|_1\left(\left\|K_\sigma - \tilde{K}_\sigma\right\|_\infty + \left\|\tilde{K}_\sigma - K_{\sigma,N}\right\|_\infty\right)\end{aligned}$$

and Lemma 4.6 yield the first assertion. We obtain the spectral norm estimate by applying Hölders inequality, i.e., we proceed as in the proof of Corollary 3.6. □

Corollary 4.8. Let $\sigma = a + b\mathrm{i} = |\sigma|\mathrm{e}^{\mathrm{i}\varphi} \in \mathbb{C}$ with bounded phase $\cos\varphi > 10^{-2}$, i.e. $|\varphi| \leq 89.4°$ is sufficient, and a target accuracy $\varepsilon < \frac{1}{2}$ be given. Choosing $p = \frac{2}{\sqrt{a}}\sqrt{|\ln\frac{\varepsilon}{5}|}$ and a polynomial degree $N > \frac{4}{\pi\cos\varphi}|\ln\frac{\varepsilon}{5}|$ yields the error estimate

$$\frac{\|g - g_N\|_\infty}{\|\boldsymbol{\alpha}\|_1} < \varepsilon$$

within the previous Theorem 4.7.

Proof. We estimate the first part of the error by

$$2\mathrm{e}^{-\frac{ap^2}{4}}\left(1 + \frac{1}{ap^2}\right) < \frac{2\varepsilon}{5}\left(1 + \frac{1}{4}\right) = \frac{\varepsilon}{2}$$

and the second one by

$$\frac{\sqrt{\pi}}{p\sqrt{|\sigma|}}\mathrm{e}^{-\frac{N^2\pi^2\cos^2\varphi}{4p^2a}}\left(1 + \frac{2p^2a}{N\pi^2\cos^2\varphi}\right) \leq \frac{\sqrt{\pi\cos\varphi}}{2\sqrt{|\ln\frac{\varepsilon}{5}|}}\frac{\varepsilon}{5}\left(1 + \frac{2}{\pi\cos\varphi}\right) < \frac{\varepsilon}{2}\,.$$

□

The error estimate of Theorem 4.7 consists of two parts. In Corollary 4.8 these two components are balanced. As the corollary also reveals, we can cope with the first summand solely by choosing a suited period p with respect to the target accuracy and the real part a of the Gaussian parameter σ. Then the cut-off degree N does only depend on the target accuracy and on the phase φ of the parameter - loosely speaking only on the chirp.

4.1.3 Numerical experiments

Subsequently, we illustrate the error estimate of Theorem 4.7 in Example 4.9 and test the accuracy and the computation time of Algorithm 4.1 in Example 4.10 and Example 4.11, respectively. The accuracy of the algorithm is measured by

$$E_\infty := \frac{\max_{j=0,\dots,M-1}|g(x_j) - g_N(x_j)|}{\sum_{l=0}^{L-1}|\alpha_l|} \tag{4.7}$$

which is of course, bounded sharply from above in Theorem 4.7.

Example 4.9. The behaviour of the error is illustrated in Figure 4.1. The right-hand sides of the figure show the level lines for the error estimate in Theorem 4.7, while the left-hand sides present the second term of this estimate only. The middle and the bottom images demonstrate the role of the parameter p, whereas comparing the top and the bottom images shows the relationship between p and the polynomial degree N.

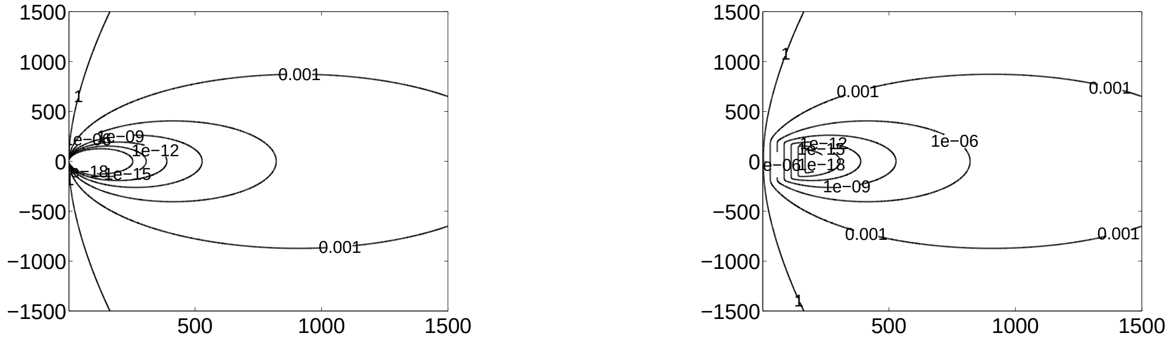
In the next two examples, we have always chosen pseudo-random source and target nodes in $[-\frac{1}{4}, \frac{1}{4})$ and coefficients α_l uniformly distributed in the complex box $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]\mathrm{i}$.

Example 4.10. We consider the Gaussian kernels with

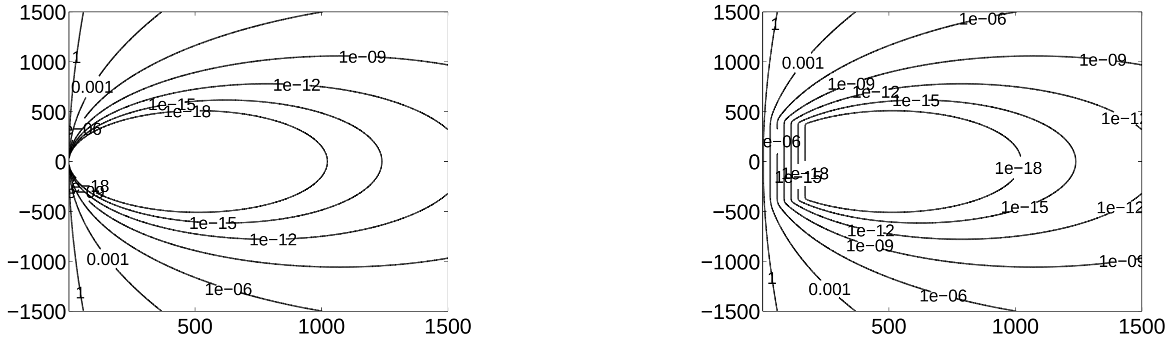
i) $\sigma = 4(138 + 100\mathrm{i})$,

ii) $\sigma = 20 + 40\mathrm{i}$.

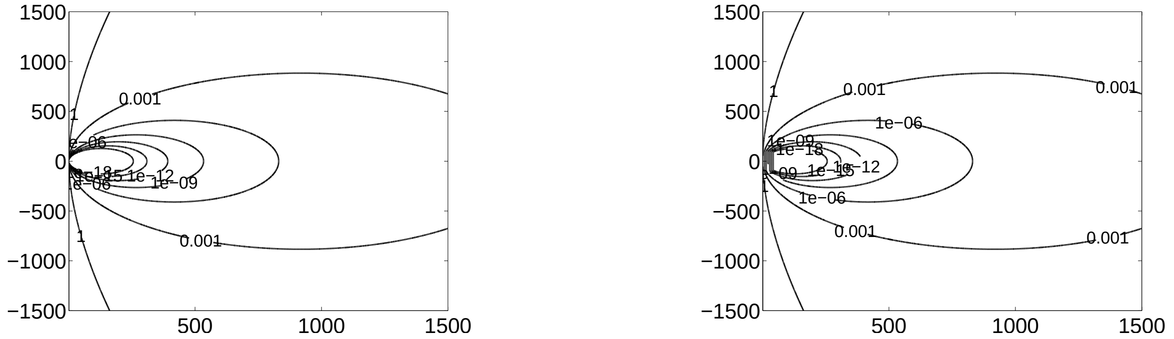
The first parameter σ is taken from [AB05] in order to make the results comparable. The second choice of a considerably smaller σ serves to demonstrate the influence of the parameter p. First we examine the errors that are generated by our fast Gauss transform. Figure 4.2 presents the error introduced by our algorithms as function of the parameter N. These results confirm the error estimates in Theorem 4.7.



(a) $p = 1$, $N = 64$ (b) $p = 1$, $N = 64$



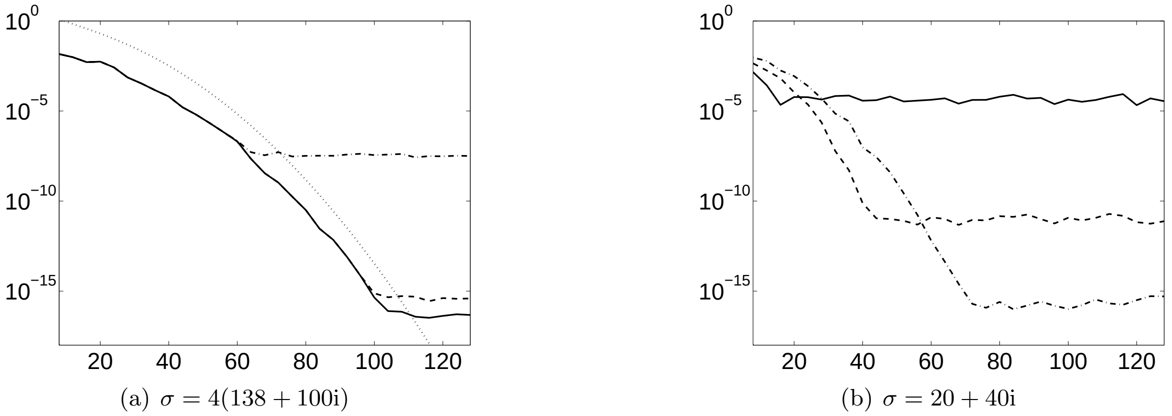
(c) $p = 1$, $N = 128$ (d) $p = 1$, $N = 128$



(e) $p = 2$, $N = 128$ (f) $p = 2$, $N = 128$

Figure 4.1: Level sets in the complex (a, b) plane for the second term in the error estimate (left) and for the total estimate (right) of Theorem 4.7.

Example 4.11. Finally, we compare the computation time of the straightforward summation of (4.1), the multiplication with the precomputed matrix $\boldsymbol{K}_\sigma$, the fast Gauss transform with NDFT, and the fast Gauss transform with NFFT for increasing $M, L \in \mathbb{N}$. The computation time required by the four algorithms is shown in Table 4.1. As expected the fast Gauss transforms outperform the straightforward algorithms, yielding an $\mathcal{O}(M)$ complexity in both variants, whereas the NFFT-version is considerably faster.



(a) $\sigma = 4(138 + 100\mathrm{i})$ (b) $\sigma = 20 + 40\mathrm{i}$

Figure 4.2: Error E_∞ for $N = 8, 12, 16, \dots, 128$ and $L = M = 1000$. Left: FGT with NDFT (solid), NFFT($m = 3$) (dash-dot), NFFT($m = 7$) (dashed), and error estimate (dotted). Right: FGT with NFFT($m = 7$) and $p = 1$ (solid), $p = 1.5$ (dashed), $p = 2$ (dash-dot).

$\log_2 M$	DGT	mult. $\boldsymbol{K}_\sigma$	FGT, NDFT	FGT, NFFT
6	$6.0e-04$	$3.0e-05$	$1.9e-03$	$1.2e-04$
7	$2.4e-03$	$1.4e-04$	$3.9e-03$	$2.3e-04$
8	$9.6e-03$	$1.3e-03$	$7.6e-03$	$4.3e-04$
9	$3.8e-02$	$5.0e-03$	$1.5e-02$	$8.5e-04$
10	$1.5e-01$	$2.0e-02$	$3.0e-02$	$1.8e-03$
11	$6.2e-01$	$8.1e-02$	$6.1e-02$	$3.5e-03$
12	$2.5e+00$	$3.7e-01$	$1.2e-01$	$6.9e-03$
13	$9.9e+00$	$1.4e+00$	$2.4e-01$	$1.4e-02$
14	$4.0e+01$	+	$4.9e-01$	$2.7e-02$
15	$1.6e+02$	+	$9.7e-01$	$5.4e-02$
16	$6.4e+02$	+	$2.0e+00$	$1.1e-01$
17	$2.6e+03$	+	$3.9e+00$	$2.1e-01$
18	$1.0e+04$	+	$7.8e+00$	$4.3e-01$
19	*	+	$1.6e+01$	$8.8e-01$
20	*	+	$3.1e+01$	$1.7e+00$
21	*	+	$6.5e+01$	$3.6e+00$

Table 4.1: Computation time of the discrete Gauss transform (DGT) and Algorithm 4.1 with respect to increasing problem size $M \in \mathbb{N}$, $L = M$, and for a fixed parameter $\sigma = 4(138 + 100\mathrm{i})$. The fast Gauss transform, i.e. Algorithm 4.1, is applied with $N = 128$ and its NFFT-based version also with cut-off parameter $m = 7$, oversampling factor $\sigma = 2$, and the Kaiser-Bessel window function, altogether assuring an error $E_\infty \approx 10^{-15}$. Note that we used accumulated measurements in case of small times and measurements are left out due to the large response time (*) or the limited size of memory (+).

4.2 Spherical Gaussian

This section deals with the spherical analogue to the univariate Gauss transform. Given complex coefficients $\alpha_l \in \mathbb{C}$ and source nodes $\boldsymbol{\eta}_l \in \mathbb{S}^2$, $l = 0, \ldots, L-1$, our goal consists in the fast evaluation of the sum

$$g(\boldsymbol{\xi}) := \sum_{l=0}^{L-1} \alpha_l \mathrm{e}^{-\sigma \|\boldsymbol{\xi}-\boldsymbol{\eta}_l\|_2^2} = \sum_{l=0}^{L-1} \alpha_l \mathrm{e}^{2\sigma(\boldsymbol{\eta}_l \cdot \boldsymbol{\xi} - 1)} \tag{4.8}$$

at the target nodes $\boldsymbol{\xi}_j \in \mathbb{S}^2$, $j = 0, \ldots, M-1$, where $\sigma > 0$ denotes a real parameter. The Gaussian kernel in this task obeys the following properties.

Lemma 4.12. *For $\sigma > 0$, the* spherical Gaussian kernel $K_\sigma : [-1, 1] \to \mathbb{R}$, *given by*

$$K_\sigma(x) := \mathrm{e}^{2\sigma(x-1)}$$

possesses the Fourier-Legendre expansion

$$K_\sigma(x) = \sum_{k \in \mathbb{N}_0} \frac{2k+1}{4\pi} \hat{w}_k P_k(x) \tag{4.9}$$

with coefficients

$$\hat{w}_k = \frac{2\pi\sigma^k}{\mathrm{e}^{2\sigma} k!} \int_{-1}^{1} \mathrm{e}^{2\sigma x} \left(1 - x^2\right)^k \mathrm{d}x.$$

Proof. Using Definition 2.25 and integration by parts k times yields the assertion. □

4.2.1 Derivation of Algorithm 4.2

Similar to the approach for the univariate Gauss transform, we simply propose to truncate the series (4.9) at a fixed polynomial degree $N \in \mathbb{N}_0$, i.e.,

$$K_{\sigma,N}(\boldsymbol{\eta} \cdot \boldsymbol{\xi}) := \sum_{k=0}^{N} \frac{2k+1}{4\pi} \hat{w}_k P_k(\boldsymbol{\eta} \cdot \boldsymbol{\xi}). \tag{4.10}$$

Substituting (4.10) into (4.8), applying the Addition theorem 2.31, and interchanging the order of summation we obtain finally the approximation

$$g_N(\boldsymbol{\xi}) := \sum_{k=0}^{N} \sum_{n=-k}^{k} \hat{w}_k \left(\sum_{l=0}^{L-1} \alpha_l \overline{Y_k^n(\boldsymbol{\eta}_l)} \right) Y_k^n(\boldsymbol{\xi}), \tag{4.11}$$

to be evaluated at the target nodes $\boldsymbol{\xi}_j$, $j = 0, \ldots, M-1$.

Our algorithm now works as follows: The expression in the inner brackets can be evaluated by an adjoint nonequispaced fast spherical Fourier transform (adjoint NFSFT) with $\mathcal{O}(L + N^2 \log^2 N)$ arithmetic operations involving the L source nodes $\boldsymbol{\eta}_l$. This is followed by $(N+1)^2$ multiplications with the precomputed Fourier-Legendre coefficients $\hat{w}_k$, and completed by an NFSFT to evaluate the outer sum at the M target nodes $\boldsymbol{\xi}_j$ with $\mathcal{O}(M + N^2 \log^2 N)$ arithmetic operations, see Algorithm 3.5.

Remark 4.13. *In matrix-vector notation the original problem* (4.8) *reads* $\boldsymbol{g} = \boldsymbol{K}_\sigma \boldsymbol{\alpha}$*, where*

$$\begin{aligned}
\boldsymbol{g} &:= \left(g\left(\boldsymbol{\xi}_j\right)\right)_{j=0,\dots,M-1} \in \mathbb{C}^M,\\
\boldsymbol{K}_\sigma &:= \left(K_\sigma\left(\boldsymbol{\eta}_l \cdot \boldsymbol{\xi}_j\right)\right)_{j=0,\dots,M-1;l=0,\dots,L-1} \in \mathbb{R}^{M\times L},\\
\boldsymbol{\alpha} &:= \left(\alpha_l\right)_{l=0,\dots,L-1} \in \mathbb{C}^L.
\end{aligned}$$

Our approach is a particular approximation to the matrix $\boldsymbol{K}_\sigma$ *of at most rank* $(N+1)^2$ *and takes the form* $\boldsymbol{g}_N = \boldsymbol{Y}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{Y}_{\mathcal{Y}}^{\mathsf{H}} \boldsymbol{\alpha}$ *with*

$$\begin{aligned}
\boldsymbol{g}_N &:= \left(g_N\left(\boldsymbol{\xi}_j\right)\right)_{j=0,\dots,M-1} \in \mathbb{C}^M,\\
\boldsymbol{Y}_{\mathcal{X}} &:= \left(Y_k^n\left(\boldsymbol{\xi}_j\right)\right)_{j=0,\dots,M-1;k=0,\dots,M,\, n=-k,\dots,k} \in \mathbb{C}^{M\times(N+1)^2},\\
\hat{\boldsymbol{W}} &:= \operatorname{diag}\left(\hat{\boldsymbol{w}}\right),\ \hat{\boldsymbol{w}} := \left(\hat{w}_k\right)_{k=0,\dots,N,\, n=-k,\dots,k} \in \mathbb{R}^{(N+1)^2},\\
\boldsymbol{Y}_{\mathcal{Y}} &:= \left(Y_k^n\left(\boldsymbol{\eta}_l\right)\right)_{l=0,\dots,L-1;k=0,\dots,N,\, n=-k,\dots,k} \in \mathbb{C}^{L\times(N+1)^2}.
\end{aligned}$$

The proposed method is summarised in Algorithm 4.2.

Algorithm 4.2 *Fast spherical Gauss transform*

```
Input:  $L,\ M \in \mathbb{N},\ N \in \mathbb{N}_0$,
        $\boldsymbol{\eta}_l \in \mathbb{S}^2,\ \alpha_l \in \mathbb{C},\ l = 0,\dots,L-1$,
        $\boldsymbol{\xi}_j \in \mathbb{S}^2,\ j = 0,\dots,M-1$.

  Compute $\tilde{\boldsymbol{\alpha}} = \boldsymbol{Y}_{\mathcal{Y}}^{\mathsf{H}} \boldsymbol{\alpha}$ by an adjoint NFSFT.

  Evaluate $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{W}} \tilde{\boldsymbol{\alpha}}$.

  Compute $\boldsymbol{g}_N = \boldsymbol{Y}_{\mathcal{X}} \hat{\boldsymbol{\alpha}}$ by an NFSFT, cf. Algorithm 3.5.

Output:      $\boldsymbol{g}_N$ approximating $\boldsymbol{g} = \boldsymbol{K}_\sigma \boldsymbol{\alpha}$.
Complexity:  $\mathcal{O}(M + L + N^2 \log^2 N)$.
```

Remark 4.14. *Replacing the NFSFT algorithms by their slow versions, i.e., by the non-equispaced discrete spherical Fourier transform (NDSFT) and its adjoint which need* $\mathcal{O}(LN^2)$ *and* $\mathcal{O}(MN^2)$ *arithmetic operations for the multiplications with the matrices* $\boldsymbol{Y}_{\mathcal{Y}}^{\mathsf{H}}$ *and* $\boldsymbol{Y}_{\mathcal{X}}$*, respectively, yields an* $\mathcal{O}(M+L)$ *algorithm, too. Nevertheless, the fast algorithms are the key for applications to large data sets since they decouple the degree* N *of the expansion from the number of nodes* L *and* M*.*

Furthermore, note that the precomputation of the Fourier-Legendre coefficients $\hat{w}_k$*, i.e., the evaluation of the entries of the diagonal matrix* $\hat{\boldsymbol{W}}$*, needs some care. Lemma 2.26 and integration by parts yield the difference equation* $2\sigma\hat{w}_{k-1} - 2\sigma\hat{w}_{k+1} = (2k+1)\hat{w}_k$ *for* $k \in \mathbb{N}$ *where* $\hat{w}_0 = 2\pi\sigma^{-1}\mathrm{e}^{-2\sigma}\sinh 2\sigma$ *and* $\hat{w}_1 = \pi\sigma^{-2}\mathrm{e}^{-2\sigma}(2\sigma\cosh 2\sigma + \sinh\sigma)$*. Using this equation in a forward recursion turns out to be numerically unstable. But due to the fact that* $\hat{w}_k = 2\sigma^{-\frac{1}{2}}\mathrm{e}^{-2\sigma}\pi^{\frac{3}{2}} I_{k+\frac{1}{2}}(2\sigma)$*, where* $I_{k+\frac{1}{2}}$ *denotes the modified Bessel function of first kind, we use routines for evaluating Bessel functions provided by the GNU scientific library (GSL) [gsl] in the pre-computation of this Fourier-Legendre coefficients.*

4.2.2 Error estimates

Beyond the finite accuracy of the NFSFT, our algorithm raises the following error.

Theorem 4.15. *Let $N \in \mathbb{N}_0$ and g_N denote the approximation of g as given above, then the uniform error estimate*

$$\frac{\|g - g_N\|_\infty}{\|\boldsymbol{\alpha}\|_1} \leq \frac{\sqrt{\pi}\,(\mathrm{e}^\sigma - 1)\,\sigma^N}{\Gamma\left(N + \frac{1}{2}\right)}$$

holds true. Moreover, there exists a constant $C > 1$, such that for $N > C\sigma$ the spectral norm estimate

$$\left\|\boldsymbol{K}_\sigma - \boldsymbol{Y}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{Y}_{\mathcal{Y}}\right\|_2 \leq \sqrt{LM} \cdot \left(\frac{C\sigma}{N}\right)^N$$

is fulfilled.

Proof. Using Lemma 4.12 gives

$$\frac{2k+1}{4\pi}\,|\hat{w}_k| \leq \frac{\left(k + \frac{1}{2}\right)\sigma^k}{\Gamma(k+1)} \int_{-1}^{1} \left(1 - x^2\right)^k \mathrm{d}x = \frac{\sqrt{\pi}\sigma^k}{\Gamma\left(k + \frac{1}{2}\right)}.$$

Due to $l! \cdot \Gamma(N + \frac{1}{2}) \leq \Gamma(N + \frac{1}{2} + l)$, the assertion is obtained by applying Lemma 2.33 to

$$\sum_{k>N} \frac{\sqrt{\pi}\sigma^k}{\Gamma\left(k + \frac{1}{2}\right)} \leq \frac{\sqrt{\pi}\sigma^N}{\Gamma\left(N + \frac{1}{2}\right)} \sum_{l\in\mathbb{N}} \frac{\sigma^l}{l!} = \frac{\sqrt{\pi\sigma}\,(\mathrm{e}^\sigma - 1)\,\sigma^{N-\frac{1}{2}}}{\Gamma\left(N + \frac{1}{2}\right)}.$$

Since $\Gamma(N + \frac{1}{2}) \geq \Gamma(N)$, Stirling's approximation, see e.g. [AS72, p. 257], and $N > \sigma$ yield

$$\frac{\sqrt{\pi\sigma}\,(\mathrm{e}^\sigma - 1)\,\sigma^{N-\frac{1}{2}}}{\Gamma\left(N + \frac{1}{2}\right)} \leq \frac{\sqrt{\pi}\,(\mathrm{e}^\sigma - 1)\,\sigma^N}{\sqrt{2\pi}\,(N-1)^{N-\frac{1}{2}}\,\mathrm{e}^{-(N-1)}} \leq \left(\frac{C\sigma}{N}\right)^N$$

and thus, by Hölders inequality the second assertion. ☐

4.2.3 Numerical experiments

Subsequently, we test the accuracy and the computation time of Algorithm 4.2 in Example 4.16 and Example 4.17, respectively. In our tests we have always chosen pseudo-random coefficients b_l from $\left[-\frac{1}{2}, \frac{1}{2}\right]$ and pseudo-random source and target nodes $\boldsymbol{\eta}, \boldsymbol{\xi}$ with uniform distribution in their spherical coordinates $(\vartheta, \varphi) \in [0, \pi] \times [-\pi, \pi)$.

Example 4.16. First we examine the errors that are generated by our fast Gauss transform. Figure 4.3 presents the error introduced by our algorithms as function of the parameter N. These results confirm the error estimates in Theorem 4.15. The comparison with respect to the spectral norm surprisingly shows that the best possible low rank approximation by means of the truncated singular value decomposition achieves a much better approximation only for very large rank.

Example 4.17. As for the univariate case, we finally compare the computation time of the straightforward summation of (4.8), the multiplication with the precomputed matrix $\boldsymbol{K}_\sigma$, the fast spherical Gauss transform with NDSFT, and the fast spherical Gauss transform with NFSFT for increasing $M, L \in \mathbb{N}$. The computation time required by the four algorithms is shown in Table 4.2. As expected the fast Gauss transforms outperform the straightforward algorithms, yielding an $\mathcal{O}(M)$ complexity in both variants, whereas the NFSFT-version is considerably faster.

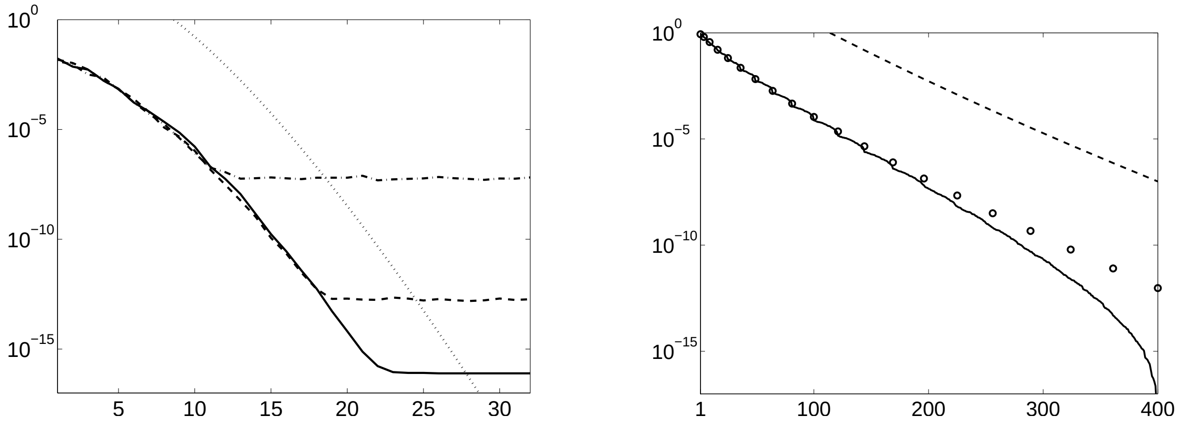


Figure 4.3: Error for the spherical Gaussian kernel, $\sigma = 2.5$. Left: E_∞, cf. equation (4.7), with respect to increasing $N = 1, 2, \ldots, 32$, fixed $L = M = 1000$. FGT with NDFT (solid), NFFT($m = 3$) (dash-dot), NFFT($m = 6$) (dashed), and error estimate (dotted). Right: Spectral norm error with respect to increasing $N = 0, 1, 2, \ldots, 19$, i.e., rank $1, 4, 9, \ldots, 400$; fixed $M = L = 400$. Proposed approximation (circle), truncated singular value decomposition (solid), and spectral norm estimate (dashed).

$\log_2 M$	DGT	mult. $\boldsymbol{K}_\sigma$	FGT, NDSFT	FGT, NFSFT
6	$1.6e-03$	$5.0e-05$	$8.2e-02$	$4.8e-01$
7	$6.2e-03$	$3.4e-04$	$1.6e-01$	$4.8e-01$
8	$2.5e-02$	$1.3e-03$	$3.2e-01$	$4.9e-01$
9	$1.0e-01$	$5.3e-03$	$6.4e-01$	$5.0e-01$
10	$4.0e-01$	$2.1e-02$	$1.3e+00$	$5.1e-01$
11	$1.6e+00$	$8.3e-02$	$2.6e+00$	$5.4e-01$
12	$6.4e+00$	$3.6e-01$	$5.1e+00$	$5.9e-01$
13	$2.5e+01$	+	$1.0e+01$	$6.9e-01$
14	$1.0e+02$	+	$2.0e+01$	$8.9e-01$
15	$4.1e+02$	+	$4.1e+01$	$1.3e+00$
16	$1.6e+03$	+	$8.2e+01$	$2.2e+00$
17	*	+	$1.6e+02$	$3.9e+00$
18	*	+	$3.3e+02$	$7.3e+00$
19	*	+	$6.6e+02$	$1.4e+01$
20	*	+	$1.3e+03$	$2.8e+01$

Table 4.2: Computation time of the discrete Gauss transform (DGT) and Algorithm 4.2 with respect to increasing problem size $M \in \mathbb{N}$, $L = M$ and for a fixed parameter $\sigma = 100$. The fast Gauss transform, i.e. Algorithm 4.2, is applied with $N = 128$ and its NFSFT-based version also with FPT-threshold 1000, NFFT-oversampling factor $\sigma = 2$, NFFT-cut-off parameter $m = 7$, and the Kaiser-Bessel window function, altogether assuring an error $E_\infty \approx 10^{-14}$. Note that we used accumulated measurements in case of small times and measurements are left out due to the large response time (*) or the limited size of memory (+).

4.3 Notes and comments

Fast and approximate schemes for the computation of the discrete Gauss transform for real σ in (4.1) have initially been proposed in [Str91, GS91, GS98]. Recently, a univariate version for complex parameters with arithmetic complexity $\mathcal{O}(L \log L + M)$ was introduced in [AB05]. In contrast, Algorithm 4.1 and Algorithm 4.2 take $\mathcal{O}(L + M)$ operations for a fixed target accuracy and a fixed Gaussian parameter σ.

For the univariate case, the only common Example 4.11 shows that our algorithm is faster than the scheme in [AB05]. Note however, that if we spread the nodes quasi-uniformly and scale a so that the Gaussian has fixed effective width when measured in units of the average spacing M^{-1} between nodes, the total computational cost is of order $\mathcal{O}(M \log M)$.

Furthermore, Algorithm 4.1 has a simple structure and belongs to a general class of fast summation schemes developed in [PS03, FS04, PSN04]. This gives also rise to the following other discretisation techniques for the weights $\hat{w}_k$, $k \in I_N$:

1. Instead of $\tilde{K}_\sigma$ we can use the truncated function

   $$\bar{K}_\sigma(x) := \sum_{r\in\mathbb{Z}} \chi_{[-\frac{p}{2},\frac{p}{2})}(x-pr)K(x-pr)$$

   together with an appropriate boundary regularisation as described in [PS03, FS04]. By the boundary regularisation, $\bar{K}_\sigma$ becomes a smooth p-periodic function with uniformly convergent Fourier series. For its corresponding truncated version $\bar{K}_{\sigma,N}$ we use the approximation of the Fourier coefficients by the discrete Fourier coefficients

   $$\bar{w}_k := \frac{p}{N} \sum_{j\in I_N} \bar{K}_\sigma\left(\frac{jp}{N}\right) \mathrm{e}^{2\pi\mathrm{i}\frac{jk}{pN}}.$$

   Thus, the computation scheme needs some additional precomputation effort due to the boundary regularisation and trades the availability of Fourier coefficients for the need of derivatives near the boundary $\pm\frac{p}{2}$.

2. From another point of view, the function $K_{\sigma,N}$ was obtained by applying the trapezoidal quadrature rule within the interval $[-\frac{N}{2p}, \frac{N}{2p}]$ as follows:

   $$\mathrm{e}^{-\sigma x^2} = \frac{\sqrt{\pi}}{\sqrt{\sigma}} \int_{-\infty}^{\infty} \mathrm{e}^{-y^2\pi^2/\sigma} \mathrm{e}^{2\pi\mathrm{i}xy} \mathrm{d}y \approx \frac{\sqrt{\pi}}{\sqrt{\sigma}} \frac{1}{p} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \mathrm{e}^{-\left(\frac{l}{p}\right)^2 \pi^2/\sigma} \mathrm{e}^{2\pi\mathrm{i}x\frac{l}{p}}.$$

   Beyond the scope of this thesis, more advanced quadrature rules as [BM02, BM05] might be used to further reduce the number of summands in the right hand side.

Of course, one might wish to have the fast summation scheme for a larger interval $[-\frac{p}{4}, \frac{p}{4}]$ for source and target nodes, as well. For some multiple of the period p as given in Corollary 4.8, it suffices to use the same multiple of the cut-off degree N to guarantee the same accuracy. If the nodes are "widespread" with respect to the real part a of the parameter, our approach is somewhat limited. Nevertheless, one might use it as the low rank building block of hierarchical methods [Hac99, HKS00, FS02a].

Section 4.2 presents a multivariate Gauss transform for the spherical setting. More general, we consider in [KKP06] fast summation schemes also for the Poisson, the Singularity, and locally supported kernels, see also [FGS98a]. Loosely speaking, we only use the decay of the Fourier-Legendre coefficients, cf. [Sch97, BH01, zCF05], to obtain a accurate degenerate approximation of the kernel.

5

Inverse NFFT

The inverse NFFT constructs a trigonometric polynomial $f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}$, cf. Section 2.1, such that for given data points $(\boldsymbol{x}_j,\, y_j) \in \mathbb{T}^d \times \mathbb{C},\ j = 0, \ldots, M-1$, the approximate identity

$$f(\boldsymbol{x}_j) \approx y_j \tag{5.1}$$

is fulfilled. Thus, we aim to solve the linear system of equations $\boldsymbol{A}\hat{\boldsymbol{f}} \approx \boldsymbol{y}$ for the vector of Fourier coefficients $\hat{\boldsymbol{f}} \in \mathbb{C}^{|I_{\boldsymbol{N}}|}$, $\hat{f}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, which is *inverse* to the matrix vector multiplication in (3.1). In contrast to the ordinary Fourier matrix, its nonequispaced analogue $\boldsymbol{A}$ is in general neither unitary nor square.

This chapter is devoted to the meaningful variants of (5.1) and efficient iterative schemes for its actual solution. The main tool in our algorithms is the NFFT, i.e., the fast matrix times vector multiplication with $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{H}}$, respectively. We focus on the least squares approximation in Section 5.1, where rigorous bounds for the condition number of the involved matrix have been proven in [Grö92]. Theorem 5.2 summarises these results and concludes certain convergence rates of Algorithm 5.1 in Corollary 5.3. Convergence of Algorithm 5.2 for the optimal interpolation of scattered data is proven in Section 5.2. The main result is Theorem 5.16, whereas Corollary 5.20 shows a couple of applications to more specific settings and Theorem 5.22 presents a generalisation for the interpolation on the sphere.

Subsequently, we propose a recently developed algorithm for the recovery of trigonometric polynomials with few non-zero Fourier coefficients and apply the inverse NFFT in magnetic resonance imaging. We conclude with our generic solver component within the NFFT software package [KP06b] and comment on related work. Parts of the material in this chapter are submitted for publication, see [KP04b, KKP05, KR06b].

5.1 Least squares approximation

A standard method to determine the vector of Fourier coefficients $\hat{\boldsymbol{f}}$ in (5.1) is to solve the general linear least squares problem, see, e.g., [Bjö96, p. 15],

$$\|\hat{\boldsymbol{f}}\|_2 \to \min \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}\|_2 = \min\,. \tag{5.2}$$

If a vector of Fourier coefficients $\hat{\boldsymbol{f}}$ exists such that $\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}$, then the right hand side $\boldsymbol{y}$ is called consistent. The actual solution of the general linear least squares problem can be computed by means of the singular value decomposition which is very expensive here and no practical way at all.

For $|I_{\boldsymbol{N}}| < M$, the linear system (5.1) is over-determined. Hence, in general the given data $\boldsymbol{y}$ will be only approximated up to a residual $\boldsymbol{r} := \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}$. In order to compensate for clusters in the sampling set $\mathcal{X}$, it is also useful to incorporate weights $w_j > 0$ into our problem, i.e., to consider the weighted least squares approximation

$$\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}\|_{\boldsymbol{W}}^2 = \sum_{j=0}^{M-1} w_j |y_j - f(\boldsymbol{x}_j)|^2 \xrightarrow{\hat{\boldsymbol{f}}} \min, \tag{5.3}$$

where $\boldsymbol{W} := \operatorname{diag}(w_j)_{j=0,\dots,M-1}$.

5.1.1 Derivation of Algorithm 5.1

Basic linear algebra reveals that the least squares problem (5.3) is equivalent to a system of linear equations which can be solved efficiently by means of an iterative algorithm.

Lemma 5.1. *The least squares problem* (5.3) *is equivalent to weighted normal equation of first kind*

$$\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{y}. \tag{5.4}$$

Moreover, the involved matrix $\boldsymbol{T} := \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}$ *possesses a multilevel Toeplitz structure, i.e., has constant entries* $T_{\boldsymbol{k},\boldsymbol{l}} = \sum_{j=0}^{M-1} w_j \mathrm{e}^{2\pi\mathrm{i}(\boldsymbol{k}-\boldsymbol{l})\boldsymbol{x}_j}$, $\boldsymbol{k}, \boldsymbol{l} \in I_{\boldsymbol{N}}$, *along "diagonals".*

Proof. The first assertion is due to [Bjö96, Thm. 1.1.2] for the matrix $\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{A}$. *Straightforward computation of the entries* $T_{\boldsymbol{k},\boldsymbol{l}}$ *yields the matrix structure.* ☐

The iterative solution of (5.3) has been addressed in [Grö92, Grö93, FGS95, BG04a]. The adaptive weights conjugate gradient Toeplitz method (ACT) applies the conjugate gradient method to the weighted normal equation (5.4) and uses the multilevel Toeplitz structure for fast matrix vector multiplications.

In contrast, we solve problem (5.3) by a factorised variant of conjugated gradients (CGNR, N for "Normal equation" and R for "Residual minimisation"). Moreover, we exploit the factorisation in (5.4) to iterate the original residual $\boldsymbol{r}_l$ instead of the residual of the normal equation. In particular, we use the NFFT for fast matrix vector multiplications with $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{H}}$, respectively. Both variants generate the same sequence of approximations in exact arithmetic, but the CGNR approach is considered to be more stable with respect to round-off errors, cf. [PS82, Sec. 7.1]. In summary, we suggest the following Algorithm 5.1.

5.1.2 Convergence results

Concerning the regularity only, the nonequispaced Fourier matrix $\boldsymbol{A}$ has full rank almost surely, whenever $|I_{\boldsymbol{N}}| \le M$ and the nodes $\boldsymbol{x}_j$ are drawn independently from the uniform distribution on $\mathbb{T}^d$, see [BG04a, Thm. 3.2]. The following theory on the condition number of the Toeplitz matrix $\boldsymbol{T}$, cf. Lemma 5.1, and hence for the convergence of Algorithm 5.1, is due to [Grö92, Grö93, BG04a]. A similar result has been obtained independently in [MNW01] in the context of Marcinkiewicz-Zygmund inequalities. We analyse the convergence of Algorithm 5.1 in the univariate case $d = 1$ and comment on the multivariate case only.

Algorithm 5.1 Inverse NFFT, CGNR

```
Input: $d, M \in \mathbb{N}$, $\boldsymbol{N} \in \mathbb{N}^d$, $\hat{\boldsymbol{f}}_0 \in \mathbb{C}^{|I_{\boldsymbol{N}}|}$, $\boldsymbol{W} = \operatorname{diag}(w_j)_{j=0,\ldots,M-1}$,
       $(\boldsymbol{x}_j, y_j) \in \mathbb{T}^d \times \mathbb{C},\ j = 0, \ldots, M-1$.

$\boldsymbol{r}_0 = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_0$
$\hat{\boldsymbol{p}}_0 = \hat{\boldsymbol{z}}_0 = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{r}_0$
for $l = 0, \ldots$ do
  $\boldsymbol{v}_l = \boldsymbol{A}\hat{\boldsymbol{p}}_l$
  $\alpha_l = \hat{\boldsymbol{z}}_l^{\mathsf{H}}\hat{\boldsymbol{z}}_l \,/\, \boldsymbol{v}_l^{\mathsf{H}}\boldsymbol{W}\boldsymbol{v}_l$
  $\hat{\boldsymbol{f}}_{l+1} = \hat{\boldsymbol{f}}_l + \alpha_l\hat{\boldsymbol{p}}_l$
  $\boldsymbol{r}_{l+1} = \boldsymbol{r}_l - \alpha_l\boldsymbol{v}_l$
  $\hat{\boldsymbol{z}}_{l+1} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{r}_{l+1}$
  $\beta_l = \hat{\boldsymbol{z}}_{l+1}^{\mathsf{H}}\hat{\boldsymbol{z}}_{l+1} \,/\, \hat{\boldsymbol{z}}_l^{\mathsf{H}}\hat{\boldsymbol{z}}_l$
  $\hat{\boldsymbol{p}}_{l+1} = \hat{\boldsymbol{z}}_{l+1} + \beta_l\hat{\boldsymbol{p}}_l$
end for
Output:     vector of coefficients $\hat{\boldsymbol{f}}_l$.
Complexity: $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$ per iteration.
```

Theorem 5.2. *[Grö92] Let $d = 1$ and a δ-dense sampling set $\mathcal{X} \subset \mathbb{T}$ of cardinality $M \in \mathbb{N}$ with $x_0 < x_1 < \ldots < x_{M-1}$ be given. Moreover, let the entries of the diagonal matrix $\boldsymbol{W}$ be given by the Voronoi weights, i.e., $w_j = \frac{1}{2}(x_{j+1} - x_{j-1})$, $j = 0, \ldots, M-1$, where for notational convenience $x_{-1} = x_{M-1} - 1$ and $x_M = x_0 + 1$.*

Then for $N \in \mathbb{N}$, $N < \delta^{-1}$, the norm of an arbitrary vector of Fourier coefficients $\hat{\boldsymbol{f}} \in \mathbb{C}^N$ and the norm of the corresponding sample values $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$ are equivalent in the sense that

$$\left| \left\| \hat{\boldsymbol{f}} \right\|_2 - \|\boldsymbol{f}\|_{\boldsymbol{W}} \right| \leq \delta N \left\| \hat{\boldsymbol{f}} \right\|_2 .$$

Proof. We abbreviate $z_j = \frac{1}{2}(x_{j-1} + x_j)$ for $j = 0, \ldots, M$. Now, let the functions $\chi_j : \mathbb{R} \to \mathbb{R}$, $\chi_j(x) = 1$ for $x \in [z_j, z_{j+1})$ and $\chi_j(x) = 0$ otherwise, form a partition of unity and hence the previously defined weights be given by $w_j = \int_{\mathbb{T}} \chi_j(x) \mathrm{d}x$. Using the Parseval identity (2.1) and the triangle inequality yields

$$\left| \left\| \hat{\boldsymbol{f}} \right\|_2 - \|\boldsymbol{f}\|_{\boldsymbol{W}} \right| \leq \left\| f - \sum_{j=0}^{M-1} f(x_j)\, \chi_j \right\|_{L^2} .$$

We calculate further

$$\begin{aligned} \int_{\mathbb{T}} |f(x) - f(x_j)|^2 \chi_j(x)\, \mathrm{d}x &= \int_{z_j}^{z_{j+1}} |f(x) - f(x_j)|^2 \,\mathrm{d}x \\ &= \int_{z_j}^{x_j} |f(x) - f(x_j)|^2 \,\mathrm{d}x + \int_{x_j}^{z_{j+1}} |f(x) - f(x_j)|^2 \,\mathrm{d}x. \end{aligned}$$

Next, we invoke the Wirtinger inequality, cf. [HLP89, pp. 184] that states for continously differentiable g with either $g(a) = 0$ or $g(b) = 0$, the inequality

$$\int_a^b |g(x)|^2 \,\mathrm{d}x \leq \frac{4(b-a)^2}{\pi^2} \int_a^b |g'(x)|^2 \,\mathrm{d}x.$$

We proceed in the estimate by

$$\begin{aligned}\int_{z_j}^{x_j} |f(x)-f(x_j)|^2\,\mathrm{d}x + \int_{x_j}^{z_{j+1}} |f(x)-f(x_j)|^2\,\mathrm{d}x\\ \le \frac{4|z_j-x_j|^2}{\pi^2}\int_{z_j}^{x_j} |f'(x)|^2\,\mathrm{d}x + \frac{4|z_{j+1}-x_j|^2}{\pi^2}\int_{x_j}^{z_{j+1}} |f'(x)|^2\,\mathrm{d}x\\ \le \frac{\delta^2}{\pi^2}\int_{z_j}^{z_{j+1}} |f'(x)|^2\,\mathrm{d}x.\end{aligned}$$

Summing over the partition and applying the Bernstein inequality from Lemma 2.4, i.e.,

$$\sum_{j=0}^{M-1}\frac{\delta^2}{\pi^2}\int_{z_j}^{z_{j+1}} |f'(x)|^2\,\mathrm{d}x = \frac{\delta^2}{\pi^2}\|f'\|_{L^2}^2 \le \left(\delta N \|f\|_{L^2}\right)^2,$$

concludes our proof. □

We immediately obtain the following convergence result for Algorithm 5.1.

Corollary 5.3. Under the assumptions of Theorem 5.2, the Toeplitz matrix $\boldsymbol{T} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}$ is positive definite. Moreover, the l-th residual $\boldsymbol{r}_l$ of Algorithm 5.1 fulfils for every consistent right hand side $\boldsymbol{y} \in \mathbb{C}^M$ in (5.1) and for the initial guess $\hat{\boldsymbol{f}}_0 = \boldsymbol{0}$ the a-priori estimate

$$\|\boldsymbol{r}_l\|_{\boldsymbol{W}} \le 2\,(\delta N)^l\,\|\boldsymbol{y}\|_{\boldsymbol{W}}. \tag{5.5}$$

In particular, Algorithm 5.1 takes only a constant number of iterations and hence $\mathcal{O}(N\log N + M)$ arithmetic operations to achieve a fixed relative target residual.

Proof. Induction over $l \in \mathbb{N}_0$ *indeed shows that the iterated residual in Algorithm 5.1 fulfils* $\boldsymbol{r}_l = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_l$. *Theorem 5.2 states, due to* $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$,

$$(1-\delta N)^2 \le \frac{\hat{\boldsymbol{f}}^{\mathsf{H}}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}\hat{\boldsymbol{f}}}{\hat{\boldsymbol{f}}^{\mathsf{H}}\hat{\boldsymbol{f}}} \le (1+\delta N)^2$$

and hence, the matrix $\boldsymbol{T}$ *possesses a bounded condition number* $\mathrm{cond}(\boldsymbol{T}) \le (\frac{1+\delta N}{1-\delta N})^2$. *Applying the standard estimate for the convergence of the conjugate gradient method, see e.g. [Axe96, pp. 566], to the normal equation* (5.4) *yields*

$$\left\|\boldsymbol{T}^{-1}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{y} - \hat{\boldsymbol{f}}_l\right\|_{\boldsymbol{T}} \le 2\left(\frac{\sqrt{\mathrm{cond}(\boldsymbol{T})}-1}{\sqrt{\mathrm{cond}(\boldsymbol{T})}+1}\right)^l \left\|\boldsymbol{T}^{-1}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{y} - \hat{\boldsymbol{f}}_0\right\|_{\boldsymbol{T}},$$

where the energy norm is given by $\|\hat{\boldsymbol{f}}\|_{\boldsymbol{T}}^2 := \hat{\boldsymbol{f}}^{\mathsf{H}}\boldsymbol{T}\hat{\boldsymbol{f}}$. *The consistency of* $\boldsymbol{y}$ *yields* $\|\boldsymbol{T}^{-1}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{y} - \hat{\boldsymbol{f}}_l\|_{\boldsymbol{T}} = \|\boldsymbol{r}_l\|_{\boldsymbol{W}}$ *and thus the second assertion.*

Hence, by a constant number of iterations the residual is decreased to a certain fraction. Using Algorithm 3.1 and Algorithm 3.2, the total number of floating point operations per iteration is bounded by $\mathcal{O}(N\log N + M)$. □

We remark that the estimate (5.5) can be generalised for an inconsistent right hand side $\boldsymbol{y}$ and for an arbitrary initial guess $\hat{\boldsymbol{f}}_0$. In general, Algorithm 5.1 obeys a monotonic decrease of the residual norm $\|\boldsymbol{r}_l\|_{\boldsymbol{W}}$ and the error $\|\boldsymbol{T}^{-1}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{y} - \hat{\boldsymbol{f}}_l\|_2$, whereas the residual $\hat{\boldsymbol{z}}_l = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{r}_l$ of the normal equation (5.4) might exhibit oscillations, see [Bjö96, pp. 288] for details.

5.1.3 Numerical experiments

We give numerical verification of the presented estimates on the conditioning of the least squares problem and on the merit of including density compensation weights.

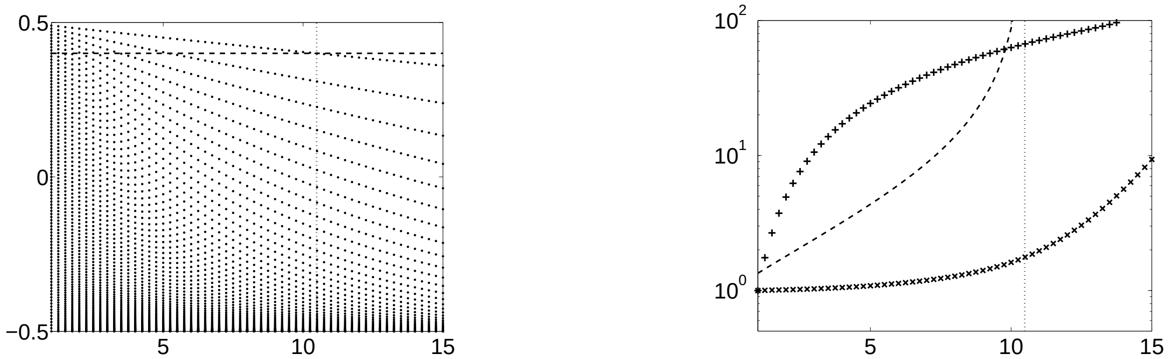


Figure 5.1: The increasingly clustered sampling sets $\mathcal{X}^{(\alpha)}$ are shown with respect to the parameter $\alpha = 1, \ldots, 15$ (left). We plot the condition number of the matrix $\boldsymbol{T} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}$ ($\times$), of the non-weighted matrix $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ ($+$), and the estimate from Corollary 5.3 (dashed) with respect to the parameter α (right). We also show the constant line $\frac{1}{2} - N^{-1}$ (dashed, left) and the vertical line at the critical parameter α such that $\delta^{(\alpha)} = N^{-1}$ (dotted).

Example 5.4. We compute the condition number of the matrix $\boldsymbol{T} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}$, of the non-weighted matrix $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$, and the estimate from Corollary 5.3. In this test, the polynomial degree $N = 10$ and the number of nodes $M = 100$ are fixed while we use increasingly clustered sampling nodes

$$x_j^{(\alpha)} := \frac{j^\alpha}{M^\alpha} - \frac{1}{2}, \qquad j = 0, \ldots, M-1.$$

The parameter $\alpha \geq 1$ controls how fast the density of the nodes in the sampling set $\mathcal{X}^{(\alpha)} = \{x_j^{(\alpha)}\}$ increases towards $-\frac{1}{2}$.

Figure 5.1 reveals that the weights indeed lead to an almost constant condition number until the sampling set approaches the critical mesh norm $\delta = N^{-1}$. In contrast, the condition number of the non-weighted matrix quickly increases for clustered sampling nodes.

5.1.4 Concluding remarks

Theorem 5.2 has been generalised for the multivariate case $d > 1$, whereas regularity of the multilevel Toeplitz matrix $\boldsymbol{T}$ is assured if $N < \frac{\log 2}{\pi d}\delta^{-1}$, cf. [Grö92, Grö93, BG04a]. Note however, that it is an open problem to obtain this result with a right hand side independent of the dimension d.

In various applications, one wants to use knowledge on the decay of the Fourier coefficients to favour a particular solution or obtain a scheme with a prescribed behaviour during the iterations. More formally, we use non-negative weights $\hat{w}_{\boldsymbol{k}} \geq 0$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, denoted as *damping factors* subsequently, and consider the seminorm-penalised least squares problem

$$\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}\|_{\boldsymbol{W}}^2 + \mu^2 \|\hat{\boldsymbol{f}}\|_{\hat{\boldsymbol{W}}^{-1}}^2 \xrightarrow{\hat{\boldsymbol{f}}} \min, \qquad \mu > 0, \tag{5.6}$$

where

$$(\hat{\boldsymbol{W}}^{-1})_{\boldsymbol{k},\boldsymbol{l}} = \begin{cases} \hat{w}_{\boldsymbol{k}}^{-1} & \text{if} \quad \hat{w}_{\boldsymbol{k}} \neq 0 \text{ and } \boldsymbol{k} = \boldsymbol{l}, \\ 0 & \text{otherwise}, \end{cases}$$

denotes the pseudoinverse of $\hat{\boldsymbol{W}} := \operatorname{diag}(\hat{w}_{\boldsymbol{k}})_{\boldsymbol{k} \in I_{\boldsymbol{N}}}$. We illustrate this idea by the following example.

Example 5.5. Let $d = 2$, $\boldsymbol{N} \in \mathbb{N}^2$, and the "smoothness functional" $\Delta : T_{\boldsymbol{N}} \to \mathbb{R}$,

$$\Delta f := \int_{\mathbb{T}} \int_{\mathbb{T}} \left(\frac{\partial^2 f\left(x_0, x_1\right)}{\partial x_0^2} \right)^2 + 2 \left(\frac{\partial^2 f\left(x_0, x_1\right)}{\partial x_0 x_1} \right)^2 + \left(\frac{\partial^2 f\left(x_0, x_1\right)}{\partial x_1^2} \right)^2 \mathrm{d}x_0 \, \mathrm{d}x_1$$

be given. Obviously, this functional can be expressed solely on the Fourier coefficients of f by

$$\Delta f = 16\pi^4 \sum_{k_0 \in I_{N_0}} \sum_{k_1 \in I_{N_1}} \left(|k_0| + |k_1|\right)^4 \left| \hat{f}_{k_0,k_1} \right|^2 = \|\hat{\boldsymbol{f}}\|^2_{\hat{\boldsymbol{W}}^{-1}},$$

where

$$\hat{w}_{k_0,k_1} = \begin{cases} \frac{1}{16\pi^4 (|k_0| + |k_1|)^4} & \text{if } k_0 \neq 0 \text{ or } k_1 \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

One easily obtains sufficient and necessary conditions for the uniqueness of a solution of (5.6).

Lemma 5.6. *Problem* (5.6) *has a unique solution for every right hand side* $\boldsymbol{y} \in \mathbb{C}^M$ *if and only if the null-spaces of* $\boldsymbol{A}$ *and* $\hat{\boldsymbol{W}}$ *intersect only trivially.*

Proof. The penalised least squares problem (5.6) *can be rewritten by*

$$\left\| \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}} \right\|^2_{\boldsymbol{W}} + \mu^2 \left\| \hat{\boldsymbol{f}} \right\|^2_{\hat{\boldsymbol{W}}^{-1}} = \left\| \begin{pmatrix} \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{A} \hat{\boldsymbol{f}} \\ \mu \hat{\boldsymbol{W}}^{-\frac{1}{2}} \hat{\boldsymbol{f}} \end{pmatrix} \right\|^2_2$$

and is thus equivalent to the normal equation

$$\left(\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A} + \mu^2 \hat{\boldsymbol{W}}^{-1} \right) \hat{\boldsymbol{f}} = \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{y},$$

which is non-singular if and only if the assumption on the null-spaces holds. □

In particular, the condition $\arg\max_{k \in I_{\boldsymbol{N}}} \{\hat{w}_k = 0\} < \delta^{-1}$ suffices in the univariate case for a unique solution of the seminorm-penalised problem (5.6). In contrast, Theorem 5.2 assures a unique solution of the original least squares problem (5.3) only under the stronger assumption $N < \delta^{-1}$.

Note furthermore, that a frequency damping was first applied in a similar context in [RS98]. The authors suggest to solve a weighted least squares problem

$$\left\| \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \left(\boldsymbol{y} - \boldsymbol{A} \hat{\boldsymbol{f}} \right) \right\|_{\hat{\boldsymbol{W}}^{-1}} \xrightarrow{\hat{\boldsymbol{f}}} \min$$

to incorporate knowledge about the decay of the solution $\hat{\boldsymbol{f}}$, whereas damping is applied to the residual $\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{A} \hat{\boldsymbol{f}}_l)$. Note, that the weights $\hat{\boldsymbol{W}}^{-1}$ do only change the solution if the matrix $\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A}$ is rank deficient or an iterative method is stopped early.

Subsequently, we focus on the under-determined consistent reconstruction problem and actually prove how the damping improves the stability.

5.2 Optimal interpolation

For $|I_{\boldsymbol{N}}| > M$, we focus on the under-determined consistent linear system $\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}$, i.e., we expect to interpolate the given data $y_j \in \mathbb{C}$, $j = 0, \ldots, M-1$, exactly. More formally, we incorporate positive weights $\hat{w}_{\boldsymbol{k}} > 0$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, denoted as damping factors subsequently, and consider the *optimal interpolation problem*

$$\|\hat{\boldsymbol{f}}\|^2_{\hat{\boldsymbol{W}}^{-1}} = \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \frac{|\hat{f}_{\boldsymbol{k}}|^2}{\hat{w}_{\boldsymbol{k}}} \xrightarrow{\hat{\boldsymbol{f}}} \min \quad \text{subject to} \quad \boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}, \tag{5.7}$$

where $\hat{\boldsymbol{W}} := \operatorname{diag}(\hat{w}_{\boldsymbol{k}})_{\boldsymbol{k}\in I_{\boldsymbol{N}}}$. Besides the introduced weights, this problem resembles (5.2) for consistent right hand sides.

5.2.1 Derivation of Algorithm 5.2

The following lemma reformulates the interpolation problem. In particular, it turns out that a linear system with the kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$, cf. Definition 2.10, has to be solved.

Lemma 5.7. *The optimal interpolation problem* (5.7) *is equivalent to the* damped normal equations of second kind

$$\boldsymbol{K}_{\boldsymbol{N}}\tilde{\boldsymbol{f}} = \boldsymbol{y}, \qquad \hat{\boldsymbol{f}} = \hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}\tilde{\boldsymbol{f}}. \tag{5.8}$$

Proof. A solution $\hat{\boldsymbol{f}}$ of $\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}$ has minimal weighted norm if and only if it is perpendicular with respect to the weights to the null-space of $\boldsymbol{A}$, i.e., $\hat{\boldsymbol{W}}^{-1/2}\hat{\boldsymbol{f}} \perp \mathcal{N}(\boldsymbol{A}\hat{\boldsymbol{W}}^{1/2})$. We conclude (5.8) *by the fact that the orthogonal complement of the null-space of a matrix is just the range of its adjoint.* ☐

The factorisation $\boldsymbol{K}_{\boldsymbol{N}} = \boldsymbol{A}\hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}$, cf. Lemma 2.11, allows to solve the interpolation problem by the following variant of the conjugate gradient method (CGNE, N for "Normal equation" and E for "Error minimisation"). Similar to Algorithm 5.1, we iterate the original vector $\hat{\boldsymbol{f}}$ instead of the vector $\tilde{\boldsymbol{f}}$ in (5.8).

Algorithm 5.2 Inverse NFFT, CGNE

```
Input:  $d, M \in \mathbb{N}$, $\boldsymbol{N} \in \mathbb{N}^d$, $\hat{\boldsymbol{f}}_0 \in \mathbb{C}^{|I_{\boldsymbol{N}}|}$, $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{w}_{\boldsymbol{k}})_{\boldsymbol{k}\in I_{\boldsymbol{N}}}$,
        $(\boldsymbol{x}_j, y_j) \in \mathbb{T}^d \times \mathbb{C}$, $j = 0, \ldots, M-1$.

$\boldsymbol{r}_0 = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_0$
$\hat{\boldsymbol{p}}_0 = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{r}_0$
for $l = 0, \ldots$ do
    $\alpha_l = \boldsymbol{r}_l^{\mathsf{H}}\boldsymbol{r}_l \,/\, \hat{\boldsymbol{p}}_l^{\mathsf{H}}\hat{\boldsymbol{W}}\hat{\boldsymbol{p}}_l$
    $\hat{\boldsymbol{f}}_{l+1} = \hat{\boldsymbol{f}}_l + \alpha_l\hat{\boldsymbol{W}}\hat{\boldsymbol{p}}_l$
    $\boldsymbol{r}_{l+1} = \boldsymbol{r}_l - \alpha_l\boldsymbol{A}\hat{\boldsymbol{W}}\hat{\boldsymbol{p}}_l$
    $\beta_l = \boldsymbol{r}_{l+1}^{\mathsf{H}}\boldsymbol{r}_{l+1} \,/\, \boldsymbol{r}_l^{\mathsf{H}}\boldsymbol{r}_l$
    $\hat{\boldsymbol{p}}_{l+1} = \beta_l\hat{\boldsymbol{p}}_l + \boldsymbol{A}^{\mathsf{H}}\boldsymbol{r}_{l+1}$
end for
Output:      vector of coefficients $\hat{\boldsymbol{f}}_l$.
Complexity:  $\mathcal{O}(|I_{\boldsymbol{N}}|\log|I_{\boldsymbol{N}}| + M)$ per iteration.
```

5.2.2 Convergence results

We first review some well known results for the interpolation of scattered data. As can be seen from Definition 2.10, the kernel $K_{\boldsymbol{N}}$ is in general positive definite, i.e., the kernel matrix is positive semidefinite $\boldsymbol{a}^{\mathsf{H}}\boldsymbol{K}_{\boldsymbol{N}}\boldsymbol{a} \geq 0$ for arbitrary $\boldsymbol{a} \in \mathbb{C}^M$. If in the univariate case $d = 1$ the number of sampling nodes is bounded by the degree, i.e., $M \leq N$, then $\boldsymbol{K}_N$ is positive definite. Due to the Theorem of Mairhuber-Curtis, cf. [Wen05, pp. 18], no such simple criteria is available for the multivariate case $d > 1$.

The following analysis derives sufficient conditions such that the kernel matrix is positive definite and moreover presents explicit convergence rates for Algorithm 5.2. The standard estimate for the convergence of the conjugate gradient method relies on the extremal eigenvalues of the kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$ only.

Definition 5.8. *We denote by*

$$\Lambda := \Lambda\left(\boldsymbol{K}_{\boldsymbol{N}}\right), \quad \lambda := \lambda\left(\boldsymbol{K}_{\boldsymbol{N}}\right), \quad \operatorname{cond}(\boldsymbol{K}_{\boldsymbol{N}}) := \frac{\Lambda}{\lambda}$$

the largest eigenvalue, the smallest eigenvalue, and the condition number of the kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$, respectively.

Lemma 5.9. *Let the kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$ in Definition 2.10 be positive definite. The l-th error $\hat{\boldsymbol{e}}_l := \hat{\boldsymbol{f}}_l - \hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{K}_{\boldsymbol{N}}^{-1}\boldsymbol{y}$ of Algorithm 5.2 fulfils for every consistent right hand side $\boldsymbol{y} \in \mathbb{C}^M$ and for the initial guess $\hat{\boldsymbol{f}}_0 = \boldsymbol{0}$ the a-priori error bound*

$$\|\hat{\boldsymbol{e}}_l\|_{\hat{\boldsymbol{W}}^{-1}} \leq \frac{2}{\sqrt{\lambda}} \left(\frac{\sqrt{\Lambda} - \sqrt{\lambda}}{\sqrt{\Lambda} + \sqrt{\lambda}}\right)^l \|\boldsymbol{y}\|_2 \,. \tag{5.9}$$

Proof. We note that $\|\hat{\boldsymbol{e}}_l\|_{\hat{\boldsymbol{W}}^{-1}} = \|\tilde{\boldsymbol{f}}_l - \boldsymbol{K}_{\boldsymbol{N}}^{-1}\boldsymbol{y}\|_{\boldsymbol{K}_N}$, where $\tilde{\boldsymbol{f}}_l$ denotes the l-th iterate of the conjugate gradient method applied to equation $\boldsymbol{K}_{\boldsymbol{N}}\tilde{\boldsymbol{f}} = \boldsymbol{y}$, cf. Lemma 5.7. The standard error estimate for the conjugate gradient method, cf. [Axe96, pp. 566], yields

$$\|\hat{\boldsymbol{e}}_l\|_{\hat{\boldsymbol{W}}^{-1}} \leq 2 \left(\frac{\sqrt{\Lambda} - \sqrt{\lambda}}{\sqrt{\Lambda} + \sqrt{\lambda}}\right)^l \|\hat{\boldsymbol{e}}_0\|_{\hat{\boldsymbol{W}}^{-1}} \,.$$

Due to $\hat{\boldsymbol{f}}_0 = \boldsymbol{0}$ we conclude by $\|\hat{\boldsymbol{e}}_0\|^2_{\hat{\boldsymbol{W}}^{-1}} = \boldsymbol{y}^{\mathsf{H}}\boldsymbol{K}_{\boldsymbol{N}}^{-1}\boldsymbol{y} \leq \lambda^{-1}\|\boldsymbol{y}\|_2^2$. ☐

We remark that the estimate (5.9) can be generalised for an arbitrary initial guess $\hat{\boldsymbol{f}}_0$. In general, Algorithm 5.2 minimises in each step the native error $\|\hat{\boldsymbol{e}}_l\|_{\hat{\boldsymbol{W}}^{-1}}$ over the current Krylov subspace, whereas the residual $\boldsymbol{r}_l = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_l$ might exhibit oscillations, see [Bjö96, pp. 288]. Lemma 5.9 includes the special case of $M = |I_{\boldsymbol{N}}|$ equispaced nodes $\mathcal{X} = \boldsymbol{N}^{-1} \odot I_{\boldsymbol{N}}$ and equal weights $\hat{w}_{\boldsymbol{k}} = M^{-1}$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$, where the first iterate of Algorithm 5.2 is already the solution to the interpolation problem (5.7).

Remark 5.10. *The weighted norm in* (5.7) *is induced by the inner product*

$$\langle f, g\rangle_{\hat{\boldsymbol{W}}^{-1}} := \hat{\boldsymbol{g}}^{\mathsf{H}}\hat{\boldsymbol{W}}^{-1}\hat{\boldsymbol{f}} = \sum_{\boldsymbol{k}\in I_{\boldsymbol{N}}} \frac{\hat{f}_{\boldsymbol{k}}\overline{\hat{g}_{\boldsymbol{k}}}}{\hat{w}_{\boldsymbol{k}}}$$

on the space of trigonometric polynomials. This inner product also makes $T_{\boldsymbol{N}}$ to a reproducing kernel Hilbert space, its reproducing kernel is given by $K_{\boldsymbol{N}}$, cf. Definition 2.10. In particular, *the point evaluations obey* $f(\boldsymbol{x}) = \langle f, K_N(\cdot - \boldsymbol{x})\rangle_{\hat{\boldsymbol{W}}^{-1}}$. *Analogously to Theorem 5.2 for the least squares approximation, the solution* $f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in I_N} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}$ *of the optimal interpolation problem* (5.7) *has comparable norm to the given samples, i.e.,*

$$\Lambda^{-1} \|\boldsymbol{y}\|_2^2 \le \langle f, f\rangle_{\hat{\boldsymbol{W}}^{-1}} \le \lambda^{-1} \|\boldsymbol{y}\|_2^2 .$$

This norm equivalence is due to the identity

$$\boldsymbol{y}^{\mathsf{H}} \boldsymbol{K}_N^{-1} \boldsymbol{y} = \tilde{\boldsymbol{f}}^{\mathsf{H}} \boldsymbol{K}_N \tilde{\boldsymbol{f}} = \hat{\boldsymbol{f}}^{\mathsf{H}} \hat{\boldsymbol{W}}^{-1} \hat{\boldsymbol{f}} = \langle f, f\rangle_{\hat{\boldsymbol{W}}^{-1}}.$$

Subsequently, we derive estimates for the extremal eigenvalues λ, Λ dependent only on the localisation of the kernel $K_{\boldsymbol{N}}$ and the separation distance $q_{\mathcal{X}}$ of the sampling set. Simply put, a localised kernel yields a diagonal dominated kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$ for well separated nodes.

We prove stability results for the trigonometric interpolation problem at q-separated nodes in the univariate case, cf. Theorem 5.11, and the multivariate case, cf. Theorem 5.16. Furthermore, we prove stability results for a slightly generalised interpolation problem at equispaced nodes and subsets of equispaced nodes, cf. Theorem 5.18. These results are applied to the kernels from Section 2.1.2 in Corollary 5.19 and Corollary 5.20. Analogous interpolation problems on the hyperbolic cross and on the sphere are considered in Lemma 5.21 and Theorem 5.22, respectively.

The univariate setting

The following theorem gives estimates for the extremal eigenvalues of the matrix $\boldsymbol{K}_N$ under reasonable assumptions on the kernel K_N.

Theorem 5.11. *Let* $N \in \mathbb{N}$ *be given and let the normalised kernel* K_N, *cf. Definition 2.10, fulfil for some* $\beta > 1$ *and all* $x \in \mathbb{T} \setminus \{0\}$ *the localisation property*

$$|K_N(x)| \le \frac{C_\beta}{N^\beta |x|^\beta}.$$

Furthermore, let a q*-separated sampling set* $\mathcal{X} \subset \mathbb{T}$, *cf. Definition 2.6, be given. Then, the extremal eigenvalues* λ, Λ *of the matrix* $\boldsymbol{K}_N$ *are bounded by*

$$1 - \frac{2\zeta(\beta) C_\beta}{N^\beta q^\beta} \le \lambda \le 1 \le \Lambda \le 1 + \frac{2\zeta(\beta) C_\beta}{N^\beta q^\beta}.$$

Proof. As usual, let M *denote the number of nodes in* $\mathcal{X}$. *Due to* $K_N(0) = 1$, *cf. Definition 2.10, we obtain* $\operatorname{trace}(\boldsymbol{K}_N) := \sum_{j=0}^{M-1} K_N(0) = M$. *Since the trace is invariant under similarity transforms, all eigenvalues sum up to* M *and thus, the inequality* $\lambda \le 1 \le \Lambda$ *is fulfilled.*

Now let $\lambda_\star$ *be an arbitrary eigenvalue of* $\boldsymbol{K}_N$. *Then for some index* $j \in \{0, \dots, M-1\}$ *the Gershgorin circle theorem yields*

$$|\lambda_\star - 1| \le \sum_{l=0; l\ne j}^{M-1} |K_N(x_j - x_l)| .$$

Furthermore, by using that the separation distance of the sampling set is at least q, *and by the localisation of the kernel* K_N, *we obtain*

$$|\lambda_\star - 1| \le \frac{C_\beta}{N^\beta} \sum_{l=0; l\ne j}^{M-1} \frac{1}{|x_j - x_l|^\beta} \le \frac{2\,C_\beta}{N^\beta q^\beta} \sum_{l=1}^{\lfloor M/2 \rfloor} l^{-\beta} < \frac{2\zeta(\beta) C_\beta}{N^\beta q^\beta}.$$

□

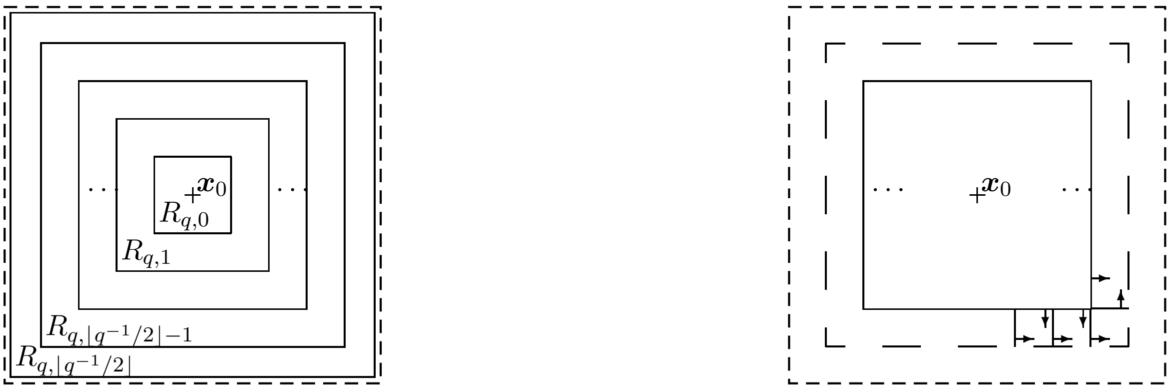

$+\boldsymbol{x}_0$
$R_{q,0}$
$R_{q,1}$
$\cdots$
$R_{q,\lfloor q^{-1}/2\rfloor-1}$
$R_{q,\lfloor q^{-1}/2\rfloor}$
$+\boldsymbol{x}_0$

Figure 5.2: Partitioning of the torus $\mathbb{T}^2$ into the rings $R_{q,m}$, $m = 0, \dots \lfloor q^{-1}/2 \rfloor$ (left). Further subdivision into shifted and rotated versions of the cube $[0, q)^d$, whereas arrows indicate the "ownership" of the faces to a particular cube (right).

Note that the kernels constructed in Section 2.1.2 meet exactly the assumptions of the previous theorem. As an immediate consequence of Theorem 5.11 we state a stability result for an equispaced grid disturbed by so-called jitter.

Corollary 5.12. Let the assumptions of Theorem 5.11 hold true. Furthermore, let the sampling nodes be of the form $x_j = -\frac{1}{2} + \frac{j-\varepsilon_j}{M}$, $j = 0, \dots, M-1$, where $0 \le \varepsilon_j \le \varepsilon < 1$. Then the extremal eigenvalues of the matrix $\boldsymbol{K}_N$ are bounded by

$$1 - \frac{2\,\zeta\,(\beta)\,C_\beta M^\beta}{N^\beta\,(1-\varepsilon)^\beta} \le \lambda \le 1 \le \Lambda \le 1 + \frac{2\,\zeta\,(\beta)\,C_\beta M^\beta}{N^\beta\,(1-\varepsilon)^\beta}\,.$$

Proof. Since the separation distance is bounded by $q \ge M^{-1}(1-\varepsilon)$ *the result follows by Theorem 5.11.* ☐

The multivariate setting

First, we show in Lemma 5.14 how many q-separated nodes can be placed in a certain distance to a reference node, see also Figure 5.2.

Definition 5.13. *For* $d \in \mathbb{N}$*, a separation distance* $q \le \frac{1}{2}$*, and* $0 \le m < \lfloor q^{-1}/2 \rfloor$*, we define the sets*

$$R_{q,m} := \left\{ \boldsymbol{x} \in \mathbb{T}^d : mq \le \operatorname{dist}(\boldsymbol{x}, \boldsymbol{0}) < (m+1)\,q \right\}$$

and

$$R_{q,\lfloor q^{-1}/2 \rfloor} := \left\{ \boldsymbol{x} \in \mathbb{T}^d : \lfloor q^{-1}/2 \rfloor\, q \le \operatorname{dist}(\boldsymbol{x}, \boldsymbol{0}) \le 1/2 \right\}.$$

Their restrictions to the sampling set $\mathcal{X}$ *will be denoted by* $R_{\mathcal{X},q,m} := R_{q,m} \cap \mathcal{X}$.

Lemma 5.14. *Let* $d \in \mathbb{N}$ *and an* q*-separated sampling set* $\mathcal{X}$ *with* $q \le \frac{1}{2}$ *be given. Then, each of the sets* $R_{\mathcal{X},q,m}$*,* $m = 1, \dots, \lfloor q^{-1}/2 \rfloor$*, has bounded cardinality*

$$|R_{\mathcal{X},q,m}| \le 2^d \left(2^d - 1\right) m^{d-1}.$$

Proof. *We use a packing argument for the partition $\{R_{q,m},\, m = 0, \dots, \lfloor q^{-1}/2 \rfloor\}$ of the torus $\mathbb{T}^d$. Each ring $R_{q,m}$ is subdivided into shifted and rotated versions of the cube $[0,q)^d$, cf. Figure 5.2 (right). This is done such that each point in $R_{q,m}$ is contained in at least one of these boxes and the boxes share no interior points with each other. Every box contains at most one node of the sampling set and hence, the estimate*

$$\begin{aligned} |R_{\mathcal{X},q,m}| &\le \frac{1}{q^d} \int_{R_{q,m}} \mathrm{d}\boldsymbol{x} \\ &\le 2^d \left((m+1)^d - m^d \right) \\ &= 2^d \sum_{t=1}^{d} \binom{d}{t} m^{d-t} \\ &\le 2^d m^{d-1} \sum_{t=1}^{d} \binom{d}{t} \end{aligned}$$

concludes our proof. □

Remark 5.15. *Using the slightly weaker packing argument, that for each node in $R_{\mathcal{X},q,m}$ the centred box around it of side length q is contained in the larger ring $\tilde{R}_{q,m} := R_{q,m-\frac{1}{2}} \cup R_{q,m+\frac{1}{2}}$ and has no interior points common with the box of another node, we might estimate*

$$|R_{\mathcal{X},q,m}| \le \frac{1}{q^d} \int_{\tilde{R}_{q,m}} \mathrm{d}\boldsymbol{x} = 2^d \left(\left(m + \frac{3}{2} \right)^d - \left(m - \frac{1}{2} \right)^d \right) = 2^d \left(3^d - 1 \right) m^{d-1}.$$

Using localised kernels, cf. Section 2.1.2, in conjunction with the previous lemma for q-separated sampling sets, we state the following theorem on the stability of the interpolation problem. The result includes Theorem 5.11 if we set $d = 1$.

Theorem 5.16. *Let $d, N \in \mathbb{N}$, and $\boldsymbol{N} = (N, \dots, N)^\top$ be given and let the normalised kernel $K_{\boldsymbol{N}}$, cf. Definition 2.10, fulfil for some $\beta > d$ and $\boldsymbol{x} \in \mathbb{T}^d \setminus \{\boldsymbol{0}\}$ the localisation property*

$$|K_N(\boldsymbol{x})| \le \frac{C_\beta}{N^\beta \|\boldsymbol{x}\|_\infty^\beta}.$$

Furthermore, let a q-separated sampling set $\mathcal{X} \subset \mathbb{T}^d$, cf. Definition 2.6, be given. Then, the extremal eigenvalues of the matrix $\boldsymbol{K_N}$ are bounded by

$$1 - \frac{2^d \left(2^d - 1\right) \zeta\left(\beta - d + 1\right) C_\beta}{N^\beta q^\beta} \le \lambda \le 1 \le \Lambda \le 1 + \frac{2^d \left(2^d - 1\right) \zeta\left(\beta - d + 1\right) C_\beta}{N^\beta q^\beta}.$$

Proof. *The inequality $\lambda \le 1 \le \Lambda$ is due to the trace argument in Theorem 5.11. Now let $\lambda_\star$ be an arbitrary eigenvalue of $\boldsymbol{K_N}$. Without loss of generality, let the diagonal element of the matrix $\boldsymbol{K_N}$ used in Gershgorin's circle theorem correspond to $\boldsymbol{x}_0 = \boldsymbol{0}$. Then we conclude by $K_{\boldsymbol{N}}(\boldsymbol{0}) = 1$, cf. Definition 2.10, that*

$$|\lambda_\star - 1| \le \sum_{l=1}^{M-1} |K_{\boldsymbol{N}}(-\boldsymbol{x}_l)|.$$

Using $q \leq \frac{1}{2}$, the partition from Definition 5.13, Lemma 5.14, and the localisation of the kernel $K_{\boldsymbol{N}}$, we proceed

$$\begin{aligned}
|\lambda_\star - 1| &\leq \sum_{m=1}^{\lfloor q^{-1}/2 \rfloor} \sum_{\boldsymbol{x}_l \in R_{\mathcal{X},q,m}} |K_N(-\boldsymbol{x}_l)| \\
&\leq \frac{2^d \left(2^d - 1\right) C_\beta}{N^\beta} \sum_{m=1}^{\lfloor q^{-1}/2 \rfloor} m^{d-1} \max_{\boldsymbol{x} \in R_{q,m}} \|\boldsymbol{x}\|_\infty^{-\beta} \\
&\leq \frac{2^d \left(2^d - 1\right) \zeta(\beta - d + 1) C_\beta}{N^\beta q^\beta}
\end{aligned}$$

what finally concludes our proof. □

Equispaced nodes

In the case of equispaced nodes we employ the fact that the kernel matrix is multilevel circulant. We present a slightly generalised result in the following Theorem 5.18.

Definition 5.17. *We define for $d \in \mathbb{N}$ and absolute summable real weights $\hat{w}_{\boldsymbol{k}} \in \mathbb{R}$, $\boldsymbol{k} \in \mathbb{Z}^d$, $\sum_{\boldsymbol{k} \in \mathbb{Z}^d} |\hat{w}_{\boldsymbol{k}}| < \infty$, the kernel*

$$K(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} \hat{w}_{\boldsymbol{k}} \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}.$$

The particular class of weights $\hat{w}_{\boldsymbol{k}} := \prod_{t=0}^{d-1} \hat{w}_{k_t}$ for absolute summable real weights $\hat{w}_{k_t} \in \mathbb{R}$, $k_t \in \mathbb{Z}$, $\sum_{k_t \in \mathbb{Z}} |\hat{w}_{k_t}| < \infty$, $t = 0, \ldots, d-1$, are called tensor product weights.

We evaluate at the equispaced sampling set $\mathcal{X} = \boldsymbol{N}^{-1} \odot I_{\boldsymbol{N}}$ and obtain the matrix

$$\boldsymbol{K} := \left(K(\boldsymbol{j} - \boldsymbol{l})\right)_{\boldsymbol{j},\boldsymbol{l} \in \mathcal{X}} \in \mathbb{C}^{|I_{\boldsymbol{N}}| \times |I_{\boldsymbol{N}}|}.$$

Theorem 5.18. *The matrix $\boldsymbol{K}$, cf. Definition 5.17, possesses the following properties. Its eigenvalues are given by*

$$\lambda_{\boldsymbol{s}}(\boldsymbol{K}) = |I_{\boldsymbol{N}}| \sum_{\boldsymbol{r} \in \mathbb{Z}^d} \hat{w}_{\boldsymbol{s}+\boldsymbol{r}\odot\boldsymbol{N}} \tag{5.10}$$

for $\boldsymbol{s} \in I_{\boldsymbol{N}}$. For tensor product weights this simplifies to

$$\lambda_{\boldsymbol{s}}(\boldsymbol{K}) = |I_{\boldsymbol{N}}| \prod_{t=0}^{d-1} \sum_{r_t \in \mathbb{Z}} \hat{w}_{s_t + r_t N_t} \tag{5.11}$$

for $\boldsymbol{s} \in I_{\boldsymbol{N}}$. Moreover, the extremal eigenvalues of

$$\boldsymbol{K}_{\mathcal{Y}} := \left(K(\boldsymbol{j} - \boldsymbol{l})\right)_{\boldsymbol{j},\boldsymbol{l} \in \mathcal{Y}} \tag{5.12}$$

are bounded by the extremal eigenvalues of $\boldsymbol{K}$ for $\mathcal{Y} \subset \mathcal{X}$.

Proof. *The matrix* $\boldsymbol{K}$ *is multilevel circulant and thus diagonalised by the Fourier matrix* $\boldsymbol{F_N}$*, cf. Theorem 2.9. We calculate*

$$\left(\boldsymbol{F}_{\boldsymbol{N}}^{\mathsf{H}}\boldsymbol{K}\boldsymbol{F}_{\boldsymbol{N}}\right)_{\boldsymbol{s},\boldsymbol{t}} = \sum_{\boldsymbol{j},\boldsymbol{l}\in\mathcal{X}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{s}\odot\boldsymbol{j}} K\left(\boldsymbol{j}-\boldsymbol{l}\right)\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{t}\odot\boldsymbol{l}}$$
$$= \sum_{\boldsymbol{k}\in\mathbb{Z}^d} \hat{w}_{\boldsymbol{k}} \sum_{\boldsymbol{j}\in\mathcal{X}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{j}\odot(\boldsymbol{s}-\boldsymbol{k})} \sum_{\boldsymbol{l}\in\mathcal{X}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{l}\odot(\boldsymbol{t}-\boldsymbol{k})}$$

for $\boldsymbol{s},\boldsymbol{t}\in I_{\boldsymbol{N}}$ *and use*

$$\sum_{\boldsymbol{j}\in\mathcal{X}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{j}\odot(\boldsymbol{s}-\boldsymbol{k})} = \begin{cases} |I_{\boldsymbol{N}}| & \text{if} \quad \boldsymbol{N}^{-1}\odot(\boldsymbol{s}-\boldsymbol{k})\in\mathbb{Z}^d, \\ 0 & \text{otherwise.}\end{cases}$$

See also [NSW98, Cor. 3.10, Thm. 3.11] for the univariate case.

The second assertion is due to the Kronecker product structure of the matrix $\boldsymbol{K}$ *in the case of tensor product kernels. The last assertion follows from the fact that removing a node is nothing else than removing its corresponding row and column in* $\boldsymbol{K}$ *and from the interlacing property for eigenvalues, see [HJ85, pp. 185].* □

Specific kernels

We apply the Theorems 5.11, 5.16, and 5.18 to the B-Spline kernels from Definition 2.15 and the kernels from Example 2.18 to obtain the following estimates on the extremal eigenvalues of the corresponding kernel matrix.

Corollary 5.19. Let $d,n\in\mathbb{N},\ n\geq 2,\ \boldsymbol{n}=(n,\ldots,n)^\top$, a polynomial degree $N\in 2\mathbb{N},\ N\geq n$, and an equispaced sampling set $\mathcal{X}=n^{-1}\boldsymbol{I_n}$ be given.

The Dirichlet kernel, cf. Example 2.18, i.e., equal weights $\hat{w}_{\boldsymbol{k}}=N^{-d}$ possesses a kernel matrix $\boldsymbol{K_N}=\boldsymbol{A}\hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}$, cf. Lemma 2.11, with extremal eigenvalues

$$\left(\frac{\lfloor Nq\rfloor}{Nq}\right)^d = \lambda \leq 1 \leq \Lambda = \left(\frac{\lceil Nq\rceil}{Nq}\right)^d,$$

where $q=n^{-1}$ denotes the separation distance of $\mathcal{X}$. Furthermore, the Fejér kernel, cf. Definition 2.15, yields

$$\left(1-\frac{1}{N^2q^2}\right)^d \leq \lambda \leq 1 \leq \Lambda \leq \left(1+\frac{1}{N^2q^2}\right)^d.$$

Proof. *First, let us prove the assertion for the univariate case* $d=1$*. We extend the weights* $\hat{w}_k=N^{-1}$*,* $k\in I_N$*, of the univariate Dirichlet kernel by* $\hat{w}_k=0$ *for* $k\in\mathbb{Z}\setminus I_N$ *and apply the identity* (5.10) *of Theorem 5.18.*

The assertion is little more delicate for the univariate Fejér kernel. We use the representation

$$|B_{2,N}(x)| = \frac{4}{N^2}\sum_{r=0}^{\frac{N}{2}-1}\sum_{k=-r}^{r}\mathrm{e}^{2\pi\mathrm{i}kx}.$$

Now, let $\lambda_\star$ *be an arbitrary eigenvalue of the kernel matrix* $\boldsymbol{K}_N$*, then Gershgorin's circle theorem yields*

$$|\lambda_\star-1| \leq \sum_{l=1}^{n-1}\left|B_{2,N}\left(\frac{l}{n}\right)\right| = \frac{4n}{N^2}\sum_{r=0}^{\frac{N}{2}-1}\left(2\left\lfloor\frac{r}{n}\right\rfloor+1\right)-1.$$

Since for $Q := \left\lfloor \frac{N-2}{2n} \right\rfloor$ *and* $R := \frac{N}{2} - 1 - nQ$ *the identity*

$$\sum_{r=0}^{\frac{N}{2}-1} \left\lfloor \frac{r}{n} \right\rfloor = \sum_{s=0}^{Q-1} \sum_{r=sn}^{(s+1)n-1} s + \sum_{r=nQ}^{nQ+R} Q = \frac{(N-n)^2 - (2(R+1)-n)^2}{8n}$$

holds, we proceed

$$\begin{aligned} \frac{4n}{N^2} \sum_{r=0}^{\frac{N}{2}-1} \left(2 \left\lfloor \frac{r}{n} \right\rfloor + 1 \right) - 1 &= \frac{4n}{N^2} \left(2 \frac{(n-N)^2 - (2(R+1)-n)^2}{8n} + \frac{N}{2} \right) - 1 \\ &= \frac{n^2}{N^2} - \left(\frac{2(R+1)-n}{N} \right)^2 \\ &\leq \frac{n^2}{N^2}. \end{aligned}$$

The case $d > 1$ *for the Dirichlet as well as for the Fejér kernel is due to the identity* (5.11) *in Theorem 5.18.* □

Thus, we obtain a nonsingular kernel matrix for $N > q^{-1}$. In the case of the Fejér kernel, one can show equality for the upper and lower bounds if $N = (2\sigma + 1)n$, $\sigma \in \mathbb{N}$. Note furthermore, that the above inequalities remain true if an arbitrary subset of the nodes is removed, see (5.12) in Theorem 5.18. Next, we give estimates for the stability of interpolation problem at arbitrary nodes.

Corollary 5.20. Let $d \in \mathbb{N}$ and a q-separated sampling set $\mathcal{X} \subset \mathbb{T}^d$ be given, cf. Definition 2.6. Then the extremal eigenvalues of the kernel matrices $\boldsymbol{K_N}$, $\boldsymbol{N} = (N, \dots, N)^\top$, $N \in 2\mathbb{N}$, are bounded as follows.

1. The Dirichlet kernel possesses for $d = 1$ and $N > (1 + |\log(2q)|)\, q^{-1}$ bounded extremal eigenvalues
$$0 < 1 - (1 + |\log(2q)|) \frac{1}{Nq} \leq \lambda \leq 1 \leq \Lambda \leq 1 + (1 + |\log(2q)|) \frac{1}{Nq}.$$
2. The B-Spline kernel of order $\beta = d + 1$ possesses for $N \geq 2\beta$ and $N > 2dq^{-1}$ bounded extremal eigenvalues
$$0 < 1 - \left(\frac{2d}{Nq} \right)^{d+1} \leq \lambda \leq 1 \leq \Lambda \leq 1 + \left(\frac{2d}{Nq} \right)^{d+1}.$$
3. The Jackson kernel of order $\beta = 2\lceil \frac{d+1}{2} \rceil$ possesses for $N > 2.1dq^{-1}$ bounded extremal eigenvalues
$$0 < 1 - \left(\frac{2.1d}{Nq} \right)^{d+1} \leq \lambda \leq 1 \leq \Lambda \leq 1 + \left(\frac{2.1d}{Nq} \right)^{d+1}.$$

Proof. We apply Theorem 5.16 for 2. and 3. where we use the estimates for C_β *given in Corollary 2.16 and Example 2.18. The first assertion follows along the same lines, i.e., we use the decay as given in Example 2.18 and only change the last step of Theorem 5.11 where we use*

$$\sum_{l=1}^{\lfloor M/2 \rfloor} l^{-1} \leq 1 + \ln \frac{M}{2}$$

and $M \leq q^{-1}$. □

Generalisation on the hyperbolic cross and on the sphere

Localised kernels and well separated sampling sets yield stable interpolation for trigonometric polynomials on the hyperbolic cross and polynomials on the sphere as well.

Lemma 5.21. *Let a sampling set $\mathcal{X} \subset \mathbb{T}^2$ be q^*-separated, cf. Definition 2.20, and the normalised hyperbolic kernel $K_N^{\mathcal{H}}$, cf. Definition 2.23, obey the localisation*

$$\left|K_N^{\mathcal{H}}(\boldsymbol{x})\right| \leq \frac{C_\beta}{2N^\beta}\left(|x_0|^{-\beta} + |Nx_0x_1|^{-\beta} + |x_1|^{-\beta}\right) \tag{5.13}$$

as stated in Lemma 2.24. Then for $N > \sqrt[\beta]{3C_\beta\zeta(\beta)}q^{-1}$ the corresponding kernel matrix

$$\boldsymbol{K}_N^{\mathcal{H}} = \left(K_N^{\mathcal{H}}(\boldsymbol{x}_j - \boldsymbol{x}_l)\right)_{j,l=0,\dots,M-1}$$

has bounded eigenvalues $1 - 3C_\beta\zeta(\beta)N^{-\beta}q^{-\beta} \leq \lambda(\boldsymbol{K}_N^{\mathcal{H}}) \leq 1 + 3C_\beta\zeta(\beta)N^{-\beta}q^{-\beta}$ and is positive definite.

Proof. The Gershgorin theorem yields, assuming $\boldsymbol{x}_0 = \boldsymbol{0}$,

$$\left|\lambda(\boldsymbol{K}_N^{\mathcal{H}}) - 1\right| \leq \sum_{j=1}^{M-1}\left|K_N^{\mathcal{H}}(\boldsymbol{x}_j)\right|.$$

We partition the torus in boxes of side-length q and note that each row and each column are allowed to contain at most one sampling node, respectively. Owing to the mixed term in the localisation estimate (5.13), the worst configuration of sampling nodes is "diagonal", i.e., we estimate further

$$\left|\lambda(\boldsymbol{K}_N^{\mathcal{H}}) - 1\right| \leq \frac{C_\beta}{N^\beta}\sum_{j=1}^{\infty}\left((jq)^{-\beta} + \left(jNq^2\right)^{-\beta} + (jq)^{-\beta}\right) \leq \frac{3C_\beta\zeta(\beta)}{N^\beta q^\beta},$$

what already concludes our proof. □

Theorem 5.22. *Let a sampling set $\mathcal{X} \subset \mathbb{S}^2$ be q-separated, cf. Definition 2.28, and the normalised kernel on the sphere $K_N^{\mathcal{S}}$, cf. Definition 2.32, obey the localisation*

$$K_N^{\mathcal{S}}(\boldsymbol{\eta}, \boldsymbol{\xi}) \leq \tilde{C}_\beta\left|(N+1)\frac{\arccos(\boldsymbol{\eta}\cdot\boldsymbol{\xi})}{2\pi}\right|^{-\beta}$$

as stated in Theorem 2.35. Then the kernel matrix

$$\boldsymbol{K}_N^{\mathcal{S}} = \left(K_N^{\mathcal{S}}(\boldsymbol{\xi}_j, \boldsymbol{\xi}_l)\right)_{j,l=0,\dots,M-1},$$

cf. Definition 2.32, has bounded eigenvalues $1 - \eta \leq \lambda(\boldsymbol{K}_N^{\mathcal{S}}) \leq 1 + \eta$ with

$$\eta := 25\tilde{C}_\beta\zeta(\beta - 1)\left(\frac{2\pi}{(N+1)q}\right)^\beta. \tag{5.14}$$

Proof. *Similar to Lemma 5.14 and [NSW98, Thm 2.3], we prove how many q-separated nodes can be placed in a certain distance to the north pole $\boldsymbol{\xi}_0 = (0,0,1)^\top$. For a separation distance $q \leq \pi$, and $0 \leq m < \lfloor \pi q^{-1} \rfloor$, we define the sets*

$$S_{q,m} := \left\{ \boldsymbol{\xi} \in \mathbb{S}^2 : mq \leq \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}, \boldsymbol{\xi}_0) < (m+1)\, q \right\}$$

and

$$S_{q,\lfloor \pi q^{-1} \rfloor} := \left\{ \boldsymbol{\xi} \in \mathbb{S}^2 : \lfloor \pi q^{-1} \rfloor\, q \leq \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{\xi}, \boldsymbol{\xi}_0) \leq \pi \right\}.$$

Their restrictions to the sampling set $\mathcal{X}$ will be denoted by $S_{\mathcal{X},q,m} := S_{q,m} \cap \mathcal{X}$. Analogously to Remark 5.15, we use [NSW98, Thm. 2.3] that states that for each node in $S_{\mathcal{X},q,m}$ the centred cap around it of colatitude $q/2$ is contained in the larger ring $\tilde{S}_{q,m} := S_{q,m-\frac{1}{2}} \cup S_{q,m+\frac{1}{2}}$ and has no interior points common with the cap of another node. Hence, we estimate for $m = 1, \ldots, \lfloor \pi q^{-1} \rfloor - 2$

$$|S_{\mathcal{X},q,m}| \leq \frac{\tilde{A}_{q,m}}{A_{q,0}} = \frac{\int_{(m-\frac{1}{2})q}^{(m+\frac{3}{2})q} \sin\theta \,\mathrm{d}\theta}{\int_0^{\frac{q}{2}} \sin\theta \,\mathrm{d}\theta} = \frac{\cos\left((2m-1)\frac{q}{2}\right) - \cos\left((2m+3)\frac{q}{2}\right)}{1 - \cos\frac{q}{2}},$$

where $\tilde{A}_{q,m}$ and $A_{q,0}$ denote the surface area of the sets $\tilde{S}_{q,m}$ and $S_{q,0}$, respectively. Using an identity for the de la Vallée Poussin kernel, see e.g. [PS01b, equation (3.4) and (3.5)], we calculate further

$$\begin{aligned}
|S_{\mathcal{X},q,m}| &\leq \frac{\cos\left((2m-1)\frac{q}{2}\right) - \cos\left((2m+3)\frac{q}{2}\right)}{1 - \cos\frac{q}{2}} \\
&= \frac{\sin\left((2m+1)\frac{q}{2}\right)\sin\left(2\frac{q}{2}\right)}{\sin^2\frac{q}{2}} \\
&= 4\left(1 + 2\sum_{l=1}^{2m-1} \cos\frac{lq}{2} + \frac{3}{2}\cos\frac{2mq}{2} + \cos\frac{(2m+1)\,q}{2} + \frac{1}{2}\cos\frac{(2m+2)\,q}{2}\right) \\
&\leq 8\,(2m+1)\,.
\end{aligned}$$

In conjunction with a similar argument, starting from the south pole, see also the first estimate in [NSW98, inequality (2.32)], i.e.,

$$|S_{\mathcal{X},q,\lfloor \pi q^{-1} \rfloor - 1} \cup S_{\mathcal{X},q,\lfloor \pi q^{-1} \rfloor}| \leq \frac{1 - \cos\left(5\frac{q}{2}\right)}{1 - \cos\left(\frac{q}{2}\right)} \leq 25,$$

we obtain the general estimate $|S_{\mathcal{X},q,m}| \leq 25m$ for $m = 1, \ldots, \lfloor \pi q^{-1} \rfloor$.

Now, the Gershgorin theorem yields, assuming $\boldsymbol{\xi}_0 = (0,0,1)^\top$,

$$\begin{aligned}
\left|\lambda(\boldsymbol{K}_N^{\mathcal{S}}) - 1\right| &\leq \sum_{j=1}^{M-1} \left|K_N^{\mathcal{S}}(\boldsymbol{\xi}_0, \boldsymbol{\xi}_j)\right| \\
&\leq \sum_{m=1}^{\lfloor \pi q^{-1} \rfloor} 25m \max_{\boldsymbol{\xi} \in S_{q,m}} \left|K_N^{\mathcal{S}}(\boldsymbol{\xi}_0, \boldsymbol{\xi})\right| \\
&\leq \sum_{m=1}^{\lfloor \pi q^{-1} \rfloor} 25m \tilde{C}_\beta \left((N+1)\frac{mq}{2\pi}\right)^{-\beta} \\
&\leq 25 \tilde{C}_\beta \zeta(\beta - 1) \left(\frac{2\pi}{(N+1)\,q}\right)^{\beta},
\end{aligned}$$

where we apply Theorem 2.35 to bound the quantity $\left|K_N^{\mathcal{S}}(\boldsymbol{\xi}_0, \boldsymbol{\xi})\right|$. □

We are now ready to prove that a polynomial degree, bounded with respect to the inverse of the separation distance, suffices for the uniqueness of the interpolation problem (5.7). Similar to the second item in Corollary 5.20, a parameter $\beta = 3$ seems most appropriate for the two-dimensional sphere.

Corollary 5.23. Let $N \in \mathbb{N}$, $N \geq 2$, and the diagonal matrix

$$\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{\boldsymbol{w}}), \ \hat{\boldsymbol{w}} = (\hat{w}_k)_{k=0,\ldots,N,\, n=-k,\ldots,k} \in \mathbb{R}^{(N+1)^2},$$

$$\hat{w}_k = \frac{1}{\|g_3\|_{1,2(N+1)}} \sum_{l=k}^{N} C_{k,l} \cdot g_3\left(\frac{l}{2(N+1)}\right)$$

be given, where g_3 denotes the normalised quadratic B-spline, cf. Definition 2.15, and the coefficients $C_{k,l}$ are given in Lemma 2.34. Then for sampling sets $\mathcal{X} \subset \mathbb{S}^2$ with bounded separation distance $q > 1.7 \cdot 2\pi(N+1)^{-1}$ the kernel matrix $\boldsymbol{K}_N^{\mathcal{S}} = \boldsymbol{Y}_{\mathcal{X}} \hat{\boldsymbol{W}} \boldsymbol{Y}_{\mathcal{X}}^{\mathsf{H}}$ is positive definite.

Proof. Applying Lemma 2.13 yields

$$\frac{1}{\|g_3\|_{1,2(N+1)}} \left| \sum_{l=-N}^{N} g_3\left(\frac{l}{2(N+1)}\right) \mathrm{e}^{-2\pi \mathrm{i} l x} \right| \leq \frac{189\zeta(3)}{64\pi^3 - 8\zeta(3)} \cdot |(N+1)\, x|^{-3}.$$

Hence, we have an explicit estimate with $\tilde{C}_3 = 189\zeta(3)(64\pi^3 - 8\zeta(3))^{-1}$ *in Theorem 2.35. Applying Theorem 5.22 with this constant in* (5.14) *yields the assertion.* ☐

5.2.3 Numerical experiments

In this section, we exemplify our findings on the stability of the optimal interpolation problem (5.7) and its iterative solution by Algorithm 5.2.

Example 5.24. The estimates for the condition number of the kernel matrix $\boldsymbol{K}_N$ for equispaced nodes, cf. Corollary 5.19, are shown in Figure 5.3 (left). For $Nq \in \mathbb{N}$ and the univariate Dirichlet kernel D_N the matrix $\boldsymbol{K}_N$ is just the identity. However, using the better localised Fejér kernel $B_{2,N}$ improves the condition number already for $N > \sqrt{3}q^{-1}$ when $Nq \notin \mathbb{N}$.

We present the effect on the stability of the interpolation problem when the equispaced nodes are perturbed by jitter error, cf. Corollary 5.12, in Figure 5.3 (right). We choose different sampling sets of size $M = 1, \ldots, 100$ with equispaced nodes disturbed by 10% jitter error and evaluate the maximum condition number over 100 reruns for the Dirichlet kernel D_{6M} and the Fejér kernel $B_{2,6M}$, respectively. The Fejér kernel produces a lower condition number which is also validated by the shown upper bound. These results confirm the theoretical results of Corollary 5.19 and Corollary 5.12.

Example 5.25. Furthermore, we apply Algorithm 5.2 to reconstruct a univariate signal from randomly taken samples. Figure 5.4 shows how the error decays. In all cases, the scheme converges within 15 iteration, whereas this is justified only for the Fejér- and for the B-spline kernel.

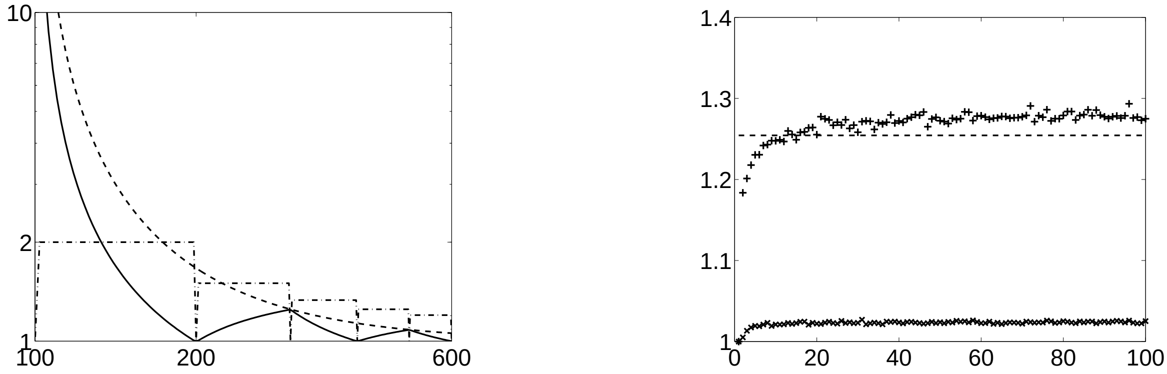


Figure 5.3: Condition number of the kernel matrix $\boldsymbol{K}_N$ for the univariate case $d = 1$. Left: Condition number with respect to polynomial degree $N = 100, \ldots, 600$, equal weights, i.e., Dirichlet kernel (dash-dot); weight function g_2, i.e., Fejér kernel (solid), and the estimate of Corollary 5.19 (dashed); here, the number of equispaced nodes is $M = 100$. Right: Condition number with respect to the number of nodes $M = 1, \ldots, 100$, the nodes are equispaced perturbed by $\varepsilon_{\text{rel.}} = 0.1$ jitter error, the polynomial degree is $N = 6M$; no weights, i.e., Dirichlet kernel (+); weight function g_2, i.e., Fejér kernel (×), and its estimate by Corollary 5.12 (dashed).

Example 5.26. The last example shows a typical test case known in radial basis function methods. We reconstruct from a data set of $M = 8345$ samples on level curves of a glacier a total number of $2^{16} \approx 8M$ Fourier coefficients. Note however, that the sampling set is highly nonuniform in the sense that the separation distance is very small compared to the mesh norm. The assumptions of Theorem 5.16 are not fulfilled, nevertheless, the proposed method yields a very good approximation to the given data after 40 iterations.

5.2.4 Concluding remarks

We have shown that the optimal interpolation problem with trigonometric polynomials at q-separated nodes in d dimensions is well conditioned for a polynomial degree $N \geq 2dq^{-1}$. Note however, that the condition $N > 2dq^{-1}$, sufficient for a positive definite kernel matrix $\boldsymbol{K}_{\boldsymbol{N}}$, is not optimal for high dimensional problems. Similar to the least squares approximation (5.3), it is an open problem to obtain a condition independent of d. Estimates for the extremal eigenvalues in case of the interpolation by radial and zonal functions were obtained in [NSW98, Wen05].

If we further assume a quasi uniform sequence $\{\mathcal{X}_M\}_{M \in \mathbb{N}}$ of sampling sets, cf. Definition 2.6, then the total arithmetical complexity for solving (5.7) by means of Algorithm 5.2 up to a prescribed error is of order $\mathcal{O}(M \log M)$.

We generalise our considerations from positive to non-negative damping factors $\hat{w}_{\boldsymbol{k}} \geq 0$, $\boldsymbol{k} \in I_{\boldsymbol{N}}$ in (5.7) by defining the *seminorm-optimal interpolation problem*

$$\|\hat{\boldsymbol{f}}\|_{\hat{\boldsymbol{W}}^{-1}} \xrightarrow{\hat{\boldsymbol{f}}} \min \quad \text{subject to} \quad \boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}, \tag{5.15}$$

where $\hat{\boldsymbol{W}}^{-1}$ denotes the pseudoinverse of $\hat{\boldsymbol{W}}$, see also the seminorm-penalised least squares problem (5.6). We state the following lemma on solutions of (5.15).

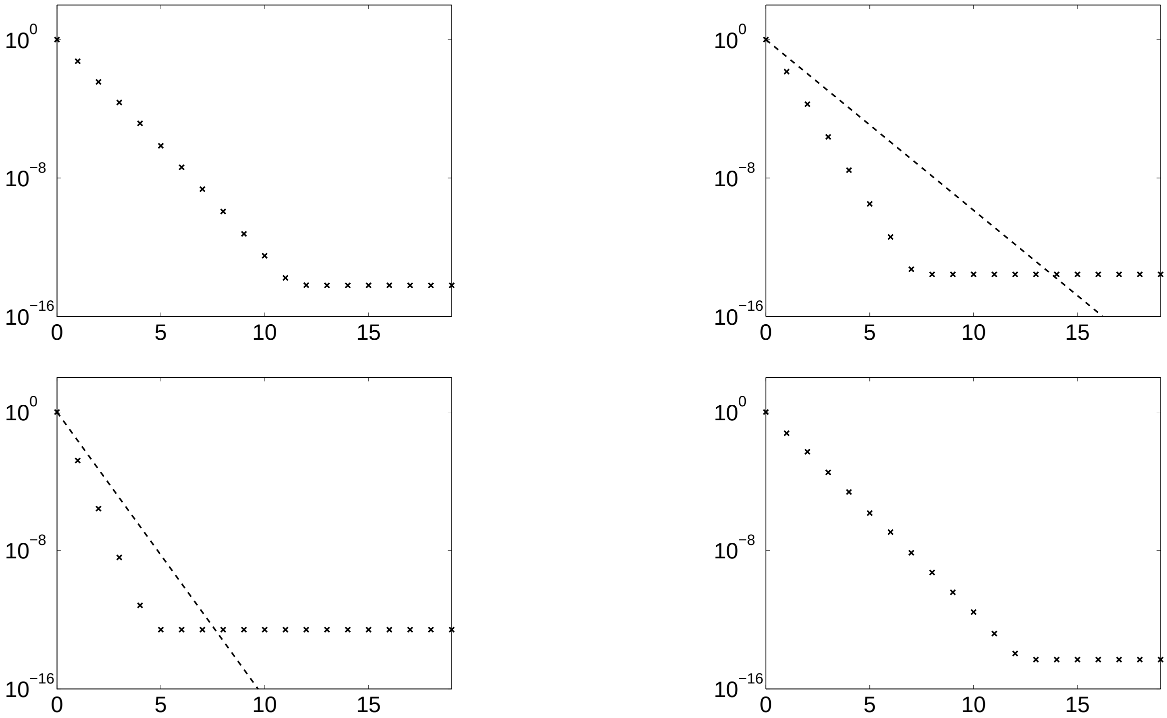


Figure 5.4: Native error $\|\hat{\boldsymbol{f}}_l - \hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{K}_N^{-1}\boldsymbol{y}\|_{\hat{\boldsymbol{W}}^{-1}}$ for the univariate interpolation problem with respect to the current iteration l. The number of samples is $M = 100$, the number of computed Fourier coefficients is $N = 1000$, and the separation distance of the nodes is $q = 4\times 10^{-3}$. Top left: no weights, i.e., Dirichlet kernel; Top right: weight function g_2, i.e., Fejér kernel, predicted decay rate (dashed); Bottom left: weight function g_4, i.e., cubic B-spline kernel, predicted decay rate (dashed); Bottom right: weight function $g_{1,2,10^{-2}}$, i.e., Sobolev kernel.

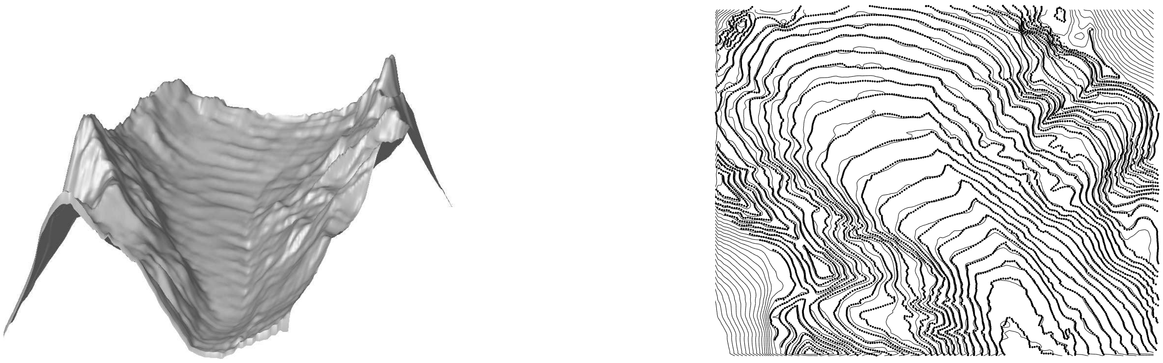


Figure 5.5: Reconstruction of the glacier data set `vol87.dat` from [Fra], $M = 8345$ nodes, $N = 256$ as polynomial degree, 40 iterations, tensor product damping factors $\hat{w}_{\boldsymbol{k}}$ to the weight function $g_{\frac{1}{2},3,10^{-3}}$. Left: surface plot of the reconstructed trigonometric polynomial, Right: contour plot of the polynomial with level curves at heights of the given samples, the sampling nodes are shown dotted.

Lemma 5.27. *Problem* (5.15) *has a solution for every right hand side* $\boldsymbol{y} \in \mathbb{C}^M$ *if and only if the nonequispaced Fourier matrix has* $\operatorname{rank}(\boldsymbol{A}) = M$. *This solution is unique if and only if the null-spaces* $\mathcal{N}(\boldsymbol{A}) := \{\hat{\boldsymbol{f}} \in \mathbb{C}^{|I_N|} : \ \boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{0}\}$ *and* $\mathcal{N}(\hat{\boldsymbol{W}})$ *intersect only trivially.*

Proof. The equality constraints $\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y}$ *are consistent for every* $\boldsymbol{y} \in \mathbb{C}^M$ *if and only if* $\boldsymbol{A}$ *has* M *linearly independent columns. At least one of these solutions attains minimal semi-norm. Since*

$$\left\|\hat{\boldsymbol{f}}\right\|_{\hat{\boldsymbol{W}}^{-1}} = \left\|\hat{\boldsymbol{f}} + \hat{\boldsymbol{f}}_*\right\|_{\hat{\boldsymbol{W}}^{-1}} \Leftrightarrow \hat{\boldsymbol{f}}_0 \in \mathcal{N}\left(\hat{\boldsymbol{W}}^{-1}\right) = \mathcal{N}\left(\hat{\boldsymbol{W}}\right),$$
$$\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{A}\left(\hat{\boldsymbol{f}} + \hat{\boldsymbol{f}}_*\right) \Leftrightarrow \hat{\boldsymbol{f}}_* \in \mathcal{N}\left(\boldsymbol{A}\right),$$

a vector $\hat{\boldsymbol{f}}$ *remains a solution of problem* (5.15) *if and only if it is perturbed by a vector* $\hat{\boldsymbol{f}}_* \in \mathcal{N}(\boldsymbol{A}) \cap \mathcal{N}(\hat{\boldsymbol{L}})$ ☐

5.3 Generalisation and application

In this section, we report on a particular version of the inverse NFFT that has been developed recently in our joint paper [KR06b] and is available as MATLAB toolbox [KR06a]. We propose Algorithm 5.3 for the recovery of trigonometric polynomials with few non-zero Fourier coefficients. Such trigonometric polynomials are commonly called sparse or compressible and surprisingly, they can be reconstructed from a small number of taken samples.

Moreover, we present a simple application of the inverse NFFT, more precisely Algorithm 5.1, in magnetic resonance imaging, see also [KKP05]. Besides showing the equivalence of gridding [JMNM91] and the first iterate of our algorithm, we give numerical evidence that density compensation weights are useful within the iterative reconstruction from data acquired by an MR scanner and for simulated data.

5.3.1 Sparse reconstruction

Recently, the surprising fact that it is possible to recover functions having only few non-zero coefficients with respect to some basis from vastly incomplete information has gained much attention. Such functions are commonly called sparse or compressible and they naturally appear in a wide range of applications. We deal with nonlinear space of *sparse trigonometric polynomials*

$$T_{\leq S} := \bigcup_{\Omega \subset I_N, |\Omega| \leq S} \operatorname{span}\left\{x \mapsto \mathrm{e}^{-2\pi \mathrm{i} k x} : \ k \in \Omega\right\},$$

i.e., we assume that the sequence of coefficients $\hat{f}_k$ is supported only on a set $\Omega \subset I_N$, which is much smaller than $|I_N|$. A priori nothing is known about Ω apart from a maximum size. The following Figure 5.6 shows a sparse trigonometric polynomial.

Our aim is to sample a trigonometric polynomial $f \in T_{\leq S}$ at M nodes and try to reconstruct f from these samples - at least in the majority of all cases. If the sparsity S is small and the dimension N large, then we hope that a number M of samples much smaller than N suffices for reconstruction. In addition to our previous definitions, let the *restricted* nonequispaced Fourier matrix be given by $(\boldsymbol{A}_{\mathcal{X},\Omega})_{j,k} := \mathrm{e}^{-2\pi \mathrm{i} k x_j}$ for $j = 0, \ldots, M-1$ and $k \in \Omega$. We suggest Algorithm 5.3 (orthogonal matching pursuit, OMP) as an inverse NFFT for trigonometric polynomials with few non-zero Fourier coefficients.

Algorithm 5.3 contains two costly computations. The multiplication of the adjoint "measurement matrix" $\boldsymbol{A}^{\mathsf{H}}$ with the current residual vector $\boldsymbol{r}_l$ takes $\mathcal{O}(N \log N + M)$ by using Algorithm 3.2. Moreover, we have to solve the system of linear equations $\boldsymbol{A}_{\mathcal{X},\Omega_l}^{\mathsf{H}} \boldsymbol{A}_{\mathcal{X},\Omega_l} \hat{\boldsymbol{f}}_l = \boldsymbol{A}_{\mathcal{X},\Omega_l}^{\mathsf{H}} \boldsymbol{y}$

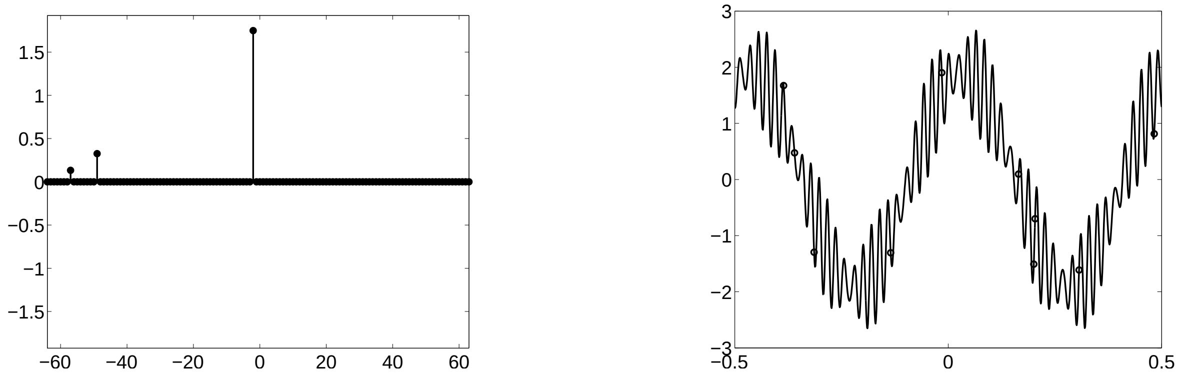


Figure 5.6: Sparse vector of Fourier coefficients (left) and the real part of the corresponding trigonometric polynomial (right) with a few samples (circle).

Algorithm 5.3 Inverse NFFT, OMP

```
Input:  $M, N, S \in \mathbb{N}$,
        $(x_j, y_j) \in \mathbb{T} \times \mathbb{C},\ j = 0, \ldots, M-1$.

  $\boldsymbol{r}_0 = \boldsymbol{y}$
  $\Omega_0 = \emptyset$
  for $l = 1, \ldots, S$ do
    $k_l = \arg\max_{k \in I_N} |\boldsymbol{A}^{\mathsf{H}} \boldsymbol{r}_{l-1}|$
    $\Omega_l = \Omega_{l-1} \cup \{k_l\}$.
    Solve $\boldsymbol{A}_{\mathcal{X},\Omega_l}^{\mathsf{H}} \boldsymbol{A}_{\mathcal{X},\Omega_l} \hat{\boldsymbol{f}}_l = \boldsymbol{A}_{\mathcal{X},\Omega_l}^{\mathsf{H}} \boldsymbol{y}$ for $\hat{\boldsymbol{f}}_l$
    $\boldsymbol{r}_l = \boldsymbol{f} - \boldsymbol{A}_{\mathcal{X},\Omega_l} \hat{\boldsymbol{f}}_l$
  end for
Output:      vector of coefficients $\hat{\boldsymbol{f}}_S$ and its support $\Omega_S$.
Complexity:  $\mathcal{O}(SN \log N + S^2 M)$.
```

of equivalently the least squares problem

$$\left\| \boldsymbol{A}_{\mathcal{X},\Omega_l} \hat{\boldsymbol{f}}_l - \boldsymbol{y} \right\|_2 \xrightarrow{\hat{\boldsymbol{f}}_l} \min .$$

A straightforward implementation yields costs $\mathcal{O}(MS^2)$ per iteration. Speed up for this computation is obtained by the QR factorisation of $\boldsymbol{A}_{\mathcal{X},\Omega_l}$ obtained from the factorisation of $\boldsymbol{A}_{\mathcal{X},\Omega_{l-1}}$, cf. [Bjö96, pp. 132], or by the use of the iterative algorithm LSQR, cf. [PS82], reducing the costs for solving one least squares problem to $\mathcal{O}(M^2)$ or $\mathcal{O}(SM)$, respectively. The latter assertion is true for a uniformly bounded condition number of $\boldsymbol{A}_{\mathcal{X},\Omega_l}$ [Rau06]. Algorithm 5.3 takes $\mathcal{O}(SN \log N + S^2 M)$ arithmetic operation and is implemented in [KR06a]. This MATLAB-toolbox comes with a simple univariate version of the Algorithms 3.1 and Algorithm 3.2, too.

We focus on the success of reconstruction and validate the sufficiency of $M = CS \log(\frac{N}{\nu})$ randomly taken samples of a given polynomial $f \in T_{\leq S}$ to succeed with probability $1 - \nu$ where C denotes an absolute constant.

Example 5.28. For a fixed $N = 1024$, we draw a support set Ω uniformly from all the subsets of increasing sizes $S = 1, \ldots, 40$, complex-valued Fourier coefficients (normal distributed with mean zero and standard deviation one), and sampling sets of size $M = 2.5S$, $M = 3S$, and $M = 3.5S$ (uniformly distributed nodes in $\mathbb{T}$), respectively. For 200 runs of each experiment, we count the number of perfect reconstructions. As Figure 5.7 (left) reveals, the success rate stays (almost) constant or might even increase slightly for an increasing number of non-vanishing coefficients if the ratio $\theta = M/S$ remains constant.

In the second part of this example, we are concerned with the dependence of this ratio $\theta = M/S$ to reach a certain success rate when N increases. For an increasing number $N = 2^6, 2^7, \ldots, 2^{14}$, we draw sets Ω of sizes $S = 4, 8, 16, 32$ and test for the *smallest* number M of samples, such that 90% (180 out of 200) of the runs result in a perfect recovery of the given Fourier coefficients. Figure 5.7 (right) confirms the relation $\theta = C \log_2(N)$ to reach a fixed success rate, whereas the constant $C \leq \frac{2}{3}$ even decreases mildly for a larger number S of non-zero coefficients.

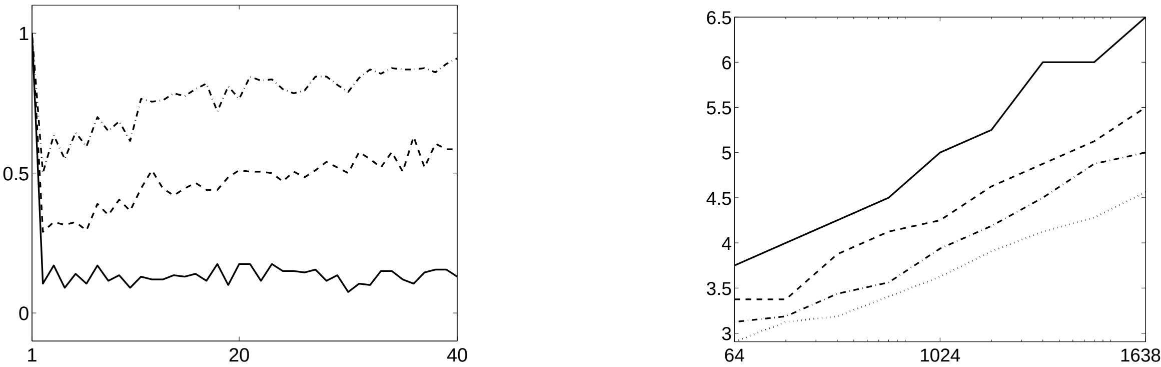


Figure 5.7: Left: The rate of successful reconstruction with respect to number S of non-zero Fourier coefficients for three ratios $M/S =: \theta = 2.5$ (solid), $\theta = 3$ (dashed), and $\theta = 3.5$ (dash-dot), where $N = 2^{10}$. Right: Ratio $\theta = M/S$, necessary to reach a success rate of 90%, with respect to fixed numbers of non-vanishing coefficients $S = 4$ (solid), $S = 8$ (dashed), and $S = 16$ (dash-dot) and increasing N.

5.3.2 Gridding in magnetic resonance imaging

The primal goal of early NFFT-approaches, denoted as gridding, cf. [O'S85, JMNM91, ST95], was an approximate inverse Fourier transform computed from nonequispaced samples. Typically, this has been done by a density compensation followed by the adjoint NFFT, i.e.,

$$\hat{\boldsymbol{f}}_* := \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{y}. \tag{5.16}$$

Such gridding approaches still play a crucial role in computerised tomography and magnetic resonance imaging, cf. [SBC01, BBZ02]. On the other hand, iterative image reconstruction algorithms are used in modern tomographic systems [MFK04] and have been applied in combination with nonequispaced FFTs in magnetic resonance imaging, cf. [SFN01, SNF03].

Iterate our particular inverse NFFT Algorithm 5.1 for only one step, we obtain an "optimised" gridding solution. As the following lemma reveals, the gridding solution $\hat{\boldsymbol{f}}_*$ in (5.16) is scaled such that its residual is minimised.

Lemma 5.29. *Let $\hat{\boldsymbol{f}}_0 = \boldsymbol{0}$ be the initial guess in Algorithm 5.1, then the first iterate fulfils*

$$\hat{\boldsymbol{f}}_1 = \arg \min_{\hat{\boldsymbol{f}} = \alpha \hat{\boldsymbol{f}}_*} \left\| \boldsymbol{y} - \boldsymbol{A} \hat{\boldsymbol{f}} \right\|_{\boldsymbol{W}}.$$

Proof. *Note that the first iterate indeed fulfils* $\hat{\boldsymbol{f}}_1 \in \{\alpha \hat{\boldsymbol{f}}_* : \alpha \geq 0\}$. *Moreover, the first residual* $\boldsymbol{r}_1 = \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_1$ *obeys*

$$\boldsymbol{r}_1^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A} \hat{\boldsymbol{f}}_* = \left(\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{y}\right)^{\mathsf{H}} \hat{\boldsymbol{f}}_* - \frac{\left\|\hat{\boldsymbol{f}}_*\right\|_2^2}{\left\|\boldsymbol{A}\hat{\boldsymbol{f}}_*\right\|_{\boldsymbol{W}}^2} \left(\boldsymbol{A}\hat{\boldsymbol{f}}_*\right)^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A} \hat{\boldsymbol{f}}_* = 0,$$

i.e., is perpendicular to $\hat{\boldsymbol{f}}_*$. *Hence, the assertion follows.* ☐

Discretisation in magnetic resonance imaging

In magnetic resonance imaging (MRI) the raw data is measured in k-space, the domain of spatial frequencies. Samples of the MR signal lie along k-space trajectories determined by the magnetic field gradients. In contrast to the use of the computationally efficient FFT for the reconstruction from Cartesian grids, the more general sampling trajectories ask for the application of Algorithm 3.1 and 3.2. Given a trajectory $\boldsymbol{k} = \boldsymbol{k}(t)$, the relation between the MR signal s_{MR} during the readout and the object p can be modelled by the simplified signal equation

$$s(\boldsymbol{k}) = \int_{\mathbb{R}^3} p(\boldsymbol{r})\, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{r}\boldsymbol{k}} \mathrm{d}\boldsymbol{r}. \tag{5.17}$$

For convenience let the available samples in k-space be contained in the shifted unit cube, i.e. $\boldsymbol{k} \in \mathbb{T}^3$, and the field of view be restricted to $\Omega_{\boldsymbol{N}} \subset [-\frac{N_1}{2}, \frac{N_1}{2}) \times [-\frac{N_2}{2}, \frac{N_2}{2}) \times [-\frac{N_3}{2}, \frac{N_3}{2})$, where $\boldsymbol{N} = (N_1, N_2, N_3)^\top \in \mathbb{N}^3$. Then, the discretisation of integral (5.17) on equispaced points leads to

$$s(\boldsymbol{k}) \approx \tilde{s}(\boldsymbol{k}) := \sum_{\boldsymbol{r} \in I_{\boldsymbol{N}}} p(\boldsymbol{r})\, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{r}\boldsymbol{k}}. \tag{5.18}$$

Thus, the unknown object p is given *implicitly* and the authors of [DvdWL02] call this the inverse model. Sampling the MR signal s at discrete time instance t_j and asking for a least squares fit to the samples $s(\boldsymbol{k}) = s(\boldsymbol{k}(t_j))$ yields the formulation from Section 5.1, i.e., we want to solve the weighted normal equation of first kind, cf. (5.4),

$$\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A} \boldsymbol{p} = \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{s}$$

for the unknown vector $\boldsymbol{p}$.

A second possible discretisation uses the Fourier inversion theorem

$$p(\boldsymbol{r}) = \int_{\mathbb{R}^3} s(\boldsymbol{k})\, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{r}\boldsymbol{k}} \mathrm{d}\boldsymbol{k}$$

and leads to

$$p(\boldsymbol{r}) \approx \tilde{p}(\boldsymbol{r}) := \sum_{j=0}^{M-1} s(\boldsymbol{k}_j)\, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{r}\boldsymbol{k}_j} w_j,$$

with density compensation weights $w_j > 0$. Here, the unknown object $p \approx \tilde{p}$ can be computed *explicitly* by one matrix vector multiplication

$$\tilde{\boldsymbol{p}} = \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{s}.$$

Numerical experiments

We are concerned with the application to data acquired by an MR scanner and the reconstruction quality for simulated data. The sampling scheme consists of 36 equidistant planes with 48 interleaved spiral arms with 1625 data points each, i.e., $2,808,000$ k-space samples in total, whereas the reconstructed image contains $256 \times 256 \times 36 = 2,359,296$ voxels. Due to the particular structure of this sampling set, we reconstruct each slice separately by bivariate inverse NFFTs only and use 256×256 univariate FFTs of length 36 within the third component. Note furthermore that this sampling set violates a “Nyquist criteria” in the sense that the mesh norm δ is larger than the inverse field of view $\boldsymbol{N}^{-1}$.

Algorithm 5.1 has been tested with MR measurements of a physical phantom by the Philips Achieva 1.5T device. Moreover, we test the reconstruction algorithm on simulated MR data obtained by Algorithm 3.1 from the $3d$-Shepp-Logan phantom of $256 \times 256 \times 36$, cf. Figure 5.8. Comparison is done with respect to the number of iterations and sampling density compensation weights $\boldsymbol{W}$. We propose formulations

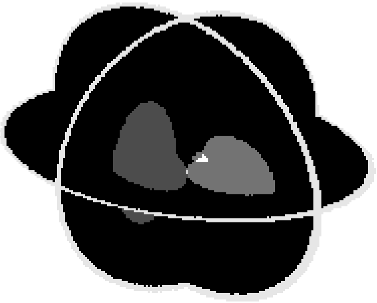


Figure 5.8: Slice plot of the $3d$-Shepp-Logan phantom.

1. with no weights, i.e. $\boldsymbol{W} = \boldsymbol{I}_M$,
2. with approximate weights obtained by counting the number of sample point in a regular partition of size 256×256 in the k-space, and
3. with weights obtained as the area of the Voronoi cell

$$\Omega_j = \left\{ \tilde{\boldsymbol{k}} \in \left[-\frac{1}{2}, \frac{1}{2}\right)^2 : \left\| \tilde{\boldsymbol{k}} - \tilde{\boldsymbol{k}}_j \right\|_2 \le \min_{\substack{l=0,\ldots,M-1,\\ l \neq j}} \left\| \tilde{\boldsymbol{k}} - \tilde{\boldsymbol{k}}_l \right\|_2 \right\}$$

around each sample point $\tilde{\boldsymbol{k}}_j$, see also [BH].

Example 5.30. In our first example, we apply Algorithm 5.1 to the MR measurements taken by the Philips Achieva 1.5T device. Figure 5.9 shows the result after one iteration, where we used Voronoi weights for sampling density compensation.

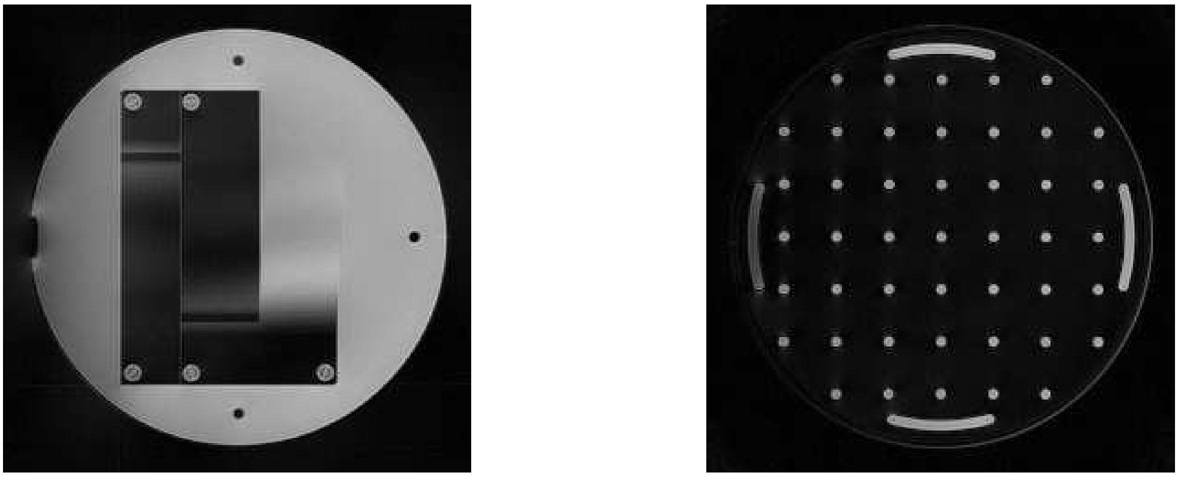


Figure 5.9: Two slices of the reconstruction from MR measurements with Voronoi weights after one iteration. Algorithm 3.1 is used with the Kaiser-Bessel window function, cut-off $m = 6$ and an oversampling factor $\sigma = 2$.

Example 5.31. Furthermore, we compare the reconstruction quality with respect to different sampling density compensation weights. Table 5.1 shows the normalised root-mean-square error

$$\mathrm{RMS}(\tilde{\boldsymbol{p}}, \boldsymbol{p}) := \frac{\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_2}{\|\boldsymbol{p}\|_2},$$

where $\tilde{\boldsymbol{p}}$ is our reconstruction and $\boldsymbol{p}$ denotes the original Shepp-Logan phantom.

weights \ iter.	1	2	5	10
none	0.7120	0.5389	0.2524	0.0803
approximation	0.1764	0.0796	0.0775	0.0774
Voronoi	0.1049	0.0781	0.0776	0.0776

Table 5.1: RMS for different weights after 1, 2, 5, and 10 iterations.

Discussion

A large number of numerical tests is presented online in conjunction with [KKP05]. These reconstructions include in particular animated graphics showing the progress during the iterations or slicing through the $3d$ data set. We see that already gridding might lead to very good results if Voronoi weights are used for sampling density compensation. Nevertheless, comparable results can be obtained by using few iterations when only an estimation of the local sampling density is available. As expected, using no weights slows down the reconstruction process seriously.

5.4 Notes and comments

The main purpose of early gridding approaches in astrophysics, tomography, and engineering was the computation of Fourier coefficients from given nonequispaced samples, see for example [O'S85, JMNM91, ST95, Pel97]. In contrast, the proposed least squares and interpolation formulations yield inverse NFFTs with specified properties. Direct solvers for the univariate trigonometric interpolation problem with computational costs $\mathcal{O}(NM)$ were derived in [RAG91, Faß97]. An approximation method in [DR95] takes $\mathcal{O}(N \log N)$ for $M = N$, samples but can be applied only for almost equispaced nodes.

On the other hand, iterative algorithms for the least squares problem (5.3) were suggested in [Grö92, Grö93, FGS95, AD96, BG04a]. In conjunction with uniformly bounded condition numbers for dense enough sampling sets and the use of the NFFT schemes, Algorithm 5.1 yields a solution with given target residual in $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$ floating point operations. Using techniques from radial basis function interpolation [NSW98, Wen05], we estimated the eigenvalues for trigonometric and spherical kernel matrices. We have shown that Algorithm 5.2 takes computational work $\mathcal{O}(|I_{\boldsymbol{N}}| \log |I_{\boldsymbol{N}}| + M)$ to obtain a certain accuracy in the sought solution for well separated sampling sets. Moreover, we would like to emphasise that Algorithm 5.1 minimises the residual $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}}_l\|_{\boldsymbol{W}}$ in each iteration over the current Krylov subspace, whereas Algorithm 5.2 minimises the native error $\|\hat{\boldsymbol{W}}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{K}_N^{-1}\boldsymbol{y} - \hat{\boldsymbol{f}}_l\|_{\hat{\boldsymbol{W}}^{-1}}$ over a related Krylov subspace. In both cases, the proven convergence rate depends on simple geometric properties of the sampling set.

The applications of inverse NFFTs range from scattered data interpolation to inverse polar FFTs [FKPar], computerised tomography, and magnetic resonance imaging, where density or separation conditions on the sampling set are typically not strictly fulfilled. Nevertheless, Algorithm 5.1 and 5.2 compute reliable approximations that improve over gridding solutions with respect to reconstruction quality, whereas only a small overhead in computations arises.

Bibliography

[AB05] F. Andersson and G. Beylkin. The fast Gauss transform with complex parameters. *J. Comput. Physics*, 203:274 – 286, 2005.

[ACD⁺06] A. Averbuch, R. Coifman, D. L. Donoho, M. Elad, and M. Israeli. Fast and accurate polar Fourier transform. *Appl. Comput. Harmon. Anal.*, 21:145 – 167, 2006.

[AD96] C. Anderson and M. Dahleh. Rapid computation of the discrete Fourier transform. SIAM *J. Sci. Comput.*, 17:913 – 919, 1996.

[Ale93] V. G. Alekseev. Jackson– and Jackson–Vallee Poussin-type kernels and their probability applications. *Theory Probab. Appl.*, 41:137 – 143, 1993.

[AS72] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions.* National Bureau of Standards, Washington, D.C., 1972.

[Axe96] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, Cambridge, 1996.

[BBZ02] M. Bronstein, A. Bronstein, and M. Zibulevsky. Iterative reconstruction in diffraction tomography using non-uniform FFT. IEEE *Trans. Med. Imag.*, 2002. submitted.

[BD89] G. Baszenski and F.-J. Delvos. A discrete Fourier transform scheme for Boolean sums of trigonometric operators. In C. K. Chui, W. Schempp, and K. Zeller, editors, *Multivariate Approximation Theory IV*, ISNM 90, pages 15–24. Birkhäuser, Basel, 1989.

[Bey95] G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comput. Harmon. Anal.*, 2:363 – 381, 1995.

[BG97] R. K. Beatson and L. Greengard. A short course on fast multipole methods. In M. Ainsworth, J. Levesley, W. A. Light, and M. Marletta, editors, *Wavelets, Multilevel Methods and Elliptic PDEs.* Clarendon Press, 1997.

[BG04a] R. F. Bass and K. Gröchenig. Random sampling of multivariate trigonometric polynomials. SIAM *J. Math. Anal.*, 36:773 – 795, 2004.

[BG04b] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147 – 269, 2004.

[BH] C. B. Barber and H. Huhdanpaa. Qhull, Softwarepackage. http://www.qhull.org.

[BH01] B. J. C. Baxter and S. Hubbert. Radial basis functions for the sphere. *Progress in Multivariate Approximation*, 137:33 – 47, 2001.

[Bjö96] Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.

[BM01] S. Bagchi and S. K. Mitra. The nonuniform discrete Fourier transform . In F. Marvasti, editor, *Nonuniform Sampling: Theory and Practice.* Kluwer/Plenum, 2001.

[BM02] G. Beylkin and L. Monzon. On generalized Gaussian quadratures for exponentials and their applications. *Appl. Comput. Harmon. Anal.*, 12:332 – 373, 2002.

[BM05] G. Beylkin and L. Monzon. On approximations of functions by exponential sums. *Appl. Comput. Harmon. Anal.*, 19:17 – 48, 2005.

[BNP05] P. J. Beatty, D. G. Nishimura, and J. M. Pauly. Rapid gridding reconstruction with a minimal oversampling ratio. IEEE *Trans. Med. Imag.*, 24:799 – 808, 2005.

[BR02] B. J. C. Baxter and G. Roussos. A new error estimate of the fast Gauss transform. SIAM *J. Sci. Comput.*, 24:257–259, 2002.

[Bra91] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Comp. Phys. Comm.*, 93:24 – 38, 1991.

[CDDY06] E. J. Candes, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. SIAM *Multiscale Model. Simul.*, 3:861 – 899, 2006.

[Chu88] C. K. Chui. *Multivariate Splines.* SIAM, Philadelphia, 1988.

[CT65] J. W. Cooley and J. W. Tukey. An algorithm for machine calculation of complex Fourier series. *Math. Comput.*, 19:297 – 301, 1965.

[DH94] J. R. Driscoll and D. Healy. Computing Fourier transforms and convolutions on the 2–sphere. *Adv. in Appl. Math.*, 15:202 – 250, 1994.

[DHR96] J. R. Driscoll, D. Healy, and D. Rockmore. Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs. SIAM *J. Comput.*, 26:1066 – 1099, 1996.

[DR93] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. SIAM *J. Sci. Stat. Comput.*, 14:1368 – 1393, 1993.

[DR95] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data II. *Appl. Comput. Harmon. Anal.*, 2:85 – 100, 1995.

[Dro96] J. O. Droese. Verfahren zur schnellen Fourier–Transformation mit nichtäquidistanten Knoten. Diplomarbeit, TU Darmstadt, 1996.

[DS99] A. J. W. Duijndam and M. A. Schonewille. Nonuniform fast Fourier transform. *Geophysics*, 64:539 – 551, 1999.

[DS00] J. Dongarra and S. Sullivan. The top 10 algorithms. *Computing in Science and Engg.*, 2:22 – 23, 2000.

[DvdWL02] B. Desplanques, R. van de Walle, and I. Lemahieu. Iterative reconstruction of magnetic resonance images from arbitrary samples in k-space. IEEE *Trans. Nucl. Sci.*, 49:2268 – 2273, 2002.

[EKPar] H. Eggers, T. Knopp, and D. Potts. Field inhomogeneity correction based on gridding reconstruction. IEEE *Trans. Med. Imag.*, to appear.

[ES98] B. Elbel and G. Steidl. Fast Fourier transform for nonequispaced data. In C. K. Chui and L. L. Schumaker, editors, *Approximation Theory IX*, Nashville, 1998. Vanderbilt University Press.

[Faß97] H. Faßbender. On numerical methods for discrete least-squares approximation by trigonometric polynomials. *Math. Comput.*, 66:719 – 741, 1997.

[Fen06] M. Fenn. Fast Fourier transform at nonequispaced nodes and applications. Dissertation, Universität Mannheim, 2006.

[FGS95] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numer. Math.*, 69:423 – 440, 1995.

[FGS98a] W. Freeden, O. Glockner, and M. Schreiner. Spherical panel clustering and its numerical aspects. *J. of Geodesy*, 72:586–599, 1998.

[FGS98b] W. Freeden, T. Gervens, and M. Schreiner. *Constructive Approximation on the Sphere*. Oxford University Press, Oxford, 1998.

[FJ] M. Frigo and S. G. Johnson. FFTW, C subroutine library. http://www.fftw.org.

[FJ05] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the* IEEE, 93:216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[FKP06] M. Fenn, S. Kunis, and D. Potts. Fast evaluation of trigonometric polynomials from hyperbolic crosses. *Numer. Algorithms*, 41:339 – 352, 2006.

[FKPar] M. Fenn, S. Kunis, and D. Potts. On the computation of the polar FFT. *Appl. Comput. Harmon. Anal.*, to appear.

[Fou03] K. Fourmont. Non equispaced fast Fourier transforms with applications to tomography. *J. Fourier Anal. Appl.*, 9:431 – 450, 2003.

[FP05] M. Fenn and D. Potts. Fast summation based on fast trigonometric transforms at nonequispaced nodes. *Numer. Linear Algebra Appl.*, 12:161 – 169, 2005.

[Fra] R. Franke. http://www.math.nps.navy.mil/∼rfranke/README.

[FS02a] M. Fenn and G. Steidl. FMM and H-matrices: a short introduction to the basic idea. *Preprint Universität Mannheim*, 2002.

[FS02b] J. A. Fessler and B. P. Sutton. NUFFT - nonuniform FFT toolbox for Matlab. http://www.eecs.umich.edu/∼fessler/code, 2002.

[FS03] J. A. Fessler and B. P. Sutton. Nonuniform fast Fourier transforms using min-max interpolation. IEEE *Trans. Signal Process.*, 51:560 – 574, 2003.

[FS04] M. Fenn and G. Steidl. Fast NFFT based summation of radial functions. *Sampling Theory in Signal and Image Processing*, 3:1 – 28, 2004.

[GL04] L. Greengard and J.-Y. Lee. Accelerating the nonuniform fast Fourier transform. SIAM *Rev.*, 46:443 – 454, 2004.

[GM06] M. Ganesh and H. N. Mhaskar. Matrix-free interpolation on the sphere. SIAM *J. Numer. Anal.*, 44:1314 – 1331, 2006.

[GR87] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325 – 348, 1987.

[Gre88] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, 1988.

[Grö92] K. Gröchenig. Reconstruction algorithms in irregular sampling. *Math. Comput.*, 59:181 – 194, 1992.

[Grö93] K. Gröchenig. A discrete theory of irregular sampling. *Lin. Alg. Appl.*, 193:129 – 150, 1993.

[GS91] L. Greengard and J. Strain. The fast Gauss transform. SIAM *J. Sci. Stat. Comput.*, 12:79 – 94, 1991.

[GS98] L. Greengard and X. Sun. A new version of the fast Gauss transform. *Doc. Math. J. DMV*, 3:575 – 584, 1998.

[GS01] K. Gröchenig and T. Strohmer. Numerical and theoretical aspects of non-uniform sampling of band-limited images. In F. Marvasti, editor, *Nonuniform Sampling: Theory and Practice.* Kluwer/Plenum, 2001.

[gsl] GSL - The GNU Scientific Library. http://www.gnu.org/software/gsl.

[Hac99] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$–matrices, Part I: introduction to $\mathcal{H}$–matrices. *Computing*, 62:89 – 108, 1999.

[Hal92] K. Hallatschek. Fouriertransformation auf dünnen Gittern mit hierarchischen Basen. *Numer. Math.*, 63:83–97, 1992.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.

[HKMR03] D. Healy, P. Kostelec, S. Moore, and D. Rockmore. FFTs for the 2-sphere – Improvements and variations. *J. Fourier Anal. Appl.*, 9:341 – 385, 2003.

[HKS00] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In H.-J. Bungartz, R. H. W. Hoppe, and C. Zenger, editors, *Lectures on applied mathematics*, pages 9 – 29, Berlin, 2000. Springer-Verlag.

[HL05] J. A. Hogan and J. D. Lakey. *Time-Frequency and Time-Scale Methods: Wavelets, Sampling, Uncertainty Principles.* Applied and Numerical Harmonic Analysis series. Birkhauser, Boston, 2005.

[HLP89] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities.* Cambridge University Press, Cambridge, 1989.

[HN89] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463 – 491, 1989.

[JMNM91] J. I. Jackson, C. H. Meyer, D. G. Nishimura, and A. Macovski. Selection of a convolution function for Fourier inversion using gridding. IEEE *Trans. Med. Imag.*, 10:473 – 478, 1991.

[Kei05] J. Keiner. Fast Spherical Fourier Transforms and Applications. Diplomarbeit, Institut für Mathematik, Universität zu Lübeck, 2005.

[KKP05] T. Knopp, S. Kunis, and D. Potts. Fast iterative reconstruction for MRI from nonuniform k-space data. revised Preprint A-05-10, Universität zu Lübeck, 2005.

[KKP06] J. Keiner, S. Kunis, and D. Potts. Fast summation of Radial Functions on the Sphere. *Computing*, 78:1–15, 2006.

[Kla05] S. Klatt. Schnelle trigonometrische Transformationen an nichtäquidistanten Knoten. Studienarbeit, Institut für Mathematik, Universität zu Lübeck, 2005.

[KP03] S. Kunis and D. Potts. Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 161:75 – 98, 2003.

[KP04a] S. Kunis and D. Potts. NFFT2.0. Preprint B-04-07, Universität zu Lübeck, http://www.tu-chemnitz.de/∼potts/nfft, 2004.

[KP04b] S. Kunis and D. Potts. Stability results for scattered data interpolation by trigonometric polynomials. revised Preprint A-04-12, Universität zu Lübeck, 2004.

[KP06a] J. Keiner and D. Potts. Fast evaluation of quadrature formulae on the sphere. Preprint A-06-07, Universität zu Lübeck, 2006.

[KP06b] S. Kunis and D. Potts. NFFT, Softwarepackage, C subroutine library. http://www.tu-chemnitz.de/∼potts/nfft, 2002 – 2006.

[KP06c] S. Kunis and D. Potts. Time and memory requirements of the nonequispaced FFT. Preprint 06-01, TU-Chemnitz, 2006.

[KPS02] S. Kunis, D. Potts, and G. Steidl. Fast Fourier transforms for nonequispaced data – A user's guide to a C-library. Preprint B-02-13, Universität zu Lübeck, 2002.

[KPS06] S. Kunis, D. Potts, and G. Steidl. Fast Gauss transform with complex parameters using NFFTs. *J. Numer. Math.*, 14:295 – 303, 2006.

[KR06a] S. Kunis and H. Rauhut. OMP4NFFT, MATLAB-toolbox for orthogonal matching pursuit on sparse trigonometric polynomials. http://www.tu-chemnitz.de/∼skunis/software.php, 2006.

[KR06b] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials II - orthogonal matching pursuit versus basis pursuit. Preprint 06-06, TU-Chemnitz, 2006.

[LG05] J.-Y. Lee and L. Greengard. The type 3 nonuniform FFT and its applications. *J. Comput. Physics*, 206:1 – 5, 2005.

[Loa92] C. V. Loan. *Computational Frameworks for the Fast Fourier Transform.* SIAM, Philadelphia, 1992.

[MF06] J. Ma and M. Fenn. Combined complex ridgelet shrinkage and total variation minimization. SIAM *J. Sci. Comput.*, 28:984 – 1000, 2006.

[MFK04] S. Matej, J. A. Fessler, and I. G. Kazantsev. Iterative tomographic image reconstruction using Fourier-based forward and back- projectors. IEEE *Trans. Med. Imag.*, 23:401 – 412, 2004.

[MNW01] H. N. Mhaskar, F. J. Narcowich, and J. D. Ward. Spherical Marcinkiewicz-Zygmund inequalities and positive quadrature. *Math. Comput.*, 70:1113 – 1130, 2001.

[Moh99] M. J. Mohlenkamp. A fast transform for spherical harmonics. *J. Fourier Anal. Appl.*, 5:159 – 184, 1999.

[MP00] H. N. Mhaskar and J. Prestin. On the detection of singularities of a periodic function. *Adv. Comput. Math.*, 12:95 – 131, 2000.

[MR95] D. K. Maslen and D. N. Rockmore. Generalized FFTs - A Survey of Some Recent Results. In L. Finkelstein and W. Kantor, editors, *DIMACS Workshop in Groups and Computation*, volume 28, pages 183 – 238, 1995.

[NL99] N. Nguyen and Q. H. Liu. The regular Fourier matrices and nonuniform fast Fourier transforms. SIAM *J. Sci. Comput.*, 21:283 – 293, 1999.

[NS03] A. Nieslony and G. Steidl. Approximate factorizations of Fourier matrices with nonequispaced knots. *Linear Algebra Appl.*, 266:337 – 351, 2003.

[NSW98] F. J. Narcowich, N. Sivakumar, and J. D. Ward. Stability results for scattered-data interpolation on euclidean spheres. *Adv. Comput. Math.*, 8:137 – 163, 1998.

[O'S85] J. D. O'Sullivan. A fast sinc function gridding algorithm for Fourier inversion in computer tomography. IEEE *Trans. Med. Imag.*, 4:200 – 207, 1985.

[Pel97] J. Pelt. Fast computation of trigonometric sums with applications to frequency analysis of astronomical data. In D. Maoz, A. Sternberg, and E. Leibowitz, editors, *Astronomical Time Series*, pages 179 – 182, Kluwer, 1997.

[Pot98] D. Potts. Schnelle Polynomtransformation und Vorkonditionierer für Toeplitz–Matrizen. Dissertation, Universität Rostock, 1998.

[Pot03] D. Potts. Schnelle Fourier-Transformationen für nichtäquidistante Daten und Anwendungen. Habilitation, Universität zu Lübeck, 2003.

[PS82] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8:43 – 71, 1982.

[PS01a] D. Potts and G. Steidl. A new linogram algorithm for computerized tomography. IMA *J. Numer. Anal.*, 21:769 – 782, 2001.

[PS01b] J. Prestin and K. Selig. On a constructive representation of an orthogonal trigonometric schauder basis for $C_{2\pi}$. In *Operator Theory: Advances and Applications*, pages 402 – 425. Birkhäuser, 2001.

[PS02] D. Potts and G. Steidl. Fourier reconstruction of functions from their nonstandard sampled Radon transform. *J. Fourier Anal. Appl.*, 8:513 – 533, 2002.

[PS03] D. Potts and G. Steidl. Fast summation at nonequispaced knots by NFFTs. SIAM *J. Sci. Comput.*, 24:2013 – 2037, 2003.

[PSN04] D. Potts, G. Steidl, and A. Nieslony. Fast convolution with radial kernels at nonequispaced knots. *Numer. Math.*, 98:329 – 351, 2004.

[PST98a] D. Potts, G. Steidl, and M. Tasche. Fast algorithms for discrete polynomial transforms. *Math. Comput.*, 67:1577 – 1590, 1998.

[PST98b] D. Potts, G. Steidl, and M. Tasche. Fast and stable algorithms for discrete spherical Fourier transforms. *Linear Algebra Appl.*, 275/276:433 – 450, 1998.

[PST01] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, pages 247 – 270. Birkhäuser, Boston, 2001.

[RAG91] L. Reichel, G. S. Ammar, and W. B. Gragg. Discrete least squares approximation by trigonometric polynomials. *Math. Comput.*, 57:273 – 289, 1991.

[Rau06] H. Rauhut. Stability results for random sampling of sparse trigonometric polynomials. *Preprint*, 2006. submitted.

[RS98] M. Rauth and T. Strohmer. Smooth approximation of potential fields from noisy scattered data. *Geophysics*, 63:85 – 94, 1998.

[RT06] V. Rokhlin and M. Tygert. Fast Algorithms for Spherical Harmonic Expansions. SIAM *J. Sci. Comput.*, 27:1903 – 1928, 2006.

[SBC01] G. E. Sarty, R. Bennett, and R. W. Cox. Direct reconstruction of non-cartesian k-space data using a nonuniform fast Fourier transform. *Magn. Reson. Med.*, 45:908 – 915, 2001.

[Sch97] M. Schreiner. Locally supported kernels for spherical spline interpolation. *J. Approx. Theory*, 89:172 – 194, 1997.

[SFN01] B. P. Sutton, J. A. Fessler, and D. C. Noll. A min-max approach to the nonuniform N-dimensional FFT for rapid iterative reconstruction of MR images. In *Proc. ISMRM 9th Scientific Meeting*, page 763, 2001.

[SNF03] B. P. Sutton, D. C. Noll, and J. A. Fessler. Fast, iterative, field-corrected image reconstruction for MRI in the presence of field inhomogeneities. IEEE *Trans. Med. Imag.*, 22:178 – 188, 2003.

[SP01] X. Sun and N. P. Pitsianis. A matrix version of the fast multipole method. SIAM *Rev.*, 43:289 – 300, 2001.

[Spr97] F. Sprengel. Interpolation und Waveletzerlegung multivariater periodischer Funktionen. Dissertation, Universität Rostock, 1997.

[Spr00] F. Sprengel. A class of function spaces and interpolation on sparse grids. *Numer. Funct. Anal. Optim.*, 21:273 – 293, 2000.

[ST95] H. Schomberg and J. Timmer. The gridding method for image reconstruction by Fourier transformation. IEEE *Trans. Med. Imag.*, MI–14:596 – 607, 1995.

[ST02] R. Suda and M. Takami. A fast spherical harmonics transform algorithm. *Math. Comput.*, 71:703 – 715, 2002.

[Ste98] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. *Adv. Comput. Math.*, 9:337 – 353, 1998.

[Str91] J. Strain. The fast Gauss transform with variable scales. SIAM *J. Sci. Stat. Comput.*, 12:1131 – 1139, 1991.

[Sze75] G. Szegö. *Orthogonal Polynomials.* Amer. Math. Soc., Providence, RI, 4th edition, 1975.

[Tyr96] E. E. Tyrtyshnikov. Mosaic–skeleton approximations. *Calcolo*, 33:47 – 57, 1996.

[Vol06] A. Vollrath. Fast Fourier transforms on the rotation group and applications. Diplomarbeit, Institut für Mathematik, Universität zu Lübeck, 2006.

[War98] A. F. Ware. Fast approximate Fourier transforms for irregularly spaced data. SIAM *Rev.*, 40:838 – 856, 1998.

[Wen05] H. Wendland. *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, 2005.

[zCF05] W. zu Castell and F. Filbir. Radial basis functions and corresponding zonal series expansions on the sphere. *J. Approx. Theory*, 134:65 – 79, 2005.

[Zen91] C. Zenger. Sparse grids. In *Parallel algorithms for partial differential equations (Kiel, 1990)*, volume 31 of *Notes Numer. Fluid Mech.*, pages 241–251. Vieweg, Braunschweig, 1991.

[Zyg93] A. Zygmund. *Trigonometric Series.* Cambridge University press, 1993.

Curriculum vitae

Personal information

Name	Stefan Kunis
Date of Birth	12 August 1978
Place of Birth	Oelsnitz, Vogtland
Nationality	German
Adress	Erich-Mühsam-Straße 26 09112 Chemnitz
E-mail	kunis@mathematik.tu-chemnitz.de

Education

since 04/2006	Research assistant, Faculty of Mathematics, Chemnitz University of Technology
2003 – 2006	Doctoral studies, Thesis defense 23/08/2006, Summa cum laude Advisor Prof. Daniel Potts and Prof. Jürgen Prestin
05/05–03/06	Fellowship of the German Academic Exchange Service, Numerical Harmonic Analysis Group, Faculty of Mathematics, University Vienna, Austria
09/03–04/05	Fellowship of the Graduate School 357: Efficient Algorithms and Multiscale Methods, Institute of Mathematics, University of Lübeck
1998–2003	Studies of computer science, Diplom 29/08/2003, Grade 1.0 Minor in medical informatics, University of Lübeck
03/03–08/03	Diploma thesis: Iterative Fourier reconstruction
01/02–06/02	Erasmus-Fellowship, University Linköping, Sweden
1997–1998	Military service
1985–1997	School, Abitur 11/07/1997, Grade 1.3