

DFG-Schwerpunktprogramm 1324

„Extraktion quantifizierbarer Information aus komplexen Systemen“

Nonequispaced Hyperbolic Cross Fast Fourier Transform

M. Döhler, S. Kunis, D. Potts

Preprint 12



Edited by

AG Numerik/Optimierung
Fachbereich 12 - Mathematik und Informatik
Philipps-Universität Marburg
Hans-Meerwein-Str.
35032 Marburg

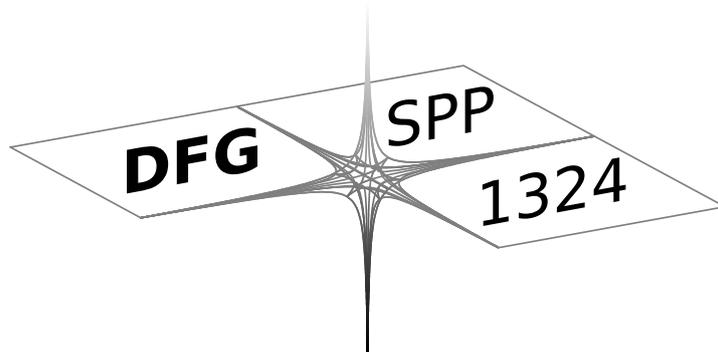
DFG-Schwerpunktprogramm 1324

„Extraktion quantifizierbarer Information aus komplexen Systemen“

Nonequispaced Hyperbolic Cross Fast Fourier Transform

M. Döhler, S. Kunis, D. Potts

Preprint 12



The consecutive numbering of the publications is determined by their chronological order.

The aim of this preprint series is to make new research rapidly available for scientific discussion. Therefore, the responsibility for the contents is solely due to the authors. The publications will be distributed by the authors.

Nonequispaced hyperbolic cross fast Fourier transform

Michael Döhler Stefan Kunis Daniel Potts

A straightforward discretisation of problems in d spatial dimensions often leads to an exponential growth in the number of degrees of freedom. Thus, even efficient algorithms like the fast Fourier transform (FFT) have high computational costs. Hyperbolic cross approximations allow for a severe decrease in the number of used Fourier coefficients to represent functions with bounded mixed derivatives. We propose a nonequispaced hyperbolic cross fast Fourier transform relying on one hyperbolic cross FFT and a dedicated interpolation by splines on sparse grids.

Key words and phrases : trigonometric approximation, hyperbolic cross, sparse grid, fast Fourier transform, nonequispaced FFT

2000 AMS Mathematics Subject Classification : 65F10, 65F15, 65T40

1 Introduction

The discretisation of high dimensional problems often leads to an exponential growth in the number of degrees of freedom. Even efficient algorithms like the d -dimensional fast Fourier transform (FFT) that computes

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \hat{G}_n} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k} \mathbf{x}}, \quad \mathbf{x} \in G_n \quad (1.1)$$

with equispaced nodes in frequency domain $\hat{G}_n = \times_{l=1}^d \hat{G}_{n_l}$, $\hat{G}_n = \mathbb{Z} \cap (-2^{n-1}, 2^{n-1}]$ and space domain $G_n = \times_{l=1}^d G_{n_l}$, $G_n = 2^{-n}(\mathbb{Z} \cap [0, 2^n))$ need $\mathcal{O}(|G_n| \log |G_n|)$ arithmetic operations, i.e., $\mathcal{O}(2^{nd}nd)$ if $n_l = n$, $l = 1, \dots, d$. This is labelled as the curse of dimensions and the use of sparsity has become a very popular tool for handling such problems. For moderately high dimensional problems the use of sparse grids and the approximation on hyperbolic crosses has decreased the problem size dramatically from $\mathcal{O}(2^{nd})$ to $\mathcal{O}(2^n n^{d-1})$ while hardly deteriorating the approximation error, see e.g. [22, 23, 20, 6, 19]. Of course, an import issue is the adaption of efficient algorithms to these thinner discretisations such that their total complexity is within logarithmic factors still linear in the decreased problem size. The FFT has been refined to the hyperbolic cross fast Fourier transform (HCFFFT) in [2, 14, 12].

Chemnitz University of Technology, Department of Mathematics, 09107 Chemnitz, Germany,
Michael.Doehler@s2003.tu-chemnitz.de, {kunis,potts}@mathematik.tu-chemnitz.de

On the other hand, the FFT has been generalised to the nonequispaced fast Fourier transform (NFFT), cf. [10, 4, 21, 18, 13, 15], which requires $\mathcal{O}(|G_n| \log |G_n| + M)$ arithmetic operations for the approximate evaluation of the trigonometric polynomial (1.1) at M arbitrary nodes $\mathbf{x} \in \mathbb{R}^d$. FFT algorithms are usually exact algorithms with small errors in the result caused only by limitations of floating point arithmetic. The NFFT algorithm purposely introduces a systematic error in the computation to achieve its favourable arithmetic cost. This additional error can be controlled, and, if deemed necessary, can be reduced to the order of machine precision. To this end, the NFFT uses an *oversampled* FFT internally and a local approximation. This introduces two additional parameters, an *oversampling factor* and a *truncation parameter*, that control the accuracy of the NFFT.

In this paper, we generalise the HCFFT [2, 14, 12] for arbitrary sampling nodes. Earlier work in [11] was based on a partition of the hyperbolic cross and several NFFTs. The complicated partition of the hyperbolic cross makes an implementation of this algorithm hard already for the three dimensional case. Moreover, each NFFT uses one oversampled FFT and a local approximation scheme which have to be glued together in a final step. In contrast, our new evaluation scheme is based on one oversampled HCFFT, which is implemented easily for arbitrary spatial dimension by using the unidirectional scheme [12], and a local approximation by interpolating splines on the sparse grid [5], which yields a global approximant to the given trigonometric polynomial.

The paper is organised as follows: After introducing the necessary notation, we discuss the periodic spline interpolation for the univariate case and its use for interpolation on sparse grids. Section 4 introduces our novel nonequispaced hyperbolic cross fast Fourier transform (NHCFFT), estimates its approximation error in Theorem 4.1, and gives a complexity analysis with respect to the problem size, the target accuracy, and fixed spatial dimension. Finally, we present our numerical experiments for the hyperbolic cross FFT and its nonequispaced version and conclude our findings.

2 Prerequisite

Let a spatial dimension $d \in \mathbb{N}$ and a refinement $n \in \mathbb{N}_0$ be given. We denote by $\mathbb{T}^d \cong [0, 1)^d$ the d -dimensional torus, consider Fourier series $f : \mathbb{T}^d \rightarrow \mathbb{C}$, $f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k} \mathbf{x}}$, and restrict the frequency domain to the hyperbolic cross

$$H_n^d := \bigcup_{\substack{\mathbf{k} \in \mathbb{N}_0^d \\ \|\mathbf{k}\|_1 = n}} \hat{G}_{\mathbf{k}} = \{\mathbf{q} \in \hat{G}_{\mathbf{k}} : \|\mathbf{k}\|_1 = |k_1| + |k_2| + \dots + |k_d| = n\}, \quad (2.1)$$

see Figure 2.1(a). Our aim is the fast approximate evaluation of d -variate trigonometric polynomials $f \in \Pi_n^{\text{hc}}(\mathbb{T}^d)$, i.e.,

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in H_n^d} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k} \mathbf{x}} \quad (2.2)$$

at nodes $\mathbf{x}_\ell \in \mathbb{T}^d$, $\ell = 0, \dots, M-1$. For the moment, we restrict ourselves to sparse grids

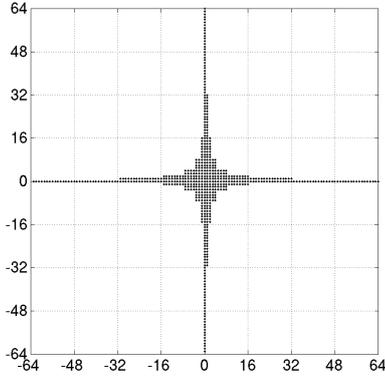
$$S_n^d := \bigcup_{\substack{\mathbf{k} \in \mathbb{N}_0^d \\ \|\mathbf{k}\|_1 = n}} G_{\mathbf{k}} = \{\mathbf{x} \in G_{\mathbf{k}} : \|\mathbf{k}\|_1 = n\} \subset \mathbb{T}^d, \quad (2.3)$$

see Figure 2.1(b), as sampling sets and obtain the following well known results, cf. [14]. For fixed dimension $d \in \mathbb{N}$ and arbitrary refinement $n \in \mathbb{N}_0$, we have the partition

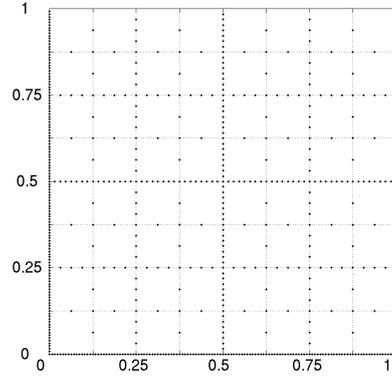
$$H_n^d = \bigcup_{s=0}^n H_{n-s}^{d-1} \times (\hat{G}_s \setminus \hat{G}_{s-1}), \quad \hat{G}_{-1} := \emptyset, \quad (2.4)$$

and the cardinality estimates

$$|S_n^d| = |H_n^d| = \sum_{j=0}^{\min(n, d-1)} 2^{n-j} \binom{n}{j} \binom{d-1}{j} = \frac{2^n n^{d-1}}{2^{d-1} (d-1)!} + \mathcal{O}(2^n n^{d-2}). \quad (2.5)$$



(a) Hyperbolic cross $H_7^2 \subset \mathbb{Z}^2$.



(b) Sparse grid $S_7^2 \subset \mathbb{T}^2$.

Figure 2.1: Two dimensional hyperbolic cross and corresponding sparse grid.

Moreover, we denote the efficient evaluation of (2.2) at the sparse grid nodes $\mathbf{x}_\ell \in S_n^d$, $\ell = 0, \dots, |S_n^d| - 1$, by hyperbolic cross fast Fourier transform (HCFFT) and the reconstruction of Fourier coefficients on the hyperbolic cross from samples at the sparse grid nodes by inverse HCFFT. Both transforms can be computed in $\mathcal{O}(2^n n^d)$ floating point operations, cf. [2, 14]. In contrast, a naive evaluation of (2.2) at the sparse grid nodes (hyperbolic cross discrete Fourier transform, HCDFT) or equivalently a matrix vector multiplication with the hyperbolic cross Fourier matrix

$$\mathbf{F}_n^d := \left(e^{2\pi i \mathbf{k} \mathbf{x}} \right)_{\mathbf{x} \in S_n^d, \mathbf{k} \in H_n^d} \quad (2.6)$$

takes $\mathcal{O}(2^{2n} n^{2d-2})$ floating point operations. Analogously, the naive evaluation of (2.2), i.e., the matrix vector multiplication with the nonequispaced hyperbolic cross Fourier matrix

$$\mathbf{A}_n^d := \left(e^{2\pi i \mathbf{k} \mathbf{x}} \right)_{\mathbf{x} \in \mathcal{X}, \mathbf{k} \in H_n^d}, \quad \mathcal{X} := \{\mathbf{x}_\ell \in \mathbb{T}^d : \ell = 0, \dots, M-1\}, \quad (2.7)$$

takes $\mathcal{O}(2^n M)$ floating point operations.

3 Periodic splines

3.1 Univariate interpolation

Let the cardinal B-spline $N_m : \mathbb{R} \rightarrow \mathbb{R}$,

$$N_{m+1} := N_{m-1} * N_1 = \int_0^1 N_{m-1}(\cdot - t) dt, \quad N_1 := \chi_{[0,1)}, \quad (3.1)$$

of order $m \in \mathbb{N}$ be given. Subsequently, we always assume a B-spline of even order $m \in 2\mathbb{N}$.

For a given spline refinement $r \in \mathbb{N}_0$, we define the periodic spline $\phi_r : \mathbb{T} \rightarrow \mathbb{C}$, its translates $\phi_{r,k} : \mathbb{T} \rightarrow \mathbb{C}$,

$$\phi_r := \sum_{j \in \mathbb{Z}} N_m(2^r(\cdot + j)), \quad \phi_{r,k} := \phi_r \left(\cdot - \frac{k}{2^r} \right), \quad k = 0, \dots, 2^r - 1. \quad (3.2)$$

and the corresponding spaces

$$V_r := \text{span}\{\phi_{r,k} : k = 0, \dots, 2^r - 1\}. \quad (3.3)$$

For $f \in C(\mathbb{T})$, let the interpolation operator $\mathcal{L}_r^{(1)} = \mathcal{L}_r : C(\mathbb{T}) \rightarrow V_r$ be uniquely defined, see [17] for details, by

$$\mathcal{L}_r f(x) = f(x), \quad x \in G_r.$$

Then the spline coefficients $a_{r,k} \in \mathbb{C}$, $k = 0, \dots, 2^r - 1$, in the representation

$$\mathcal{L}_r f = \sum_{k=0}^{2^r-1} a_{r,k} \phi_{r,k}$$

can be computed by $\mathcal{O}(2^r m)$ floating point operations, cf. [3, 5].

Concerning the interpolation error, the relevant results from [16] state the following.

Lemma 3.1. *Let $r \in \mathbb{N}_0$, $m \in 2\mathbb{N}$, and the kernel $K_r : [0, 1]^2 \rightarrow \mathbb{R}$,*

$$K_r(x, y) = b_m(x, y) - \sum_{k=0}^{2^r-1} b_m \left(\frac{k}{2^r}, y \right) L_{r,k}(x)$$

built upon the Lagrange polynomials and the modified Bernoulli splines

$$L_{r,k}(x) = \prod_{l=0, l \neq k}^{2^r-1} \frac{2^r x - l}{k - l}, \quad k = 0, \dots, 2^r - 1,$$

$$b_m(x) = \sum_{k \in \mathbb{Z} \setminus \{0\}} (2\pi i k)^{-m} e^{2\pi i k x}, \quad b_m(x, y) = b_m(x) - b_m(x - y)$$

be given. Moreover, let $f : [0, 1] \rightarrow \mathbb{C}$ be m -times continuously differentiable. Then, the interpolation error allows for the representation

$$(\mathcal{I} - \mathcal{L}_r)f(x) = \int_0^1 K_r(x, y) f^{(m)}(y) dy.$$

Moreover, the error is bounded since

$$\sup_{x \in [0,1]} \|K_r(x, \cdot)\|_1 = \sup_{x \in [0,1]} \left\{ \int_0^1 |K_r(x, y)| dy \right\} = \frac{F_m}{(2^r \pi)^m},$$

where $\frac{\pi^2}{8} \leq F_m = \frac{4}{\pi} \sum_{s=0}^{\infty} (-1)^s (2s+1)^{-m-1} < \frac{4}{\pi}$ denotes the Favard constant.

3.2 Multivariate interpolation

For spatial dimension $d \in \mathbb{N}$, we define the spline interpolation operator $\mathcal{L}_r^{(d)} : C(\mathbb{T}^d) \rightarrow V_r^{(d)}$,

$$\mathcal{L}_r^{(d)} := \bigoplus_{\substack{\mathbf{j} \in \mathbb{N}_0^d \\ \|\mathbf{j}\|_1 = r}} \mathcal{L}_{j_1} \otimes \dots \otimes \mathcal{L}_{j_d},$$

with target space $V_r^{(d)} := \text{Im } \mathcal{L}_r^{(d)}$. The main difficulty in this Boolean sum approach is the structure of the basis functions in $V_r^{(d)}$. We consider the following generating system as already suggested in [5] for the bivariate case. Let

$$\phi_{\mathbf{j}, \mathbf{k}}^{(d)} = \bigotimes_{l=1}^d \phi_{j_l, k_l}, \quad \mathbf{j}, \mathbf{k} \in \mathbb{N}_0^d, \mathbf{k} < 2^{\mathbf{j}}, \text{ i.e., } k_l = 0, \dots, 2^{j_l} - 1, l = 1, \dots, d,$$

and $\tilde{V}_r^{(d)} := \text{span}\{\phi_{\mathbf{j}, \mathbf{k}}^{(d)} : \mathbf{j}, \mathbf{k} \in \mathbb{N}_0^d, \|\mathbf{j}\|_1 = r, \mathbf{k} < 2^{\mathbf{j}}\}$.

Lemma 3.2. *Let a spatial dimension $d \in \mathbb{N}$ and a spline refinement $r \in \mathbb{N}_0$ be given, then*

i) *functions $f \in C(\mathbb{T}^d)$ are interpolated on the sparse grid, i.e.,*

$$\mathcal{L}_r^{(d)} f(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in S_r^{(d)},$$

ii) *moreover, the Boolean sum can be expressed as*

$$\mathcal{L}_r^{(d)} = \sum_{j=0}^r \mathcal{L}_j^{(d-1)} \otimes \mathcal{L}_{r-j} - \sum_{j=0}^{r-1} \mathcal{L}_j^{(d-1)} \otimes \mathcal{L}_{r-j-1}, \quad (3.4)$$

iii) *and finally, we have*

$$V_r^{(d)} \subset \tilde{V}_r^{(d)}, \quad \dim V_r^{(d)} = |S_r^{(d)}|, \quad |\tilde{V}_r^{(d)}| = 2^r \binom{r-1+d}{d-1} = \mathcal{O}(2^r r^{d-1}),$$

where the last expression holds true for fixed $d \in \mathbb{N}$.

Proof. All results easily follow from [7, Sec. 1-2]. Due to its definition, the operator $\mathcal{L}_r^{(d)}$ interpolates on all grids G_j , $\|\mathbf{j}\|_1 = r$, and thus on S_r^d , cf. [7, Sec. 2.3, Prop. 2]. The second assertion is due to [7, Sec. 1.3, Prop. 2]. Moreover, the target spaces allow for the recursion

$$V_r^{(d)} = \text{Im } \mathcal{L}_r^{(d)} = \sum_{j=0}^r \text{Im } \mathcal{L}_j^{(d-1)} \otimes \text{Im } \mathcal{L}_{r-j} = \sum_{j=0}^r V_j^{(d-1)} \otimes V_{r-j}$$

and thus $\tilde{V}_r^{(d)}$ yields a generating system for $V_r^{(d)}$. Finally, [7, Sec. 2.3, Prop. 4] yields the dimension of $V_r^{(d)}$ and since the number of multi-indices \mathbf{j} with $\|\mathbf{j}\|_1 = r$ is $\binom{r+d}{d} - \binom{r-1+d}{d} = \binom{r-1+d}{d-1}$, the last assertion follows. ■

We represent the interpolating function $\mathcal{L}_r^{(d)} f$ by an expansion in $\tilde{V}_r^{(d)}$, i.e.,

$$\mathcal{L}_r^{(d)} f = \sum_{\substack{\mathbf{j} \in \mathbb{N}_0^d \\ \|\mathbf{j}\|_1 = r}} \sum_{\substack{\mathbf{k} \in \mathbb{N}_0^d \\ \mathbf{k} < 2^{\mathbf{j}}}} a_{\mathbf{j}, \mathbf{k}} \phi_{\mathbf{j}, \mathbf{k}}^{(d)}. \quad (3.5)$$

For $d \geq 2$, a simple reorganisation of the above sum yields for $\mathbf{x} \in \mathbb{T}^{d-1}, y \in \mathbb{T}$

$$\mathcal{L}_r^{(d)} f(\mathbf{x}, y) = \sum_{l=0}^r \sum_{r=0}^{2^l-1} \left(\sum_{\substack{\mathbf{j} \in \mathbb{N}_0^{d-1} \\ \|\mathbf{j}\|_1 = r-l}} \sum_{\substack{\mathbf{k} \in \mathbb{N}_0^{d-1} \\ \mathbf{k} < 2^{\mathbf{j}}}} a_{(\mathbf{j}, l), (\mathbf{k}, r)} \phi_{\mathbf{j}, \mathbf{k}}^{(d-1)}(\mathbf{x}) \right) \phi_{l, r}(y).$$

Now let l, r be fixed and solve the $d-1$ dimensional interpolation problem in the brackets. If we assume that this can be done in $Cm(r-l)^{d-1}2^{r-l}$ arithmetic operations, an induction argument yields a total complexity of $\mathcal{O}(2^r r^{d-1} m)$ for computing the spline coefficients $a_{\mathbf{j}, \mathbf{k}}$, $\mathbf{j}, \mathbf{k} \in \mathbb{N}_0^d$, $\|\mathbf{j}\|_1 = r$, $\mathbf{k} < 2^{\mathbf{j}}$, from the samples $f(\mathbf{x})$, $\mathbf{x} \in S_r^d$. Details of the two-dimensional interpolation algorithm are given in [5].

3.3 Evaluation at arbitrary nodes

Finally, we compute function values of $\mathcal{L}_r^{(d)} f$ at arbitrary sampling nodes $\mathbf{x} \in \mathbb{T}^d$. For fixed $x \in \mathbb{T}$, $j \in \mathbb{N}_0$, and $k = 0, \dots, 2^j - 1$, we have

$$\phi_{j, k}(x) \neq 0 \quad \text{if and only if} \quad k \in \{ \lfloor 2^j x \rfloor - m + 1, \dots, \lfloor 2^j x \rfloor \}.$$

Thus, the inner sum in (3.5) contains only m^d non-zero summands. In conjunction with the estimate on the number of multi-indices $\mathbf{j} \in \mathbb{N}_0^d$, $\|\mathbf{j}\|_1 = r$, from the proof of Lemma 3.2, and assuming that a single B-spline can be evaluated with constant effort, we obtain a total number of arithmetic operations $\mathcal{O}(r^{d-1} m^d)$ for evaluation of one function value $\mathcal{L}_r^{(d)} f(\mathbf{x})$.

4 The nonequispaced hyperbolic cross fast Fourier transform

We combine the hyperbolic cross FFT [2, 14, 12] and the spline approximation scheme [5] generalised to spatial dimension $d \in \mathbb{N}$ in the following algorithm.

4.1 Error estimates

Algorithm 4.1 introduces an error when approximating the trigonometric polynomial by the spline. Subsequently, we show that for a moderate oversampling exponent $\alpha \in \mathbb{N}$, $\alpha \geq d$, the error decays exponentially with the spline order $m \in 2\mathbb{N}$. Theorem 4.1 gives a slightly more general statement for arbitrary spline refinement which is then proven by induction.

Theorem 4.1. *Let a spatial dimension $d \in \mathbb{N}$, a spline refinement $r \in \mathbb{N}_0$, a spline order $m \in 2\mathbb{N}$, a refinement $n \in \mathbb{N}_0$, and a trigonometric polynomial $f \in \Pi_n^{\text{hc}}(\mathbb{T}^d)$ be given. Then the interpolation error can be bounded by*

$$\|(\mathcal{I} - \mathcal{L}_r^{(d)})f\|_\infty \leq \frac{(2r+2)^{d-1} F_m^d 2^{nm}}{2^{(r-d+1)m}} \sum_{\mathbf{k} \in H_n^d} |\hat{f}_{\mathbf{k}}|. \quad (4.1)$$

Algorithm 4.1 Nonequispaced hyperbolic cross FFT (NHCFFT)

Input:	$d \in \mathbb{N}$ $n \in \mathbb{N}_0$ $\hat{f}_{\mathbf{k}}, \mathbf{k} \in H_n^d$ $\mathbf{x}_\ell \in \mathbb{T}^d, \ell = 0, \dots, M-1$ $m \in 2\mathbb{N}$ $\alpha \in \mathbb{N}, \alpha \geq d$	Spatial dimension Refinement of H_n^d Fourier coefficients Sampling nodes Spline order Oversampling exponent
--------	---	---

1: Compute samples on the oversampled sparse grid by the HCFFT, i.e., evaluate

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in H_n^d} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k} \mathbf{x}}, \quad \mathbf{x} \in \mathbb{S}_{n+\alpha}^d.$$

2: Interpolate on the oversampled sparse grid by $\mathcal{L}_{n+\alpha}^{(d)}$, see (3.5), i.e., compute

$$a_{\mathbf{j}, \mathbf{k}} \in \mathbb{C}, \mathbf{j}, \mathbf{k} \in \mathbb{N}_0^d, \|\mathbf{j}\|_1 = n + \alpha, \mathbf{k} < 2^{\mathbf{j}}, \quad \text{from } f(\mathbf{x}), \mathbf{x} \in \mathbb{S}_{n+\alpha}^d.$$

3: For $\ell = 0, \dots, M-1$, evaluate the spline

$$\mathcal{L}_{n+\alpha}^{(d)} f(\mathbf{x}_\ell) = \sum_{\substack{\mathbf{j} \in \mathbb{N}_0^d \\ \|\mathbf{j}\|_1 = n+\alpha}} \sum_{\substack{\mathbf{k} \in \mathbb{N}_0^d \\ \mathbf{k} < 2^{\mathbf{j}}}} a_{\mathbf{j}, \mathbf{k}} \phi_{\mathbf{j}, \mathbf{k}}^{(d)}(\mathbf{x}_\ell).$$

Output:	$\mathcal{L}_{n+\alpha}^{(d)}(\mathbf{x}_\ell), \ell = 0, \dots, M-1$	Sample values
---------	---	---------------

Proof. We prove the assertion by induction over $d \in \mathbb{N}$. Due to Lemma 3.1 and by restricting to the space of trigonometric polynomials $\Pi_n^{\text{hc}}(\mathbb{T})$, the univariate complement operator $\mathcal{K}_r^{(1)} = \mathcal{I} - \mathcal{L}_r^{(1)}$ allows for the error estimate

$$|\mathcal{K}_r^{(1)} f(x)| \leq \|K_r(x, \cdot)\|_1 \|f^{(m)}\|_\infty \leq \frac{F_m 2^{nm}}{2^{rm}} \sum_{k \in H_n^1} |\hat{f}_k|$$

and shows (4.1) for $d = 1$. Using $d \geq 2$ and (3.4), we generalise to the d -variate complement operator $\mathcal{K}_r^{(d)} = \mathcal{I} - \mathcal{L}_r^{(d)}$ by

$$\mathcal{K}_r^{(d)} = \mathcal{K}_r^{(d-1)} \otimes \mathcal{I} + \mathcal{I} \otimes \mathcal{K}_r^{(1)} + \sum_{j=0}^{r-1} \mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j-1}^{(1)} - \sum_{j=0}^r \mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j}^{(1)} \quad (4.2)$$

and consider the summands individually. For $\mathbf{x} \in \mathbb{T}^{d-1}$, $y \in \mathbb{T}$, $f : \mathbb{T}^d \rightarrow \mathbb{C}$, and $g_{\mathbf{x}}(y) = (\mathcal{K}_j^{(d-1)} \otimes \mathcal{I})f(\mathbf{x}, y)$, we have

$$\mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j-1}^{(1)} f(\mathbf{x}, y) = (\mathcal{K}_j^{(d-1)} \otimes \mathcal{I})(\mathcal{I} \otimes \mathcal{K}_{r-j-1}^{(1)})f(\mathbf{x}, y) = \mathcal{K}_{r-j-1}^{(1)} g_{\mathbf{x}}(y)$$

and use Lemma 3.1 to estimate

$$|\mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j-1}^{(1)} f(\mathbf{x}, y)| \leq \frac{F_m}{2^{(r-j-1)m} \pi^m} \max_{z \in \mathbb{T}} |\mathcal{K}_j^{(d-1)} \otimes \frac{d^m}{dz^m} f(\mathbf{x}, z)|. \quad (4.3)$$

Restricting again to $f \in \Pi_n^{\text{hc}}(\mathbb{T}^d)$ and using the partition (2.4) we proceed by

$$\begin{aligned} \mathcal{K}_j^{(d-1)} \otimes \frac{d^m}{dz^m} f(\mathbf{x}, z) &= \mathcal{K}_j^{(d-1)} \otimes \frac{d^m}{dz^m} \left(\sum_{(\mathbf{k}, l) \in H_n^d} \hat{f}_{\mathbf{k}, l} e^{2\pi i \mathbf{k} \mathbf{x}} e^{2\pi i l z} \right) \\ &= \sum_{s=0}^n \sum_{l \in \hat{G}_s \setminus \hat{G}_{s-1}} (2\pi i l)^m e^{2\pi i l z} \cdot \mathcal{K}_j^{(d-1)} \left(\sum_{\mathbf{k} \in H_{n-s}^{d-1}} \hat{f}_{\mathbf{k}, l} e^{2\pi i \mathbf{k} \mathbf{x}} \right). \end{aligned}$$

Due to the induction hypothesis for $\mathcal{K}_j^{(d-1)}$ and since $|l| \leq 2^{s-1}$, we estimate for $j = 0, \dots, r-1$

$$\begin{aligned} |\mathcal{K}_j^{(d-1)} \otimes \frac{d^m}{dz^m} f(\mathbf{x}, z)| &\leq \sum_{s=0}^n \sum_{l \in \hat{G}_s \setminus \hat{G}_{s-1}} (2\pi |l|)^m \frac{(2j+2)^{d-2} F_m^{d-1} 2^{(n-s)m}}{2^{(j-d+2)m}} \sum_{\mathbf{k} \in H_{n-s}^{d-1}} |\hat{f}_{\mathbf{k}, l}| \\ &\leq \frac{(2j+2)^{d-2} F_m^{d-1} 2^{nm} \pi^m}{2^{(j-d+2)m}} \sum_{(\mathbf{k}, l) \in H_n^d} |\hat{f}_{\mathbf{k}, l}|. \end{aligned}$$

In conjunction with (4.3) this yields

$$|\mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j-1}^{(1)} f(\mathbf{x}, y)| \leq \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+1)m}} \sum_{(\mathbf{k}, l) \in H_n^d} |\hat{f}_{\mathbf{k}, l}|, \quad j = 0, \dots, r-1.$$

Analogously, all summands in (4.2) can be bounded - in particular

$$|\mathcal{K}_j^{(d-1)} \otimes \mathcal{K}_{r-j}^{(1)} f(\mathbf{x}, y)| \leq \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+2)m}} \sum_{(\mathbf{k}, l) \in H_n^d} |\hat{f}_{\mathbf{k}, l}|, \quad j = 0, \dots, r.$$

We finally use $\sum_{j=0}^{r-1} (2j+2)^{d-2} \leq \int_0^r (2j+2)^{d-2} dj = \frac{(2r+2)^{d-1}}{2(d-1)}$ to estimate

$$\begin{aligned} &\|\mathcal{K}_j^{(d)} f\|_\infty / \sum_{(\mathbf{k}, l) \in H_n^d} |\hat{f}_{\mathbf{k}, l}| \\ &\leq \frac{(2r+2)^{d-2} F_m^{d-1} 2^{nm}}{2^{(r-d+2)m}} + \frac{F_m 2^{nm}}{2^{rm}} + \sum_{j=0}^{r-1} \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+1)m}} + \sum_{j=0}^r \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+2)m}} \\ &\leq \frac{(2r+2)^{d-1} F_m^d 2^{nm}}{2^{(r-d+1)m}} \\ &\quad \left(\frac{1}{(2r+2)F_m 2^m} + \frac{1}{(2r+2)^{d-1} F_m^{d-1} 2^{(d-1)m}} + \frac{1}{2(d-1)} + \frac{1}{2^m 2(d-1)} + \frac{1}{2^m (2r+2)} \right). \end{aligned}$$

Bounding the term in parenthesis from above by one ($d \geq 2$, $r \geq 0$, $m \geq 2$, and $F_m \geq 1$) yields the assertion. \blacksquare

Remark 4.2. For the univariate case $d = 1$, an oversampling factor $\sigma > 1$, and a periodic spline interpolation of order $m \in 2\mathbb{N}$ at $\sigma 2^n$ nodes (formally let $r = \log_2(\sigma 2^n)$), the error estimate (4.1) reads as

$$\|(\mathcal{I} - \mathcal{L}_r)f\|_\infty / \sum_{k \in \hat{G}_n} |\hat{f}_k| \leq F_m \sigma^{-m},$$

which can be improved to $8(2\sigma - 1)^{-m}$, cf. [21, Cor. 4.3]. While Algorithm 4.1 computes the spline coefficients from samples in its second step, dividing the given Fourier coefficients by the discrete Fourier coefficients of the used B-spline leads to the same spline coefficients in [21, Algorithm 2.1 with (4.4)].

Moreover, the error estimate (4.1) can be refined to

$$\|(\mathcal{I} - \mathcal{L}_r^{(2)})f\|_\infty \leq \left(\frac{r}{2^{(r-1)m}} + \frac{r+3}{2^{rm}} \right) F_m^2 2^{nm} \sum_{\mathbf{k} \in H_n^2} |\hat{f}_{\mathbf{k}}|$$

for $d = 2$ and to

$$\|(\mathcal{I} - \mathcal{L}_r^{(3)})f\|_\infty \leq \frac{1}{2} \left(\frac{(r-1)r}{2^{(r-2)m}} + \frac{2r(r+4)}{2^{(r-1)m}} + \frac{(r+2)(r+6)}{2^{rm}} \right) F_m^3 2^{nm} \sum_{\mathbf{k} \in H_n^3} |\hat{f}_{\mathbf{k}}|$$

for $d = 3$, see [8, Theorems 3.19 and 3.24]. □

Corollary 4.3. Let a spatial dimension $d \in \mathbb{N}$, a refinement $n \in \mathbb{N}$ with $n \geq d$, and a target accuracy $\varepsilon > 0$ be given and choose the oversampling exponent $\alpha = d$ in Algorithm 4.1. Then for spline orders $m \in 2\mathbb{N}$ with

$$m \geq |\log_2 \varepsilon| + d \log_2 n + 3d \tag{4.4}$$

the complex exponentials are approximated such that

$$|e^{2\pi i \mathbf{k} \mathbf{x}} - \mathcal{L}_{n+d}^{(d)} e^{2\pi i \mathbf{k} \mathbf{x}}| \leq \varepsilon, \quad \mathbf{k} \in H_n^d, \mathbf{x} \in \mathbb{T}^d.$$

Proof. Setting $f(\mathbf{x}) = e^{2\pi i \mathbf{k} \mathbf{x}}$ in Theorem 4.1 yields

$$|e^{2\pi i \mathbf{k} \mathbf{x}} - \mathcal{L}_{n+d}^{(d)} e^{2\pi i \mathbf{k} \mathbf{x}}| \leq \frac{(4n+2)^{d-1} 4^d}{\pi^d 2^m} \leq \varepsilon \quad \Leftrightarrow \quad m \geq \log_2 \frac{(4n+2)^{d-1} 4^d}{\pi^d \varepsilon}$$

for which (4.4) is sufficient. ■

4.2 Complexity estimates

As outlined in Section 2, the HCFEFT and thus Step 1 of Algorithm 4.1 with a fixed oversampling exponent $\alpha \in \mathbb{N}$ takes $\mathcal{O}(2^n n^d)$ floating point operations. Moreover, the spline interpolation in Step 2, cf. Section 3.2, and the spline evaluation in Step 3, cf. Section 3.3, are realised in $\mathcal{O}(2^n n^{d-1} m)$ and $\mathcal{O}(M n^{d-1} m^d)$ operations, respectively. For $d \in \mathbb{N}$, $M = |H_n^d|$ nodes, and a target accuracy $\varepsilon > 0$ this sums up to a total complexity of

$$\mathcal{O} \left(2^n n^{2d-2} (|\log \varepsilon|^d + \log^d n) \right)$$

instead of $\mathcal{O}(2^{2n} n^{2d-2})$ for the nonequispaced HCDFT, cf. (2.7).

5 Implementation and numerical results

For the reader's convenience, we provide an efficient and reliable implementation of the presented algorithms in Matlab. Following the commonly accepted concept of *reproducible research*, all numerical experiments are included in our publicly available toolbox [9]. The toolbox includes an interface to the tensor toolbox [1] for simple handling of sparse grids and hyperbolic crosses. All numerical results were obtained on an Intel Xeon Dual Core CPU with 3GHz, 64GByte RAM running OpenSUSE Linux 10.3 X86-64 and Matlab 7.8.0.347. Time measurements use the Matlab function `cputime`. For dimensions $d \in \mathbb{N}$ and refinements $n \in \mathbb{N}$, we choose Fourier coefficients $\hat{f}_{\mathbf{k}} \in \mathbb{C}$, $\mathbf{k} \in H_n^d$, uniformly at random with $|\hat{f}_{\mathbf{k}}| \leq 1$. Within the nonequispaced HCFFT, the sampling nodes $\mathbf{x}_j \in \mathbb{T}^d$ are also drawn uniformly at random.

5.1 The hyperbolic cross fast Fourier transform

We compare the computation time of the hyperbolic cross discrete Fourier transform (HCDFT) and the hyperbolic cross fast Fourier transform (HCFFT). For the HCDFT we compare two different methods, the direct summation

$$f_{\mathbf{x}} = \sum_{\mathbf{k} \in H_n^d} \hat{f}_{\mathbf{k}} e^{2\pi i \mathbf{k} \mathbf{x}}, \quad \mathbf{x} \in S_n^d,$$

and the matrix vector multiplication with the explicitly set up Fourier matrix \mathbf{F}_n^d , cf. (2.6), which is denoted by (matrix-vector) subsequently and requires $\mathcal{O}(2^{2n} n^{2d-2})$ bytes for storage.

Figure 1(a)–1(c) show the computation time of the HCDFT and HCFFT for the two, three and ten dimensional case and increasing refinement n . Besides the better asymptotic complexity $\mathcal{O}(2^n n^d)$ instead of $\mathcal{O}(2^{2n} n^{2d-2})$, the HCFFT also has a low break even with the HCDFT variants at refinements $n \approx 6$ (direct summation) and $n \leq 12$ (matrix-vector). Thus, we observe that the HCFFT outperforms the HCDFT at small refinements and for arbitrary spatial dimension. Moreover, Figure 1(d) gives the computational times for a constant refinement $n = 6$ and increasing dimension $d = 2, \dots, 20$.

5.2 Accuracy of the nonequispaced HCFFT

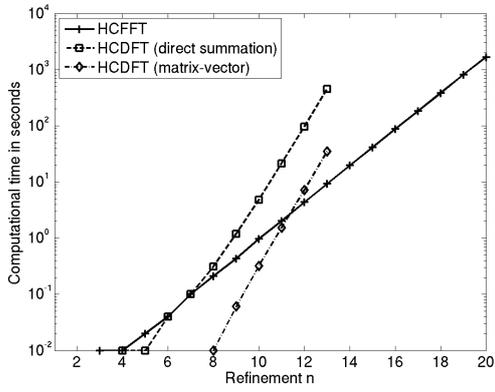
In a second experiment, we examine the accuracy of the NHCFFT against the NHCDFT with respect to an increasing spline order in Figure 5.2. The error of Algorithm 4.1 is measured by

$$E_{\infty} = \frac{\max_{\ell=0, \dots, M-1} |(\mathcal{I} - \mathcal{L}_{n+\alpha}^{(d)})f(\mathbf{x}_{\ell})|}{\sum_{\mathbf{k} \in H_n^d} |\hat{f}_{\mathbf{k}}|}$$

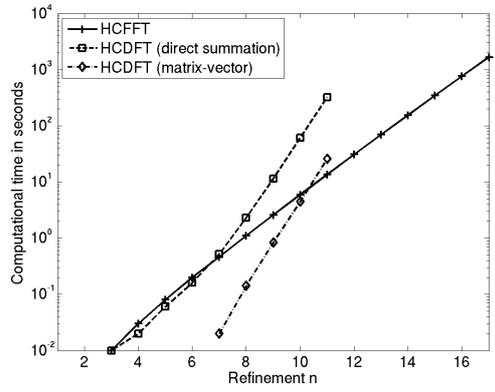
which is of course bounded by Theorem 4.1 and Remark 4.2. The error decays exponentially with increasing spline order $m \in 2\mathbb{N}$. However, a certain loss of accuracy sets in for large spline orders which might be due to the numerical precalculation of the zeros of certain Euler-Frobenius polynomials used in the spline interpolation step.

5.3 Computational times of the nonequispaced HCFFT

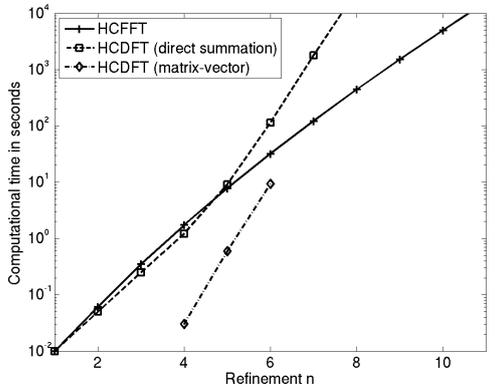
Finally, we compare the computational times for the naive evaluation of (2.2) and the NHCFFT with respect to an increasing refinement and a number of nodes $M = |H_n^d|$ in Figure 5.3. The better asymptotic complexity $\mathcal{O}(2^n n^{2d-2} (|\log \varepsilon|^d + \log^d n))$ instead of $\mathcal{O}(2^{2n} n^{2d-2})$



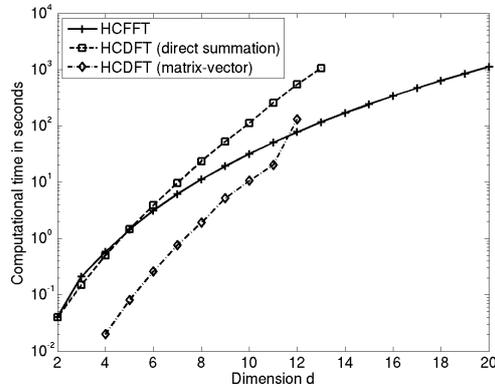
(a) Dimension $d = 2$



(b) Dimension $d = 3$

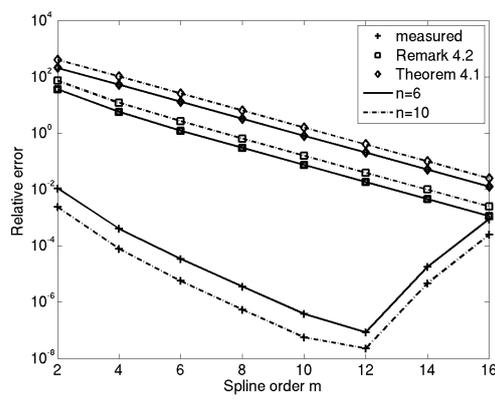
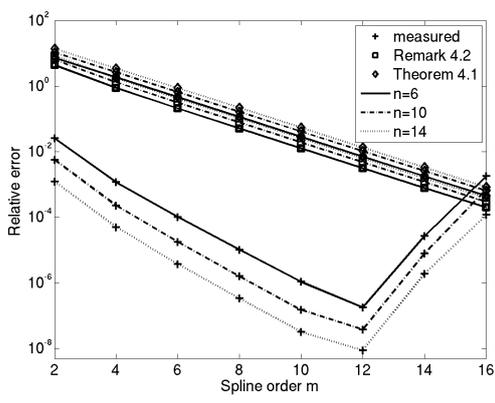


(c) Dimension $d = 10$



(d) Refinement $n = 6$

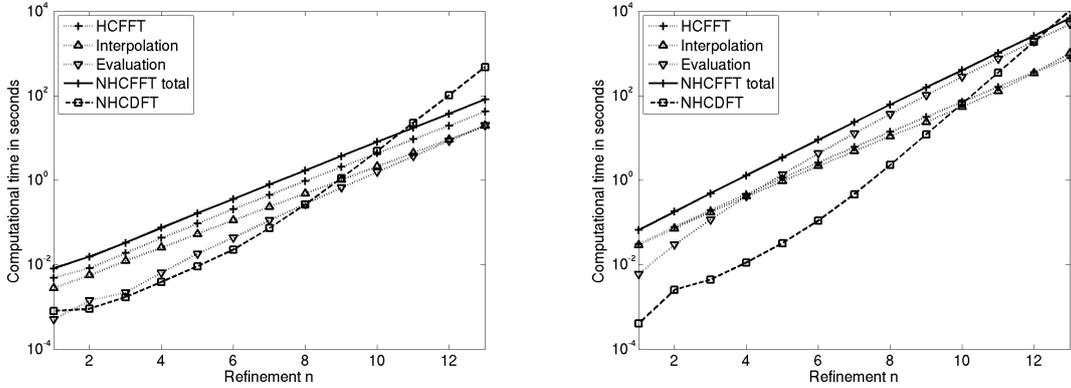
Figure 5.1: Computational time for the HCDFT and HCFFFT with respect to the refinement n and the dimension d .



(a) Dimension $d = 2$; oversampling exponent $\alpha = 2$; (b) Dimension $d = 3$; oversampling exponent $\alpha = 3$; refinement $n = 6, 10, 14$

Figure 5.2: Relative error E_∞ of NHCFFT with respect to spline order m .

is supported by the obtained computation times for the bi- and trivariate case. However note that the spline evaluation step dominates the NHCFFT already for $n \geq 5$ and $d = 3$.



(a) Dimension $d = 2$; oversampling exponent $\alpha = 2$; spline order $m = 4$
(b) Dimension $d = 3$; oversampling exponent $\alpha = 3$; spline order $m = 4$

Figure 5.3: Computational times of the NHCDFT, the NHCFFT, and its three steps with respect to the refinement n .

6 Summary

We have shown that the HCFFT can be generalised to nonequispaced nodes efficiently. Analogously to the nonequispaced FFT which relies on one oversampled FFT and a local approximation scheme, the nonequispaced HCFFT uses one oversampled HCFFT and a local approximation scheme by interpolating splines on the sparse grid. The complexity of Algorithm 4.1 is up to logarithmic factors linear in the problem size, the accuracy of the scheme is guaranteed to enter the complexity only as $|\log \varepsilon|^d$. An efficient implementation of the HCFFT for arbitrary spatial dimension and its nonequispaced version for the bi- and trivariate is made publicly available.

Our theoretical and numerical results indicate that (2.2) is computed efficiently by Algorithm 4.1 already for moderate refinements $n \in \mathbb{N}$ and spatial dimensions $d = 2, 3$. The computational dominant, i.e., most expensive step is the evaluation of the spline which scales as $\mathcal{O}(2^n n^{2d-2} (|\log \varepsilon|^d + \log^d n))$.

Acknowledgement

The authors gratefully acknowledge support by German Research Foundation within the project KU 2557/1-1.

References

- [1] B. W. Bader and T. G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Trans. Math. Software*, 32:635 – 653, 2006.

- [2] 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.
- [3] T. Berger and J.-O. Strömberg. Exact reconstruction algorithms for the discrete wavelet transform using spline wavelets. *Appl. Comput. Harmon. Anal.*, 2:392 – 397, 1995.
- [4] G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comput. Harmon. Anal.*, 2:363 – 381, 1995.
- [5] K. Bittner. Fast algorithms for spline wavelets on sparse grids. *SIAM J. Sci. Comput.*, 20:1192 – 1213, 1999.
- [6] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147 – 269, 2004.
- [7] F.-J. Delvos and W. Schempp. *Boolean methods in interpolation and approximation*. Longman Scientific & Technical, Harley, 1989.
- [8] M. Döhler. Nichtäquidistante schnelle Fouriertransformation auf dem hyperbolischen Kreuz. Diplomarbeit, Faculty of Mathematics, Chemnitz University of Technology, 2008.
- [9] M. Döhler, L. Kämmerer, S. Kunis, and D. Potts. NHCFFT, Matlab toolbox for the nonequispaced hyperbolic cross FFT. <http://www.tu-chemnitz.de/~skunis/software.php>, 2009.
- [10] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Stat. Comput.*, 14:1368 – 1393, 1993.
- [11] M. Fenn, S. Kunis, and D. Potts. Fast evaluation of trigonometric polynomials from hyperbolic crosses. *Numer. Algorithms*, 41:339 – 352, 2006.
- [12] V. Gradinaru. Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation. *Computing*, 80:1 – 22, 2007.
- [13] L. Greengard and J.-Y. Lee. Accelerating the nonuniform fast Fourier transform. *SIAM Rev.*, 46:443 – 454, 2004.
- [14] K. Hallatschek. Fouriertransformation auf dünnen Gittern mit hierarchischen Basen. *Numer. Math.*, 63:83 – 97, 1992.
- [15] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, accepted.
- [16] N. P. Korneičuk. Exact error bounds of approximation by interpolating splines in l-metric on the classes w_p^r ($1 \leq p < \infty$) of periodic functions. *Anal. Math.*, 3:109 – 117, 1977.
- [17] G. Plonka. Optimal shift parameters for periodic spline interpolation. *Numer. Algorithms*, 6:297 – 316, 1994.
- [18] 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.

- [19] W. Sickel and T. Ullrich. The Smolyak algorithm, sampling on sparse grids and function spaces of dominating mixed smoothness. *East Journal on Approximations*, 13:387 – 425, 2007.
- [20] F. Sprengel. A class of function spaces and interpolation on sparse grids. *Numer. Funct. Anal. Optim.*, 21:273 – 293, 2000.
- [21] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. *Adv. Comput. Math.*, 9:337 – 353, 1998.
- [22] V. N. Temlyakov. Approximation of functions with bounded mixed derivative. *Proc. Steklov Inst. Math.*, pages vi+121, 1989. A translation of Trudy Mat. Inst. Steklov 178 (1986).
- [23] 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, Germany, 1991.

Preprint Series DFG-SPP 1324

<http://www.dfg-spp1324.de>

Reports

- [1] R. Ramlau, G. Teschke, and M. Zhariy. A Compressive Landweber Iteration for Solving Ill-Posed Inverse Problems. Preprint 1, DFG-SPP 1324, September 2008.
- [2] G. Plonka. The Easy Path Wavelet Transform: A New Adaptive Wavelet Transform for Sparse Representation of Two-dimensional Data. Preprint 2, DFG-SPP 1324, September 2008.
- [3] E. Novak and H. Woźniakowski. Optimal Order of Convergence and (In-) Tractability of Multivariate Approximation of Smooth Functions. Preprint 3, DFG-SPP 1324, October 2008.
- [4] M. Espig, L. Grasedyck, and W. Hackbusch. Black Box Low Tensor Rank Approximation Using Fibre-Crosses. Preprint 4, DFG-SPP 1324, October 2008.
- [5] T. Bonesky, S. Dahlke, P. Maass, and T. Raasch. Adaptive Wavelet Methods and Sparsity Reconstruction for Inverse Heat Conduction Problems. Preprint 5, DFG-SPP 1324, January 2009.
- [6] E. Novak and H. Woźniakowski. Approximation of Infinitely Differentiable Multivariate Functions Is Intractable. Preprint 6, DFG-SPP 1324, January 2009.
- [7] J. Ma and G. Plonka. A Review of Curvelets and Recent Applications. Preprint 7, DFG-SPP 1324, February 2009.
- [8] L. Denis, D. A. Lorenz, and D. Trede. Greedy Solution of Ill-Posed Problems: Error Bounds and Exact Inversion. Preprint 8, DFG-SPP 1324, April 2009.
- [9] U. Friedrich. A Two Parameter Generalization of Lions' Nonoverlapping Domain Decomposition Method for Linear Elliptic PDEs. Preprint 9, DFG-SPP 1324, April 2009.
- [10] K. Bredies and D. A. Lorenz. Minimization of Non-smooth, Non-convex Functionals by Iterative Thresholding. Preprint 10, DFG-SPP 1324, April 2009.
- [11] K. Bredies and D. A. Lorenz. Regularization With Non-convex Separable Constraints. Preprint 11, DFG-SPP 1324, April 2009.
- [12] M. Döhler, S. Kunis, and D. Potts. Nonequispaced Hyperbolic Cross Fast Fourier Transform. Preprint 12, DFG-SPP 1324, April 2009.