NONEQUISPACED HYPERBOLIC CROSS FAST FOURIER TRANSFORM

MICHAEL DÖHLER[†], STEFAN KUNIS[‡], AND DANIEL POTTS[§]

Abstract. 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 based on one hyperbolic cross FFT and a dedicated interpolation by splines on sparse grids. Analogously to the nonequispaced FFT for trigonometric polynomials with Fourier coefficients supported on the full grid, this allows for the efficient evaluation of trigonometric polynomials with Fourier coefficients supported on the hyperbolic cross at arbitrary spatial sampling nodes.

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

2010 AMS Mathematics Subject Classification : 65T40

1. Introduction. The discretisation of high dimensional problems often leads to an exponential growth in the number of degrees of freedom. We consider the d dimensional discrete Fourier transform

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\hat{G}_{\boldsymbol{n}}} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}}, \qquad \boldsymbol{x}\in G_{\boldsymbol{n}}, \tag{1.1}$$

with equispaced nodes in frequency domain $\hat{G}_{\boldsymbol{n}} = \times_{l=1}^{d} \hat{G}_{n_l}$, $\hat{G}_n = \mathbb{Z}\cap(-2^{n-1}, 2^{n-1}]$, and space domain $G_{\boldsymbol{n}} = \times_{l=1}^{d} G_{n_l}$, $G_n = 2^{-n}(\mathbb{Z}\cap[0,2^n))$, which maps $|\hat{G}_{\boldsymbol{n}}| = |G_{\boldsymbol{n}}|$ Fourier coefficients to the same number of sample values. Here $|\hat{G}_{\boldsymbol{n}}|$ denotes the cardinality of the set $\hat{G}_{\boldsymbol{n}} \subset \mathbb{Z}^d$ and even efficient algorithms like the d dimensional fast Fourier transform (FFT) need $\mathcal{O}(|G_{\boldsymbol{n}}|\log|G_{\boldsymbol{n}}|)$ arithmetic operations for (1.1), 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. [27, 28, 25, 6, 24]. Of course, an important issue is the adaption of efficient algorithms to these thinner discretisations such that their total complexity, within logarithmic factors, is still linear in the reduced problem size. Such improvements were studied in [2, 17, 15] and are known as the hyperbolic cross fast Fourier transform (HCFFT).

On the other hand, the FFT has been generalised to the nonequispaced fast Fourier transform (NFFT), cf. [11, 4, 26, 23, 16, 18], which takes $\mathcal{O}(|G_{\boldsymbol{n}}|\log|G_{\boldsymbol{n}}|+M)$ arithmetic operations for the approximate evaluation of the trigonometric polynomial (1.1) at M arbitrary nodes $\boldsymbol{x}\in\mathbb{R}^d$. FFT algorithms can be considered as exact algorithms up to floating-point errors. In contrast, the NFFT algorithm systematically introduces an approximation error in the computations in order to achieve the desired computational complexity. This additional error can be controlled, and if necessary,

[†]Centre de Recherche INRIA Rennes, Michael.Dohler@irisa.fr

[‡]Helmholtz Zentrum München, Institute for Biomathematics and Biometry, and Chemnitz University of Technology, Faculty of Mathematics, kunis@mathematik.tu-chemnitz.de

[§]Chemnitz University of Technology, Faculty of Mathematics, potts@mathematik.tu-chemnitz.de

can be reduced to the order of machine precision. To this end, the NFFT uses an *oversampled* FFT internally and a local approximation, effectively introducing 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, 17, 15] for arbitrary sampling nodes. More specific, we propose a new algorithm for the efficient evaluation of trigonometric polynomials with Fourier coefficients supported on the hyperbolic cross at arbitrary spatial sampling nodes. Earlier work in [13] was based on a partition of the hyperbolic cross and several NFFTs. The complicated partition of the hyperbolic cross makes the implementation of this algorithm hard even for low dimensions e.g. three. Moreover, each NFFT uses one oversampled FFT and a local approximation scheme which have to be glued together at a later stage. In contrast, our new evaluation scheme is based on one oversampled HCFFT, which is easily implemented for arbitrary spatial dimension, using the unidirectional scheme [15], and a local approximation by splines interpolating on the sparse grid [5], which yields a global approximant to the given trigonometric polynomial. Beyond the scope of this paper, our algorithm can be used within iterative schemes for computing Fourier coefficients on the hyperbolic cross from samples at scattered sampling nodes. However note that such reconstructions are expected to be stable only under appropriate Nyquist type criteria which have been given for the full grid case recently in [12, 20].

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. In Section 4, we introduce our novel nonequispaced hyperbolic cross fast Fourier transform (NHCFFT), estimate its approximation error in Theorem 4.1, and give 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 $f : \mathbb{T}^d \to \mathbb{C}$, $f(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}$, and restrict the frequency domain to the hyperbolic cross

$$H_n^d := \bigcup_{\substack{\boldsymbol{j}\in\mathbb{N}_0^d \\ \|\boldsymbol{j}\|_1 = n}} \hat{G}_{\boldsymbol{j}} = \{\boldsymbol{k} \in \hat{G}_{\boldsymbol{j}} : \|\boldsymbol{j}\|_1 = |j_1| + |j_2| + \ldots + |j_d| = n\} \subset \mathbb{Z}^d, \tag{2.1}$$

see Figure 2.1(a). Our aim is the fast approximate evaluation of the d-variate trigonometric polynomial

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

at nodes $\boldsymbol{x}_\ell \in \mathbb{T}^d$, $\ell = 0, \ldots, M-1$. The space of all such trigonometric polynomials with Fourier coefficients supported on the hyperbolic cross is denoted by $\Pi_n^{\mathrm{hc}}(\mathbb{T}^d)$. For the moment, we restrict ourselves to sparse grids

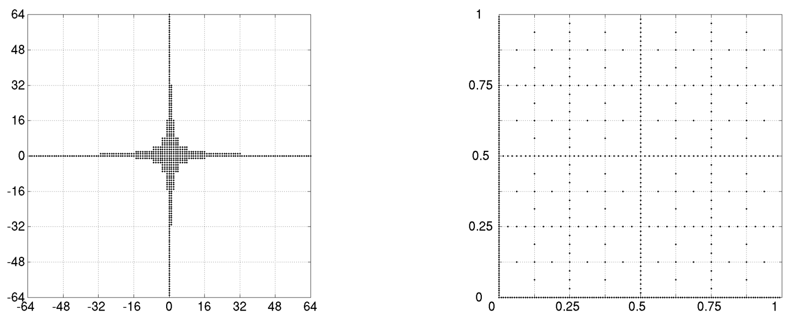
$$S_n^d := \bigcup_{\substack{\boldsymbol{j}\in\mathbb{N}_0^d \\ \|\boldsymbol{j}\|_1 = n}} G_{\boldsymbol{j}} = \{\boldsymbol{x} \in G_{\boldsymbol{j}} : \|\boldsymbol{j}\|_1 = n\} \quad \subset \quad \mathbb{T}^d, \tag{2.3}$$

see Figure 2.1(b), as sampling sets and obtain the following well known results, cf. [17]. For fixed dimension $d \in \mathbb{N}$ and arbitrary refinement $n \in \mathbb{N}_0$, we have the partition

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

and the cardinality estimates

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



(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 $\boldsymbol{x}_\ell \in S_n^d$, $\ell = 0, \ldots, |S_n^d| - 1$, 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, 17]. 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

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

takes $\mathcal{O}(2^{2n} n^{2d-2})$ floating point operations. Given an arbitrary sampling set

$$\mathcal{X} := \{\boldsymbol{x}_\ell \in \mathbb{T}^d : \ell = 0, \ldots, M-1\},$$

the naive evaluation of (2.2), i.e., the matrix vector multiplication with the nonequispaced hyperbolic cross Fourier matrix

$$\boldsymbol{A}_n^d := \left(\mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}\right)_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{k} \in H_n^d} \tag{2.7}$$

takes $\mathcal{O}(2^n n^{d-1} M)$ floating point operations.

3. Periodic splines.

3.1. Univariate interpolation. Let the cardinal B-spline $N_m : \mathbb{R} \to \mathbb{R}$,

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

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 $\phi_r : \mathbb{T} \to \mathbb{C}$, its translates $\phi_{r,k} : \mathbb{T} \to \mathbb{C}$,

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

and the corresponding spaces

$$V_r := \operatorname{span}\{\phi_{r,k} : k = 0, \dots, 2^r - 1\}. \tag{3.3}$$

For $f \in C(\mathbb{T})$, let the interpolation operator $\mathcal{L}_r : C(\mathbb{T}) \to V_r$ be uniquely defined, see [21] for details, by

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

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 [19] state the following.

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

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

built upon the Lagrange polynomials and the modified Bernoulli splines

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

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

be given. Moreover, let $f : [0,1] \to \mathbb{C}$ be m-times continuously differentiable and let $f^{(m)}$ denote its m-th derivative. Then, the interpolation error allows for the representation

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

Moreover, the error is bounded since

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

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

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

$$\mathcal{L}_r^{(d)} := \bigoplus_{\substack{\boldsymbol{j}\in\mathbb{N}_0^d \\ \|\boldsymbol{j}\|_1=r}} \mathcal{L}_{j_1} \otimes \ldots \otimes \mathcal{L}_{j_d}, \qquad V_r^{(d)} := \operatorname{Im} \mathcal{L}_r^{(d)},$$

where the superscript is skipped for $d = 1$. 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

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

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

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

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

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

2. *moreover, the Boolean sum can be expressed as*

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

3. *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 [8, Sec. 1-2]. Due to its definition, the operator $\mathcal{L}_r^{(d)}$ interpolates on all grids $G_{\boldsymbol{j}}$, $\|\boldsymbol{j}\|_1 = r$, and thus on S_r^d, cf. [8, Sec. 2.3, Prop. 2]. The second assertion is due to [8, Sec. 1.3, Prop. 2]. Moreover, the target spaces allow for the recursion

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

and thus $\tilde{V}_r^{(d)}$ yields a generating system for $V_r^{(d)}$. Finally, [8, Sec. 2.3, Prop. 4] yields the dimension of $V_r^{(d)}$ and since the number of multi-indices $\boldsymbol{j}$ with $\|\boldsymbol{j}\|_1 = r$ 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.,

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

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

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

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 $C_1 m (r-l)^{d-1} 2^{r-l}$ arithmetic operations, an induction argument yields a total complexity of $\mathcal{O}(2^r r^{d-1} m)$ for computing the spline coefficients $a_{\boldsymbol{j},\boldsymbol{k}}$, $\boldsymbol{j}, \boldsymbol{k} \in \mathbb{N}_0^d$, $\|\boldsymbol{j}\|_1 = r$, $\boldsymbol{k} < 2^{\boldsymbol{j}}$, from the samples $f(\boldsymbol{x})$, $\boldsymbol{x} \in S_r^d$. 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 $\boldsymbol{x} \in \mathbb{T}^d$. For fixed $x \in \mathbb{T}$, $j \in \mathbb{N}_0$, and $k = 0, \ldots, 2^j - 1$, we have

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

where $\lfloor \cdot \rfloor$ denotes the floor function. 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 $\boldsymbol{j} \in \mathbb{N}_0^d$, $\|\boldsymbol{j}\|_1 = r$, 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 $\mathcal{L}_r^{(d)} f(\boldsymbol{x})$.

4. The nonequispaced hyperbolic cross fast Fourier transform. We combine the hyperbolic cross FFT [2, 17, 15] and the spline approximation scheme [5] generalised to spatial dimension $d \in \mathbb{N}$ in the following Algorithm 1 for the fast and approximate multiplication with the Fourier matrix $\boldsymbol{A}_n^d$, cf. (2.7). The multiplication with the adjoint Fourier matrix $(\boldsymbol{A}_n^d)^*$ easily follows and is denoted by adjoint NHCFFT subsequently.

4.1. Error estimates. Algorithm 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 $f \in \Pi_n^{\mathrm{hc}}(\mathbb{T}^d)$ be given. Then the interpolation error can be bounded by*

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

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 $\Pi_n^{\mathrm{hc}}(\mathbb{T})$, the univariate complement operator $\mathcal{K}_r := \mathcal{I} - \mathcal{L}_r$ allows for the error estimate

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

Algorithm 1 Nonequispaced hyperbolic cross FFT (NHCFFT)

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

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

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in H_n^d} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}, \qquad \boldsymbol{x} \in S_{n+\alpha}^d.$$

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

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

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

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

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

and shows (4.1) for $d = 1$. We generalise to the d-variate complement operator $\mathcal{K}_r^{(d)} := \mathcal{I} - \mathcal{L}_r^{(d)}$, where $\mathcal{I}$ always denotes the identity operator for functions in an appropriate number of variables. Using (3.4) from Lemma 3.2, we note for $d \geq 2$ that

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

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

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

and use Lemma 3.1 to estimate

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

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

$$\mathcal{K}_j^{(d-1)} \otimes \frac{\mathrm{d}^m}{\mathrm{d}z^m} f(\boldsymbol{x}, z) = \mathcal{K}_j^{(d-1)} \otimes \frac{\mathrm{d}^m}{\mathrm{d}z^m} \left(\sum_{(\boldsymbol{k},l) \in H_n^d} \hat{f}_{\boldsymbol{k},l} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} \mathrm{e}^{2\pi \mathrm{i} l z} \right)$$

$$= \sum_{s=0}^{n} \sum_{l \in \hat{G}_s \setminus \hat{G}_{s-1}} (2\pi \mathrm{i} l)^m \mathrm{e}^{2\pi \mathrm{i} l z} \cdot \mathcal{K}_j^{(d-1)} \left(\sum_{\boldsymbol{k} \in H_{n-s}^{d-1}} \hat{f}_{\boldsymbol{k},l} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}} \right).$$

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

$$|\mathcal{K}_j^{(d-1)} \otimes \frac{\mathrm{d}^m}{\mathrm{d}z^m} f(\boldsymbol{x}, z)| \leq \sum_{s=0}^{n} \sum_{l \in \hat{G}_s \setminus \hat{G}_{s-1}} (2\pi |l|)^m \frac{(2j+2)^{d-2} F_m^{d-1} 2^{(n-s)m}}{2^{(j-d+2)m}} \sum_{\boldsymbol{k} \in H_{n-s}^{d-1}} |\hat{f}_{\boldsymbol{k},l}|$$

$$\leq \frac{(2j+2)^{d-2} F_m^{d-1} 2^{nm} \pi^m}{2^{(j-d+2)m}} \sum_{(\boldsymbol{k},l) \in H_n^d} |\hat{f}_{\boldsymbol{k},l}|.$$

In conjunction with (4.3) this yields

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

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

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

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

$$\|\mathcal{K}_j^{(d)} f\|_\infty \,/ \sum_{(\boldsymbol{k},l) \in H_n^d} |\hat{f}_{\boldsymbol{k},l}| \leq \frac{(2r+2)^{d-2} F_m^{d-1} 2^{nm}}{2^{(r-d+2)m}} + \frac{F_m 2^{nm}}{2^{rm}}$$

$$+ \sum_{j=0}^{r-1} \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+1)m}} + \sum_{j=0}^{r} \frac{(2j+2)^{d-2} F_m^d 2^{nm}}{2^{(r-d+2)m}}$$

$$\leq \frac{(2r+2)^{d-1} F_m^d 2^{nm}}{2^{(r-d+1)m} 2} C_{m,r,d},$$

where

$$C_{m,r,d} := \frac{1}{(r+1) F_m 2^m} + \frac{1}{(r+1)^{d-1} F_m^{d-1} 2^{(d-1)m}} + \frac{1}{d-1} + \frac{1}{2^m (d-1)} + \frac{1}{2^m (r+1)}.$$

Bounding the term $C_{m,r,d}$ from above by two ($d \geq 2$, $r \geq 0$, $m \geq 2$, and $F_m \geq 1$) yields the assertion. □

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. [26, Cor. 4.3]. While Algorithm 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 [26, Algorithm 2.1 with (4.4)].*

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

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

for $d=2$ *and to*

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

for $d=3$, *see [9, Theorems 3.19 and 3.24].*

Corollary 4.3. *Let a spatial dimension* $d\in\mathbb{N}$, *a refinement* $n\in\mathbb{N}$ *with* $n\ge d$, *and a target accuracy* $\varepsilon>0$ *be given and choose the oversampling exponent* $\alpha=d$ *in Algorithm 1. Then for spline orders* $m\in 2\mathbb{N}$ *with*

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

the complex exponentials are approximated such that

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

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

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

for which (4.4) is sufficient. □

4.2. Complexity estimates. As outlined in Section 2, the HCFFT and thus Step 1 of Algorithm 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 $\mathcal{O}(Mn^{d-1}m^d)$ 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

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

instead of $\mathcal{O}(2^{2n}n^{2d-2})$ for the nonequispaced HCDFT, cf. (2.7). We summarise the arithmetic complexity of our new Algorithm in comparison to other discrete and fast Fourier transforms in the following Table 4.1.

Finally, we consider the memory usage of the algorithms for the matrix vector multiplication with the Fourier matrix $\boldsymbol{A}_n^d$, cf. (2.7). The NHCDFT uses direct calls to the complex exponentials and thus takes only $\mathcal{O}(2^n n^{d-1}+M)$ bytes for storing the input and output vectors, i.e., is linear in the problem size. The NHCFFT has the same space complexity but introduces an exponential factor in d due to oversampling. The fastest method for this matrix vector multiplication and small refinements is to set up the Fourier matrix $\boldsymbol{A}_n^d$ explicitly. This takes $\mathcal{O}(2^n n^{d-1}M)$ bytes of storage, which is too costly already for moderate problem sizes.

Algorithm	$\boldsymbol{k}$	Size	$\boldsymbol{x}$	Size	$\mathcal{O}$-complexity	Reference
DFT	$\hat{G}_{\boldsymbol{n}}$	2^{nd}	$G_{\boldsymbol{n}}$	2^{nd}	2^{2nd}	
FFT	$\hat{G}_{\boldsymbol{n}}$	2^{nd}	$G_{\boldsymbol{n}}$	2^{nd}	$2^{nd}nd$	e.g. [7, 14]
NDFT	$\hat{G}_{\boldsymbol{n}}$	2^{nd}	$\mathcal{X}$	M	$2^{nd}M$	
NFFT	$\hat{G}_{\boldsymbol{n}}$	2^{nd}	$\mathcal{X}$	M	$2^{nd}+M$	e.g. [18]
HCDFT	H_n^d	$2^n n^{d-1}$	S_n^d	$2^n n^{d-1}$	$2^{2n}n^{2d-2}$	
HCFFT	H_n^d	$2^n n^{d-1}$	S_n^d	$2^n n^{d-1}$	$2^n n^d$	[2, 17, 15]
NHCDFT	H_n^d	$2^n n^{d-1}$	$\mathcal{X}$	M	$2^n M n^{d-1}$	
NHCFFT	H_n^d	$2^n n^{d-1}$	$\mathcal{X}$	M	$2^n n^d + Mn^{d-1}\log^d n$	Algorithm 1

Table 4.1

Problem sizes and computational complexities for increasing refinement n, fixed accuracy, and fixed spatial dimension.

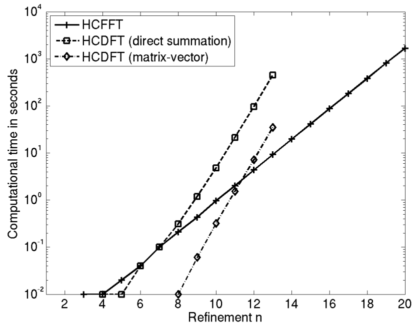
5. Implementation and numerical results. For the reader's convenience, we provide an efficient and reliable implementation of the presented algorithms in Matlab. Following the commonly accepted concept of *reproducible research*, all numerical experiments are included in our publicly available toolbox [10]. 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 were performed by employing the Matlab function `cputime`. For dimensions $d \in \mathbb{N}$ and refinements $n \in \mathbb{N}$, we choose Fourier coefficients $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$, $\boldsymbol{k} \in H_n^d$, uniformly at random with $|\hat{f}_{\boldsymbol{k}}| \leq 1$. Within the nonequispaced HCFFT, the sampling nodes $\boldsymbol{x}_j \in \mathbb{T}^d$ 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

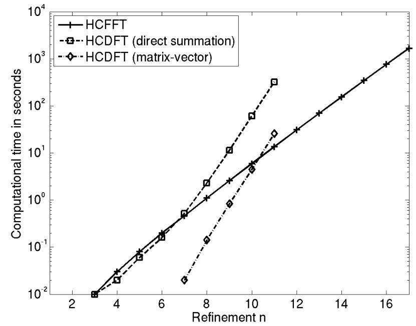
$$f_{\boldsymbol{x}} = \sum_{\boldsymbol{k} \in H_n^d} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\boldsymbol{x}}, \qquad \boldsymbol{x} \in S_n^d,$$

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

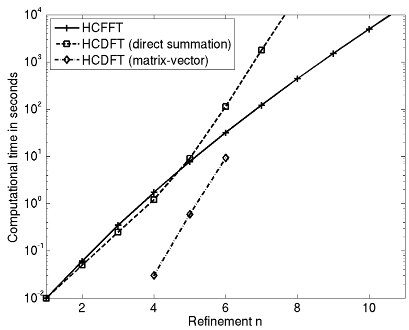
Figure 5.1(a)–5.1(c) show the computation time of the HCDFT and HCFFT for the two, three and ten dimensional case and increasing refinement n. Besides the better asymptotic complexity $\mathcal{O}\left(2^n n^d\right)$ instead of $\mathcal{O}\left(2^{2n}n^{2d-2}\right)$, the HCFFT also has a low break 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 5.1(d) gives the computational times for a constant refinement $n = 6$ and increasing dimension $d = 2, \ldots, 20$.



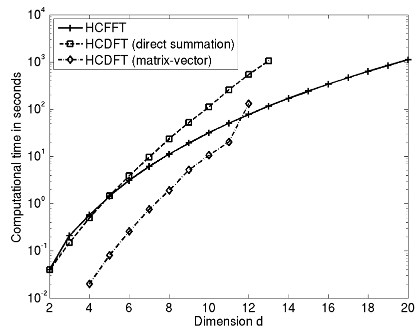
(a) Dimension $d = 2$



(b) Dimension $d = 3$



(c) Dimension $d = 10$



(d) Refinement $n = 6$

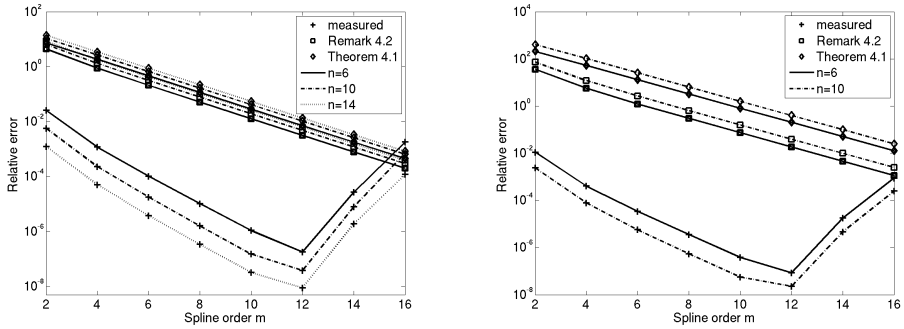
FIGURE 5.1. *Computational time for the HCDFT and HCFFT with respect to the refinement n and the dimension d.*

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 1 is measured by

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

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.

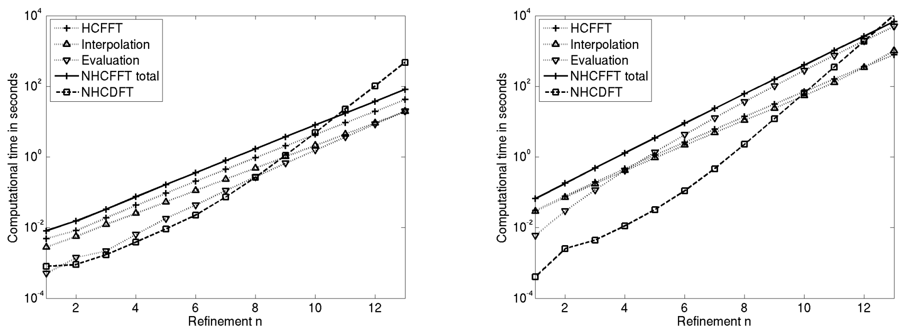
5.3. Computational times of the nonequispaced HCFFT. We compare the computational times for the naive evaluation of (2.2) and the NHCFFT with respect to an increasing refinement and a number of nodes $M = |H_n^d|$ in Figure 5.3. The better asymptotic complexity $\mathcal{O}(2^n n^{2d-2}(|\log \varepsilon| + \log n)^d)$ instead of $\mathcal{O}\left(2^{2n} n^{2d-2}\right)$ 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 $\alpha = 2$; refinement $n = 6, 10, 14$

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

FIGURE 5.2. *Relative error E_∞ of NHCFFT with respect to spline order m.*



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

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

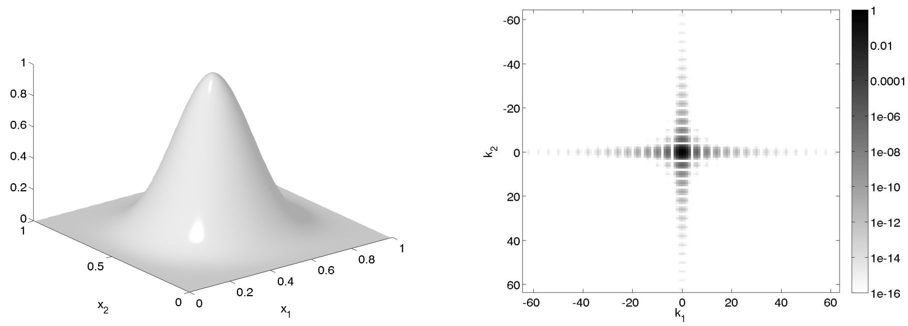
FIGURE 5.3. *Computational times of the NHCDFT, the NHCFFT, and its three steps with respect to the refinement n.*

5.4. Fast summation with kernels of dominating mixed smoothness. Finally, we consider an example which shows the compression and fast application of a kernel matrix to a vector, where the kernel is non-local, translation invariant, and of dominating mixed smoothness. Let the B-spline kernel matrix

$$\boldsymbol{K} = \left(N_4^{(2)}(4(\boldsymbol{x}_j - \boldsymbol{y}_l + \frac{1}{2}) \right)_{l=1,\dots,L; j=1,\dots,M}, \quad N_4^{(2)} = N_4 \otimes N_4, \quad \boldsymbol{x}_j, \boldsymbol{y}_l \in [\frac{1}{4}, \frac{3}{4}),$$

be given. Figure 5.4 shows $N_4^{(2)}(4\cdot)$ and the Fourier coefficients $d_{\boldsymbol{k}}$, $\boldsymbol{k} \in \hat{G}_{(7,7)}$, of the trigonometric polynomial interpolating $N_4^{(2)}(4\cdot)$ at the full grid $G_{(7,7)}$, i.e., the refinement is set to $n = 7$. Clearly, the Fourier coefficients on $\hat{G}_{(7,7)} \setminus H_7^2$ are small and might be neglected causing a relative error of 10^{-4}. The remaining Fourier coefficients are used in $\boldsymbol{d} = (d_{\boldsymbol{k}})_{\boldsymbol{k} \in H_7^2}$ and following the ideas in [22], we set up a degenerate approximation of the B-spline kernel and thus the matrix approximation

$$\boldsymbol{K} \approx \boldsymbol{A}_{\mathcal{Y}} (\mathrm{diag} \boldsymbol{d}) \boldsymbol{A}_{\mathcal{X}}^*, \quad \boldsymbol{A}_{\mathcal{Y}} = \left(\mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \boldsymbol{y}_l} \right)_{l=1,\dots,L, \boldsymbol{k} \in H_7^2}, \ \boldsymbol{A}_{\mathcal{X}} = \left(\mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \boldsymbol{x}_j} \right)_{j=1,\dots,M, \boldsymbol{k} \in H_7^2}.$$



(a) Spatial domain (b) Frequency domain

FIGURE 5.4. *B-spline* $N_4^{(2)}(4\cdot)$ *and Fourier coefficients* $d_{\boldsymbol{k}}$*, both normalised.*

Similar to the fast multiplication with a circulant matrix, the approximate matrix vector multiplication with $\boldsymbol{K}$ can be realised by an adjoint NHCFFT, a pointwise multiplication in frequency domain, and an NHCFFT, i.e.,

$$\boldsymbol{K}\boldsymbol{g} = \boldsymbol{h} \approx \tilde{\boldsymbol{h}} = \boldsymbol{A}_{\mathcal{Y}}(\mathrm{diag}\boldsymbol{d})\boldsymbol{A}_{\mathcal{X}}^{*}\boldsymbol{g}.$$

The naive matrix vector multiplication, having complexity $\mathcal{O}(LM)$, takes 5.56 seconds for $M = L = 3200$ nodes when using columnwise updates. In contrast, the above approximation yields a complexity $\mathcal{O}(L+M)$ and takes only 2.63 seconds if we set the spline order $m = 4$ and the oversampling exponent $\alpha = 2$ as NHCFFT parameters. This results in an error $\|\boldsymbol{h} - \tilde{\boldsymbol{h}}\|_\infty / \|\boldsymbol{g}\|_1 \leq 10^{-5}$, $g_j \in [-1, 1]$ drawn uniformly at random.

6. Summary. We have shown that the HCFFT can be generalised to nonequispaced nodes in order to evaluate trigonometric polynomials with Fourier coefficients supported on the hyperbolic cross at arbitrary spatial sampling 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 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 in [10].

Our theoretical and numerical results indicate that (2.2) is computed efficiently by Algorithm 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 $\mathcal{O}(2^n n^{2d-2}(|\log \varepsilon| + \log n)^d)$.

Acknowledgement. The authors gratefully acknowledge support by German Research Foundation within the project KU 2557/1-1. SK acknowledges support by the Helmholtz Association, contract number VH-NG-526. Moreover, we thank the referees for their valuable suggestions and Lutz Kämmerer for the further development of the software toolbox [10].

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] J. W. Cooley and J. W. Tukey. An algorithm for machine calculation of complex Fourier series. *Math. Comput.*, 19:297 – 301, 1965.
[8] F.-J. Delvos and W. Schempp. *Boolean methods in interpolation and approximation*. Longman Scientific & Technical, Harley, 1989.
[9] M. Döhler. Nichtäquidistante schnelle Fouriertransformation auf dem hyperbolischen Kreuz. Diplomarbeit, Faculty of Mathematics, Chemnitz University of Technology, 2008.
[10] 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.
[11] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Stat. Comput.*, 14:1368 – 1393, 1993.
[12] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numer. Math.*, 69:423 – 440, 1995.
[13] M. Fenn, S. Kunis, and D. Potts. Fast evaluation of trigonometric polynomials from hyperbolic crosses. *Numer. Algorithms*, 41:339 – 352, 2006.
[14] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the* IEEE, 93:216–231, 2005.
[15] V. Gradinaru. Fourier transform on sparse grids: Code design and the time dependent Schrödinger equation. *Computing*, 80:1 – 22, 2007.
[16] L. Greengard and J.-Y. Lee. Accelerating the nonuniform fast Fourier transform. *SIAM Rev.*, 46:443 – 454, 2004.
[17] K. Hallatschek. Fouriertransformation auf dünnen Gittern mit hierarchischen Basen. *Numer. Math.*, 63:83 – 97, 1992.
[18] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, 36:Article 19, 1 – 30, 2009.
[19] N. P. Korneĭčuk. Exact error bounds of approximation by interpolating splines in l-metric on the classes w_p^r $(1 \le p < \infty)$ of periodic functions. *Anal. Math.*, 3:109 – 117, 1977.
[20] S. Kunis and D. Potts. Stability results for scattered data interpolation by trigonometric polynomials. *SIAM J. Sci. Comput.*, 29:1403 – 1419, 2007.
[21] G. Plonka. Optimal shift parameters for periodic spline interpolation. *Numer. Algorithms*, 6:297 – 316, 1994.
[22] D. Potts, G. Steidl, and A. Nieslony. Fast convolution with radial kernels at nonequispaced knots. *Numer. Math.*, 98:329 – 351, 2004.
[23] 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, Boston, MA, USA, 2001. Birkhäuser.
[24] 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.
[25] F. Sprengel. A class of function spaces and interpolation on sparse grids. *Numer. Funct. Anal. Optim.*, 21:273 – 293, 2000.
[26] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. *Adv. Comput. Math.*, 9:337 – 353, 1998.
[27] 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).
[28] 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.