On the computation of the polar FFT

Markus Fenn[*] Stefan Kunis[†] Daniel Potts[‡]

We show that the polar as well as the pseudo-polar FFT can be computed very accurately and efficiently by the well known nonequispaced FFT. Furthermore, we discuss the reconstruction of a $2d$ signal from its Fourier transform samples on a (pseudo-)polar grid by means of the inverse nonequispaced FFT.

Key words and phrases: Nonequispaced FFT, polar Fourier transform, polar grid, linogram grid, FFT, NFFT, USFFT

2000 AMS Mathematics Subject Classification: 65T50, 65T40

1 Introduction

In recent years, the nonequispaced FFT (NFFT) as a universal tool for the fast approximate evaluation of a multivariate trigonometric polynomial at arbitrary nodes has attracted much attention. Its accuracy is adjusted to the practical requirements in a simple way by an oversampling factor and a cut-off parameter. In particular, the accuracy does not depend on the sampling nodes. The inverse NFFT can be computed with a CG-type algorithm utilising one NFFT and one adjoint NFFT per iteration. This reconstruction algorithm produces very good results for an appropriate sampling geometry and corresponding weights.

The authors in [1] propose a fast polar Fourier transform (polar FFT) based on the chirp-z transform, see e.g. [3], followed by $1d$ interpolations and conclude that their scheme might be superior to the well known nonequispaced FFT [17] for this setting. In contrast, our numerical experiments strongly indicate that the computation of the polar and the pseudo-polar FFT by means of our mature software package [12], based on [7, 5, 18] and the tutorials [19, 17, 11], is indeed highly efficient and very accurate. In particular, the oversampling factors used in [1] to achieve a certain accuracy are by no means necessary. This is also covered by the error analysis in [18, 17] showing that

[*] Markus.Fenn@uni-mannheim.de, University of Mannheim, Institute of Mathematics, D–68159 Mannheim

[†] kunis@mathematik.tu-chemnitz.de, Chemnitz University of Technology, Department of Mathematics, 09107 Chemnitz, Germany

[‡] potts@mathematik.tu-chemnitz.de, Chemnitz University of Technology, Department of Mathematics, 09107 Chemnitz, Germany

the error decays exponentially fast with respect to a cut-off parameter which enters the computational complexity quadratic for $d = 2$, while the error decays only algebraically with respect to the oversampling factor.

The slightly different Fourier transform based on the pseudo-polar grid, known as linogram grid for decades [15], and its efficient computation have been considered in [16]. Its recent applications, including the ridgelet transform [14] and the curvelet transform [6], might actually need only low accuracy what comes with another speed up of the computations.

Finally, we focus on the inverse polar FFT, i.e., the reconstruction of an image from its polar FFT. In contrast to [1], we use explicit estimates on the condition number of this problem with respect to the mesh-norm of the sampling set, cf. [10, 8], to obtain stable polar and pseudo-polar grids.

The paper is organised as follows: After introducing the necessary notation for the nonequispaced FFT and its inverse in the next section, we investigate the polar, a modified polar and the linogram grid in Section 3. Various numerical examples concerning the computation time, the accuracy of the forward transform, and the reconstruction error of the inverse transform are presented in Section 4. Finally, we draw our conclusions.

2 Nonequispaced FFT and its inverse

As usual, let the torus $\mathbb{T}^2$ be represented by the unit square $[-\frac{1}{2}, \frac{1}{2}]^2$ with opposing faces identified. For $N \in \mathbb{N}$ let the index set $I_N := \mathbb{Z} \cap [-\frac{N}{2}, \frac{N}{2})$ and its Cartesian product $I_N^2 := I_N \times I_N$ be given. For a finite number of given Fourier coefficients $\hat{f}_{\boldsymbol{k}} \in \mathbb{C}$ $(\boldsymbol{k} \in I_N^2)$, the bivariate NFFT evaluates the trigonometric polynomial

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in I_N^2} \hat{f}_{\boldsymbol{k}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}} \tag{2.1}$$

at arbitrary nodes $\boldsymbol{x}_j \in \mathbb{T}^2$ $(j = 1, \ldots, M)$ in only $\mathcal{O}(|I_N^2| \log|I_N^2| + |\log\varepsilon|^2 M)$ arithmetic operations, where ε denotes the target accuracy. In matrix-vector notation this reads as $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$, where

$$\boldsymbol{f} := (f(\boldsymbol{x}_j))_{j=1}^M, \qquad \boldsymbol{A} := \left(\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\boldsymbol{x}_j}\right)_{j=1,\boldsymbol{k}\in I_N^2}^M, \qquad \hat{\boldsymbol{f}} := \left(\hat{f}_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in I_N^2}$$

denote the vector of samples, the nonequispaced Fourier matrix, and the vector of Fourier coefficients, respectively. The accuracy of our fast algorithms for this matrix vector product do not depend on the particular distribution of the nodes but only on the used window function and its cut-off parameter, see [18, 17, 13] for details.

Furthermore, we consider the following reconstruction problem. Given the samples $(\boldsymbol{x}_j, y_j) \in \mathbb{T}^2 \times \mathbb{C}$ $(j = 1, \ldots, M)$ of a trigonometric polynomial, the aim of the *inverse NFFT* is to reconstruct its Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ $(\boldsymbol{k} \in I_N^2)$. Hence, we have to solve the linear system of equations

$$\boldsymbol{A}\hat{\boldsymbol{f}} = \boldsymbol{y} \tag{2.2}$$

for the unknown vector of Fourier coefficients $\hat{\boldsymbol{f}}$. Of course, for equally spaced nodes $\{\boldsymbol{x}_j : j = 1, \ldots, M\} = N^{-1} I_N^2$ the inverse NFFT becomes an ordinary inverse FFT which can be easily computed. We take the non-uniformity of the sampling set into account by defining the mesh-norm

$$\delta := 2 \max_{\boldsymbol{x} \in \mathbb{T}^2} \min_{j=1,\ldots,M} \min_{\boldsymbol{k} \in \mathbb{Z}^2} \|\boldsymbol{x}_j - \boldsymbol{x} + \boldsymbol{k}\|_\infty ,$$

which can be interpreted as the maximum distance between neighbouring nodes. Further, we introduce weights $w_j > 0$ to compensate for local sampling density variations. Let $\boldsymbol{W} := \operatorname{diag}(w_j)_{j=1}^M$. Motivated from the Cartesian setting, we force the mesh-norm to be smaller than N^{-1}, and hence, the number of nodes to be larger than the dimension of the space of trigonometric polynomials, i.e., $M \geq |I_N^2|$. Thus, the linear system (2.2) is overdetermined and a standard method is to use a least squares approach, solving the unconstrained minimisation problem

$$\left\| \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{f}} \right\|_{\boldsymbol{W}}^2 = \sum_{j=1}^{M} w_j \left| y_j - f(\boldsymbol{x}_j) \right|^2 \xrightarrow{\hat{\boldsymbol{f}}} \min . \tag{2.3}$$

This problem is equivalent to the weighted normal equation of the first kind

$$\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A} \hat{\boldsymbol{f}} = \boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{y} . \tag{2.4}$$

Obviously, the matrix $\boldsymbol{A}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{A}$ has two-level Toeplitz structure and approaches the identity for a sampling set tending to the Cartesian grid and equal weights $\boldsymbol{W} = N^{-2} \boldsymbol{I}$. Furthermore, for sufficiently dense sampling sets, i.e., $\delta < CN^{-1}$ with some explicitly known constant, the reconstruction problem is well conditioned and a variant of the conjugate gradients algorithm has been applied successfully, cf. [10, 8]. Based on these considerations, for the numerical solution of (2.4), the NFFT software package [12] provides a factorised variant of the conjugated gradients method (CGNR, N for 'Normal equation', R for 'Residual minimisation'), where the NFFT and its adjoint are used for the fast matrix-vector multiplications.

In contrast to the forward NFFT and its adjoint, the reconstruction error of the inverse NFFT heavily relies on the distribution of the nodes $\boldsymbol{x}_j$. Here, we investigate in particular the polar, a modified polar, and the linogram grid of radial and angular size $R, T \in 2\mathbb{N}$.

3 Grids and weights

3.1 Polar grid

The nodes of the *polar grid* lie on concentric circles around the origin. They are given for $(j, t)^\top \in I_R \times I_T$ by a signed radius $r_j := \frac{j}{R} \in [-\frac{1}{2}, \frac{1}{2})$ and an angle $\theta_t := \frac{\pi t}{T} \in [-\frac{\pi}{2}, \frac{\pi}{2})$ as

$$\boldsymbol{x}_{t,j} := r_j \left(\cos\theta_t, \sin\theta_t \right)^\top .$$

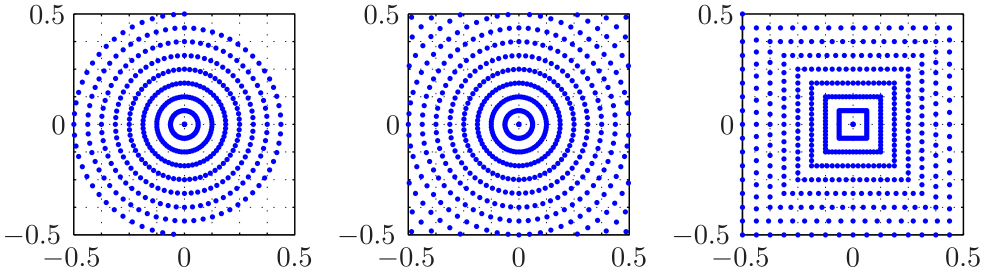

Figure 3.1: Left to right: polar, modified polar, and linogram grid of size $R = 16$, $T = 32$.

The total number of nodes is $M = TR$, whereas the origin is included multiple times.

Obviously, the nodes of the polar grid leave out the corners of the unit square, cf. Figure 3.1(left). While this poses of course no problems for the polar FFT, cf. Example 4.1, its inversion gets ill conditioned and visible artifacts are left, cf. Example 4.2. While [1, Section 3.2] aims to concentrate the Fourier transform within the disk sampled by the polar FFT, we propose to extend the sampling set to fill the corners as follows.

3.2 Modified polar grid

We add more concentric circles and exclude those nodes not located in the unit square, i.e.,

$$\boldsymbol{x}_{t,j} := r_j \left(\cos\theta_t, \sin\theta_t\right)^\top, \qquad (j,t)^\top \in I_{\sqrt{2}R} \times I_T\,.$$

with r_j and θ_t as before, cf. Figure 3.1(middle). The number of nodes for the modified polar grid can be estimated as $M \approx \frac{4}{\pi}\log(1+\sqrt{2})TR$.

3.3 Linogram grid

Instead of concentric circles, the nodes of the *linogram* or *pseudo-polar grid* lie on concentric squares around the origin. Thus, they are typically given by a slope and an intercept. Depending on the slope, we distinguish two sets of nodes, i.e.,

$$\boldsymbol{x}^{\mathrm{BH}}_{t,j} := \left(\frac{j}{R}, \frac{4t}{T}\frac{j}{R}\right)^\top, \qquad \boldsymbol{x}^{\mathrm{BV}}_{t,j} := \left(-\frac{4t}{T}\frac{j}{R}, \frac{j}{R}\right)^\top.$$

where $j \in I_R$ and $t \in I_{\frac{T}{2}}$, cf. Figure 3.1(right). Adding together, the number of nodes for the linogram grid is $M = TR$, where the origin is included multiple times and the node $(-\frac{1}{2}, \frac{1}{2})^\top$ twice.

3.4 Density of the sampling sets

Since the condition on the mesh-norm in [4, Theorem 4.1] is not optimal, we just wish to bound the mesh-norm as $\delta \le N^{-1}$, what also turns out to be a reasonable choice in our numerical examples.

Of course, the polar grid does not meet this criterion, but choosing $R \geq N$ and $T \geq \pi N$ guarantees a mesh-norm $\delta \leq N^{-1}$ on the unit disk $\{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\|_2 \leq \frac{1}{2}\}$.

In order to achieve $\delta \leq N^{-1}$ for the modified polar grid, it suffices to choose $R \geq 2N$ and the angle sampled such that $\sqrt{2}(\frac{1}{2} - N^{-1}) \leq \sin(\frac{\pi}{4} - \theta)$ what is in turn implied by $\theta \leq N^{-1}$ or equivalently by $T \geq \pi N$. In this case, the fundamental domain $[-\frac{1}{2}, \frac{1}{2})^2$ is covered by the set of cubes of side-length N^{-1} centred at such sampling nodes. However, using the slightly weaker condition $R \geq N$ still guarantees the covering within the unit disk and within the box $\{\boldsymbol{x} \in \mathbb{R}^2 : \|\boldsymbol{x}\|_\infty \leq \frac{1-N^{-1}}{2}\}$.

For the linogram grid a slightly simpler argument shows, that $R \geq N$ and $T \geq 2N$ already guarantees a dense sampling set, i.e., $\delta \leq N^{-1}$.

3.5 Choice of the weights

Weights are introduced in equation (2.3) to compensate for local sampling density variations. For every point in the sampling set, we associate a small surrounding area. In case of the polar grid, we choose small ring segments. The area of such a ring segment around $\boldsymbol{x}_{t,j}$ $(j \neq 0)$ is

$$w_{t,j} = \frac{\pi}{2TR^2}\left(\left(|j| + \frac{1}{2}\right)^2 - \left(|j| - \frac{1}{2}\right)^2\right) = \frac{\pi\,|j|}{TR^2}\,.$$

The area of the small circle of radius $\frac{1}{2R}$ around the origin is $\frac{\pi}{4R^2}$. Divided by the multiplicity of the origin in the sampling set, we get $w_{t,0} := \frac{\pi}{4TR^2}$.

Let us turn to the linogram grid, see also [16, 2]. For a point $\boldsymbol{x}_{t,j}^{\mathrm{BH}}$ $(j \neq 0$, analogous for a point $\boldsymbol{x}_{t,j}^{\mathrm{BV}})$ we use small surrounding trapezoids. Their area is equal to a corresponding rectangle and hence given by

$$w_{t,j}^{\mathrm{BH}} = \frac{1}{R} \cdot \frac{4\,|j|}{TR} = \frac{4\,|j|}{TR^2}\,.$$

Around the origin we have a small square of side length $\frac{1}{R}$, divided by the multiplicity of the origin in the sampling set, this weight is $w_{t,0}^{\mathrm{BH}} := \frac{1}{TR^2}$.

Remark 3.1 *Another possible choice for the weights associated to the nodes of the grids are Voronoi weights. However, our numerical tests showed that better results can be achieved with the analytical weights proposed here.* □

4 Numerical examples

The following numerical examples are computed with the NFFT C-subroutine library [12], where we choose the Kaiser-Bessel window functions with cut-off parameter m and oversampling factor $\sigma = 2$. We use the well known Shepp-Logan phantom of different sizes with values in $[0, 1]$ (Matlab-function `phantom(`N`)`), however, similar results can be obtained with arbitrary input images. We interpret the gray values of this image as Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ given on the grid I_N^2. The NFFT evaluates the corresponding trigonometric polynomial (2.1) at M arbitrary nodes in $\mathcal{O}((\sigma N)^2 \log(\sigma N)^2 + m^2 M)$ arithmetical operations.

Example 4.1 *In our first test, we compare the straightforward computation of the discrete (pseudo-)polar Fourier transform $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$ with the result of the NFFT, denoted by $\tilde{\boldsymbol{f}}$, on the polar grid, the modified polar grid, and the linogram grid for different values of the cut-off parameter m. We choose the phantom of size $N = 64$ and the sampling grids with $T = 3N$ and $R = \frac{3}{2}N$. As accuracy of the NFFT we take*

$$E_{\max} := \frac{\max_{t,j}|f_{t,j} - \tilde{f}_{t,j}|}{\max_{t,j}|f_{t,j}|}\,.$$

As expected, Figure 4.1 shows that these Fourier transforms can be computed very accurately. In particular, the achieved accuracy does hardly depend on the distribution of the grid points. □

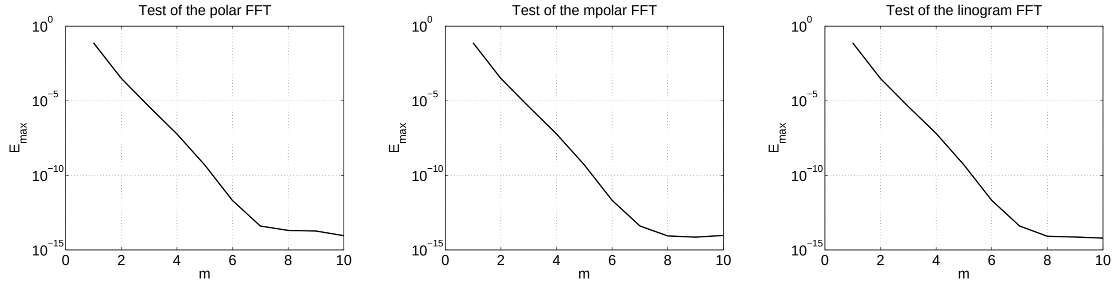

Figure 4.1: Left to right: Accuracy $E_{\max}$ for different cut-off parameter m for the polar, modified polar, and linogram grid of size $N = 64$ with $T = 3N$, $R = \frac{3}{2}N$.

Example 4.2 *Next, we compare the results of the inverse NFFT on the different grids and $N = 64$, i.e., the image size of the phantom is 64×64. The right hand side of (2.2) is computed with the slow exact transform. Then the reconstruction is done with our inverse NFFT and we obtain after the l-th iteration a reconstruction error*

$$\hat{E}_{\max} := \frac{\max_{\boldsymbol{k}\in I_N^2}|\hat{f}_{\boldsymbol{k}} - \hat{f}_{l,\boldsymbol{k}}|}{\max_{\boldsymbol{k}\in I_N^2}|\hat{f}_{\boldsymbol{k}}|}\,.$$

Here, the vector $\hat{\boldsymbol{f}}$ denotes the original image and $\hat{f}_{l,\boldsymbol{k}}$ denotes the $\boldsymbol{k}$-th entry of the l-th iterate within the CGNR method. Furthermore, different cut-off parameters m of the NFFT and its adjoint were used within the iterative scheme. This results in a limited final reconstruction quality. The same test is done for the phantom of size 256×256, where the right hand side of (2.2) is computed fast and very accurately by the NFFT ($m = 12$). The results are illustrated in Figure 4.2.

No convergence is achieved when using the polar grid, even for a large number of iterations. In contrast, a small number of iterations suffices to obtain very accurate inverse Fourier transforms from samples on the modified polar grid as well as from samples on the linogram grid, whereas the linogram grid performs slightly better. □

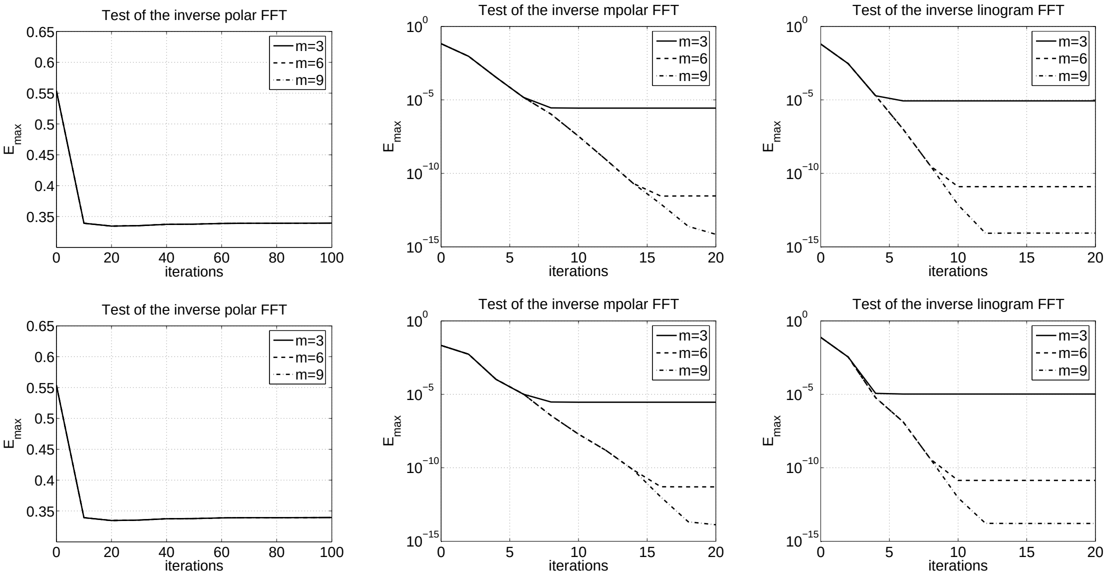

Figure 4.2: Reconstruction error $E_{\max}$ with respect to the number of iterations and for different cut-off parameter m. Top: Phantom of size $N = 64$, Bottom: $N = 256$, Grid sizes $T = 3N$, and $R = \frac{3}{2}N$. Left to right: polar, modified polar, and linogram grid.

Example 4.3 *Finally, we compare the computation time of the ordinary (Cartesian) FFT (computed by the FFTW package [9]), the discrete modified polar Fourier transform, i.e., the straightforward computation of the matrix vector product* $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$*, its fast realisation by the NFFT, and its inversion by the proposed iterative scheme. Note, that within the inverse transform, we choose the number of iterations equal to two times of the cut-off* m*, which is motivated by the final reconstruction error in Figure 4.2.*

The CPU time required by the four algorithms is shown in Table 4.1. As expected, both NFFT based algorithms show the same asymptotic performance as the ordinary FFT, i.e., doubling the size N *takes approximately* 4 *times longer. In comparison, the straightforward computation of* $\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$ *is much slower, in particular, doubling the size* N *takes approximately* 16 *times longer.*

Furthermore, for a fixed problem size, the computation time of the approximate NFFT schemes is $\mathcal{O}(|\log\varepsilon|^2)$ *for the fast forward transform and* $\mathcal{O}(|\log\varepsilon|^3)$ *for its inversion, whereas* $\varepsilon \to 0$ *denotes the target accuracy, see [17, 13] for details.* □

size N	FFTW	$\boldsymbol{f} = \boldsymbol{A}\hat{\boldsymbol{f}}$	cut-off m	mod. polar FFT	inv. mod. polar FFT
16	5.4e-06	7.3e-02	3	4.0e-03	5.9e-02
			6	1.1e-02	3.1e-01
			9	2.3e-02	9.0e-01
32	3.0e-05	1.1e+00	3	1.5e-02	2.3e-01
			6	4.3e-02	1.2e+00
			9	8.5e-02	3.5e+00
64	1.6e-04	1.7e+01	3	6.0e-02	9.1e-01
			6	1.7e-01	4.7e+00
			9	3.4e-01	1.4e+01
128	9.0e-04	2.7e+02	3	2.4e-01	3.7e+00
			6	6.8e-01	1.9e+01
			9	1.3e+00	5.4e+01
256	7.0e-03		3	9.9e-01	1.5e+01
			6	2.7e+00	7.6e+01
			9	5.4e+00	2.2e+02

Table 4.1: CPU-Time of the ordinary (Cartesian) FFT, the discrete modified polar Fourier transform, the NFFT based modified polar FFT, and its inversion. Note that we used accumulated measurements in case of small times.

5 Conclusion

We demonstrated that one can compute polar/pseudo-polar FFTs and their inverses very efficiently and accurately with our mature software package NFFT [12].

In contrast to [1], the accuracy of the forward transform is controlled by the cut-off parameter m, which allows for a small oversampling factor of the NFFT and hence for the non expansivity of the scheme. The achieved accuracy is theoretically guaranteed [17], whereas the mixed spline and Hermite-type interpolation scheme in [1] is only numerically tested. Furthermore, we applied a density argument from [10, 8] and contributed the missing explicit criteria, cf. Section '2.3 Fast Inverse Transform and Quasi-Parseval Relationship' in [1], for the considered sampling sets to allow for stable reconstruction.

Numerical experiments showed furthermore the accuracy of the NFFT and its inverse within the present setting. The computation time, although reasonable larger than for an ordinary FFT, is asymptotically optimal.

In our opinion, the discrete Fourier transform on the polar, the modified polar, and similar grids should simply be computed by a reliable approximate scheme as implemented in [12]. Asking for a highly accurate **pseudo**-polar FFT ($\varepsilon < 10^{-8}$), the non-approximate scheme based on the chirp-z transform [3, 2] or a set of $1d$ NFFTs [16] might indeed be favourable.

References

[1] A. Averbuch, R. Coifman, D. Donoho, M. Elad, and M. Israeli. Fast and accurate polar Fourier transform. *Appl. Comput. Harm. Anal.*, to appear.

[2] A. Averbuch, R. Coifman, D. Donoho, M. Israeli, and J. Walden. Fast slant stack: A notion of Radon transform for data in a cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. SIAM *J. Sci. Comput.*, to appear.

[3] D. Bailey and P. Swarztrauber. The fractional Fourier transform and applications. SIAM *Rev.*, 33:389 – 404, 1991.

[4] R. Bass and K. Gröchenig. Random sampling of multivariate trigonometric polynomials. SIAM *J. Math. Anal.*, 36:773 – 795, 2004.

[5] G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comput. Harmon. Anal.*, 2:363 – 381, 1995.

[6] E. Candes, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. SIAM *Multiscale Model. Simul.*, to appear.

[7] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. SIAM *J. Sci. Stat. Comput.*, 14:1368 – 1393, 1993.

[8] H. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numer. Math.*, 69:423 – 440, 1995.

[9] M. Frigo and S. G. Johnson. FFTW, a C subroutine library. http://www.fftw.org/.

[10] K. Gröchenig. A discrete theory of irregular sampling. *Lin. Alg. Appl.*, 193:129 – 150, 1993.

[11] 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.

[12] S. Kunis and D. Potts. NFFT, Softwarepackage, C subroutine library. http://www.math.uni-luebeck.de/potts/nfft, 2002 – 2005.

[13] S. Kunis and D. Potts. Time and memory requirements of the nonequispaced FFT. *Preprint, TU-Chemnitz*, 2006.

[14] J. Ma and M. Fenn. Combined complex ridgelet shrinkage and total variation minimization. SIAM *J. Sci. Comput.*, to appear.

[15] F. Natterer. *The Mathematics of Computerized Tomography.* John Wiley & Sons Ltd, 1986.

[16] D. Potts and G. Steidl. A new linogram algorithm for computerized tomography. IMA *J. Numer. Anal.*, 21:769 – 782, 2001.

[17] 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, 2001. Birkhäuser.

[18] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. *Adv. Comput. Math.*, 9:337 – 353, 1998.

[19] A. F. Ware. Fast approximate Fourier transforms for irregularly spaced data. SIAM *Rev.*, 40:838 – 856, 1998.