Non-equispaced fast Fourier transforms with applications to tomography

K. Fourmont

May 15, 2002

Abstract

In this paper we describe a non-equispaced fast Fourier transform based on a Fourier series representation of $\exp(-ix\xi)$. It is very similar to the approximate representations of Dutt and Rokhlin (1993) and Beylkin (1995) but it is exact. This greatly simplifies the analysis. Also, our representation is more flexible in that it involves a large class of weight functions rather than specific weights. This additional freedom can be used to achieve more efficiency. Accuracy and efficiency of the resulting algorithm are illustrated by numerical examples.

Then the non-equispaced FFT is applied to the reconstruction problem in Computerized Tomography. This results in a new view of the gridding method of O'Sullivan (1985) and in a new ultra fast reconstruction algorithm. It makes use of the fact that linear interpolation in angular direction is justified for Fourier reconstruction. Comparison of these algorithms with the filtered backprojection illustrates the improved efficiency and accuracy. The new reconstruction algorithm outperforms the filtered backprojection at least by a factor of 10.

1 Introduction

There has been much interest recently in doing the fast Fourier transform (FFT) on non-equispaced grids; see Dutt and Rokhlin (1993), Beylkin (1995), Steidl (1998) and the survey article of Ware (1998). Applications are abundant: Astronomy (Brouw (1975)), tomography (O'Sullivan (1985), Schomberg and Timmer (1995), Averbuch et al. (2000)), magnetic resonance imaging (MRI) (Twieg (1983)), and ultrasound (Kaveh and Soumekh (1987)).

In this paper we describe a non-equispaced FFT which is based on a representation of $\exp(-ix\xi)$ as a linear combination of the functions $\exp(-im\xi)$, $m \in N$, in the interval $|\xi| \leq \pi/c \leq \pi$. It is very similar to the approximate representations of Dutt and Rokhlin (1993) and Beylkin (1995) but it is exact. This greatly simplifies the analysis. Also, our representation is more flexible in that it involves a large class of weight functions while former works used specified weights. We use this additional freedom to achieve better efficiency.

The Fourier transform of equispaced data z_k evaluated at non-equispaced grid points $x_l \in [-N/2, N/2]$ can be written as

$$\hat{z}_\ell = \sum_{k=-N/2}^{N/2-1} e^{-2\pi i x_\ell k/N} z_k \ , \quad \ell = 1, \ldots, M \ . \tag{1.1}$$

This we will call the NER (non-equispaced results) case. On the other hand one can consider the Fourier transform of data sampled at non-equispaced points x_l evaluated on an equispaced grid. This we will call the NED (non-equispaced data) case:

$$\hat{z}_k = \sum_{\ell=1}^{M} e^{-2\pi i x_\ell k/N} z_\ell \ , \quad k = -N/2, \ldots, N/2 - 1 \ , \tag{1.2}$$

Obviously NER, NED are simply the transpose of each other. For $x_\ell = \ell, M = N$ we regain the usual equispaced Fourier transform of length N. The operation count of our algorithms for the evaluation of (1.1), (1.2) corresponds to 3 and 5 times the count of an equispaced FFT of the same length in single and double precision, respectively.

In the next section we derive the basic formulas for NED and NER. In section 3 we describe the algorithms to some detail and carry out numerical experiments. In section 4 we apply the 1D version of NER and the 2D version of NED to computerized tomography. In the latter case the resulting inversion algorithm for the Radon transform is essentially the gridding method.

2 Basic theory

All variants of non-equispaced Fourier algorithms utilize normal equispaced FFTs. So the task is to establish a connection between the non-equispaced Fourier nodes and its equispaced counterparts.
We start with Shannon's theorem for a bandlimited function f with bandwidth $< \pi$:

$$f(x) = \sum_{m \in \mathbb{Z}} \operatorname{sinc}\left(\pi(x-m)\right) f(m) \ . \tag{2.3}$$

With f chosen as $f(x) = e^{-ix\xi}$, $|\xi| < \pi$ this becomes

$$e^{-ix\xi} = \sum_{m \in \mathbb{Z}} \operatorname{sinc}\left(\pi(x-m)\right) e^{-im\xi}, \qquad |\xi| < \pi \ . \tag{2.4}$$

As the decay of the sinc function is not good enough for rapid computation of (2.4) the next obvious step is to use oversampling.

If we require f to have bandwidth $< \pi/c$, $c > 1$, one can introduce an interpolation function $\phi \in \mathcal{C}_0^\infty$ with support in $[-\pi, \pi]$. Unlike normal resampling scenarios we do not have to require ϕ to be 1 on $[-\pi/c, \pi/c]$; $\phi > 0$ on this interval will do. With this window function ϕ our Fourier expansion becomes

$$e^{-ix\xi} = \frac{(2\pi)^{-1/2}}{\phi(\xi)} \sum_{m \in \mathbb{Z}} \hat{\phi}(x-m) e^{-im\xi}, \qquad |\xi| < \pi/c\,. \tag{2.5}$$

However this is still not the best representation. For our special choice $f = e^{-ix\xi}$ equation (2.5) holds even if we extend the support of ϕ somewhat beyond $[-\pi, \pi]$.

The following proposition summarizes these observations:

Proposition 2.1 *Let $0 < \pi/c < \alpha$ and $\alpha < \pi(2 - 1/c)$. Let ϕ be continuous and piecewise continuously differentiable in $[-\alpha, \alpha]$, vanishing outside $[-\alpha, \alpha]$ and non-zero in $[-\pi/c, \pi/c]$. Then, for $x \in \mathbb{R}^1$ and $|\xi| \le \pi/c$,*

$$e^{-ix\xi} = \frac{(2\pi)^{-1/2}}{\phi(\xi)} \sum_{m \in \mathbb{Z}} \hat{\phi}(x-m) e^{-im\xi}\,.$$

Proof We define the 2π-periodic function

$$g(\xi) = \sum_{k \in \mathbb{Z}} \phi(\xi + 2k\pi) e^{-ix(\xi + 2k\pi)}\,. \tag{2.6}$$

Note that the support of ϕ is chosen so that

$$g(\xi) = \phi(\xi) e^{-ix\xi}, \qquad |\xi| \le \pi/c\,. \tag{2.7}$$

The Fourier expansion of g reads

$$\begin{aligned} g(\xi) &= \sum_{m \in \mathbb{Z}} \hat{g}_m e^{-im\xi}\,, \\ \hat{g}_m &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{im\xi} g(\xi) d\xi \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{im\xi} \sum_{k \in \mathbb{Z}} \phi(\xi + 2k\pi) e^{-ix(\xi+2k\pi)} d\xi \\ &= \frac{1}{2\pi} \int_{\mathbb{R}^1} e^{im\xi} \phi(\xi) e^{-ix\xi} d\xi \\ &= (2\pi)^{1/2} \hat{\phi}(x-m)\,. \end{aligned}$$

Thus,

$$g(\xi) = (2\pi)^{1/2} \sum_{m \in \mathbb{Z}} \hat{\phi}(x-m) e^{-im\xi} \tag{2.8}$$

with pointwise convergence in each point ξ of continuity. Since ϕ is continuous in $|\xi| \le \alpha$ and $\pi/c < \alpha$ we have pointwise convergence for $|\xi| \le \pi/c$. Combining (2.8), (2.7) and observing that $\phi(\xi) \neq 0$ for $|\xi| \le \pi/c$ finishes the proof.

□

Proposition 2.1 immediately leads to formulas which permit the efficient evaluation of (1.1), (1.2). We only have to put $\xi = 2\pi k/cN$, $x = cx_\ell$ to obtain for $|k| \le N/2$:

$$e^{-2\pi i x_\ell k/N} = \frac{(2\pi)^{-1/2}}{\phi(2\pi k/cN)} \sum_{m\in\mathbb{Z}} \hat{\phi}\left(cx_\ell - m\right) e^{-2\pi imk/cN} .$$

Inserting this into (1.1) we obtain the formula for the NER case:

$$\hat{z}_\ell = (2\pi)^{-1/2} \sum_{m\in\mathbb{Z}} \hat{\phi}\left(cx_\ell - m\right) \sum_{k=-N/2}^{N/2-1} e^{-2\pi imk/cN} \frac{z_k}{\phi(2\pi k/cN)} , \quad \ell = 1,\ldots,M. \tag{2.9}$$

Likewise we obtain for the NED case in (1.2)

$$\hat{z}_k = \frac{(2\pi)^{-1/2}}{\phi(2\pi k/cN)} \sum_{\ell=1}^{M} \sum_{m\in\mathbb{Z}} z_\ell \hat{\phi}\left(cx_\ell - m\right) e^{-2\pi imk/cN} , \quad k = -N/2,\ldots,N/2-1 . \tag{2.10}$$

The evaluation of (2.9), (2.10) requires equispaced FFTs of length cN. A convenient choice for the oversampling factor c is $c = 2$, however even $c = 3/2$ is sufficient to get good accuracy; see section 3 for details. α has to be chosen slightly smaller than $\pi(2 - 1/c)$.
The sum over m can be evaluated efficiently if $\hat{\phi}$ is small outside some interval $[-K, K]$. So we require our window ϕ to have compact support in $[-\alpha, \alpha]$ and its Fourier transform to be concentrated as much as possible in $[-K, K]$. Unfortunately the true solution to this problem, the prolate spheroidal wave functions (see Slepian (1983)) are too inconvenient to use. An approximate solution is the the Kaiser-Bessel window; see Kaiser (1966):

$$\hat{\phi}(x) = \sqrt{\frac{2}{\pi}} \frac{\sinh(\alpha\sqrt{K^2 - x^2})}{\sqrt{K^2 - x^2}} , \tag{2.11}$$

$$\phi(\xi) = \begin{cases} I_0\left(K\sqrt{\alpha^2 - \xi^2}\right) & , \quad |\xi| \le \alpha , \\ 0 & , \quad |\xi| > \alpha . \end{cases} \tag{2.12}$$

The periodic extension (2.6) in the proof of Proposition 2.1 can be used to improve the error estimate of the method of Dutt and Rokhlin (see Dutt and Rokhlin (1993)). They present an approach which is very similar to ours but use Gaussian bells $\phi(x) = e^{-x^2/4b}$ as window functions. They determine the window width parameter b by numerical experiments and propose $b = 0.6$ for single precision. The Gaussian bells are not compactly supported, resulting in additional error terms. They do not use the periodification of (2.6) and therefore all parts of $\phi(x)$ with $|x| \geq \pi$ contribute to the error estimate. As a result the error estimate is roughly proportional to $\phi(\pi) \approx 0.003$ for $b = 0.6$, which is much bigger than single machine precision. Applying the trick of (2.6) in their derivation, it is easily seen that only the parts of $\phi(x)$ with $|x| \geq \pi(2 - 1/c)$ contribute to the error. Hence the error is of magnitude $\phi(\pi(2 - 1/c)) \approx 1.6 \cdot 10^{-6}$. This drastically improves the error estimate and also explains the choice of b. The lack of such an explanation has been mentioned in the survey article of Ware (1998). The solution has been given by Steidl (1998). She uses a frequency based view resulting in correct periodification and suitable error estimates for the method of Dutt and Rokhlin.

3 Implementation

To get an implementation that can be compared with standard FFT implementations, all calculations depending only on the distribution of the nodes have to be precomputed and stored.
First we have to choose the oversampling factor c and the length of the interpolation K. A convenient choice is $c = 2$ and $K = 3, 6$ for single, double precision resp. We can assume $N \gg K$, e.g. $N \geq 32$ as we do not need a fast Fourier transform for very small N.
The required values of the weighting function ϕ are stored in a vector ϕ_k:

$$\phi_k = \phi(2\pi k/cN) \qquad k = -N/2, \ldots, N/2 - 1 \tag{3.13}$$

The width α of the window function ϕ has to be chosen slightly smaller than $\pi(2 - 1/c)$, $\alpha = \pi(2 - 1/c) - 0.01$ will do.
Next we define a vector $\mu_k, k = 1, \ldots, M$, where μ_k is the nearest integer to cx_k. This is the nearest equispaced node to x_k. Finally, we precompute the required values of $\hat{\phi}$ that we will use as interpolation coefficients:

$$\hat{\phi}_{\ell m} = \frac{1}{\sqrt{2\pi}} \hat{\phi}\left(cx_\ell - (\mu_\ell + m)\right), \qquad \begin{matrix} l = 1, \ldots, M, \\ m = -K, \ldots, K \end{matrix} \tag{3.14}$$

With these definitions, 2.9 reads

$$\hat{z}_\ell = \sum_{|m| \leq K} \hat{\phi}_{\ell m} \sum_{k=-N/2}^{N/2-1} e^{-2\pi i m k/cN} \frac{z_k}{\phi_k}, \qquad \ell = 1, \ldots, M.$$

We have truncated the sum over m so as to contain only values for $\hat{\phi}$ in $[-K, K]$. See section 4 for an error estimate. The algorithm consists of three steps:

Algorithm: Fast computation of the NER problem

$$\hat{z}_\ell = \sum_{k=-N/2}^{N/2-1} e^{-2\pi i x_\ell k/N} z_k \ , \quad \ell = 1, \ldots, M \ .$$

Parameters: oversampling factor c with cN FFT-friendly, interpolation length K, precomputed values $\phi_k, \hat{\phi}_{\ell j}, \mu$.

Data: The input data z_k, $k = -N/2, \ldots, N/2-1$.

Step 1: Scaling and zero padding:

$$u_k = \begin{cases} 0 & k = -cN/2, \ldots, -N/2-1 \\ z_k/\phi_k & k = -N/2, \ldots, N/2-1 \\ 0 & k = N/2, \ldots, cN/2-1 \end{cases}$$

($2N$ flops)

Step 2: compute

$$U_j = \sum_{k=-cN/2}^{cN/2-1} u_k \cdot e^{-2\pi i k j/cN}, \qquad j = -cN/2, \ldots, cN/2-1$$

using a FFT of length cN.
($\mathcal{O}(cN \log cN)$ flops)

Step 3: Interpolation to the nonuniform grid:

$$\hat{z}_\ell = \sum_{j=-K}^{K} \hat{\phi}_{\ell j} U_{\mu_\ell + j}, \qquad \ell = 1, \ldots, M \tag{3.15}$$

The index $\mu_\ell + j$ has to be taken cN-periodic.
($8KM$ flops)

Note that we compute a symmetric Fourier transform in step 2, whereas usually FFT software computes the zero based form:

$$\hat{z}_k = \sum_{j=0}^{N-1} e^{-2\pi i j k/N} z_k. \tag{3.16}$$

But this is merely a matter of notation. z as well as $\hat{z}$ can be considered cN-periodic and hence one can use a FFT software of type (3.16) to compute symmetric FFTs by just interchanging the first and second half of the data vector before as well as after execution of the standard FFT routine.

For the NED case, the formula is slightly less obvious. We start with the same notation as for the NER case, converting (2.10) into

$$\hat{z}_k = \frac{1}{\phi_k} \sum_{\ell=1}^{M} \sum_{m=-K}^{K} z_\ell \hat{\phi}_{\ell m} e^{-2\pi i k(\mu_\ell + m)/cN} \tag{3.17}$$

If we assume $\hat{\phi}$, μ, and z to be zero outside their area of definition and with a new index $j = \mu_\ell + m$ we get

$$\begin{aligned}
\hat{z}_k &= \frac{1}{\phi_k} \sum_{j\in\mathbb{Z}} \sum_{l\in\mathbb{Z}} z_\ell \hat{\phi}_{\ell,j-\mu_\ell} e^{-2\pi ikj/cN} && (3.18)\\
&= \frac{1}{\phi_k} \sum_{j=-cN/2}^{cN/2-1} \sum_{l\in\mathbb{Z}} \sum_{m\in\mathbb{Z}} z_\ell \hat{\phi}_{\ell,j+cmN-\mu_\ell} e^{-2\pi ik(j+cmN)/cN} && (3.19)\\
&= \frac{1}{\phi_k} \sum_{j=-cN/2}^{cN/2-1} u_j e^{-2\pi ikj/cN} && (3.20)
\end{aligned}$$

with

$$u_j = \sum_l \sum_m z_\ell \hat{\phi}_{\ell,j+cmN-\mu_\ell}, \quad j = -cN/2, \ldots, cN/2-1 \ . \qquad (3.21)$$

Nonzero terms occur only for $|j + cmN - \mu_\ell| \leq K \ll cN$. Thus each u_j gets contributions from all z_ℓ with distance $\leq K$; the distance is computed modulo cN.

Algorithm: Fast computation of the NED problem

$$\hat{z}_k = \sum_{\ell=1}^{M} z_\ell e^{-2\pi i x_\ell k/N} \qquad k = -N/2, \ldots, N/2-1$$

Parameters: oversampling factor c with cN FFT-friendly, interpolation length K, precomputed values $\phi_k, \hat{\phi}_{\ell j}, \mu_k$.

Data: The non-equispaced input data z_k, $k = 1, \ldots, M$.

Step 1: Evaluate (3.21):
$u := 0$
for $k = 1, \ldots, M$:
for $l = -K, \ldots, K$:

$$u_{\mu_k+l} \leftarrow u_{\mu_k+l} + \hat{\phi}_{kl} \cdot z_k \qquad (3.22)$$

Each z_k contributes to the nearest $2K+1$ equispaced nodes. Again, the index $\mu_k + l$ has to be taken cN-periodic.
($8MK$ flops)

Step 2: Compute the FFT of size cN

$$U_j = \sum_{k=-cN/2}^{cN/2-1} u_k \cdot e^{2\pi ikj/cN} \qquad , j = -N/2, \ldots, N/2-1$$

($\mathcal{O}(cN \log cN)$ flops)

Step 3: Finally, scale the result:

$$\hat{z}_j = U_j/\phi_j \qquad , j = -N/2, \ldots, N/2-1$$

($2N$ flops)

The algorithms easily extend to more than one dimension. For example, the two-dimensional NED problem looks like this:

$$\hat{z}_{kl} = \sum_{j=1}^{M} z_j e^{2\pi i k x_j/N} e^{2\pi i l y_j/N}, \qquad k, l = -N/2, \ldots, N/2-1 \tag{3.23}$$

Note that the arbitrary nodes (x_j, y_j) are not necessarily a tensor product of two coordinates and therefore cannot be written using two indices for the two dimensions. Instead we use a one-dimensional vector which contains all nodes. So the number of non-equispaced nodes M is usually of magnitude N^2.
The weighting function $\phi(x,y)$ is chosen as a tensor product $\phi(x)\cdot\phi(y)$. Thus the interpolation step uses K^2 terms, increasing the necessity to get by with small values of K.

To get an idea of the operation count, we assume the one-dimensional FFT of size N with $4.25N\log N$ flops. This is true for a standard Cooley Tukey radix 4 FFT if N is a power of two.
If we assume the typical case $M = N$, the operation count for the non-equispaced fast Fourier transform reads:

$$\mathrm{flops}_{1D}(N,c,K) = 4.25cN\log_2(cN) + (8K+2)N + \mathcal{O}(1), \tag{3.24}$$

To get an idea about the actual runtime, we compare this to the operation count for an equispaced FFT of the same size, yielding

$$N_1(N,c,K) = \frac{\mathrm{flops}_{1D}(N,c,K)}{4.25N\log_2 N} = c + \frac{4.25\log_2 c + (8K+2)}{4.25\log_2 N}. \tag{3.25}$$

Figure 1 displays this factor N_1 for different values of N, c and K.
The same can be done for the two-dimensional case. For $M = N^2$ the operation count is

$$\mathrm{flops}_{2D}(N,c,K) = 9c^2N^2\log_2(cN) + 20K^2N^2 + \mathcal{O}(1) \tag{3.26}$$

which results in a factor of

$$N_2(N,c,K) = c^2 + \frac{8.5c^2\log_2 c + (20K^2+3)}{8.5\log_2 N}. \tag{3.27}$$

Typical values for N_2 are displayed in figure 2.
We implemented one- and two-dimensional versions of our algorithm in C and tested them on a 300MHz Sun UltraSPRAC-II. The implementation can be found at `http://www.math.uni-muenster.de/num/fourmont/`.

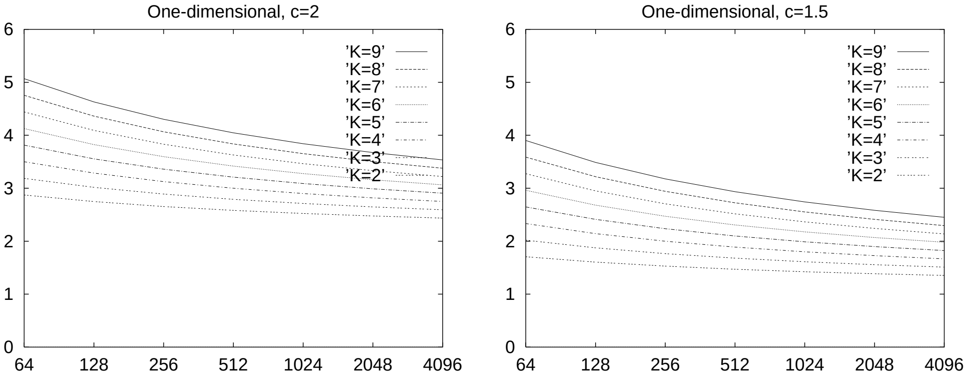

Figure 1: Factor non-equispaced/equispaced 1D-FFT

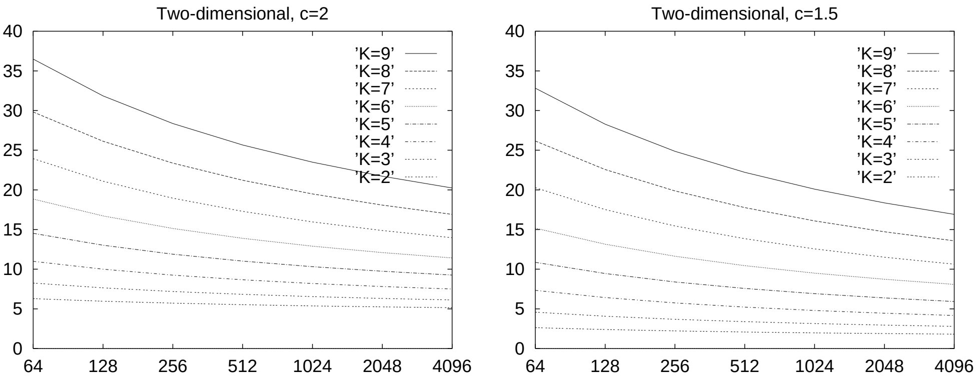

Figure 2: Factor non-equispaced/equispaced 2D-FFT

To verify the accuracy, we use the following error norms:

$$E_\infty = \frac{\max_j |\bar{z}_j - z_j|}{\max_j |\bar{z}_j|} \qquad \text{and} \qquad E_2 = \frac{\sum_j |\bar{z}_j - z_j|^2}{\sum_j |\bar{z}_j|^2},$$

where $\bar{z}$ is the result using direct computation and z is the result of our algorithm. One has to keep in mind that $\bar{z}$ itself is computed with limited accuracy and therefore we only get a rough estimate for the actual error. The input data were generated randomly on the unit square in the complex plane, defined by

$$0 \le \mathrm{Re}(z_j) \le 1, \qquad 0 \le \mathrm{Im}(z_j) \le 1.$$

The nonequispaced nodes were distributed randomly in $[-N, N]$. The IMSL software package has been used to compute the equispaced FFTs. It supplies single as well as double precision versions of the FFT.
Tables 1 to 4 display the results of the runtime measurements for the one- and two-dimensional case, both in single and double precision. *Eval* indicates the runtime for the evaluation of our algorithm, *Direct Factor* the speedup

compared to direct evaluation. *FFT* is given as a reference and indicates the runtime for an equispaced FFT. Finally *FFT-Factor* displays the factor of non-equispaced and equispaced FFT timings. These values can be compared to the analytic derivations in (3.25), (3.27) resp.
The algorithm of Dutt and Rokhlin is very similar to ours except that it uses Gaussian bells as weight functions. In their article they use oversampling of factor 2 and a total interpolation length of 10, 28 for single, double precision resp. In our notation this corresponds to values of $c = 2$ and $K = 5, 14$ resp. Our algorithm achieves the same accuracy with $c = 1.5$ and $K = 4, 7$ resp. This results in a significant speedup especially in two dimensions.

NE	N	E_∞	E_2	Eval msecs	Direct Factor	FFT msecs	FFT-Factor
ner	128	6.81e-07	1.17e-06	0.113	64	0.050	2.2
ner	256	2.43e-06	1.73e-06	0.210	124	0.071	3.0
ner	512	1.34e-06	1.33e-06	0.425	251	0.129	3.3
ner	1024	6.02e-07	1.03e-06	0.838	504	0.272	3.1
ner	2048	5.15e-07	8.73e-07	1.693	956	0.561	3.0
ner	4096	3.49e-07	8.12e-07	3.798	1822	1.278	3.0
ned	128	4.35e-07	8.04e-07	0.127	54	0.050	2.5
ned	256	4.20e-07	7.09e-07	0.241	108	0.071	3.4
ned	512	2.81e-07	6.31e-07	0.480	225	0.129	3.7
ned	1024	1.92e-07	6.55e-07	0.953	440	0.273	3.5
ned	2048	1.25e-07	6.61e-07	2.000	837	0.601	3.3
ned	4096	4.45e-07	7.64e-07	4.222	1628	1.228	3.4

Table 1: **One-dimensional Non-equispaced Fourier Transform, Single Precision,** $K = 4$**,** $c = 1.5$

NE	N	E_∞	E_2	Eval msecs	Direct Factor	FFT msecs	FFT-Factor
ner	128	4.50e-11	3.20e-11	0.152	47	0.051	3.0
ner	256	2.46e-11	2.36e-11	0.297	103	0.075	3.9
ner	512	7.78e-12	1.43e-11	0.564	211	0.161	3.5
ner	1024	3.37e-12	1.34e-11	1.133	400	0.351	3.2
ner	2048	4.33e-12	1.17e-11	2.348	783	0.698	3.4
ner	4096	3.11e-12	1.28e-11	5.036	1445	1.402	3.6
ned	128	6.96e-12	1.15e-11	0.192	37	0.050	3.8
ned	256	6.14e-12	1.13e-11	0.383	79	0.075	5.1
ned	512	3.85e-12	8.97e-12	0.746	159	0.161	4.6
ned	1024	5.52e-12	1.05e-11	1.494	301	0.349	4.3
ned	2048	4.13e-12	1.03e-11	3.098	600	0.693	4.5
ned	4096	8.34e-12	1.27e-11	6.493	1121	1.404	4.6

Table 2: **One-Dimensional Non-equispaced Fourier Transform, Double Precision, $K = 7$, $c = 1.5$**

NE	N	E_∞	E_2	Eval secs	Direct Factor	FFT secs	FFT-Factor
ner	64	3.45e-07	9.91e-07	0.029	1161	0.01	4.1
ner	128	3.29e-07	7.86e-07	0.125	4311	0.03	4.6
ner	256	5.74e-07	3.17e-06	0.567	14751	0.11	5.1
ner	512	5.05e-06	8.91e-06	2.947	50361	0.51	5.4
ner	1024	1.69e-07	4.04e-06	13.663	173800	2.51	5.5
ned	64	1.54e-07	1.36e-06	0.034	991	0.01	4.9
ned	128	1.83e-07	2.14e-06	0.141	3822	0.03	5.4
ned	256	1.88e-07	3.85e-06	0.643	13008	0.11	5.7
ned	512	2.07e-07	7.45e-06	3.157	47011	0.51	6.2
ned	1024	2.43e-07	8.21e-06	14.806	160383	2.51	5.9

Table 3: **Two-Dimensional Non-equispaced Fourier Transform, Single Precision, $K = 4$, $c = 1.5$**

NE	N	E_∞	E_2	Eval secs	Direct Factor	FFT sec	FFT-Factor
ner	64	5.98e-12	1.54e-11	0.066	577	0.01	9.4
ner	128	4.52e-12	1.69e-11	0.293	2230	0.03	9.5
ner	256	1.47e-11	2.35e-11	1.418	7068	0.14	10.4
ner	512	7.92e-11	3.12e-11	7.907	22262	0.76	10.4
ner	1024	4.50e-12	7.49e-12	34.332	84082	3.09	11.2
ned	64	3.07e-12	1.43e-11	0.077	586	0.01	9.6
ned	128	1.46e-12	1.40e-11	0.324	1960	0.03	10.5
ned	256	9.62e-13	1.40e-11	1.533	6517	0.14	11.1
ned	512	6.85e-13	1.39e-11	8.512	20977	0.76	11.2
ned	1024	2.94e-12	1.45e-11	36.539	77497	3.09	11.9

Table 4: **Two-Dimensional Non-equispaced Fourier Transform, Double Precision,** $K = 7$**,** $c = 1.5$

4 Error Estimations

In Proposition 2.1 we presented a Fourier series expansion of the exponential function $\exp(-ix\xi)$. The implementation as described in section 3 uses a truncated version of this expansion. In this section we give an estimate for the resulting truncation error when using the Kaiser-Bessel Window of (2.11),(2.12).

Proposition 4.1 *With the same assumptions as in Proposition 2.1 and ϕ,$\hat{\phi}$ as in (2.11),(2.12)*

$$\left| e^{-ix\xi} - \frac{(2\pi)^{-1/2}}{\phi(\xi)} \sum_{|m-x|\leq K} \hat{\phi}(x-m)e^{-im\xi} \right| \leq \frac{30}{\pi I_0(K\pi\sqrt{\alpha^2 - p^{-2}})}$$

holds for $K \leq 15$.

Proof Due to Proposition 2.1, the approximation error is

$$\frac{(2\pi)^{-1/2}}{\phi(\xi)} \left| \sum_{|m-x|>K} \hat{\phi}(x-m)e^{-im\xi} \right| \tag{4.28}$$

For these arguments, numerator and denominator of $\hat{\phi}$ become purly imaginary, resulting in

$$\hat{\phi}(\ell) = \sqrt{\frac{2}{\pi}} \frac{\sin(\alpha\sqrt{\ell^2 - K^2})}{\sqrt{\ell^2 - K^2}}.$$

For large values of l this behaves like $\sin(\ell)/\ell$. Unfortunately summation over $\sin(\ell)/\ell$ does not converge absolutely resulting in somewhat tedious estimations. We refer to Fourmont (1999) for the details and only present the result:

Lemma 4.1 *Let $1 < K \leq 15$ and $\alpha \leq 2\pi$. Then*

$$\left| \sum_{|l|>K} \frac{\sin(\alpha\sqrt{\ell^2 - K^2})}{\sqrt{\ell^2 - K^2}} \cdot e^{itl} \right| < 30$$

for all $t \in \mathbb{R}$.

The proof of Proposition 4.1 is finished by putting this into (4.28) and observing that ϕ is minimized by choosing $\xi = \pi/c$.

$\square$

Figure 3 shows the values of this error estimation as well as some directly computed errors. IEEE single and double machine precision have been marked. Roughly speaking, increasing K by one increases the approximation accuracy by two digits. Therefore it is not necessary to use oversampling factors greater than 2 because this would degrade efficiency while giving only moderate accuracy improvements.

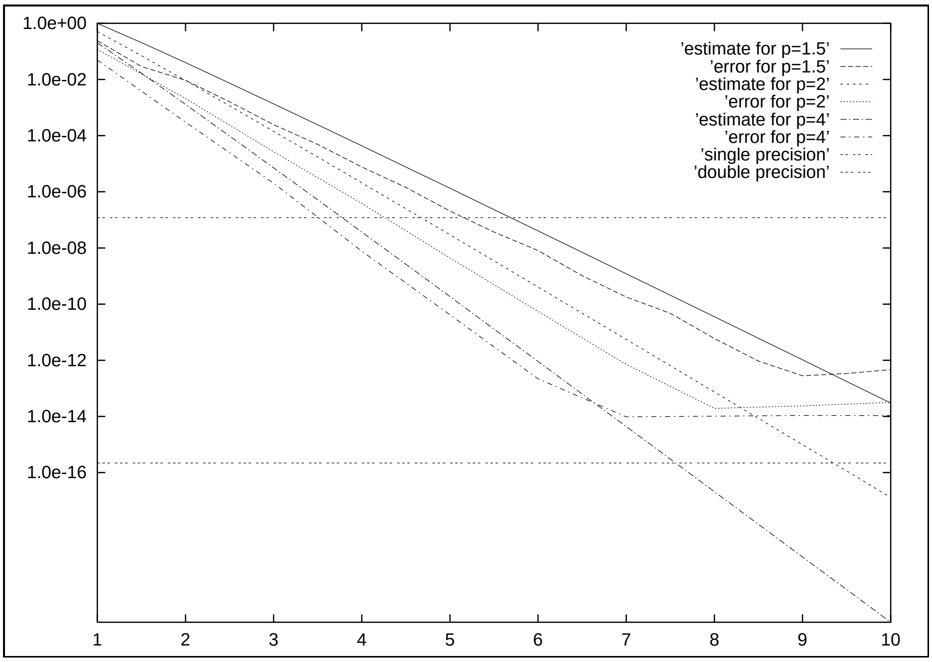

Figure 3: approximation error and error estimate for oversampling $c = 1.5; 2; 4$ and different values of K.

5 Application to tomography

In this chapter we apply the nonequispaced Fourier transform to computerized tomograpy. In tomography one reconstructs a function in $\mathbb{R}^2$ from its line integrals; see Herman (1980), Natterer (1986). More specifically, let f be a function supported in $|x| < 1$ and let

$$(Rf)(\theta, s) = \int\limits_{x\cdot\theta = s} f(x)dx \ , \qquad \theta \in S^1, \ s \in \mathbb{R}^1$$

be its Radon transform. Then the problem is to recover f from samples of the function $g = Rf$. We assume $g(\theta, s)$ to be sampled at $\theta = \theta(\varphi_j)$ and $s = s_\ell$ where

$$\begin{aligned} \theta(\varphi) &= \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} , \quad \varphi_j = \frac{\pi j}{p} \ , \quad j = 0, \ldots, p-1 \ , \\ s_\ell &= \ell/q \quad , \ \ell = -q, \ldots, q \ . \end{aligned} \tag{5.29}$$

In the language of tomography this corresponds to standard parallel scanning. The sampled values $g_{j,\ell} = g(\theta_j, s_\ell)$ permit the reliable reconstruction of a function f with essential bandwidth Ω (i.e. $\hat{f}(\xi)$ negligible for $|\xi| > \Omega$) provided that $p \geq \Omega$, $q \geq \pi/\Omega$; see Natterer (1986).

The use of Fourier transforms for the reconstruction of f is based on the "Projection Theorem"

$$\hat{g}(\theta, s) = (2\pi)^{1/2} \hat{f}(\sigma\theta) . \tag{5.30}$$

Here, $\hat{g}$ is the 1D Fourier transform of g with respect to the second variable, and $\hat{f}$ is the 2D Fourier transform of f. For the proof of (5.30) see e.g. Natterer (1986).

We describe two ways of using non-equispaced FFTs in Fourier reconstruction, the first one using one 2D NED, the second one p 1D NERs. The first approach is basically the gridding method by O'Sullivan while the second one is a new approach that is even faster than gridding while still producing good reconstruction quality.
We design the algorithms so as to reconstruct reliably functions of essential bandwidth Ω from the data (5.29).

5.1 Gridding

If we introduce polar coordinates in the Fourier inversion integral, we obtain

$$\begin{aligned} f(x) &= \frac{1}{2\pi} \int\limits_{\mathrm{R}^2} e^{ix\cdot\xi} \hat{f}(\xi) d\xi \\ &= \frac{1}{2\pi} \int\limits_0^\infty \sigma \int\limits_{S^1} e^{ix\cdot\sigma\theta} \hat{f}(\sigma\theta) d\theta d\sigma \\ &= (2\pi)^{-3/2} \int\limits_0^\infty \sigma \int\limits_{S^1} e^{ix\cdot\sigma\theta} \hat{g}(\theta, \sigma) d\theta d\sigma . \end{aligned} \tag{5.31}$$

For the success of the method it is crucial to discretize (5.31) properly.
For $x = \frac{1}{q}k$, $k \in \mathbb{Z}^2$ the discrete version reads

$$f\left(\frac{1}{q}k\right) = \frac{\pi^2/pd}{(2\pi)^{3/2}} \sum_{\ell=0}^{Q} \sum_{j=0}^{2p-1} e^{i\pi k\cdot\xi_{j,\ell}/q} \sigma_\ell \hat{g}(\theta_j, \pi\ell/d) \tag{5.32}$$

where $\xi_{j,\ell} = \frac{1}{d}\ell\theta_j$ and d is an oversampling factor. The range of integration for the σ-integral in 5.31 is $[0, \infty]$ rather than $[-\infty, \infty]$. One has to compensate for the effects of truncating the range of integration at zero. We have chosen $\sigma_0 = 1/10$, $\sigma_1 = 0.98$, $\sigma_\ell = \ell$ for $\ell > 1$ and are satisfied with the results. However a more detailed analysis is outstanding. They might be interpreted as some kind of end point correction for oscillating integrals as proposed in Stoer (1980) and Press, Teukolsky, Vetterling, Flannery (1988).
(5.32) is a 2D NED problem with non-equispaced nodes $\xi_{j,l}$ and can be evaluated using the algorithm presented in section 2.

If we insert the 2D version of (2.10) into (5.32) the result is essentially the gridding method; see Brouw (1975), O'Sullivan (1985), Kaveh and Soumekh (1987), Schomberg and Timmer (1995). Only the sampling requirements are different (in fact more favourable) than in the usual gridding method.
Our version of the gridding algorithm can be described as follows:

Algorithm: Gridding method for standard parallel geometry using 2D NED.

Parameters: Scanning geometry p, q, oversampling factor d, parameters for 2D NED

Data: The values $g_{j,\ell} = g(\theta_j, s_\ell)$, $j = 0, \ldots, p-1$, $\ell = -q, \ldots, q$ with $g = Rf$.

Step 1: For $j = 0, \ldots, p-1$ carry out the discrete Fourier transform

$$\hat{g}_{j,r} = \sum_{\ell=-q}^{q} e^{-i\pi\ell r/Q} \,, \quad r = -Q, \ldots, Q-1$$

of length $2Q$ where $Q = dq$. Extend $\hat{g}_{j,r}$ to $j = 0, \ldots, 2p-1$ by $\hat{g}_{j+p,r} = \hat{g}_{j,-r}$. $\hat{g}_{j,r}$ is an approximation to $\hat{g}(\theta_j, \pi r d)$.

Step 2: Apply filtering and scaling

$$\hat{g}_{j,r} \leftarrow \frac{\pi\sigma_r}{4qpd} F(r/Q)\hat{g}_{j,r} \qquad \begin{array}{c} j = 0, \ldots, p-1 \\ r = -Q, \ldots, Q-1 \end{array}$$

Here σ_r performs ramp filtering while $F(x)$ is an additional smoothing filter.

Step 3: Use 2D NED to compute

$$f_k = \sum_{r,l} e^{i\pi k\cdot x_{r,l}/Q} \hat{g}_{r,l} \qquad , |k| < q$$

using the non-equispaced nodes

$$x_{r,l} = r \begin{pmatrix} \cos(\pi l/p) \\ \sin(\pi l/p) \end{pmatrix} \qquad \begin{array}{c} l = 0, \ldots, p-1 \\ r = -Q, \ldots, Q-1 \end{array}.$$

Note that we use two indices for the non-equispaced data while our formulation for the NED algorithm uses only one index.

5.2 Fast Fourier Reconstruction

The gridding algorithm uses a non-equispaced Fourier transform for the two-dimensional inverse Fourier transform. Our own approach, the *Fast Fourier Reconstruction* uses a non-equispaced Fourier transform for the one-dimensional Fourier transform instead.

It makes use of the fact that angular interpolation in the polar coordinate grid is justified; see Natterer (1985). In fact linear interpolation suffices. Thus we can move each point πk, $k \in \mathbb{Z}^2$ of the Cartesian grid to a point $\pi|k|\theta_j$ with θ_j suitably chosen, leaving us with the problem of computing

$$\hat{f}\,(\pi|k|\theta_j) \sim (2\pi)^{-1}\frac{1}{q}\sum_{\ell=-q}^{q-1} e^{-i\pi|k|\ell/q} g_{j,\ell}\ , \quad k \in \mathbb{Z}^2\ , \quad |k| \leq q\ .$$

This is an instance of a non-equispaced FFT of type NER.
The interpolation scheme is displayed in figure 4.
For the algorithm to perform efficiently one has to precompute the coefficients used in the linear interpolation. For each Cartesian grid point $k \in \mathbb{Z}^2$, $|k| \leq dq$

- denote the two beams next to k by l_k and $l_k + 1$
- Add $|k|$ to the lists $\{x^{\ell_k}\}$,$\{x^{\ell_k+1}\}$ of evaluation points for the respective one-dimensional non-equispaced Fourier transforms.
- Compute the coefficients a_k b_k for the linear interpolation.

$$\begin{aligned} a_k &= F(|k|/dq)\cdot|\varphi_k - l_x\pi/p|\cdot p/\pi \\ b_k &= F(|k|/dq)\cdot|\varphi_k - (l_x+1)\pi/p|\cdot p/\pi \end{aligned}$$

Here $\varphi_k = \arctan(k_2/k_1)$ and $F(x)$ is a suitable smoothing filter.

With these precomputations the reconstruction can be performed very fast:

Algorithm: Fast Fourier reconstruction using 1D NER.

Parameters: Scanning geometry p, q, oversampling factor d, parameters l_k,$\{x^{l_k}\}$, a_k, b_k as defined above

Data: The values $g_{j,\ell} = g(\theta_j, s_\ell)$, $j = 0, \ldots, p-1$, $\ell = -q, \ldots, q$ with $g = Rf$.

Step 1: For $j = 0, \ldots, p-1$ compute

$$\hat{g}_{jl} = \sum_{k=-q}^{q} g_{kl} e^{i\pi k x_j^l/q} \quad l = -dq, \ldots, dq-1$$

using p instances of the 1D NER algorithm.

Step 2: For each point of the cartesian grid $k \in \mathbb{Z}^2$, $|k| \leq q$ compute $\hat{f}_k$ using linear interpolation in the angular direction:

$$\hat{f}_k = a_k \hat{g}_{\ell_k, j_k^1} + b_k \hat{g}_{\ell_k+1, j_k^2}$$

j_k^1 and j_k^2 are the indices of the non-equispaced nodes belonging to this cartesian point. a_k and b_k are the coefficients for the linear interpolation, l_k and $l_k + 1$ the indices for the two beams closest to k.

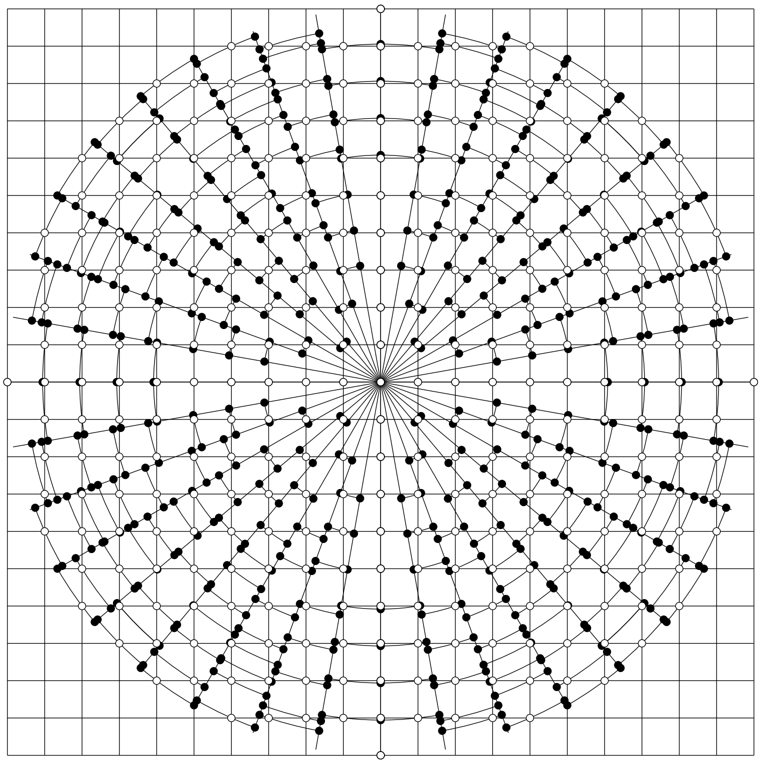

Figure 4: Interpolation scheme for fast Fourier reconstruction using one-dimensional NER

Step 3: compute

$$f_\ell = \frac{1}{2dq} \sum_{|k|<dq} \hat{f}_k e^{\pi i k \cdot l/Q} \quad |l| < q$$

using a Standard 2D inverse FFT.

Both reconstructions algorithms were implemented in C and tested on a 300MHz Sun UltraSPRAC-II. The implementations make use of the fact that the input data as well as the result is real. Hence one can restrict the complex computations to the upper half plane reducing the computational load by nearly 50%. To carry out the FFTs the FFTW software package (`www.fftw.org`) has been used. It provides the required real-to-complex and complex-to-real FFT routines.
Table 5 displays the result of runtime measurements for the Fourier reconstruction algorithms. p and q determine the scanning geometry, the reconstruction size is $2q \times 2q$. d is the oversampling factor used in step 1 of the algorithm while c is the oversampling factor used to compute the non-equispaced Fourier transforms. Thus the total oversampling factor is $c \cdot d$. The choice $c{\cdot}d = 2$ is more efficient than $c{\cdot}d = 3$ but produces slight artefacts for input data which is not properly bandlimited.

To compare the efficiency of our algorithms we used a standard filtered backprojection algorithm. Our implementation did not utilize FFT techniques to perform the FBP filtering step. Therefore the timings are displayed for the pure backprojection part to give a fair comparison. However the speedup is still impressive. Reconstructing an image with $p = 400, q = 128$ takes more than 6 seconds using filtered backprojection while Fast Fourier Reconstruction takes only 1/15th of a second. This is a speedup factor of almost 100.

	p	q	d	c	time [msec] BP	FFR	speedup BP/FFR
NER	200	64	2	3/2	781	52	15
NER	400	128	2	3/2	6295	238	26
NER	800	256	2	3/2	50870	1143	44
NER	200	64	1	3/2	781	15	52
NER	400	128	1	3/2	6295	67	94
NER	800	256	1	3/2	50870	308	165
NED	200	64	2	3/2	781	124	6
NED	400	128	2	3/2	6295	599	10
NED	800	256	2	3/2	50870	2967	17
NED	200	64	3/2	4/3	781	67	11
NED	400	128	3/2	4/3	6295	332	19
NED	800	256	3/2	4/3	50870	1779	28

Table 5: runtime measurements for Fast Fourier Reconstruction using one-dimensional NER

To verify the accuracy of the reconstruction, we used two computer generated phantoms. The first phantom is the well known Shepp Logan phantom. The second phantom consists of a high-density torus with smaller objects inside. The torus differs from the background by 1000 Houndsfield units, the small slits by 25 units. From both phantoms, line integrals have been computed analytically. To simulate real CT data, we averaged over 20 line integrals for each detector. We used 400 directions and 257 measurements per direction. This corresponds to parameters $p = 400, q = 128$. Oversampling factors have been chosen as $c = 1.5$ and $d = 2$ for the Fourier reconstruction algorithm s. The reconstructions are displayed in Figures 5 and 6. For filtered backprojection, NED Fourier reconstruction (Gridding) and Fast Fourier Reconstruction. The backprojection has been filtered with a Shepp Logan filter while for the Fourier reconstructions a sinc^3 filter has been used.
For the Shepp Logan phantom a cross section with a range of 50 Hounsfield units is shown. For the second phantom there a two cross sections: the top cross section displays a range of 35 Hounsfield units while the lower cross section zooms in to 10 Hounsfield units. The artefacts of the filtered

backprojection occur due to the sharp transitions at the edges of the slits. Both Fourier methods provide excellent reconstruction quality.
During our experiments with computer generated data the oversampling factor d was the key parameter to avoid the typical Fourier reconstruction artefacts. Surprisingly increasing the number of angles p did not reduce these artefacts but increasing d did. However for experiments with real data none such artefacts occurred and oversampling of $c \cdot d = 2$ or even $c \cdot d = 1.5$ has been sufficient.

These results clearly demonstrate that Fourier based methods can compete with filtered back projection by means of reconstruction quality while still providing the excellent computational efficiency of FFT based methods. Removing the restriction for the grids to be equispaced from the Fast Fourier Transform allows Fourier methods to extend into areas where they were inappropriate before.

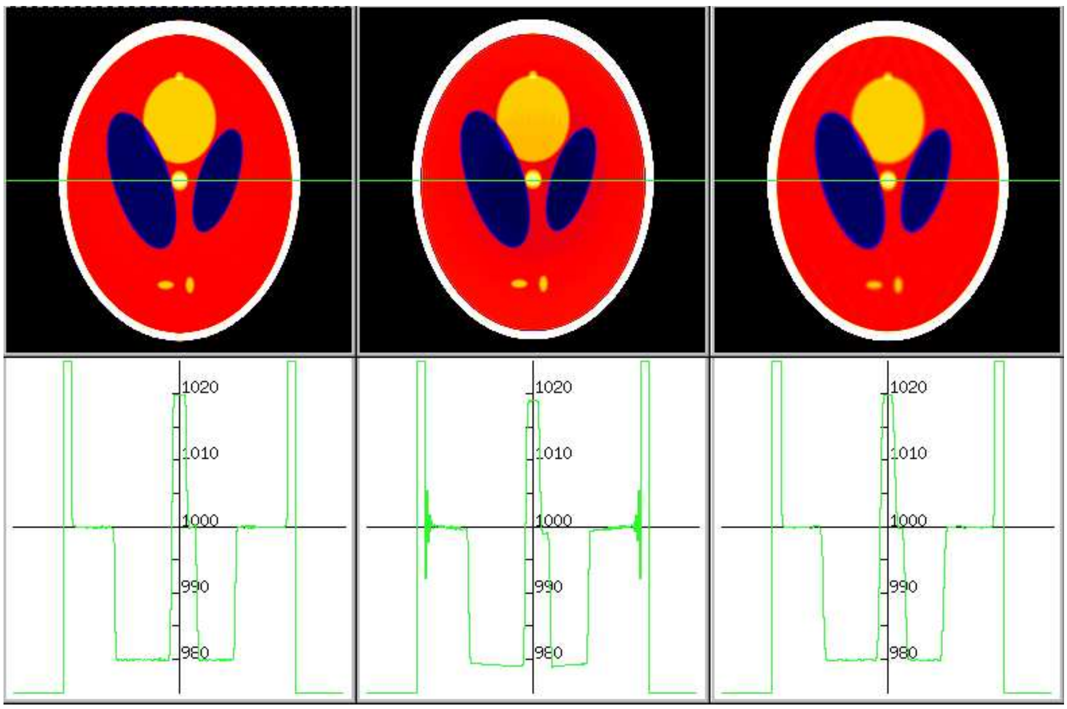

Figure 5: Image Reconstructions. From left to right: filtered backprojection, NED (grididng) Fourier reconstruction, NER fast Fourier reconstruction

References

[1] A. Averbuch, R. Coifman, D. Donoho, M. Israeli, and J. Waldén (2000), 'The pseudopolar FFT and its applications', Technical Report, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel.

[2] G. Beylkin (1995), 'On the fast Fourier transform of functions with singularities', *Appl. Comp. Harm. Anal.* **2**, 363-381.

[3] W.N. Brouw (1975), 'Aperture synthesis', in: B. Alder, S. Fernbach, and M. Rotenberg (eds.): *Methods in Computational Physics* **14**, pp. 131-175.

[4] A. Dutt and V. Rokhlin (1993), 'Fast Fourier transforms for nonequispaced data', *SIAM J. Sci. Comput.* **14**, 1368-1393.

[5] J.F. Kaiser (1966), 'Digital filters', in: F. Kuo and J.F. Kaiser (eds.): *System Analysis by Digital Computers*, 218-285.

[6] M. Kaveh and M. Soumekh (1987), 'Computer-assisted diffraction tomography', in: H. Stark (ed.): *Image Recovery: Theory and Application*, pp. 369-413.

[7] F. Natterer (1985), 'Fourier reconstruction in tomography', *Numerische Mathematik.* **47**, 343-353.

[8] F. Natterer (1986), *The Mathematics of Computerized Tomography.* John Wiley & Sons and B.G. Teubner.

[9] J.D. O'Sullivan (1985), 'A fast sinc function gridding algorithm for Fourier inversion in computer tomography', *IEEE Trans Med. Imag.*, vol. **MI-4**, no. 4, pp. 200-207.

[10] H. Schomberg and J. Timmer (1995), 'The gridding Method for Image Reconstruction by Fourier Transformation', *IEEE Trans. Med. Imag.*, vol. **14**, no. 3, 596-607.

[11] D. Slepian (1983), 'Some comments on Fourier analysis, uncertainty, and modeling', *SIAM Rev.* **25**, 379-393.

[12] G. Steidl (1998), 'A note on fast Fourier transforms for nonequispaced grids', *Advances in Computational Mathematics* **9**, 337-352.

[13] D.B. Twieg (1983),'The k-trajectory formulation of the NMR imaging process with applications in analysis and synthesis of imaging methods', *Med. Physics*, vol. **10**, no. 5, pp. 610-621.

[14] A. Ware (1998), 'Fast approximate Fourier transforms for irregularly spaced data', *SIAM Review* **40**, 838-856.

[15] Fourmont, K.: *Schnelle Fourier - Transformation bei nichtäquidistanten Gittern und tomographische Anwendungen.* Dissertation, University of Münster.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and Brian P. Flannery (1988), *Numericla Recipes.* Cambridge University Press

[17] J. Stoer and R. Bulirsch (1980), *Introduction to Numerical Analysis.* Springer Verlag

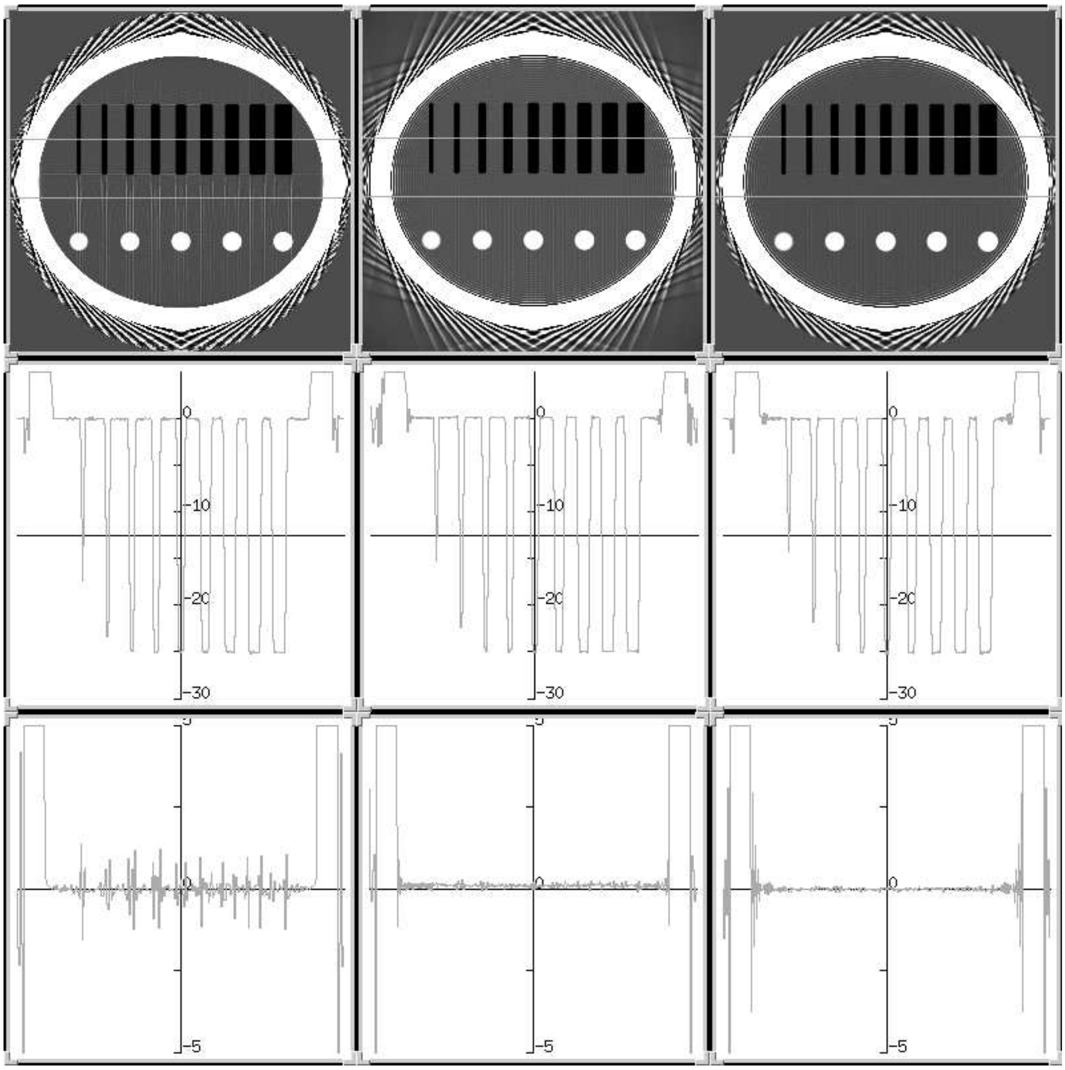

Figure 6: Image Reconstructions. From left to right: filtered backprojection, NED (grididng) Fourier reconstruction, NER fast Fourier reconstruction