An Exploration of the Application of Non-Uniform Fast Fourier Transforms to Parallel Beam Tomography

by

Hannah Stanton

An Expository Paper for Partial Fulfillment
of the Requirements for the Degree of
Masters of Science

OREGON STATE UNIVERSITY

Advisor:
Dr. Adel Faridani

9 July 2013

Acknowledgments

I would like to first and foremost thank my parents and brother. Without their unwavering love, support, and commonsense, I would not be where and who I am today. I am immeasurably thankful for all they have done and continue to do for me.

Next, I would like to thank my advisor, Dr. Adel Faridani. His guidance, help, and knowledge made this entire research paper possible.

I would also like to thank Charlie Robson for always giving just the right amount of support and positive encouragement, Nancy Scherich for putting up with me - even in my “early morning” mode, and Chelsea Hall for always lending an ear when asked.

1 Introduction

Transmission Computerized Tomography (CT) is the process of sending energy beams through an object, measuring the attenuation of the energy beams after they have passed through the object, and attempting to reconstruct an image of a cross-section of that object from gathered data. This process can be used to examine a wide variety of objects, such as insects, different portions of a human body, or even industrial equipment. CT is versatile in the sense that it can be used to construct 2-dimensional and 3-dimensional images, even without total 360° access to the object [1], [2].

There are different beams of energy that can be used in Transmission CT. X-rays are commonly used in the reconstruction of images of the human body and will be the type referred to in this paper. However, Ultrasound is used when an image is desired but radiation of the object in question is undesirable. Additionally, Electron Microscopy can be useful when a high resolution image of a small specimen is desired [1].

CT (also known as: computed axial tomography, computerized tomography, and computerized axial tomography) with X-rays is the process where a computer creates an image of an object from data collected after X-rays have been sent through the object. One procedure to gather the necessary data is to have a single source and detector directly opposite. The source sends X-rays through the object to the detector while both are moved in a parallel fashion. After the beams are sent through the length of the object, at predetermined intervals, the source and the detector are rotated and the process repeated. This is done for a specified number of rotation angles. This is known as Parallel Scanning, and is the type of scanning geometry that the data throughout this paper is based on. However, another scan type, Fan-Beam Scanning, is when a single source sends out beams at multiple angles to multiple detectors. Then both are rotate around the object [1]. Pictorial examples of these scan types are shown in Figure 1.

Either parallel or fan-beam scanning employed a single time will result in a planar cross-sectional image of the object. Implementing multiple times at various cross-sections allows reconstruction of a 3-dimensional image from the numerous 2-dimensional cross-sectional images produced.

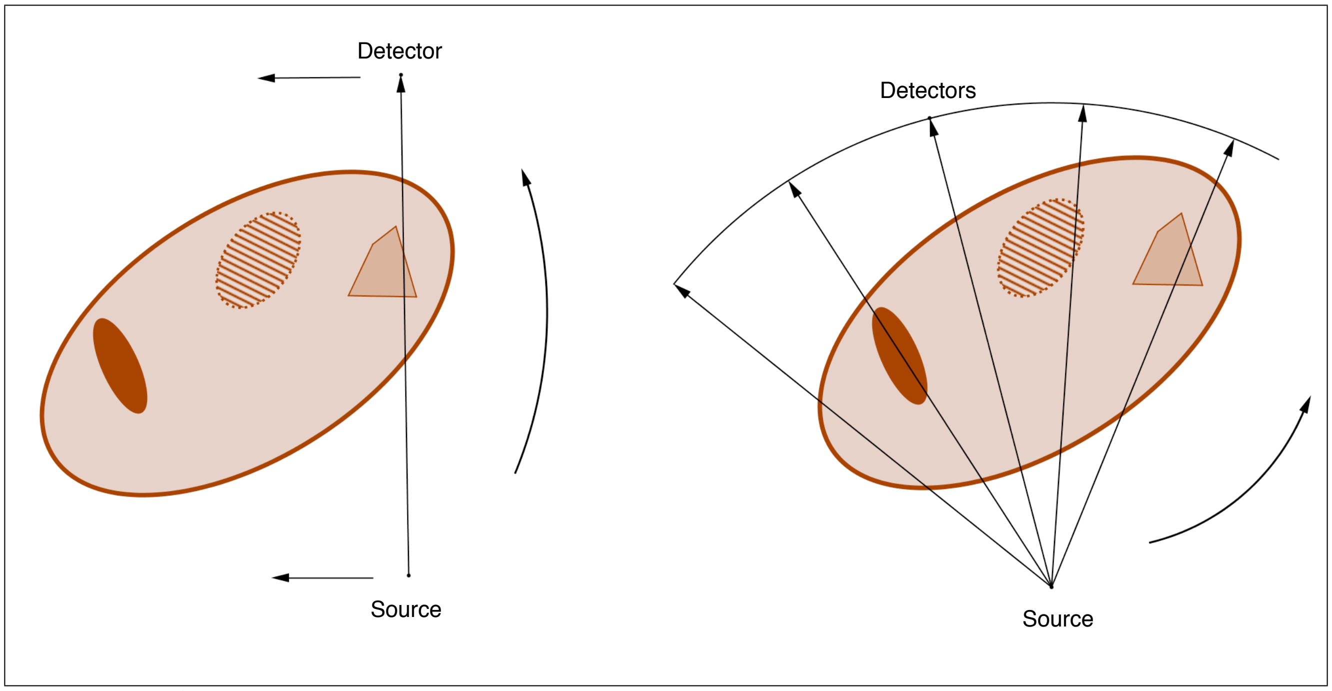

Figure 1: Above are basic examples of a cross section of an object with differing density within (the different shaded shapes). The left depicts parallel beam scanning, the right illustrates fan-beam scanning.

Behind this image reconstruction is a very interesting part of CT. To successfully create useful and accurate images, it is necessary to carefully evaluate the data gathered after an X-ray beam has traveled through the object. This reveals how much energy is lost from the X-ray beam at different locations of its path through the object, relating information about the density of the object at varying locations. Suppose some function f represents the X-ray attenuation coefficients at various points of the object. Thus, if we can figure out what values the function f takes throughout a specific cross-section of the object, then we know the image of that cross-section.

As with most fields in modern science, there remains plenty to be discovered, improved upon, and explored in the field of Computerized Tomography. While the development of modern tomography in the mid- to late-20th century was revolutionary, particularly with regards to medicine, the increased use of X-rays from CT scans is correlated with adverse affects in patients [3], [4]. This, and the fact that some of the mathematical algorithms behind image reconstruction sacrifice either accuracy or computation time during execution, results in continued research into additional mathematical models that would allow: safer data extraction, better time efficiency, and/or more accurate implementation.

This paper first presents the mathematical background that is useful in two different methods of image reconstruction. Next, we briefly examine the reconstruction methods Filtered Backprojection and one type of Fourier reconstruction. Following this brief evaluation, which motivates the later portion of this paper, we look at several non-uniform/non-equispaced 1- and 2-dimensional Fast Fourier Transform algorithms and, finally, their applications to Fourier reconstruction in Computerized Tomography.

2 Mathematical Background

To concisely and logically look at how to reconstruct a cross-sectional image of an object from gathered data, a few mathematical tools are necessary. The following subsections contain the mathematical background needed to examine several image reconstruction methods used in tomography.

2.1 The Fourier Transform

Let $f \in L_1(\mathbb{R}^n)$. Defining the Fourier Transform (FT) of f, and subsequently the accompanying Inverse Fourier Transform, can be done in multiple, equivalent ways (see [1] and [2]). This paper utilizes the following definition for the Fourier Transform. The Fourier Transform of f, denoted $\hat{f}$, is defined as follows,

$$\hat{f}(\xi) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} f(x)\ e^{-ix\cdot\xi} dx, \qquad \xi \in \mathbb{R}^n. \tag{1}$$

Theorem 2.1 states the accompanying Inverse Fourier Transform of equation (1) [2].

Theorem 2.1. *Fourier Inversion Formula.For $f \in L_1(\mathbb{R}^n)$, if $\hat{f} \in L_1(\mathbb{R}^n)$ then,*

$$f(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} \hat{f}(\xi)\ e^{i\xi\cdot x} d\xi, \qquad x \in \mathbb{R}^n. \tag{2}$$

Notice, $\hat{f}(\xi)$ exist and can be defined in this way because,

$$|\hat{f}(\xi)| = (2\pi)^{-n/2} \left| \int_{\mathbb{R}^n} f(x) e^{-ix\cdot\xi} dx \right| \leq (2\pi)^{-n/2} \int_{\mathbb{R}^n} \left| f(x) e^{-ix\cdot\xi} \right| dx = (2\pi)^{-n/2} ||f||_1 < \infty.$$

The final inequality follows by definition of f.

The Fourier Transform can be extended to $f \in L_2(\mathbb{R}^n)$ by continuity. Thus, the Inverse Fourier Transform can be similarly interpreted as in Theorem 2.1 for $f, \hat{f} \in L_2(\mathbb{R}^n)$.

2.2 Radon Transform

This sections information can be found in [1] and [2]. The purpose of Transmission Computerized Tomography is to create a 2- or 3-dimensional image from a function representing the desired image. The Radon Transform is a useful mathematical tool that reveals information about a function along lines in $\mathbb{R}^2$ or along planes in $\mathbb{R}^3$. The Radon Transform and the Fourier Transform are particularly useful when used in conjunction.

The Radon Transform, denoted $\mathbf{R}$, takes a function $f \in \mathbb{R}^n$ and maps it into the integral of f over hyperplanes in $\mathbb{R}^n$. For its use in cross-sectional image reconstruction, n is typically equal to 2 (i.e. 2-dimensions). In the 2-dimensional case, the Radon Transform takes $f(x) \in L_1(\mathbb{R}^2)$ vanishing outside some bounded domain, and maps f to the set of line integrals of f in the plane.

Let S^{n-1} denote the unit sphere in $\mathbb{R}^n$ and θ be any unit vector such that $\theta \in S^{n-1}$ (for example, S^1 is the unit circle in $\mathbb{R}^2$ with $\theta \in S^1$). Consider, $f \in L_1(\mathbb{R}^n)$ with $\theta \in S^{n-1}$, and $s \in \mathbb{R}$, then the Radon Transform of f is

$$\mathbf{R}f(\theta, s) = \int_{x\cdot\theta=s} f(x)dx = \int_{\theta^\perp} f(s\theta + t)dt, \qquad \theta^\perp := \text{hyperplane perpendicular to } \theta. \tag{3}$$

In two dimensions,

$$\mathbf{R}f(\theta, s) = \int_{x\cdot\theta=s} f(x)dx = \int_{-\infty}^{\infty} f(s\theta + t\theta^\perp)dt, \qquad \theta^\perp := 90^\circ \text{counterclockwise rotation of } \theta. \tag{4}$$

Thus, for $x \in \mathbb{R}^2$, $\mathbf{R}f(\theta, x\cdot\theta)$ = the integral of f over the line through the point x with direction perpendicular to θ.

Often, the Radon Transform is considered only applied to $f(\theta, s)$ for a single fixed $\theta \in S^{n-1}$ at a time. Thus only $s \in \mathbb{R}$ varies. The notation used in such instances is

$$\mathbf{R}_\theta f(s) := \mathbf{R}f(\theta, s).$$

Hence, the Radon Transform can be viewed as a function of a single variable for fixed θ; this holds for all $\theta \in S^{n-1}$.

Example. Consider the following example of the Radon Transform of a function $f \in L_1(\mathbb{R}^2)$.
Let

$$f(x) = \begin{cases} (1 - \|x\|^2)^m, & \|x\|^2 < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Shorthand, equation (5) can be written as follows,

$$f(x) = (1 - \|x\|^2)_+^m, \tag{6}$$

then for $s \in \mathbb{R}$ and $\theta \in S^1$ with $\theta^\perp$ the 90° counter-clockwise rotation of θ,

$$\mathbf{R}f(\theta, s) = \int_{-\infty}^{\infty} f(s\theta + t\theta^\perp)dt.$$

Let $g := \mathbf{R}f$. What does $g(\theta, s)$ equal when we evaluate the above expression for the given function f? For $x = s\theta + t\theta^\perp$, we have $\|x\|^2 = s^2 + t^2$. Thus,

$$\begin{aligned} g(\theta, s) &= \int_{-\infty}^{\infty} (1 - s^2 - t^2)_+^m dt \\ &= \int_{-\infty}^{\infty} (1 - s^2)^m \big(1 - \frac{t^2}{1 - s^2}\big)_+^m dt \end{aligned}$$

Let $u = \frac{t}{\sqrt{1-s^2}}$, therefore $du = \frac{dt}{\sqrt{1-s^2}}$. Then the $g(\theta, s)$ can be rewritten

$$\begin{aligned} g(\theta, s) &= (1 - s^2)^m \sqrt{1 - s^2} \bigg(\int_{-\infty}^{\infty} (1 - u^2)_+^m du \bigg), \\ &= (1 - s^2)^{m+1/2} \int_{-1}^{1} (1 - u^2)^m du. \end{aligned}$$

From Problem 9 (a) in [2], the above can be rewritten

$$g(\theta, s) = (1 - s^2)^{m+1/2} \bigg((2^{2m+1}) \frac{\big(\Gamma(m+1)\big)^2}{\Gamma(2m+2)} \bigg), \tag{7}$$

where $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$. This specific function, f (equation (5)), will be used in numerical experiments in later sections.

2.3 Projection Slice Theorem

The information in this section is based on [1]. An important theorem quickly follows the definitions of the Fourier and Radon Transforms. Consider applying the Fourier Transform to the Radon Transform of a function $f \in L_1(\mathbb{R}^n)$. There exists a neat relationship between the Fourier Transform of the Radon Transform of a function f and simply the Fourier Transform of f. We state this theorem for n-dimensions, but present only the 2-dimensional proof.

Theorem 2.2. *Projection Slice Theorem (PST). For $f \in L_1(\mathbb{R}^n)$,*

$$\widehat{(\mathbf{R}_\theta f)}(\sigma) = (2\pi)^{(n-1)/2} \hat{f}(\sigma\theta), \qquad \sigma \in \mathbb{R}. \tag{8}$$

Proof. Let $f \in L_1(\mathbb{R}^2)$, first consider for $\sigma \in \mathbb{R}$ and fixed $\theta \in S^1$ such that $\theta^\perp$ is the $90°$ counterclockwise rotation of θ,

$$\begin{aligned}\widehat{(\mathbf{R}_\theta f)}(\sigma) &= (2\pi)^{-1/2}\int_{\mathbb{R}} \mathbf{R}_\theta f(s) e^{-is\sigma} ds,\\ &= (2\pi)^{-1/2}\int_{\mathbb{R}} \Big(\int_{-\infty}^{\infty} f(s\theta + t\theta^\perp)dt\Big) e^{-is\sigma} ds.\end{aligned}$$

Let $x = s\theta + t\theta^\perp$, thus $s = \theta \cdot x$. By Fubini's Theorem, $dx = dt\ ds$ and

$$\begin{aligned}\widehat{(\mathbf{R}_\theta f)}(\sigma) &= (2\pi)^{-1/2}\int_{\mathbb{R}\times\mathbb{R}} f(x) e^{-i\theta\cdot x\sigma} dx,\\ &= (2\pi)^{-1/2}\int_{\mathbb{R}^2} f(x) e^{-ix\cdot\sigma\theta} dx,\\ &= (2\pi)^{-1/2}(2\pi)^{2/2}\hat{f}(\sigma\theta).\end{aligned}$$

This holds for all $\theta \in S^1$. Thus

$$\widehat{(\mathbf{R}_\theta f)}(\sigma) = \sqrt{2\pi}\hat{f}(\sigma\theta), \qquad \sigma \in \mathbb{R},$$

as desired. □

Theorem 2.2 is very powerful. It shows that given $\mathbf{R}f$, only a transformation and an inverse transformation are needed to solve for f.

Another nice result following from Theorem 2.2 is the following. Let $g(\theta, s) := \mathbf{R}f(\theta, s)$ for $f \in L_1(\mathbb{R}^n)$, $s \in \mathbb{R}$, and $\theta \in S^{n-1}$. Consider

$$\begin{aligned}\hat{g}(\theta,\sigma) &= (2\pi)^{(n-1)/2}\hat{f}(\sigma\theta),\\ &= (2\pi)^{(n-1)/2}\hat{f}\big((-\sigma)(-\theta)\big),\\ &= \hat{g}(-\theta,-\sigma).\end{aligned}$$

Thus, we obtain the equality

$$\hat{g}(\theta,\sigma) = \hat{g}(-\theta,-\sigma). \tag{9}$$

2.4 Discrete Fourier Transform and Discrete Inverse Fourier Transform

This section follows from Section VII.5 in [1]. A key aspect of the FT and IFT is their ability to be discretized, which we discuss here. This permits straightforward, numerical computation. The 1-dimensional Discrete Fourier Transform (DFT) of the input data $x_l \in \mathbb{R}$, $l = 0, \dots, p-1$, can be expressed as

$$\hat{x}_k = \sum_{l=0}^{p-1} x_l e^{-2\pi ikl/p}, \qquad k = 0, \dots, p-1. \tag{10}$$

Notice,

$$\sum_{l=0}^{p-1} e^{-2\pi ikl/p} = \begin{cases} p, & k = 0, \pm p, \pm 2p, \dots,\\ 0, & \text{otherwise.}\end{cases}$$

Thus, the expression for the Discrete Inverse Fourier Transform (DIFT) is,

$$x_l = \frac{1}{p}\sum_{k=0}^{p-1} \hat{x}_k e^{2\pi ilk/p}, \qquad k = 0, \dots, p-1. \tag{11}$$

Both of these equations, (10) and (11), require $\mathcal{O}(p^2)$ operations to evaluate. However, when p is a power of 2, the number of operations it takes to evaluate these expressions can be lowered to $\mathcal{O}(p \log p)$ using Fast Fourier Transform (FFT) techniques. One FFT technique is separating the sum into even and odd terms and using periodicity properties of the exponential to eliminate repetitive calculations. A well-known algorithm for this technique is by Cooley and Tukey, [5].

For a more general case, in 1-dimension, suppose there exists a function $f \in L_1(\mathbb{R})$ such that $f = 0$ outside $[-b, b]$ and $f_l := f(hl)$ for $l = -q, \ldots, q-1$ with step-size $h = b/q$. If $\hat{f} \in L_1(\mathbb{R})$ as well, then for $\hat{f}_k := \hat{f}(uk)$ for $k = -q, \ldots, q-1$, by the trapezoidal rule,

$$\hat{f}_k = (2\pi)^{-1/2} h \sum_{l=-q}^{q-1} f_l e^{-iuklb/q}, \qquad k = -q, \ldots, q-1. \tag{12}$$

When $u = \pi/b$, equation (12) can be evaluated with one FFT. When $u \neq \pi/b$, splitting equation (12) into multiple parts and evaluating each part via an FFT reduces the operation count to $\mathcal{O}(q \log q)$. For $u > 0$, this expression can be evaluated via an algorithm known as a chirp-z (see VII.5 in [1]). This separation is done as follows, write

$$-iuklb/q = \frac{bu}{2q}(k-l)^2 - \frac{bu}{2q}(k^2 + l^2).$$

Substituting this into (12) results in

$$\begin{aligned} \hat{f}_k &= e^{-i(bu/2q)k^2} \hat{f}'_k, \\ \hat{f}'_k &= (2\pi)^{-1/2} h \sum_{l=-q}^{q-1} e^{i(bu/2q)(k-l)^2} f'_l, \qquad k = -q, \ldots, q-1, \\ f'_l &= e^{-i(bu/2q)l^2} f_l. \end{aligned}$$

This expression for $\hat{f}'_k$ is a discrete convolution which can be evaluated with 2 FFTs – this above splitting and evaluation via FFT's is the chirp-z algorithm (see Chapter VII.5 in [1] for more details).

The default MatLab FFT and IFFT algorithms evaluate expressions of the form of (10) and (11) (with the slight modification of $k, l = 1, \ldots, p$ and $(k-1)(l-1)$ in the exponentials). However, in practice the centered Discrete Fourier Transform, similar to equation (12), often presents itself. MatLab has a built in command, fftshift, which allows the evaluation of this sum via the built in FFT command. Suppose we want to evaluate the expression,

$$\sum_{j=-N/2}^{N/2-1} u_j e^{-2\pi ikj/N}, \qquad k = -N/2, \ldots, N/2-1.$$

This can be done in MatLab with the following command, $\text{fftshift}\big(\text{fft}\big(\text{fftshift}(u)\big)\big)$. Thus, the fftshift command corresponds to swapping the first and second half of a vector (1-dimensional), or the first and third as well as the second and fourth quadrants of a matrix (2-dimensional).

2.5 Convolution of Functions

The information in this section is based on presentations in [1] and [2]. Another tool used in conjunction with the Fourier Transform is the convolution of functions.

Definition. Let $f, g \in L_2(\mathbb{R}^n)$, then "the convolution of f with g", denoted $f \star g$, is defined by

$$\big(f \star g\big)(x) = \int_{\mathbb{R}^n} f(x-y) g(y) dy, \qquad \forall\, x \in \mathbb{R}^n, \tag{13}$$

$$= \int_{\mathbb{R}^n} f(y) g(x-y) dy = \big(g \star f\big)(x), \qquad \forall\, x \in \mathbb{R}^n. \tag{14}$$

If $f, g \in L_1(\mathbb{R}^n)$ then $\big(f \star g\big)(x)$ is defined for a.e. $x \in \mathbb{R}^n$ and $f \star g \in L_1(\mathbb{R}^n)$.

A quick consequence of the definition of the convolution is that $f \star g$ is bounded. Suppose, for example, g is concentrated near the origin, then $\big(f \star g\big)(x)$ is an average of f near x. Figure 2 shows a graphical, 1-dimensional example of the convolution of two such functions.

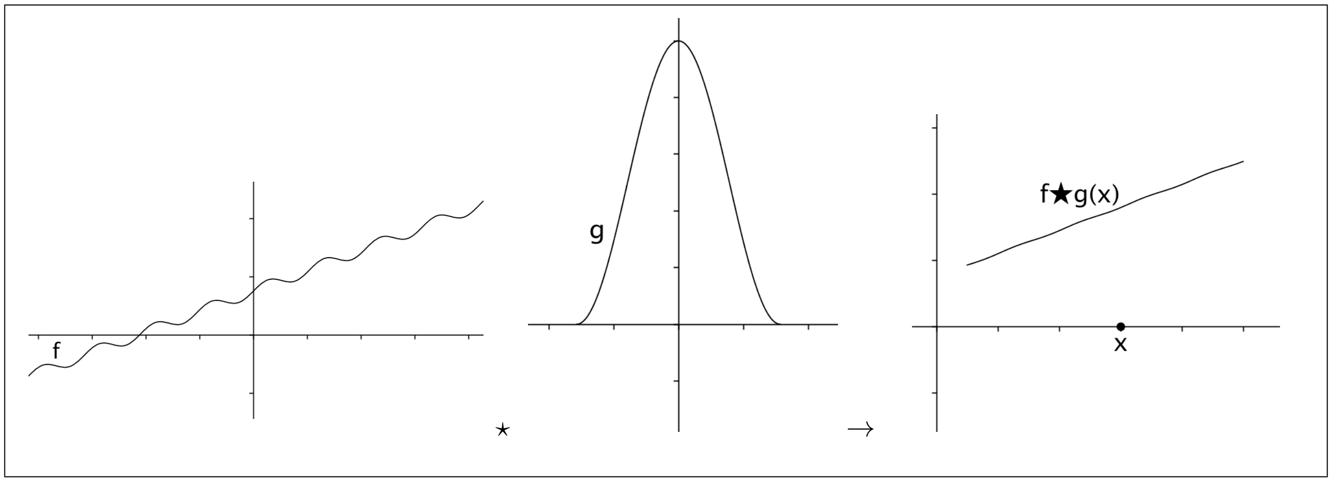

Figure 2: The left is a possible function f; middle is a possible function g, concentrated near the origin; right is the result of a convolution of f with g at some point $x \in \mathbb{R}$, i.e. the "average of f near x" due to the behavior of g.

Let $f, g \in L_1(\mathbb{R}^n)$, consider a Fourier Transform of the convolution of f with g. The result is the Convolution Theorem.

Theorem 2.3. *Convolution Theorem.Let $f, g \in L_1(\mathbb{R}^n)$, then*

$$\widehat{(f \star g)}(\xi) = (2\pi)^{n/2} \hat{f}(\xi)\hat{g}(\xi). \tag{15}$$

Proof. Take $f, g \in L_1(\mathbb{R}^n)$ and the Fourier Transform as defined in equation (1). Consider

$$\begin{aligned}
\widehat{(f \star g)}(\xi) &= (2\pi)^{-n/2} \int_{\mathbb{R}^n} \big(f \star g\big)(x) e^{-ix\cdot\xi} dx, \\
&= (2\pi)^{-n/2} \int_{\mathbb{R}^n} \Big(\int_{\mathbb{R}^n} f(x-y) g(y) dy \Big) e^{-ix\cdot\xi} dx, \\
&= (2\pi)^{-n/2} \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} f(x-y) g(y) e^{-ix\cdot\xi} dx \ dy, \\
&= (2\pi)^{-n/2} \int_{\mathbb{R}^n} g(y) e^{-iy\cdot\xi} \Big(\int_{\mathbb{R}^n} f(x-y) e^{-i(x-y)\cdot\xi} dx \Big) dy, \\
&= (2\pi)^{-n/2} \int_{\mathbb{R}^n} g(y) e^{-iy\cdot\xi} \big((2\pi)^{n/2} \hat{f}(\xi) \big) dy, \\
&= \int_{\mathbb{R}^n} g(y) \big(\hat{f}(\xi) e^{-iy\cdot\xi} \big) dy, \\
&= \hat{f}(\xi) \int_{\mathbb{R}^n} g(y) e^{-iy\cdot\xi} dy, \\
&= \hat{f}(\xi) (2\pi)^{n/2} \hat{g}(\xi).
\end{aligned}$$

Therefore,

$$\widehat{(f \star g)}(\xi) = (2\pi)^{n/2} \hat{f}(\xi)\hat{g}(\xi),$$

as desired. □

Theorem 2.3 can be found in VII.1 of [1] as R4.

2.6 Additional Useful Mathematical Tools

This section presents a few additional tools which are derived from the previous sections. First, applying the Radon transform to a convolution of functions gives an interesting result stated in Theorem 2.4 for the 2-dimensional case [1].

Theorem 2.4. *Let $f, g \in L_1(\mathbb{R}^2)$, then*

$$\mathbf{R}_\theta(f \star g) = \mathbf{R}_\theta f \star \mathbf{R}_\theta g.$$

The proof of this theorem utilizes Theorems 2.2 and 2.3 (for more background and/or detail see [1]).

Proof. Take $f, g \in L_1(\mathbb{R}^2)$ and fix $\theta \in S^1$. Consider, for $\sigma \in \mathbb{R}$,

$$\begin{aligned}(\mathbf{R}_\theta f \star \mathbf{R}_\theta g)\widehat{\ }(\sigma) &= (2\pi)^{1/2}\big(\widehat{(\mathbf{R}_\theta f)}(\sigma)\big)\big(\widehat{(\mathbf{R}_\theta g)}(\sigma)\big),\\ &= (2\pi)^{1/2}\big((2\pi)^{1/2}\hat{f}(\sigma\theta)\big)\big((2\pi)^{1/2}\hat{g}(\sigma\theta)\big),\\ &= (2\pi)^{1/2}\Big(\big(f \star g\big)\widehat{\ }(\sigma\theta)\Big),\\ &= \big(\mathbf{R}_\theta(f \star g)\big)\widehat{\ }(\sigma).\end{aligned}$$

This holds true for all $\theta \in S^1$. Thus, $\mathbf{R}_\theta(f \star g) = \mathbf{R}_\theta f \star \mathbf{R}_\theta g$ as well. □

The dual of the Radon Transform, denoted $\mathbf{R}^\sharp$, can can be useful in tomography (see Chapter II in [1]). Consider $f \in C_0^\infty(\mathbb{R}^2)$ and let $g(\theta, s) := \mathbf{R}_\theta f(s)$. The formulation of $\mathbf{R}^\sharp$ is such that it is the formal adjoint of $\mathbf{R}$. Consider,

$$\langle \mathbf{R}f, g\rangle_{L_2(S^1\times\mathbb{R})} = \langle f, \mathbf{R}^\sharp g\rangle_{L_2(\mathbb{R}^2)}, \tag{16}$$

where S^1 indicates the unit circle in $\mathbb{R}^2$. So, consider for $s \in \mathbb{R}, \theta \in S^1$ such that $\theta = \binom{\cos(\varphi)}{\sin(\varphi)}$,

$$\begin{aligned}\langle \mathbf{R}f, g\rangle_{L_2(S^1\times\mathbb{R})} &= \int_0^{2\pi}\int_{-\infty}^{\infty} \mathbf{R}f(\theta, s)\overline{g(\theta, s)}ds\ d\varphi,\\ &= \int_0^{2\pi}\int_{-\infty}^{\infty}\Big(\int_{-\infty}^{\infty} f(s\theta + t\theta^\perp)dt\Big)\overline{g(\theta, s)}ds\ d\varphi.\end{aligned}$$

Let $x = s\theta + t\theta^\perp$ then $s = x \cdot \theta$; by Fubini's Theorem,

$$\begin{aligned}\langle \mathbf{R}f, g\rangle_{L_2(S^1\times\mathbb{R})} &= \int_0^{2\pi}\int_{\mathbb{R}^2} f(x)\overline{g(\theta, x\cdot\theta)}dx\ d\varphi,\\ &= \int_{\mathbb{R}^2} f(x)\overline{\Big(\int_0^{2\pi} g(\theta, x\cdot\theta)d\varphi\Big)}dx.\end{aligned}$$

If $\mathbf{R}^\sharp g(x) := \int_0^{2\pi} g(\theta, x\cdot\theta)d\varphi$ (*) for $x \in \mathbb{R}^2$, then the desired inner product equality results. Thus, let $\mathbf{R}^\sharp$ be defined by (*). Then

$$\begin{aligned}\langle \mathbf{R}f, g\rangle_{L_2(S^1\times\mathbb{R})} &= \int_{\mathbb{R}^2} f(x)\overline{\Big(\mathbf{R}^\sharp g(x)\Big)}dx\\ &= \langle f, \mathbf{R}^\sharp g\rangle_{L_2(\mathbb{R}^2)}.\end{aligned}$$

Thus the pair of $\mathbf{R}$ and $\mathbf{R}^\sharp$ are duals. $\mathbf{R}^\sharp$ is called the *Back Projection Operator* because, for example, if $x \in \mathbb{R}^2$, then

$$\mathbf{R}^\sharp \mathbf{R} f(x) = \int_0^{2\pi} \mathbf{R} f(\theta, x \cdot \theta) d\varphi, \qquad \theta \in S^1.$$

In words, $\mathbf{R}^\sharp \mathbf{R} f(x)$ sums up the values of the integrals of f over all lines that pass through x.

Finally, again let f be the function corresponding to the image we wish to reconstruct. An expression for f stated solely in terms of gathered data would be ideal. Through a combination of the Fourier Inversion Formula, Theorem 2.1, and the Projection Slice Theorem, Theorem 2.2, such an expression is possible (see [6]).

Take $g := \mathbf{R} f$ for $f \in L_1(\mathbb{R}^2)$. Consider the Fourier Transform of f,

$$f(x) = (2\pi)^{-1} \int_{\mathbb{R}^2} \hat{f}(\xi) e^{ix\cdot\xi} d\xi, \qquad x \in \mathbb{R}^2,$$

such that $\hat{f} \in L_1(\mathbb{R}^2)$. Now, let $\xi = \sigma\theta$, where $\theta = \binom{\cos\varphi}{\sin\varphi}$ and $\sigma \in \mathbb{R}$, and reparametrize the previous expression with polar coordinates

$$\begin{aligned} f(x) &= (2\pi)^{-1} \int_0^{2\pi} \Big(\int_0^\infty \hat{f}(\sigma\theta) e^{ix\cdot\sigma\theta} |\sigma| \; d\sigma \Big) d\varphi, \\ &= (2\pi)^{-1} \int_0^{2\pi} \Big(\int_0^\infty (2\pi)^{-1/2} \widehat{\mathbf{R} f}(\theta, \sigma) e^{ix\cdot\sigma\theta} |\sigma| \; d\sigma \Big) d\varphi, \\ &= (2\pi)^{-3/2} \int_0^{2\pi} \Big(\int_0^\infty \hat{g}(\theta, \sigma) e^{ix\cdot\sigma\theta} |\sigma| \; d\sigma \Big) d\varphi. \end{aligned}$$

Recall $\hat{g}(\theta, \sigma) = \hat{g}(-\theta, -\sigma)$ (equation (9)), hence we can extend the lower limit of integration over σ to $-\infty$. Thus

$$f(x) = (2\pi)^{-3/2} \int_0^{2\pi} \frac{1}{2} \Big(\int_{-\infty}^\infty \hat{g}(\theta, \sigma) e^{ix\cdot\sigma\theta} |\sigma| \; d\sigma \Big) d\varphi.$$

Therefore, an equation for f, solely expressed in terms of g, employing the previously presented mathematical tools, is

$$f(x) = \frac{1}{2} (2\pi)^{-3/2} \int_0^{2\pi} \Big(\int_{-\infty}^\infty \hat{g}(\theta, \sigma) e^{ix\cdot\sigma\theta} |\sigma| \; d\sigma \Big) d\varphi. \tag{17}$$

3 Two Algorithms for Image Reconstruction

This section's information follows from presentations found in Chapter V of [1], [2], and lecture notes from *Numerical Analysis: Numerical Harmonic Analysis with Applications to Image Reconstruction and Processing* with Dr. Faridani, Fall '12.

Here we present the mathematical theory behind two algorithms for image reconstruction. This presentation is done in 2-dimensions. Following the theory are numerical examples.

3.1 Filtered Backprojection

Section 2 presented sufficient background information and theory to introduce the algorithm known as *Filtered Backprojection* (FBP). This reconstruction method combines the Back Projection Operator, $\mathbf{R}^\sharp$, with a filter function to reproduce a function, f, from $g := \mathbf{R} f$ sampled in $\mathbb{R}^2$; thus coining the method's name.

Consider a function W such that

$$\int_{\mathbb{R}^2} W(x)dx = 1, \qquad x \in \mathbb{R}^2.$$

Define $W_b(x) := b^2 W(bx)$, hence $W_b \to \delta$ as $b \to \infty$. Then for $f \in L_1(\mathbb{R}^2)$,

$$\lim_{b\to\infty} \|f - W_b \star f\|_1 = 0.$$

We also choose W_b to be a radial function.

Thus, instead of attempting to reconstruct f exactly, we can attempt to reconstruct $W_b \star f$, an approximation of f. Therefore, consider

$$\begin{aligned}(W_b \star f)(x) &= (2\pi)^{-1} \int_{\mathbb{R}^2} (\widehat{W_b \star f})(\xi) e^{ix\cdot\xi} d\xi, \\ &= (2\pi)^{-1} \int_{\mathbb{R}^2} 2\pi\ \widehat{W_b}(\xi)\hat{f}(\xi)e^{ix\cdot\xi}d\xi, \\ &= \int_{\mathbb{R}^2} \widehat{W_b}(\xi)\hat{f}(\xi)e^{ix\cdot\xi}d\xi.\end{aligned}$$

Take $\sigma\theta := \xi$ where $\sigma \in \mathbb{R}$ and $\theta \in S^1$ such that $\theta = \binom{\cos(\varphi)}{\sin(\varphi)}$; then the previous can be rewritten as

$$(W_b \star f)(x) = \int_0^{2\pi} \int_0^{\infty} \widehat{W_b}(\sigma\theta)\hat{f}(\sigma\theta)e^{ix\cdot\sigma\theta}|\sigma|\ d\sigma\ d\varphi.$$

For $g := \mathbf{R}f$, we know $\hat{g}(\theta, \sigma) = \hat{g}(-\theta, -\sigma)$. Hence, we have the following equality,

$$\int_0^{2\pi} \int_0^{\infty} |\sigma|\ \widehat{W_b}(\sigma\theta)\hat{f}(\sigma\theta)e^{ix\cdot\sigma\theta}d\sigma d\varphi = \int_0^{2\pi} \int_{-\infty}^{0} |\sigma|\ \widehat{W_b}(\sigma\theta)\hat{f}(\sigma\theta)e^{ix\cdot\sigma\theta}d\sigma d\varphi.$$

Thus,

$$\begin{aligned}(W_b \star f)(x) &= \int_0^{2\pi} \frac{1}{2} \int_{-\infty}^{\infty} \widehat{W_b}(\sigma\theta)\hat{f}(\sigma\theta)e^{ix\cdot\sigma\theta}|\sigma|\ d\sigma\ d\varphi, \\ &= \frac{1}{2}\int_0^{2\pi} \int_{-\infty}^{\infty} |\sigma|\big((2\pi)^{-1/2}\widehat{\mathbf{R}_\theta W_b}(\sigma)\big)\big((2\pi)^{-1/2}\widehat{\mathbf{R}_\theta f}(\sigma)\big)e^{ix\cdot\sigma\theta}d\sigma d\varphi, \\ &= \int_0^{2\pi} \int_{-\infty}^{\infty} \frac{1}{2}(2\pi)^{-1}|\sigma|\widehat{\mathbf{R}_\theta W_b}(\sigma)\widehat{\mathbf{R}_\theta f}(\sigma)e^{ix\cdot\sigma\theta}d\sigma d\varphi.\end{aligned}$$

By definition, W_b does not depend on θ, thus neither does $\widehat{W_b}$. Therefore, let $\widehat{w_b}(\sigma) := \frac{1}{2}(2\pi)^{-3/2}|\sigma|\widehat{\mathbf{R}_\theta W_b}(\sigma)$. Then

$$\begin{aligned}(W_b \star f)(x) &= \int_0^{2\pi} \int_{-\infty}^{\infty} (2\pi)^{1/2}\widehat{w_b}(\sigma)\widehat{\mathbf{R}_\theta f}(\sigma)e^{ix\cdot\sigma\theta}d\sigma d\varphi, \\ &= \int_0^{2\pi} \int_{-\infty}^{\infty} \big(w_b \star \mathbf{R}_\theta f\big)\widehat{\ }(\sigma)e^{ix\cdot\sigma\theta}d\sigma d\varphi, \\ &= \int_0^{2\pi} \big(w_b \star \mathbf{R}_\theta f\big)(x\cdot\theta)d\varphi.\end{aligned}$$

Thus,

$$\big(W_b \star f\big)(x) = \int_0^{2\pi} \big(w_b \star \mathbf{R}_\theta f\big)(x\cdot\theta)d\varphi$$

which, by definition of $\mathbf{R}^\sharp$, is equivalent to

$$\big(W_b \star f\big)(x) = \mathbf{R}^\sharp\big(\big(w_b \star \mathbf{R}_\theta f\big)(x)\big). \tag{18}$$

Recall, $\mathbf{R}^\sharp$ is the Back Projection Operator, w_b is the "filter", and $\mathbf{R}_\theta f(x)$ is the data. Equation (18) is a succinct formula for $\big(W_b \star f\big)$ based on $\mathbf{R}f$. Discretization is the next step towards numerically solving for f.

Let $\Omega :=$ unit disc in $\mathbb{R}^2$. Consider $f \in C_0^\infty(\Omega)$ and sample $g := \mathbf{R}f$ at (θ_j, s_l) for $j = 0,\dots,p-1$ and $l = -q,\dots,q-1$, where $\theta_j = \binom{\cos(\varphi_j)}{\sin(\varphi_j)}$ with $\varphi_j = \frac{2\pi j}{p}$, and $s_l = hl$ with $h = \frac{1}{q}$. This type of sampling scheme corresponds to Parallel Beam scanning geometry (see Figure 1, Section 1). Utilizing this sampling scheme, we discretize the convolutions, $w_b \star g$, and represent the discrete versions with a subscript h:

$$\big(w_b \star g\big)(\theta_j, s) = \int_{-\infty}^{\infty} w_b(s-y)g(\theta_j, y)dy, \qquad s\in\mathbb{R},$$

$$\Longrightarrow$$

$$\big(w_b \star g\big)_h(\theta_j, s) = h\sum_{l=-q}^{q-1} w_b(s-s_l)g(\theta_j, s_l), \qquad s\in\mathbb{R}. \tag{19}$$

Next, we discretize the Back Projection Operator with the Trapezoidal Rule for p nodes. Using the Trapezoidal rule is accurate because g, and thus $w_b\star g$, is 2π-periodic with respect to θ. The discrete version is signified with a subscript p:

$$\mathbf{R}^\sharp\big(w_b \star g\big)(x) = \int_0^{2\pi} \big(w_b \star g\big)(\theta, x\cdot\theta)d\varphi, \qquad x\in\mathbb{R}^2,$$

$$\Longrightarrow$$

$$\mathbf{R}^\sharp{}_p\big(w_b \star g\big)(x) = \frac{2\pi}{p}\sum_{j=0}^{p-1} \big(w_b \star g\big)(\theta_j, x\cdot\theta_j), \qquad x\in\mathbb{R}^2. \tag{20}$$

So, combining these two discretizations, (19) and (20), the expression the discrete Filtered Backprojection evaluates, is

$$\mathbf{R}^\sharp{}_p\big(w_b \star g\big)_h(x) = \frac{2\pi}{p}\sum_{j=0}^{p-1} h\sum_{l=-q}^{q-1} w_b(x\cdot\theta_j - s_l)g(\theta_j, s_l), \qquad x\in\mathbb{R}^2.$$

Thus, for $x\in\mathbb{R}^2$ and $f\in L_1(\mathbb{R}^2)$,

$$\big(W_b \star f\big)(x) \simeq \mathbf{R}^\sharp{}_p\big(w_b \star g\big)_h(x). \tag{21}$$

The filtering portion of equation (21), takes $\mathcal{O}(pq)$ operations for each x, totaling $\mathcal{O}(pq^3)$ operations. The back projection portion takes $\mathcal{O}(p)$ operations for each x. Thus, when computed in this fashion Filtered Backprojection takes $\mathcal{O}(p^4)$ operations for $p \simeq q$.

Computing the convolution first at equidistant points and then using linear interpolation lowers the convolution operation count. Using FFT's the number of operations for the convolutions reduces to

$\mathcal{O}(pq\log q)$. However, the backprojection portion still takes $\mathcal{O}(pq^2)$ operations. Thus, when $p \simeq q$ the number of operations is still $\mathcal{O}(p^3)$. The interpolation step is denoted I_h. Thus

$$\big(w_b \star g\big)(\theta, s) \simeq I_h\big(w_b \star g\big)_h(\theta, s).$$

Therefore, the Filtered Backprojection Algorithm computes an approximation of f, which we denote f_{FBI},

$$f_{FBI}(x) = \big(W_b \star f\big)(x) \simeq \mathbf{R}^\sharp{}_p I_h\big(w_b \star g\big)_h(\theta_j, x \cdot \theta_j), \qquad x \in \mathbb{R}^2. \tag{22}$$

We briefly outline the steps of the Filtered Backprojection reconstruction algorithm.

- *Filtered Backprojection Algorithm Outline:*
  - Step 1: Compute discrete convolutions, $(w_b \star g)_h(\theta, s)$, for $s = s_l$, $l = -q, \dots, q$.
  - Step 2: For each reconstruction point x find $(2\pi/p)I_h(w_b \star g)_h(\theta_j, x \cdot \theta_j)$ using linear interpolation and add this value to the pixel representing x.

Note. The Shepp-Logan Phantom (with one added ellipse along the "skull") is used in upcoming Sections 3.2, 3.4, 5.2, and 5.4. It corresponds to a fictional cross-section of a human brain with specific cross-sectional areas – ellipses – of different sizes within the unit circle. Each ellipse has one of the following "density values":

$$[1,\ -0.98,\ -0.02,\ -0.02,\ 0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.03].$$

These "density values" tell us for x in the unit circle, if x is within multiple ellipses, then the value of x is equal to the sum of the multiple ellipse values (this is why negative densities are allowed). The first ellipse with density 1 represents the skull, while the other ellipses are ventricles, "tumors", and grey matter.

3.2 Image Reconstruction via Filtered Backprojection

The Filtered Backprojection algorithm executed in MatLab for this section is attributed to Dr. Faridani; code last revised April 14, 2008. The code follows from the algorithm briefly outlined at the end of Section 3.1 and is used to reconstruct the 2 dimensional Shepp-Logan phantom. The execution time, and the relative error are briefly analyzed. The implementation time is what leads us to explore alternative methods utilizing Fast Fourier Transforms in image reconstruction.

Consider the function f from the example in Section 2.2, equation (6),

$$f(x) = (1 - \|x\|^2)_+^m, \qquad x \in \mathbb{R}^2 \text{ and } \ m > -1 \tag{23}$$

(see [7] for the negative case). Let $g := \mathbf{R}f$ be sampled at (θ_j, s_l) for $j = 0, \dots, p-1$ and $l = -q, \dots, q-1$, where $\theta_j = \binom{\cos(\varphi_j)}{\sin(\varphi_j)}$ with $\varphi_j = \frac{2\pi j}{p}$, and $s_l = hl$ with $h = \frac{1}{q}$. The experimentation is carried out with multiple values of p, q and m. Figure 3 shows an image of $2q \times 2q$ pixels for $m = 0, q = 128$ and $p = 400$. The two tables following figure 3 display the implementation time and the relative error for the varying values of p and q for $m = 0$ or $m = 3$, respectively.

In the reconstructions, we look at the relative error for the $m = 3$ case. This is because, although the $m = 0$ case produces useful images, the error in that case is *very* high - roughly 25% - due to the discontinuous changes in density. Thus we use the smoother, $m = 3$ case to analyze the relative error.

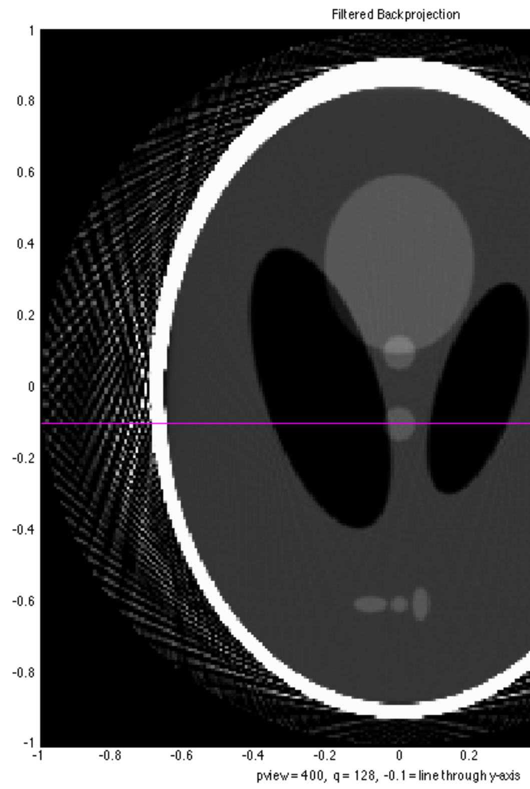

Figure 3: Filtered Backprojection reconstructed image created in MatLab with a Shepp-Logan filter (see [1] page 110-111); code attributed to Dr. Faridani.

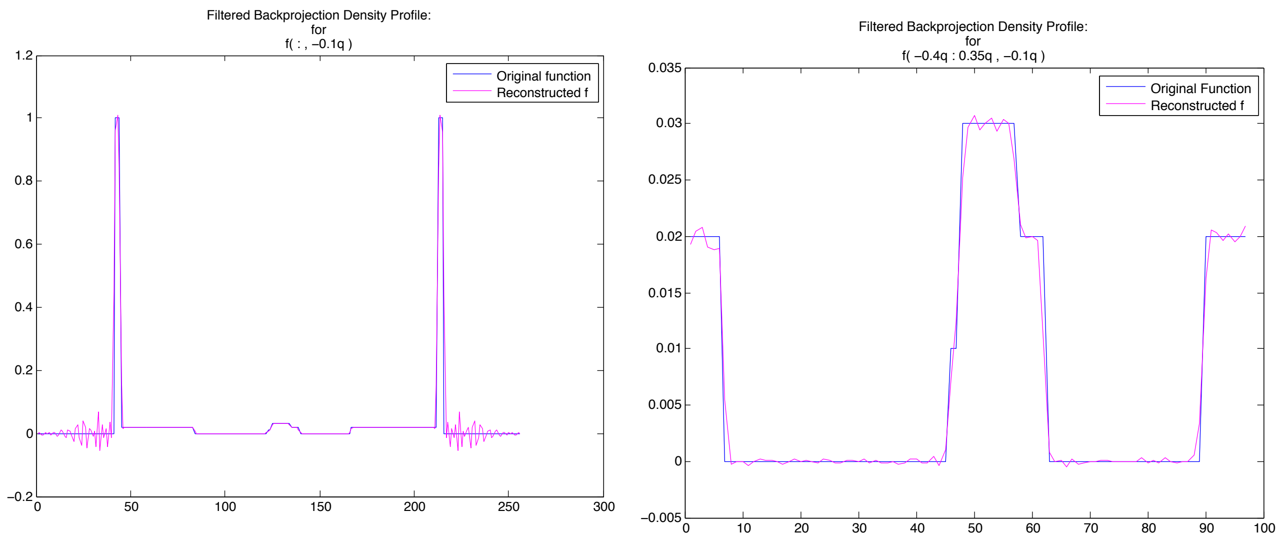

Figure 4: Density Profile corresponding to the line through the image in Figure 3. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; code attributed to Dr. Faridani.

Table 1: Time for $m = 0$:
(time in seconds)

p \ q	64	128	256	512
100	0.08	0.20	0.89	3.82
200	0.16	0.40	1.65	7.59
400	0.30	0.80	3.34	15.16
800	0.60	1.73	7.02	30.45

Table 2: Relative Error for $m = 3$:
(x10$e - 3$)

p \ q	64	128	256	512
100	6.97	2.92	2.44	2.47
200	6.76	2.19	0.76	0.39
400	6.76	2.16	0.73	0.32
800	6.76	2.16	0.73	0.31

The image produced by the Filtered Backprojection method is a nice clear image (see Figure 3). It is easy to differentiate areas of different density and the borders between the abrupt changes are clear, as one would expect the border of two actual different physical objects. This differentiation can really be seen in Figure 4. The reconstructed f density profile aligns with the density profile of the original function with very minor deviations. There is a slight amount of ringing artifact but this is mainly outside of the region we are most interested in.

The relative error in the $m = 3$ case is also satisfactory; the error is below 1% for every pairing of p and q and below 0.22% for $q = 128$ and $p = 400$.

While this method has small error, the time efficiency for certain values of p and q is less than satisfactory. For small p and q the amount of time the reconstruction takes is reasonable, but the image quality is too poor to be useful for those values. However, increasing p and q significantly increases the method's implementation time. Looking at the high-lighted, lower off-diagonal entries in Table 1, we see that as p and q increase by a factor of 2 the execution time increases by a factor of about 8. This is *not* optimal.

Thus, although Filtered Backprojection produces a clear image, a faster reconstruction method is desirable. Hence, we look at Fourier reconstruction. Fourier reconstruction attempts to directly compute f using the Fourier Inversion Formula - Theorem 2.1, the Projection Slice Theorem - Theorem 2.2, and FFTs.

3.3 Fourier Reconstruction

The Fourier reconstruction discussed here follows from the presentation on pages 126-127 in Chapter V.2 of [1].

Let $f \in L_1(\mathbb{R}^2)$ and take $g := \mathbf{R}f$. Consider

$$\begin{aligned} f(x) &= (2\pi)^{-1} \int_{\mathbb{R}^2} \hat{f}(\xi) e^{ix\cdot\xi} d\xi, \\ &\simeq (2\pi)^{-1} \tilde{d} \sum_{k\in\mathbb{Z}^2} \hat{f}(\tilde{d}k) e^{ix\cdot\tilde{d}k}, \qquad \tilde{d} \in \mathbb{R}. \end{aligned}$$

This requires $\hat{f}$ on a cartesian grid. Then the summation can be evaluated via a 2-dimensional FFT. Thus, we must find or approximate $\hat{f}(\tilde{d}k)$. This is done by interpolation from $\hat{g}(\theta_j, \sigma) = \sqrt{2\pi}\hat{f}(\sigma\theta_j)$. So, if we know $\hat{g}(\theta_j, \sigma)$ and, hence $\hat{f}$ by Theorem 2.2, on a polar grid, interpolation is required to find $\hat{f}$ on a cartesian grid. However, it is known that this interpolation has to be implemented very carefully. A relatively simple method using nearest neighbor in either the vertical or horizontal direction is suggested by Natterer (see pages 126-127 [1]). The following is an outline of this basic Fourier reconstruction algorithm.

Let $\Omega :=$ unit disc in $\mathbb{R}^2$ and take $f \in C_0^\infty(\Omega)$, and sample $g := \mathbf{R}f$ at (θ_j, s_l) for $j = 0, \dots, p-1$ and $l = -q, \dots, q-1$, where $\theta_j = \binom{\cos(\varphi_j)}{\sin(\varphi_j)}$ with $\varphi_j = \frac{2\pi j}{p}$ and $s_l = hl$, where $h = \pi^2/(p\sqrt{2})$, and define

$c_j = 1/(\max\{|\sin(\varphi_j)|, |\cos(\varphi_j)|\})$ (see Chapter V of [1] for more details on these choices for h and c_j). Then for $j = 0, \dots, p-1$,

$$\hat{g}(\theta_j, r\pi c_j) \simeq \hat{g}_{j,r} := (2\pi)^{-1/2} h \sum_{l=-q}^{q-1} g(\theta_j, s_l) e^{-ilr\pi c_j/q}, \qquad r = -q, \dots, q-1.$$

Thus,

$$\hat{f}(r\pi c_j \theta_j) \simeq (2\pi)^{-1/2} \hat{g}_{j,r}, \qquad r = -q, \dots, q-1. \tag{24}$$

These approximations of $\hat{f}$ are on a polar coordinate grid, thus this expression cannot be evaluated via an FFT. To have an efficient algorithm, we want to use FFTs. Thus, we interpolate $\hat{f}$ to a cartesian grid.

Let $k \in \mathbb{Z}^2$ such that $|k| < q$. For k in Quadrant I such that $|k| < q$, if $k_1 > k_2$, move that point horizontally to a point on the nearest ray, φ_j; if $k_1 < k_2$, move that point vertically to a point on the nearest ray, φ_j. Go through a similar process for the remaining $k \in \mathbb{Z}^2$ such that $|k| < q$. This can be summarize as: for each $k \in \mathbb{Z}^2$ such that $|k| < q$, choose r and j such that $|\pi k - r c_j \theta_j|$ is minimized, i.e. nearest neighbor interpolation in either the horizontal or vertical direction (see Figure 5, depicted after Figure V.7 on page 126 in [1]). Then, for all $k \in \mathbb{Z}^2$ such that $|k| < q$, approximate $\hat{f}(\pi k)$ by

$$\hat{f}(\pi k) := \hat{f}_k \simeq (2\pi)^{-1/2} \hat{g}_{j,r}.$$

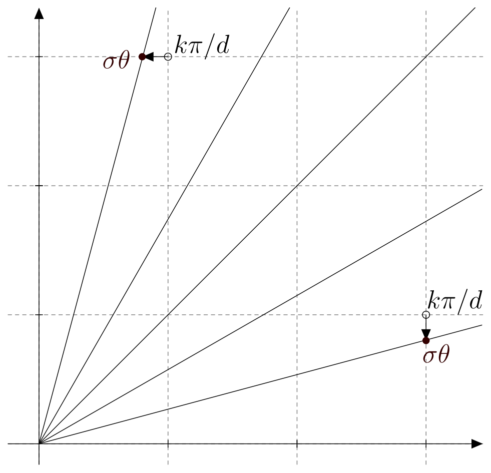

Figure 5: Natterer's Nearest Neighbor interpolation scheme, in either the vertical or horizontal direction, used in his Fourier reconstruction algorithm.

Finally compute a discrete 2-dimensional Inverse Fourier Transform of $\hat{f}_k$. An IFFT approximates $f(hm)$ for $m \in \mathbb{Z}^2$, i.e.,

$$f(hm) := f_m \simeq \left(\frac{\pi}{2}\right) \sum_{|k|<q} \hat{f}_k e^{i\pi m \cdot k/q}, \qquad |m| < q. \tag{25}$$

The numerical evaluation of equation (25) is one type of reconstruction referred to as *Fourier reconstruction* (FR). We briefly outline its algorithmic implementation.

- *Fourier Reconstruction Algorithm Outline:*

– Step 1: Approximate $\hat{g}_{j,r}$ on a polar grid via p FFTs with respect to the second variable of g.

– Step 2: Interpolate $\hat{g}_{j,r}$ from Step 1 to a cartesian grid to acquire $\hat{f}_k$.

– Step 3: Finally, take a 2-dimensional IFFT of $\hat{f}_k$ from Step 2 for an approximation of f.

The above outline does not produce a useful image, and compared to the later Fourier reconstruction methods, it is lacking an oversampling factor. Thus, for experimentation, and to allow fair comparison, we introduce an oversampling factor, d. This is done in Fourier space. Instead of approximating $\hat{f}_k$ for only $|k| < q$, we approximate $\hat{f}_k$ for all $k \in \mathbb{Z}^2$ such that $|k| \leq dq$. The first and last steps of the outlined algorithm remain the same with the exception of $|k| < dq$ in Step 3.

3.4 Image Reconstruction via Basic Fourier Reconstruction

Similar to Section 3.2, here we apply the algorithm outlined at the end of Section 3.3 to a 2-dimensional image reconstruction of the Shepp-Logan phantom from the Radon Transform of the function from the example in Section 2.2. The Fourier reconstruction algorithm implemented in MatLab for this section is based on the algorithm presented on pages 126-127 of [1]. The coding is attributed to Dr. Faridani; code last revised July 6, 2013. We analyze the execution time, and the relative error for multiple experiments. The outcomes of this section, yet again, encourage additional exploration of modified FFTs for Fourier reconstruction methods.

Consider the function $f \in L_1(\mathbb{R}^2)$, again, defined by equation (6)/(23). Sample $g := \mathbf{R}f$ at (θ_j, s_l) for $j = 0, \ldots, p-1$ and $l = -q, \ldots, q-1$, where $\theta_j = \binom{\cos(\varphi_j)}{\sin(\varphi_j)}$ with $\varphi_j = \frac{2\pi j}{p}$ and $s_l = hl$ with $h = \pi^2/(p\sqrt{2})$. We evaluate this Fourier Reconstruction algorithm for the same varying values of p, q, and m as in Section 3.2 to allow quick comparison with Filtered Backprojection. Figure 6 is the $2q \times 2q$ pixel image produced for $q = 128, q = 400$, and $m = 0$ with an oversampling factor of 2 in Fourier space. The tables following Figure 6, however, list the experimental time for $m = 0$ and the relative error for $m = 3$ for varying values of p and q with no oversampling to compare with Filtered Backprojection.

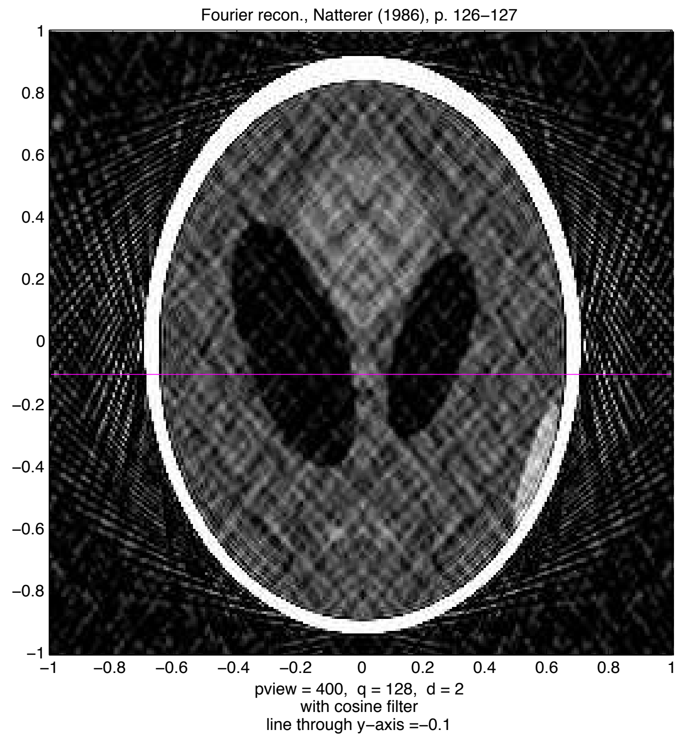
Fourier recon., Natterer (1986), p. 126–127
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
pview = 400, q = 128, d = 2
with cosine filter
line through y−axis =−0.1

Figure 6: Basic Fourier reconstruction image reconstructed with a cosine smoothing filter applied to $\hat{f}$ in between Steps 2 and 3 and oversampling factor of 2, created in MatLab; code attributed to Dr. Faridani; last revised: July 6, 2013

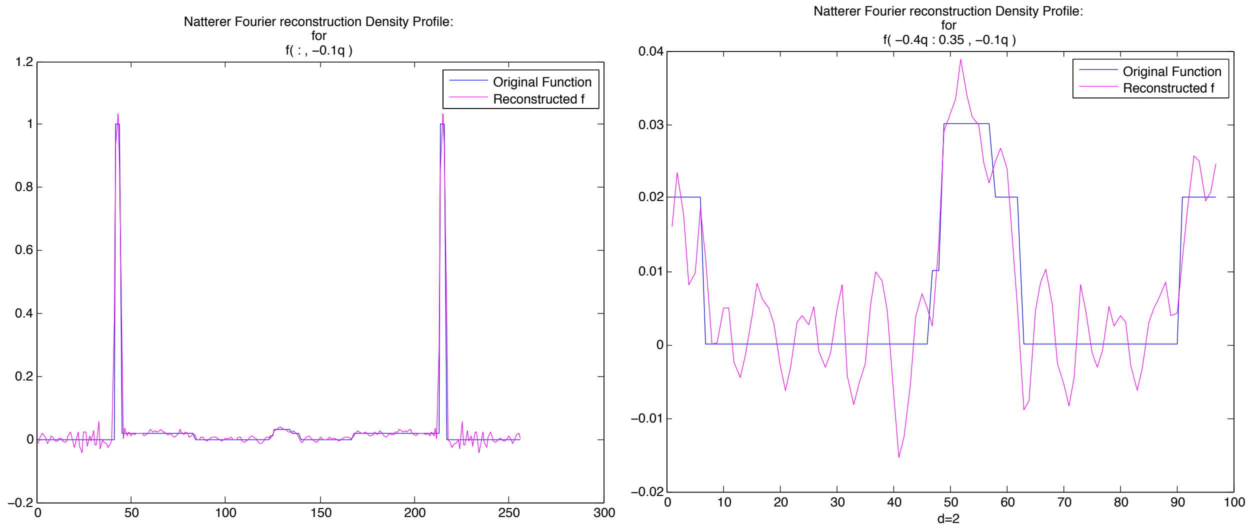

Figure 7: Density Profile corresponding to the line through the image in Figure 6. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; code attributed to Dr. Faridani.

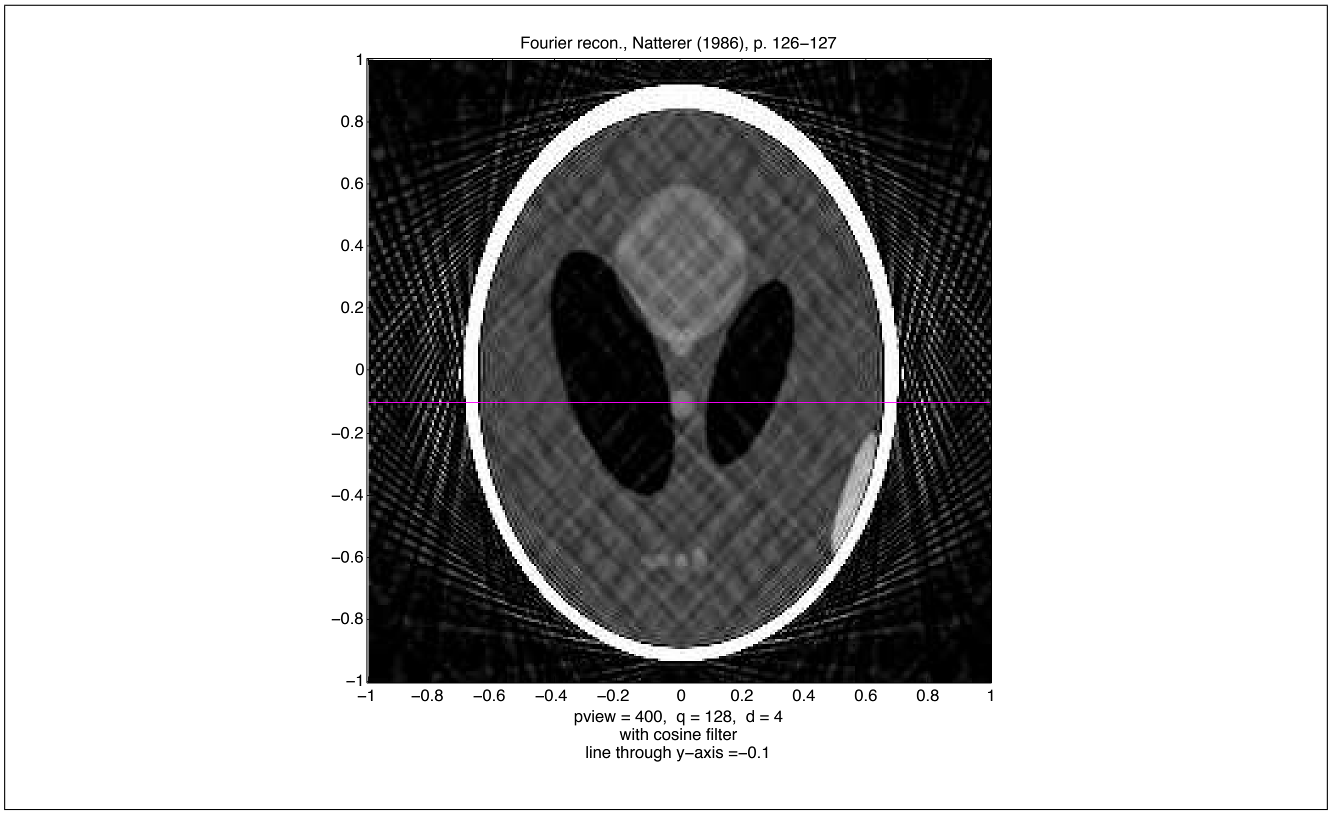
Fourier recon., Natterer (1986), p. 126–127
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
pview = 400, q = 128, d = 4
with cosine filter
line through y−axis =−0.1

Figure 8: Basic Fourier reconstruction image reconstructed with a cosine smoothing filter applied to $\hat{f}$ in between Steps 2 and 3 and oversampling factor of 4, created in MatLab; code attributed to Dr. Faridani; last revised: July 6, 2013

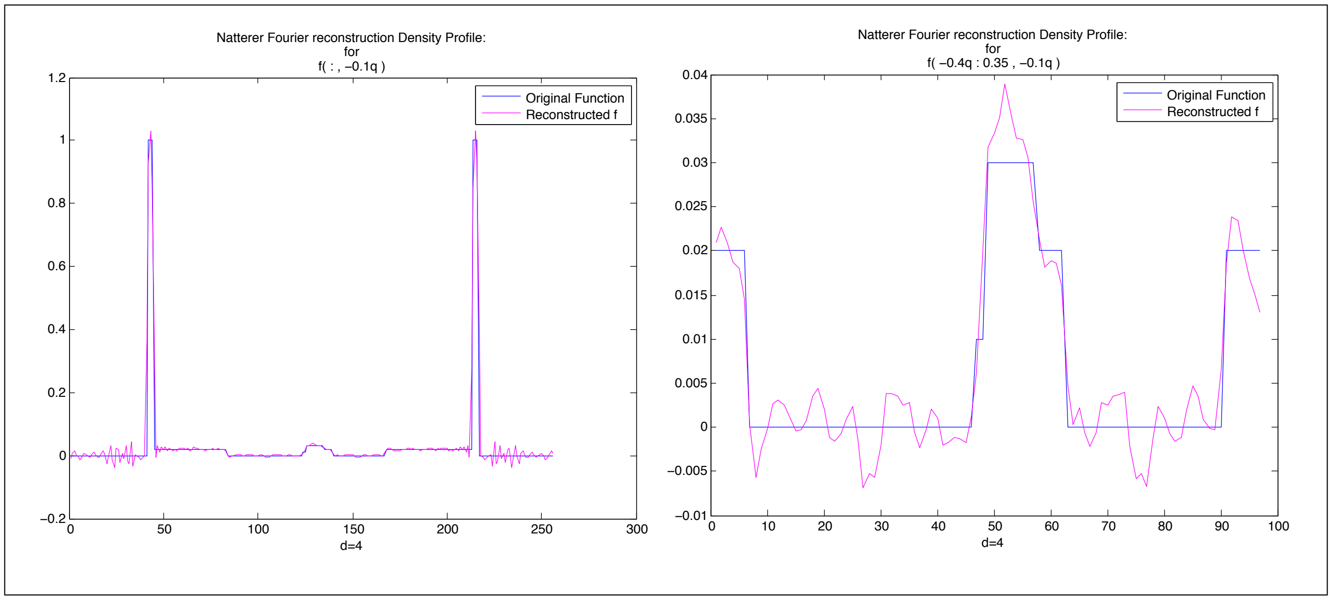

Figure 9: Density Profile corresponding to the line through the image in Figure 8. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; code attributed to Dr. Faridani.

For comparison with Filtered Backprojection:

Table 3: Time for $m = 0$: no oversampling (time in seconds)

p \ q	64	128	256	512
100	0.05	0.09	0.20	0.71
200	0.09	0.13	0.26	0.78
400	0.15	0.20	0.37	0.90
800	0.29	0.35	0.55	1.18

Table 4: Relative Error for $m = 3$: (x10$e-3$)

p \ q	64	128	256	512
100	12.16	12.03	12.03	12.03
200	8.17	7.92	7.92	7.92
400	3.15	2.32	2.32	2.32
800	2.49	1.27	1.26	1.26

For later comparisons:

Table 5: Time for $m = 0$: d = 2 (time in seconds)

p \ q	64	128	256
200	0.11	0.20	0.56
400	0.20	0.29	0.67
800	0.33	0.49	0.94

Table 6: Relative Error for $m = 3$: d = 2 (x10$e-3$)

p \ q	64	128	256
200	7.4	4.1	3.7
400	6.6	2.2	1.4
800	6.5	1.4	0.9

Table 7: Time for $m = 0$: d = 4 (time in seconds)

p \ q	64	128	256
200	0.19	0.49	1.62
400	0.28	0.61	1.82
800	0.46	0.86	2.24

Table 8: Relative Error for $m = 3$: d = 4 (x10$e-3$)

p \ q	64	128	256
200	6.8	2.6	1.9
400	6.5	2.0	0.9
800	6.4	1.8	0.7

Immediately, it is clear this Fourier reconstruction method is nowhere near comparable with Filtered Backprojection with regards to producing a useful image, even with an oversampling factor as large as 4. The image it constructs with an oversampling factor of 2 (see Figure 6), even for large p and q, is much too blurry and noisy to be serviceable in any real world application. This is even more evident by the large oscillations in the density profiles; these large oscillations are even *after* an additional smoothing filter - cosine; see Figure 7. In a broad sense, the image and the density profile indicate where different areas of density are, but with nowhere near the precision *and* accuracy of Filtered Backprojection for the same values of p and q. With the oversampling factor of 4, the image is better (see Figure 8) but still not as clear as Filtered backprojection.

However, the execution time for larger p and q is better, even with the oversampling factor of 4. As the sub-diagonal values of p and q increase by a factor of 2 the high-lighted execution time entries in Table 3 increase by a factor of about 3, less than 1/2 that of Filtered Backprojection.

Another indicator that Fourier Reconstruction methods warrant further exploration, is the size of the relative error in the $m = 3$ case even with no oversampling, see Table 4. Particularly for increasing values of p, these errors are only about twice that of the relative error for the same p and q of Filtered Backprojection, which, recall, were all below 1%. This can particularly be seen in density profiles of the image reproduced in the $m = 3$ case. Figure 10 shows the density profiles for this basic Fourier reconstruction algorithm when $m = 3$ compared with the same for Filtered Backprojection. This figure shows clearly that Fourier reconstruction holds promise as a useful reconstruction algorithm.

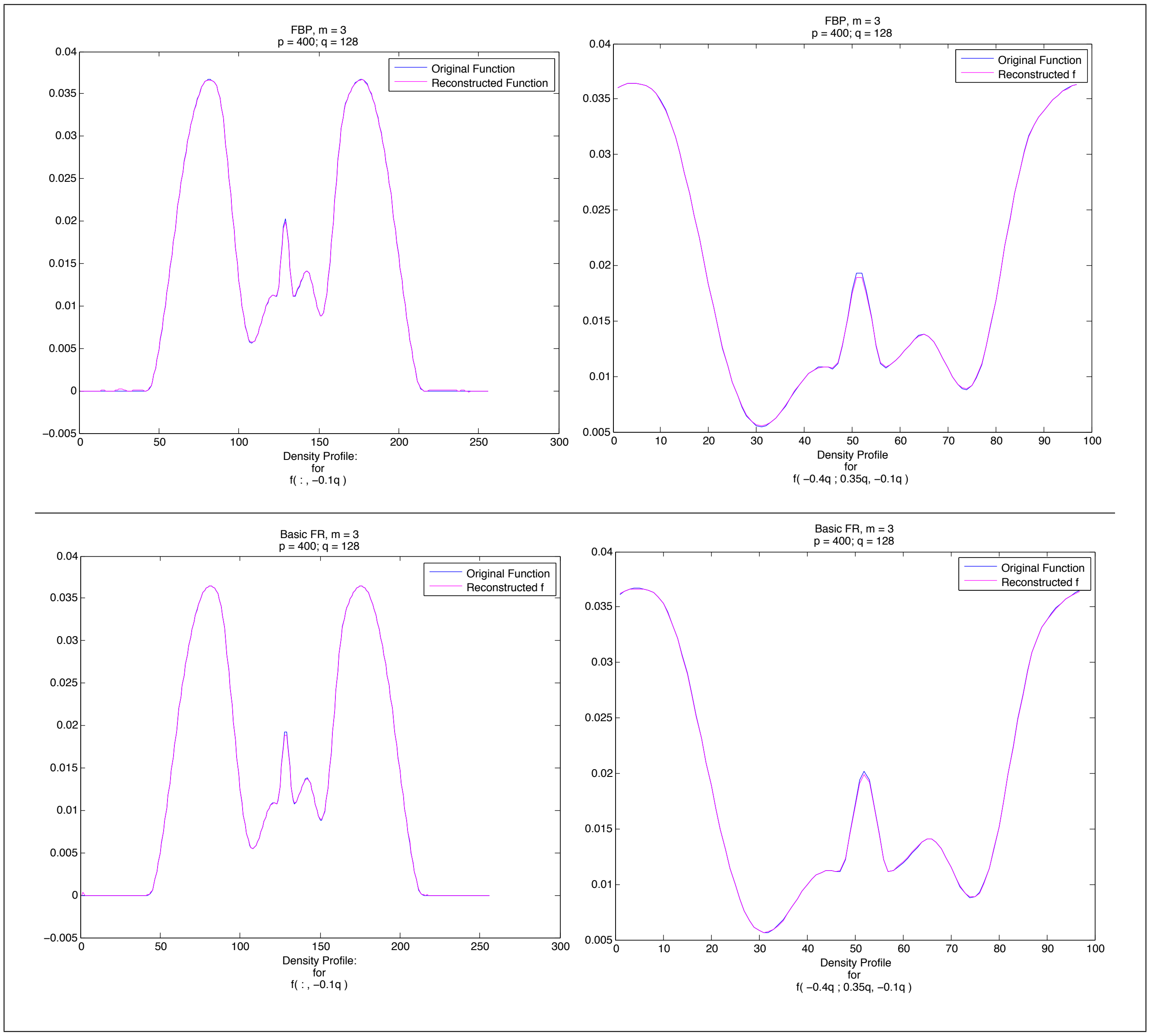

Figure 10: The top two figures show the density profile for Filtered Backprojection for $p = 400, q = 128$ and $m = 3$ along the same density line as shown in Figure 3. The bottom two show the same plots but for the basic Fourier reconstruction with no oversampling (and Figure 6).

The issue in Fourier reconstruction is the interpolation that takes place in Fourier space. Oversampling helps but is not the most ideal process. From Natterer, [1], straightforward nearest neighbor interpolation to a cartesian grid is not good enough. He also explains that theoretically, angular interpolation is "good enough" to reconstruct a useful image. We see, following these experiments which utilize an interpolation somewhere in between nearest neighbor and angular interpolation, that to obtain a useful image, some adjustments to the interpolation step must occur. The radial portion of the interpolation needs to be eliminated.

3.5 Conclusions from Filtered Backprojection and Basic Fourier Reconstruction

An ideal reconstruction method would produce an image as clear as Filtered Backprojection but execute with the speed of Fourier Reconstruction. This difference in the evaluation times can particularly be seen in Figure 11. This figure shows the relationship between q, when $p \simeq q$, and the operation time for Filtered Backprojection,

$$\begin{aligned}
\text{FBP time } &\simeq \mathcal{O}(q^3), \\
&\longrightarrow \\
\log(\text{FBP time }) &= \log(q^3), \\
\log(\text{FBP time }) &= 3\log(q).
\end{aligned}$$

Thus, a plot of $\log(q)$ versus log(FBP time) should have a slope of 3. Similarly, for Fourier reconstruction,

$$\begin{aligned}
\text{FR time } &\simeq \mathcal{O}(q^2\log(q)), \\
&\longrightarrow \\
\log(\text{FR time }) &= \log(q^2) + \underbrace{\log(\log(q))}_{=\text{small number}}, \\
\log(\text{FR time }) &= 2\log(q).
\end{aligned}$$

Thus, a plot of $\log(q)$ versus log(FR time) should have a slope of 2. Figure 11 shows our experiments support these relationships.

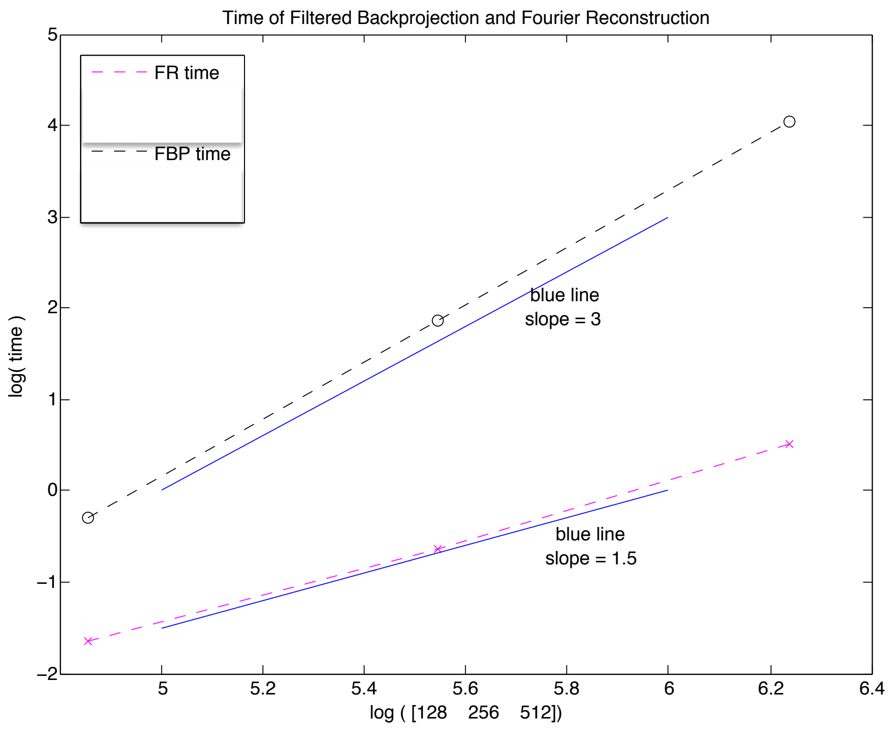

Figure 11: Plots of the log of q for $q = 128, 256$, and 512 versus the log of the implementation times for Filtered Backprojection and Basic Fourier reconstruction with no oversampling (see plot's key for more details).

The key issue with Filtered Backprojection is time; the key problem with Fourier reconstruction is the interpolation step in Fourier space. Thus, we move on and look at some modified Fourier reconstruction methods that attempt to quickly yet accurately produce a useful image.

4 Non-Uniform / Non-Equispaced Fast Fourier Transforms

In the upcoming sections, we explore the use of Non-Uniform Fast Fourier Transforms (NUFFTs) in Fourier reconstruction methods (NUFFTs are also sometimes called Non-Equispaced FFTs). NUFFTs amend FFTs so evaluation of non-equispaced/non-uniform input or output on a non-equispaced (non-uniform) grid is possible. Using an NUFFT allows less rigidity on data, but it introduces error (for additional information on NUFFTs see: [8], [9], [10], [11], and [12]).

The remainder of this paper aligns with Karsten Fourmont's paper, *Non-Equispaced Fast Fourier Transform with Applications to Tomography* [6]; additional background and information can be found in his dissertation [13]. The following sections present the same ideas as [6].

Fourmont presents two NUFFT methods, their numerical algorithms, and the application of a 2-dimensional method and a 1-dimensional method in Fourier reconstruction. We examine and carry out these algorithms. Note, the upcoming algorithms come from Fourmont's paper, [6]. However, the presentation here differs slightly from Fourmont's. We attempt to state the algorithms as clearly as possible, with the hope that algorithm recreation can be done with ease.

4.1 1-Dimensional NUFFTs: The NED and NER Cases

Quick evaluation of non-uniform spaced input data, or quick evaluation of uniform input data producing non-uniformly spaced output data is the main idea behind these types of Non-Uniform Fast Fourier Transforms. There are several interesting questions. Can we figure out how the Fourier Transforms of either non-uniform input data *or* uniform input data evaluated to non-uniform points relate to FFTs? And, is it possible to manipulate these Fourier Transform expressions to something able to be evaluated with FFTs? The answer to both of these questions is yes. A little mathematical theory and algebraic manipulation yield expressions, portions of which, can be evaluated with FFTs for both instances.

Fourmont discusses two types of NUFFTs. First, consider the two 1-dimensional versions. One type evaluates the Fourier Transform of equispaced data on a non-equispaced grid generating non-equispaced results. Fourmont refers to this as the NER case (non-equispaced *results*). The second type takes the Fourier Transform of data on a non-equispaced grid. This returns equispaced results; Fourmont calls this the NED case (non-equispaced *data*).

First, suppose we know z_k for $k = -N/2, \dots, N/2-1$, and we also know the nodes $x_l \in [-N/2, N/2-1]$, for $l = 1, \dots, M$, which are *not* necessarily equispaced (although equispacing is possible). We evaluate the discrete Fourier Transform of z_k at these non-equispaced nodes with

$$\hat{z}_l = \sum_{k=-N/2}^{N/2-1} z_k e^{-2\pi i x_l k/N}, \qquad l = 1, \dots, M, \tag{26}$$

via the 1-dimensional NER algorithm.

Alternatively, suppose we know $z_l := z(x_l)$, where the nodes $x_l \in [-N/2, N/2-1]$ are *not* necessarily equispaced, and we want the Fourier Transform of z_l at equispaced grid points over the set $\{-N/2, \dots, N/2-1\}$. Hence, we evaluate

$$\hat{z}_k = \sum_{l=1}^{M} z_l e^{-2\pi i x_l k/N}, \qquad k = -N/2, \dots, N/2-1, \tag{27}$$

via the 1-dimensional NED algorithm. Notice, these 1-dimensional expressions, (26) and (27), are transposes of one another, *and* if $x_l = l$ and $M = N$, then both expressions are DFTs.

The NER and the NED cases resemble FFTs. However, the exponentials in expressions (26) and (27) contain troublesome x_l's. These points are in $[-N/2, N/2-1]$, but we do *not* know if they uniformly partition the set. We rectify this issue by rewriting these complex exponentials as a sum of different complex exponentials. This is done using Fourmont's Proposition 1 which stems from Shannon's Sampling Theorem for finite bandwidth, bandlimited functions oversampled by a factor c within the bandwidth. The proposition provides a way to rewrite the previous exponentials with*out* x_l's.

Proposition 1. *Fourmont Proposition 1 [6]. Take $0 < \pi/c < \alpha$ as well as $\alpha < \pi(2 - 1/c)$. If ϕ is continuous and piecewise continuously differentiable on $[-\alpha, \alpha]$, non-zero on $[-\pi/c, \pi/c]$, and zero outside $[-\alpha, \alpha]$, then for $x \in \mathbb{R}$ and $|\xi| \leq \pi/c$,*

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

The hypotheses of Proposition 1 imply a little more than is immediately apparent. Consider, $0 < \pi/c < \alpha$ and $\alpha < \pi(2 - 1/c)$. This implies,

$$\begin{aligned} 0 < \pi/c &< 2\pi - \pi/c, \\ 0 < 1 &< 2c - 1, \\ 1/2 < 1 &< c. \end{aligned}$$

Thus, c *must* be a true *over*sampling factor. The proof of this proposition utilizes Fourier series as well as the Poisson Summation Formula [6].

Proof. Let c, α, and ϕ satisfy the hypotheses of Proposition 1. Take $x, \xi \in \mathbb{R}$ and consider the 2π-periodic function,

$$g(\xi) = \sum_{k=-\infty}^{\infty} \phi(\xi + 2\pi k) e^{-ix(\xi + 2k\pi)}.$$

Then,

$$g(\xi) = \sum_{m\in\mathbb{Z}} \hat{g}_m e^{-im\xi} \tag{28}$$

is the Fourier expansion of g with Fourier coefficients $\hat{g}_m$. Consider these Fourier coefficients,

$$\begin{aligned} \hat{g}_m &= (2\pi)^{-1} \int_{-\pi}^{\pi} g(\xi) e^{im\xi} d\xi, \\ &= (2\pi)^{-1} \int_{-\pi}^{\pi} \sum_{k=-\infty}^{\infty} \phi(\xi + 2\pi k) e^{-ix(\xi + 2k\pi)} e^{im\xi} d\xi. \end{aligned}$$

Suppose $|\xi| \leq \pi/c$, the smallest $\xi + 2\pi k$ can be for $k \neq 0$ is when $\xi = \pm\pi/c$ and $k = \mp 1$, respectively; in these cases we have $\mp\pi(2 - 1/c)$ which is either less than $-\alpha$ or greater than α, both of which result in $\phi(\xi + 2\pi k) = 0$. Note, even if $\pi/c < |\xi| < \alpha$ and, without loss of generality, suppose $\xi > 0$, then

$$\xi + 2\pi > \pi/c + 2\pi = \pi(2 + 1/c) > \alpha,$$

so $\phi(\xi + 2\pi k) = 0$ for $k > 0$. Also,

$$\xi - 2\pi < \pi/c - 2\pi = -\pi(2 - 1/c) < -\alpha,$$

so $\phi(\xi + 2\pi k) = 0$ for $k < 0$. Thus $\phi(\xi + 2\pi k)$ is only non-zero for $k = 0$ when $|\xi| < \alpha$. Therefore for $|\xi| \leq \pi/c$ ($|\xi| < \alpha$ in fact), we have $g(\xi) = \phi(\xi)e^{-ix\xi}$. Hence, since $\phi(\xi) = 0$ for $\xi > \alpha$ we can extend the limits of integration as well,

$$\begin{aligned}\hat{g}_m &= (2\pi)^{-1} \int_{-\infty}^{\infty} \phi(\xi) e^{-ix\xi} e^{im\xi} d\xi, \\ &= (2\pi)^{-1/2} \hat{\phi}(x - m).\end{aligned}$$

Combining this expression for $\hat{g}_m$ and equation (28), we obtain

$$\phi(\xi)e^{-ix\xi} = \sum_{m\in\mathbb{Z}} (2\pi)^{-1/2} \hat{\phi}(x - m) e^{-im\xi}, \tag{29}$$

when $|\xi| \leq \pi/c$, the desired result. □

Now we can revisit the 1-dimensional expressions (26) and (27) with the $e^{-2\pi i x_l k/N}$ terms. Let $\xi = 2\pi k/(cN)$ and $x = cx_l$ for $|k| \leq N/2$, then rewriting equation (26) with the substitution from Proposition 1 yields,

$$\hat{z}_l = (2\pi)^{-1/2} \sum_{m\in\mathbb{Z}} \hat{\phi}(cx_l - m) \sum_{k=-N/2}^{N/2-1} \frac{z_k}{\phi(2\pi k/(cN))} e^{-2\pi i m k/(cN)}, \qquad l = 1, \ldots, M. \tag{30}$$

This inner sum can be evaluated via an FFT, and the outer sum consists of few terms when ϕ is wisely chosen. Similarly, equation (27) can be rewritten,

$$\hat{z}_k = \frac{(2\pi)^{-1/2}}{\phi(2\pi k/(cN))} \sum_{m\in\mathbb{Z}} \sum_{l=1}^{M} z_l \hat{\phi}(cx_l - m) e^{-2\pi i m k/(cN)}, \qquad k = -N/2, \ldots, N/2 - 1. \tag{31}$$

Equation (31) requires some manipulation before evaluation via an FFT is clear. The presentation of this manipulation will directly precede the 1-dimensional NED algorithm forthcoming in Section 4.2.

Finally, before we move on to numerical implementation, we must choose an appropriate ϕ function. This choice for ϕ must satisfy the hypotheses in Proposition 1, *and* the Fourier Transform of ϕ should be almost negligible outside some bounded interval. This second constraint upon ϕ permits truncation of the summation over m in equations (30) and (31). A function that satisfies these stipulations is the Kaiser-Bessel Window [14]. The Kaiser-Bessel Window has compact support on $[-\alpha, \alpha]$ and $\hat{\phi}$ is insignificant outside an interval $[-K, K]$ (see Section 2 [6] [14]). The Kaiser-Bessel Window is,

$$\phi(\xi) = \begin{cases} I_0\left(K\sqrt{\alpha^2 - \xi^2}\right), & \text{if } |\xi| \leq \alpha, \\ 0, & \text{otherwise,} \end{cases} \tag{32}$$

$$\hat{\phi}(x) = \sqrt{\frac{2}{\pi}} \frac{\sinh\left(\alpha\sqrt{K^2 - x^2}\right)}{\sqrt{K^2 - x^2}}. \tag{33}$$

This choice of ϕ enables the summations over m in equations (30) and (31) to be truncated to a small subset of $\mathbb{Z}$, since for $x > K$ we see that $\hat{\phi}$ is insignificant (see Section 4.4 and Section 4 of [6] for error analysis).

It appears problems could arise with $\hat{\phi}$ if $x^2 > K^2$. However, when this happens both the numerator as well as the denominator become strictly imaginary. Hence, $\frac{\text{imag. \#}}{\text{imag. \#}} \to$ real #. Thus, $\hat{\phi}$ is a continuous real valued function on $\mathbb{R}$.

Dutt and Rokhlin, [8], present a very similar method where they use $\phi(x) = e^{-x^2/(4b)}$, i.e. Gaussian bells, as their window function, of width b. Steidl, [10], and Ware, [11] expound and modify Dutt and Rokhlin's error estimations to complete their method. We examined Greengard and Lee's presentation of this type of NUFFT in Dr. Faridani's Fall '12 Numerical Analysis course [9].

4.2 1-Dimensional NER and NED Algorithms

First, notice the 1-dimensional equations, (30) and (31), involve the *same* evaluations of ϕ and $\hat{\phi}$. Hence, the same pre-compupations will be useful. To generate these pre-computations we need to know α, K, c, N, and the x_l's.

We know $c > 1$ by Proposition 1, but if c is too large our algorithm will be inefficient; thus we choose $1 < c \leq 2$ (a particularly convenient choice is $c = 2$, because then cN is divisible by 2). Then, define

$$\alpha := \pi(2 - 1/c) - .01$$

to satisfy Proposition 1. K is determined by ϕ; it is the interpolation length and chosen to be 3 or 6. Fourmont's choice is based on whether single or double precision, respectively, is desired. However, we use MatLab to evaluate all the algorithms in this paper. MatLab is double precision, so *our* choice between 3 and 6 corresponds to the size of the error introduced by truncating the summation over m (see Section 4.4).

Thus, so long as we know the x_l's and N, the pre-computations are feasible . Typically, $N >> K$, i.e. $N \geq 32$, otherwise evaluation via FFT is not required. We pre-calculate ϕ, approximations of x_l's, and $\hat{\phi}$ as defined below.

- 1-dimensional NER/NED pre-computations:
  - for $k = -N/2, \ldots, N/2 - 1$,
    $\phi_k = \phi(2\pi k/(cN))$,
  - for $l = 1, \ldots, M$,
    $\mu_l = \text{round}(cx_l)$,
  - for $l = 1, \ldots, M$,
    for $m = -K, \ldots, K$,
    $\hat{\phi}_{l,m} = (2\pi)^{-1/2}\hat{\phi}(cx_l - (\mu_l + m))$,

In the final pre-computation, we only look at $\hat{\phi}_{l,m}$ for $|m| \leq K$ because by definition $\hat{\phi}(x)$ is negligible for $|x| > K$. This implies $\hat{\phi}(cx_l - (\mu_l + m))$ is insignificant for $|cx_l - \mu_l - m| > K$. If $|m| \geq K + 1$, then

$$|cx_l - \mu_l - m| \geq |m| - \underbrace{|cx_l - \mu_l|}_{\leq 1/2} > K + 1/2 > K.$$

So, we set $\hat{\phi}_{l,m} = 0$ for $|m| > K$. Now we are ready to develop the NER and NED algorithms for fast computation of the 1-dimensional equations, (30) and (31).

First, consider the 1-dimensional expression (30). Since ϕ was chosen carefully, we may assume $\hat{\phi}$ is insignificant outside the interval $[-K, K]$, therefore $\hat{\phi}_{l,m} = 0$ for $|m| > K$. Thus, the 1-dimensional NER algorithm computes,

$$\begin{aligned}\hat{z}_l &= (2\pi)^{-1/2} \sum_{m \in \mathbb{Z}} \hat{\phi}(cx_l - m) \sum_{k=-N/2}^{N/2-1} \frac{z_k}{\phi(2\pi k/(cN))} e^{-2\pi imk/(cN)}, \qquad l = 1, \ldots, M, \\ &= (2\pi)^{-1/2} \sum_{m \in \mathbb{Z}} \hat{\phi}(cx_l - (\mu_l + m)) \sum_{k=-N/2}^{N/2-1} \frac{z_k}{\phi(2\pi k/(cN))} e^{-\pi i(\mu_l + m)k/(cN/2)}, \qquad l = 1, \ldots, M.\end{aligned}$$

With our pre-computations and assumptions regarding $\hat{\phi}$, the above can be rewritten,

$$\hat{z}_l \simeq \sum_{m=-K}^{K} \hat{\phi}_{l,m} \sum_{k=-N/2}^{N/2-1} \frac{z_k}{\phi_k} e^{-\pi i(\mu_l + m)k/(cN/2)}, \qquad l = 1, \ldots, M. \tag{34}$$

Let $z_k = 0$ for $k < -N/2$ and $k > N/2 - 1$, and define $u_k = z_k/\phi_k$ for $k = -cN/2, \ldots, cN/2 - 1$. Let $j := -cN/2, \ldots, cN/2 - 1$ and consider,

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

which is equivalent to the inner sum of (34) for $j = (\mu_l + m) \mod cN$, by properties of the complex exponential; additionally this sum can be evaluated with an FFT. Then, let $j := \mu_l + m$ with $\mu_l + m$ cN-periodic. Therefore,

$$\hat{z}_l \simeq \sum_{m=-K}^{K} \hat{\phi}_{l,m} U_{\mu_l+m}, \qquad l = 1, \ldots, M,$$

which interpolates U_{μ_l+m} to the non-uniform grid, producing the desired non-equispaced results.

Algorithm 1 1D NER (Section 3 [6]): Fast computation of

$$\hat{z}_l = \sum_{k=-N/2}^{N/2-1} z_k e^{-\pi i x_l k/(N/2)}, \qquad l = 1, \ldots, M.$$

Parameters: Oversampling factor c, interpolation length K, and pre-computations: ϕ_k, μ_l, and $\hat{\phi}_{l,m}$
Input: z_k for $k = -N/2, \ldots N/2 - 1$
.

Step 1: Adjust size of z_k and scale to allow utilization of an FFT,

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

Step 2: Compute an FFT of length cN,

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

Step 3: Interpolate back to the nonequispaced grid,

$$\hat{z}_l = \sum_{m=-K}^{K} \hat{\phi}_{l,m} U_{\mu_l+m}, \qquad l = 1, \ldots, M,$$

where we use that $U_j = U_{j \pm cN}$ in case $\mu_l + m$ is outside $[-cN/2, \ldots, cN/2 - 1]$.

Next, consider the 1-dimensional expression (31). Again, by choice of ϕ, we assume $\hat{\phi}$ is insignificant outside $[-K, K]$. Therefore $\hat{\phi}_{l,m} = 0$ for $|m| > K$. Obtaining an expression that can be evaluated by the 1-dimensional NED algorithm requires reindexing of equation (31). Consider,

$$\begin{aligned}\hat{z}_k &= \frac{(2\pi)^{-1/2}}{\phi(2\pi k/(cN))}\sum_{m\in\mathbb{Z}}\sum_{l=1}^{M} z_l\hat{\phi}(cx_l - m)e^{-2\pi imk/(cN)}, \qquad k = -N/2,\dots,N/2-1,\\ &= \frac{(2\pi)^{-1/2}}{\phi(2\pi k/(cN))}\sum_{m\in\mathbb{Z}}\sum_{l=1}^{M} z_l\hat{\phi}(cx_l - (\mu_l + m))e^{-\pi i(\mu_l+m)k/(cN/2)}, \qquad k = -N/2,\dots,N/2-1.\end{aligned}$$

Inserting the pre-computations, and truncating the sum over m,

$$\hat{z}_k \simeq \frac{1}{\phi_k}\sum_{l=1}^{M}\sum_{m=-K}^{K} z_l\hat{\phi}_{l,m}e^{-\pi i(m+\mu_l)k/(cN/2)}, \qquad k = -N/2,\dots,N/2-1. \tag{35}$$

The formulation of this expression, is not yet able to be evaluated with an FFT.

Assume $\hat{\phi}$, z_l, and μ_l zero outside their region of definition. Consider the slightly more general expression of equation (35), where $j := m + \mu_l$,

$$\hat{z}_k = \frac{1}{\phi_k}\sum_{j\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} z_l\hat{\phi}_{l,j-\mu_l}e^{-\pi ijk/(cN/2)}, \qquad k = -N/2,\dots,N/2-1,$$

Now, $j \in \mathbb{Z}$ so there exists $j' \in \{-cN/2,\dots,cN/2-1\}$ such that $j = j' + c\tilde{m}N$ for some $\tilde{m} \in \mathbb{Z}$. Thus,

$$\hat{z}_k = \frac{1}{\phi_k}\sum_{j'=-cN/2}^{cN/2-1}\sum_{\tilde{m}\in\mathbb{Z}}\sum_{l\in\mathbb{Z}} z_l\hat{\phi}_{l,j'+c\tilde{m}N-\mu_l}e^{-\pi i(j'+c\tilde{m}N)k/(cN/2)}, \qquad k = -N/2,\dots,N/2-1,$$

and by properties of the complex exponential, $e^{-\pi i(j'+c\tilde{m}N)k/(cN/2)} = e^{-2\pi ij'k/(cN/2)}$ for any $\tilde{m} \in \mathbb{Z}$.

Therefore, recalling $z_l \neq 0$ only when $l = 1,\dots,M$, define

$$\begin{aligned}u_{j'} &:= \sum_{l=1}^{M}\sum_{\tilde{m}\in\mathbb{Z}} z_l\hat{\phi}_{l,j'+c\tilde{m}N-\mu_l}, \qquad j' = -cN/2,\dots,cN/2-1,\\ &= \sum_{l=1}^{M} z_l\hat{\phi}_{l,(j'-\mu_l) \mod cN}, \qquad j' = -cN/2,\dots,cN/2-1.\end{aligned}$$

Hence, if we let $m_{j',l} := (j' - \mu_l) \mod cN$, then

$$u_{j'} = \sum_{l=1}^{M} z_l\hat{\phi}_{l,m_{j',l}}, \qquad j' = -cN/2,\dots,cN/2-1, \tag{36}$$

where, by definition and previous argument, $\hat{\phi}_{l,m_{j',l}} = 0$ for $|m_{j',l}| > K$, i.e. $(|j' - \mu_l| \mod cN) > K$.

Then the final discrete 1-dimensional expression the 1-dimensional NED algorithm evaluates for $\hat{z}_k$ is,

$$\hat{z}_k \simeq \frac{1}{\phi_k}\sum_{j'=-cN/2}^{cN/2-1} u_{j'}e^{-\pi ikj'/(cN/2)}, \qquad k = -N/2,\dots,N/2-1, \tag{37}$$

with $u_{j'}$ as defined in equation (36). Hence, this sum over j' can be evaluated via an FFT.

Algorithm 2 1D NED (Section 3 [6]): Fast computation of

$$\hat{z}_k = \sum_{l=1}^{M} z_l e^{-\pi i x_l k/(N/2)}, \qquad k = -N/2, \dots, N/2-1.$$

Parameters: Oversampling factor c, interpolation length K, and pre-computations: ϕ_k, μ_l, and $\hat{\phi}_{l,m}$
Input: z_l for $l = 1, \dots M$
.

Step 1: Each z_l contributes to the $2K+1$ nodes closest by; compute those contributions,

```
for $l = 1, \dots, M$,
    for $m = -K, \dots, K$,
```

$$u_{\mu_l+m} \leftarrow u_{\mu_l+m} + z_l \hat{\phi}_{l,m},$$

where $\mu_l + m$ is cN-periodic.

Step 2: Let $j := \mu_l + m$, compute an FFT of length cN,

$$U_k = \sum_{j=-cN/2}^{cN/2-1} u_j e^{-i\pi kj/(cN/2)}, \qquad k = -cN/2, \dots, cN/2-1.$$

Step 3: Scale the result,

$$\hat{z}_k = U_k/\phi_k, \qquad k = -N/2, \dots, N/2-1 \; ^*.$$

*In this final step we only want $k = -N/2, \dots, N/2-1$ although Step 2 computed U_k for $k = -cN/2, \dots, cN/2-1$.

4.3 2-Dimensional NED Algorithm

The last type of NUFFT we consider is the 2-dimensional equivalent of the NED type NUFFT. Fourmont does not fully present this algorithm, but he uses it in one of his applications to tomography. We present it here.

The extension of the 1-dimensional NED expression to a 2-dimensional equivalent is intuitive. Take z_j on $\mathbb{R}^2$ located at non-equispaced node $(x_j, y_j) \in [-N/2, N/2-1] \times [-N/2, N/2-1]$, $j = 1, \dots, M$. The Fourier Transform of z_j at these points is,

$$\hat{z}_{k,l} = \sum_{j=1}^{M} z_j e^{-2\pi i(kx_j + ly_j)/N}, \qquad l, k = -N/2, \dots, N/2-1. \tag{38}$$

Notice, there is only one summation over the single index j despite the fact that we are now in 2-dimensions. This is because before proceeding with anything else, we vectorize the matrix containing all the input data point z's and their corresponding x and y node point matrices.

Vectorization of a matrix is done with the command $z(:)$ in MatLab if z was the matrix containing the input data. For example, if the first data point $z(1,1) = 0$ is located at node $(1,1) \in \mathbb{R}^2$, then the matching vectorized components would be: $z(1) = 0, x(1) = 1$, and $y(1) = 1$.

With this done, we return to the 2-dimensional equation (38) and utilize Proposition 1 twice:

once with $\xi := 2\pi k/cN$ and $x := cx_j$, and once with $\xi := 2\pi l/cN$ and $x := cy_j$. Then,

$$\begin{aligned}
\hat{z}_{k,l} &= \sum_{j=1}^{M} z_j e^{-2\pi i(kx_j+ly_j)/N}, \qquad k,l=-N/2,\ldots,N/2-1,\\
&= \sum_{j=1}^{M} z_j e^{-2\pi ikx_j/N} e^{-2\pi ily_j/N}, \qquad k,l=-N/2,\ldots,N/2-1,\\
&= \sum_{j=1}^{M} \left[z_j \Big(\frac{(2\pi)^{-1/2}}{\phi(2\pi k/(cN))} \sum_{m\in\mathbb{Z}} \hat{\phi}(cx_j-m) e^{-2\pi ikm/(cN)} \Big) \Big(\frac{(2\pi)^{-1/2}}{\phi(2\pi l/(cN))} \sum_{n\in\mathbb{Z}} \hat{\phi}(cy_j-n) e^{-2\pi iln/(cN)} \Big) \right],\\
&\qquad\qquad k,l=-N/2,\ldots,N/2-1.
\end{aligned}$$

The pre-computations in 2-dimensions are only slightly more complicated than in 1-dimension. Let, ϕ, and thus $\hat{\phi}$, be the same 1-dimensional Kaiser-Bessel Window defined in equations (32) and (33). In this 2-dimensional type NUFFT, the pre-computed ϕ is a product of two 1-dimensional ϕ's, and the pre-computed $\hat{\phi}$ is a product of two 1-dimensional $\hat{\phi}$'s.

- 2-dimensional NED pre-computations:
  - for $k=-N/2,\ldots,N/2-1$,
    for $l=-N/2,\ldots,N/2-1$,
    $\phi_{k,l} = \phi(2\pi k/(cN)) \cdot \phi(2\pi l/(cN))$,
  - for $j=1,\ldots,M$,
    $\mu_j = \text{round}(cx_j)$,
    $\nu_j = \text{round}(cy_j)$,
  - for $j=1,\ldots,M$,
    for $m=-K,\ldots,K$,
    for $n=-K,\ldots,K$,
    $\hat{\phi}_{j,m,n} = (2\pi)^{-1}\hat{\phi}(cx_j-(\mu_j+m)) \cdot \hat{\phi}(cy_j-(\nu_j+n))$.

Notice, by assumptions on ϕ, that for each j, outside $[-K,K]\times[-K,K]$ we may again assume $\hat{\phi}_{j,m,n}$ is insignificant. (Also, it is possible to have the third pre-computation only store: $\hat{\phi}_{j,m} := (2\pi)^{-1/2}\hat{\phi}(cx_j-(\mu_j+m))$, $m=-K,\ldots,K$ and $\hat{\phi}_{j,n} := (2\pi)^{-1/2}\hat{\phi}(cy_j-(\nu_j+n))$, $n=-K,\ldots,K$ for each j.)

We now have the tools to present the 2-dimensional algorithm. The 2-dimensional NED algorithm computes

$$\hat{z}_{k,l} \simeq \frac{1}{\phi_{k,l}} \sum_{j=1}^{M} \sum_{m=-K}^{K} \sum_{n=-K}^{K} z_j \hat{\phi}_{j,m,n} e^{-2\pi i(k(\mu_j+m)+l(\nu_j+n))/(cN)}, \qquad k,l=-N/2,\ldots,N/2-1. \tag{39}$$

Before presenting the algorithm that quickly computes the 2-dimensional equation (39), some index manipulation, similar to the 1-dimensional case, must take place. Note, z, x, and y are all in vector form, so the reindexing process is nearly identical to the 1-dimensional case reindexing. Thus, by the exact some process used to arrive at equation (37) – once with respect to m, where $v := \mu_j + m$ and once with respect to n, where $w := \nu_j + n$ – we arrive at the 2-dimensional expression, that we evaluate via the 2-dimensional NED algorithm.

$$\hat{z}_{k,l} \simeq \frac{1}{\phi_{k,l}} \sum_{v'=-cN/2}^{cN/2-1} \sum_{w'=-cN/2}^{cN/2-1} u_{v',w'} e^{-2\pi i(kv'+lw')/(cN)}, \qquad k,l=-N/2,\ldots,N/2-1, \tag{40}$$

where

$$u_{v',w'} = \sum_{j=1}^{M} z_j \hat{\phi}_{j,m_{v',j},n_{w',j}}, \qquad v', w' = -cN/2, \ldots, cN/2-1,$$

with $m_{v',j} := (v' - \mu_j) \mod cN$ and $n_{w',j} := (w' - \nu_j) \mod cN$ where $v', w' = -cN/2, \ldots, cN/2-1$. Additionally, $\hat{\phi}_{j,m_{v',j},n_{w',j}} \neq 0$ only if $|m_{v',j}|, |n_{w',j}| \leq K$ which is if and only if ($|v' - \mu_j| \mod cN$) and ($|w' - \nu_j| \mod cN$) are both $\leq K$.

Algorithm 3 2D NED (End of Section 3 [6]): Fast computation of

$$\hat{z}_{k,l} = \sum_{j=1}^{M} z_j e^{-2\pi i(kx_j + ly_j)/N}, \qquad k, l = -N/2, \ldots, N/2-1.$$

Parameters: Oversampling factor c, interpolation length K, and pre-computations: $\phi_{k,l}$, μ_j, ν_j, and $\hat{\phi}_{j,m,n}$

Input: z_j for $j = 1, \ldots, M$

.

Step 1: Each z_j contributes to the $(2K+1)^2$ nodes in the $(2K+1) \times (2K+1)$ grid in $\mathbb{Z}^2$ centered at z_j. Compute these contributions,

```
for $j = 1, \ldots, M$,
    for $m, n = -K, \ldots, K$,
```

$$u_{\mu_j+m,\nu_j+n} \leftarrow u_{\mu_j+m,\nu_j+n} + z_j \hat{\phi}_{j,m,n},$$

where $\mu_j + m$ and $\nu_j + n$ are cN-periodic.

Step 2: Let $v := \mu_j + m$ and $w := \nu_j + n$, compute a 2-dimensional FFT,

$$U_{k,l} = \sum_{v=-cN/2}^{cN/2-1} \sum_{w=-cN/2}^{cN/2-1} u_{v,w} e^{-2i\pi(kv+lw)/(cN)}, \qquad k, l = -cN/2, \ldots, cN/2-1.$$

Step 3: Scale the result,

$$\hat{z}_{k,l} = U_{k,l}/\phi_{k,l}, \qquad k, l = -N/2, \ldots, N/2-1\ ^*.$$

*In the final step we only consider $k, l = -N/2, \ldots, N/2-1$ whereas Step 2 computed $U_{k,l}$ for $k, l = -cN/2, \ldots, cN/2-1$.

4.4 Error Estimates

Before applying any of the NUFFT algorithms to Computerized Tomography, it is important to check that the discretization and truncation process did *not* introduce too much error. We evaluate the error caused by truncating the expanded version of $e^{-ix\xi}$ obtained from Proposition 1 while using the Kaiser-Bessel Window, equations (32) and (33) [6].

Proposition 2. *Fourmont's Proposition 2 [6]. Under the same assumptions as in Proposition 1 and ϕ and $\hat{\phi}$ defined by equations* (32), (33), *then*

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

holds true for all $K \leq 15$.

Proof. By Proposition 1, the approximate error caused by truncating the sum for $|m-x| > K$ is,

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

For each term in the summation, the numerator and the denominator of $\hat{\phi}$ become imaginary. Hence,

$$\hat{\phi}(x-m) = \sqrt{\frac{2}{\pi}}\ \frac{\sin\big(\alpha\sqrt{(x-m)^2-K^2}\big)}{\sqrt{(x-m)^2-K^2}} \tag{41}$$

which for $|x-m| >> K$ acts like $\sin(x-m)/(x-m)$. This does not converge absolutely, but can be bounded using the following lemma.

Lemma 1. *Fourmont Lemma 1. Let* $0 < K \leq 15$ *and* $\alpha \leq 2\pi$, *then*

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

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

Proof. See [13]. □

Thus,

$$\text{error} = \frac{(2\pi)^{-1/2}}{\phi(\xi)} \left| \sum_{|m-x|>K} \hat{\phi}(x-m)\ e^{-im\xi} \right| \leq \frac{(2\pi)^{-1/2}}{\phi(\xi)} \sqrt{\frac{2}{\pi}}\ 30.$$

Notice, $\phi(\xi)$ is minimized by $\xi = \pi/c$; hence error is maximized when $\xi = \pi/c$. So,

$$\text{error} \leq \frac{(2\pi)^{-1/2}}{\phi(\pi/c)} 30 = \frac{30}{\pi I_0\big(K\sqrt{\alpha^2-(\pi/c)^2}\big)}$$

□

From the graphs of the error estimate in Figure 12, it is clear that increasing K by one results in noticeable accuracy improvements for each increase. Increasing c from $3/2-4$ improves of the error estimate as well, but at much higher efficiency costs than increasing K. Thus, we remain with $1 < c \leq 2$ for the oversampling factor. For K, 6 is a good choice; it ensures the truncation error is small (see Figure 12), and keeps the truncated sum to only 13 terms.

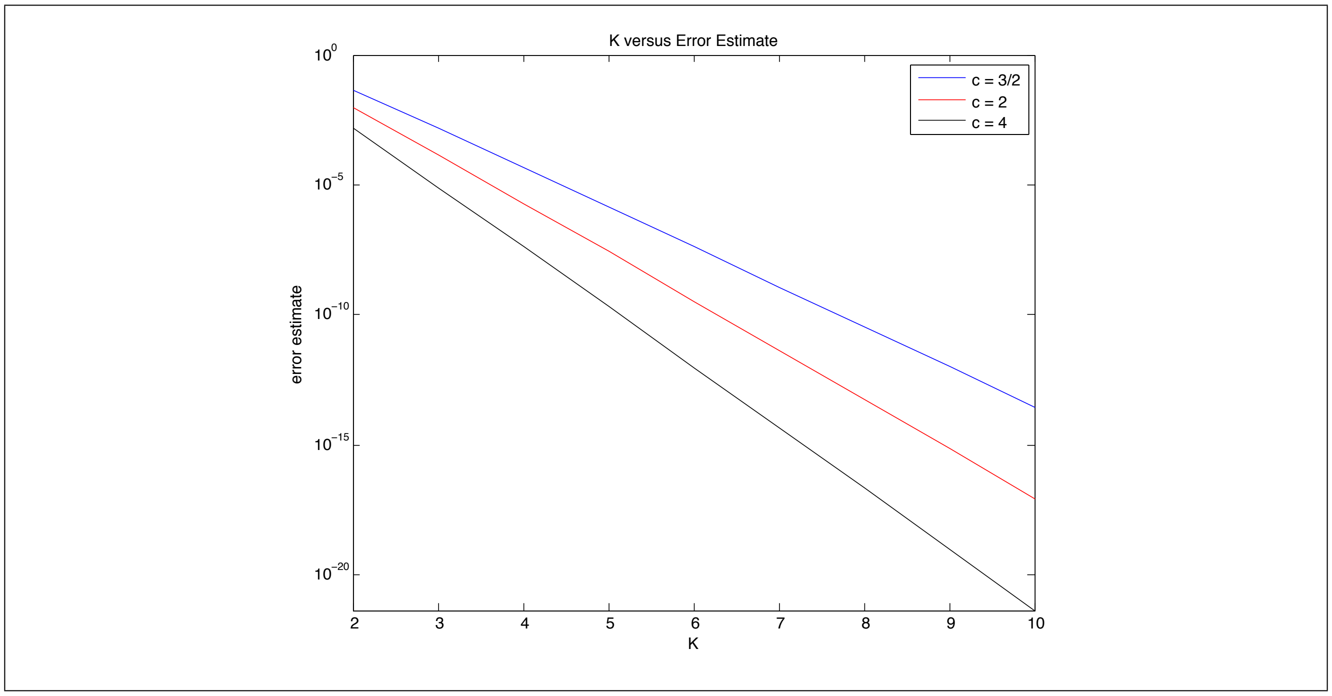

Figure 12: Plot of $K = 2:1:10$ versus corresponding error estimates for $c = 3/2, 2$, and 4 (4 is included for comparison sake).

5 Application of NUFFTs to Tomography

In the remaining sections, we present two different Fourier reconstruction methods for image reconstruction of a function f. Each uses a different NUFFT algorithm presented in the preceding sections. These reconstruction algorithms are presented in Sections 5.1 and 5.2 of Fourmont's paper [6]. Here, we present these algorithms with a few modifications. Lastly, we implement these algorithms and examine the images produced, the relative error, and the execution time.

Take $f(x)$, $x \in \mathbb{R}^2$, such that $f(x) = 0$ for all $|x| \geq 1$. Consider $\mathbf{R}f$, the Radon Transform of f. Sample $g := \mathbf{R}f$ using parallel beam scanning with $\theta = \theta_j$ and $s = s_l$ where

$$\theta_j = \begin{pmatrix} \cos(\varphi_j) \\ \sin(\varphi_j) \end{pmatrix}, \qquad \varphi_j = \frac{j\pi}{p}, \qquad j = 0, \ldots, p,$$
$$s_l = l/q, \qquad \text{and} \quad l = -q, \ldots, q-1.$$

We know if f has an essential bandwidth of b, then we can reasonably reconstruct f from this set of sampled data, $g(\theta_j, s_l)$, so long as $p \geq b$ and $q \geq b/\pi$ (see [1] III.3 for proof). Remember the relationship the Projection Slice Theorem, Theorem 2.2, gives between $\hat{g}$ and $\hat{f}$,

$$\hat{g}(\theta, \sigma) = (2\pi)^{1/2} \hat{f}(\sigma\theta), \qquad \theta \in S^1, \ \sigma \in \mathbb{R}.$$

First, we consider the application of a 2-dimensional NED to Fourier reconstruction. Second, we apply p, 1-dimensional NERs to Fourier reconstruction (i.e. we implement the 1-dimensional NER p times). This first type, is known as "gridding" (for more articles on gridding methods and some applications to CT please see: Brouw [15], O'Sullivan [16], Kaveh and Soumekh [17], Schomberg and Timmer [18], and Greengard and Lee [9]). Fourmont, specifically, cites O'Sullivan's gridding method [6],[16]. Fourmont claims his second type of Fourier reconstruction is a new approach (at the time of publication) and states it "is even faster than gridding while still producing good reconstruction quality" [6].

5.1 Application of the 2-dimensional NED to Tomography: "Gridding"

Let $f(x) = 0$ for $|x| \geq 1, x \in \mathbb{R}^2$. Consider, using the Fourier Inversion Theorem (Theorem 2.1), polar reparametrization, and the Projection Slice Theorem (Theorem 2.2),

$$\begin{aligned} f(x) &= (2\pi)^{-1} \int_{\mathbb{R}^2} \hat{f}(\xi) e^{ix\cdot\xi} d\xi, \\ &= (2\pi)^{-1} \int_0^\infty \int_0^{2\pi} \hat{f}(\sigma\theta) e^{ix\cdot\sigma\theta} d\varphi \ |\sigma| \ d\sigma, \\ &= (2\pi)^{-1} \int_0^\infty |\sigma| \int_0^{2\pi} (2\pi)^{-1/2} \hat{g}(\theta,\sigma) e^{ix\cdot\sigma\theta} d\varphi \ d\sigma. \end{aligned}$$

Recall, $\hat{g}(\theta,\sigma) = \hat{g}(-\theta,-\sigma)$. Therefore we get the following equation for f,

$$f(x) = \frac{1}{2}(2\pi)^{-3/2} \int_{-\infty}^{\infty} \int_0^{2\pi} \hat{g}(\theta,\sigma) \ |\sigma| \ e^{ix\cdot\sigma\theta} d\varphi \ d\sigma \tag{42}$$

Equation (42) requires careful and accurate discretization before we can move on to a gridding algorithm (see Section 5.1 [6]).

The integrand in equation (42) is a periodic function in φ; thus, the trapezoidal rule is applicable with minimal error. Consider, $\hat{g}(\theta_j, \sigma_r)$ with θ_j, $j = 0, \dots, p-1$, defined at the start of Section 5 and $\sigma_r := \pi r/d$, $r = -dq, \dots, dq-1$, with d an oversampling factor, then

$$\int_{-\infty}^{\infty} |\sigma| \ \dots \ d\sigma \to \frac{\pi}{d} \sum_{r=-dq}^{dq-1} \left|\frac{\pi r}{d}\right| \ \dots = \frac{\pi^2}{d^2} \sum_{r=-dq}^{dq-1} |r|,$$

$$\text{and} \quad \int_0^{2\pi} \dots \ d\varphi \to \frac{2\pi}{2p} \sum_{j=0}^{2p-1} \dots = \frac{\pi}{p} \sum_{j=0}^{2p-1} \dots \ .$$

Hence, for $x = \frac{1}{q}k$, $k \in \mathbb{R}^2$ and $\tilde{\sigma}_r := |r|$,

$$f\left(\frac{1}{q}k\right) = \frac{1}{2(2\pi)^{3/2}} \frac{\pi^3}{pd^2} \sum_{r=-dq}^{dq-1} \sum_{j=0}^{2p-1} \tilde{\sigma}_r \ \hat{g}(\theta_j, \pi r/d) \ e^{i(k/q)\cdot(\theta_j \pi (r/d))}.$$

Define $\xi_{j,r} := \frac{1}{d} r\theta_j$, then the preceding line becomes

$$f\left(\frac{1}{q}k\right) = \frac{\pi^{3/2}}{4\sqrt{2}pd^2} \sum_{r=-dq}^{dq-1} \sum_{j=0}^{2p-1} \tilde{\sigma}_r \ \hat{g}(\theta_j, \pi r/d) \ e^{i\pi k\cdot\xi_{j,r}/q}. \tag{43}$$

With this discretization, terms involved with $\tilde{\sigma}_0 = 0$ would be zero. We do not want this because these terms contain important information about $\hat{g}(\theta_j, 0) \simeq \hat{f}(0)$, that is, the average value of f. If left zero, this can result in a shift in the output. Fourmont takes "$\tilde{\sigma}_0 = 1/10$, $\tilde{\sigma}_{\pm1} = 0.98$, $\tilde{\sigma}_r = |r|$ for $|r| > 1$ and [he is] satisfied with the results." But, he states "a more detailed analysis is outstanding"[6]. Through numerical experimentation with MatLab, we found $\tilde{\sigma}_0 = .2$ to be a better choice. Thus, the implementation of the upcoming algorithm for image reconstruction via gridding, uses: $\tilde{\sigma}_0 = .2$, $\tilde{\sigma}_{\pm1} = .98$, and $\tilde{\sigma}_r = |r|$ for all $|r| \geq 2$, for the following algorithms for image reconstructions (for experimentation results see [19]).

This discrete equation for f, equation (43), as well as how the $\xi_{j,l}$'s are defined, enable the application of the 2-dimensional NED algorithm.

Algorithm 4 Gridding via 2D NED (Section 5.1 [6]):

Parameters: Parallel Beam scanning geometry p, q, oversampling factor d, and parameters/pre-computations for the 2-dimensional NED (dependent upon p, q, d and $\xi_{j,r}$ from Step 3)

Input: $g_{j,l} = g(\theta_j, s_l)$, $j = 0, \dots, p-1$ and $l = -q, \dots q-1$ for $g := \mathbf{R}f$
.

Step 1: Compute p, 1-dimensional Fourier Transforms with respect to the second variable of $g_{j,l}$. This can be done via an FFT with $g_{j,l} := 0$ for $|l| > q$ when $d > 1$,
for $j = 0, \dots, p-1$,

$$\hat{g}_{j,r} = (2\pi)^{-1/2} \frac{1}{q} \sum_{l=-dq}^{dq-1} g_{j,l} \, e^{-ir\pi l/(dq)}, \qquad r = -dq, \dots, dq-1.$$

Extend $\hat{g}_{j,r}$ to $j = 0, \dots, 2p-1$ since $\hat{g}_{j+p,r} = \hat{g}_{j,-r}$ in polar coordinates.

Step 2: Filter and scale $\hat{g}_{j,r}$,
for $j = 0, \dots, 2p-1$,
for $r = -dq, \dots, dq-1$,

$$\hat{g}_{j,r} \leftarrow \frac{\pi^{3/2}\tilde{\sigma}_r}{4\sqrt{2}pd^2} \, F\big(r/(dq)\big) \, \hat{g}_{j,r}.$$

$\tilde{\sigma}_0 = 0.2, \tilde{\sigma}_{\pm 1} = 0.98$, and $\tilde{\sigma}_r = |r|$ for $r > 1$. $F(\sigma)$ is a smoothing function; i.e.

$$F(\sigma) := \begin{cases} 1, & \text{for } \textit{no} \text{ additional smoothing,} \\ \cos(\sigma\pi/2);\ \operatorname{sinc}(\sigma);\ \operatorname{sinc}^3(\sigma);\ \text{etc.,} & \text{for additional smoothing.} \end{cases}$$

Step 3: Compute

$$f_k = \sum_{j=0}^{2p-1} \sum_{r=-dq}^{dq-1} \hat{g}_{j,r} \, e^{i\pi k \cdot \xi_{j,r}/q}, \qquad k \in \mathbb{R}^2 \text{ s.t. } |k| < q,$$

via a 2-dimensional NED with the non-equispaced nodes, $\xi_{j,r}$,

$$\xi_{j,r} = \frac{r}{d}\theta_j = \frac{r}{d} \left(\begin{smallmatrix} \cos(\pi j/p) \\ \sin(\pi j/p) \end{smallmatrix}\right).$$

Evaluation of this step via the 2D NED Algorithm, Algorithm 3, is only possible after $\hat{g}_{j,r}$, $\xi_{j,r}(1)$, and $\xi_{j,r}(2)$ are turned into vectors (i.e. for $m = 1, \dots, 2p2dq$, in MatLab notation: $\hat{g}_m = \hat{g}_{j,r}(:)$. Similarly procure vectors $\xi(1)$ and $\xi(2)$).

5.2 Results of Gridding Algorithm

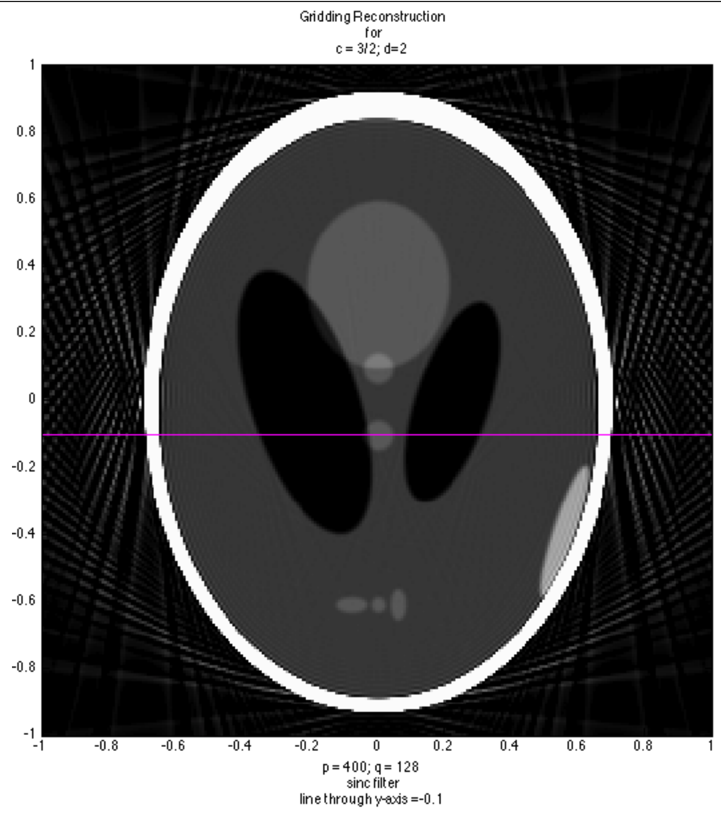

Figure 13: Gridding Method reconstructed image created in MatLab with a sinc filter and oversampling factors $c = 3/2$ and $d = 2$; for code, see Appendix **??**.

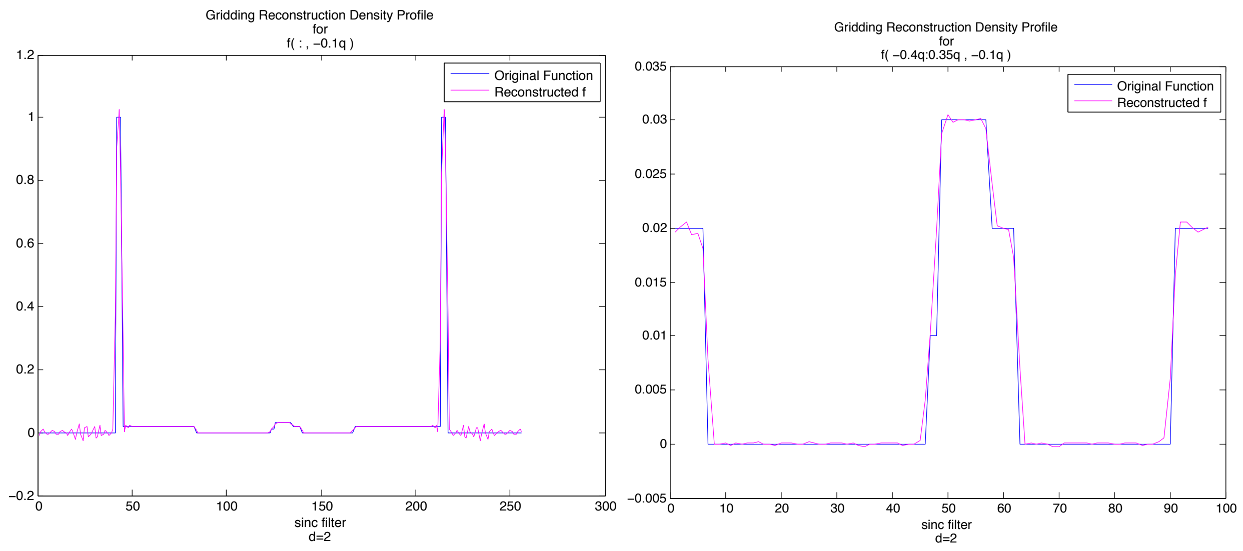

Figure 14: Density Profile corresponding to the line through the image in Figure 13. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

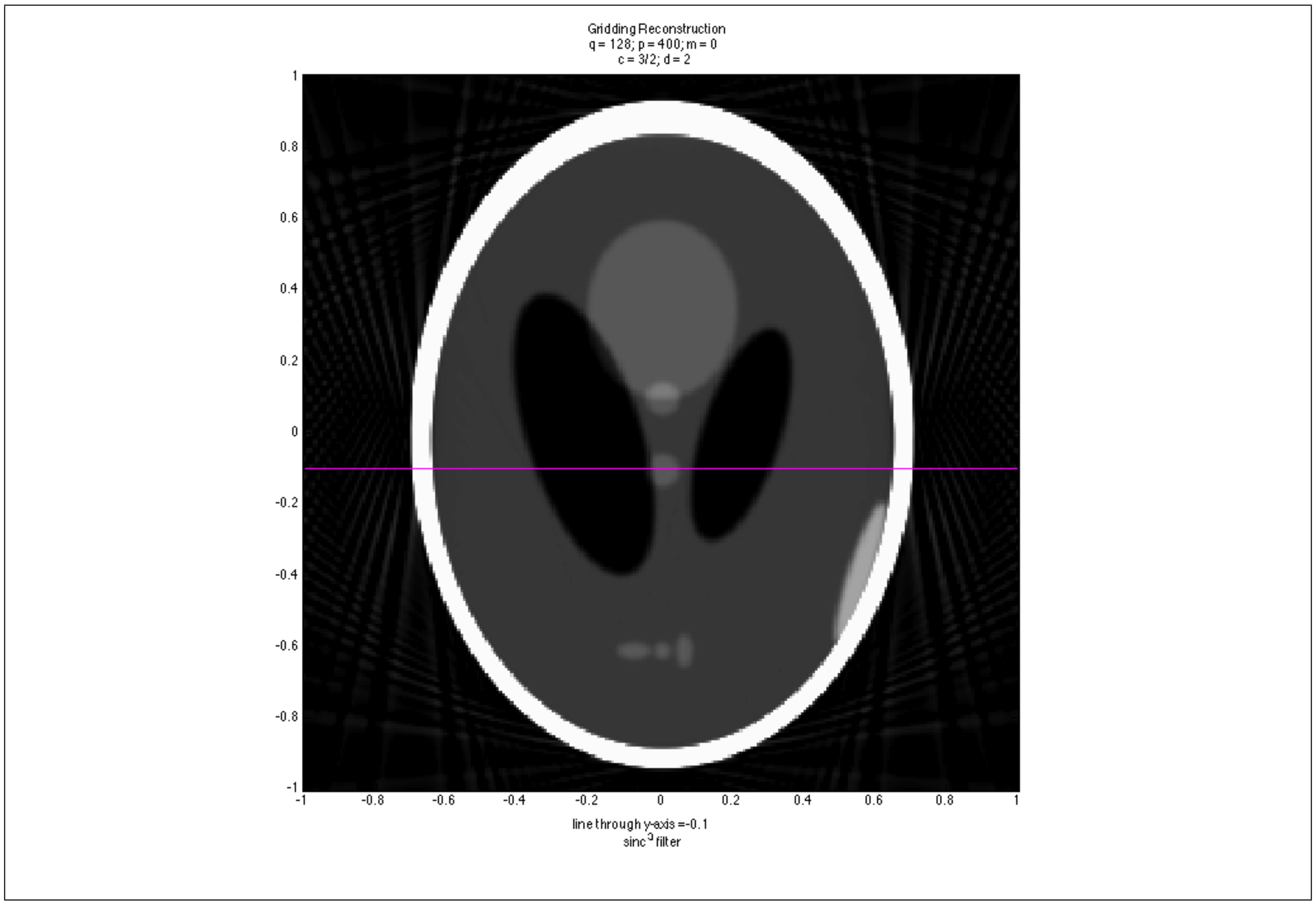

Figure 15: Gridding Method reconstructed image created in MatLab with a sinc^3 filter and oversampling factors $c = 3/2$ and $d = 2$; for code, see Appendix **??**.

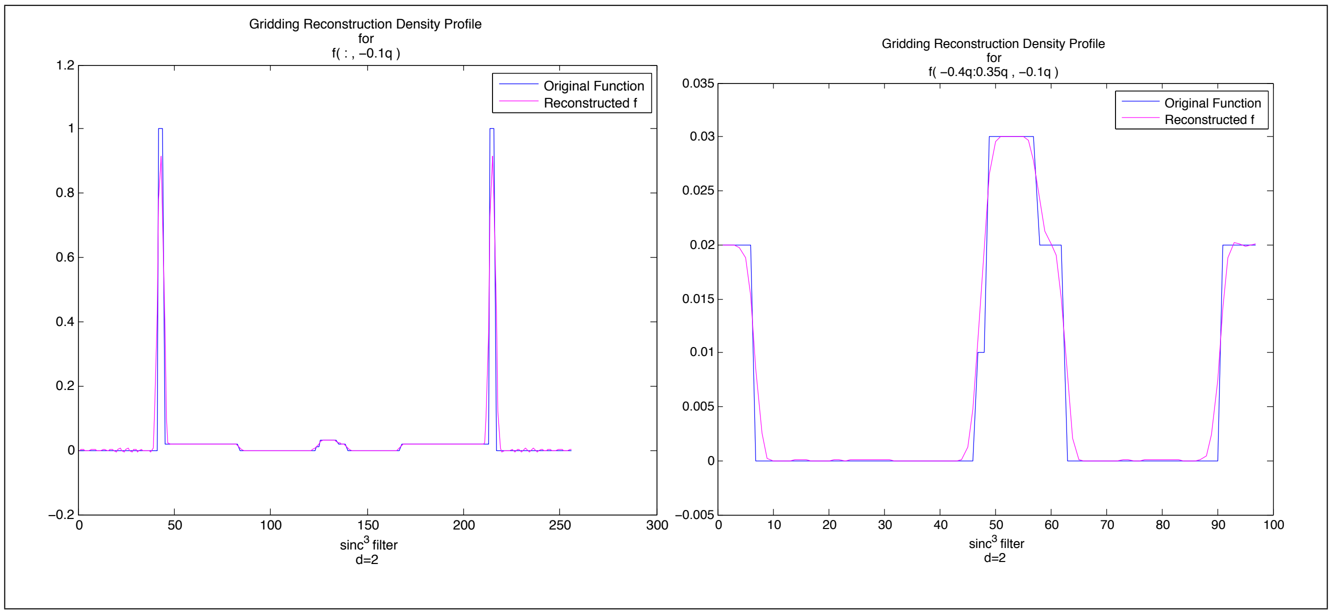

Figure 16: Density Profile corresponding to the line through the image in Figure 15. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

Table 9: Time for $m = 0$: $c = 3/2$, $d = 1$ (time in seconds)

p \ q	64	128	256
200	0.8	1.30	2.43
400	1.33	2.40	4.62
800	2.49	4.55	8.87

Table 10: Rel. Error for $m = 3$: $c = 3/2$, $d = 1$ ($\times 10e-2$)

p \ q	64	128	256
200	4.94	4.73	4.71
400	4.94	4.73	4.71
800	4.94	4.73	4.71

Table 11: Time for $m = 0$: $c = 3/2$, $d = 2$ (time in seconds)

p \ q	64	128	256
200	1.28	2.41	4.67
400	2.40	4.54	8.68
800	4.49	8.63	17.34

Table 12: Rel. Error for $m = 3$: $c = 3/2$, $d = 2$ ($\times 10e-2$)

p \ q	64	128	256
200	1.65	0.63	0.21
400	1.65	0.63	0.21
800	1.65	0.63	0.21

Immediately, this method proves an improvement on the Fourier reconstruction method in Section 3.4. The images produced by the same p and q, for $d = 2$ and $c = 3/2$, have almost the same level of clarity as Filtered Backprojection. We show the images corresponding to when a sinc filter is used, Figure 13, and when a sinc^3 filter is used, Figure 15, along with their density profiles, Figures 14 and 16. The sinc filtered images have more clear differentiation between the differing densities but more outside noise, while the sinc^3 filtered images have less noise but are more blurred.

With no oversampling, i.e. $d = 1$, this algorithm is almost as efficient as Filtered Backprojection time wise, however, it does not produce as useful an image. When $d = 2$, the time is only comparable to Filtered Backprojection for large values of p and q, see Table 11.

Thus, we were not able to see the same time efficiency Fourmont achieved. This could be due to our use of MatLab on a personal MacBook Pro for implementation software and hardware while Fourmont used C and tested on a 300MHz Sun UltraSPRAC-II [6]. But, Fourmont's gridding method achieves equivalent images at no higher cost and with a more optimal asymptotic time increase as $q \simeq p$ increase (except for 1 round of pre-computations for each p and q pair). It is interesting to note the asymptotic behavior of the time increase for both $d = 1$ and $d = 2$ (see Figure 17). In both of these cases, as $q \simeq p$ increases by a factor of 2 along the diagonal entries of Tables 9 and 11 the corresponding time increase is around a factor of 4, half that of Filtered Backprojection. Thus, if larger p and q are required, this gridding method would become a more efficient algorithm than Filtered Backprojection.

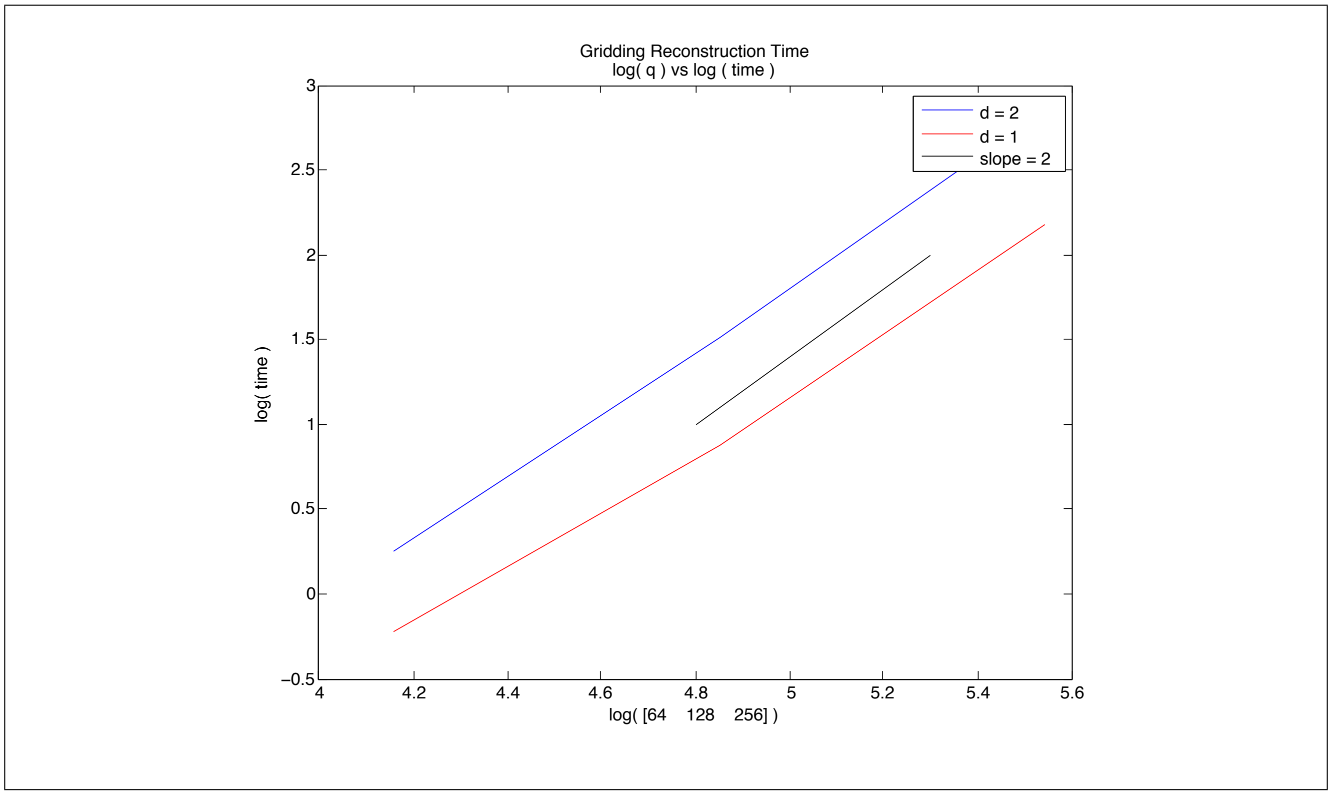

Figure 17: Plots of log (q) versus log of implementation times for the gridding method for $d = 1$ and $d = 2$. The slope plot indicates the number of operations to be around $\mathcal{O}(p^2)$.

5.3 Application of the 1-dimensional NER to Tomography: "Fast Fourier Reconstruction"

The last reconstruction method we examine is Fourmont's "Fast Fourier Reconstruction" and its algorithm (see Section 5.2 [6]). This method utilizes p, 1-dimensional NERs. Applying the Projection Slice Theorem, Theorem 2.2, to the Fourier Transform of a function f produced a succinct double integral (see equation (42)) solely based on the Radon Transform of f. However, the resulting discretized version of equation (42), equation (43), was in polar coordinates. This is not the ideal case. We now consider the specific function defined at the beginning of Section 5 in this case.

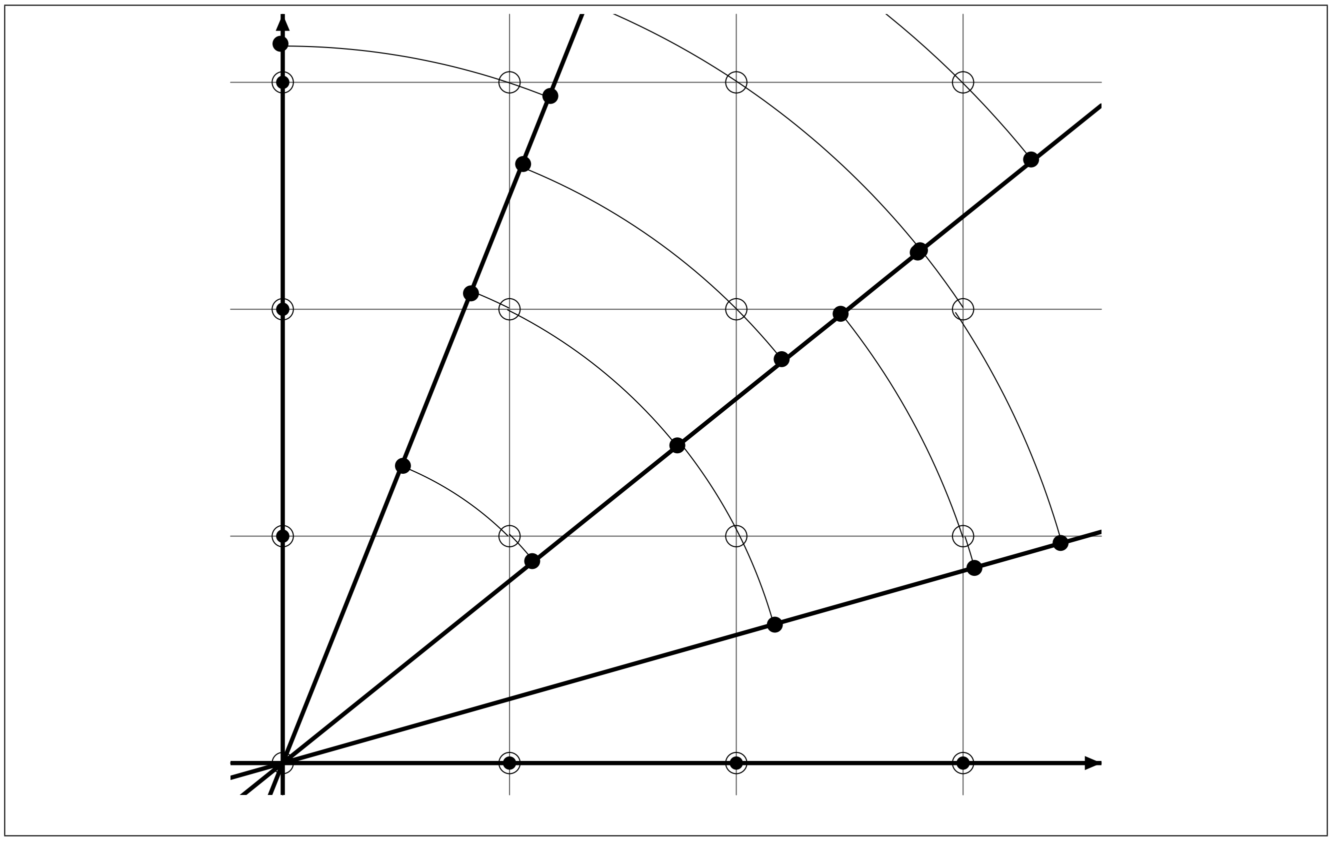

Figure 18: Zoomed in view of a portion of the Interpolation Scheme for "Fast Fourier Reconstruction" in Quadrant I of Fourier space. The circles - ○ - correspond to cartesian grid points where we *want* to know $\hat{f}$. The points - ● - on the rays (φ_j's) from the origin are where we *know* $\hat{f}$. The arcs indicate which of the points we *know* contribute to which cartesian grid points under angular interpolation. If a circle is already on a ray it gets 100% of the contribution of that point.

Natterer discusses why nearest neighbor interpolation is insufficient to produce a usable image [1]. He justifies an inequality to guide choosing reasonable points ξ_k at which to sample $\hat{f}_k$ to approximate $\hat{f}(\pi k)$ (see equation (2.4) page 121 [1]). However, as shown in Section 3.4, nearest neighbor interpolation in the horizontal or vertical direction, which satisfies Natterer's stipulations, is substandard. *Angular* interpolation of the Fourier Transform on a polar grid satisfies the requirements (V.2 [1]), and is shown to improve upon the previous stipulations in [20]. In actuality, linear interpolation (in strictly the angular direction) is adequate. Thus $\hat{f}$ can be approximated at cartesian points via $\hat{f}$ located at polar coordinates using angular interpolation, see Figure 18 for the schematic outline.

Consider the point πk, $k \in \mathbb{Z}^2$. We interpolate $\hat{f}(\pi|k|\theta_j)$, for θ_j chosen correctly, to $\hat{f}(\pi k)$. Therefore, we accurately approximate $\hat{f}(\pi k)$ with $\hat{f}(\pi|k|\theta_j)$. Hence we are left to compute, for $k \in \mathbb{Z}^2$ such that $|k| \leq q$,

$$\begin{aligned}\hat{f}(\pi|k|\theta_j) &= (2\pi)^{-1/2}\hat{g}(\theta_j, \pi|k|),\\ &\simeq (2\pi)^{-1/2}(2\pi)^{-1/2}\frac{1}{q}\sum_{l=-q}^{q-1} g_{j,l}\ e^{-i\pi|k|l/q},\\ &= (2\pi q)^{-1}\sum_{l=-q}^{q-1} g_{j,l}\ e^{-i\pi|k|l/q}.\end{aligned}$$

Thus, for $j = 0, \dots, p-1$,

$$\hat{f}(\pi|k|\theta_j) \simeq (2\pi q)^{-1} \sum_{l=-q}^{q-1} g_{j,l}\; e^{-\pi i |k| l/q}, \qquad k \in \mathbb{Z}^2,\ |k| \leq q, \tag{44}$$

is now in the form of an NER type NUFFT. Following p implementations of the 1-dimensional NER and angular interpolation, a 2-dimensional Inverse Fourier Transform produces f on a cartesian grid.

Suppose, d is an oversampling factor. Then, instead we have $\hat{f}(\pi(|k|/d)\theta_j)$. Hence equation (44) becomes,

$$\hat{f}(\frac{\pi}{d}|k|\theta_j) = (2\pi q)^{-1} \sum_{l=-q}^{q-1} g_{j,l}\; e^{-\pi i (|k|/d) l/q}, \qquad k \in \mathbb{Z}^2,\ |k| \leq dq$$

Completely evaluating this requires the angular interpolation step previously discussed and schematically depicted in Figure 18. To have Fourmont's "Fast Fourier Reconstruction" algorithm be at *all* "fast," the linear interpolation coefficients shall be pre-computed. These pre-computations are described below. The pre-computations need only be done for $k \in \mathbb{Z}^2$ in the upper-half plane, because f is real valued, therefore for k in the lower-half plane $\hat{f}(-k) = \overline{\hat{f}(k)}$. There are 3 pre-computations for a cartesian grid point, $k \in \mathbb{Z}^2$, in the upper-half plane with $|k| \leq dq$.

- First, let $l_k, l_{k+1} \in \{\frac{p}{\pi}\varphi_j\}_{j=0}^{p-1}$ denote the two rays closest k; i.e. l_k is associated with the nearest φ_j traveling clockwise from k around the origin and $l_k + 1$ corresponds to the nearest φ_j traveling *counter*clockwise from k around the origin (if k is *on* φ_j then $l_k = \frac{p}{\pi}\varphi_j$ and $l_k + 1 = \frac{p}{\pi}(\varphi_{j+1})$). The k and $k+1$ indexes are taken p periodic.

- Next, create lists $\{x^{l_k}\}$ and $\{x^{l_k+1}\}$ of evaluation points for the corresponding NER evaluations by adding $|k|$ to each list.

- Last, compute the appropriate linear interpolation coefficients:

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

  Here, $\varphi_k = \operatorname{atan2}(k_2, k_1)$ and $F(\sigma)$ is a reasonable smoothing filter (similar to Algorithm 4, $F(\sigma)$ can be 1, $\cos(\frac{\pi}{2}\sigma)$, $\operatorname{sinc}(\sigma)$, $\operatorname{sinc}^3(\sigma)$, etc).

Note. The a_k and b_k are mislabeled in Fourmont's paper [6]. From Step 2 in the upcoming algorithm, Algorithm 5, it follows that for k on l_k, a_k should equal 1 and b_k should equal 0, while for k on $l_k + 1$, a_k and b_k's values are switched. Consider, if k on l_k, then

$$\begin{aligned} a_k &= 1, \\ &= |\varphi_k - \frac{\pi}{p}l_k - \pi/p|\pi/p, \\ &= |\varphi_k - (l_k+1)\pi/p|\pi/p. \end{aligned}$$

Similarly, if k on $l_k + 1$ then

$$\begin{aligned} b_k &= 1, \\ &= |\varphi_k - \frac{\pi}{p}l_k + \pi/p - \pi/p|\pi/p, \\ &= |\varphi_k - l_k\pi/p|\pi/p. \end{aligned}$$

Before beginning this "Fast Fourier Reconstruction" algorithm, recall that the NER algorithm requires pre-computations as well. Thus, for each θ_j, $j = 0, \ldots, p-1$, there is a set of NER pre-computations that must be made *after* the interpolation pre-computations. The order is important because the "x_l's" for the p instances of the NER algorithm correspond to the p lists, $\{x^{l_k}\}/d$. So, while the upcoming algorithm itself may be short and compute fairly quickly, it is worthwhile to remember all the pre-computations necessary for successful fast algorithmic evaluation. In particular, for larger p and q the NER pre-computations become time consuming. However, in many practical scenarios only one set of pre-computations is needed for a vast number of experiments. Thus proving this method highly efficient (i.e. a CT scan machine in a hospital with a preset number of angles, p, and image pixels, q).

Algorithm 5 Fast Fourier Reconstruction via 1D NER (Section 5.2 [6]):

Parameters: Parallel Beam scanning geometry p, q, oversampling factor d, pre-computations: l_k, a_k, b_k, and parameters/pre-computations for p versions of the 1D NER determined by: $\{x^{l_k}\}/d$

Input: $g_{j,l} = g(\theta_j, s_l)$, $j = 0, \ldots, p-1$ and $l = -q, \ldots q-1$ for $g := \mathbf{R}f$

.

Step 1: Compute an NER evaluation of $g_{j,l}$ for each θ_j,
for $j = 0, \ldots, p-1$,

$$\hat{g}_{j,m} = \sum_{l=-q}^{q-1} g_{j,l}\ e^{-\pi i l (x_m^j/d)/q} \qquad \forall\ m \text{ s.t. } x_m^j \in \{x^{l_k}\} \text{ for } l_k = \frac{p}{\pi}\varphi_j.$$

Step 2: Linearly interpolate $\hat{g}_{j,m}$ in the angular direction to obtain $\hat{f}_k$,
for $k \in \mathbb{Z}^2$, $|k| \leq dq$,

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

Recall, l_k and $l_k + 1$ denote the rays (φ_j's) nearest k clockwise and counterclockwise. j_k^1 is the index of $|k|$ in list $\{x^{l_k}\}$ and j_k^2 is the index of $|k|$ in list $\{x^{l_k+1}\}$.

Step 3: Finally, compute a standard 2-dimensional Inverse FFT,

$$f_l = \frac{q}{(2dq)^2} \sum_{|k|<dq} \hat{f}_k\ e^{2\pi i k \cdot l/(2dq)}, \qquad l \in \mathbb{Z}^2 \text{ s.t. } |l| < q.$$

where the front constant term on the right hand side is the accumulation of all of the discretization constants throughout Steps 1-3.

5.4 Results of Fast Fourier Reconstruction Algorithm

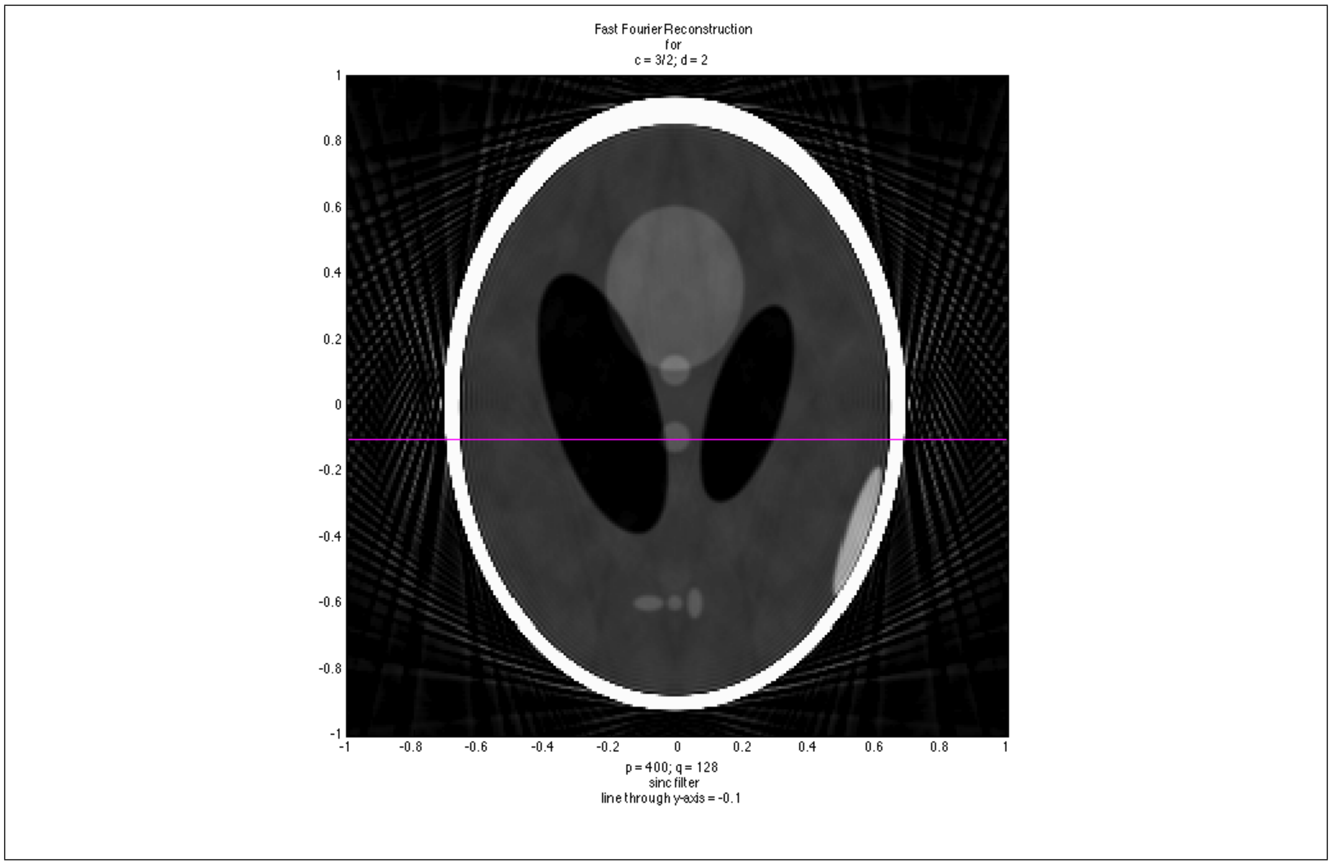

Figure 19: Fast Fourier Reconstruction image reconstructed in MatLab with a sinc filter and over-sampling factors $c = 3/2$ and $d = 2$; for code, see Appendix **??**.

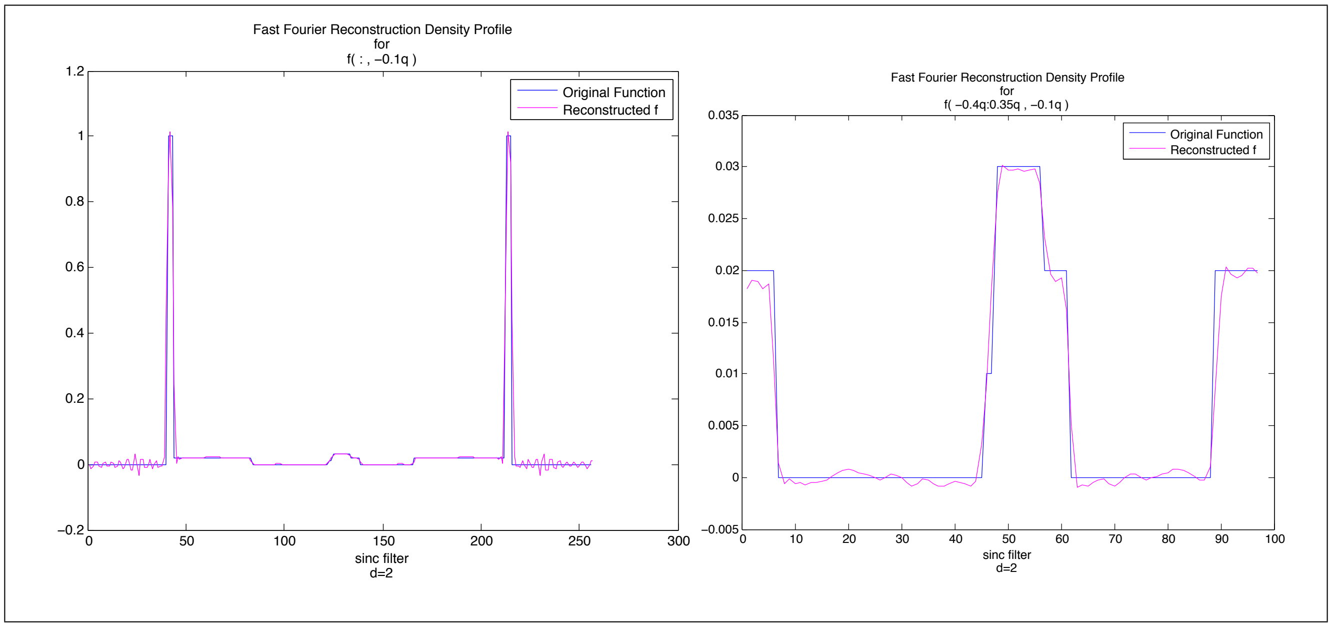

Figure 20: Density Profile corresponding to the line through the image in Figure 19. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

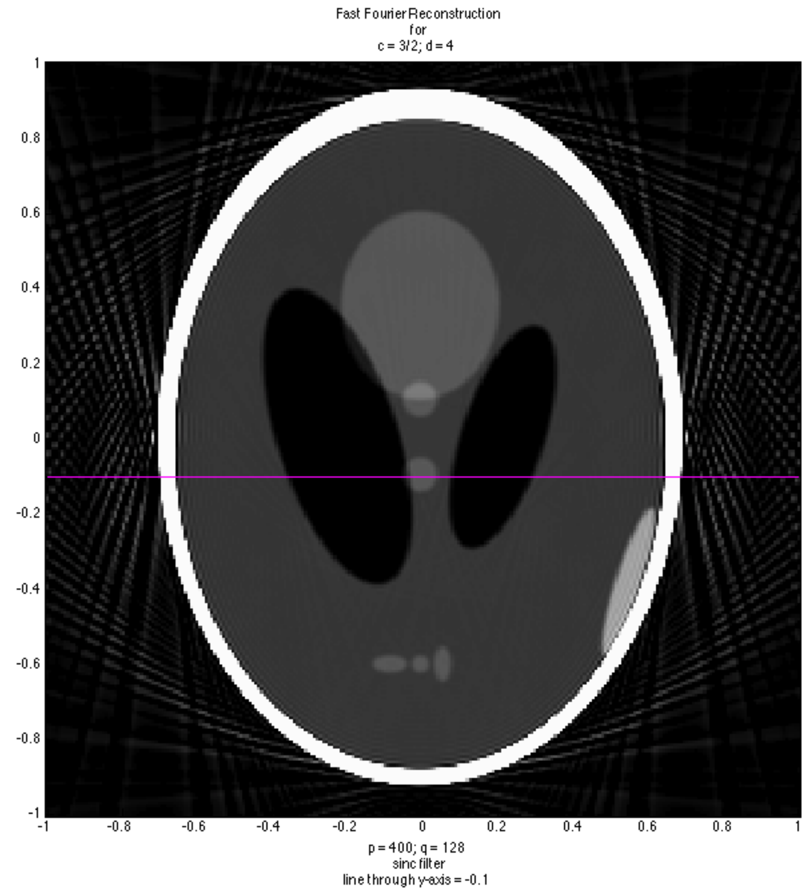

Figure 21: Fast Fourier Reconstruction image reconstructed in MatLab with a sinc filter and oversampling factors $c = 3/2$ and $d = 4$; for code, see Appendix **??**.

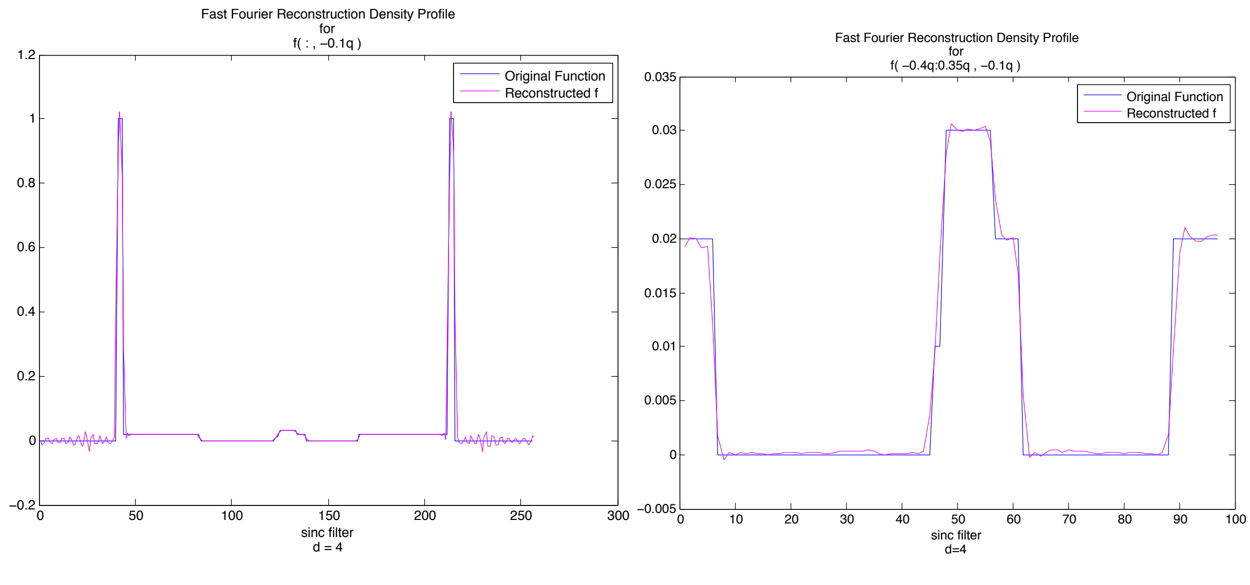

Figure 22: Density Profile corresponding to the line through the image in Figure 21. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

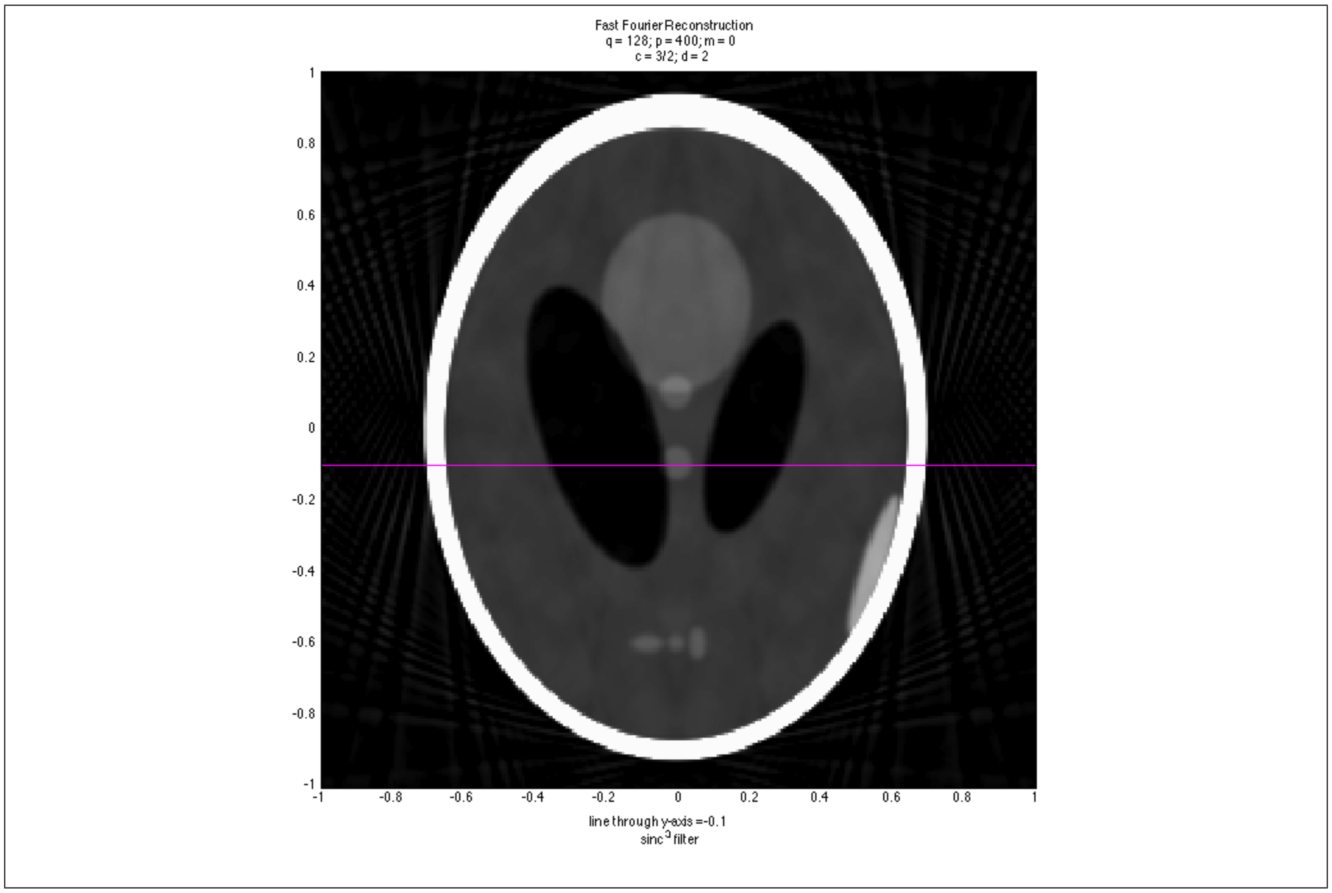

Figure 23: Fast Fourier Reconstruction image reconstructed in MatLab with a sinc^3 filter and oversampling factors $c = 3/2$ and $d = 2$; for code, see Appendix **??**.

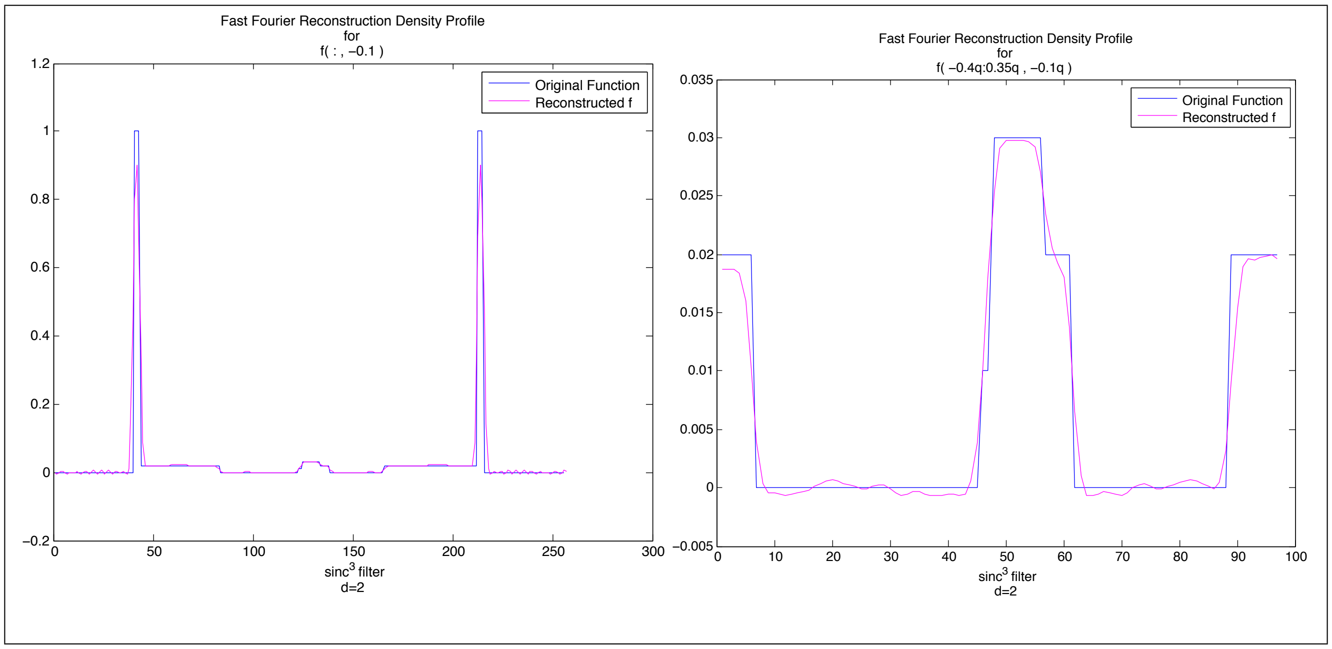

Figure 24: Density Profile corresponding to the line through the image in Figure 23. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

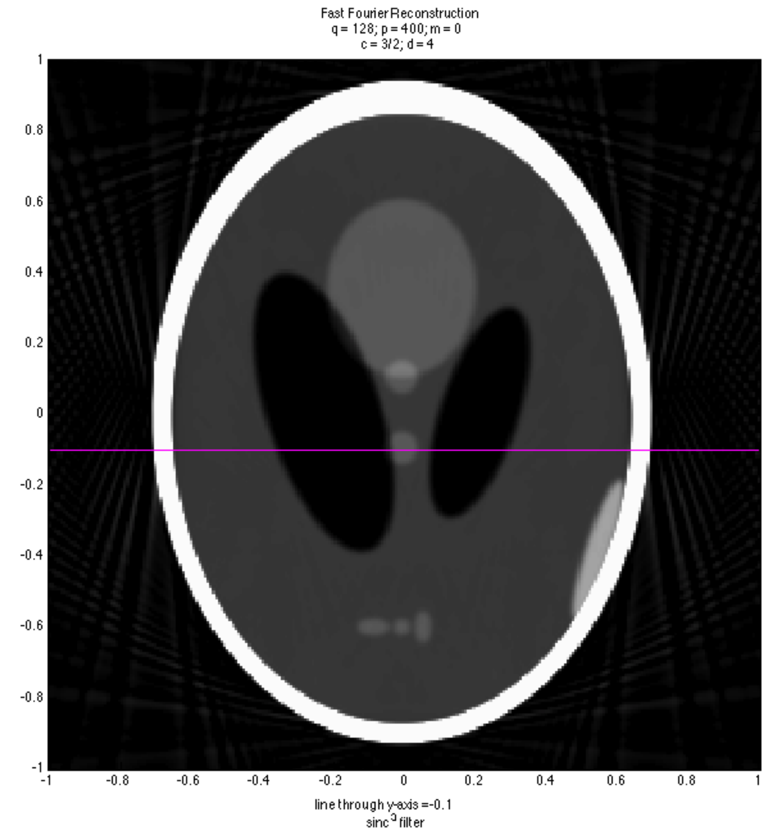

Figure 25: Fast Fourier Reconstruction image reconstructed in MatLab with a sinc^3 filter and over-sampling factors $c = 3/2$ and $d = 4$; for code, see Appendix **??**.

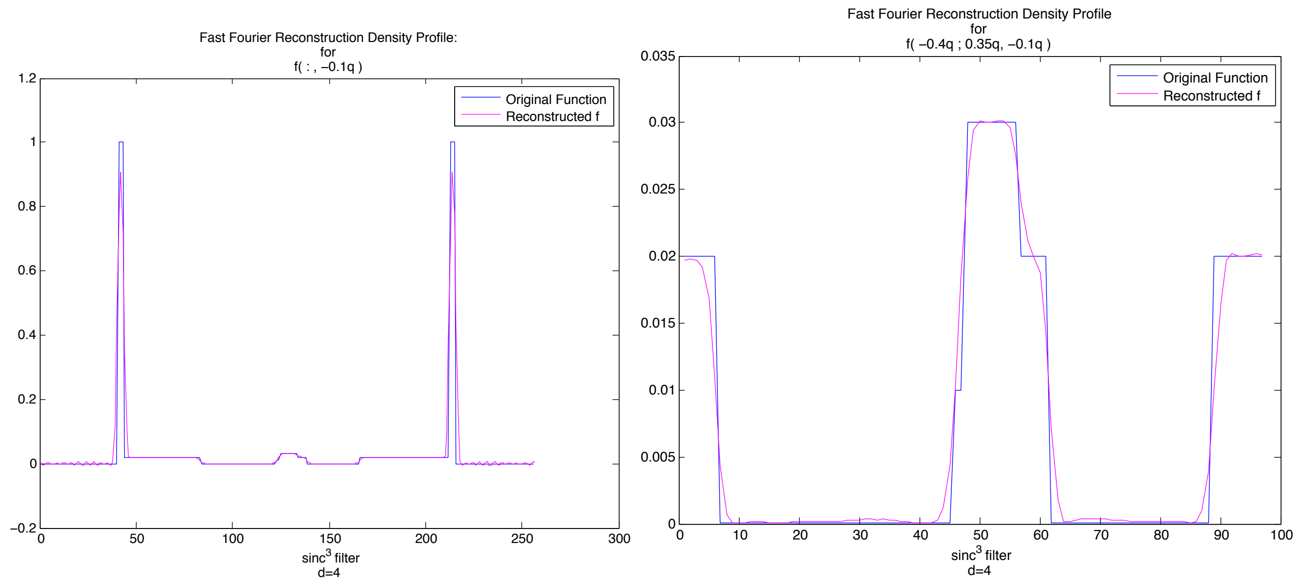

Figure 26: Density Profile corresponding to the line through the image in Figure 25. Left graph is the density profile from $x = -1$ to $x = 1$ at $y = -0.1$; the right graph is a zoomed-in portion, $x = -0.4$ to $x = 0.35$, of the left graph to show detail. Plots created in MatLab; for code, see Appendix **??**.

Table 13: Time for $m = 0$:
$c = 3/2$, $d = 2$ (time in seconds)

p \ q	64	128	256
200	0.14	0.26	0.69
400	0.25	0.54	0.89
800	0.49	0.93	1.83

Table 14: Rel. Error for $m = 3$:
$c = 3/2$, $d = 2$ ($\times 10e - 2$)

p \ q	64	128	256
200	2.03	0.87	0.36
400	2.03	0.86	0.34
800	2.03	0.85	0.34

Table 15: Time for $m = 0$:
$c = 3/2$, $d = 4$ (time in seconds)

p \ q	64	128	256
200	0.29	0.72	2.77
400	0.61	1.20	3.52
800	1.03	2.13	6.57

Table 16: Rel. Error for $m = 3$:
$c = 3/2$, $d = 4$ (sinc filter;$\times 10e - 2$)

p \ q	64	128	256
200	1.04	0.40	0.18
400	1.03	0.38	0.16
800	1.03	0.38	0.16

This final reconstruction method, immediately yields clearer, more useful images than the Fourier reconstruction method in Section 3.4. Akin to the gridding method, Figures 19 and 23 and their accompanying density profiles, Figures 20 and 24, show that a sinc filter results in a sharper image but with more exterior noise, while a sinc^3 filter has less overall error, but is less focused. Additionally, we only consider this method for $d \geq 2$ because otherwise the interpolation step introduces too much error.

However, even with d chosen as large as 4, the execution time for this method is somewhere between Filtered Backprojection and the first Fourier reconstruction method we examined. When $d = 2$ it is almost as efficient the first Fourier reconstruction method and it produces a much better quality image. The speed up factor is about 3 for $p = 400$, $q = 128$ and about 8 for $p = 800$, $q = 256$ from Filtered Backprojection! This still is not as good as the speedup Fourmont cites, but it is still notable [6].

Another important thing to note, is the asymptotic time increase as $p \simeq q$ increase by a factor of 2. Figure 27 illustrates the relationship between the size of $p(\simeq q)$ and the implementation time. For both $d = 2$ and $d = 4$; the number of operations seems to be right at about $\mathcal{O}(p^2)$. Thus, indicating for larger p and q this is an efficient method, provided one set of pre-computations suffices for many subsequent image reproductions.

Something novel worth mentioning for this method is an interesting shadowing effect that occurs for small d. It is particularly visible in the $d = 2$ case and still can be seen when $d = 3$. However, it is no longer evident when $d = 4$. This artifact is unlike any other I have come across in image reconstruction thus far. It could be something introduced from the weighting of the interpolation coefficients in the interpolation step; but this is just speculation.

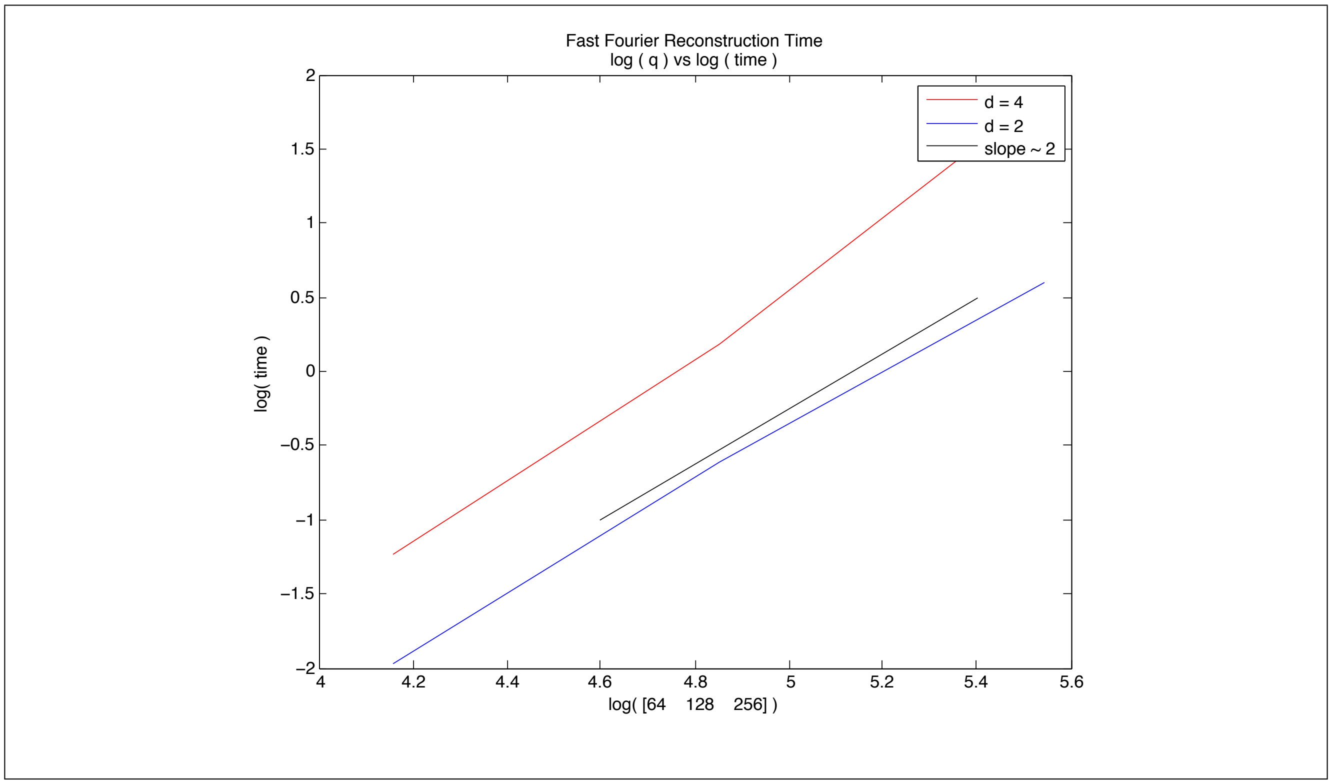

Figure 27: Plots of log (q) versus log of implementation times for the Fast Fourier Reconstruction method for $d = 2$ and $d = 4$. The slope plot indicates the number of operations to be about $\mathcal{O}(p^2)$.

6 Conclusion

The applications of these NUFFTs to Fourier reconstruction do greatly improve upon the image quality of the first Fourier reconstruction method we examined. They produce images on par with Filtered Backprojection from data FBP could evaluate no faster than on the order of $\mathcal{O}(p^3)$ operations, and they do so in the same amount of time, if not faster. In fact, as Figure 28 indicates, the asymptotic time growth for gridding and, particularly, it confirms that Fast Fourier Reconstruction computes in the expected $\mathcal{O}(p^2 \log p)$ operations (see the slope of approximately 2 in the plot for $d = 2$ and 4 in the FFR).

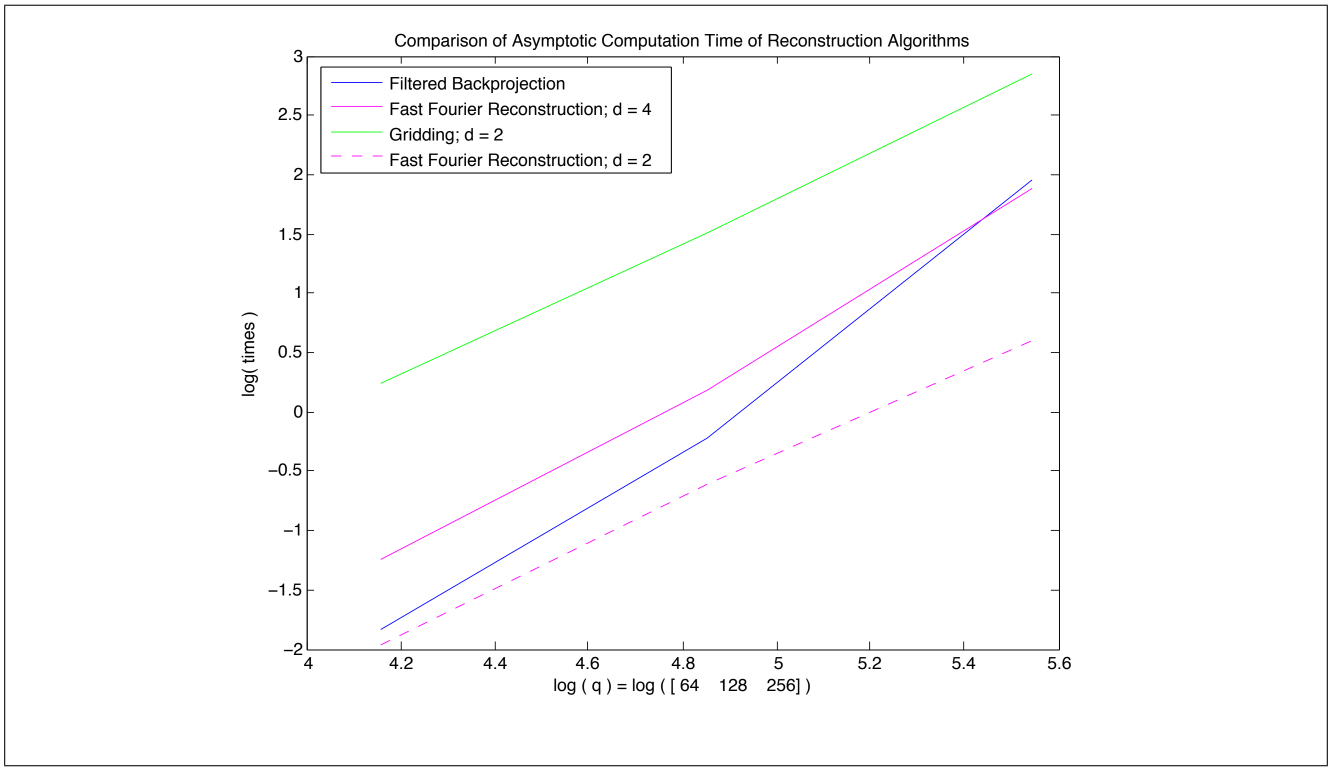

Figure 28: Plots of log (q) versus log of implementation times for Filtered Backprojection, Fourmont's gridding method, and Fourmont's Fast Fourier Reconstruction method.

It is clear that Fourier reconstruction methods are faster for sufficiently large p and q. Additionally, in practical applications one set of pre-computations would suffice for a large number of image reconstructions eliminating concerns over the implementation time for pre-computations. Again, the difference between our time results and Fourmont's could be due to the different software and hardware used for experimentation and, also, there may be slight modifications possible within the code that would enable even more efficient evaluation times.

Thus, in conclusion, these applications of several Non-Uniform Fast Fourier Transforms to Fourier reconstruction are comparable with Filtered Backprojection. Additional work with NUFFTs in general as well as cross-referencing and comparing these specific Fourier reconstruction methods with others would be an interesting and quite possibly rewarding avenue of further investigation.

References

[1] Frank Natterer. *The Mathematics of Computerized Tomography.* John Wiley and Sons Ltd and B G Teubner, Stuttgart, Chichester; New York; Brisban; Toronto; Singapore, 1986.

[2] Adel Faridani. Introduction to the mathematics of computed tomography. *Inside Out: Inverse Problems and Applications*, 47:1–46, 2003.

[3] Amy Berrington de Gonzalez, Mahadevappa Mahesh, Kwang-Pyo Kim, Mythreyi Bhargavan, Rebecca Lewis, Fred Mettler, and Charles Land. Projected cancer risks from computed tomographic scans performed in the united states in 2007. *Archives of internal medicine*, 169(22):2071, 2009.

[4] David J Brenner and Eric J Hall. Computed tomography - an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, 2007.

[5] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[6] Karsten Fourmont. Non-equispaced fast fourier transforms with applications to tomography. *Journal of Fourier Analysis and Applications*, 9(5):431–450, 2003.

[7] Andreas Rieder and Adel Faridani. The semidiscrete filtered backprojection algorithm is optimal for tomographic inversion. *SIAM journal on numerical analysis*, 41(3):869–892, 2003.

[8] Alok Dutt and Vladimir Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific computing*, 14(6):1368–1393, 1993.

[9] Leslie Greengard and June-Yub Lee. Accelerating the nonuniform fast fourier transform. *SIAM review*, 46(3):443–454, 2004.

[10] Gabriele Steidl. A note on fast fourier transforms for nonequispaced grids. *Advances in computational mathematics*, 9(3-4):337–352, 1998.

[11] Antony F Ware. Fast approximate fourier transforms for irregularly spaced data. *SIAM review*, 40(4):838–856, 1998.

[12] Daniel Potts and Gabriele Steidl. Fourier reconstruction of functions from their nonstandard sampled radon transform. *Journal of Fourier Analysis and Applications*, 8(6):513–534, 2002.

[13] Karsten Fourmont. *Schnelle FourierTransformation bei nichtquidistanten Gittern und tomographische Anwendungen.* PhD thesis, University of Mnster, 1999.

[14] Franklin F. Kuo and James F. Kaiser. Digital filters. In *System Analysis by Digital Computer*, pages 218–285. John Wiley, 1966.

[15] W.N. Brouw. Aperture synthesis. In *Image Processing Techniques in Astronomy*, pages 301–307. Springer, 1975.

[16] J.D. O'Sullivan. A fast sinc function gridding algorithm for fourier inversion in computer tomography. *Medical Imaging, IEEE Transactions on*, 4(4):200–207, 1985.

[17] M. Kaveh and M. Soumekh. Computer-assisted diffraction tomography. *Image recovery: theory and application*, pages 369–413, 1987.

[18] Hermann Schomberg and Jan Timmer. The gridding method for image reconstruction by fourier transformation. *Medical Imaging, IEEE Transactions on*, 14(3):596–607, 1995.

[19] Hannah Stanton. Mth 656: Final project. Class Term Paper, 2013.

[20] Frank Natterer. Fourier reconstruction in tomography. *Numerische Mathematik*, 47(3):343–353, 1985.