SPECT Reconstruction Using the Expectation Maximization Algorithm and an Exact Inversion Formula

Kyle Champley
Oregon State University

December 2004

Abstract

We develop the theory behind the Expectation Maximization algorithm and an exact inversion formula for the attenuated Radon transform, two reconstruction methods used in SPECT. We also implement both methods and present a few numerical experiments.

Contents

1 Introduction **3**

2 Expectation Maximization **4**
2.1 Model 4
2.2 Theory 6

3 Inversion Formula **13**
3.1 Model 13
3.2 Theory 14

4 Implementation **25**

5 Numerical Results **27**
5.1 Inversion Formula 27
5.2 EM Algorithm 28

6 Conclusions **28**

7 MATLAB Code **29**

List of Figures

1 Line parameterized by $x \cdot \theta = s$. 14
2 Geometry of the convergence of z to t_0. 16
3 Inversion formula reconstructions for experiment 1. 43
4 Cross section plots for inversion formula, experiment 1. 44
5 Inversion formula reconstructions for experiment 2. 45
6 Cross section plots for inversion formula, experiment 2. 46
7 Inversion formula reconstructions with added noise for experiment 2. 47
8 EM algorithm reconstructions for experiment 2. 48
9 Cross section plots for the EM algorithm, experiment 2. 49
10 EM algorithm reconstructions with added noise for experiment 2. . . 50
11 Cross section plots for the EM algorithm with added noise, experiment 2. 51

1 Introduction

Single Photon Emission Computed Tomography (SPECT) is a medical imaging modality used to measure activity levels of certain regions of the body. After the activity distribution is recovered, a colored image is created from its level sets. The brain, throat, heart, and lungs are the most common regions of the body that SPECT scans are used on. For instance, one might use a SPECT scan to find clogged arteries on a patient with heart disease. Clogged arteries would show up as dark spots in the image, giving the physician information on the source of the problem. Often times SPECT scans are compared to the more well-known CAT scan. Mathematically speaking, the reconstruction methods used in creating CAT scan images are simplified versions of those that create SPECT scan images. Although there are some similarities between SPECT and CAT scans, they are quite different in their use in the field of medical diagnosis. While SPECT scans try to retrieve information relating to the functionality of the body, CAT scans try to retrieve the physical makeup of the body.

These images are produced by first introducing a radiopharmaceutical into the area of interest in the body. This radiopharmaceutical gives off radiation in the form of photons emitted from the source of radioactive decay. These photons are measured by detectors outside the body over a certain period of time. Since the amount of radiation activity is proportional to the concentration of the radiopharmaceutical, one can determine the concentration from measuring the number of photons detected outside the body. This radiation is measured by a gamma camera that encircles the patient in a ring. The face of the gamma camera is covered with collimators, so that it only measures photons traveling essentially parallel to the collimators. Thus when a photon is measured we know that it was emitted from a source somewhere along a certain line in space. An image of this activity distribution can then be reconstructed in a tomographic fashion with various algorithms, two of which we will discuss in this paper. Since we measure the radiation over a period of time, we may even create images of the flow of the radioactive isotope over time, giving a rare glimpse into how the body works in a way that cannot even be seen through surgery.

Most of the photons never make it through the collimators to the gamma camera. This makes an accurate recovery of the activity distribution much more difficult. In other words, much of the data is lost. Since the radiation detectors only lie in a ring around the patient, only photons traveling in the plane of the camera and essentially parallel to a collimator can be measured. Another problem in SPECT imaging lies with the interaction between photons and the body tissue in which they pass through. Photons may be absorbed or their trajectory may get deflected (e.g. Compton Scattering) by the tissue they travel through. For example, when using Tc-99m for the radiotracer, only about 20% of photons will reach the detector after passing through 10 cm of soft tissue and about 20% of those photons are detected in the wrong location due to scatter. When a photon is scattered the energy level of that photon is reduced. Since all the photons of a certain radioactive isotope have the same energy levels, the gamma camera can (with varying accuracy) discount photons whose measurements do not have this expected energy level. We discount these measurements so that we can assume that all measured photons originated from some point along a certain line. This modeling assumption is essential in tomography. Fortunately, the probabilities in which these events occur (called

attenuation) are well known and can be determined by a CAT scan. Thus a transmission scan (e.g. a CAT scan) is used in many SPECT activity reconstructions to recover the attenuation map of the tissue. With this, one can try to compensate for the lost or attenuated data.

Another obstacle in accurate activity reconstruction is the statistical nature of Emission Tomography. Although these radioactive isotopes have a constant half-life, radioactive decay is naturally statistical because measurements of the radioactive decay of the radioisotopes over identical periods of time vary. This variation in the sampling of the radioactive decay comes from the intrinsic probabilistic nature of radioactive decay. Although decay events can fluctuate randomly, from the averaging of many values, the true radiation activity distribution can been deduced. Thus the accuracy increases with larger numbers of events. Unfortunately, the number of measured events in SPECT is relatively small, so accurate reconstructions must take into account these statistical characteristics. Therefore stochastic methods have so far been shown to be more useful in practice. These stochastic models are completely discrete and are solved with iterative techniques. The most popular (and the one that has been used for the past twenty years in most medical scanners) is the Expectation Maximization (EM) algorithm developed by Shepp and Vardi in 1982.

SPECT can also be modeled explicitly with the attenuated Radon transform (see section 3). In the past three years there have been many developments in this area. Including three different inversion formulae. With these new formulae, we may implement a new reconstruction algorithm that could possibly create images that are more accurate than those created by the EM algorithm. Currently the EM algorithm is still used in practice, but it is certainly a much slower algorithm than the inversion formula and produces images that lack in smoothness.

In this paper we will develop the EM algorithm (section 2) and an exact inversion formula for the attenuated Radon Transform (in 2-D) (section 3) that was first found by Novikov, but we will develop the inversion formula found by Natterer[7]. Then in section 4 we discuss the implementation of both of these reconstruction methods. Finally in section 5 we discuss some numerical results of MATLAB experiments with elliptical phantoms. The last section includes all the MATLAB code used to perform the numerical experiments.

2 Expectation Maximization

2.1 Model

For the Expectation Maximization algorithm we first subdivide the reconstruction region into pixels and enumerate these pixels $0 \leq j \leq m$. We also enumerate the detectors $0 \leq i \leq n$. Radioactive decay events over a fixed time interval $[0, \tau]$ are most accurately modeled by Poisson random variables. We will often exploit the fact that the sum of Poisson random variables is also a Poisson random variable and the expectation of the Poisson random variable is equal to its parameter.

Let $\mathbf{P}(\cdot)$ and $\mathbf{E}(\cdot)$ be the probability and expectation functions, respectively. The following random variables are measurements over a fixed time interval and all random variables are assumed to be independent identically distributed (iid). Let N_j be Poisson random variables with expectation v_j that model the measurement of the

number of photons emitted from pixel j per unit area. Let $p_{ij} = \mathbf{P}$(photon emitted from pixel j is detected in detector i). The p_{ij} are derived from the geometry of the scanner and the attenuation map of the tissue. We will explain the determination of these later. Now let N_{ij} be Poisson random variables with expectation $v_j p_{ij}$ that model the number of detected events in detector i from activity in pixel j per unit area. Let $\overline{N}_j$ be Poisson random variables with expectation $f_j = v_j \sum_{i=1}^{n} p_{ij}$ that model the number of photons emitted from pixel j per unit area that are detected by some detector. Finally let g_i be the realization of the Poisson random variables $\gamma_i = \sum_{j=1}^{m} N_{ij}$ with expectation $\sum_{j=1}^{m} v_j p_{ij}$ that denote the number of photons detected in detector i. It is reasonable to assume that $\sum_{j=1}^{m} v_j p_{ij} > 0$ for all i. This is equivalent to assuming that the ith detector has measured some radiation from some region of the body. We may also just remove the ith detector from the model if it so happens that $\sum_{j=1}^{m} v_j p_{ij} = 0$.

Then we see that g_i and p_{ij} are known measurements. We seek to find $v_j = (v_1, v_2, \ldots, v_m)^T$. Clearly, we can get v_j from f_j. Thus we will develop a scheme to find the f_j from which we can find the v_j.

Let $A = (a_{ij})$, where

$$a_{ij} = \frac{p_{ij}}{\sum_{l=1}^{n} p_{lj}}.$$

Then

$$\mathbf{E}[\gamma_i] = \sum_{j=1}^{m} v_j p_{ij} = \sum_{j=1}^{m} \frac{f_j}{\sum_{l=1}^{n} p_{lj}} p_{ij} = \sum_{j=1}^{m} f_j a_{ij},$$

so

$$\mathbf{E}[\gamma] = Af.$$

The values of the p_{ij} can be modeled as follows. Let S be the area of a pixel, r_{ij} be the ratio of the maximum range of angles of trajectories for an event in pixel j to be measured at detector i with 2π, x_j be the position of the jth pixel, θ_i be the angle of the collimator of the ith detector, μ be the attenuation function of the tissue, and $D\mu(\cdot,\cdot)$ be the divergent beam transform (see section 3). Then

$$p_{ij} = r_{ij} S e^{-D\mu(x_j, \theta_i)}.$$

The exponential term may seem curious at this time. Its purpose in the model is described at the beginning of section 3.

Since the elements of A are probabilities, g are measurements of photon activity, and f are expectations of the photon activity measurements, A, g, and f are element-wise nonnegative. Moreover, from above we see that $(Af)_i > 0$ for all i and $\sum_{i=1}^{n} a_{ij} = 1$ for all j. Then we determine f by maximizing the likelihood function which is described as follows.

For iid r.v.'s Λ_1, Λ_2,...Λ_n with unknown parameter η and probability mass function $p(\cdot;\eta)$, the likelihood function is given by

$$L(\eta;\lambda) = p(\lambda_1;\eta)p(\lambda_2;\eta)\cdots p(\lambda_n;\eta).$$

In other words the likelihood function is a function of the unknown parameter, η, of the iid r.v.'s Λ_1, Λ_2,...Λ_n which gives the probability (or likelihood) that a given event will happen for that parameter. Thus if a maximum of the likelihood does exist in a certain instance we can find the most likely parameter for the r.v.'s.

Since we use a Poisson r.v. in our likelihood function, the parameter of the r.v. is equal to its expectation. Thus if we maximize the likelihood function, we can find the f_j that has the greatest probability to have the measured events g_i occur. Let p be the probability mass function of the poisson r.v. γ. Thus $p(k) = e^{-(Af)}\frac{(Af)^k}{k!}$. Then

$$L(f) \equiv L((Af); g) = \prod_{i=1}^{n} p(g_i; (Af)_i) = \prod_{i=1}^{n} \frac{(Af)_i^{g_i}}{(g_i)!} e^{-(Af)_i}.$$

Since maximizing $L(f)$ is the same as maximizing $\log(L(f))$, we maximize the function

$$\log(L(f)) = \log\left(\prod_{i=1}^{n} \frac{(Af)_i^{g_i}}{(g_i)!} e^{-(Af)_i})\right) = \sum_{i=1}^{n} \left[g_i \log(Af)_i - (Af)_i - \log(g_i)!\right].$$

We may also omit the $\log(g_i)!$ term in the sum. Then we arrive at the log likelihood function

$$l(f) \equiv \sum_{i=1}^{n} \left[g_i \log(Af)_i - (Af)_i\right]. \tag{1}$$

The EM algorithm seeks to find an f that maximizes this function. Now we will develop an iterative formula and prove that it converges to a unique maximum point of $l(f)$.

2.2 Theory

Lemma 2.1. *Let H be the hessian matrix of $l(f)$. Then $(x, Hx) \leq 0$ for $x_i \geq 0$ i.e. H is negative semidefinite for $x_i \geq 0$.*

Proof. Now $H = \left(\frac{\partial^2 l}{\partial f_i \partial f_j}\right)$. Then

$$\begin{aligned} \frac{\partial l}{\partial f_i} &= \frac{\partial}{\partial f_i} \sum_{k=1}^{n} g_k \log(Af)_k - (Af)_k \\ &= \sum_{k=1}^{n} \frac{g_k a_{ki}}{\sum_{q=1}^{m} a_{kq} f_q} - a_{ki} \end{aligned} \tag{2}$$

and thus

$$\begin{aligned} \frac{\partial^2 l}{\partial f_i \partial f_j} &= \frac{\partial}{\partial f_j} \sum_{k=1}^{n} \frac{g_k a_{ki}}{\sum_{q=1}^{m} a_{kq} f_q} - a_{ki} \\ &= \sum_{k=1}^{n} g_k a_{ki} \left(\frac{-a_{kj}}{\left(\sum_{q=1}^{m} a_{kq} f_q\right)^2}\right) \\ &= -\sum_{k=1}^{n} \frac{g_k a_{ki} a_{k_j}}{\left(\sum_{q=1}^{m} a_{kq} f_q\right)^2}. \end{aligned}$$

Then since $(H)_{ij} \leq 0$, $(x, Hx) \leq 0$ for $x_i \geq 0$. ∎

Since H is negative semidefinite, l is concave. Thus the local maxima of $l(f)$ (for $f_i \geq 0$) are also global ones and f is a global maximum if and only if the Kuhn-Tucker conditions

$$\begin{aligned} \frac{\partial l}{\partial f_j}(f) &= 0 \quad \text{for } f_j > 0, \\ \frac{\partial l}{\partial f_j}(f) &\leq 0 \quad \text{for } f_j = 0 \end{aligned}$$

are satisfied. From (2) we see that

$$\nabla l(f) = A^T \left(\frac{g}{Af} - 1 \right),$$

where 1 is a vector of ones and all arithmetic operations between vectors are componentwise. Then each global maximum $f \geq 0$ of l satisfies

$$fA^T \left(\frac{g}{Af} - 1 \right) = 0.$$

Since $A^T 1 = 1$,

$$f = fA^T \frac{g}{Af}. \tag{3}$$

Now we define the EM algorithm as the iterative method that solves (3):

$$f^{k+1} = f^k A^T \frac{g}{Af^k}, \qquad \text{k} = 0,\ 1,\ 2,\ ... \tag{4}$$

Theorem 2.2. *Let $f_j^0 > 0$ for $0 \leq j \leq m$. Then (4) converges to a maximizer of $l(f)$.*

Proof. For the proof we follow an outline by [8]. We also follow [11] in step 2 of our proof. We use the fact that $f_j^0 > 0$ for all j, so that we don't have any problems with dividing by zero and to show that (4) converges to a maximizer of $l(f)$ in the last step of our proof. We make use of the Kullback-Leiber distance

$$KL(x, y) = \sum_{i=1}^{n} \left(x_i \log \left(\frac{x_i}{y_i} \right) + y_i - x_i \right),$$

where $x, y \in \mathbb{R}^n$, $x \geq 0$, and $y > 0$ component-wise. We will use the convention that $0 \log 0 = 0$. Note that since $KL(x, y)$ is not symmetric and the triangle inequality does not apply, it is not a metric. Then since

$$\frac{\partial KL}{\partial x_i}(x, y) = \log \left(\frac{x_i}{y_i} \right)$$

and

$$\frac{\partial KL}{\partial y_i}(x, y) = \frac{-x_i}{y_i} + 1 = \frac{y_i - x_i}{y_i},$$

$KL(x, y)$ assumes its minimum at $x = y$ and $KL(x, x) = 0$. Then $KL(x, y) \geq 0$. It is also obviously true that $KL(x, y) \to KL(x, x)$ if and only if $y \to x$.

We also make use of the equality

$$\sum_{i=1}^{n}(Af^k)_i = \sum_{j=1}^{m} f_j^k = \sum_{i=1}^{n} g_i, \text{ for } k \geq 1. \tag{5}$$

The first equality is true, since

$$\begin{aligned}\sum_{i=1}^{n}(Af^k)_i &= \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} f_j^k \\ &= \sum_{j=1}^{m} f_j^k \sum_{i=1}^{n} a_{ij} \\ &= \sum_{j=1}^{m} f_j^k.\end{aligned}$$

For the second equality we use (4). Then

$$\begin{aligned}\sum_{j=1}^{m} f_j^k &= \sum_{j=1}^{m} f_j^{k-1} \sum_{i=1}^{n} a_{ij} \frac{g_i}{(Af^{k-1})_i} \\ &= \sum_{i=1}^{n} \frac{g_i}{(Af^{k-1})_i} \sum_{j=1}^{m} a_{ij} f_j^{k-1} \\ &= \sum_{i=1}^{n} \frac{g_i}{(Af^{k-1})_i} (Af^{k-1})_i \\ &= \sum_{i=1}^{n} g_i.\end{aligned}$$

In the first step of our proof, we will show that

$$l(f^{k+1}) \geq l(f^k), \quad k = 1, 2, ... \tag{6}$$

We will use the identity

$$\sum_{i=1}^{n} g_i \log(Af)_i = \sum_{i=1}^{n} g_i \sum_{j=1}^{m} \frac{a_{ij} h_j}{(Ah)_i} \left[\log(a_{ij} f_j) - \log\left(\frac{a_{ij} f_j}{(Af)_i}\right)\right], \tag{7}$$

which holds for $h, f \in \mathbb{R}^n$ and $h, f > 0$. Then from (5) we have

$$l(f^{k+1}) - l(f^k) = \sum_{i=1}^{n} g_i \log(Af^{k+1})_i - \sum_{i=1}^{n} g_i \log(Af^k)_i.$$

And from (7) we have

$$\begin{aligned}
l(f^{k+1}) - l(f^k) &= \sum_{i=1}^{n} g_i \sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \left[\log(a_{ij} f_j^{k+1}) - \log\left(\frac{a_{ij} f_j^{k+1}}{(Af^{k+1})_i} \right) \right] \\
&- \sum_{i=1}^{n} g_i \sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \left[\log(a_{ij} f_j^{k}) - \log\left(\frac{a_{ij} f_j^{k}}{(Af^{k})_i} \right) \right] \\
&= \sum_{i=1}^{n} g_i \sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \left[\log\left(\frac{f_j^{k+1}}{f_j^k} \right) - \log\left(\frac{f_j^{k+1}(Af^k)_i}{f_j^k (Af^{k+1})_i} \right) \right] \\
&= \sum_{j=1}^{m} \log\left(\frac{f_j^{k+1}}{f_j^k} \right) \sum_{i=1}^{n} f_j^k a_{ij} \frac{g_i}{(Af^k)_i} \\
&- \sum_{i=1}^{n} g_i \sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \log\left(\frac{f_j^{k+1}(Af^k)_i}{f_j^k (Af^{k+1})_i} \right).
\end{aligned}$$

Now we have that

$$\sum_{i=1}^{n} f_j^k a_{ij} \frac{g_i}{(Af^k)_i} = f_j^{k+1}.$$

And since

$$\sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} = 1,$$

$\forall i$ such that $1 \leq i \leq n$ and log is a concave function, we may apply Jensen's inequality to get that

$$\begin{aligned}
\sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \log\left(\frac{f_j^{k+1}(Af^k)_i}{f_j^k (Af^{k+1})_i} \right) &\leq \log \sum_{j=1}^{m} \frac{a_{ij} f_j^k}{(Af^k)_i} \left(\frac{f_j^{k+1}(Af^k)_i}{f_j^k (Af^{k+1})_i} \right) \\
&= \log \sum_{j=1}^{m} \frac{a_{ij} f_j^{k+1}}{(Af^{k+1})_i}.
\end{aligned}$$

Then

$$\begin{aligned}
l(f^{k+1}) - l(f^k) &\geq \sum_{j=1}^{m} f_j^{k+1} \log\left(\frac{f_j^{k+1}}{f_j^k} \right) - \sum_{i=1}^{n} g_i \log \sum_{j=1}^{n} \frac{a_{ij} f_j^{k+1}}{(Af^{k+1})_i} \\
&= KL(f^{k+1}, f^k) + \sum_{j=1}^{m} (f_j^{k+1} - f_j^k) - \sum_{i=1}^{n} g_i \log 1.
\end{aligned}$$

By (5) we see that

$$\sum_{j=1}^{m} (f_j^{k+1} - f_j^k) = 0.$$

Thus

$$l(f^{k+1}) - l(f^k) \geq KL(f^{k+1}, f^k) \geq 0. \tag{8}$$

And we have our first step.

Now we will address our possible divide by zero and log(0) problems. The instances where we have $f_j^{k+1}\log(f_j^{k+1}/f_j^k)$ are covered by the fact that $f_j^k = 0$ implies that $f_j^{k+1} = 0$ and that we use the convention $0/0 = 0$ and $0\log 0 = 0$. Our remaining problems are with the $(Af^k)_i$ as a denominator. Apply A to both sides of (4), the iteration of the EM Algorithm. Then we have $Af^{k+1} = Af^k A^T\frac{g}{A^k}$. Since A and f are element-wise nonnegative and g is element-wise positive, by an easy inductive argument we see that

$$(Af^{k+1})_i \geq \frac{g_i\sum_{j=1}^m a_{ij}^2 f_j^k}{\sum_{j=1}^m a_{ij}f_j^k} > 0$$

for $i = 1,\ldots,n$. Thus $(Af^k)_i > 0$ for all $k \geq 0$, $1 \leq i \leq n$.

For our second step, we will show that each accumulation point f^* of the sequence $\{f^k\}$ is a fixed point of (4). Such an accumulation point does exist, since $0 \leq ||f^k||_1 = ||g||_1\ \forall k \geq 2$. Let $B(f^k) = l(f^{k+1}) - l(f^k)$. Since $l(f^k)$ is monotonically nondecreasing and bounded above (l is concave), $l(f^k)$ has a limit. Let $l^* = \lim_{k\to\infty} l(f^k)$. Hence $B(f^k) \to 0$. Let f^* be an accumulation point of $\{f^k\}$. Then there exists a subsequence f^{k_s} such that $f^{k_s} \to f^*$. Since $l(\cdot)$ is a continuous function, $B(f^{k_s}) \to B(f^*)$. But $B(f^k) \to 0$, so $B(f^*) = 0$. Now let $f^{*'} = f^* A^T\frac{g}{Af^*}$. Then from (8) $B(f^*) = l(f^{*'}) - l(f^*) \geq KL(f^{*'}, f^*) \geq 0$. Since $B(f^*) = 0$, $KL(f^{*'}, f^*) = 0$. Then $KL(f^{*'}, f^*) = 0$ implies that $f^{*'} = f^*$ i.e. f^* is a fixed point of (4).

For our third step, we will show that

$$KL(f^*, f^{k+1}) \leq KL(f^*, f^k), \quad k = 1, 2, ..., \tag{9}$$

where f^* is an accumulation point. Let

$$x_i^j = \frac{a_{ij}g_i/(Af^*)_i}{(A^T(g/Af^*))_j}, \quad y_i^j = \frac{a_{ij}g_i/(Af^k)_i}{(A^T(g/Af^k))_j}, \quad i = 1, ..., n.$$

Then $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i = 1$. The following sums are over those j such that $f_j^* > 0$. Then we have

$$\begin{aligned}
0 &\leq \sum_{j=1}^m f_j^* KL(x^j, y^j) \\
&= \sum_{j=1}^m f_j^* \sum_{i=1}^n x_i^j \log\frac{x_i^j}{y_i^j} \\
&= \sum_{j=1}^m f_j^* \sum_{i=1}^n \frac{a_{ij}g_i/(Af^*)_i}{(A^T(g/Af^*))_j} \log\frac{(Af^k)_i(A^T(g/Af^k))_j}{(Af^*)_i(A^T(g/Af^*))_j}.
\end{aligned}$$

Now by (4) and the fact that f^* is a fixed point of (4) we have

$$\begin{aligned}
0 &\leq \sum_{j=1}^{m} f_j^* \sum_{i=1}^{n} \frac{a_{ij} g_i/(Af^*)_i}{(A^T(g/Af^*))_j} \log \frac{(Af^k)_i f_j^{k+1} f_j^*}{(Af^*)_i f_j^* f_j^k} \\
&= \sum_{j=1}^{m} f_j^* \sum_{i=1}^{n} \frac{a_{ij} g_i/(Af^*)_i}{(A^T(g/Af^*))_j} \log \frac{(Af^k)_i f_j^{k+1}}{(Af^*)_i f_j^k} \\
&= \sum_{j=1}^{m} f_j^* \sum_{i=1}^{n} a_{ij} \frac{g_i}{(Af^*)_i} \log \frac{(Af^k)_i f_j^{k+1}}{(Af^*)_i f_j^k} \\
&= \sum_{j=1}^{m} f_j^* \sum_{i=1}^{n} a_{ij} \frac{g_i}{(Af^*)_i} \left(\log \frac{(Af^k)_i}{(Af^*)_i} + \log \frac{f_j^{k+1}}{f_j^k} \right) \\
&= \sum_{i=1}^{m} \frac{g_i}{(Af^*)_i} \log \frac{(Af^k)_i}{(Af^*)_i} \sum_{j=1}^{n} f_j^* a_{ij} + \sum_{j=1}^{m} f_j^* \log \frac{f_j^{k+1}}{f_j^k} \sum_{i=1}^{n} a_{ij} \frac{g_i}{(Af^*)_i} \\
&= \sum_{i=1}^{n} g_i \log \frac{(Af^k)_i}{(Af^*)_i} + \sum_{j=1}^{m} f_j^* \log \frac{f_j^{k+1}}{f_j^k}.
\end{aligned}$$

We see that

$$0 \leq \sum_{i=1}^{n} g_i \log \frac{(Af^k)_i}{(Af^*)_i} + \sum_{j=1}^{m} f_j^* \log \frac{f_j^{k+1}}{f_j^k}$$

even when we let the j sum extend over those such that $f_j^* = 0$. Then by (4) we have

$$\sum_{i=1}^{n} f_i^k = \sum_{i=1}^{n} f_i^{k+1} = \sum_{i=1}^{n} f_i^*$$

and thus

$$KL(f^*, f^k) = \sum_{j=1}^{m} f_j^* \log \frac{f_j^*}{f_j^k} + f_j^k - f_j^* = \sum_{j=1}^{m} f_j^* \log \frac{f_j^*}{f_j^k}.$$

Then we have

$$\begin{aligned}
\sum_{i=1}^{n} g_i \log \frac{(Af^k)_i}{(Af^*)_i} &= l(f^k) - l(f^*) + \sum_{i=1}^{n} ((Af^k)_i - (Af^*)_i) \\
&= l(f^k) - l(f^*)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{j=1}^{m} f_j^* \log \frac{f_j^{k+1}}{f_j^k} &= \sum_{j=1}^{m} f_j^* \log f_j^{k+1} - \sum_{j=1}^{m} f_j^* \log f_j^k \\
&= \sum_{j=1}^{m} f_j^* \log f_j^{k+1} - \sum_{j=1}^{m} f_j^* \log f_j^* + \sum_{j=1}^{m} f_j^* \log f_j^* - \sum_{j=1}^{m} f_j^* \log f_j^k \\
&= \sum_{j=1}^{m} f_j^* \log \frac{f_j^{k+1}}{f_j^*} + \sum_{j=1}^{m} f_j^* \log \frac{f_j^*}{f_j^k} \\
&= KL(f^*, f^k) - KL(f^*, f^{k+1}).
\end{aligned}$$

Thus we have

$$0 \leq l(f^k) - l(f^*) + KL(f^*, f^k) - KL(f^*, f^{k+1})$$

and since $l(f^{k+1}) \geq l(f^k)$ implies $l(f^*) \geq l(f^k)$,

$$KL(f^*, f^{k+1}) \leq KL(f^*, f^k).$$

Now we are ready to prove our theorem. Since $\left\{f^k\right\}$ is bounded, there exists a convergent subsequence. Take f^* as the limit of the subsequence $\left\{f^{k_s}\right\}$. It follows that $KL(f^*, f^{k_s}) \to KL(f^*, f^*) = 0$ as $s \to \infty$. But since $KL(f^*, f^k)$ is nonincreasing, $KL(f^*, f^k) \to 0$ as $k \to \infty$. Hence $f^k \to f^*$. Now that we have proved convergence we must show that the Kuhn-Tucker conditions are satisfied. If $f_j^* > 0$, then $1 = (A^T \frac{g}{(Af^*)})_j$ by (4). Then the Kuhn-Tucker conditions are satisfied for $f_j^* > 0$. Now for $f_j^* = 0$, we have

$$f_j^{k+1} = f_j^0 \left(A^T \frac{g}{Af^0}\right)_j \cdots \left(A^T \frac{g}{Af^k}\right)_j \to f_j^* = 0.$$

Since $f_j^0 > 0$, we see that $(A^T g/Af^k)_j \to (A^T g/Af^*)_j$. But for this to converge we must have $(A^T g/Af^*)_j \leq 1$. Thus

$$\frac{\partial l}{\partial f_j}(f^*) = \left(A^T \frac{g}{Af^*}\right)_j - 1 \leq 1 - 1 = 0.$$

□

We have proven more than convergence in the proof above. From (8) we see that the iterate f^k gets closer to the maximum likelihood with every iteration.

The problems with the EM algorithm are the slow computation time for (4) and the lack of smoothness of the solution. The smoothness can be increased by adding a penalty term to the iteration of (4) in the form of a Bayesian function. The computations can be accelerated by splitting A and g into submatrices A_j and g_j for $j = 1, ...p$, such that

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix}, \quad g = \begin{pmatrix} g_1 \\ \vdots \\ g_p \end{pmatrix},$$
$$A_j : \mathbb{R}^m \to \mathbb{R}^{n_j}, \qquad g_j \in \mathbb{R}^{n_j},$$

and performing (4) on each submatrix. Here we assume that $a = A_j^T 1$ for all j. Then we get

$$\begin{aligned} f^{k,0} &= f^k, \\ f^{k,j} &= f^{k,j-1} \frac{1}{a} A_j^T \frac{g_j}{A_j f^{k,j-1}}, \quad j = 1, ..., p, \\ f^{k+1} &= f^{k,p}. \end{aligned}$$

This is called the ordered subset EM (OSEM) algorithm. The convergence of OSEM has not yet been proven.

3 Inversion Formula

3.1 Model

For the inversion formula we model the activity distribution by an integral equation. Let $a : \mathbb{R}^2 \to \mathbb{R}$ and $f : \mathbb{R}^2 \to \mathbb{R}$ be sufficiently smooth functions with compact support, where a is the density and f is the concentration of the radiopharmaceutical along a plane in the body. Let $\theta = \begin{pmatrix}\cos\varphi \\ \sin\varphi\end{pmatrix}$, $\theta^\perp = \begin{pmatrix}-\sin\varphi \\ \cos\varphi\end{pmatrix}$, and $x \in \mathbb{R}^2$. Let $I_{\theta^\perp,x}(y)$ be the sum of the photons (intensity) measured over a fixed time interval $[0,\tau]$ emitted from a point x, traveling in direction $\theta^\perp$, and at the point y that have not interacted with the body. Note that $y = x + t\theta^\perp$ for some $t \geq 0$. Then from the physics of the absorbtion and deflection rates of the photons in relation to the attenuation factor we have the relation

$$I_{\theta^\perp,x}(x + (t+\Delta t)\theta^\perp) - I_{\theta^\perp,x}(x + t\theta^\perp) = -a(x + t\theta^\perp) I_{\theta^\perp,x}(x + t\theta^\perp)\Delta t$$

for small Δt. Then if we take $\Delta t \to 0$ and rearrange terms, we have that

$$\frac{dI_{\theta^\perp,x}(x + t\theta^\perp)}{I_{\theta^\perp,x}(x + t\theta^\perp)} = -a(x + t\theta^\perp)\, dt.$$

By integrating both sides from $t = 0$ to ∞ we get that

$$\ln(I_{\theta^\perp,x}(x + t\theta^\perp))\Big|_0^\infty = -\int_0^\infty a(x + t\theta^\perp)\, dt.$$

The infinity term is slightly misleading. In practice, the function a has compact support, so integrating all the way to infinity makes no difference. Then we define $I_m(\theta^\perp, x)$ to be the measured intensities of the photons that were emitted from a point x and traveled in the direction $\theta^\perp$. We define the divergent beam transform by

$$(Da)(x,\theta^\perp) = \int_0^\infty a(x + t\theta^\perp)\, dt.$$

Then we get that

$$I_m(\theta^\perp, x) = I_{\theta^\perp,x}(x)e^{-Da(x,\theta^\perp)}.$$

In practice, we do not know the values of $I_{\theta^\perp,x}(x)$ because all photons emitted along a straight line emanating from a detector are measured all at once without knowing where the photons originated on that line. Thus the detectors actually measure the sum of the $I_m(\theta^\perp, x)$ with x varying over the lines emanating from the detector and parallel to $\theta^\perp$. This restriction can be expressed as $x \cdot \theta = s$ (see figure 1). These measurements are simply the line integrals

$$I_M(\theta, s) \equiv \int_{x\cdot\theta=s} I_{\theta^\perp,x}(x)e^{-Da(x,\theta^\perp)}\, dx.$$

Now since f is proportional to I and we assume that photons are given off in an isotropic manner, there exists c such that $I_{\theta^\perp,x}(x) = cf(x)$. Let $g(\theta, s) = \frac{1}{c}I_M(\theta, s)$. Thus we have

$$R_a f(\theta, s) \equiv g(\theta, s) = \int_{x\cdot\theta=s} f(x)e^{-Da(x,\theta^\perp)}\, dx.$$

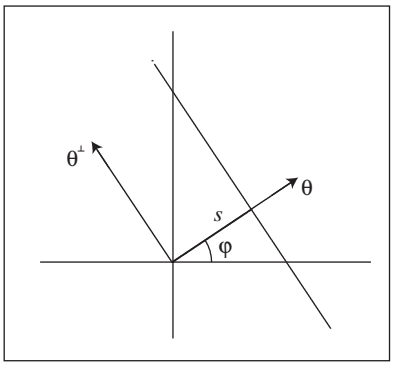

Figure 1: Line Parameterized by $x \cdot \theta = s$.

The integral transform, $R_a f$, is referred to as the attenuated Radon transform and

$$Rf(\theta, s) \equiv R_0 f(\theta, s) = \int_{x \cdot \theta = s} f(x)\, dx$$

is the Radon transform. In this paper we assume that a is known (in practice a is determined by a transmission scan) and g are the photon intensities measured by the gamma camera. We will then develop an inversion formula for f that was first found by Novikov, but we will follow a derivation by Natterer[7].

3.2 Theory

We will use the following function throughout this discussion. Let $\mathrm{h} = \frac{1}{2}(I + iH)(Ra)$ where I is the identity transformation and H is the Hilbert transform given by,

$$(Hg)(s) = \frac{1}{\pi}\,\text{p.v.} \int_{R^1} \frac{g(t)}{s - t}\, dt$$

acting on the second argument of Ra. From here on we will omit the "p.v.". If the integral does not exist in the traditional sense it should always be interpreted as the principle value.

Necessary and sufficient conditions on the functions f and a such that the inversion formula holds have not yet been found. That is why we simply stated above that a and f are "sufficiently smooth". Thus we will consider $a, f \in \mathcal{C}_0^\infty(\mathbb{R}^2)$ in the following discussion.

Lemma 3.1. *Let*

$$G(z) = \frac{1}{2\pi i} \int_L \frac{g(t)}{t - z}\, dt,$$

where L is an oriented (counterclockwise) simple smooth closed curve (with a finite number of arcs) and g satisfies the Holder condition on L

$$|g(t) - g(t_0)| \leq A|t - t_0|^\mu \quad \forall t, t_0 \in L \quad A, \mu > 0.$$

Then G is analytic on $\mathbb{C}\backslash L$. Moreover, if $t_0 \in L$, then the limit from the left (inside) of L exists and is given by

$$G_+(t_0) = \frac{1}{2} g(t_0) + \frac{1}{2\pi i} p.v. \int_L \frac{g(t)}{t - t_0}\, dt$$

and the limit from the right (outside) of L exists and is given by

$$G_-(t_0) = -\frac{1}{2}g(t_0) + \frac{1}{2\pi i} p.v. \int_L \frac{g(t)}{t - t_0}\, dt.$$

This is just a generalization the Plemelj-Sokhozki formulae. See [9] for the proof of L as a C^1 oriented simple path (L not necessarily closed).

Proof. This proof comes from a collection of theorems and lemmas in [9]. Note that the function g here has no connection to the function g in the description of the inversion formula model. Now clearly G exists and is analytic on $\mathbb{C}\backslash L$. Now we will show that $G(t_0)$ exists for $t_0 \in L$.

Let l be the part of L that has length $2\varepsilon_1$ and center t_0. Denote the ends of l by t' and t''. Then

$$\int_{L-l} \frac{g(t)}{t - t_0}\, dt = \int_{L-l} \frac{g(t) - g(t_0)}{t - t_0}\, dt + g(t_0) \int_{L-l} \frac{dt}{t - t_0}.$$

Then the last integral is

$$\begin{aligned} \int_{L-l} \frac{dt}{t - t_0} &= \log(t - t_0)\Big|_{t'}^{t''} \\ &= \log(t'' - t_0) - \log(t' - t_0) \\ &= \log\left(\frac{|t'' - t_0|}{|t' - t_0|}\right) + i(\arg(t'' - t_0) - \arg(t' - t_0)). \end{aligned}$$

Here, $\log(t - t_0)$ is a branch which continuously changes on the arc and thus

$$\int_L \frac{dt}{t - t_0} = \lim_{\varepsilon \to 0} \log\left(\frac{|t'' - t_0|}{|t' - t_0|}\right) + i(\arg(t'' - t_0) - \arg(t' - t_0)) = i\pi, \tag{10}$$

since L is smooth. Then for the other integral

$$\left| \int_{L-l} \frac{g(t) - g(t_0)}{t - t_0}\, dt \right| \leq A \int_{L-l} |t - t_0|^{\mu - 1}\, |dt| < \infty. \tag{11}$$

Therefore the principle value of the integral exists and is given by

$$\begin{aligned} G(t_0) &= \frac{1}{2\pi i} \int_L \frac{g(t)}{t - t_0}\, dt \\ &= \frac{1}{2} g(t_0) + \frac{1}{2\pi i} \int_L \frac{g(t) - g(t_0)}{t - t_0}\, dt. \end{aligned}$$

Moreover, we see from (11) that G is bounded on L.

Let $\varepsilon_2 > 0$ and $0 < \beta_0 \leq \frac{\pi}{2}$ be given and let z approach t_0 in such a way that the non-obtuse angle β between the tangent to L at the point t_0 and the segment $t_0 z$ is greater than or equal to β_0. Then we will show that

$$F(z) = \frac{1}{2\pi i} \int_L \frac{g(t) - g(t_0)}{t - z}\, dt$$

converges uniformly to

$$F(t_0) = \frac{1}{2\pi i} \int_L \frac{g(t) - g(t_0)}{t - t_0}\, dt.$$

We will then use this uniform convergence to show that $F(z) \to F(t_0)$ for $z \to t_0$ with the only restriction that z stays either on this inside or outside of L.

Parameterize L by $x = x(s)$, $y = y(s)$ such that $[x'(s)]^2 + [y'(s)]^2 = 1$. Then the arc length of the arc t_1t_2, where $t_i = (x(s_i), y(s_i))$ is $|s_2 - s_1|$. Let γ be a circle with center t_0 and radius ρ taken so small that γ intersects L at exactly two points, say t_1 and t_2, and the non-obtuse angle between the chord joining any two points on the arc t_1t_2 and the tangent at t_1 does not exceed α_0, for $0 \leq \alpha_0 < \beta_0 \leq \pi/2$. Now let $r = r(t)$ be the distance between t_0 and $t \in t_1t_2$ and α be the acute angle between the chord t_0t and the tangent at the point t. We will show that

$$\frac{dr}{ds} = \pm \cos \alpha.$$

The "+" refers to the part $s > s_0$ and the "-" refers to $s < s_0$. Without loss of generality, suppose that $t_0 = (x(s_0), y(s_0)) = (0, 0)$. Let $\tau = -(x'(s), y'(s))$ and $\xi = -(x(s), y(s))$. Note that $||\xi|| = r$ and $||\tau|| = 1$. Then for $s > s_0$

$$\frac{dr}{ds} = \frac{1}{r}(\xi, \tau) = \frac{1}{r}||\xi||\,||\tau|| \cos \alpha = \cos \alpha.$$

Then since $0 \leq \alpha \leq \alpha_0 < \pi/2$, $\cos \alpha \geq \cos \alpha_0 = 1/K > 0$ and thus $|ds| \leq K|dr|$. We see that since L has only a finite number of arcs, we may also take ρ so small that t_1t_2 consists of a single arc. We will refer to the arc t_1t_2 with this restriction as the standard arc. Now since L is compact, we see that L can be decomposed into a finite number of standard arcs.

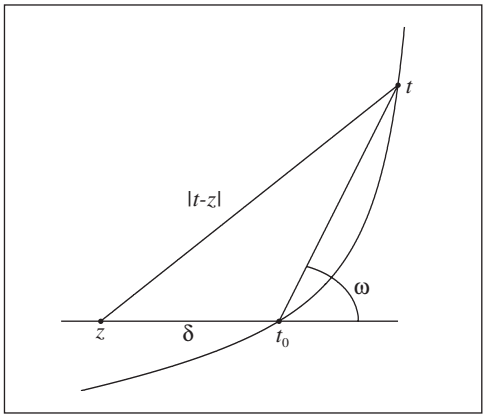

Figure 2: Geometry of the Convergence of z to t_0.

Consider the difference

$$F(z) - F(t_0) = \frac{z - t_0}{2\pi i} \int_L \frac{g(t) - g(t_0)}{(t - t_0)(t - z)}\, dt.$$

Now split $|F(z) - F(t_0)| = |I_1 + I_2| \leq |I_1| + |I_2|$, where

$$I_1 = \frac{z - t_0}{2\pi i} \int_{t_1t_2} \frac{g(t) - g(t_0)}{(t - t_0)(t - z)}\, dt, \quad I_2 = \frac{z - t_0}{2\pi i} \int_{L - t_1t_2} \frac{g(t) - g(t_0)}{(t - t_0)(t - z)}\, dt.$$

First we will consider I_1. Since $\frac{dr}{ds} = \pm\cos\alpha$, one obtains $|dt| = |ds| \leq K|dr|$. Then

$$\begin{aligned} |I_1| &\leq \frac{|z-t_0|}{2\pi}\int_{t_1t_2}\frac{|g(t)-g(t_0)|}{|t-t_0||t-z|}\,dt \\ &\leq \frac{A|z-t_0|}{2\pi}\int_{t_1t_2}\frac{|t-t_0|^{\mu-1}}{|t-z|}\,dt \\ &\leq \frac{\delta_2 AK}{2\pi}\int_{t_1t_2}\frac{r^{\mu-1}}{|t-z|}\,|dr|, \end{aligned}$$

where $\delta_2 = |z-t_0|$. Now let ω be the non-obtuse angle between the segments t_0t and t_0z. If $|t-z| \geq \delta_2$, then $|t-z| \geq \delta_2\sin(\omega)$. Otherwise, ω is an interior angle of the z, t_0, t triangle and we also have $|t-z| \geq \delta_2\sin(\omega)$. Then $|t-z| \geq \delta_2\sin(\omega) \geq \delta_2\sin(\omega_0)$, where ω_0 is a constant such that $0 < \omega_0 \leq \omega \leq \pi/2$. Such a ω_0 does exist, since z approaches t_0 in such a way that the non-obtuse angle β between the tangent to L at t_0 and the segment t_0z is greater than or equal to $0 < \beta_0 \leq \pi/2$. Explicitly, we have $\omega_0 = \beta_0 - \alpha_0 > 0$. Thus

$$|I_1| \leq \frac{AK}{2\pi\sin(\omega_0)}\int_{t_1t_2} r^{\mu-1}\,|dr| \leq \frac{AK}{\pi\sin(\omega_0)}\int_0^\rho r^{\mu-1}\,dr = \frac{AK\rho^\mu}{\pi\mu\sin(\omega_0)}.$$

Take ρ so small that $|I_1| < \varepsilon_2/2$. Notice that the choice of ρ may be taken independent of the position of t_0 on L and of the position of z. Now let $\delta_2 \leq \rho/2$. Then for t on $L - t_1t_2$, i.e., outside of the circle γ, $|t-t_0| \geq \rho$ and $|t-z| \geq |t-t_0| - |z-t_0| \geq \rho - \rho/2 = \rho/2$ and thus

$$\begin{aligned} |I_2| &\leq \frac{|z-t_0|}{2\pi}\int_{L-t_1t_2}\frac{|g(t)-g(t_0)|}{|t-t_0||t-z|}\,dt \\ &\leq \frac{\delta_2}{2\pi}\int_{L-t_1t_2}\frac{|g(t)-g(t_0)|}{\rho(\rho/2)}\,dt \\ &\leq \frac{\delta_2}{\pi\rho^2}\int_{L-t_1t_2}|g(t)-g(t_0)|\,dt \\ &\leq \frac{\delta_2 M}{\pi\rho^2}, \end{aligned}$$

where M is a constant which depends neither on the position of t_0 on L nor the position of z. Hence for sufficiently small δ_2, $|I_2| < \varepsilon_2/2$. Therefore $|F(z)-F(t_0)| < \varepsilon_2$, i.e. $F(z)$ converges uniformly to $F(t_0)$ for the above set of trajectories.

Let U_{t_0} denote the set of those $z \in \mathbb{C}\backslash L$ such that the non-obtuse angle β between the tangent to L at t_0 and the segment t_0z is greater that or equal to β_0. In other words, U_{t_0} is a double cone-shaped region with vertex t_0. If $z \to t_0$ while z stays in U_{t_0}, then we have shown above that $F(z)$ converges uniformly to $F(t_0)$. Now we will show that $F(z) \to F(t_0)$ as $z \to t_0$ for any path. To do this we will first show that $F(t)$ is continuous for $t \in L$.

Let $\varepsilon_3 > 0$ be given. Then $\exists\delta_3 > 0$ such that $|z-t| < \delta_3$ and $z \in U_t$ implies that $|F(z)-F(t)| < \varepsilon_3/2$ for all $t \in L$ because of the uniformity of the convergence. Let $t_0 \in L$ be given. Let $z'_n \to t_0$ such that the segment z'_nt_0 is perpendicular to the tangent of L at t_0 for all $n \geq 0$. Clearly $z'_n \in U_{t_0}$ for all $n \geq 0$. Then there exists $N_1 \in \mathbb{N}$ such that $|z'_n - t_0| < \delta_3$ for all $n \geq N_1$. Now let $0 < \overline{\delta_3} < \delta_3/2$ be taken so small that $z'_{N_1} \in U_t$ for all $t \in B_{\overline{\delta_3}}(t_0) \cap L$. Now let $t' \in B_{\overline{\delta_3}}(t_0) \cap L$. Then

$$|z'_{N_1} - t'| \leq |z'_{N_1} - t_0| + |t_0 - t'| < \frac{\delta_3}{2} + \frac{\delta_3}{2} = \delta_3.$$

Hence $|F(z'_{N_1}) - F(t')| < \varepsilon_3/2$. Then

$$|F(t_0)-F(t')| \le |F(t_0)-F(z'_{N_1})|+|F(z'_{N_1})-F(t')| < \epsilon_3 \quad \forall t' \in L \text{ with } |t_0-t'| < \overline{\delta_3}.$$

Thus $F(t)$ is continuous on L.

Now let ab be a standard arc on L. Let Π be the family of parallel lines which make with the tangents to ab a non-obtuse angle not less than β_0. Then we see that each straight line Δ of the family Π which lies between Δ_a and Δ_b (the straight lines through a and b, respectively) cuts the arc ab in one and only one point. We also note that if $z \to t_0 \in ab$ and z varies over the straight line in Π that cuts L at t_0, then $F(z) \to F(t_0)$ uniformly. Now let $\{z_n\}$ be a sequence such that $z_n \to t_0$ along any path that stays entirely to the left (or right) of L. Then for each $n \in \mathbb{N}$ $\exists t_n \in ab$ and $\exists \Delta_n \in \Pi$ with $z_n, t_n \in \Delta_n$. Let ψ_n be the non-obtuse angle between $t_n t_0$ and $t_n z_n$. Then since Δ_n makes an angle no less than β_0 with the tangent at t_0, $\exists \psi_0$ such that $0 < \psi_0 \le \psi_n \le \pi/2$. Now consider the $z_n t_n t_0$ triangle with inner angle ψ_n. Then we see that

$$|t_n - t_0| \le \frac{|z_n - t_0|}{\sin\psi_n} \le \frac{|z_n - t_0|}{\sin\psi_0} \text{ and } |z_n - t_n| \le \frac{|z_n - t_0|}{\sin\psi_n} \le \frac{|z_n - t_0|}{\sin\psi_0}.$$

Thus we may make $|t_n - t_0|$ and $|z_n - t_n|$ arbitrarily small. Let $\epsilon_4 > 0$ be given. Now choose $\delta_4 > 0$ so small that $|t_n - t_0| < \delta_4$ implies that $|F(t_n) - F(t_0)| < \epsilon_4/2$ and $|z_n - t_n| < \delta_4$ implies that $|F(z_n) - F(t_n)| < \epsilon_4/2$. Take $N_2 \in \mathbb{N}$ so large that $|z_n - t_n| < \delta_4$ and $|t_n - t_0| < \delta_4$ for all $n \ge N_2$. Then

$$|F(z_n) - F(t_0)| \le |F(z_n) - F(t_n)| + |F(t_n) - F(t_0)| < \frac{\epsilon_4}{2} + \frac{\varepsilon_4}{2} = \varepsilon_4$$

for all $n \ge N_2$. Therefore $F(z) \to F(t_0)$ for all $t_0 \in L$.

Now we have that

$$G(z) = \frac{g(t_0)}{2\pi i}\int_L \frac{dt}{t-z} + F(z).$$

Then by using (10) and the fact that

$$\frac{1}{2\pi i}\int_L \frac{dt}{t-z} = \begin{cases} 0, & \text{z is outside of L,} \\ 1, & \text{z is inside of L,} \end{cases}$$

we have

$$\begin{aligned} G_+(t_0) &= g(t_0) + F(t_0) \\ &= g(t_0) + \frac{1}{2\pi i}\int_L \frac{g(t)-g(t_0)}{t-t_0}\,dt \\ &= \frac{1}{2}g(t_0) + \frac{1}{2\pi i}\int_L \frac{g(t)}{t-t_0}\,dt \end{aligned}$$

and

$$\begin{aligned} G_-(t_0) &= F(t_0) \\ &= \frac{1}{2\pi i}\int_L \frac{g(t)-g(t_0)}{t-t_0}\,dt \\ &= -\frac{1}{2}g(t_0) + \frac{1}{2\pi i}\int_L \frac{g(t)}{t-t_0}\,dt. \end{aligned}$$

□

For the rest of this discussion let L be the unit circle oriented counterclockwise.

Corollary 3.2. *Let G and g be as in the previous lemma. Then for $\theta \in L$,*

$$G_+(-\theta^\perp) - G_+(\theta^\perp) = \frac{1}{2}(g(-\theta^\perp) - g(\theta^\perp)) + \frac{1}{2\pi i}\int_{S^1} \frac{g(\omega)}{\omega \cdot \theta}\, d\omega.$$

Proof. The proof of this corollary was outlined in [4]. Let $\xi = e^{i\sigma}$, where $\theta = (\cos\sigma, \sin\sigma)$ and $v = e^{i\psi}$, for $\omega = (\cos\psi, \sin\psi)$. Then $\theta^\perp = i\xi$ and thus

$$\begin{aligned} G_+(-\theta^\perp) - G_+(\theta^\perp) &= G_+(-i\xi) - G_+(i\xi) \\ &= \frac{1}{2}(g(-i\xi) - g(i\xi)) + \frac{1}{2\pi i}\int_L \left(\frac{1}{v + i\xi} - \frac{1}{v - i\xi}\right) g(v)\, dv \\ &= \frac{1}{2}(g(-i\xi) - g(i\xi)) + \frac{1}{2\pi i}\int_L \left(\frac{-2i\xi}{v^2 + \xi^2}\right) g(v)\, dv. \end{aligned}$$

Since $dv = iv\, d\psi$,

$$G_+(-i\xi) - G_+(i\xi) = \frac{1}{2}(g(-i\xi) - g(i\xi)) + \frac{1}{2\pi i}\int_0^{2\pi} \frac{2v\xi}{v^2 + \xi^2}\, g(e^{i\psi})\, d\psi.$$

But

$$\begin{aligned} \frac{1}{2}(\bar{\xi}v + \bar{v}\xi) &= \frac{1}{2}(\cos\sigma\cos\psi + \sin\sigma\sin\psi + i(-\sin\sigma\cos\psi + \cos\sigma\sin\psi) \\ &+ \cos\sigma\cos\psi + \sin\sigma\sin\psi + i(-\cos\sigma\sin\psi + \sin\sigma\cos\psi)) \\ &= \cos\sigma\cos\psi + \sin\sigma\sin\psi \\ &= \omega \cdot \theta \end{aligned}$$

and

$$(\bar{\xi}v + \bar{v}\xi)v\xi = |\xi|^2 v^2 + |v|^2\xi^2 = v^2 + \xi^2,$$

so

$$\omega \cdot \theta = \frac{1}{2}(\bar{\xi}v + \bar{v}\xi) = \frac{v^2 + \xi^2}{2v\xi}.$$

Therefore we have

$$\int_0^{2\pi} \frac{2v\xi}{v^2 + \xi^2}\, g(e^{i\psi})\, d\psi = \int_{S^1} \frac{1}{\omega \cdot \theta}\, g(\omega)\, d\omega,$$

which concludes our proof. □

Lemma 3.3. *Let $u(x,\theta) = h(\theta, x\cdot\theta) - (Da)(x,\theta^\perp)$, where $h = \frac{1}{2}(I + iH)Ra$. Now let $g_x(\theta^\perp) = Da(x,\theta^\perp)$. If we replace the function g in lemma 3.1 with this g_x we get that $u(x,\theta) = G_{x,+}(-\theta^\perp) - G_{x,+}(\theta^\perp)$, where x is a fixed parameter.*

Proof. We will follow an outline for this proof given in [4]. Since $Ra(\theta, x\cdot\theta) = Da(x, -\theta^\perp) + Da(x,\theta^\perp)$,

$$\begin{aligned} u(x,\theta) &= \frac{1}{2}Ra(\theta, x\cdot\theta) + \frac{i}{2}HRa(\theta, x\cdot\theta) - (Da)(x,\theta^\perp) \\ &= \frac{1}{2}\left(Da(x,-\theta^\perp) - Da(x,\theta^\perp)\right) - \frac{1}{2i}HRa(\theta, x\cdot\theta). \end{aligned}$$

Now we will show that

$$(HRa)(\theta, x\cdot\theta) = -\frac{1}{\pi}\int_{S^1}\frac{Da(x,\omega)}{\theta\cdot\omega}\,d\omega.$$

Then

$$\begin{aligned}
(HRa)(\theta, x\cdot\theta) &= \frac{1}{\pi}\int_{\mathbb{R}}\frac{Ra(\theta,t)}{x\cdot\theta - t}\,dt \\
&= \frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a(t\theta + v\theta^{\perp})}{x\cdot\theta - t}\,dv\,dt \\
&= -\frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a((x\cdot\theta - t)\theta + v\theta^{\perp})}{t}\,dv\,dt \\
&= -\frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a((x\cdot\theta + t)\theta + v\theta^{\perp})}{t}\,dv\,dt \\
&= -\frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a(x - (x\cdot\theta^{\perp})\theta^{\perp} + t\theta + v\theta^{\perp})}{t}\,dv\,dt \\
&= -\frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a(x + t\theta + v\theta^{\perp})}{t}\,dv\,dt.
\end{aligned}$$

Now let $y = t\theta + v\theta^{\perp}$. Then

$$-\frac{1}{\pi}\int_{\mathbb{R}}\int_{\mathbb{R}}\frac{a(x + t\theta + v\theta^{\perp})}{t}\,dv\,dt = -\frac{1}{\pi}\int_{\mathbb{R}^2}\frac{a(x+y)}{y\cdot\theta}\,dy.$$

Then let $y = r\omega$ for $\omega \in S^1$, $r \in \mathbb{R}$. Then we have

$$\begin{aligned}
-\frac{1}{\pi}\int_{\mathbb{R}^2}\frac{a(x+y)}{y\cdot\theta}\,dy &= -\frac{1}{\pi}\int_{S^1}\int_0^{\infty}\frac{a(x + r\omega)}{\omega\cdot\theta}\,dr\,d\omega \\
&= -\frac{1}{\pi}\int_{S^1}\frac{Da(x,\omega)}{\theta\cdot\omega}\,d\omega.
\end{aligned}$$

Now recall the functions G and g from our previous lemma. Now let $g_x(\theta^{\perp}) = Da(x, \theta^{\perp})$, where x is a fixed parameter. Since $a \in \mathcal{C}_0^{\infty}(\mathbb{R}^2)$, $Da(x,\theta)$ certainly satisfies the Holder condition in θ. Then $u(x,\theta) = G_{x,+}(-\theta^{\perp}) - G_{x,+}(\theta^{\perp})$.

□

Lemma 3.4. *Let* $\omega^{\perp} = \frac{x^{\perp}}{|x|} = (\cos\psi, \sin\psi)^T$ *and* $\theta = (\cos\varphi, \sin\varphi)^T$. *Then*

$$\int_0^{2\pi}\frac{\theta}{x\cdot\theta}e^{il\varphi}d\varphi = \begin{cases} 0, & l \text{ odd}, \\ 2\pi x/|x|^2, & l = 0, \\ -sgn(l)2\pi i e^{il\psi}x^{\perp}/|x|^2, & l \text{ even}. \end{cases}$$

The integral is to be understood as the principle value.

Proof. Then

$$\int_0^{2\pi}\frac{\theta}{x\cdot\theta}e^{il\varphi}\,d\varphi = \frac{1}{|x|}\int_{S^1}\frac{\theta^{l+1}}{\omega\cdot\theta}\,d\theta.$$

Let $g(z) = z^{l+1}$ for $l \in \mathbb{Z}$. Clearly g satisfies that hypothesis in lemma 3.1. From lemma 3.2

$$\int_{S^1}\frac{g(\theta)}{\omega\cdot\theta}\,d\theta = 2\pi i\left[G(-\omega^{\perp}) - G(\omega^{\perp})\right] - \pi i\left[g(-\omega^{\perp}) - g(\omega^{\perp})\right].$$

Then

$$G(z) = \frac{1}{2\pi i}\int_{S^1} \frac{t^{l+1}}{t-z}\,dt = \begin{cases} Res_z(\frac{t^{l+1}}{t-z}) = z^{l+1} = g(z), & \text{if l} \geq \text{-1,} \\ Res_z(\frac{t^{l+1}}{t-z}) + Res_0(\frac{t^{l+1}}{t-z}) = z^{l+1} - z^{l+1} = 0, & \text{if l} \leq \text{-2.} \end{cases}$$

Thus $G(\omega^\perp) = \lim_{r\to 1^-} g(re^{i\psi}) = g(\omega^\perp)$ and $G(-\omega^\perp) = g(-\omega^\perp)$ for $l \geq -1$. Then

$$\int_{S^1} \frac{g(\theta)}{\omega\cdot\theta}\,d\theta = \begin{cases} \pi i[g(-\omega^\perp) - g(\omega^\perp)], & \text{l} \geq \text{-1,} \\ -\pi i[g(-\omega^\perp) - g(\omega^\perp)], & \text{l} \leq \text{-2.} \end{cases}$$

Then

$$\begin{aligned} \pi i[g(-\omega^\perp) - g(\omega^\perp)] &= \pi i[(-1)^{l+1}e^{i(l+1)\psi} - e^{i(l+1)\psi}] \\ &= -\pi i e^{il\psi}e^{i\psi}[(-1)^l + 1] \\ &= -\pi i e^{il\psi}\omega^\perp[(-1)^l + 1]. \end{aligned}$$

and thus

$$\int_{S^1} \frac{g(\theta)}{\omega\cdot\theta}\,d\theta = -\text{sgn(l)}\pi i e^{il\psi}\omega^\perp[(-1)^l + 1]$$

for $l \neq 0$. Then for $l \in \mathbb{Z}$

$$\begin{aligned} \int_{S^1} \frac{g(\theta)}{\omega\cdot\theta}\,d\theta = \pi i\left[-\omega^\perp(-1)^l e^{il\psi} - \omega^\perp e^{il\psi}\right] &= \pi(-i\omega^\perp)e^{il\psi}[(-1)^l + 1] \\ &= \begin{cases} 0 & \text{l odd,} \\ 2\pi\omega & \text{l} = 0, \\ -\text{sgn(l)}2\pi i e^{il\psi}\omega^\perp & \text{l even.} \end{cases} \end{aligned}$$

This concludes our proof. □

Lemma 3.5. *Let u and h be as in lemma 3.3. Then*

$$Re\int_{S^1} \frac{\theta}{(x-y)\cdot\theta} e^{u(y,\theta)-u(x,\theta)}d\theta = 2\pi\frac{x-y}{|x-y|^2}.$$

Proof. We again follow an outline in [7] for the proof of this lemma. Now observe that $e^{(u(y,\theta)-u(x,\theta))} = \cosh(u(y,\theta)-u(x,\theta)) + \sinh(u(y,\theta)-u(x,\theta))$. From lemma 3.3 we have $u(y,\theta)-u(x,\theta) = (G_{y,+}(-\theta^\perp) - G_{y,+}(\theta^\perp)) - (G_{x,+}(-\theta^\perp) - G_{x,+}(\theta^\perp))$.

Since e^z is an entire analytic function and $G_x(z)$ is analytic for $|z| < 1$,

$$e^{(G_y(-z)-G_y(z))-(G_x(-z)-G_x(z))}$$

is analytic for $|z| < 1$. Then $\exists u_l(x,y)$ such that

$$e^{(G_y(-z)-G_y(z))-(G_x(-z)-G_x(z))} = \sum_{l\geq 0} u_l(x,y)z^l$$

for $|z| < 1$. It follows that

$$e^{(G_y(-re^{i\phi})-G_y(re^{i\phi}))-(G_x(-re^{i\phi})-G_x(re^{i\phi}))} = \sum_{l\geq 0} u_l(x,y)r^l e^{il\phi}$$

for $0 \le r < 1$. Let $\theta^\perp = \begin{pmatrix}\cos\phi \\ \sin\phi\end{pmatrix}$. Since $e^z = \cosh(z) + \sinh(z)$ and $\cosh(z)$ is even,

$$\cosh[(G_y(-z) - G_y(z)) - (G_x(-z) - G_x(z))] = \sum_{l \ge 0 \text{ even}} u_l(x,y) z^l.$$

Now we will calculate the Fourier coefficients of $\cosh(u(y,\theta) - u(x,\theta))$. Then

$$\begin{aligned}
& \frac{1}{2\pi}\int_0^{2\pi} \cosh(u(y,\theta) - u(x,\theta)) e^{il\phi}\, d\phi \\
= \; & \frac{1}{2\pi}\int_0^{2\pi} \cosh[(G_{y,+}(-\theta^\perp) - G_{y,+}(\theta^\perp)) - (G_{x,+}(-\theta^\perp) - G_{x,+}(\theta^\perp))] e^{il\phi}\, d\phi \\
= \; & \lim_{r\to 1-} \frac{1}{2\pi}\int_0^{2\pi} \cosh[(G_y(-re^{i\varphi}) - G_y(re^{i\varphi})) - (G_x(-re^{i\varphi}) - G_x(re^{i\varphi}))] e^{il\phi}\, d\phi \\
= \; & \lim_{r\to 1-} u_l(x,y) r^l \\
= \; & u_l(x,y).
\end{aligned}$$

Since $G(z)$ is bounded for z inside the unit disk and $\cosh(z)$ is a continuous function, we may indeed exchange the limit and integration as we have done above. We also have that $u_0(x,y) = \cosh[(G_y(-0) - G_y(0)) - (G_x(-0) - G_x(0))] = \cosh(0) = 1$. Then

$$\begin{aligned}
\cosh(u(y,\theta) - u(x,\theta)) &= 1 + \sum_{l>0 \text{ even}} u_l(x,y) e^{il\varphi}, \\
\sinh(u(y,\theta) - u(x,\theta)) &= \sum_{l>0 \text{ odd}} u_l(x,y) e^{il\varphi}.
\end{aligned}$$

Since $a \in \mathcal{C}_0^\infty(\mathbb{R}^2)$, $u(y,\theta) - u(x,\theta)$ is an absolutely continuous function in θ and thus the Fourier series above converge pointwise.

Let

$$D \equiv \left\{ h \in L^2(S^1) : h(\theta) = \sum_{l=-n}^{n} a_l e^{il\varphi}, \quad n \in \mathbb{N} \right\}.$$

Consider $K_x : D \to D$ defined by

$$K_x(h) = \int_{S^1} \frac{\theta}{x \cdot \theta} h(\theta)\, d\theta,$$

where x is a parameter. Clearly K_x is linear and D is a dense subset of $L^2(S^1)$. From lemma 3.4 we see that $\|K_x\| \le 2\pi/|x|$. Since K_x is uniformly continuous and D is dense in $L^2(S^1)$, there exists a unique continuous extension of K_x to all of $L^2(S^1)$. Let $\overline{K}_x$ be this unique extension. By Parseval's Theorem, $L^2(S^1)$ is isomorphic to l^2. Thus we may uniquely describe any $h \in L^2(S^1)$ by its Fourier coefficients. Then we see that $\overline{K}_x(h)$ multiplies the Fourier coefficients of h by the

values given in lemma 3.4. Then using the continuity of K_{x-y} we have

$$\begin{aligned} & K_{x-y}(\cosh(u(y,\theta)-u(x,\theta))) \\ = \; & K_{x-y}\left(1+\lim_{n\to\infty}\sum_{l=0}^{n}u_{2l}(x,y)e^{2li\phi}\right) \\ = \; & 2\pi\frac{x-y}{|x-y|^2}+\sum_{l=0}^{\infty}u_{2l}(x,y)\frac{-2\pi i(x-y)^{\perp}}{|x-y|^2}e^{2li\psi} \\ = \; & 2\pi\frac{x-y}{|x-y|^2}-2\pi i\frac{(x-y)^{\perp}}{|x-y|^2}(\cosh(u(y,\omega)-u(x,\omega))-1) \end{aligned}$$

and

$$\begin{aligned} K_{x-y}(\sinh(u(y,\theta)-u(x,\theta))) &= K_{x-y}\left(\lim_{n\to\infty}\sum_{l=0}^{n}u_{2l+1}(x,y)e^{(2l+1)i\phi}\right) \\ &= \sum_{l=0}^{\infty}u_{2l+1}(x,y)0 \\ &= 0. \end{aligned}$$

Since $(x-y)\cdot\omega=0$, $x\cdot\omega=y\cdot\omega$ and thus $h(\omega,x\cdot\omega)=h(\omega,y\cdot\omega)$. Then

$$u(y,\omega)-u(x,\omega)=-(Da)(y,\omega^{\perp})+(Da)(x,\omega^{\perp})=-\int_y^x a\,ds$$

i.e. $u(y,\omega)-u(x,\omega)$ is real. Hence

$$Re\int_{S^1}\frac{\theta}{(x-y)\cdot\theta}e^{u(y,\theta)-u(x,\theta)}d\theta=2\pi\frac{x-y}{|x-y|^2}.$$

□

Lemma 3.6. *For $x\in\mathbb{R}^2$, we have*

$$div\frac{x}{|x|^2}=2\pi\delta(x),$$

where δ is the Dirac delta function.

Proof. Let $\mathrm{x}=\begin{bmatrix}x_1\\x_2\end{bmatrix}$. Then

$$\nabla(\log|x|)=\begin{bmatrix}\frac{x_1}{x_1^2+x_2^2}\\\frac{x_2}{x_1^2+x_2^2}\end{bmatrix}=\frac{x}{|x|^2}.$$

Hence $\nabla\cdot\left(\frac{x}{|x|^2}\right)=\nabla\cdot\nabla(\log|x|)=\Delta(\log|x|)$. Now we will show that

$$\int_{\mathbb{R}^2}\Delta\log|x|\,v(x)\,dx=2\pi v(0)$$

for $v\in\mathcal{C}_0^{\infty}(\mathbb{R}^2)$.

First we will switch to polar coordinates. Thus

$$\begin{aligned}\Delta v(r(x_1,x_2),\theta(x_1,x_2)) &= \frac{\partial}{\partial x_1}\left[v_r r_{x_1}+v_\theta\theta_{x_1}\right]+\frac{\partial}{\partial x_2}\left[v_r r_{x_2}+v_\theta\theta_{x_2}\right]\\ &= v_{rr}\left[r_{x_1}^2+r_{x_2}^2\right]+2v_{r\theta}\left[r_{x_1}\theta_{x_1}+r_{x_2}\theta_{x_2}\right]+v_{\theta\theta}\left[\theta_{x_1}^2+\theta_{x_2}^2\right]\\ &+ v_r\left[r_{x_1x_1}+r_{x_2x_2}\right]+v_\theta\left[\theta_{x_1x_1}+\theta_{x_2x_2}\right]\\ &= v_{rr}+\frac{1}{r}v_r+\frac{1}{r^2}v_{\theta\theta}.\end{aligned}$$

Then

$$\begin{aligned}\int_{\mathbb{R}^2}\Delta\log|x|\,v(x)\,dx &= \int_{\mathbb{R}^2}\log|x|\,\Delta v(x)\,dx\\ &= \int_0^{2\pi}\int_0^\infty r\log(r)\left[\frac{1}{r}v_r(r,\theta)+v_{rr}(r,\theta)+\frac{1}{r^2}v_{\theta\theta}(r,\theta)\right]dr\;d\theta\\ &= \int_0^{2\pi}\left[\int_0^\infty \log(r)v_r(r,\theta)\,dr+r\log(r)v_r(r,\theta)|_0^\infty\right.\\ &- \left.\int_0^\infty(\log(r)+1)v_r(r,\theta)\;dr\right]d\theta+\int_0^\infty\frac{1}{r}\log(r)\left[\int_0^{2\pi}v_{\theta\theta}(r,\theta)\;d\theta\right]dr\\ &= \int_0^{2\pi}\int_0^\infty -v_r(r,\theta)\,dr\;d\theta\\ &= \int_0^{2\pi}-v(r,\theta)|_0^\infty\;d\theta\\ &= \int_0^{2\pi}v(0,\theta)\;d\theta\\ &= 2\pi v(0).\end{aligned}$$

□

Theorem 3.7. *Let $g=R_af$ and h and u be as above. Then*

$$f(x)=\frac{1}{4\pi}Re\;div\int_{S^1}\theta e^{(Da)(x,\theta^\perp)}(e^{-h}He^hg)(\theta,x\cdot\theta)\,d\theta, \tag{12}$$

where H acts on the second argument.

Proof. This is the main result of this section and we follow an outline of the proof of this theorem developed in [7]. Then

$$(He^hg)(\theta,s)=\frac{1}{\pi}\int_{\mathbb{R}}\frac{e^{h(\theta,t)}}{s-t}\left[\int_{y\cdot\theta=t}e^{-(Da)(y,\theta^\perp)}f(y)\,dy\right]dt.$$

Now let $y=t\theta+r\theta^\perp$. Then we have that

$$\begin{aligned}(He^hg)(\theta,s) &= \frac{1}{\pi}\int_{\mathbb{R}}\frac{e^{h(\theta,t)}}{s-t}\left[\int_{\mathbb{R}}e^{-(Da)(t\theta+r\theta^\perp,\theta^\perp)}f(t\theta+r\theta^\perp)\,dr\right]dt\\ &= \frac{1}{\pi}\int_{\mathbb{R}^2}\frac{f(y)}{s-y\cdot\theta}e^{-(Da)(y,\theta^\perp)+h(\theta,y\cdot\theta)}dy\\ &= \frac{1}{\pi}\int_{\mathbb{R}^2}\frac{f(y)}{s-y\cdot\theta}e^{u(y,\theta)}dy.\end{aligned}$$

Therefore

$$\begin{aligned}
& \frac{1}{4\pi} Re \text{ div} \int_{S^1} \theta e^{(Da)(x,\theta^\perp)} (e^{-h} H e^h g)(\theta, x\cdot\theta)\, d\theta \\
= & \frac{1}{4\pi} Re \text{ div} \int_{S^1} \theta e^{(Da)(x,\theta^\perp)} e^{-h(\theta,x\cdot\theta)} \left[\frac{1}{\pi} \int_{\mathbb{R}^2} \frac{f(y)}{(x-y)\cdot\theta} e^{u(y,\theta)} dy \right] d\theta \\
= & \frac{1}{4\pi^2} Re \text{ div} \int_{S^1} \theta e^{-u(x,\theta)} \left[\int_{\mathbb{R}^2} \frac{f(y)}{(x-y)\cdot\theta} e^{u(y,\theta)} dy \right] d\theta \\
= & \frac{1}{4\pi^2} \text{div} \int_{\mathbb{R}^2} f(y) \left[Re \int_{S^1} \frac{\theta}{(x-y)\cdot\theta} e^{u(y,\theta)-u(x,\theta)} d\theta \right] dy \\
= & \frac{1}{4\pi^2} \text{div} \int_{\mathbb{R}^2} f(y)\, 2\pi \frac{x-y}{|x-y|^2}\, dy \\
= & \frac{1}{2\pi} \text{div} \int_{\mathbb{R}^2} f(y) \frac{x-y}{|x-y|^2}\, dy \\
= & \int_{\mathbb{R}^2} f(y)\delta(x-y)\, dy \\
= & f(x).
\end{aligned}$$

□

4 Implementation

We will implement the exact inversion formula (12) due to Natterer[7] for the parallel-beam geometry. For those interested in the fan-beam case, see [1]. The implementation will be divided into two parts. Let

$$g_a = Re\, e^{-h} H e^h g, \quad h = \frac{1}{2}(I + iH)Ra.$$

The first part will be the computation of this g_a. Then we will compute

$$f(x) = \frac{1}{4\pi} \text{div} \int_{S^1} \theta e^{(Da)(x,\theta^\perp)} g_a(\theta, x\cdot\theta)\, d\theta.$$

Notice that this last formula is very similar to the filtered backprojection and thus the reconstruction proceeds similarly. Now we will write out g_a explicitly. Let $h = h_1 + ih_2$, where $h_1 = \frac{1}{2}Ra$ and $h_2 = \frac{1}{2}H\,Ra$. Then

$$\begin{aligned}
Re\,[e^{-h} H e^h g] &= Re\,[e^{-h_1} e^{-ih_2} H e^{h_1} e^{ih_2} g] \\
&= Re\,[e^{-h_1}(\cos(h_2) - i\,\sin(h_2)) H e^{h_1} (\cos(h_2) + i\,\sin(h_2)) g] \\
&= e^{-h_1}[\cos(h_2) H e^{h_1} \cos(h_2) g + \sin(h_2) H e^{h_1} \sin(h_2) g].
\end{aligned}$$

We will compute the Hilbert transform by convolution. The formulation of this convolution is given below in the following propositions which apply to functions in the Schwartz space on $\mathbb{R}$. We shall denote the Schwartz space by $\mathcal{S}$. This is the linear space of smooth functions, v, such that $\sup_{x\in\mathbb{R}} |x^k D^l v(x)|$ is finite for all multi-indices $k, l \in \mathbb{N}$. The Schwartz space is used because the Fourier transform is an isomorphism of $\mathcal{S}$ onto itself.

Proposition 4.1. *If $v \in \mathcal{S}(\mathbb{R})$, then $\widehat{(Hv)}(\sigma) = \frac{sgn(\sigma)}{i}\widehat{v}(\sigma)$.*

For the proof, see [6], section VII.1.

Now let ϕ be a low pass filter, i.e. $\phi(\sigma) \approx 1$ for $|\sigma| \leq 1$ and $\phi(\sigma) \approx 0$ otherwise. Let $\widehat{(H_b v)}(\sigma) = \phi(\sigma/b)\widehat{(Hv)}(\sigma)$, where $b > 0$ is a bandwidth. Then if $\widehat{Hv}(\sigma)$ is essentially bandlimited with bandwidth b we have $\widehat{(H_b v)}(\sigma) \approx \widehat{(Hv)}(\sigma)$ and thus $(H_b v)(s) \approx (Hv)(s)$.

Proposition 4.2. *If*

$$k_b(s) = \frac{1}{2\pi}\int_{\mathbb{R}} \phi(\sigma/b)\frac{sgn(\sigma)}{i}e^{is\sigma}\, d\sigma,$$

then $H_b v = k_b * v$.

Proof. Then

$$\begin{aligned}
k_b * v(s) &= \int_{\mathbb{R}} k_b(s-t)v(t)\, dt \\
&= \frac{1}{2\pi}\int_{\mathbb{R}}\int_{\mathbb{R}} \phi(\sigma/b)\frac{sgn(\sigma)}{i}e^{i(s-t)\sigma}v(t)\, d\sigma\, dt \\
&= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \phi(\sigma/b)\frac{sgn(\sigma)}{i}e^{is\sigma}\left[\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} v(t)e^{-it\sigma}\, dt\right] d\sigma \\
&= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \phi(\sigma/b)\frac{sgn(\sigma)}{i}e^{is\sigma}\widehat{v}(\sigma)\, d\sigma \\
&= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \phi(\sigma/b)\widehat{(Hv)}(\sigma)e^{is\sigma}\, d\sigma \\
&= (H_b v)(s).
\end{aligned}$$

□

Proposition 4.3. *If ϕ is an ideal low pass filer, i.e. $\phi(s) = 1$ for $|s| \leq 1$ and $\phi(s) = 0$ otherwise, then*

$$k_b(s) = \frac{1-\cos(bs)}{\pi s}.$$

Proof. Then

$$\begin{aligned}
k_b(s) &= \frac{1}{2\pi}\int_{\mathbb{R}} \phi(\sigma/b)\frac{sgn(\sigma)}{i}e^{is\sigma}\, d\sigma \\
&= \frac{1}{2\pi}\int_{-b}^{b} \frac{sgn(\sigma)}{i}e^{is\sigma}\, d\sigma \\
&= \frac{1}{2\pi}\int_{-b}^{b} \frac{sgn(\sigma)}{i}(\cos(s\sigma) + i\,\sin(s\sigma))\, d\sigma \\
&= \frac{1}{2\pi}\int_{-b}^{b} sgn(\sigma)\sin(s\sigma)\, d\sigma,
\end{aligned}$$

since $sgn(\sigma)\cos(s\sigma)$ is odd. Then we have

$$\begin{aligned}\frac{1}{2\pi}\int_{-b}^{b} sgn(\sigma)\sin(s\sigma)\,d\sigma &= \frac{1}{2\pi}\left[\int_0^b \sin(s\sigma)\,d\sigma - \int_{-b}^{0}\sin(s\sigma)\,d\sigma\right] \\ &= \frac{1}{2\pi s}\left[-\cos(s\sigma)|_0^b + \cos(s\sigma)|_{-b}^0\right] \\ &= \frac{1-\cos(bs)}{\pi s}.\end{aligned}$$

□

Therefore we may use $k_b * v$ as an approximation for Hv in our calculation if $\widehat{Hv}(\sigma)$ is essentially bandlimited with bandwidth b. The bandwidth, b, is subject to the same restrictions as the bandwidth in the normal filtered backprojection algorithm. See [6] for details.

5 Numerical Results

I created two different elliptical test phantoms in MATLAB for my numerical experiments. The elliptical test phantoms are contained in a 35 cm by 35 cm square and their SPECT activities (in thousands) and attenuation coefficients (units cm^{-1}) are displayed in figures 3 and 5. Data was computed and used to produce images of 128 by 128 pixels using the parallel-beam geometry with 400 projections equispaced in 360 degrees with 128 detectors each. This agrees with the sampling conditions given in [6] for the filtered backprojection with the modification to 360 degree scanning. The MATLAB code for the inversion formula is a modification from Faridani's code in [2] for the computation of the standard filtered backprojection algorithm.

5.1 Inversion Formula

The first experiment (see figures 3 and 4) reproduces the numerical experiment in [7]. The same cross section of the reconstruction in [7] was sampled and displayed in figure 4. Results were almost identical.

The results of the second experiment, whose SPECT activities and attenuation map are displayed in figure 5, are displayed in figures 5, 6 and 7. Cross sections of the reconstruction are displayed in figure 6. The location of these cross sections are shown as horizontal lines in the top two frames of figure 5. These plots were used to sample the accuracy of the reconstruction. The step function displays the true activity distribution and the other curve is the reconstruction. I also included a plot of the reconstruction that used no attenuation correction (assumes attenuation map is zero) in the top left plot of figures 4 and 6. This is the bottom curve in both cases. Here, one can see the affect of attenuation on the data and the strength of accuracy of this new algorithm.

Oftentimes the attenuation map is quite different from the activity distribution. With this in mind, I created the phantoms for experiment two. In particular, the boundaries of the regions of varying attenuation and activity are different. As one can see, not only did the algorithm correct for attenuation, but it did rather well in detecting the boundaries of the attenuation map, so that they don't show up

in the reconstruction. Of course, the algorithm did not execute this flawlessly. If one looks closely, they can see a faint boundary from the attenuation map show up near the bottom of the phantom. Early SPECT reconstructions assumed that the attenuation map was either zero or constant. From this experiment one can see that an algorithm that does not take the attenuation into account can be grossly inaccurate.

I also ran the code with noise on the data. The same cross sections that were used to display results of the noiseless data were used here. For both experiments, I plotted results of adding 10% and 17% noise on the SPECT data and 10% noise on both the SPECT data and the attenuation data. These can be found in figures 4, 6, and 7. As one can see, the results were very good. The algorithm displays good stability and accuracy in the face of noisy data. Moreover, the attenuation correction is not affected by such noise for the affect of the attenuation does not show up in the reconstruction images (see figure 7).

5.2 EM Algorithm

Only the second numerical experiment was tested on the EM algorithm. Figures 8 and 10 show the reconstruction images and figures 9 and 11 show the cross section plots, where figures 10 and 11 have 10% noise added to the SPECT data. This algorithm doesn't seem to handle attenuation correction as well as the inversion formula. A few shadows from the attenuation can be seen towards to the bottom of the reconstruction images. One can definitely see that the smoothness of the solution deteriorates with the number of steps performed. In practice, the smoothness of the images can be improved slightly with a penalty term which is usually a Bayesian function.

6 Conclusions

It is difficult to compare the accuracy of these two methods with this numerical experiment, since they are both sensitive to the model. My programs have assumed perfect collimation, perfect decay rates, and no scatter. These modeling particulars will inherently improve the accuracy of the inversion formula, while not exploiting the power of the EM algorithm's statistical nature. I should also note that due to extremely long run times and complexity, the calculation of the matrix of probabilities for the EM algorithm is quite simplistic. Reconstructions using real hospital data would lead to a more informative and accurate comparison of these SPECT reconstruction algorithms. These modifications could be added to the implementation of the code given below.

Even after considering the circumstances given above, it seems to me that the inversion formula produces smoother, more accurate images in a much more timely manner. I believe that my numerical results show that the inversion formula shows promise in medical scanners, if work can be done to control the noise and account for scatter in the reconstruction. It is noteworthy to observe that the two algorithms performed equally well with noise added to the data (see figures 7 and 11). Not that the EM algorithm was more accurate with added noise, but that the algorithm seemed equally as stable.

Computation time had a noticeable difference in the two algorithms. Run times for the EM algorithm experiments were about forty times as long as the inversion formula. Most of the computation time was spent on the determination of the matrix of probabilities. The difference in memory usage was extreme because of the sparse, but very large matrix of probabilities used in the EM algorithm.

7 MATLAB Code

Below we include MATLAB code to reproduce the numerical experiments described above.

```
%%%%%%%% AttRad.m %%%%%%%%

function [RF] = AttRad(theta, phi, s, centerX, centerY, u, v, alpha, a, f)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Calculates attenuated Radon transform for phantom ellipses.
% First for loop gathers boundaries.  Second for loop sets a's, and f's.
% Third for loop carries out the calculation.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

phiP = phi + pi/2;
ts = [];
vals = [];
for mu=1:max(size(centerX))
    p1 = theta(1)*s - centerX(mu);
    p2 = theta(2)*s - centerY(mu);

    %one*t^2 + 2*two*t + three <= 0
    one = (cos(phiP-alpha(mu))/u(mu))^2 + (sin(phiP-alpha(mu))/v(mu))^2;
    temp1 = p1*cos(alpha(mu)) + p2*sin(alpha(mu));
    temp2 = -1*p1*sin(alpha(mu)) + p2*cos(alpha(mu));
    two = cos(phiP-alpha(mu))*temp1/u(mu)^2 + sin(phiP-alpha(mu))*temp2/v(mu)^2;
    three = (temp1/u(mu)).^2 + (temp2/v(mu)).^2 - 1;

    one = one*ones(size(two));

    vals = [vals one' two' three'];

    disc = two .* two - one .* three;
    ind = disc < 0;
    disc(ind) = 1; one(ind) = 1; two(ind) = 11;
    t0 = (-two - sqrt(disc))./one; t0 = t0';
    t1 = (-two + sqrt(disc))./one; t1 = t1';
    %Above assumes that the radius of roi is < 10

    ts = sort([ts t0 t1],2);
end

[m, n] = size(ts);
midPts = (ts(:,2:n) + ts(:,1:n-1))/2;

fs = zeros(m,n-1);
as = fs;
for mu=1:max(size(centerX))
    one = vals(:, 3*mu - 2); one = one(1);
    two = vals(:, 3*mu - 1);
    three = vals(:, 3*mu);

    temp = one * midPts(:,1:n-1).^2 + 2 * (two*ones(1,n-1)) .* midPts(:,1:n-1);
    ind = temp + (three*ones(1,n-1)) <= 0;
    fs = fs +  f(mu)*ind;
```

```
    as = as + a(mu)*ind;
end
vals = [];

tj_m_tjm1 = ts(:,2:n) - ts(:,1:n-1);
temp = as .* tj_m_tjm1;
temp = temp(:,2:n-1);
temp = fliplr(temp);
topSum = cumsum(temp,2);
topSum = fliplr(topSum);

RF = zeros(size(s));
for mu=1:n-1
    if mu < n-1
        temp1 = fs(:,mu).*exp(-1*topSum(:,mu));
    else
        temp1 = fs(:,mu);
    end

    oneA = as(:,mu);
    oneT = tj_m_tjm1(:,mu);

    ind = oneA == 0;
    indNot = ~ind;
    temp2 = zeros(size(temp1));
    temp2(ind) = oneT(ind);
    temp2(indNot) = (1 - exp(-1*oneA(indNot) .* oneT(indNot)))./oneA(indNot);

    RF = RF + (temp1.*temp2)';
end


%%%%%%%% ConvKernel.m %%%%%%%%

function u = ConvKernel(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Calculates k_b(s) = (1-cos(bs))(pi*s)
% for the SPECT inversion formula ie (Hg)(s) = k_b*g(s).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u = zeros(size(t));
il = abs(t) <= 1.e-6;
t1 = t(abs(t) > 1.e-6);
v = (1-cos(t1))./t1;
u(abs(t) > 1.e-6) = v;
u = u / pi;


%%%%%%%% DivBeam.m %%%%%%%%

function [RF] = DivBeam(phi, x1, x2, centerX, centerY, u, v, alpha, a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Computes the Divergent Beam transform of ellipses.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = max(size(x1));
RF = zeros(1, m);
phiP = phi;
for mu=1:max(size(centerX))
    p1 = x1 - centerX(mu);
    p2 = x2 - centerY(mu);
    p1 = p1';
    p2 = p2';

    %one*t^2 + 2*two*t + three <= 0
    one = (cos(phiP-alpha(mu))/u(mu))^2 + (sin(phiP-alpha(mu))/v(mu))^2;
    temp1 = p1*cos(alpha(mu)) + p2*sin(alpha(mu));
    temp2 = -1*p1*sin(alpha(mu)) + p2*cos(alpha(mu));
```

```
    two = cos(phiP-alpha(mu))*temp1/u(mu)^2 + sin(phiP-alpha(mu))*temp2/v(mu)^2;
    three = (temp1/u(mu)).^2 + (temp2/v(mu)).^2 - 1;

    val1 = zeros(size(RF));
    val2 = val1; val = val1;
    disc = two .* two - one .* three;
    ind = disc > 0;
    sqDisc = sqrt(disc(ind));

    val1(ind) = (-two(ind) - sqDisc)./one;
    val2(ind) = (-two(ind) + sqDisc)./one;
    val(ind) = max(val2(ind),0) - max(val1(ind), 0);

    RF(ind) = RF(ind) + a(mu)*val(ind);
end


%%%%%%%% ExpectMax.m %%%%%%%%

function ExpectMax(p,q,n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Main file for EM algorithm experiment.
% p = number of projections
% q = number of detectors
% n = number of iterations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('EM Algorithm');

if nargin < 3, n = 32; end
if nargin < 2, p = 200; q = 64; end

% Set phantom and window ranges
phanNum = 2;
crossSections = [-.31 floor(q/(64/20)+.5)+q; -.55 floor(q/(64/35)+.5)+q];
Ptrue = trueActivity(phanNum,2*q);
val = crossSections(phanNum,2);
x = [-q:q-1]/q;
[windowLeft,i] = min(x+.75);
[windowRight,i] = min(x-.75);
windowLeft = windowLeft - .5;
windowRight = -windowRight - 1;
ytrue = Ptrue(val,1:2*q);

% General Set up
MX = 2*q;  MY = 2*q;
circle = 1;

pmin = -0.05;
pmax = 1;

[retVal, roi] = phantom(phanNum);
centerX = retVal(1,:);
centerY = retVal(2,:);
axisA = retVal(3,:);
axisB = retVal(4,:);
alpha = retVal(5,:);
f = retVal(6,:);
a = retVal(7,:);
alpha = alpha*pi/180;

if MX > 1
    hx = (roi(2) - roi(1))/MX;
    xrange = roi(1) + hx*[0:MX-1] + hx/2;
else
    return;
end
if MY > 1
    hy = (roi(4) - roi(3))/MY;
```

```
    yrange = flipud((roi(3) + hy*[0:MY-1] +hy/2)');
else
    return;
end

center = [(roi(1) + roi(2)), (roi(3)+roi(4))]/2;
x1 = ones(MY,1)*xrange;  %x-corrdinate matrix
x2 = yrange*ones(1,MX);  %y-coordinate matrix
if circle == 1
    re = min([roi(2)-roi(1), roi(4)-roi(3)])/2;
    chi = ((x1-center(1)).^2 + (x2-center(2)).^2 <= re^2);  %char. fcn of roi;
else
    chi = isfinite(x1);
end

x1 = x1(chi);  x2 = x2(chi);
newQ = max(size(x1));

h = 1/q;
s = h*[-q:q-1];
shiftS = (s + 1/(2*q))';
g = zeros(p*q,1);
xj = x1'; yj = x2';
A = sparse(2*p*q,newQ);

disp('Building Matrix of Probabilities');
for i=1:p
    % set A and g
    if mod(i,10) == 0
        i
    end
    phi = (2*pi*(i-1)/p);
    theta = [cos(phi); sin(phi)];

    %detector coordinates
    xdi = -shiftS*theta(2) + center(1);
    ydi = shiftS*theta(1) + center(2);

    tempx = repmat(xdi,1,max(size(xj))) - repmat(xj,max(size(xdi)),1);
    tempy = repmat(ydi,1,max(size(yj))) - repmat(yj,max(size(ydi)),1);

    disc = (1/(2*q))^2 - (tempx*theta(2)-tempy*theta(1)).^2;
    ind = disc >= 0;
    Ai = sparse(2*q,newQ);
    Ai(ind) = 1;

    scaler = DivBeam(phi+pi/2, x2, flipud(x1), centerX, centerY, axisA, axisB, alpha, a);
    scaler = exp(-scaler);
    Ai = repmat(scaler, 2*q, 1) .* Ai;

    g((i-1)*2*q+1:i*2*q) = AttRad(theta, phi, s, centerX, centerY, axisA, axisB, alpha, a, f);
    A((i-1)*2*q+1:i*2*q,:) = Ai;
    clear Ai;
    clear tempx;
    clear tempy;
end

% add noise
g = g + g .* (.1 * 2*(rand(size(g)) - .5));

colSums = sum(A,1);
for i=1:newQ
    A(:,i) = A(:,i) ./ colSums(i);
end

ks = [4 8 16 32];
counter = 1;

% Original EM
fk = 2*ones(newQ,1);
```

```
disp('Computing Iterates');
for k=1:n
    fk = fk .* (A'*(g ./ (A*fk)));
    flag = find(ks-k == 0);
    [qw,as] = size(flag);
    if as > 0
        P = zeros(MY,MX);
        P(chi) = (fk ./ colSums') ./ hx;
        P = P';
        P = fliplr(P);
        [k min(min(P)) max(max(P))]
        figure(1);
        subplot(2,2,counter), window3(pmin, pmax, roi, P);
        axis off;

        yapprox = P(val,1:2*q);
        figure(2);
        subplot(2,2,counter), plot(x,ytrue,'k',x,yapprox,'k');
        temp = axis;
        temp(1) = windowLeft; temp(2) = windowRight;
        if phanNum == 1
            temp(3) = -0.1; temp(4) = 1.1;
        else
            temp(3) = -0.15; temp(4) = 1.05;
        end
        axis(temp);
        axis('square');
        P = 0;
        counter = counter + 1;
    end
end


%%%%%%%% fbp.m %%%%%%%%

function P=fbp(p,q,phanNum,job,noise)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Adel Faridani with
% very small slight modifications by Kyle Champley
% Description: Computes Parallel-beam filtered backprojection
% algorithm for the standard lattice.
% Last revision: Aug 29, 2001
%
% job 1: original calculation
% job 2: SPECT true activity
% job 3: SPECT w/ no attenuation correction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if nargin < 5, noise = 0; end
if nargin < 4, job = 1; end
if nargin < 3, phanNum = 1; end
if nargin < 2, p = 200; q = 64; end

%specify input parameters here

%p = 200; %number of view angles between 0 and pi
%q = 64; %q=1/d, d = detector spacing
MX = 128;  MY = 128; %matrix dimensions
%roi = [-1 1 -1 1]; %roi=[xmin xmax ymin ymax]
                    %region of interest where
                    %reconstruction is computed
circle = 1; % If circle = 1 image computed only inside
            % circle inscribed in roi

[retVal, roi] = phantom(phanNum);

centerX = retVal(1,:);
centerY = retVal(2,:);
axisA = retVal(3,:);
axisB = retVal(4,:);
```

```
alpha = retVal(5,:);
f = retVal(6,:);
a = retVal(7,:);

b = pi*q;
rps = 1/b;
alpha = alpha*pi/180;

if MX > 1
    hx = (roi(2) - roi(1))/(MX-1);
    xrange = roi(1) + hx*[0:MX-1];
else
    return;
end

if MY > 1
    hy = (roi(4) - roi(3))/(MY-1);
    yrange = flipud((roi(3) + hy*[0:MY-1])');
else
    return;
end

center = [(roi(1) + roi(2)), (roi(3)+roi(4))]/2;
x1 = ones(MY,1)*xrange;  %x-corrdinate matrix
x2 = yrange*ones(1,MX);  %y-coordinate matrix
if circle == 1
    re = min([roi(2)-roi(1), roi(4)-roi(3)])/2;
    chi = ((x1-center(1)).^2 + (x2-center(2)).^2 <= re^2);  %char. fcn of roi;
else
    chi = isfinite(x1);
end

x1 = x1(chi);  x2 = x2(chi);
P = zeros(MY,MX);  Pchi = P(chi);

h = 1/q;
s = h*[-q:q-1];
bs = [-2*q:2*q-1]/(q*rps);
wb = slkernel(bs)/(rps^2);  %compute discrete convolution kernel.


for j=1:p
    phi = (pi*(j-1)/p);
    theta = [cos(phi); sin(phi)];
    if job == 3
        RF = AttRad(theta, phi, s, centerX, centerY, axisA, axisB, alpha, a, f);
    else
        RF = Rad(theta, phi, s, centerX, centerY, axisA, axisB, alpha, f);
    end
    if noise > 0
        RF = RF + RF .* (noise * 2*(rand(size(RF)) - .5));
    end

%   Convolution
    C = conv(RF,wb);
    Q = h*C(2*q+1:4*q);  Q(2*q+1) = 0;

%   Interpolation and backprojection
    Q = [real(Q)'; 0];
    t = theta(1)*x1 + theta(2)*x2;
    k1 = floor(t/h);
    u = (t/h-k1);
    k = max(1,k1+q+1);  k = min(k,2*q);
    Pupdate = ((1-u).*Q(k)+u.*Q(k+1));
    Pchi = Pchi + Pupdate;
end
P(chi) = Pchi*(2*pi/p);


%%%%%%%% phantom.m %%%%%%%%
```

```
function [retVal, roi]=phantom(job)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Sets parameters for phantom ellipses.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if job == 1
    % Natterer's SPECT Phantom
    centerX = [0 -.45 .45 .1];
    centerY = [0 0 0 -.4];
    axisA = [.96 .25 .27 .13];
    axisB = [.80 .35 .27 .13];
    alpha = [0 0 0 0];
    f = [100 -100 -100 900];
    a = [0.1 -0.1 -0.1 0];
    f = f / 1000;
    a = a * 100 * (.5 *.35);

    roi = [-1 1 -1 1];
end
if job == 2
    % Phantom for experiment 2.
    centerX = [0 0 -.1 .2 -.2 .53 -.08 .08];
    centerY = [0 0 .4 .1 -.1 -.5 -.55 -.55];
    axisA = [.8 .75 .3 .15 .12 .05 .06 .03];
    axisB = [.95 .9 .3 .5 .12 .2 .03 .06];
    alpha = [0 0 0 -20 0 -29 0 0];
    f = [0 0.5 0.5 -0.35 0.25 0.25 -0.15 0.4];
    a = [0.1 -0.075 0 0 0 0 0 0];

    centerX = [centerX 0 -.25 .27 -.35];
    centerY = [centerY -.12 -.54 -.54 .2];
    axisA = [axisA .12 .25 .27 .2];
    axisB = [axisB .5 .05 .05 .3];
    alpha = [alpha 0 0 0 -20];
    f = [f 0 0 0 0];
    a = [a 0.0350 0.0350 0.0350 0.0250];
    a = a * 100 * (.5 * .35);

    roi = [-1 1 -1 1];
end

retVal = [centerX; centerY; axisA; axisB; alpha; f; a];


%%%%%%%% Rad.m %%%%%%%%

function [RF] = Rad(theta, phi, s, x, y, u, v, alpha, rho)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Adel Faridani
% Description: This function computes the Radon transform of
% ellipses centered at (x,y) with major axis u, minor axis v,
% rotated through angle alpha, with weight rho.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

RF = zeros(size(s));

for mu=1:max(size(x))
    a = (u(mu)*cos(phi-alpha(mu)))^2+(v(mu)*sin(phi-alpha(mu)))^2;
    test = a-(s-[x(mu); y(mu)]'*theta).^2;
    ind = test > 0;
    RF(ind) = RF(ind) + rho(mu)*(2*u(mu)*v(mu)*sqrt(test(ind)))/a;
end


%%%%%%%% slkernel.m %%%%%%%%

function u = slkernel(t)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Adel Faridani
% Description: Calculate Shepp-Logan kernel
% for the filtered backprojection algorithm.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u = zeros(size(t));
i1 = abs(abs(t) - pi/2) <= 1.e-6;
u(i1) = ones(size(u(i1)))/pi;
t1 = t(abs(abs(t) - pi/2) > 1.e-6);
v = (pi/2 - t1.*sin(t1))./((pi/2)^2 - t1.^2);
u(abs(abs(t) - pi/2) > 1.e-6) = v;
u = u / (2*pi^3);


%%%%%%%%% spect.m %%%%%%%%%

function P=spect(p, q, phanNum, noise, both)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:        Kyle Champley
% Description:   Calculates SPECT image.
% This code was modified from Adel Faridani's code to calculate the
% Parallel-beam filtered backprojection algorithm for the standard lattice.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if nargin < 2, p = 200; q = 64; end
if nargin < 3, phanNum = 1; end
if nargin < 4, noise = 0; end
if nargin < 5, both = 0; end

if noise >= 1
    disp('Please enter 0 <= noise < 1');
    return;
end

%p = 400; %number of view angles between 0 and 2pi
%q = 64; %q=1/d, d = detector spacing
MX = 128;  MY = 128;
circle = 1;

[retVal roi] = phantom(phanNum);

% Phantom Ellipses
centerX = retVal(1,:);
centerY = retVal(2,:);
axisA = retVal(3,:);
axisB = retVal(4,:);
alpha = retVal(5,:);
f = retVal(6,:);
a = retVal(7,:);

b = pi*q;  rps = 1/b;
alpha = alpha*pi/180;

hx = (roi(2) - roi(1))/(MX-1);
xrange = roi(1) + hx*[0:MX-1];

hy = (roi(4) - roi(3))/(MY-1);
yrange = flipud((roi(3) + hy*[0:MY-1])');

center = [(roi(1) + roi(2)), (roi(3)+roi(4))]/2;
x1 = ones(MY,1)*xrange;
x2 = yrange*ones(1,MX);
if circle == 1
    re = min([roi(2)-roi(1), roi(4)-roi(3)])/2;
    chi = ((x1-center(1)).^2 + (x2-center(2)).^2 <= re^2);
else
    chi = isfinite(x1);
end
```

```
x1 = x1(chi);  x2 = x2(chi);
P1 = zeros(MY,MX);  P1chi = P1(chi);
P2 = zeros(MY,MX);  P2chi = P2(chi);
h = 1/q;
s = h*[-q:q-1];

bs = b*[-2*q:2*q-1]/q;
wb = ConvKernel(bs)*b;

deltaTheta = 2*pi/p;

for j=1:p
    if mod(j,50) == 0
        j
    end
    phi = (2*pi*(j-1)/p);
    theta = [cos(phi); sin(phi)];
    Ra = Rad(theta, phi, s, centerX, centerY, axisA, axisB, alpha, a);

    % Compute g(theta, theta*x), here s = theta*x
    g = AttRad(theta, phi, s, centerX, centerY, axisA, axisB, alpha, a, f);
    if noise > 0
        g = g + g .* (noise * 2*(rand(size(g)) - .5));
    end

    % Compute ga(theta, theta*x)
    C = conv(Ra, wb);
    vRa = h*C(2*q+1:4*q);

    cos_vRa = cos(1/2*vRa);
    sin_vRa = sin(1/2*vRa);

    C = conv(exp(1/2*Ra).*cos_vRa.*g, wb);
    part1 = h*C(2*q+1:4*q);
    part1 = cos_vRa .* part1;

    C = conv(exp(1/2*Ra).*sin_vRa.*g, wb);
    part2 = h*C(2*q+1:4*q);
    part2 = sin_vRa .* part2;

    ga = exp(-1/2*Ra) .* (part1 + part2);
    %end compute ga

    % Caculate exp((Da)(x,thetaPerp))
    Da = DivBeam(phi+pi/2, x1, x2, centerX, centerY, axisA, axisB, alpha, a);
    if both == 1 & noise > 0
        Da = Da + Da .* (noise * 2*(rand(size(Da)) - .5));
    end
    eDa = (exp(Da))';

    % Interpolation and backprojection
    ga = [real(ga)'; 0; 0];
    t = theta(1)*x1 + theta(2)*x2;
    k1 = floor(t/h);
    u = (t/h-k1);
    k = max(1,k1+q+1);  k = min(k,2*q);
    Pupdate = eDa.*((1-u).*ga(k)+u.*ga(k+1));
    P1chi = P1chi + theta(1)*Pupdate;
    P2chi = P2chi + theta(2)*Pupdate;
end

P1(chi) = P1chi*deltaTheta;
P2(chi) = P2chi*deltaTheta;

%Calculate Divergence here
x1 = ones(MY,1)*xrange;
x2 = yrange*ones(1,MX);
P = divergence(x1, x2, P1, P2);

P = P/(4*pi);
```

```
% Clear junk for |x|>1 created
% from divergence calculation.
% Assumes there is nothing for
% x within 0.01 of the boundary.
% Commenting this out doesn't change much
re = min([roi(2)-roi(1), roi(4)-roi(3)])/2 - .01;
chi = ((x1-center(1)).^2 + (x2-center(2)).^2 > re^2);
P(chi) = 0;


%%%%%%%% spectMain.m %%%%%%%%

function spectMain(part, phanNum)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Performs numerical experiments on the
% exact inversion formula.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if nargin < 2, phanNum = 1; end

disp('Research Folder');

[retVal, roi] = phantom(phanNum);

p = 400;
q = 64;

crossSections = [-.31 84; -.55 99];

if part == 1
    centerX = retVal(1,:);
    centerY = retVal(2,:);
    axisA = retVal(3,:);
    axisB = retVal(4,:);
    alpha = retVal(5,:);
    f = retVal(6,:);
    a = retVal(7,:);
    alpha = alpha*pi/180;

    for i=1:max(size(a))
        x1 = [-1:.01:1];
        x1 = axisA(i)*x1;
        x2 = x1;
        y1 = axisB(i)/axisA(i)*sqrt(axisA(i)^2-x1.^2);
        y2 = -y1;

        temp1 = x1*cos(alpha(i)) - y1*sin(alpha(i));
        temp2 = x1*sin(alpha(i)) + y1*cos(alpha(i));
        x1 = temp1 + centerX(i);
        y1 = temp2 + centerY(i);

        temp1 = x2*cos(alpha(i)) - y2*sin(alpha(i));
        temp2 = x2*sin(alpha(i)) + y2*cos(alpha(i));
        x2 = temp1 + centerX(i);
        y2 = temp2 + centerY(i);

        if f(i) ~= 0
            subplot(2,2,1), plot(x1, y1, 'b', x2, y2, 'b');
            axis(roi);
        else
            subplot(2,2,1), plot(0,0,'w');
        end
        %TITLE('SPECT Activity');
        axis('square');
        if (i == 1)
            hold on;
        end
        if a(i) ~= 0
```

```
            subplot(2,2,2), plot(x1, y1, 'b', x2, y2, 'b');
            axis(roi);
        else
            subplot(2,2,2), plot(0,0,'w');
        end
        %TITLE('Attenuation Coefficients');
        axis('square');
        if (i == 1)
            hold on;
        end
    end

    x3 = [-2/3:.01:2/3];
    y3 = crossSections(phanNum,1)*ones(size(x3));
    subplot(2,2,1), plot(x3,y3,'r');
    subplot(2,2,2), plot(x3,y3,'r');

    P2 = spect(p,q,phanNum);
    P1 = trueActivity(phanNum);

    pmin = min(min(P1) - 0.025);
    pmax = max(max(P1));
    if phanNum == 1, pmax = 0.5; end
    if phanNum == 2, pmin = -0.05; end

    subplot(2,2,3), window3(pmin, pmax, roi, P1);
    axis off;
    %TITLE('True Activity');
    subplot(2,2,4), window3(pmin, pmax, roi, P2);
    axis off;
    %TITLE('Reconstruction');
end
if part == 2
    % Cross-Section Plots
    P1 = trueActivity(phanNum,128);
    P2a = spect(p,q,phanNum);
    P2b = spect(p,q,phanNum,.1);
    P2c = spect(p,q,phanNum,.17);
    P2d = spect(p,q,phanNum,.1,1);
    P3a = fbp(p/2,q,phanNum,3);

    val = crossSections(phanNum,2);
    x = [-48:48]/64;
    y1 = P1(val,16:128-16);
    y2a = P2a(val,16:128-16);
    y2b = P2b(val,16:128-16);
    y2c = P2c(val,16:128-16);
    y2d = P2d(val,16:128-16);
    y3a = P3a(val,16:128-16);

    figure(1);
    %clrs = ['r' 'b' 'g'];
    clrs = ['k' 'k' 'k'];
    subplot(2,2,1), plot(x,y1,clrs(1),x,y2a,clrs(2),x,y3a,clrs(3));
    %TITLE('Cross Section with No Noise');
    temp = axis;
    temp(1) = x(1); temp(2) = x(length(x));
    if phanNum == 1
        temp(3) = -0.1; temp(4) = 1.1;
    else
        temp(3) = -0.15; temp(4) = 1.05;
    end
    axis(temp);
    axis('square');
    subplot(2,2,2), plot(x,y1,clrs(1),x,y2b,clrs(2));
    %TITLE('Cross Section with 10% Noise');
    temp = axis;
    temp(1) = x(1); temp(2) = x(length(x));
    if phanNum == 1
        temp(3) = -0.1; temp(4) = 1.1;
```

```
    else
        temp(3) = -0.15; temp(4) = 1.05;
    end
    axis(temp);
    axis('square');
    subplot(2,2,3), plot(x,y1,clrs(1),x,y2c,clrs(2));
    %TITLE('Cross Section with 17% Noise');
    temp = axis;
    temp(1) = x(1); temp(2) = x(length(x));
    if phanNum == 1
        temp(3) = -0.1; temp(4) = 1.1;
    else
        temp(3) = -0.15; temp(4) = 1.05;
    end
    axis(temp);
    axis('square');
    subplot(2,2,4), plot(x,y1,clrs(1),x,y2d,clrs(2));
    %TITLE('Cross Section with 10% Noise on Both Scans');
    temp = axis;
    temp(1) = x(1); temp(2) = x(length(x));
    if phanNum == 1
        temp(3) = -0.1; temp(4) = 1.1;
    else
        temp(3) = -0.15; temp(4) = 1.05;
    end
    axis(temp);
    axis('square');


    % Noise Pics (figure 5)
    pmin = min(min(P1) - 0.025);
    pmax = max(max(P1));
    if phanNum == 1, pmax = 0.5; end
    if phanNum == 2, pmin = -0.05; end

    figure(2);
    subplot(2,2,1),window3(pmin, pmax, roi, P3a);
    axis off;
    subplot(2,2,2),window3(pmin, pmax, roi, P2b);
    axis off;
    subplot(2,2,3),window3(pmin, pmax, roi, P2c);
    axis off;
    subplot(2,2,4),window3(pmin, pmax, roi, P2d);
    axis off;
end


%%%%%%%% trueActivity.m %%%%%%%%

function P=trueActivity(phanNum,numPixels)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Kyle Champley
% Description: Calculates exact image.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


if nargin < 2, numPixels = 256; end
MX = numPixels;  MY = numPixels; %matrix dimensions
%roi = [-1 1 -1 1]; %roi=[xmin xmax ymin ymax]

[retVal, roi] = phantom(phanNum);

if MX > 1
    hx = (roi(2) - roi(1))/(MX-1);
    xrange = roi(1) + hx*[0:MX-1];
else
    return;
end

if MY > 1
```

```
    hy = (roi(4) - roi(3))/(MY-1);
    yrange = flipud((roi(3) + hy*[0:MY-1])');
else
    return;
end

centerX = retVal(1,:);
centerY = retVal(2,:);
axisA = retVal(3,:);
axisB = retVal(4,:);
alpha = retVal(5,:);
f = retVal(6,:);
a = retVal(7,:);
alpha = alpha*pi/180;

center = [(roi(1) + roi(2)), (roi(3)+roi(4))]/2;
x1 = ones(MY,1)*xrange;  %x-corrdinate matrix
x2 = yrange*ones(1,MX);  %y-coordinate matrix
chi = isfinite(x1);
x1 = x1(chi);  x2 = x2(chi);
P = zeros(MY,MX);  Pchi = P(chi);

for mu=1:max(size(centerX))
    p1 = x1 - centerX(mu);
    p2 = x2 - centerY(mu);
    p1 = p1'; p2 = p2';
    A = [cos(alpha(mu))/axisA(mu) sin(alpha(mu))/axisA(mu);
         -sin(alpha(mu))/axisB(mu) cos(alpha(mu))/axisB(mu)];
    temp = [A(1,1)*p1 + A(1,2)*p2; A(2,1)*p1 + A(2,2)*p2];
    ind = temp(1,:).^2+temp(2,:).^2 <= 1;
    ind = ind';
    update = zeros(size(P));
    update(ind) = f(mu);
    Pchi(ind) = Pchi(ind) + update(ind);
end
P(chi) = Pchi;


%%%%%%%% window3.m %%%%%%%%

function pic1 = window3(mi, ma, roi, pic);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Adel Faridani
% Description: Displays image pic with coordinates given by
% roi = [xmin xmax ymin ymax].
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = [roi(1), roi(2)];  y = [roi(3), roi(4)];
colors = 128;  co = colors - 1;
pic1 = pic - mi*ones(size(pic));
pic1 = (co/(ma-mi))*pic1;
P = (pic1 >= 0);
pic1 = pic1.*P;
P = (pic1 <= co);
pic1 = pic1.*P + co*(ones(size(pic1)) - P);
colormap(gray(colors));
image(x,fliplr(y), flipud(pic1));
axis('square');
```

References

[1] Bukhgeim A and Kazantsev S, *Inversion formula for the fan-beam attenuated radon transform in a unit disk*, Sobolev Institute of Math (2002), 3–33.

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

[3] Iusem A, *Convergence analysis for a multiplicity relaxed em algorithm*, Math Meth Appl Sci **14** (1991), 573–593.

[4] Finch D, *The attenuated x-ray transform: Recent developments*, Inside Out: Inverse Problems and Applications (2003), 47–66.

[5] Williams D, *Weighing the odds: A course in probability and statistics*, Cambridge University Press, 2001.

[6] Natterer F, *The mathematics of computerized tomography*, John Wiley & Sons, Inc., 1986.

[7] ———, *Inversion of the attenuated radon transform*, Inverse Problems **17** (2001), 113–119.

[8] Natterer F and Wübbeling F, *Mathematical methods in image reconstruction*, Society for Industrial & Applied Mathematics, 2001.

[9] Muskhelishvili N, *Singular integral equations*, P. Noordhoff N.V., Groningen, Holland, 1953.

[10] Lewitt R and Matej S, *Overview of methods for image reconstruction from projections in emission computed tomography*, Proceedings of the IEEE **10** (2003), 1588–1609.

[11] Cover T, *An algorithm for maximizing expected log investment return*, IEEE Trans on Information Theory **2** (1984), 369–373.

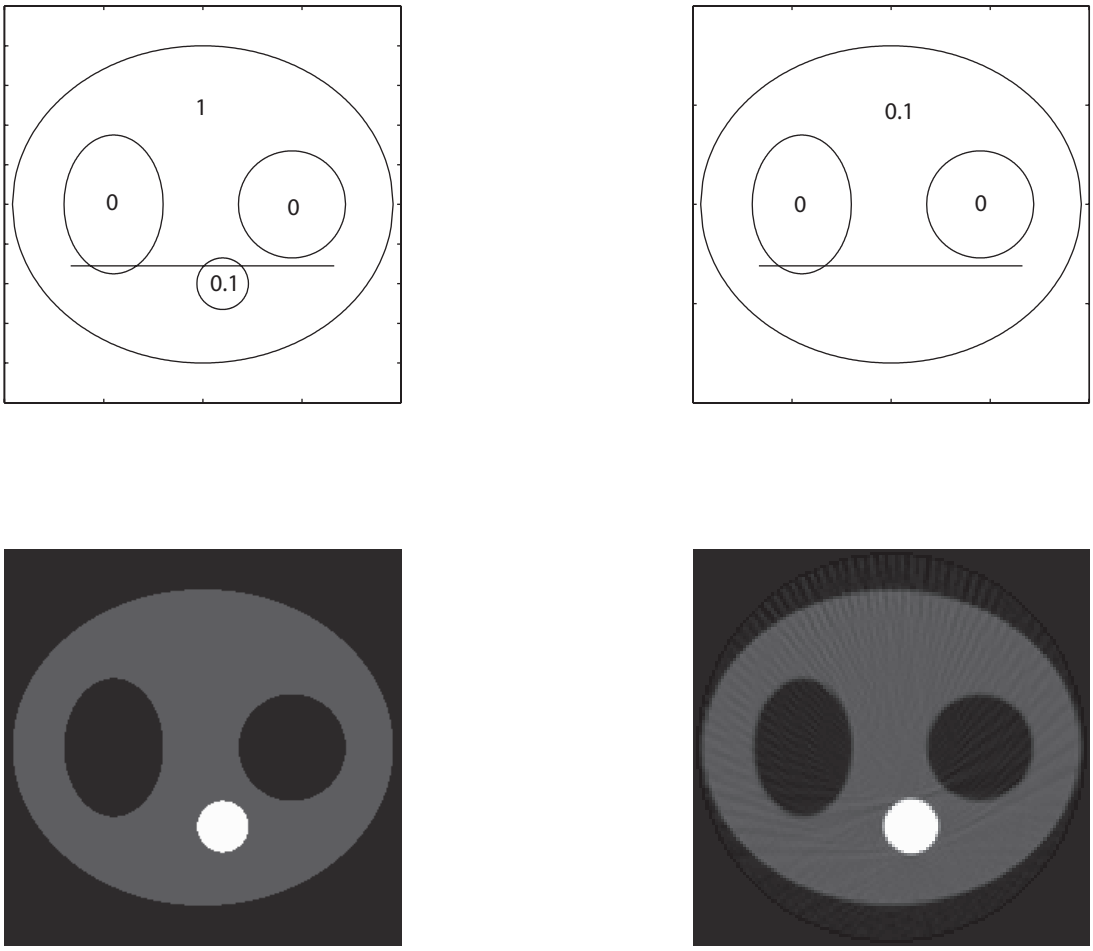

Figure 3: Inversion formula reconstructions for experiment 1. (top left): SPECT Map, (top right): Attenuation Map, (bottom left): True Activity, (bottom right): Reconstruction

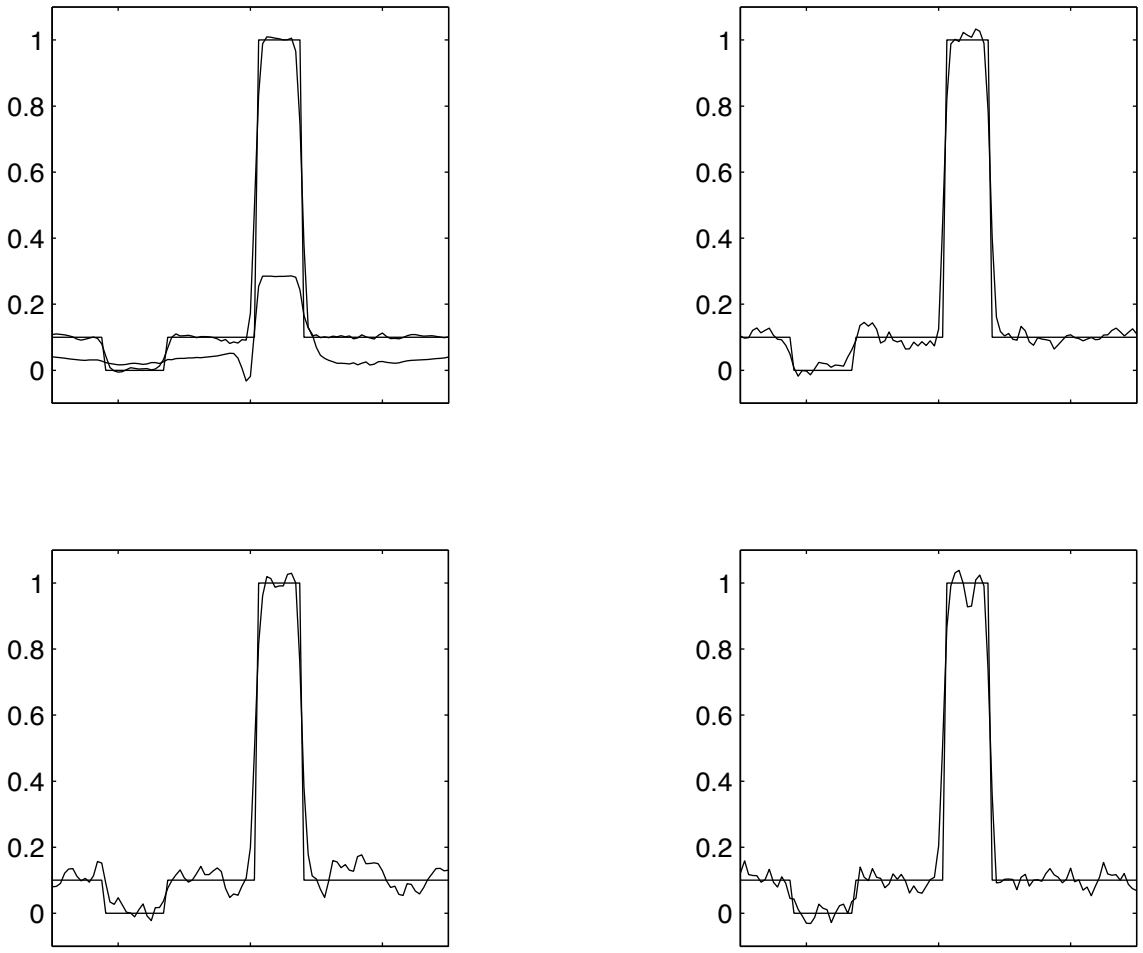

Figure 4: Cross section plots for inversion formula, experiment 1. (top left): No noise added to the data. The bottom line plots the cross section of the reconstruction that does not use attenuation correction. (top right): 10% noise added to the SPECT activity. (bottom left): 17% noise added to the SPECT data. (bottom right): 10% noise added to both the SPECT activity and the attenuation map.

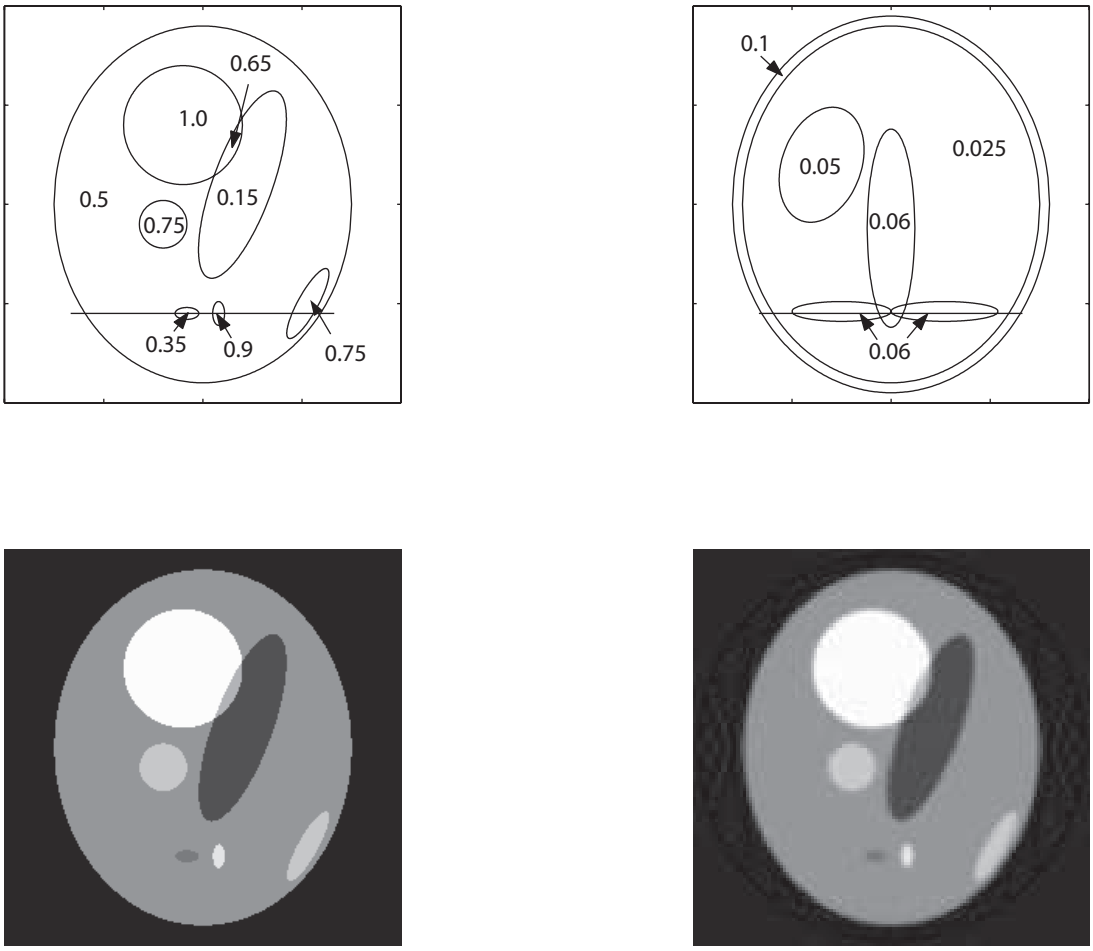

Figure 5: Inversion formula reconstructions for experiment 2. (top left): SPECT Map, (top right): Attenuation Map, (bottom left): True Activity, (bottom right): Reconstruction

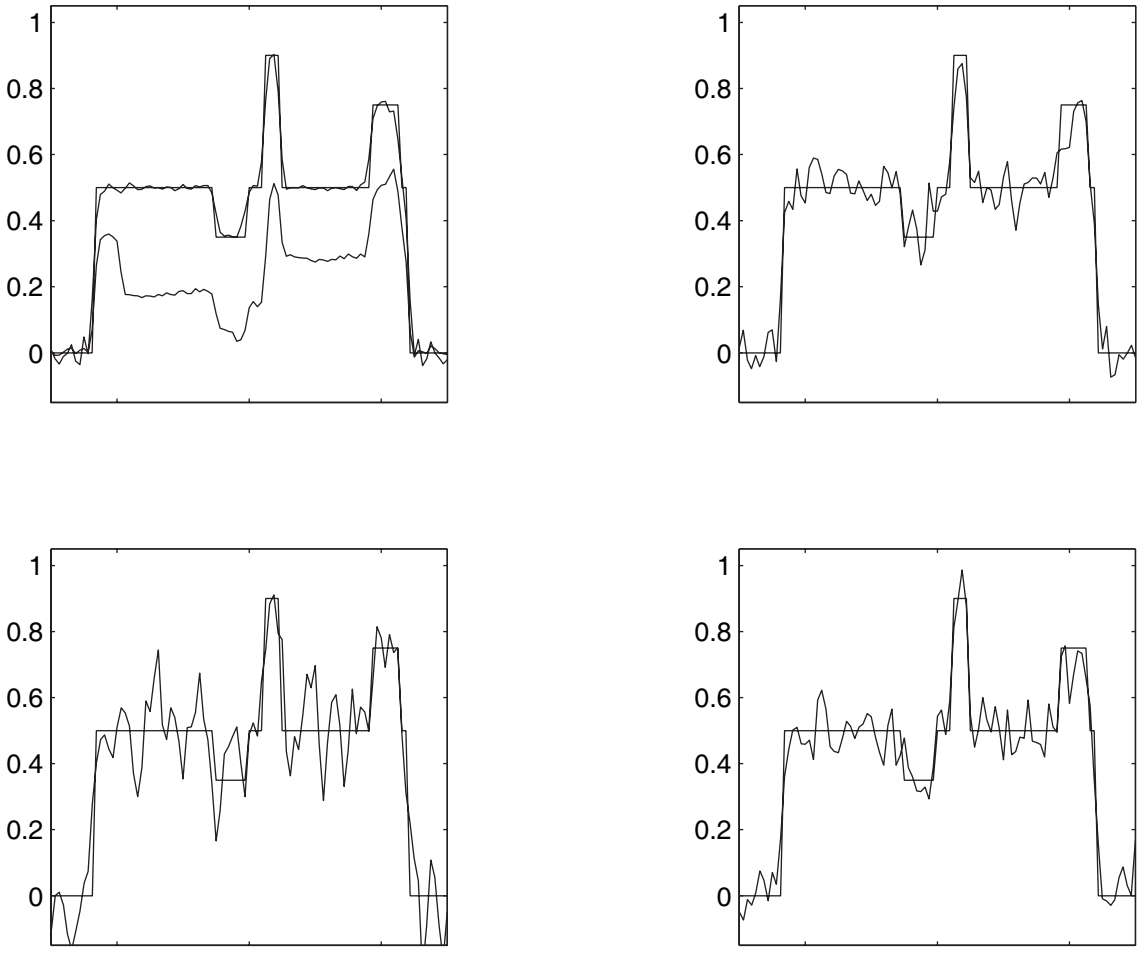

Figure 6: Cross section plots for inversion formula, experiment 2. (top left): No noise added to the data. The bottom line plots the cross section of the reconstruction that does not use attenuation correction. (top right): 10% noise added to the SPECT activity. (bottom left): 17% noise added to the SPECT data. (bottom right): 10% noise added to both the SPECT activity and the attenuation map.

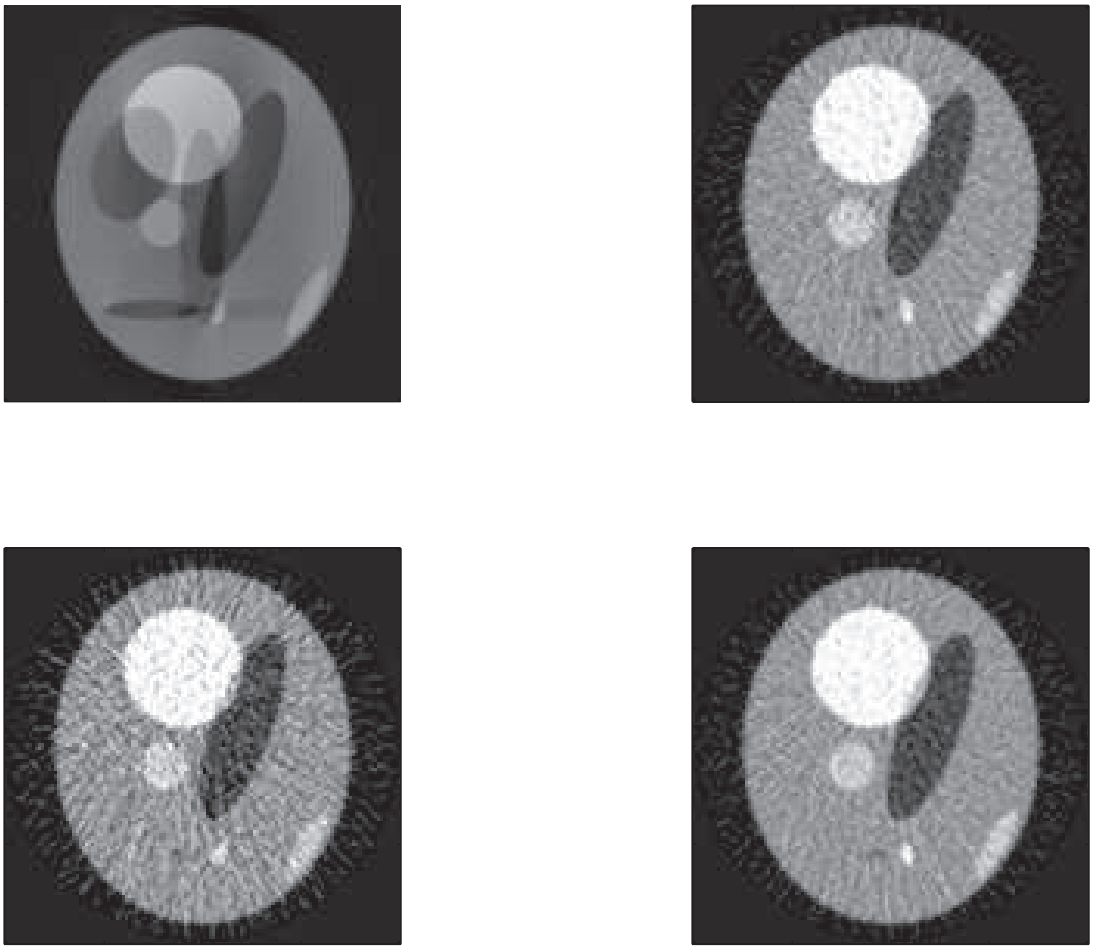

Figure 7: Inversion formula reconstructions with added noise and one that does not use attenuation correction for experiment 2. (top left): Reconstruction that does not use attenuation correction. (top right): 10% noise added to the SPECT activity. (bottom left): 17% noise added to the SPECT data. (bottom right): 10% noise added to both the SPECT activity and the attenuation map.

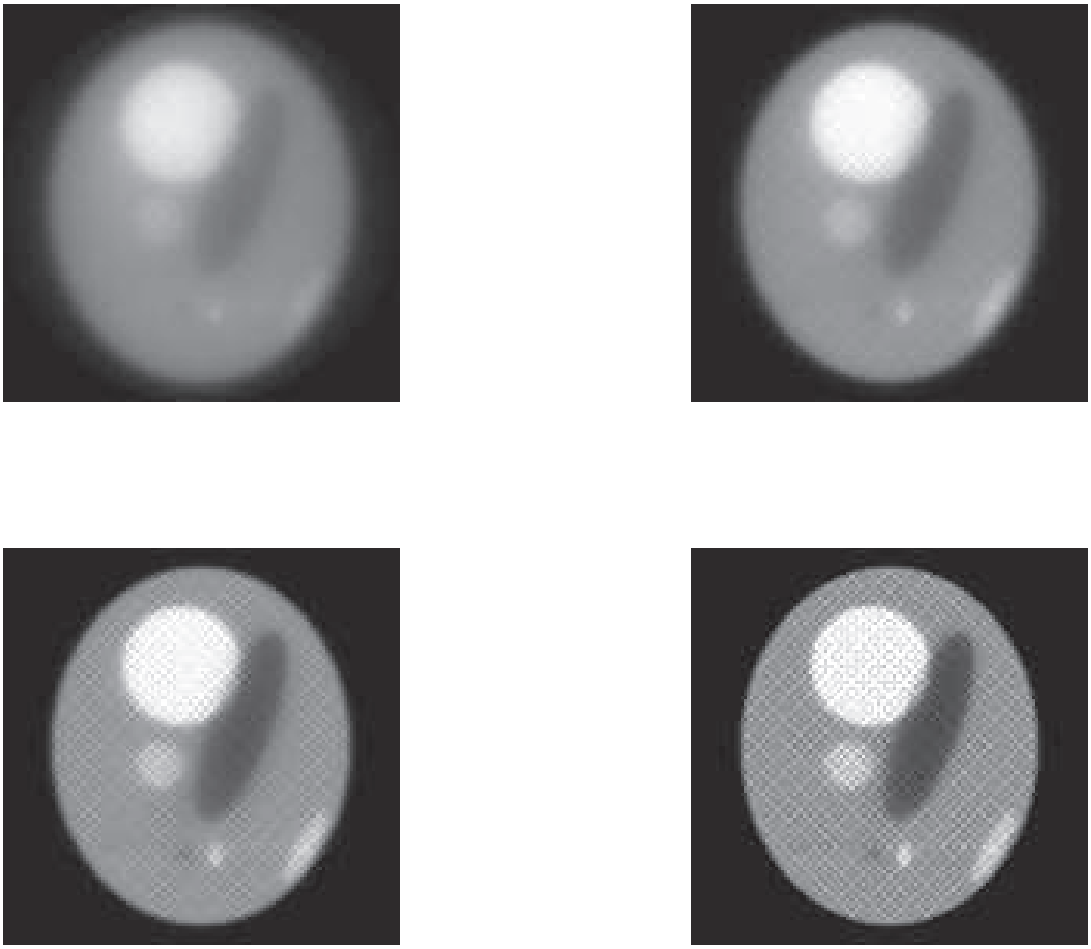

Figure 8: EM algorithm reconstructions for experiment 2. (top left): Image after 4 iterations of the EM algorithm. (top right): Image after 8 iterations of the EM algorithm. (bottom left): Image after 16 iterations of the EM algorithm. (bottom right): Image after 32 iterations of the EM algorithm.

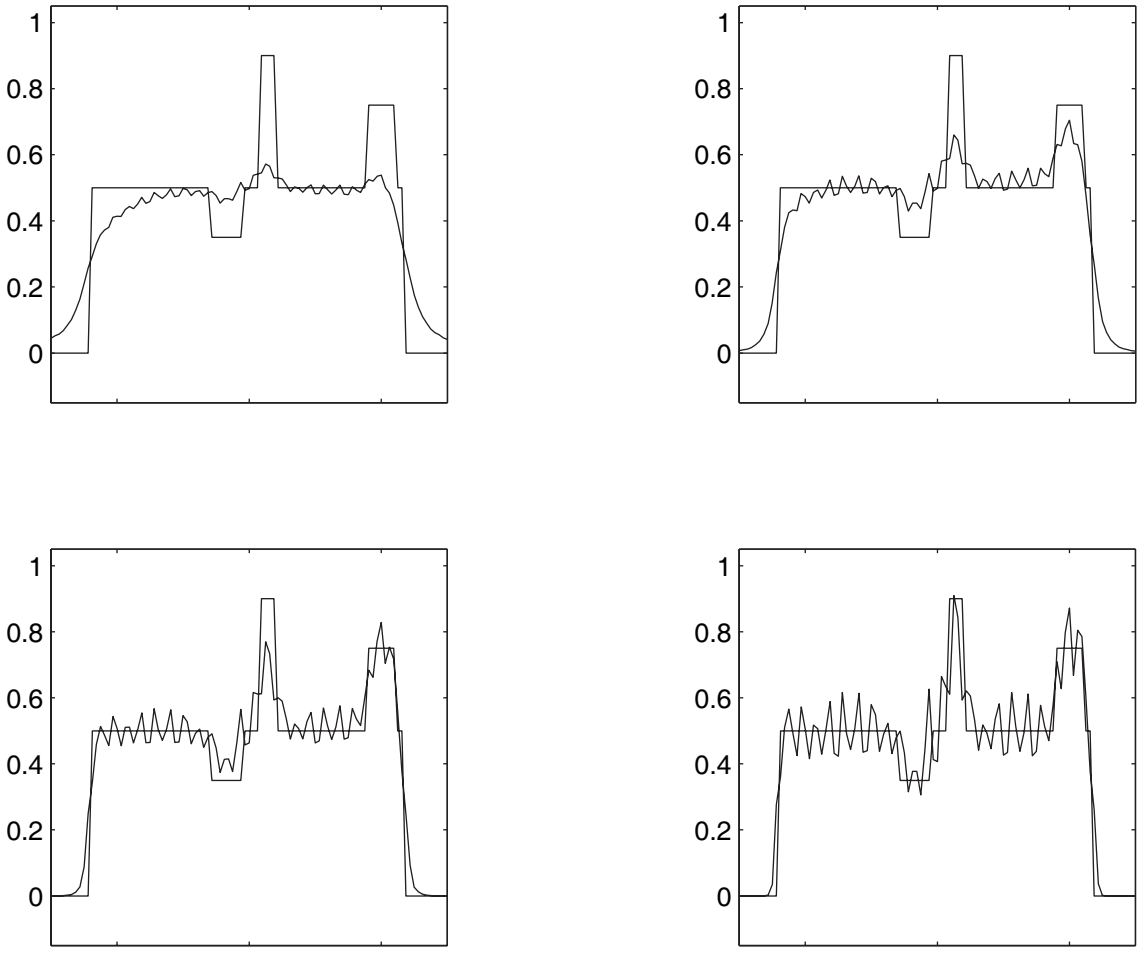

Figure 9: Cross section plots for the EM algorithm, experiment 2. (top left): Plot after 4 iterations of the EM algorithm. (top right): Plot after 8 iterations of the EM algorithm. (bottom left): Plot after 16 iterations of the EM algorithm. (bottom right): Plot after 32 iterations of the EM algorithm.

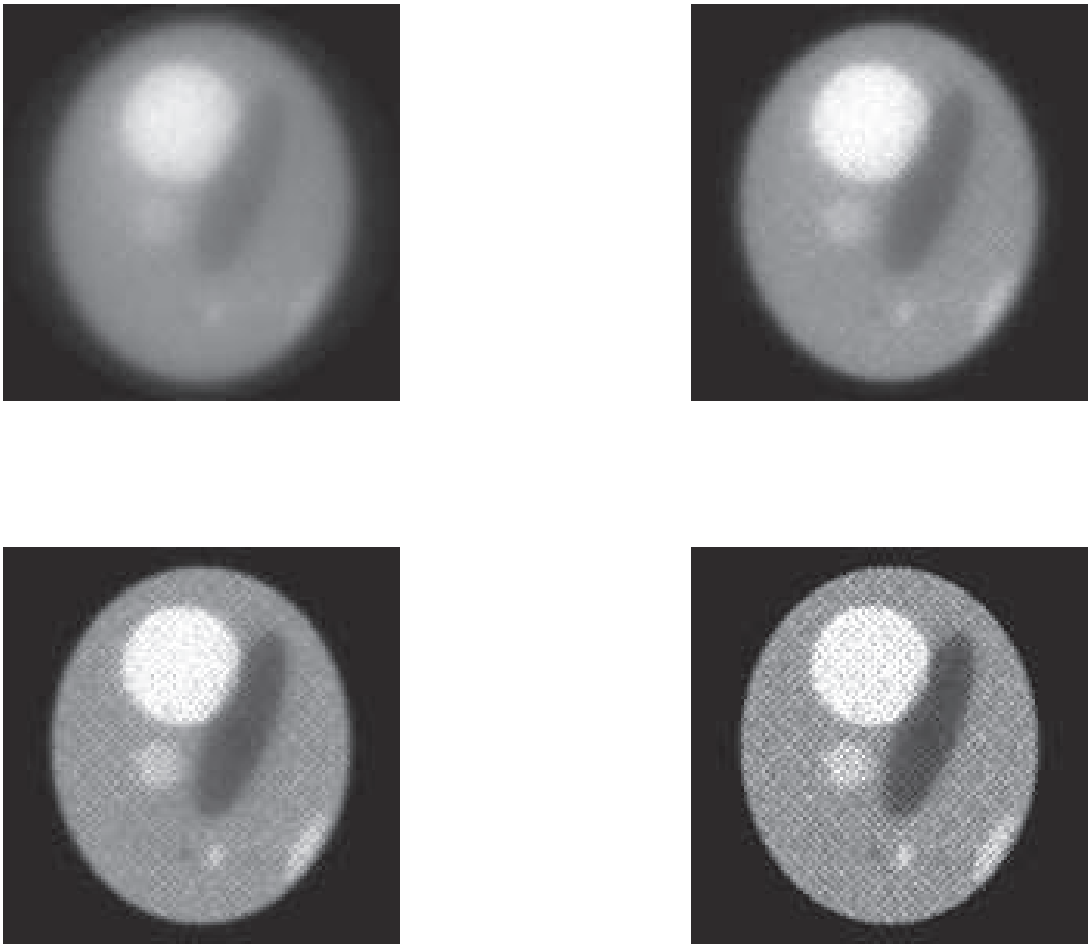

Figure 10: EM algorithm reconstructions with 10% noise added to the SPECT data for experiment 2. (top left): Image after 4 iterations of the EM algorithm. (top right): Image after 8 iterations of the EM algorithm. (bottom left): Image after 16 iterations of the EM algorithm. (bottom right): Image after 32 iterations of the EM algorithm.

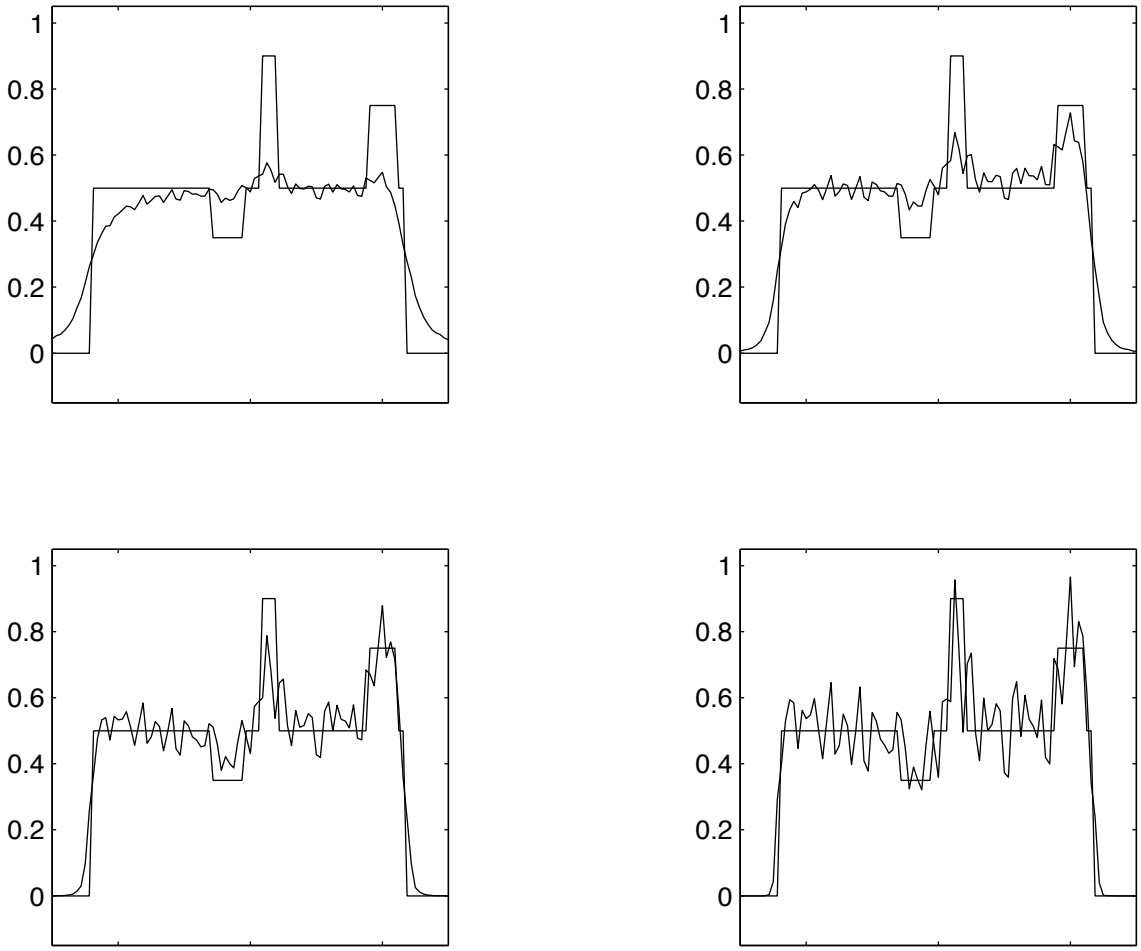

Figure 11: Cross section plots for the EM algorithm with 10% noise added to the SPECT data, experiment 2. (top left): Plot after 4 iterations of the EM algorithm. (top right): Plot after 8 iterations of the EM algorithm. (bottom left): Plot after 16 iterations of the EM algorithm. (bottom right): Plot after 32 iterations of the EM algorithm.