PENALIZED MAXIMUM LIKELIHOOD RECONSTRUCTION METHODS FOR EMISSION TOMOGRAPHY

By

JEFFREY A. ZAHNEN

A DISSERATATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

This dissertation is dedicated to my grandmother, Lena Zahnen, for always believing in me.

ACKNOWLEDGMENTS

For his patience and advice through the years, I offer my greatest thanks to my advisor, Dr. Bernard Mair. In addition, I would like to thank Dr. David Gilland, Dr. William Hager, Dr. James Keesling, Dr. Scott McCullough and Dr. David Wilson for taking the time to serve on my committee and providing suggestions to improve the work. Finally, I would like to thank my parents for their unwavering support of my studies.

TABLE OF CONTENTS

page

LIST OF TABLES

Table page

LIST OF FIGURES

Abstract of Disseratation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

PENALIZED MAXIMUM LIKELIHOOD RECONSTRUCTION METHODS FOR EMISSION TOMOGRAPHY

By

Jeffrey A. Zahnen

December 2006

Chair: Bernard Mair
Major Department: Mathematics

Positron emission tomography (or PET) has been utilized for years in medical diagnosis. The PET reconstruction problem consists of recovering the image from the data gathered by the detectors in the machine. Penalized maximum likelihood techniques allow for the recovery of the image with a smoothing term to improve the image. Previous methods using this technique forced the use of a weak penalty term in order to gain convergence. Using a new method, this restriction is removed while retaining convergence. In addition, the method is extended to the problem of recovering multiple PET images along with a vector describing the motion between the images.

CHAPTER 1
MATHEMATICS OF POSITRON EMISSION TOMOGRAPHY (PET)

1.1 Introduction To PET

Emission tomography has been used for years to study metabolic processes in the body. In this procedure, a patient is injected with a short-lived radioactive isotope. While the substance is being absorbed by the tissues in the body, the patient is placed in a scanner. As the isotope decays, it begins to emit positrons. These emitted positrons travel only a short distance before interacting with a nearby electron. In positron emission tomography (PET), the subsequent annihilation produces a pair of photons traveling in opposite directions on a line oriented in a uniformly random direction. A PET scanner consists of individual detector crystals arranged around the patient in a cylindrical manner. Each pair of detectors forms a virtual detection tube. Figure 1–1 displays a tube formed by detectors α and β. A near-simultaneous arrival of two photons at opposite ends of

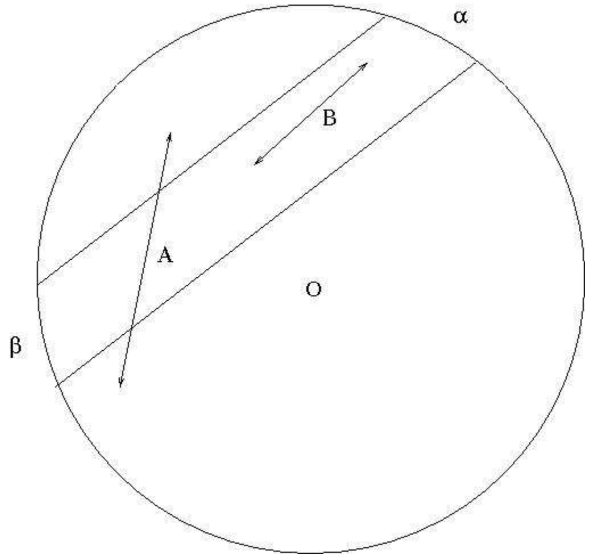


Figure 1–1: A 2-D PET tube registering an emission

the tube is then registered as a single detection. The tube shown in Figure 1–1 will register the displayed emission at point B but not the emission at point A.

In single photon emission computed tomography (SPECT), a single photon is emitted. In this method (more widely applied than PET due to the relative costs of the scanning machines), the detectors are located only on one side of the patient and a single detection results in a count. This scan is performed for different angles between the patient and detector to provide a full image.

In both procedures, the aim is to determine the number of emissions from the various locations in the patient. The resulting intensity images are used to make medical diagnoses. This work will consider the problem of PET image reconstruction using the maximum likelihood (ML) method with the addition of a penalty term. Although developed for PET reconstruction, the method can be extended for SPECT reconstruction. Unlike previous methods, the revised algorithm presented uses a more general penalty term, allowing for a wider range of settings. Proof of convergence and reconstruction results are obtained using this method.

1.2 Mathematical Model for PET

The 2-dimensional PET scanner consists of a circle of detectors surrounding the region of interest, Ω, which can be taken to be rectangular. There are I virtual detector tubes, each one formed by two of the detectors composing the scanner ring. Ω is partitioned into J smaller-sized rectangles, called voxels. An emission in voxel j is detected by tube i if the emitted positrons are detected simultaneously in the two detectors forming the tube. Shepp and Vardi [16] introduced a mathematical model for PET in 1982. In the model, the random emissions from voxel j are modeled by a Poisson process with mean f_j. Thus the detector count in tube i, Y_i, will also be governed by a Poisson process with mean

$$\lambda_i = \sum_j p_{i,j} f_j \tag{1–1}$$

where $p_{i,j}$ is the geometric probability that an emission in voxel j is registered by detector tube i. Using the properties of the Poisson distribution, the probability that there are y_i counts registered in detector i is given by

$$\frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}.$$

To recover the emission intensity, it is necessary to estimate $\mathbf{f} = [f_1, f_2, ..., f_J]$ based on a random sample from $\mathbf{Y} = [Y_1, Y_2, ..., Y_I]$.

1.3 PET Probability Function

The probability function $p_{i,j}$ for the 2-dimensional PET scanner can be defined using the geometry of the scanner. Given any tube i formed by two detectors, it can be divided into six regions as shown in Figure 1–2. The voxels are assumed to be so small that the probability can be assumed approximately constant on the entire voxel. If $\mathbf{z}$ is the center of voxel j, then $p_{i,j}$ is defined by the

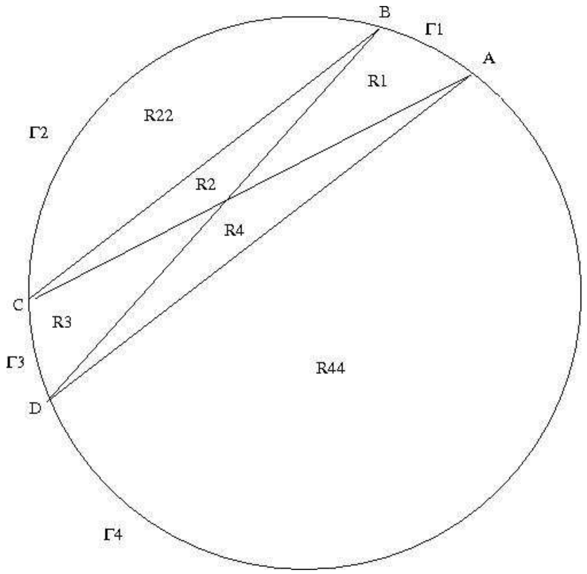


Figure 1–2: Tube regions

probability that the path of a pair of photons emitted from the center of the voxel, **z**, intersects both detectors defining the tube. This probability is proportional to the angle-of-view of the tube from **z**. Thus, if $C\hat{\mathbf{z}}D$ is defined to be the acute angle formed by the lines passing through C and **z** and D and **z**, the following definition results [12].

$$\pi p_{i,j} = \begin{cases} C\hat{z}D & \text{if } z \in R_1 \\ \pi - B\hat{z}C & \text{if } z \in R_2 \\ A\hat{z}B & \text{if } z \in R_3 \\ \pi - D\hat{z}A & \text{if } z \in R_4 \\ 0 & \text{if } z \in R_{22} \text{ or } R_{44} \end{cases} \tag{1–2}$$

1.4 Reconstruction Techniques

The reconstruction problem is to recover the emission intensity **f** given the scanner data **y**, assumed to be Poisson-distributed. Several methods have been used to accomplish this reconstruction. If the detectors are treated as points, the model reduces to line integrals along detector lines. Filtered-back projection (FBP) is a popular method utilized to recover the emission intensity, as it provides a quick way to invert the Radon transform. However, it has been seen to produce images with streaking artifacts [16]. In 1998, Carroll [3] decomposed the data, probability matrix and intensity functions into basis functions, transforming the problem into an infinite set of linear equations to solve. Our algorithm though is based on a third method, Maximum likelihood estimation (ML).

1.4.1 Maximum Likelihood

In the ML method, a search is made for the emission intensity **f** that maximizes the probability of obtaining the Poisson-distributed data. Each detector count y_i is assumed to be an independent sample from a Poisson distribution with a mean corresponding to $\lambda_i = Pf_i = \sum_j p_{i,j} f_j$. Thus the probability that there are

y_i counts in detector i is given by

$$P(Y_i = y_i|\lambda_i) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} \tag{1–3}$$

The detector counts are independent random variables, so these probabilities are multiplied together to yield the probability of obtaining the data from an emission intensity map of $\mathbf{f}$. An ML estimate is an intensity $\mathbf{f}$ that maximizes the following probability.

$$\Pi_i \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} = \Pi_i e^{-Pf_i}(Pf_i)^{y_i}/y_i!$$

In addition, the intensity map, $\mathbf{f}$ is constrained to be non-negative $(\{\mathbf{f}|f_j \geq 0\})$.

An equivalent problem is to apply the negative logarithm and perform a minimization of the term

$$\sum_i (Pf_i - y_i \log Pf_i - \log y_i!).$$

As $y_i!$ does not depend on the unknown intensity $\mathbf{f}$, it can be safety dropped from the term resulting in the ML objective function.

$$L(\mathbf{f}) = \sum_i Pf_i - y_i \log Pf_i \tag{1–4}$$

The Expectation-Maximization (or EM) algorithm developed by Shepp and Vardi [16] can then applied to find the minimizer of this term, resulting in the EMML algorithm.

$$f_j^{n+1} = f_j^n \frac{\sum_i y_i p_{i,j}/Pf_i}{\sum_i p_{i,j}} \tag{1–5}$$

where $\mathbf{f}^n$ refers to the n^{th} step in the iteration.

Alternately, the EMML iteration can be expressed as a descent algorithm.

$$f_j^{n+1} = f_j^n - f_j^n \frac{\partial L}{\partial f_j}(\mathbf{f}^n)/\sum_i p_{i,j}$$

Restated in matrix form yields the equation

$$\mathbf{f}^{n+1} = \mathbf{f}^n - \alpha \mathbf{D}(\mathbf{f}^n) \nabla L(\mathbf{f}^n).$$

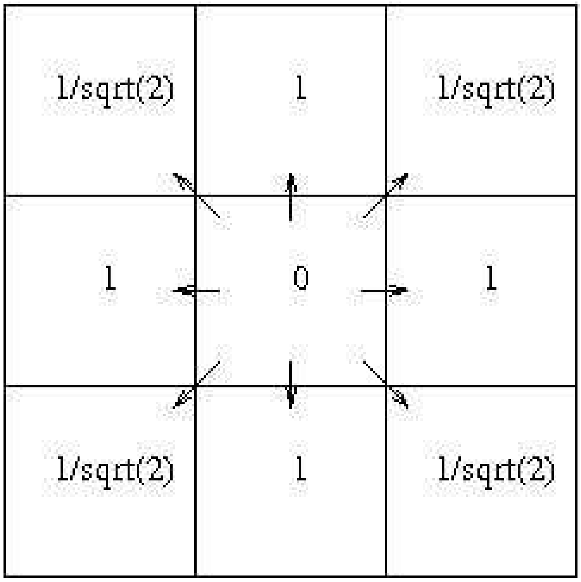
where $\alpha = 1$ is the step-size and $\mathbf{D}$ is the diagonal matrix whose diagonal entries consist of $d_{j,j} = f_j^n / \sum_i p_{i,j}$. While the EM algorithm has been shown to monotonically decrease the objective function and be globally convergent, a well-known deficiency is a tendency to produce noisy images if the iteration is run too long [8].

1.4.2 Penalized Maximum Likelihood (PML)

To help reduce the noise, a smoothing term can be added to the likelihood term resulting in the penalized maximum likelihood (PML) method. Equivalently, this method can be considered as an application of MAP (maximum a-posteriori), in that the recovered image is assumed to have a known local Gibbs probability distribution. This method, used by Geman and McClure [7], is employed by first choosing a penalty function ϕ and then forming the term

$$U(f) = \sum_i \sum_j \omega_{i,j} \phi(f_i - f_j) \tag{1–6}$$

where ω is a weighting factor that associates pixels. This can be used to link neighboring pixels, forcing them to have similar intensities. For example, N_i, can be defined on each voxel i to be the set of pixels in a neighborhood lattice of that voxel. Thus the weight $\omega_{i,j} = 0$ if $j \notin N(i)$. If $j \in N(i)$, the weight is defined to be a nonzero constant. One possibility is to let N_i consist of the eight neighboring voxels as shown in Figure 1–3 with associated weights. In addition, a constant value, γ is multiplied to the penalty term to control the influence of the penalty in the reconstruction. As $\gamma \to 0$, the reconstructions approach those of the EM algorithm. However, if γ is chosen to be too large, the images will be oversmoothed.



1/sqrt(2)	1	1/sqrt(2)
1	0	1
1/sqrt(2)	1	1/sqrt(2)

Figure 1–3: Sample neighborhood system with associated weights

There exist many choices for the penalty function $\phi(r)$. A frequently used choice is a penalty of the form $\phi(r) = \log\cosh\frac{r}{\delta}$ [8],where δ is a previously chosen constant that allows for some fine-tuning of the penalty. This constant determines how the prior is scaled with respect to r, as the $\log\cosh$ penalty will smooth all differences above a certain cutoff value, determined by δ. As $\delta \to \infty$, the method will approach results of the EM Algorithm (as no smoothing will occur). This penalty is useful as it approximates quadratic smoothing for small values of r, but its derivative remains bounded.

In 1993, Lalush and Tsui [9] developed a new penalty function that allowed for even greater control over the penalized range of intensities. Noting that current PML methods did not require the penalty function to be calculated, they instead built the function from properties of its derivative. Thus, they produced

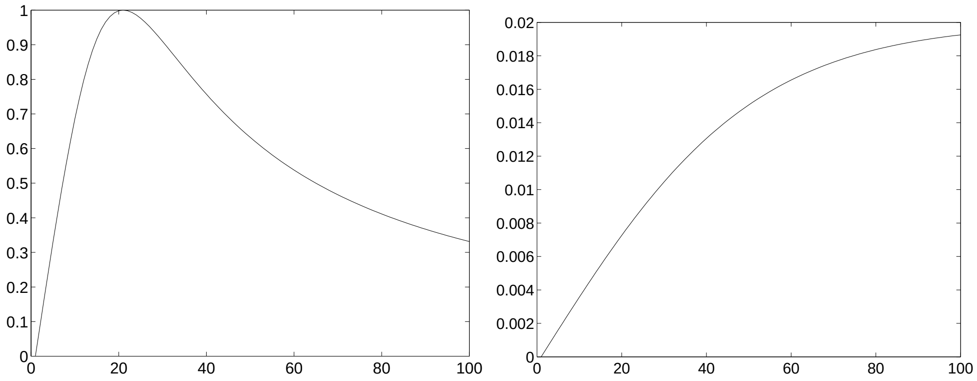
$$\frac{\partial\phi}{\partial r} = \frac{r}{|r|}[1 + \alpha^2(\frac{r}{\delta} - \frac{\delta}{r})^2]^{-1/2} \tag{1–7}$$

where α and δ are constants that affect the spread and location respectively of peak of the derivative. As α increases, the peak becomes more narrow and as δ

increases, the peak away from the origin. The plot of the derivative shows the majority of the smoothing occurring in the region under the peak of the derivative [9]. Thus altering the values of α and δ will affect the range of smoothing in the image. By comparing this function to the derivative of the logcosh function,

$$\phi'(r) = (1/\delta)\tanh\frac{r}{\delta}$$

it can be seen that while the logcosh function does not have a peaked derivative (it increases to an asymptotic limit), it can be considered as an infinitely-wide peak. Thus, Green's logcosh penalty can be approximated as a Lalush-Tsui penalty by taking α close to zero (producing a near infinitely-wide peak). Integrating the Lalush-Tsui penalty term results in their penalty function:



(a) Derivative of the Lalush-Tsui penalty (b) Derivative of the logcosh penalty

Figure 1–4: Derivatives of the penalty terms

$$\phi(r) = \frac{\delta}{2\alpha}\log[1 - 2\alpha^2(1 - \frac{r^2}{\delta^2}) + \frac{2\alpha}{\delta^2}\sqrt{\alpha^2(\delta^4 + r^4) + \delta^2 r^2(1 - 2\alpha^2)}].$$

While Green's original OSL algorithm does not require the calculation of this term, other algorithms do require the frequent calculation of the objective function. In that case, the Lalush-Tsui penalty will have a larger computation time than other penalties. In addition, the general Lalush-Tsui penalty does not have the strict convexity condition needed for convergence to a unique minimizer.

1.5 Minimizing Function

Given a choice of penalty term U and constant γ, the PML objective function thus becomes

$$E(\mathbf{f}) = \sum_i Pf_i - y_i \log Pf_i + \gamma U(\mathbf{f}) \tag{1–8}$$

In Maximum Likelihood, this objective function must be minimized over the set of positive intensity values $\{\mathbf{f} | f_j \geq 0\}$. This minimum must satisfy the Kuhn-Tucker optimality conditions for the problem. In this situation, those two conditions are that for all j

$$\frac{\partial E}{\partial f_j}(\mathbf{f}) \geq 0 \tag{1–9}$$

$$f_j \frac{\partial E}{\partial f_j}(\mathbf{f}) = 0$$

It will be shown that the revised algorithm presented in Chapter 3 will meet these conditions at convergence.

CHAPTER 2
PML RECONSTRUCTION METHODS

2.1 Green's One Step Late Algorithm

Applying the Maximum Likelihood method, a search is made to minimize the penalized objective function:

$$E(\mathbf{f}) = \sum_i (Pf_i - y_i \log Pf_i) + \gamma U(\mathbf{f})$$

Differentiating the objective function and replacing it in the 2nd Kuhn-Tucker optimality condition (1–9) yields the condition that for all j:

$$f_j[\sum_i (p_{i,j} - \frac{y_i p_{i,j}}{Pf_i}) + \gamma \frac{\partial U}{\partial f_j}(\mathbf{f})] = 0.$$

Regrouping terms in this sum produces

$$f_j[\sum_i p_{i,j} + \gamma \frac{\partial U}{\partial f_j}(\mathbf{f})] = f_j \sum_i \frac{y_i p_{i,j}}{Pf_i}.$$

Now define

$$r_j(\mathbf{f}) = (\sum_i p_{i,j} + \gamma \frac{\partial U}{\partial f_j}(\mathbf{f}))^{-1}.$$

With this definition, it can be seen that the second Kuhn-Tucker condition is equivalent to

$$f_j = f_j r_j(\mathbf{f}) \sum_i \frac{y_i p_{i,j}}{Pf_i}. \tag{2–1}$$

This form for the optimality condition is useful as it reformulates the equation into a fixed point condition.

In 1990, Green developed the One-Step-Late (OSL) algorithm [8] to apply the EM algorithm to the Penalized Maximum Likelihood problem. The One-Step-Late name comes from the choice to evaluate the penalty at the current iterate rather than the (unknown) update, as it allowed the EM algorithm to be applied to the PML problem. In Green's OSL, the update to iterate $\mathbf{f}^n$ is given by

$$f_j^{n+1} = f_j^n r_j(\mathbf{f}^n) \sum_i \frac{y_i p_{i,j}}{P f_i^n}.$$

Using the fact that $\frac{\partial E}{\partial f_j}(\mathbf{f}) = \sum_i p_{i,j} - \frac{y_i p_{i,j}}{P f_i} \gamma \frac{\partial U}{\partial f_j}(\mathbf{f})$, the update can be expressed in the alternative form:

$$\mathbf{f}^{n+1} = \mathbf{f}^n + \mathbf{v}(\mathbf{f}^n) \tag{2–2}$$

where

$$v_j(\mathbf{f}) = -f_j r_j(\mathbf{f}) \frac{\partial E}{\partial f_j}(\mathbf{f}). \tag{2–3}$$

Thus if Green's OSL algorithm converges, it will satisfy the same fixed point condition as the second Kuhn-Tucker condition. Note that if $\gamma = 0$, Green's OSL algorithm reduces to the EMML Algorithm (1–5). However, Green's OSL algorithm does not guarantee positivity of the iterates or convergence.

2.1.1 Lange's Improvement to Green's OSL

In 1990, Lange showed that altering Green's algorithm to include a line search could enforce convergence and positivity [10]. In Lange's algorithm, $r_j(\mathbf{f}^n)$ is required to be be positive for all j. To meet this condition, the penalty parameter γ be must be chosen small enough to keep $r_j(\mathbf{f}^n) > 0$ for all possible iterates $\mathbf{f}^n$ and voxels j. In the method, Green's OSL update is altered to include a step-size parameter s (taking s=1 reduces the method to the original OSL method). Thus the update becomes

$$\mathbf{f}^{n+1} = \mathbf{f}^n + s^n \mathbf{v}(\mathbf{f}^n) \tag{2–4}$$

where v is defined as before. The step-size s^n is chosen to be a positive value that gives both positivity of the new iterate (taking $s^n < 1$ enforces this condition) and a decrease in the objective function E. To decrease the objective function, define

$$P(s) = E(\mathbf{f}^n + s\mathbf{v}(\mathbf{f}^n)) \tag{2–5}$$

If $P'(0) < 0$, the algorithm will be decreasing for small values of the step-size. With this condition that $r_j(\mathbf{f}^n) > 0$, it follows that $P'(0) = -\sum_j f_j^n r_j(\mathbf{f}^n)(\frac{\partial E}{\partial f_j})^2(\mathbf{f}^n) < 0$ if $\mathbf{f}_j^n \frac{\partial E}{\partial f_j}(\mathbf{f}^n) \neq 0$ for some j. Since $P'(0) < 0$, the objective function will be decreasing for small values of the step-size s. A line-search is then performed for an appropriate step-size that will give a positive update and a decrease in the objective function. In Lange's algorithm, the step-size that gives the greatest decrease in the objective function is denoted $\bar{s}$ and a search is performed to find a value $s < \bar{s}$ within ϵ of $\bar{s}$. This condition requires an exact line search, eliminating the possibility of faster inexact search algorithms.

2.1.2 Mair-Gilland Method

One way to remove the restriction of positivity of the r-term is to allow the constant s to be a negative value. In this method (developed for use in the RM Algorithm [14]), the step-size s^n is allowed to be negative. The sign of s^n depends on the sign of $P'(0)$. If $P'(0) < 0$, the objective function will be decreasing for small positive values of the step-size. In this case, the step-size is chosen as in Lange's method. A search is made for a value that ensures positivity of the iterate, but is also within ϵ of the value that gives the greatest decrease. If $P'(0) > 0$, then E will be increasing for small positive values of s. In this case, s^n is allowed to be negative. The search is made in the negative direction, again for a value that ensures positivity and is within ϵ of the value that gives the greatest decrease in the objective function. Although this method will decrease the objective function and convergence, the limit may not satisfy the second Kuhn-Tucker optimality condition.

2.2 Accelerated Penalized Maximum Likelihood Method (APML)

The accelerated penalized maximum likelihood method (APML) was developed in 2004 by Chang et al [5]. In the APML method, surrogate functions to decrease the objective function and provide convergence. In the method, a penalty function

$\phi(r)$ and associated neighborhood system is chosen as described in Chapter 1. In addition, the related function $\gamma(r) = \dot{\phi(r)}/r$ is required to be non-increasing. In the method, the trial iterate is formed by

$$\tilde{\mathbf{f}}_j^{n+1} = -G_j(\mathbf{f}^n) + \sqrt{G_j(\mathbf{f}^n)^2 - 8E_j(\mathbf{f}^n)F_j(\mathbf{f}^n)}/4F_j(\mathbf{f}^n) \tag{2–6}$$

with the defined functions given by

$$\begin{aligned} E_j &= -\frac{y_i p_{i,j} f_j^n}{\sum_j p_{i,j} f_j^n} \\ F_j &= 2\beta \sum_{k \in N_j} \omega_{i,j}\gamma(f_j - f_k) \\ G_j &= \sum_i p_{i,j} - 2\beta \sum_{k \in N_j} \omega_{i,j}\gamma(f_j - f_k)(f_j + f_k) \end{aligned}$$

As this was seen to converge slowly, an acceleration factor (the APML Method) was added to the algorithm [4]. As in Lange's method, a step-size factor is added to the algorithm. In APML, the accelerated update is given by

$$\mathbf{f}^{n+1} = \tilde{\mathbf{f}}^n + s\mathbf{v}(\mathbf{f}^n) \tag{2–7}$$

with $\mathbf{v}(\mathbf{f}^n) = \tilde{\mathbf{f}}^{n+1} - \mathbf{f}^n$. In Lange's method, a search method is employed to find a suitable choice for the step-size. Here, surrogate functions are used to find the best possible step-size needed to decrease the objective function. Surrogate functions were first applied to PET reconstruction by DePierro [6]. In order to minimize a function f over a set S using the method of surrogate functions, another function ψ must be found satisfying

- $f(\mathbf{x}^n) = \psi(\mathbf{x}^n)$
- $f(\mathbf{x}) \geq \psi(\mathbf{x})$ for all $\mathbf{x} \in S$.

The difficulty in using this method lies in finding an appropriate function ψ. In APML, creating the surrogate function requires the calculation of four separate functions related to the penalty function, possibly increasing the computation time. Reconstructions gathered using the APML algorithm will be compared to reconstructions using the revised algorithms presented in this paper.

CHAPTER 3
GENERAL PML ALGORITHM

3.1 General Algorithm Statement

In the maximum likelihood method, a search is made for an emission intensity $\mathbf{f}$ that minimizes the objective function

$$E(\mathbf{f}) = L(\mathbf{f}) + \gamma U(\mathbf{f})$$

under the restriction that $f_j \geq 0$ for all j. In this chapter, an iterative algorithm to find this minimum will be presented. Under specific conditions, convergence to a minimizer will be proven.

3.1.1 General Algorithm Update

The new algorithm is formed by taking Lange's modification to OSL, but treating the step-size as a vector instead of a constant. To apply the algorithm, initialize first by taking $\mathbf{f}_j^0 = \sum_i y_i / J$. Given a current iterate $\mathbf{f}^n$, the next iterate, $\mathbf{f}^{n+1}$ can be constructed by the following update:

$$\mathbf{f}^{n+1} = \mathbf{f}^n + \mathbf{s}^n \odot \mathbf{v}(\mathbf{f}^n) \tag{3–1}$$

where

- $\mathbf{s}^n$ is the step-size vector
- $r_j(\mathbf{f}) = (\sum_i p_{i,j} + \gamma \frac{\partial U}{\partial f_j}(\mathbf{f}))^{-1}$
- $v_j(\mathbf{f}) = -f_j r_j(\mathbf{f}) \frac{\partial E}{\partial f_j}(\mathbf{f})$
- $(\mathbf{x} \odot \mathbf{y})_j = x_j y_j$ (the Hadamard product).

Note that $\mathbf{v}(\mathbf{f})$ and $\mathbf{r}(\mathbf{f})$ are the same defined functions as in the OSL algorithm.

Given a current iterate $\mathbf{f}^n$, the step-size vector $\mathbf{s}^n = [s_1^n, ..., s_J^n]$ is chosen to produce the next step in the algorithm. This vector must satisfy the conditions:

$$\begin{aligned} 0 < |s_j^n| &< K \text{ for some large constant K} && (3–2)\\ |s_j^n| &< (|r_j(\mathbf{f}^n)\frac{\partial E}{\partial f_j}(\mathbf{f}^n)|)^{-1} && (3–3)\\ E(\mathbf{f}^n + \mathbf{s}^n \odot \mathbf{v}(\mathbf{f}^n)) &< E(\mathbf{f}^n)\text{if}||\mathbf{v}(\mathbf{f^n})|| \neq 0 && (3–4)\\ (\mathbf{s}^n \odot \mathbf{r}(\mathbf{f}^n))_j &> 0. && (3–5) \end{aligned}$$

The first three conditions on the step-size vector enforce respectively boundedness of the step-size, positivity of the iterate and decrease in the objective function. The final condition requires that for each voxel j, the sign of s_j matches the sign of $r_j(\mathbf{f})$. Note that a unique $\mathbf{s}$ is not guaranteed. The revised algorithm presented after the proof provides an example of one way to choose a suitable step-size vector. Once a penalty function and method of constructing a step-size vector (both matching the required criteria) have been chosen, the algorithm can proceed. Continue iterating until a suitable stopping criteria has been reached. Convergence of this algorithm will now be investigated.

3.1.2 Penalty function criteria

The general penalty term $U(\mathbf{f})$ is of the form

$$U(\mathbf{f}) = \sum_i \sum_j \omega_{i,j}\phi(f_i - f_j) \qquad (3–6)$$

where $\phi(r)$ is required to satisfy the following properties:

- $\phi(0) = 0$
- $\lim_{r\to\infty} \phi(r) \to \infty$
- $\phi(-r) = \phi(r)$ (ϕ is an even function)
- $\phi \in C^1$ (ϕ has a differentiable first derivative)
- ϕ is a strictly convex function ($\phi''(r) > 0$).

The first three conditions are basic conditions for a penalty function, while the latter two conditions are required for convergence. Functions that match these conditions include the logcosh and quadratic penalties. Note that the general Lalush-Tsui penalty does not meet these conditions due its lack of convexity.

3.2 Proof of Convergence

Theorem 1. *Let ϕ be a penalty function satisfying the conditions stated in Section (3.1.2). Let E be the PML objective function $E(\mathbf{f}) = L(\mathbf{f}) + \gamma U(\mathbf{f})$. Define $\mathbf{f}_j^0 = \sum_i y_i/J$ for all j (the algorithm is initialized with a flat emission intensity). Let the algorithm update be given by (3–1) with the step-size vector chosen such that conditions (3–2 - 3–5) are satisfied. The algorithm will then converge to a unique minimizer.*

The proof (extended from Lange's PML proof [10]) will demonstrate that the algorithm converges to a unique vector that satisfies both Kuhn-Tucker optimality conditions, hence is the minimizer of E. It consists of two parts. In the first, Zangwill's convergence theorem is applied to prove that the objective function converges and that all limit points will fulfil the second Kuhn-Tucker condition. In the second part of the proof, it will be shown that there is a single limit point and this point will also fulfil the first Kuhn-Tucker condition.

Definition 1. *A stationary point is defined to be any intensity function $\mathbf{f}$ where for all j, $\mathbf{f} \odot \nabla E(\mathbf{f}) = 0$.*

By this definition, the 2nd Kuhn-Tucker condition will be met at a stationary point. Zangwill's Convergence Theorem [11] is now applied.

Theorem 2. *Zangwill's Convergence Theorem. Let T be a point-to-set function defined on a metric space X. Generate a sequence $\{\mathbf{x}^n\}$ such that $\mathbf{x}^{n+1} \in T(\mathbf{x}^n)$. Let a solution set, $S \subseteq X$, be given along with a continuous function E. Suppose that*

1. *the sequence $\{\mathbf{x}^n\}$ is contained in a compact subset of X*
2. *if $\mathbf{x} \notin S$, $E(\mathbf{y}) < E(\mathbf{x})$ for all $\mathbf{y} \in T(\mathbf{x})$*

3. if $\mathbf{x} \in S$*,* $E(\mathbf{y}) \leq E(\mathbf{x})$ *for all* $\mathbf{y} \in T(\mathbf{x})$

4. the mapping T *is closed at points outside of* S*.*

Then the limit of any convergent subsequence lies in S *and* $\{E(\mathbf{x}^n)\}$ *converges monotonically to* $E_S = E(\mathbf{y})$ *for some* $\mathbf{y} \in S$*.*

To apply Zangwill, let $X = R^n_+$, S be the set of stationary points and let E the PML objective function. For any emission intensity $\mathbf{f}$, define the set $S_{\mathbf{f}}$ to be the set of all possible step-size vectors $\mathbf{s}$ fulfilling condition (3–2 - 3–5). Then define the point-to-set function T by the equation:

$$T(\mathbf{f}) = \{\mathbf{f} + \mathbf{s} \odot \mathbf{v}(\mathbf{f}) : \mathbf{s} \in S_{\mathbf{f}}\}. \tag{3–7}$$

Note that if $\mathbf{f} \in S$, $\mathbf{v}(\mathbf{f}) = \mathbf{0}$, thus $T(\mathbf{f}) = \{\mathbf{f}\}$. In the PML algorithm, the next iterate is created by taking $\mathbf{f}^{n+1} = \mathbf{f}^n + \mathbf{s} \odot \mathbf{v}(\mathbf{f}^n)$ for a particular choice of $\mathbf{s} \in S_{\mathbf{f}^n}$, hence $\mathbf{f}^{n+1} \in T(\mathbf{f}^n)$.

Lemma 1. *The point-set mapping* T *is closed at all nonstationary points* θ*.*

The closure definition requires that if $\theta^n \to \theta \notin S$, $\psi^n \in T(\theta^n)$, and $\psi^n \to \psi$, then $\psi \in T(\theta)$. Hence it must be shown that there exists a step-size vector $\mathbf{s} \in S_\theta$ such that $\psi = \theta + \mathbf{s} \odot \mathbf{v}(\theta)$ holds for all $\theta \notin S$. By the definition of the update, for each n there is a step-size vector $\mathbf{s}^n \in S_{\theta^n}$ such that

$$\psi^n = \theta^n + \mathbf{s}^n \odot \mathbf{v}(\theta^n).$$

Since $\frac{\partial U}{\partial f_j}$ is assumed to be continuous, $r_j(\mathbf{f})$ is continuous and hence $v_j(\mathbf{f})$ will also be continuous. Let $W = \{j : v_j(\theta) \neq 0\}$. Since θ is not a stationary point, W is nonempty. For each $j \in W$, there exists a constant $M_j > 0$ such that for all $m_j > M_j$, $v_j(\theta^m) \neq 0$. Therefore for all $j \in W$ and $m > max_j(M_j)$,

$$s_j^m = \frac{\psi_j^m - \theta_j^m}{v_j(\theta^m)}$$

which converges to

$$\frac{\psi_j - \theta_j}{v_j(\theta)}.$$

For $j \in W$, define s_j to be $\frac{\psi_j - \theta_j}{v_j(\theta)}$. Therefore for all $j \in W$, $\psi_j = \theta_j + s_j v_j(\theta)$. If $j \notin W$, $v_j(\theta) = 0$ and hence $\psi_j = \theta_j$. Thus for all j, $\psi_j = \theta_j + s_j v_j(\theta)$ where $s_j = 1$ for $j \notin W$. By the continuity of E, $\mathbf{v}, \theta$ and ψ, $s \in S_\theta$ and hence $\psi \in T(\theta)$.

Lemma 2. *The iterates belong to a bounded and compact set.*

It will be shown that if $||\mathbf{f}^n|| \to \infty$, $E(\mathbf{f}^n) \to \infty$ as well. The log-likelihood term L is a function of the form $\sum_i c_i - y_i log(c_i)$, where $c_i = Pf_i$ and y_i are both positive terms. If $||\mathbf{f}^n|| \to \infty$, then for some i, $f_i^n \to \infty$ as well. For each voxel j, there exists at least one tube i with $p_{i,j} \neq 0$. Hence if the intensity at voxel j becomes unbounded, $c_i = Pf_i \to \infty$ forcing $c_i - y_i log(c_i) \to \infty$. Thus if the iterates become unbounded, the log-likelihood function will also become unbounded. Hence if $||\mathbf{f}^n|| \to \infty$, $E(\mathbf{f}^n) = L(\mathbf{f}^n) + \gamma U(\mathbf{f}^n) \to \infty$. By construction, $E(\mathbf{f}^n) \leq E(\mathbf{f}^0)$ for all n. Thus, it cannot be true that $E(\mathbf{f}^n) \to \infty$. Therefore, the iterates must belong to a bounded set. By construction, $E(\mathbf{f}^{n+1}) \leq E(\mathbf{f}^n)$. Thus $E(\mathbf{f}^n) \leq E(\mathbf{f}^0)$ for all n. Therefore the iterates belong to the set $\{\theta | E(\theta) \leq E(\mathbf{f}^0)\}$. Due to the the continuity of E, the set is closed. Since the set is closed and bounded, it is a compact set.

Theorem 3. *The limit of any convergent subsequence is a stationary point.*

Zangwill's theorem is now applied. Let $X = \mathbf{R}_+^n$ and S = the set of stationary points. If $\mathbf{f}^n \in S$, $\mathbf{v}(\mathbf{f}^n) = \mathbf{0}$ and hence $T(\mathbf{f}^n) = \{\mathbf{f}^n\}$. If $\mathbf{f}^n \notin S$, by construction, $E(\mathbf{f}^n + \mathbf{s}^n \odot \mathbf{v}(\mathbf{f}^n)) < E(\mathbf{f}^n)$ for any step-size vector $\mathbf{s}^n$ satisfying conditions (3–2 - 3–5). By Lemma 1, the point-to-set mapping T is closed at non-stationary points. Thus Zangwill's theorem applies and all limit points are stationary points. In addition, $E(\mathbf{f}^n) \to E_S = E(\mathbf{y})$ for some $\mathbf{y} \in S$. It must be shown that E_S is well-defined. If $\mathbf{x} \in S$, then $\mathbf{x}$ is a limit point of the sequence. Hence there is a subsequence $\{\mathbf{f}^{n_j}\}$ converging to $\mathbf{x}$. Since $E(\mathbf{f}^n) \rightarrow E(\mathbf{y})$, it must be true that $E(\mathbf{x}) = E(\mathbf{y})$.

Lemma 3. *If $E(\mathbf{f}^n) = E_S$ for some n, then $\mathbf{f}^n \in S$.*

In the proof of Zangwill's conditions, $E(\mathbf{f}^{n+1}) < E(\mathbf{f}^n)$ if $\mathbf{f} \notin S$ and $E(\mathbf{f}^{n+1}) \leq E(\mathbf{f}^n)$ if $\mathbf{f} \in S$. Thus if $\mathbf{f}^n \notin S$, $E(\mathbf{f}^{n+1}) < E(\mathbf{f}^n) = E_S$. But $E(\mathbf{f}^n)$ converges monotonically to E_S, giving a contradiction. Thus $\mathbf{f}^n \in S$.

Lemma 4. *The set of limit points is finite.*

The results of Zangwill's Theorem state that all limit points of the sequence are stationary points. A stationary point has been defined to be one where for each index j, $f_j \frac{\partial E}{\partial f_j}(\mathbf{f}) = 0$. With the requirement that U is a strictly convex (penalty) term, E will be a strictly convex function [10]. Therefore, given any subset W of $[1, J]$, E will remain a strictly convex function on the plane $\{f_j = 0 | j \in W\}$. Hence there is a unique point on this plane that is a minimizer for E. Denote this point f_W. Thus for any subset W, there is only one point $\mathbf{f}_W$ with $\frac{\partial E}{\partial f_j}(\mathbf{f}_W) = 0$ for $j \notin W$ and $f_j = 0$ for $j \in W$. Since there are a finite number of possible subsets W, the set of stationary points is finite.

Lemma 5. *The distance between successive iterates tends to zero.*

$||\mathbf{f}^{n+1} - \mathbf{f}^n|| = ||\mathbf{s}^n \odot \mathbf{v}(\mathbf{f}^n)||$. Since $|\mathbf{s}_j^n| < K$ for all n, j, it suffices to prove that $v_j(\mathbf{f}^n)$ goes to zero. The function $\mathbf{v}$ is continuous and is zero for any stationary point. Since the set of limit points is finite, given $\epsilon > 0$, a single $\delta > 0$ can be found such that if $||\mathbf{f}^n - \mathbf{y}|| < \delta$ for any $\mathbf{y} \in S$, then $||\mathbf{v}(\mathbf{f}^n)|| = ||\mathbf{v}(\mathbf{f}^n) - \mathbf{v}(\mathbf{y})|| < \epsilon$. Let Z be the set of all sequence points within δ of a limit point and the W be the remaining sequence points. If W is infinite, it contains a convergent subsequence (the iterates are contained in a compact set). Thus there are points in W arbitrarily close to a limit point, a contradiction. Hence W must be finite. Let N be the largest sequence index in W. Thus if $n > N$, $||\mathbf{f}^n - \mathbf{y}|| < \delta$ for some $\mathbf{y} \in S$. Hence $||\mathbf{v}(\mathbf{f}^n)|| < \epsilon$. Therefore $\mathbf{v}(\mathbf{f}^n) \rightarrow \mathbf{0}$.

Lemma 6. *The set of limit points consists of a single stationary point.*

By Lemma 5, the distance between iterates goes to zero. Ostrowski proved that the set of limit points of such a sequence is both compact and connected [15]. Since this set is finite, it must consist of a single point. Denote this intensity by $\bar{\mathbf{f}}$.

Theorem 4. *The iterates converge to the minimizer of E.*

There is a single limit point for the sequence, hence the sequence will converge to $\bar{\mathbf{f}}$. Assume this function is not the minimizer of E. Since the limit is a stationary point, the second Kuhn-Tucker criterion $\bar{f}_j \frac{\partial E}{\partial f_j}(\bar{\mathbf{f}}) = 0$ is satisfied, so the first optimality condition must be violated. Therefore, an index j can be found such that $\bar{f}_j = 0$ and $\frac{\partial E}{\partial f_j}(\bar{\mathbf{f}}) < 0$ and hence there is an N such that $\frac{\partial E}{\partial f_j}(\mathbf{f}^n) < 0$ for all $n > N$. By construction, $r_j(\mathbf{f}^n)$ and $s_j(\mathbf{f}^n)$ have the same sign and $f_j^n > 0$. Thus for $n > N$,

$$f_j^{n+1} = f_j^n + s_j^n v_j(\mathbf{f}^n) = f_j^n - s_j^n f_j^n r_j(\mathbf{f}^n) \frac{\partial E}{\partial f_j}(\mathbf{f}^n) > f_j^n$$

and f_j^n does not converge to zero, contrary to the assumption. Therefore, both Kuhn-Tucker conditions hold at the convergence point and hence it is the minimizer of E.

If the penalty function lacks convexity (as in the generalized Lalush-Tsui penalty (1–7), the proof of Theorem 4 will not apply as it requires convexity. Thus, as with the modified RM algorithm discussed in the following chapter, the only result that can be proved is that all of the limit points generated from the sequence will be minimizers.

3.3 New PML Algorithm

The general statement does not guarantee the existence of a method of choosing the step-size vector with the needed conditions. However, there do exist several ways of meeting the necessary conditions. In Lange's algorithm [10], all entries in the step-size vector are the same positive value. In this method, conditions (3–2 - 3–5) are satisfied by assuming that $r_j(\mathbf{f})$ is positive for all emission intensities $\mathbf{f}$ and voxels j. With this restriction, the objective function will be decreasing for small positive values of the step-size s. However, in order to ensure the positivity of $\mathbf{r}(\mathbf{f})$, the strength of the penalty term γ may need to be decreased, which alters the objective function needed to be minimized. The revised algorithm presented below has been designed to remove this strong condition on the penalty term by using a different method to choose the step-size, while still providing for the convergence of the iterates.

3.3.1 Non-Uniform Step-Size Method

In this algorithm, the step-size vector consists of two values, one positive and one negative. To find these values at the n^{th} step in the iteration, form the two sets, $D_+(\mathbf{f}^n) := \{j | r_j(\mathbf{f}^n) > 0\}$ and $D_-(f^n) := \{j | r_j(\mathbf{f}^n) < 0\}$. Since $\sum_i p_{i,j} + \gamma \frac{\partial U}{\partial f_j}(\mathbf{f}^n)$ will be finite for each j, $\mathbf{r}(\mathbf{f}^n)$ can never be zero. Therefore, $D_+(\mathbf{f}^n)$ and $D_-(\mathbf{f}^n)$ will partition the set of voxels. These sets will determine which value of the step-size will be applied to the appropriate voxel. Define the equations

$$g_j^n(t_1, t_2) = \begin{cases} f_j^n + t_1 v_j(\mathbf{f}^n) & \text{if } j \in D_+(\mathbf{f}^n) \\ f_j^n + t_2 v_j(\mathbf{f}^n) & \text{if } j \in D_-(\mathbf{f}^n) \end{cases} \tag{3–8}$$

$$\psi(t_1, t_2) = E(\mathbf{g}^n(t_1, t_2)). \tag{3–9}$$

The current value of the objective function at step n is given by $\psi(0,0) = E(\mathbf{f}^n)$. Since the goal is to decrease the value of E, a search is made for a new update in the direction of the negative gradient. This will give the direction of the steepest descent of E. If a single value is used for our constant (as in Lange's method [10] or the Mair-Gilland method [14]), it corresponds to using (t,t) as the search direction in (3–8). This may not be the direction of the greatest decrease. In order to achieve the greatest decrease in the objective function, the vector (t_1, t_2) is assumed to be in the direction of the negative gradient. Therefore, define $(t_1, t_2) = -s\nabla\psi(0,0)$ for some $s \geq 0$ to be chosen later. Calculating the two partial derivatives yields:

$$\begin{aligned}\frac{\partial \psi}{\partial t_1}(0,0) &= \sum_{j \in D_+^n} \frac{\partial E}{\partial f_j}(\mathbf{f}^n) v_j(\mathbf{f}^n) = -\sum_{j \in D_+^n} f_j^n r_j(\mathbf{f}^n) (\frac{\partial E}{\partial f_j})^2(\mathbf{f}^n) \\ \frac{\partial \psi}{\partial t_2}(0,0) &= \sum_{j \in D_-^n} \frac{\partial E}{\partial f_j}(\mathbf{f}^n) v_j(\mathbf{f}^n) = -\sum_{j \in D_-^n} f_j^n r_j(\mathbf{f}^n) (\frac{\partial E}{\partial f_j})^2(\mathbf{f}^n).\end{aligned}$$

Therefore by defining

$$\begin{aligned}\tau_+(\mathbf{f}) &= \sum_{j \in D_+(f)} f_j r_j(\mathbf{f}) (\frac{\partial E}{\partial f_j})^2(\mathbf{f}) \\ \tau_-(\mathbf{f}) &= \sum_{j \in D_-(f)} f_j r_j(\mathbf{f}) (\frac{\partial E}{\partial f_j})^2(\mathbf{f})\end{aligned} \tag{3–10}$$

it can be seen that $(t_1, t_2) = s(\tau_+(\mathbf{f}^n), \tau_-(\mathbf{f}^n))$. Since (τ_+, τ_-) provides a search direction only, it can be normalized at this stage to produce a unit vector $(\tau_+(\mathbf{f}), \tau_-(\mathbf{f}))$ (τ_+ and τ_- will now refer to the normalized vector). The new iterate is thus formed by:

$$\mathbf{f}^{n+1} = \mathbf{f}^n + s\tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n) \tag{3–11}$$

where $\tau_j(\mathbf{f}) = \tau_+(\mathbf{f})$ if $j \in D_+(\mathbf{f})$ and $\tau_j(\mathbf{f}) = \tau_-(\mathbf{f})$ if $j \in D_-(\mathbf{f})$. The problem has thus been reduced from a 2-dimensional search for (t_1, t_2) to a 1-dimensional search for s. Defining $P(s) = E(\mathbf{f}^n + s\tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n))$, yields

$$\begin{aligned}P'(0) &= \nabla E(\mathbf{f}^n + s\tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n)) \cdot \frac{d}{ds}(\mathbf{f}^n + s\tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n))|_{s=0} \\ &= \sum_j -(\frac{\partial E}{\partial f_j})^2(\mathbf{f}^n) \tau_j(\mathbf{f}^n) f_j^n r_j(\mathbf{f}^n) \leq 0.\end{aligned}$$

The final inequality comes from the fact that τ_j has the same sign as r_j (and so (s $\tau_j \mathbf{r}_j \geq 0$ for all j, proving the fourth criteria for the step-size vector). Thus for small positive values of the step-size coefficient s, E will be decreasing. It remains to be shown that the method of selecting the step-size coefficient s will result in a step-size vector, $s\tau$, fulfilling the conditions stated in (3–2 - 3–5).

3.3.2 Line Search Methods

In the non-uniform step-size method discussed above, the step-size vector consists of $s\tau$. Now, a search must be made for an appropriate step-size coefficient to ensure the decrease in the objective function and positivity of the iterate. First, a bounding value, s_{max} must be found to ensure positivity. Using the definition of the update,

$$\begin{aligned} f_j^{n+1} &= f_j^n + s^n \tau_j(\mathbf{f}^n) v_j(\mathbf{f}^n) \\ &= f_j^n - s^n \tau_j(\mathbf{f}^n) f_j^n r_j(\mathbf{f}^n) \frac{\partial E}{\partial f_j}(\mathbf{f}^n) \\ &= f_j^n (1 - s^n \tau_j(\mathbf{f}^n) r_j(\mathbf{f}^n) \frac{\partial E}{\partial f_j}(\mathbf{f}^n)). \end{aligned}$$

Therefore, by defining

$$s_{max}^n = min_j(|r_j(\mathbf{f}^n)\tau_j(\mathbf{f}^n)\frac{\partial E}{\partial f_j}(\mathbf{f}^n)|)^{-1}, \tag{3–12}$$

it can be seen that if $s^n < s_{max}^n$, then $f_j^{n+1} \geq 0$. If $K < s_{max}^n$, let $s_{max}^n = K$. Now a search is made for a step-size coefficient within the bounding value that decreases the algorithm. If $P(s)$ is defined to be $E(\mathbf{f}^{n+1}(s)) = E(\mathbf{f}^n + s\tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n))$, then the algorithm will decrease the objective function if the constant s^n satisfies $P'(s^n) < 0$. Using the definition of $P(s)$,

$$\begin{aligned} P'(s) &= \nabla E(\mathbf{f}^{n+1}(s)) \cdot \frac{d\mathbf{f}^{n+1}}{ds} \\ &= -\sum_j \frac{\partial E}{\partial f_j}(\mathbf{f}^{n+1}(s)) \tau_j(\mathbf{f}^n) f_j^n r_j(\mathbf{f}^n) \frac{\partial E}{\partial f_j}(\mathbf{f}^n). \end{aligned}$$

In order to find a constant s^n to produce a negative value, a search algorithm is employed. One possible choice is the bisection rule, which produces a result within a specified precision. The algorithm proceeded with this choice, but with slow computation speeds as it requires multiple calculations of $P'(s)$. In order to decrease the iteration time, the Armijo rule [1] was employed. The Armijo rule is an inexact search algorithm, here used to approximate where $P'(s) = 0$. The value returned by the Armijo rule will be a constant $\mathbf{s}$ that will decrease the objective function as required. To apply the Armijo rule, the search begins with an initial guess (here taken to be the minimum of s_{max} and K). The initial value is decremented this by a constant factor β (taken to be 1/3) until the following relationship holds (δ is a positive constant chosen to be close to 0):

$$P(0) + s\delta P'(0) \geq P(s).$$

The following figure illustrates one application of the Armijo rule and the value that would be returned by the method. The use of the Armijo rule produced much smaller computation times than the bisection method. The former requires the calculation of the derivative only at the origin and thereafter calculates values of $P(s)$ to give a rough approximation to the solution. Even though it does not give the best value for the step-size coefficient, it was seen that the number of iterations needed to reach a stopping point did not significantly increase when the Armijo Rule was applied. Hence, the savings in computation time caused it to be used in the reconstructions.

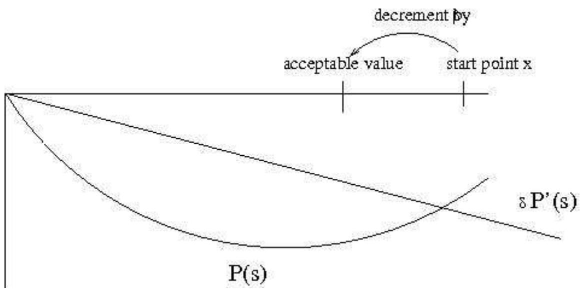


Figure 3–1: Armijo rule

Since the step-size coefficient s^n satisfies $P'(s^n) < 0$, the objective function decreases as needed. The update is then formed by

$$\mathbf{f}^{n+1} = \mathbf{f}^n + s^n \tau(\mathbf{f}^n) \odot \mathbf{v}(\mathbf{f}^n)$$

and the iteration proceeds. It remains to be shown that this choice of step-size vector will satisfy the required conditions for convergence. By construction, $(s\tau_j(\mathbf{f})) < K$. By specifying that $(s\tau_j(\mathbf{f})) < s_{max}$, (3–3) is met. $(s\tau)$ decreases the objective function, satisfying (3–4). Finally, by construction, $\tau_j(\mathbf{f})$ and $r_j(\mathbf{f})$ have the same sign, proving (3–5). Since conditions (3–2 - 3–5) have been satisfied, the iteration will converge as proven in Theorem 1.

3.3.3 Stopping Criteria

The algorithm should terminate when the Kuhn-Tucker optimality conditions (1–9) are realized. The projected gradient function [1] can be used to accomplish this. This function is defined by

$$pgd(\mathbf{f}) = |max(f_j - \frac{\partial E}{\partial f_j}(\mathbf{f})), 0) - f_j|_\infty. \tag{3–13}$$

Theorem 5. *$pgd(\mathbf{f}) = 0$ if and only if the Kuhn-Tucker optimality conditions are met.*

Assume the Kuhn-Tucker conditions are not met. If the first condition is not true, then $\frac{\partial E}{\partial f_j}(\mathbf{f}) < 0$, resulting in $pgd(\mathbf{f}) > 0$. If the second condition is not met, f_j and $\frac{\partial E}{\partial f_j}(\mathbf{f})$ are nonzero for some j. If $f_j \geq |\frac{\partial E}{\partial f_j}(\mathbf{f})|$, $pgd(\mathbf{f}) \geq |\frac{\partial E}{\partial f_j}(\mathbf{f})| > 0$. If the other inequality holds, then $pgd(\mathbf{f}) \geq f_j \geq 0$. Thus if the Kuhn-Tucker conditions do not hold, the projected gradient function is nonzero. Now assume the Kuhn-Tucker conditions are met. For each j, $\frac{\partial E}{\partial f_j}(\mathbf{f}) > 0$ and at least one of f_j and $\frac{\partial E}{\partial f_j}(\mathbf{f})$ are zero. If $f_j \geq \frac{\partial E}{\partial f_j}(\mathbf{f})$, then $\frac{\partial E}{\partial f_j}(\mathbf{f}) = 0$ and $|max(f_j - \frac{\partial E}{\partial f_j}(\mathbf{f}), 0) - f_j| = \frac{\partial E}{\partial f_j}(\mathbf{f}) = 0$. If $\frac{\partial E}{\partial f_j}(\mathbf{f}) \geq f_j$, then $f_j = 0$ and $|max(f_j - \frac{\partial E}{\partial f_j}(\mathbf{f}), 0) - f_j| = f_j = 0$.

Given some choice of small $\epsilon > 0$, the algorithm can be terminated when $pgd(\mathbf{f}) < \epsilon$.

[1] as suggested by Prof. Hager

CHAPTER 4
MULTIPLE IMAGE-MOTION RECONSTRUCTION

4.1 Algorithm Desription

4.1.1 RM Algorithm

The Penalized Maximum Likelihood method discussed in Chapter 3 can be extended for use in the reconstruction of time-delayed images. In this problem, a series of CT-generated images of an object are reconstructed along with any motion of the object between the scans. The RM algorithm, as developed by Gilland and Mair [13] [2] [14], can be used to reconstruct this series. However, its convergence to a minimizer has not been proven. A modification of their work allows for convergence of an altered objective function. As in their work, the algorithm will be described initially in the continuous case, then discretized to produce the iteration. In this reconstruction, a total of K time-lapsed scans (or frames) are analyzed. In the continuous case, the frame number will be denoted in the subscript, $f_k(\mathbf{z})$. In the discrete case, the frame number will be the first subscript, $f_{k,j}$ referring to the jth voxel in frame k. Along with the unknown emission intensities in each frame k, $f_k(\mathbf{w})$, the motion vector, $\mathbf{m}_k(\mathbf{z})$ from frame k to frame $k+1$ is also unknown. This motion vector describes the movement of each point $\mathbf{z}$ in frame k, as $\mathbf{z}$ moves to $\mathbf{z}+\mathbf{m}_k(\mathbf{z})$ in frame $k+1$. In the continuous case, the negative log-likelihood of the reconstruction is given by

$$L(\mathbf{f}) = \sum_{k=0}^{K} \sum_{i} [Pf_k(i) - y_k(i) \log Pf_k(i)] \tag{4–1}$$

where $Pf_k(i)$ is the continuous projection operator given by $Pf_k(i) = \int p_i(\mathbf{z}) f_k(\mathbf{z}) d\mathbf{z}$ and $p_i(\mathbf{z})$ is the probability that an emission at $\mathbf{w}$ will be registered in detector i

(compare with the discrete projection operator given by $Pf_i = \sum_i p_{i,j} f_j$). The penalty term U now measures how well the motion term matches up points between frames. Ideally, $f_k(\mathbf{z}) = f_{k+1}(\mathbf{z} + \mathbf{m}_k(\mathbf{z}))$ but it is to be expected that this will not occur with real data. Hence the image-matching term between frames k and $k+1$ is given by

$$U_k(f_k, f_{k+1}, \mathbf{m}_k) = \int [f_k(\mathbf{z}) - f_{k+1}(\mathbf{z} + \mathbf{m}_k(\mathbf{z}))]^2 d\mathbf{z} \tag{4–2}$$

and the penalty term becomes the sum of these individual terms.

$$U(\mathbf{f}, \mathbf{m}) = \sum_{k=1}^{K} U_k(f_k, f_{k+1}, \mathbf{m}_k) \tag{4–3}$$

In addition, the objective function includes a third term $E_S = \sum_{k=1}^{K} E_{Sk}$ modeling the elastic strain of the material.

$$\begin{aligned} E_{Sk}(\mathbf{m}_k) &= 1/2 \int \lambda(\mathbf{z})(u_{k,x} + v_{k,y} + w_{k,z})^2 d\mathbf{z} \\ &+ \int \mu(\mathbf{z})(u_{k,x}^2 + v_{k,y}^2 + w_{k,z}^2) d\mathbf{z} \\ &+ 1/2 \int \mu(\mathbf{z})[(u_{k,y} + v_{k,x})^2 + (u_{k,x} + w_{k,x})^2 + (v_{k,z} + w_{k,y})^2] d\mathbf{z} \end{aligned}$$

where λ and μ are material parameters,u_k, v_k, w_k are the components of $\mathbf{m}_k$ and $u_{k,x}$ is the partial derivative of u_k with respect to x.

Thus after adding constants to balance the contribution of each term, the continuous objective function becomes

$$E(\mathbf{f}, \mathbf{m}) = \alpha L(\mathbf{f}) + U(\mathbf{f}, \mathbf{m}) + \beta E_S(\mathbf{m}). \tag{4–4}$$

The reconstruction problem thus becomes a minimization of $E(\mathbf{f}, \mathbf{m})$ with the restriction that the intensities in each frame remain positive.

At each iteration of the RM Algorithm (the outer loop), two processes are run in succession - the R-Step and the M-Step. In the R-Step (an inner loop), the previous motion iterate is used to improve the reconstruction of the frames. Since E_S does not depend on $\mathbf{f}$, the R-Step is applied only to $\alpha L(\mathbf{f}) + U(\mathbf{f}, \mathbf{m})$ with the motion vector held constant. A PML reconstruction method can be applied (such as the non-uniform step-size method described Chapter 3) to decrease this term and produce the next iterate. In the M-Step (an inner loop), the motion vector is updated using the frames generated from the R-Step. In this step, the conjugate gradient algorithm is applied to decrease $U(\mathbf{f}, \mathbf{m}) + \beta E_S(\mathbf{m})$ with $\mathbf{f}$ now held constant.

4.1.2 Image-Matching Penalty

In order to run the algorithm, the image space is partitioned into J equally sized voxels. However, after discretizing the image space, it cannot be assumed that the center of a voxel is mapped to the center of another voxel by the motion vector. Thus, it is necessary to distribute the motion over adjoining voxels.
To accomplish this, let $(u_{k,i}, v_{k,i}, w_{k,i})$ be the components of the motion vector mapping voxel i in frame k into frame $k+1$. Now define d_i, d_j and d_k to be the decimal portions of u, v and w respectively. If (a_i, b_i, c_i) are the coordinates of the center of voxel i, then the motion vector will map this voxel to the "voxel" in frame $k+1$ centered at $(a_i + u_{k,i}, b_i + v_{k,i}, c_i + w_{k,i})$. The target voxel to be approximated will lie in at most eight real voxels. Define d_{000} to be the voxel located at $([a_i + u_{k,i}], [b_i + v_{k,i}], [c_i + w_{k,i}])$. Then the target voxel will lie in a cube of dimension 2-voxels with d_{000} in the lower left corner. The emission intensity in the target voxel can then be found by a trilinear approximation using the voxels forming the cube, with the weights proportional to the distance from center of each voxel to the center of the target voxel. Thus, the discrete version of the image-matching term (4–3) is given by:

$$U_k(\mathbf{f}_k, \mathbf{f}_{k+1}, \mathbf{m}_k) = \sum_i (f_{k,i} - \sum_{j \in N_{k,i}} c_{k,i,j} f_{k+1,j})^2 \tag{4–5}$$

where $N_{k,i}$ is the 8-voxel cube formed by $m_{k,i}$ and the weights, $c_{k,i,j}$, are given by Table 4–1. This form for the image-matching term is currently implemented in the objective function used in the RM Algorithm. However, it is difficult to perform the conjugate gradient (M-Step) on this term. it has proven difficult to implement the M-Step on the this term since the neighborhood system $N_{k,i}$ depends on the motion vector. To solve this problem, the M-Step has instead been performed on a linearization of the image-matching term [14] given by

voxel	voxel center	weight
d_{000}	$([a_i + u_{k,i}], [b_i + v_{k,i}], [c_i + w_{k,i}])$	$(1 - d_i)(1 - d_j)(1 - d_k)$
d_{100}	$([a_i + u_{k,i}] + 1, [b_i + v_{k,i}], [c_i + w_{k,i}])$	$d_i(1 - d_j)(1 - d_k)$
d_{010}	$([a_i + u_{k,i}], [b_i + v_{k,i}] + 1, [c_i + w_{k,i}])$	$(1 - d_i)d_j(1 - d_k)$
d_{110}	$([a_i + u_{k,i}] + 1, [b_i + v_{k,i}] + 1, [c_i + w_{k,i}])$	$d_i d_j(1 - d_k)$
d_{001}	$([a_i + u_{k,i}], [b_i + v_{k,i}], [c_i + w_{k,i}] + 1)$	$(1 - d_i)(1 - d_j)d_k$
d_{101}	$([a_i + u_{k,i}] + 1, [b_i + v_{k,i}], [c_i + w_{k,i}] + 1)$	$d_i(1 - d_j)d_k$
d_{011}	$([a_i + u_{k,i}], [b_i + v_{k,i}] + 1, [c_i + w_{k,i}] + 1)$	$(1 - d_i)d_j d_k$
d_{111}	$([a_i + u_{k,i}] + 1, [b_i + v_{k,i}] + 1, [c_i + w_{k,i}] + 1)$	$d_i d_j d_k$

Table 4–1: RM voxel neighborhood

$$Q_k(\mathbf{m}) = \int f_k(\mathbf{w}) - f_{k+1}(\mathbf{w}) - (\mathbf{m}_k(\mathbf{w}) - \tilde{\mathbf{m}}_k(\mathbf{w})) \cdot \nabla \mathbf{f}_{k+1}(\mathbf{w} + \tilde{\mathbf{m}}_k(\mathbf{w}) d\mathbf{w}$$

with the previous motion estimate given by $\tilde{m}$. However, it has not been proven that a decrease in the linearized term Q_k will result in a decrease in the original image-matching term U_k as required. One possible way to solve this problem is to perform the linearization at the objective function level rather than at each inner loop. Hence, the original objective function is modified to replace U_k with a linearized version $\tilde{U}_k$. With this change, the algorithm will now be referred to as the modified RM algorithm. Since the objective functions are different, a minimizer for the modified RM algorithm is not expected to be a minimizer for the original RM algorithm. However, a decrease in the modified objective function can be proven, allowing for convergence results. To describe the modified objective function, define the term

$$\begin{aligned}\tilde{U} &= \sum_{k=1}^{K} \tilde{U}_k(f_k, f_{k+1}, \mathbf{m}_k) \\ \tilde{U}_k &= \int (f_k(\mathbf{z}) - f_{k+1}(\mathbf{z}) - \mathbf{m}_k(\mathbf{z})) \cdot \nabla f_{k+1}(\mathbf{z}))^2 d\mathbf{z} \qquad (4\text{–}6)\end{aligned}$$

and update the objective function to become

$$\tilde{E}(\mathbf{f}, \mathbf{m}) = \alpha L(\mathbf{f}) + \tilde{U}(\mathbf{f}, \mathbf{m}) + \beta E_S(\mathbf{m}). \qquad (4\text{–}7)$$

In addition, linearizing the image-matching term gains convexity in the objective function (the RM image-matching term is not convex in $\mathbf{m}$). However, this linearization is valid only for small displacements. If the motion between the frames is large, the reconstruction may not be valid. Using this modified objective function, the two inner loop sub-processes can now be described in detail.

4.1.3 Modified RM Iteration

The R-Step consists of minimizing $\alpha L(\mathbf{f}) + \tilde{U}(\mathbf{f}, \mathbf{m})$ over the set of positive intensities $\mathbf{f}$. As this in the same form as the objective function in Chapter 3, the non-uniform step-size algorithm (3–1) can be utilized. Thus, the continuous R-Step update for each frame k is given by

$$\begin{aligned}f_k^{n+1} &= f_k^n + s\mathbf{v}(\mathbf{f}^n) \qquad (4\text{–}8) \\ &= f_k^n - s f_k^n r_k(\mathbf{f}^n) D_k E(\mathbf{f}, \mathbf{m})\end{aligned}$$

with $r_k(\mathbf{f})(\mathbf{z}) = (\alpha \sum_i p_i(\mathbf{z}) + D_k U(\mathbf{f},\mathbf{m}))^{-1}$. As derived in the Appendix of [14], the derivatives $D_k E$ (taken with respect to $\mathbf{f}_k$) are given by

$$\begin{aligned}
D_k E(\mathbf{f},\mathbf{m}) &= \alpha \sum_i (p_i(\mathbf{z}) - \frac{y_k(i) p_i(\mathbf{z})}{P f_k(i)}) + D_k \tilde{U}(\mathbf{f},\mathbf{m}) \\
D_k \tilde{U}(\mathbf{f},\mathbf{m}) &= D_k \tilde{U}_k(f_k, f_{k+1}, \mathbf{m}_k) + D_k \tilde{U}_{k-1}(f_{k-1}, f_k, \mathbf{m}_{k-1}) \\
D_k \tilde{U}_k(f_k, f_{k+1}, \mathbf{m}_k) &= 2(f_k - f_{k+1} - \mathbf{m}_k \cdot \nabla f_{k+1}) \\
D_k \tilde{U}_{k-1}(f_{k-1}, f_k, \mathbf{m}_{k-1}) &= -2(f_{k-1} - f_k - \mathbf{m}_{k-1} \cdot \nabla f_k) \\
&\quad +2(\nabla f_{k-1} - \nabla f_k - \nabla(\mathbf{m}_k \cdot \nabla f_k)) \cdot \mathbf{m}_{k-1} \\
&\quad +2(f_{k-1} - f_k - \mathbf{m}_{k-1} \cdot \nabla f_k)(\nabla \cdot \mathbf{m}_{k-1}).
\end{aligned}$$

The last derivative can be seen (in the 2-frame case without loss of generality) by using the formula

$$D_2 \tilde{U}_1(f_1, f_2, \mathbf{m}_1) = \frac{d}{dt} E(f_1(\mathbf{z}), f_2(\mathbf{z}) + t h(\mathbf{z}), \mathbf{m}_1(\mathbf{z}))|_{t=0} \tag{4–9}$$

to obtain

$$\begin{aligned}
D_2 \tilde{U}_1(f_1, f_2, \mathbf{m}_1) &= -2 \int (f_1(\mathbf{z}) - f_2(\mathbf{z}) - \mathbf{m}_1(\mathbf{z}) \cdot \nabla f_2(\mathbf{z})) h(\mathbf{z}) d\mathbf{z} \\
&\quad -2 \int (f_1(\mathbf{z}) - f_2(\mathbf{z}) - \mathbf{m}_1(\mathbf{z}) \cdot \nabla f_2(\mathbf{z}))(\mathbf{m}_1(\mathbf{z}) \cdot \nabla h(\mathbf{w})) d\mathbf{z}.
\end{aligned}$$

$D_2 \tilde{U}_1(f_1, f_2, \mathbf{m}_1) = \psi$ if equation (4–9) can be written as $\int \psi(\mathbf{z}) h(\mathbf{z}) d\mathbf{z}$. As the first term is already in this form, only the second needs to be simplified. This can be done by applying the identity

$$\int g\mathbf{m} \cdot \nabla h(\mathbf{z}) = -\int (\nabla \cdot g\mathbf{m}) h(\mathbf{z}) = -\int (\nabla g \cdot \mathbf{m}) h(\mathbf{z}) + (g \nabla \cdot \mathbf{m}) h(\mathbf{z})$$

to yield the equality for the second term:

$$\begin{aligned}
&= 2 \int ((\nabla(f_1(\mathbf{z}) - f_2(\mathbf{z}) - \mathbf{m}_1(\mathbf{z}) \cdot \nabla f_2(\mathbf{z})) \cdot \mathbf{m}_1) h(\mathbf{z}) d\mathbf{z} \\
&\quad +2 \int (f_1(\mathbf{z}) - f_2(\mathbf{z}) - \mathbf{m}_1(\mathbf{z}) \cdot \nabla f_2(\mathbf{z}))(\nabla \cdot \mathbf{m}_1) h(\mathbf{z}) d\mathbf{z}.
\end{aligned}$$

Distributing the ∇ in the first integral and adding the results to the first term yields the desired derivative.

After discretizing the voxels (and approximating derivatives with central differences), (4–8) reduces to the discrete iteration:

$$f_{k,j}^{n+1} = f_{k,j}^{n} + s_{k,j}^{n} v_{k,j} \tag{4–10}$$

where $v_{k,j} = -f_{k,j} r_{k,j}(\mathbf{f}) \frac{\partial \tilde{E}}{\partial f_{k,j}}(\mathbf{f})$ and $r_{k,j}(\mathbf{f}) = (\alpha \sum_i p_{k,i,j} + D_k \tilde{U}(\mathbf{f}, \mathbf{m}))^{-1}$. As before, the step-size $\mathbf{s}^n$ vector must be chosen so that (3–2 - 3–5) are satisfied. The method described in Chapter 3 can be used to find such a vector.

Once the emission intensity has been updated by the R-Step, the M-Step is run to update the motion vector. This step consists of minimizing $\tilde{U}(\mathbf{f}, \mathbf{m}) + \beta E_S(\mathbf{m})$ over all possible motion vectors $\mathbf{m}$. The conjugate gradient algorithm (using the Polak-Ribiere variant) is then applied to the discretized voxels to update the motion vector [14].

In practice, at each iteration of the RM algorithm (the outer loop), one iteration each of the R-Step and M-Step occurs (the inner loop). Continue running the outer loop under a stopping criteria has been reached, such as the projected gradient (3–13) falling under a chosen critical value.

4.2 Convergence Results

Although the new objective function $\tilde{E} = \alpha L + \tilde{U} + \beta E_S$ is convex in $\mathbf{f}$, Lemma 4 does not apply in this situation (unless a single motion vector is fixed). Hence, convergence to a single minimizer is not guaranteed.

Theorem 6. *If $\{\mathbf{f}^n\}$ is a sequence of iterates produced by (4–10), then each limit point of the sequence is a minimizer for the objective function $\tilde{E}$ and $\tilde{E}(\mathbf{f}^n)$ converges.*

As before, Zangwill's Convergence Theorem is applied. The definition of the point-to-set function T is expanded to the following:

$$T_1(\mathbf{f},\mathbf{m}) = \{(\mathbf{g},\mathbf{m}_{CG})|\mathbf{g} \in T(\mathbf{f})\} \tag{4–11}$$

where T is the previous point-to-set function defined in (3–7) (with a range of possible step-size vectors) and $\mathbf{m}_{CG}$ is the motion vector outputted from applying the conjugate gradient algorithm to $\tilde{U}(\mathbf{g},\mathbf{m}) + \beta E_S(\mathbf{m})$. Let $\tilde{E}$ be the linearized objective function and define the set of stationary points S to be the set of all points $(\mathbf{f},\mathbf{m})$ with $\mathbf{v}(\mathbf{f}) = \mathbf{0}$. If $f \in S$, then $T(\mathbf{f}) = \{\mathbf{f}\}$ and the R-Step will not change the value of $L + \gamma\tilde{U}$. The M-Step (which does not affect the reconstructed image $\mathbf{f}$) has been defined to decrease $\gamma\tilde{U} + E_S$. Thus $\tilde{E}(T_1(\mathbf{f},\mathbf{m})) \leq \tilde{E}(\mathbf{f},\mathbf{m})$ if $\mathbf{f} \in S$. If $f \notin S$, then the R-Step will decrease $L + \gamma\tilde{U}$ as desired. Following this, the M-Step will decrease the value of $\gamma\tilde{U} + E_S$. Hence $\tilde{E}(T_1(\mathbf{f},\mathbf{m})) < \tilde{E}(\mathbf{f},\mathbf{m})$ if $\mathbf{f} \notin S$. The closure condition is proved by noting that T_1 can be split into its two parts, thus allowing the critical closure condition on f to be proved as in Lemma 1.

Thus by Zangwill, all limit points of the sequence $\{(\mathbf{f}^n,\mathbf{m}^n)\}$ lie in S and $\tilde{E}(\mathbf{f}^n,\mathbf{m}^n)$ will monotonically converge to $\tilde{E}(\mathbf{y},\mathbf{m})$ for some $(\mathbf{y},\mathbf{m}) \in S$.

Now to prove all limit points are minimizers, let y be a limit point for the algorithm. Consider the subsequence $\mathbf{f}^{n_j}$ where for each j, $||\mathbf{f}^{n_j} - \mathbf{y}|| < 1/j$ (the definition of a limit point gives the existence of such points). Now as in Theorem 4, assume $\mathbf{y}$ does not meet the first K-T condition for a minimizer (By Zangwill, $\mathbf{y} \in S$, hence the second K-T condition minimizer of E is met). Therefore, an index k can be found such that $y_k = 0$ and $\frac{\partial E}{\partial f_k}(\mathbf{y}) < 0$ and hence there is an J such that $\frac{\partial E}{\partial f_j}(\mathbf{f}^{n_j}) < 0$ for all $j > J$. By construction, $r_k(f^{n_j})$ and s_k have the same sign and $f_k^{n_j} > 0$. Thus for $j > J$,

$$f_k^{n_{j+1}} = f_k^{n_j} + s_k^{n_j} v_k(\mathbf{f}^{n_j}) = f_k^{n_j} - s_k^{n_j} f_k^{n_j} r_k(\mathbf{f}^{n_j}) \frac{\partial E}{\partial f_k}(\mathbf{f}^{n_j}) > f_k^{n_j}$$

and $f_k^{n_j}$ does not converge to $\mathbf{y}_k = 0$, contrary to the assumption. Therefore, both K-T conditions hold at the limit point.

If the modified M-Step is terminated after iteration K (locking in a specific value for the motion vector) and the modified R-Step is used exclusively afterwards, the convergence proof for penalized maximum likelihood presented in Chapter 3 will suffice to give convergence for the modified RM algorithm.

CHAPTER 5
RECONSTRUCTIONS

5.1 PML Reconstructions

To test the non-uniform step-size algorithm presented in Chapter 3, a series of reconstructions was performed on a phantom source image [1] , $\lambda = [\lambda_1, ..., \lambda_J]$, of dimension J=128x128. As displayed below, the phantom image is that of a typical cardiac scan, with lungs, heart, spine and other tissues. Higher intensity values correspond to regions of greater metabolic activity (in particular the heart and lung regions). The detector data, $\mathbf{Y} = [y_1, ..., y_I]$, was created by calculating the probability matrix P using (1–2). The chest emission was then projected into the detector space using $y_i = \sum_j p_{i,j}\lambda_j$. Finally, random Poisson noise was added to the

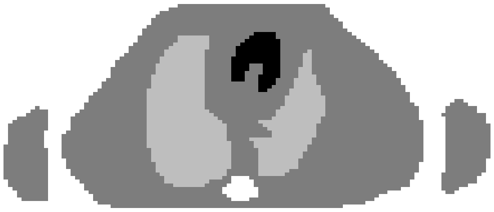


Figure 5–1: Phantom chest source image

[1] created by Dr.Anderson

data. The PML objective function is formed by the sum

$$E(\mathbf{f}) = L(\mathbf{f}) + \gamma U(\mathbf{f}).$$

The penalty term, $U(\mathbf{f})$ was formed using the logcosh function with δ taken to be 50.

$$U(\mathbf{f}) = \sum_i \sum_{j \in N_i} w_{i,j} \log\cosh \frac{f_i - f_j}{\delta} \tag{5–1}$$

The corresponding neighborhood system, N_i is formed by the 8 closest voxels, with the diagonal elements having a weight of $\frac{1}{\sqrt{2}}$, and the 4 axial neighbors a weight of 1, as seen in Figure 1–3. The strength of the penalty term, γ, ranged over a set of values to provide a comparison. The reconstructions were formed by the non-uniform step-size method (3–1). In separate trials, the search for the step-size coefficient, s, was performed with the Armijo Rule and the bisection method. For each value of γ, the algorithm was run until the projected gradient (3–13) fell under 10^{-2}. The resulting reconstructions were then compared to the source image with a root-mean-square calculation

$$\sqrt{\frac{\sum_j (f_j - \lambda_j)^2}{128 * 128}}$$

to determine optimal values to use for the penalty strength.

As a comparison, the same phantom scanner data was used in the APML algorithm (2–7). The reconstructions outputted from APML were visually identical to the reconstructions gained using the non-uniform step-size algorithm. In all trials, the objective function decreased rapidly, then leveled off. It was seen that the image quality did not improve significantly after this leveling off of the objective function. As can be seen in Figure 5–2, initially the objective function decreases faster with the non-uniform step method than with APML, but eventually both fall to the same level. The computation times for APML were approximately the same as the computation time using the non-uniform

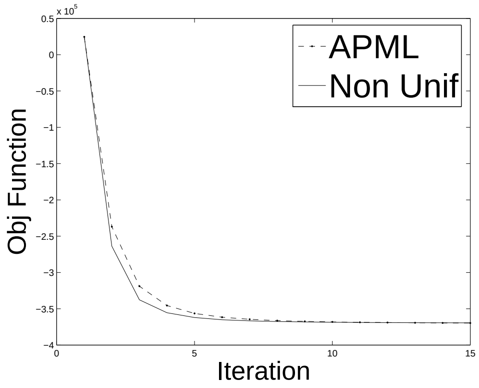


Figure 5–2: Comparing the decrease in the objective function

step-size method with the Armijo Rule. The APML algorithm was able to reduce the projected gradient factor in less iterations than with the non-uniform step-size method.

Table 5–1 gives both the RMS error and the iterations required for algorithm termination, occuring when the projected gradient fell below 10^{-2}. In Figure 5–3, the RMS error is plotted against the penalty paramter γ. Reconstruction using the non-uniform step-size algorithm (applying either bisection or Armijo in the line search produced an identical RMS error) produced a smaller RMS error than reconstruction of the image than using the APML reconstruction method. Thus, these methods were able to produce a reconstruction closer to the source phantom image. Due to its inexact search, the Armijo rule reduced the projected gradient

γ	Method	Iterations	CPU time	RMS error
0.015	Bisect	219	475	49.5
0.020	Bisect	227	492	46.5
0.025	Bisect	235	631	45.5
0.030	Bisect	183	530	45.4
0.015	Armijo	220	118	49.6
0.020	Armijo	228	125	46.4
0.025	Armijo	268	129	45.5
0.030	Armijo	229	100	45.4

Table 5–1: PML reconstruction data

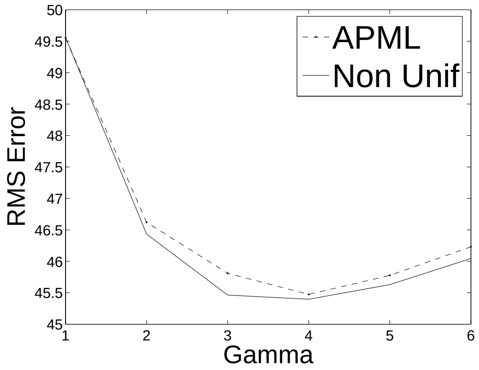
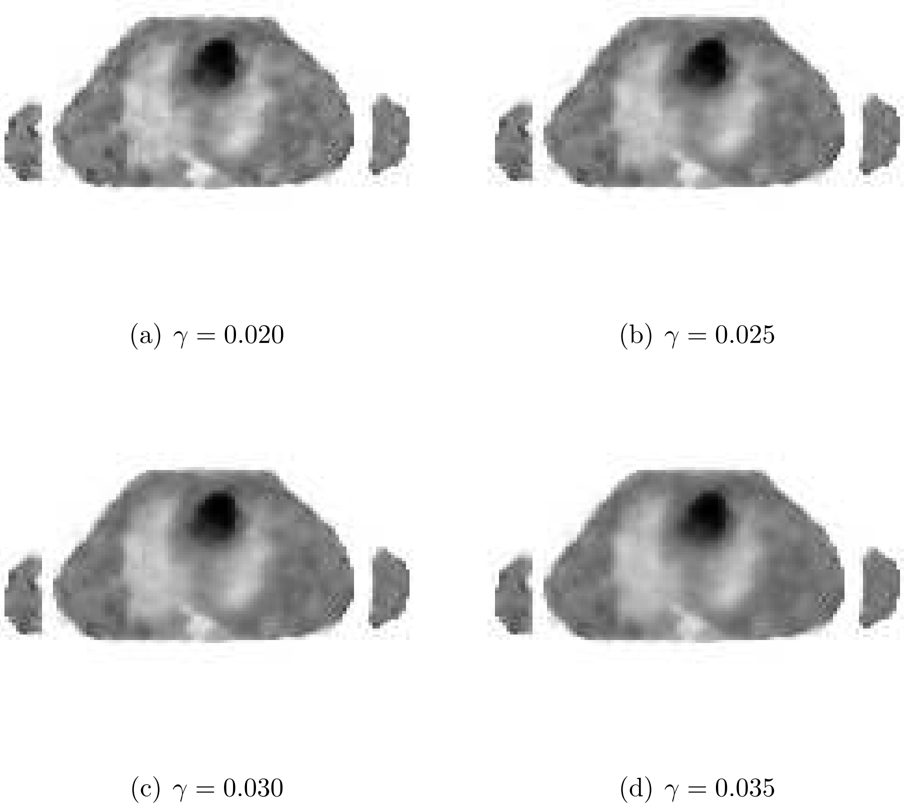


Figure 5–3: Comparing the root-mean-square (RMS) error

criteria in a significantly shorter cputime than the bisection method (although the iterations were approximately the same). The RMS plot in Figure 5–3 shows that a penalty parameter taken from [0.03,0.04] resulted in the closest reconstruction to the original phantom.

The reconstructions seen in Figure 5–4 were formed by applying the bisection method. Although the iteration time was longer than using the Armijo method, the bisection method gave a visually better image with higher values of the penalty parameter, γ. The next set of reconstructions found in Figure 5–5 were formed by applying the Armijo Rule. In this case, as γ increased, the Armijo rule did not perform as well in high-intensity regions. This can be observed in the heart region of the later reconstructions. Figure 5–6 displays a plot formed by a vertical line thru the heart region. In this region, the exact bisection rule was able to produce a smoother image in the high-intensity heart region than the non-exact Armijo rule. Note that both methods produced similar intensities except in the heart region. With further iterations, it was seen that the non-smooth intensity in the heart region was reduced (using a smaller value for the projected gradient cutoff would accomplish this). A tradeoff has thus been made between reconstruction speed (the Armijo rule) and accuracy in the step-size choice (bisection). One possible solution



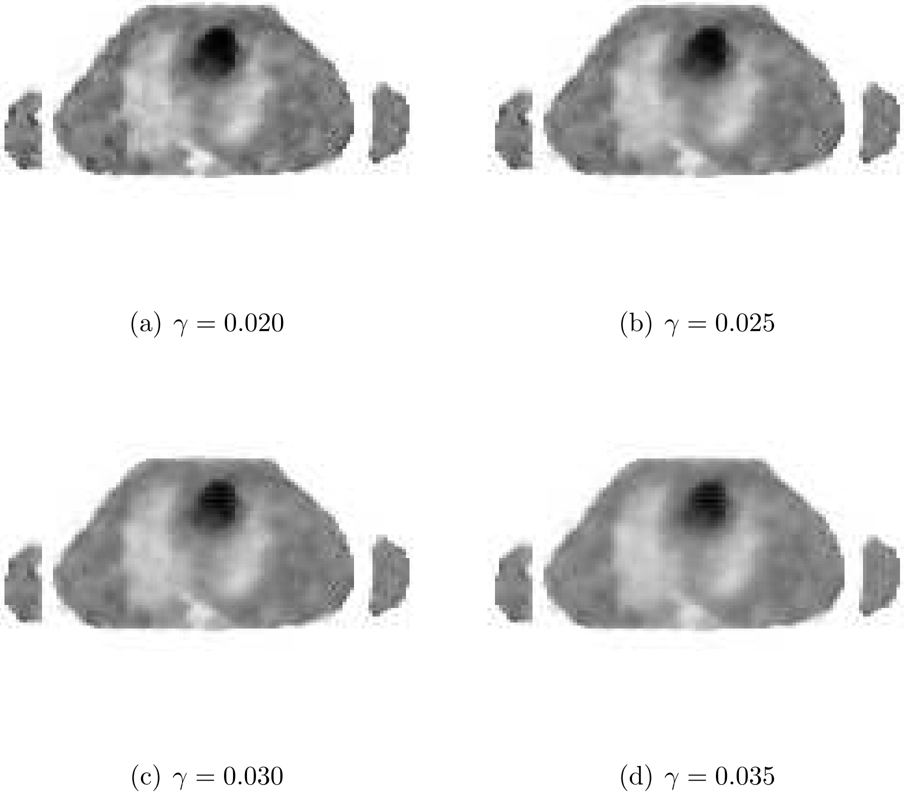
(a) $\gamma = 0.020$

(b) $\gamma = 0.025$

(c) $\gamma = 0.030$

(d) $\gamma = 0.035$

Figure 5–4: Phantom image reconstruction using bisection rule



(a) $\gamma = 0.020$

(b) $\gamma = 0.025$

(c) $\gamma = 0.030$

(d) $\gamma = 0.035$

Figure 5–5: Phantom image reconstruction using Armijo rule

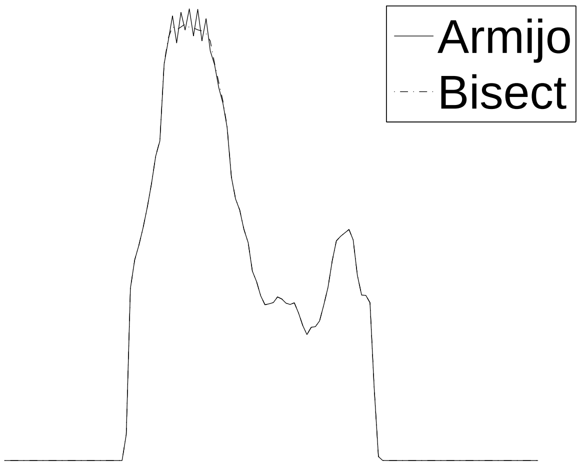
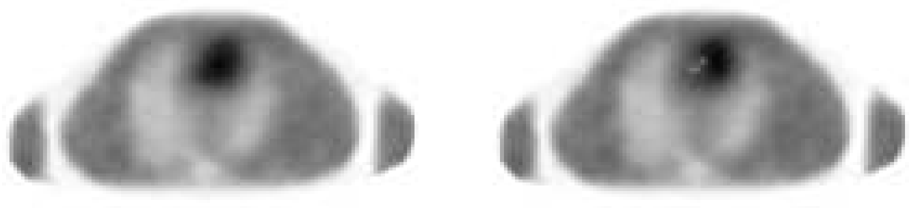


Figure 5–6: Line plot of heart region

is to combine these methods - begin with the Armijo method to approximate the solution, then use bisection for precision.

Originally, the new algorithm was designed to remove the restrictions on the penalty function imposed by Lange's algorithm (2–4). Figure 5–7 shows two reconstructions of the chest, the first using the non-uniform step-size method and the second using Lange's algorithm (and the Armijo Rule). In this case, a quadratic penalty was used in place of the logcosh function. With a penalty parameter of $\gamma = 0.03$, the $\mathbf{r}(f)$ function becomes negative (a situation not covered in Lange's algorithm). As can be seen from the reconstructions, the non-uniform step-size method is able to reconstruct the image correctly, but Lange's algorithm (together with the Armijo Rule) produces artifacts.



(a) Non-uniform step-size method (b) Lange's method

Figure 5–7: Comparing reconstruction algorithms

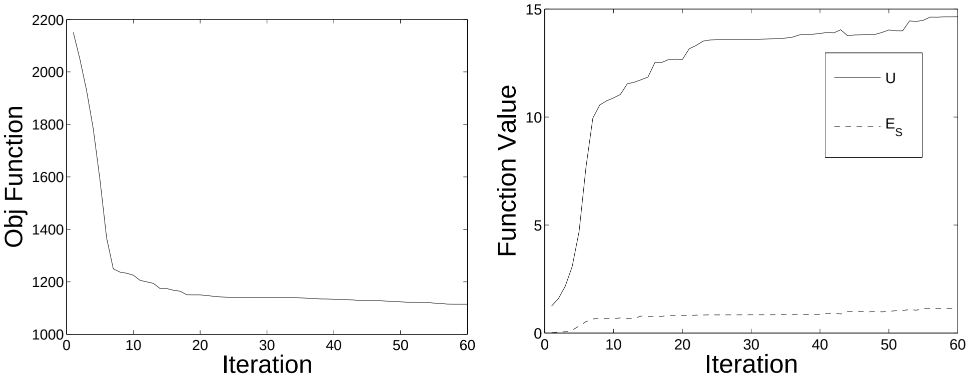
5.2 Modified RM Reconstructions

The modified RM algorithm presented in Chapter 4 was tested on a 2-frame NCAT phantom image of a heart. The algorithm reconstructs both images with the assistance of the motion vector between the frames. The objective function used in the modified RM algorithm is given by

$$\tilde{E}(\mathbf{f}, \mathbf{m}) = \alpha L(\mathbf{f}) + \tilde{U}(\mathbf{f}, \mathbf{m}) + \beta E_S(\mathbf{m})$$

with the constants taken to be $\alpha = 0.02$ and $\beta = 0.005$ as in previous reconstructions. The linearized penalty term $\tilde{U}$ was used in the reconstruction instead of the virtual voxel method used in the RM algorithm of Mair and Gilland [14]. Although the use of the virtual voxels was seen to produce useful reconstructions, the convergence of the RM objective function was unproven (but was observed numerically).

In addition, the use of the linearized penalty produced a decreasing objective function as seen in Figure 5–8. The second plot in Figure 5–8 shows the functions involved in the M-Step, $\tilde{U}$ and βE_S. As can be seen, both $\tilde{U}$ and βE_S are



(a) Modified RM objective function

(b) $\tilde{U}$ and βE_S

Figure 5–8: Modified RM function plots

increasing for each outer loop. Despite this, the M-Step is reducing their sum for each M-Step as can be seen in the sample RM data found in Table 5–2. Typically, the R-step increases the value of $\tilde{U}$ while reducing αL. Then the M-Step decreases $\tilde{U}$ and increases βE_S to give a decrease in the objective function. The reconstructions formed using the modified RM algorithm (with the linearized penalty, $\tilde{U}$), were comparable to those outputted using the original RM algorithm (using the approximated voxels penalty method (4–1)). Figure 5–9 displays a quiver plot of the reconstructed Slice 20 combined with the motion vector. Figure 5–10 displays a comparison of the methods. The top reconstructions were

Iteration	αL	$\tilde{U}$	βE_S	Objective
1 - R-Step	2866.4	1.326		2867.76
1 - M-Step		1.262	0.0203	2867.72
2 - R-Step	2728.4	1.678		2730.11
2 - M-Step		1.673	0.0228	2730.11
3 - R-Step	2569.1	2.341		2571.48
3 - M-Step		2.232	0.0750	2571.42
4 - R-Step	2374.0	3.367		2377.44
4 - M-Step		3.315	0.104	2377.42
5 - R-Step	2116.4	5.638		2122.11
5 - M-Step		5.330	0.277	2121.97

Table 5–2: RM data $\alpha = 0.02$, $\beta = 0.005$

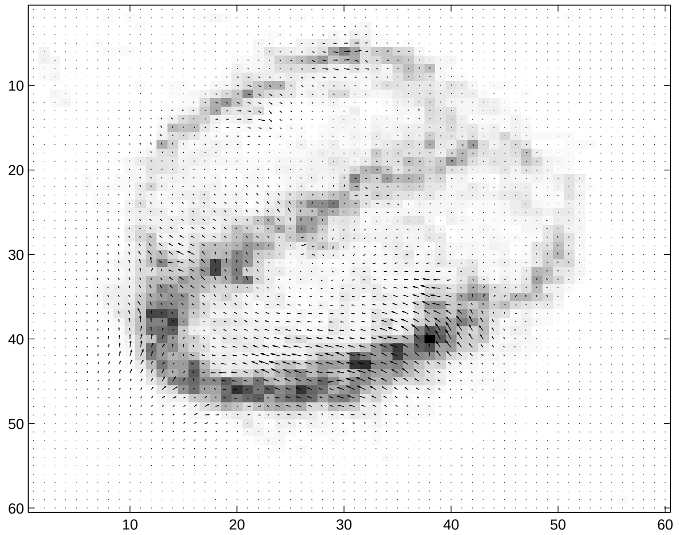
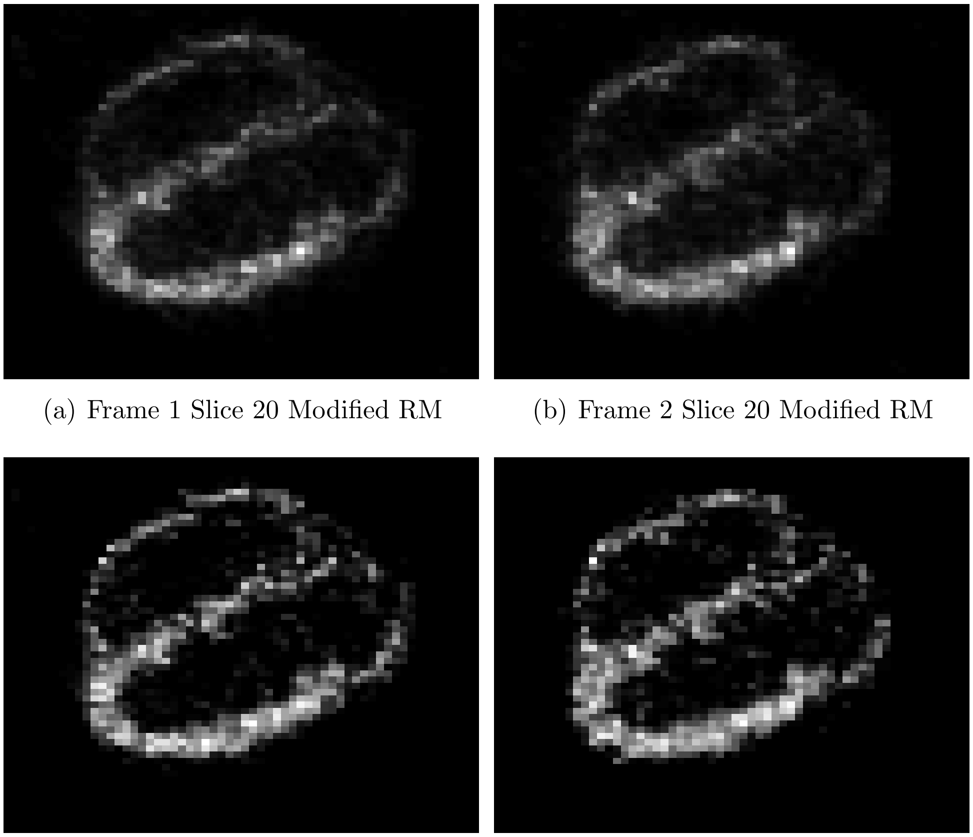


Figure 5–9: Quiver plot of Slice 20

performed with the modified RM algorithm. The lower plots were reconstructed using the original RM algorithm, both using the parameters α 0.02 and $\beta = 0.005$. Although the original RM method gives a better estimate of the motion penalty, a decrease in the objective function (and hence convergence) remains unproven. The use of the linearized penalty in the modified RM algorithm produces a similar reconstruction, while in addition giving the needed decrease in the objective function.



(a) Frame 1 Slice 20 Modified RM

(b) Frame 2 Slice 20 Modified RM

(c) Frame 1 Slice 20 Original RM

(d) Frame 2 Slice 20 Original RM

Figure 5–10: RM reconstructions

5.3 Conclusions

Several methods exist to solve the problem of image recovery in Emission Tomography. Green's OSL algorithm as extended by Lange provides for convergence in a reduced set of penalty functions (which do not include the image-matching penalty used in the RM algorithm). Although Lange's algorithm can be employed in larger settings, neither its convergence to a minimizer nor a decrease in the objective function is guaranteed. If the step-size is allowed to become negative [2], a decrease in the objective function and algorithm convergence is once again guaranteed, but this minimizer may not satisfy the first Kuhn-Tucker optimality condition. However, a further extrapolation to a step-size vector will both guarantee convergence to a minimizer and allow for a larger set of penalty functions. Unlike previous methods, the convergence proof of the generalized method allows for the use of inexact faster line searches, such as the Armijo Rule. The non-uniform step-size algorithm provides one way of constructing a step-size vector that meets the conditions required for convergence. This generalized method can be extended to other image recovery problems, such as multiple image-motion recovery. In this case, the generalized algorithm is incorporated into the modified RM algorithm for reconstruction. Future work in this area will concentrate on refining the algorithm to increase speed without losing image quality.

APPENDIX
SAMPLE ALGORITHM FORTRAN CODE

This Fortran code can be utilized to run the non-uniform step method as described in Chapter 3. Code to calculate the penalty term (and its derivative) is required. The probability matrix $p_{i,j}$ is stored in a sparce matrix format consisting of hrow, hcol and hv preserving the non-zero entries of p.

The iteration loop consists of the following code:

```
do j=1,nhdim

   temp = hcol(j)

   hsum(temp) = hsum(temp) + hv(j)

enddo

  do j=1,nhdim

     temp1 = hrow(j)

     temp2 = hcol(j)

     AX(temp1) = AX(temp1) + hv(j)*f(temp2)

  enddo

  do j=1,nhdim

     temp1 = hrow(j)

     temp2 = hcol(j)

     if(AX(temp1).ne.0)then

        AY(temp2) =  AY(temp2) + hv(j)*Y(temp1)/AX(temp1)

     endif

  enddo

  call findderivpenalty(f,dU)

  do j=1,NVOXX
```

```
   dE(j) = hsum(j) - AY(j) + gamma*dU(j)
enddo


tau(1) = 0
tau(2) = 0
do j=1,NVOXX
   r(j) = sA(j) + gamma*dU(j)
   f1(j) = f(j)*AY(j)/r(j)
   v(j) = f1(j) - f(j)
enddo


temp1 = 0
do j=1,NVOXX
   if(r(j).le.0)temp1 = 1
enddo
if(temp1.eq.1)then
   do j=1,NVOXX
      if(r(j).ge.0.)then
         tau(1)=tau(1)+(f(j)*dE(j)**2)/r(j)
      else
         tau(2)=tau(2)+(f(j)*dE(j)**2)/r(j)
      endif
   enddo
else
   tau(1) = 1
endif
tau(3) = sqrt(tau(1)**2 + tau(2)**2)
tau(1) = tau(1) / tau(3)

tau(2) = tau(2) / tau(3)

do j=1,NVOXX

   if (r(j).ge.0.)then

      v(j) = v(j)*tau(1)

   else

      v(j) = v(j)*tau(2)

   endif

enddo

smax=0

do j=1,NVOXX

   if(r(j).ge.0)then

      smax = max(smax, tau(1)*dE(j)/r(j))

   else

      smax = max(smax, tau(2)*dE(j)/r(j))

   endif

enddo

if(smax.eq.0.)then

   smax = 1

else

   smax = 1/smax

endif


sopt =find-s(f,smax,v,hcol,hrow, hv,Y,dE,gamma)

do j=1,NVOXX

   f(j) = f(j) + sopt*v(j)

enddo
```

Utilizing the Armijo rule in the line-search can be accomplished by the code (called by find-s in the main loop):

```
      dEdotv = 0
      do j=1,NVOXX
         dEdotv = dEdotv + dE(j)*v(j)
      enddo
      tol = 0.1
      k = 0
      rho = 3
 30   continue
      alpha = smax / (rho**k)
      do j=1,NVOXX
         f1(j) = f(j) + alpha*v(j)
      enddo
      do j=1,nhdim
         temp1 = hrow(j)
         temp2 = hcol(j)
         AX(temp1) = AX(temp1) + hv(j)*f1(temp2)
      enddo
      call findpenalty(f1,U)

      E = 0
      do j=1,ndet
         if(AX(j).ne.0)then
            E = E  + AX(j) - Y(j)*log(AX(j))
         endif
      enddo
```

```
do j=1,NVOXX

   E = E + gamma*U(j)

enddo

flag = E0 + alpha * tol * dEdotv

k = k+1

if(E.gt.flag)goto 30


find_s = alpha
```

REFERENCES

[1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, 1993. *Nonlinear programming: theory and algorithms.* New York : J. Wiley & Sons, 2nd edition.

[2] Z. Cao, D. Gilland, B. Mair, and R. Jaszczak. Three-dimensional motion estimation with image reconstruction for gated cadiac ECT. *IEEE Trans. Nuclear Science*, pages 384–388.

[3] R. B. Carroll. *An orthogonal series approach to positron emission tomography.* PhD thesis, University of Florida, Gainesville, FL, October 1998.

[4] J. Chang, J.M.M. Anderson, and B.Mair. An accelerated penalized maximum likelihood algorithm for positron emission tomography. *IEEE Trans. Image Processing*, forthcoming, 2006.

[5] J. Chang, J.M.M. Anderson, and J. Votaw. Regularized image reconstruction algorithms for positron emission tomography. *IEEE Trans. Med. Imag.*, 23(9):1165–1175, 2004.

[6] A.R. DePierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Trans. Med. Imag.*, 14(1):132–137, 1995.

[7] S. Geman and D. McClure. Bayesian image analysis: an application to single photon emission tomography. *Proc. American Statistical Society*, pages 12–18, 1985.

[8] P. Green. On use of the EM algorithm for penalized likelihood estimation. *J.R. Statist. Soc*, 52(3):443–452, 1990.

[9] D. Lalush and B. Tsui. A generalized Gibbs prior for maximum a posterioro reconstruction in SPECT. *Phys. Med. Biol.*, 38:729–741, 1993.

[10] K. Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *IEEE Trans. Med. Imag.*, 9(4):439–446, 1990.

[11] D. Luenberger, 1973. *Introduction to linear and nonlinear programming.* Reading : Addison-Wesley.

[12] B. Mair. A mathematical model incorporating the effects of detector width in 2D PET. *Inverse Problems*, 16:223–224, 2000.

[13] B. Mair, D. Gilland, and Z. Cao. Simultaneous motion estimation and image reconstruction from gated data. *Proc. 2002 IEEE Internat. Symp. Biomed. Imaging*, pages 661–664, 2002.

[14] B. Mair, D. Gilland, and J.Sun. Estimation of images and nonrigid deformations in gated emission CT. *IEEE Trans. Med. Imag.*, 25(9):1130–1144, 2006.

[15] A. M. Ostrowski, 1973. *Solutions of equations in euclidean and banach spaces.* New York : Academic Press, 3rd edition.

[16] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.*, 1:113–122, 1982.

BIOGRAPHICAL SKETCH

Jeff Zahnen was born on August 7, 1974, in Chicago, Illinois. He received his Bachelor of Science degree in mathematics with a minor in history from the University of Florida in 1996. Following this, he received his master's degree in 1998. In 2000, he began to study medical imaging techniques under the supervision of Dr. Bernard Mair. In 2006, he completed his Ph.D. work on emission tomography methods. During this time, he also served as the coach of the University of Florida College Bowl Team.