1994-06

A multilevel approach to the algebraic image reconstruction problem

Robinson, Bruce T.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/28382

A Multilevel Approach to the Algebraic Image Reconstruction Problem

by

Bruce T. Robinson
Major, United States Army
B.S., U.S. Military Academy, 1976
M.S., University of Colorado, 1985

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 1994

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1994	3. REPORT TYPE AND DATES COVERED Doctoral Dissertation

4. TITLE AND SUBTITLE

A MULTILEVEL APPROACH TO THE ALGEBRAIC IMAGE RECONSTRUCTION PROBLEM

5. FUNDING NUMBERS

6. AUTHOR(S)

Robinson, Bruce T.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)   The problem of reconstructing an image from its Radon transform profiles is outlined. This problem has medical, industrial and military applications. Using the computer assisted tomography (CAT) scan as an example, a discretization of the problem based on natural pixels is described, leading to a symmetric linear system that is in general smaller than that resulting from the conventional discretization. The linear algebraic properties of the system matrix are examined, and the convergence of the Gauss-Seidel iteration applied to the linear system is established. Next, multilevel technology is successfully incorporated through a multilevel projection method (PML) formulation of the problem. This results in a V-cycle algorithm, the convergence of which is established. Finally, the problem of spotlight computed tomography, where high quality reconstructions for only a portion of the image are required, is outlined. We establish the formalism necessary to apply fast adaptive composite (FAC) grids in this setting, and formulate the problem in a block Gauss-Seidel form. Numerical results and reconstructed images are presented which demonstrate the usefulness of these two multilevel approaches.

14. SUBJECT TERMS

Multilevel Method, Radon Transform, Image Reconstruction

15. NUMBER OF PAGES

187

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

ABSTRACT

The problem of reconstructing an image from its Radon transform profiles is outlined. This problem has medical, industrial and military applications. Using the computer assited tomography (CAT) scan as an example, a discretization of the problem based on natural pixels is described, leading to a symmetric linear system that is in general smaller than that resulting from the conventional discretization. The linear algebraic properties of the system matrix are examined, and the convergence of the Gauss-Seidel iteration applied to the linear system is established. Multilevel technology is successfully incorporated through a multilevel projection method (PML) formulation of the problem. This results in a V-cycle algorithm, the convergence of which is established. Finally, the problem of spotlight computed tomography, where high quality reconstructions for only a portion of the image are required, is outlined. We establish the formalism necessary to apply fast adaptive composite (FAC) grids in this setting, and formulate the problem in a block Gauss-Seidel form. Numerical results and reconstructed images are presented which demonstrate the usefulness of these two multilevel approaches.

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge the guidance and encouragement of my advisor Professor Van Emden Henson, whose support made this thesis possible. I would like to thank the Lord God, who has given me the talent, inspiration and perseverence to conduct this research. I would also like to thank my wife Nancy, who stood firmly behind me through the many high and low periods of the past three years. Finally, I would like to dedicate this work to Ben Cooper, whose goodness of heart and wit will be truly missed - his time on earth was far too short.

I. INTRODUCTION

The problem of determining the nature or properties of some object though a non-invasive procedure has many real-world applications. In the medical field, a doctor might want to determine whether or not a patient exhibiting symptoms associated with a brain tumor actually has such a tumor without opening the patient's skull to look inside. In industry, an inspector might need to verify the integrity of a pump encased in a pipe, without opening the pipe to do so. Many other similar situations exist. In all of these cases, it is often possible to determine the nature of the object's interior by measuring its density or some other physical property, and then using this measured data to *reconstruct* the object in question in terms of the measured property.

The example we use throughout this study is Computer Assisted Tomography (CAT), the well-known CAT scan. In addition to the obvious medical use, CAT has important military applications as well, such as jet aircraft engine manufacturing. Here, the density of unknown object, say a jet engine, is measured by passing x-rays of known intensity through it and recording the emergent intensities. Given this collection of x-ray intensity data, the density of the engine is then reconstructed for quality control.

Other medical tomographic applications include Positron Emission Tomogra-

phy (PET) and Single Photon Emission Tomography (SPECT), where the patient is administered a dose of a radioactive drug, which collects in the region of interest and then emits radiation which can be detected outside the patient.

Yet another application is Ionospheric Tomography, in which the electromagnetic density of the ionosphere is reconstructed. Data is collected by transmitting radio waves from beacons on the earth's surface through the ionosphere, where they are detected by an orbiting satellite.

A. GOALS OF THE RESEARCH

Our research has as its primary objective the expansion of the collection of problem types that can be approached with a multilevel method. We use the image reconstruction problem as the vehicle in this study. Currently, there are several established methods for reconstructing the density, or *image* . They include Fourier methods, backprojection methods, and algebraic methods - all of which have advantages and disadvantages in terms speed, accuracy and scope. We restrict ourselves to the algebraic methods, whose major limitation is that they are slow, and attempt to accelerate them by incorporating multilevel technology. We accomplish this by applying the principles of multilevel projection methods (PML) to the algebraic image reconstruction problem. A secondary goal of our research is the improved performance of an algebraic method. Finally, we develop a multilevel based method for solving the problem of *Spotlight Computed Tomography (CT),*in which a high-resolution reconstruction is desired for only a small portion of the image. We accomplish this by

applying the principles of PML to a composite grid image space, developing a fast adaptive composite grid (FAC) method.

Our work is new in that the natural pixel discretization, to a large extent, has never been analyzed in depth. This is the first rigorous application of (PML) to a problem outside of partial differential equations, as well as the first application of FAC to the Spotlight CT problem, or of natural pixels to a composite grid discretization. This study will generally follow the course we now outline. The image reconstruction problem is formally posed, and its relation to the *Radon transform* established. Some of the properties of the Radon transform are summarized, as are several inversion techniques. Formally, inverting the Radon transform solves the image reconstruction problem.

B.   STANDARD APPROACH

We look at one particular inversion method, the Algebraic Reconstruction Technique (ART), in greater detail. In the standard approach to ART, the space containing the image is discretized into small elements called *pixels,* and the image density is assumed to be constant throughout each pixel. This approach yields a large, sparse, underdetermined system of linear equations, the solution of which approximates the image. The system is normally solved with the method of Kaczmarz, which is examined and analyzed.

C. NATURAL PIXEL DISCRETIZATION

Next, we adopt an alternative discretization based on *natural pixels* . This discretization was originally proposed by Buonocore [Ref. 1], but a careful analysis of the properties of the resulting system has not previously been performed. This approach produces a linear system that is square, symmetric and in general smaller than that generated using square pixels. The system matrix is analyzed, revealing a rich collection of linear algebraic properties. The rank of the matrix is shown to be determined by the x-ray geometry used to generate it, and its null space is characterized. The square pixel and natural pixel discretizations are compared.

D. GAUSS-SEIDEL ITERATION

We consider the Gauss-Seidel iterative method for solving the natural pixel discretized problem, and convergence properties of Gauss-Seidel iteration when applied to this problem are established. A spectral analysis of a typical Gauss-Seidel iteration matrix for this problem is examined and serves to illuminate the numerical performance. The behavior of Gauss-Seidel applied to several test systems is analyzed, and numerical results are presented, along with several reconstructed images. This behavior, which can be characterized by rapid initial convergence followed by stalling, makes the problem a candidate for a multilevel approach.

E. MULTILEVEL METHODS

A review of the traditional multilevel methodology is presented. As the image reconstruction problem is not traditional, in that it shares few characteristics with problems arising from elliptic PDEs, we consider more general multilevel projection methods (PML). In PML [Ref. 2], the problem is discretized by orthogonal projections, and the projections themselves implicitly define the other multilevel components that make up the method. We show that the natural pixel discretization is a discretization by orthogonal projections, and formally cast the image reconstruction problem in a PML setting. Convergence of the method is established, and its behavior applied to several test systems is analyzed.

F. SPOTLIGHT COMPUTED TOMOGRAPHY

Finally, we consider the problem of *Spotlight CT,* where a portion of the image is desired at a high resolution. Reconstructing the entire image at high resolution is expensive, so a composite natural pixel discretization at different levels of resolution is developed. The resulting system matrix is again analyzed for its linear algebraic properties. As in the case of uniform discretization, the rank of the matrix can be determined by the geometry used to produce it, and the null space of the matrix is characterized. The composite linear system of equations can be solved using a block Gauss-Seidel method. This approach is formally shown to be equivalent to the Fast Adaptive Composite (FAC) multilevel method, for which rigorous theory has been

previously developed [Ref. 3]. Numerical results are presented, and composite grid images are reconstructed.

II. THE RADON TRANSFORM

Consider the problem of determining the internal structure of an object without having to cut or otherwise damage the object. We refer to such a problem as a *reconstruction problem,* and it will be the basis for the work that follows. An important category of reconstruction problems is medical imagery, where the object of interest is the human body, or some particular organ inside the body. The profile is then used to reconstruct the object. In the medical field, probes includes such things as x-rays, sound waves, and nuclear magnetic resonance signals. We will focus on the x-ray, and the resulting computer assisted tomography problem. For foundational reading see [Ref. 4, 5, 6, 7, 8, 9, 10].

A.   COMPUTER ASSISTED TOMOGRAPHY (CAT)

If a mono-energetic x-ray is passed along a straight line through some homogeneous object, then the intensity of the x-ray is observed to decrease according to the equation

$$I = I_0 e^{-\mu x},$$

where I_0 is the initial intensity, I is the emergent intensity, and μ is the *linear attenuation coefficient,* which depends on the material making up the object [Ref. 11]. If the x-ray passes through two different materials, traveling a distance x_1 through the first and a distance x_2 through the second, then the emerging x-ray will be attenuated

by

$$I = I_0 e^{-\mu_1 x_1 - \mu_2 x_2},$$

where μ_1 and μ_2 are the attenuation coefficients of the two materials. For several materials the relation is

$$I = I_0 e^{-\sum_j \mu_j x_j}.$$

For a non-homogeneous object, we may formally let the number of materials go to infinity while the distances traveled through each material become infinitesimally small. Then $\mu = \mu(x)$, the linear attenuation function, and the summation becomes an integral over the x-ray path L, yielding

$$I = I_0 e^{-\int_L \mu(x) dx}.$$

Now, consider passing many x-rays, along many paths, through an object, with the x-ray paths directed so that they are all coplanar. Then we may write the linear attenuation function as a function of two variables $\mu(x, y)$. The attenuation of one x-ray is then given by

$$I = I_0 e^{-\int_L \mu(x,y) ds}, \tag{2.1}$$

where the line integral is along the x-ray path L. As the linear attenuation function characterizes the object of interest, we will often refer to it as the *image* to be reconstructed. Equation (2.1) can be rewritten as

$$P = -log(\frac{I}{I_0}) = \int_L \mu(x, y) ds.$$

Figure 1 illustrates the path of one such x-ray through an object.

8
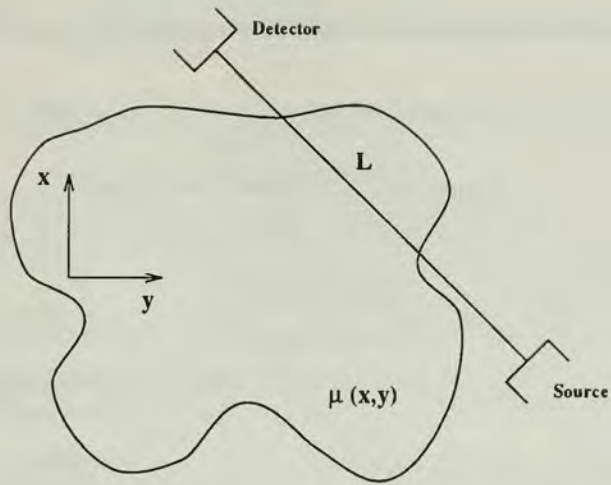
Figure 1. *An x-ray through an object.*

If the x-ray source and detector are moved along parallel straight lines past

the object as indicated in Figure 2, then it is possible to collect a set of attenuated

intensities, or a *profile,* for the object at some fixed angle ϕ, as

$$P(\rho, \phi) \;=\; \int_L \mu(x, y) ds. \tag{2.2}$$
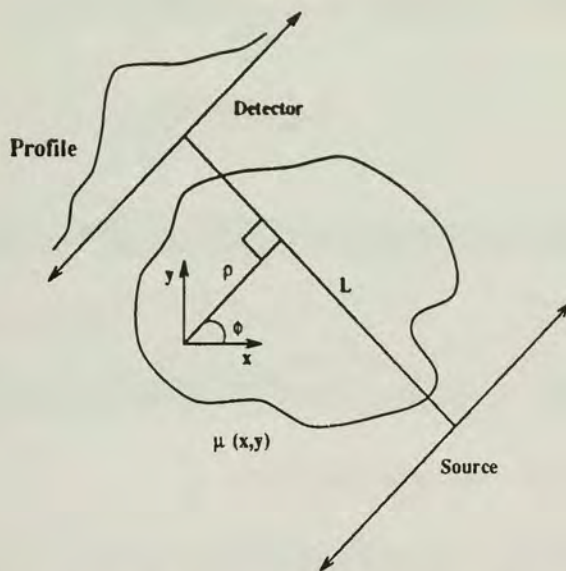
Figure 2. *A profile for a fixed angle ϕ.*

9

The basic problem of computer assisted tomography is to reconstruct the image $\mu(x, y)$ from a collection of profiles measured at various angles. Solving this problem is equivalent to finding the inverse of the Radon transform.

B.   THE RADON TRANSFORM

Let $u(x,y)$ be an arbitrary function defined on some region $D \in \mathcal{R}^2$. If L is any line in $\mathcal{R}^2$, then the mapping defined by the line integral of u along all possible lines L, also a function of two variables, is the *Radon Transform* of u, provided the integral exists. Formally,

$$R(L) = \int_L u(x, y) ds, \tag{2.3}$$

where the domain D may be all of $\mathcal{R}^2$, or some portion thereof. The mapping (2.3) was first studied in 1917 by Johann Radon, who also discovered an inversion formula by which u can be obtained from $R(L)$ [Ref. 12].

Consider a parameterization of the line L according to

$$\rho = x \cos \phi + y \sin \phi, \tag{2.4}$$

where ρ is a real number and ϕ is an angle measured from the positive x-axis. Then (2.4) determines the equation of a line through the xy-plane, normal to the unit vector $\vec{\xi} = (\cos \phi, \sin \phi)^T$, and a distance ρ from the origin, measured along $\vec{\xi}$. Defining $\vec{x} = (x, y)^T$, then (2.4) can also be written as

$$\rho = \vec{x} \cdot \vec{\xi}.$$

Using either (ρ, ϕ) or $(\rho, \vec{\xi})$ as the variables of the Radon transform, (2.3) can be written as

$$[Ru](\rho, \phi) = \int_L u(x, y)ds \quad \text{or} \quad [Ru](\rho, \vec{\xi}) = \int_L u(\vec{x})ds.$$

Figure 3 shows the geometry of the Radon transform of a function $u(x,y)$ in terms of ρ and ϕ, where ϕ is the angle defining a line normal to L, the path of the x-ray, and is measured counterclockwise from the positive x-axis. The parameter ρ is the signed distance from the origin to the line L.
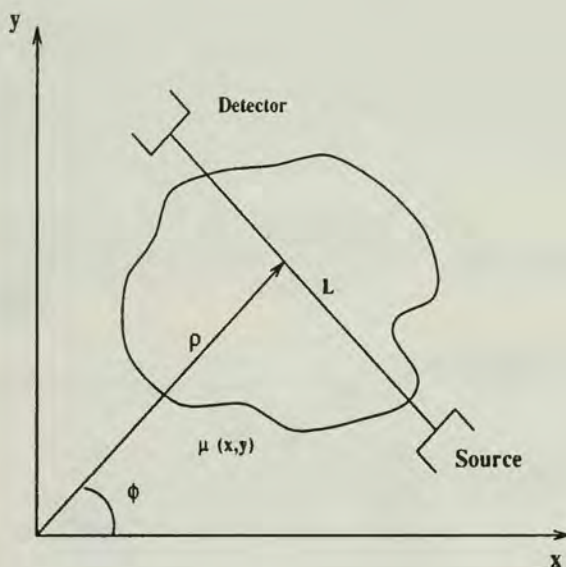
Figure 3. *The geometry of the Radon transform.*

It is sometimes useful to express the Radon transform, in terms of the Dirac delta function δ, as

$$[Ru](\rho, \phi) = \int_{\mathcal{R}^2} u(x, y)\delta(\rho - x\cos\phi - y\sin\phi)dxdy,$$

11

or

$$[Ru](\rho, \vec{\xi}) = \int_{\mathcal{R}^2} u(\vec{x}) \delta(\rho - \vec{x} \cdot \vec{\xi}) d\vec{x}.$$

For some objects, the Radon transform can be computed analytically. Consider a constant density disk of radius R centered at the origin. Explicitly

$$u(x, y) = \begin{cases} 1, & x^2 + y^2 \le R^2 \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Since this object is symmetric with respect to the angle ϕ, only one profile is required to determine the Radon transform. Let $\phi = 0$, so that the line integrals are computed along lines parallel to the y-axis, at a distance ρ from the origin. For values of $|\rho| > R$, the lines do not intersect the disk and the profile is zero. For values of $|\rho| \le R$, the transform is

$$[Ru](\rho, 0) = \int_{-\sqrt{R^2 - \rho^2}}^{\sqrt{R^2 - \rho^2}} dy = 2\sqrt{R^2 - \rho^2}.$$

The symmetry of $u(x,y)$ yields

$$[Ru](\rho, \phi) = \begin{cases} 2\sqrt{R^2 - \rho^2}, & |\rho| \le R \\ 0, & \text{otherwise.} \end{cases}$$

A graphical representation of the Radon transform of this disk is given in Figure 4 [Ref. 13].

The Radon transform can be extended to higher dimensions. Integrating $u(\vec{x})$, for $\vec{x} \in \mathcal{R}^n$, over all subspaces of dimension n-1 is also a Radon transform. For example, if $n = 3$, then the Radon transform is the set of all integrals of u over all planes in $\mathcal{R}^3$. We will restrict ourselves exclusively to functions of two variables.
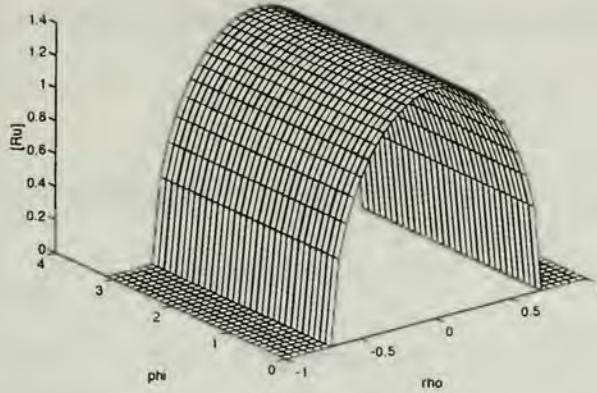
Figure 4. *The Radon transform of the function of a constant density disk.*

C.   PROPERTIES OF THE RADON TRANSFORM

The Radon transform operator has many properties that are needed when developing inversion techniques. We will outline several of the more important properties. A detailed examination can be found in [Ref. 12].

1.   Linearity

Given two functions f and g , and two scalars α and β, consider

$$
\begin{aligned}
R\{\alpha f + \beta g\} &= \int_{\mathcal{R}^2} [\alpha f(x,y) + \beta g(x,y)] \delta(\rho - x\cos\phi - y\sin\phi) dx dy \\
&= \alpha \int_{\mathcal{R}^2} f(x,y) \delta(\rho - x\cos\phi - y\sin\phi) dx dy \\
&\quad + \beta \int_{\mathcal{R}^2} g(x,y) \delta(\rho - x\cos\phi - y\sin\phi) dx dy \\
&= \alpha[Rf] + \beta[Rg].
\end{aligned}
$$

Thus, the Radon transform is a linear operator. This property is important in that fairly complex objects can be modeled with relative ease by contructing the transforms of a collection of simple objects and then combining the results. For example, to

13

analytically compute the Radon transform of an annulus centered at the origin, one would compute the Radon transforms of two disks of different radii, and subtract the smaller from the larger. So if

$$u(x,y) = \begin{cases} 1, & r_2^2 \leq x^2 + y^2 \leq r_1^2 \\ 0, & \text{otherwise,} \end{cases}$$

then the Radon transform of $u(x,y)$, by applying linearity, would be

$$[Ru](\rho, \phi) = \begin{cases} 2\sqrt{r_1^2 - \rho^2}, & r_2 \leq |\rho| \leq r_1 \\ 2\left(\sqrt{r_1^2 - \rho^2} - \sqrt{r_2^2 - \rho^2}\right), & |\rho| \leq r_2 \\ 0. & \text{otherwise.} \end{cases}$$

Figure 5 shows the graphical representation of the Radon transform of the annulus.
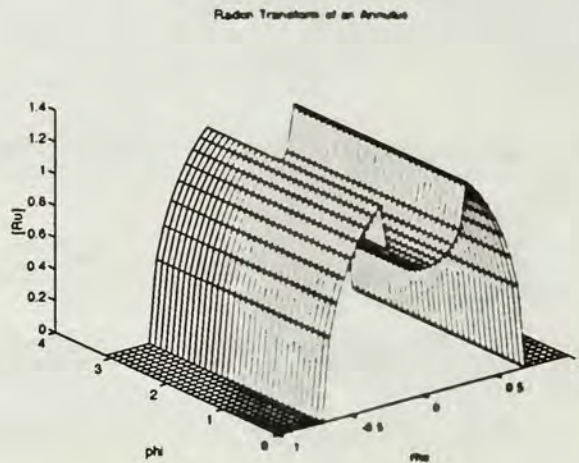
Figure 5. *The Radon transform of the function of a constant density annulus.*

14

2. The Shifting Property

Given a function $u(\vec{x})$, we consider the effect of shifting the argument of u by a vector $\vec{a}$. The Radon transform is

$$[Ru(\vec{x} - \vec{a})](\rho, \phi) = \int_{\mathcal{R}^2} u(\vec{x} - \vec{a})\delta(\rho - \vec{x} \cdot \vec{\xi})d\vec{x}.$$

Letting $\vec{y} = \vec{x} - \vec{a}$ results in

$$
\begin{aligned}
[Ru(\vec{y})](\rho, \phi) &= \int_{\mathcal{R}^2} u(\vec{y})\delta(\rho - (\vec{y} + \vec{a}) \cdot \vec{\xi})d\vec{y} \\
&= \int_{\mathcal{R}^2} u(\vec{y})\delta(\rho - \vec{a} \cdot \vec{\xi} - \vec{y} \cdot \vec{\xi})d\vec{y} \\
&= [Ru](\rho - \vec{a} \cdot \vec{\xi}, \vec{\xi}).
\end{aligned}
$$

Thus, shifting the argument of u by a vector $\vec{a}$ has the effect of shifting the resulting Radon transform a distance $\vec{a} \cdot \vec{\xi}$ along the ρ−axis. The shifting property allows for simplified computation of transforms of objects not centered at the origin.

Consider a disk of radius r centered at the point (a,b). We desire the transform of $u(x\text{-}a, y\text{-}b)$, where u is given by (2.5). Applying the shifting property results in

$$[Ru](\rho, \phi) = \begin{cases} 2\sqrt{r^2 - (\rho - a\cos\phi - b\sin\phi)^2}, & |\rho - a\cos\phi - b\sin\phi| \le r \\ \\ 0, & \text{otherwise.} \end{cases}$$

A plot of the Radon transform of the shifted disk is shown in Figure 6.

3. The Scaling Property

Consider

$$[Ru](\alpha\rho, \alpha\vec{\xi}) = \int_{\mathcal{R}^2} u(\vec{x})\delta(\alpha\rho - \vec{x} \cdot \alpha\vec{\xi})d\vec{x}$$

15
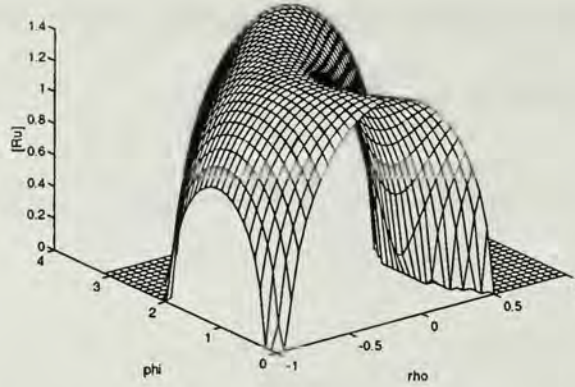
Figure 6. *The Radon transform of the function of a constant density disk shifted away from the origin.*

$$= \frac{1}{|\alpha|} \int_{\mathcal{R}^2} u(\vec{x}) \delta(\rho - \vec{x} \cdot \vec{\xi}) d\vec{x}$$

$$= \frac{1}{|\alpha|} [Ru](\rho, \vec{\xi}).$$

This is the scaling property.

As a special case where $\alpha = -1$, we have

$$[Ru](-\rho, -\vec{\xi}) = [Ru](\rho, \vec{\xi}),$$

so that the Radon transform is an even function of $(\rho, \vec{\xi})$. This eveness is significant in a practical sense, in that when on object is x-rayed, only the angular range from 0 to π need be considered.

The Radon transform can be viewed as a projection operator. For a fixed value of ϕ, the set of all line integrals as ρ varies is a projection of u into $\mathcal{R}$. This projection is identical to that defined by the reconstruction problem, as the right-hand-sides of (2.2) and (2.3) are the same, hence the relationship between the

16

Radon transform and the reconstruction problem. If we define

$$\breve{u}(\rho, \phi) = [Ru](\rho, \phi) = \int_L u(x, y) ds,$$

and view $\breve{u}(\rho, \phi)$ as measured data obtained from x-raying the object, then u can be reconstructed by inverting the Radon transform. We do not use Radon's inversion formula directly, as it has been shown to be numerically sensitive to inaccuracies in the data [Ref. 5], which in practice are present since the data is measured. Practical inversion techniques have been developed, a few of which will now be overviewed.

D. STANDARD INVERSION TECHNIQUES

There are several categories of inversion techniques for the Radon transform, and many variations within each category. The main categories are Fourier methods, backprojection methods, and iterative methods. This work will concentrate on the latter category, but we briefly discuss the other methods here.

1. Fourier Methods

The Fourier methods are based on the relationship between the Fourier transform and the Radon transform. The relationship is formalized in the *Central Slice Theorem*. Consider the 2-dimensional Fourier transform, $F_2\{f\}$, of a function of two variables, $f(x,y)$. We have

$$F_2\{f\} = \hat{f}(\omega_x, \omega_y) \equiv \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i(x\omega_x + y\omega_y)} dx\, dy.$$

Next, consider the one-dimensional Fourier transform of a profile, that is, the transform of $[Ru](\rho, \vec{\xi})$ taken with respect to ρ, for $\vec{\xi}$ fixed. This can be expressed as

$$
\begin{aligned}
F_1\{Ru\} &= \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} [Ru](\rho, \vec{\xi}) e^{-i\omega\rho} d\rho \\
&= \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} \left(\int_{\mathcal{R}^2} u\vec{x}) \delta(\rho - \vec{x} \cdot \vec{\xi} d\vec{x}) \right) e^{-i\omega\rho} d\rho \\
&= \frac{1}{2\sqrt{2\pi}} \int_{\mathcal{R}^2} u(\vec{x}) \left(\int_{-\infty}^{\infty} \delta(\rho - \vec{x} \cdot \vec{\xi}) e^{-i\omega\rho} d\rho \right) d\vec{x} \\
&= \frac{1}{2\sqrt{2\pi}} \int_{\mathcal{R}^2} u(\vec{x}) e^{-i\omega\vec{x}\cdot\vec{\xi}} d\vec{x} \\
&= \frac{1}{2\sqrt{2\pi}} \int_{\mathcal{R}^2} u(\vec{x}) e^{-i\vec{\omega}\cdot\vec{x}} d\vec{x} \\
&= \sqrt{2\pi}\hat{u}(\vec{\omega}),
\end{aligned}
$$

where $\vec{\omega} = \omega\vec{\xi}$ ranges over all of $\mathcal{R}^2$. The Central Slice Theorem, in two dimensions, says that the Fourier transform of a projection $\check{u}(\rho, \vec{\xi})$ is equal to a constant multiple of the two-dimensional Fourier transform $\hat{u}(\vec{\omega})$. Explicitly [Ref. 12]

Theorem 2.1: *Let the image u(x,y) have a two- dimensional Fourier transform, $\hat{u}(\omega_x, \omega_y)$, and a Radon transform, $\check{u}(\rho, \phi) = [Ru](\rho, \phi)$. If $F_1\{Ru\} = \hat{\check{u}}(\omega, \phi)$ is the one-dimensional Fourier transform (with respect to ρ) of the profile $[Ru](\rho, \phi)$, then*

$$
\sqrt{2\pi}\hat{u}(\omega_x, \omega_y) = \hat{\check{u}}(\omega, \phi),
$$

where $\omega^2 = \omega_x^2 + \omega_y^2$ and $\phi = \tan^{-1}(\omega_x, \omega_y)$. That is, the Fourier transform of the projection of u(x,y) onto the line in the direction of the vector $(\cos\phi, \sin\phi)^T$ is exactly a slice through the two-dimensional Fourier transform of u(x,y) along that direction.

The essence of Theorem 2.1 is

$$
\sqrt{2\pi}F_2\{u(x,y)\} = F_1\{Ru\} = F_1\{\check{u}(\rho, \phi)\}, \tag{2.6}
$$

18

where F_2 and F_1 indicate the two and one-dimensional Fourier transforms, respectively. Figure 7 gives a schematic diagram of the Central Slice Theorem in two dimensions. The interpolation step arises from numerical implementation using the Fast Fourier Transfrom (FFT).
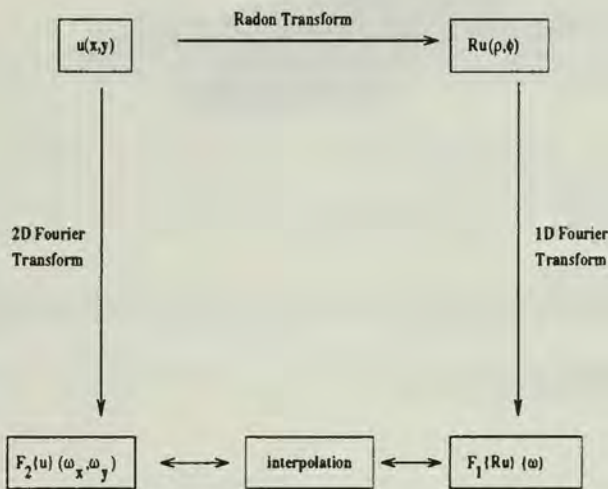
Figure 7. *Relationship between Fourier and Radon transforms.*

To invert the Radon transform, consider

$$u(x,y) \; = \; F_2^{-1} \left\{ \frac{1}{\sqrt{2\pi}} F_1\{\dot{u}(\rho,\phi)\} \right\},$$

which says that given the Radon transform of some image, we first take the one-dimesional Fourier transform of each profile. This gives data in polar coordinates. Fast implementation occurs through the use of the FFT, which requires data in Cartesian coordinates. Therefore, these results are interpolated to Cartesian coordinates and then an inverse 2-dimensional Fourier transform is taken to recover the image. A family of such Fourier methods exists, based on how the interpolation is carried out [Ref. 5, 14, 13, 15].

2. Backprojection Methods

Let $\psi(\rho, \vec{\xi})$ be an arbitrary function, where $\rho = \vec{x} \cdot \vec{\xi} = x \cos \phi + y \sin \phi$, as before. The *backprojection operator B* is defined as

$$[B\psi](x,y) = 2 \int_0^\pi \psi(x \cos \phi + y \sin \phi, \vec{\xi}) d\phi.$$

The action of the backprojection operator can be interpreted as follows. Fix a point (x, y). Then for every angle ϕ, $u(x, y)$ is a value contributing to the line integral along the line $\rho = x \cos \phi + y \sin \phi$. That is, $u(x, y)$ is part of $[Ru](\rho, \phi)$ for every $\phi \in [0, 2\pi)$. Backprojection assembles at (x, y) the sum (integral) of all values to which it *could have* contributed, i.e.

$$B\{Ru\}(x,y) = \int_{-\pi}^\pi [Ru](\rho, \phi) d\phi,$$

which by the eveness of the Radon transform becomes

$$B\{Ru\}(x,y) = 2 \int_0^\pi [Ru](\rho, \phi) d\phi.$$

Geometrically, backprojection is a form of image reconstruction. For each profile, the values corresponding to a point (x, y) are spread over the image region. The linear superposition of values that results is an approximation of the image. Figure 8 shows the backprojection operation for two profiles taken of a rectangular object.

Backprojection by itself is not a satisfactory reconstruction technique, as is evident in Figure 8 by the areas surrounding the rectangle that have been shaded. Normally, backprojection is used as an intermediate step in other inversion techniques

20
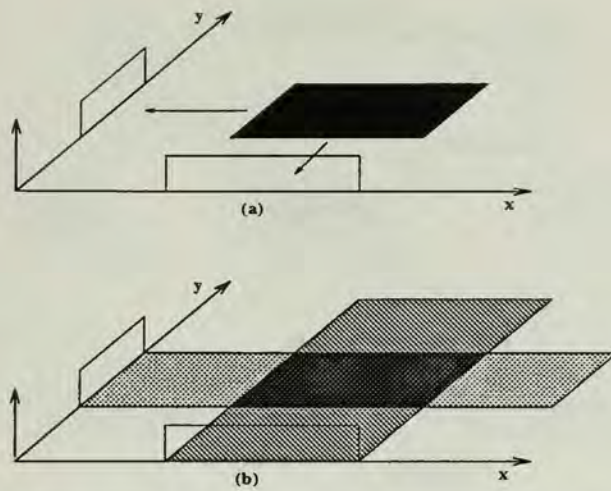
Figure 8. *(a) Two profiles of a rectangle. (b) Backprojection.*

that are very effective. One of these techniques, known as *backprojection of filtered projections*, is the most widely used reconstruction method [Ref. 16]. This method can be formulated as

$$u(x, y) = BF_1^{-1}\{|\omega|\, F_1\{Ru\}\}.$$

The method can be summarized follows. Given the Radon transform of some image, take the one-dimensional Fourier transform of each projection and then *weight* the results with a factor $|\omega|$. This weight factor is the *filter*. Next, take the inverse Fourier transform of the weighted projections and then backproject the results to recover the image. Figure 9 gives a schematic diagram of the backprojection of filtered projections method. It should be pointed out that this method can be performed entirely in image/projection space without the use of Fourier transforms, by using convolutions instead.
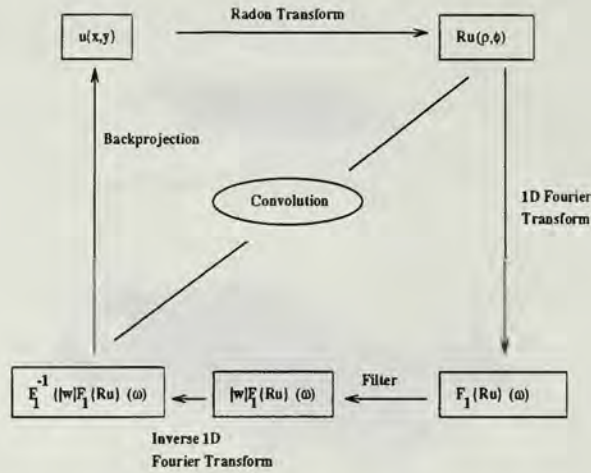
Figure 9. *Relationships in backprojection of filtered projections.*

This alternate path to the image is also shown in Figure 9. For ease of development, we only describe the frequency domain implementation.

Another closely related technique involves backprojecting the projections first, and then filtering the backprojection. This method is known as the *filter of backprojected projections,* and can be expressed as

$$u(x,y) = F_2^{-1}\{|\omega|\, F_2\{B\{Ru\}\}\}.$$

Given the Radon transform of some image, backproject it and then take the 2-dimensional Fourier transform of the result. This quantity is then filtered by multiplication by $|\omega|$ and then the inverse 2-dimensional Fourier transform is taken, recovering the image. Figure 10 illustrates schematically the implementation of this method.

The final category of inversion techniques are iterative. They involve discretizing the problem into a linear system of equations, which is subsequently solved to recover the image. We wish to concentrate exclusively on this category of meth-

22
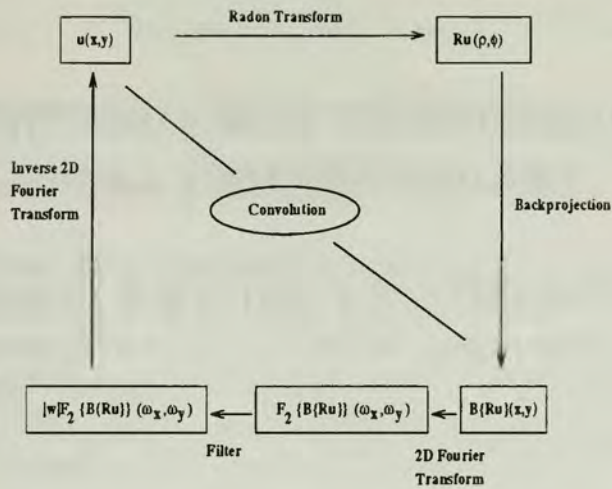
Figure 10. *Relationships in filter of backprojected projections.*

ods, first discussing the standard techniques, and then developing a more efficient method. In the next chapter we will develop and analyze the standard technique of discretization by *square pixels* , as well as the iterative method of *Kaczmarz* to solve the resulting linear system.

III. ALGEBRAIC RECONSTRUCTION TECHNIQUES (ART)

A. DISCRETIZATION BY SQUARE PIXELS

The algebraic reconstruction technique involves discretizing the Radon transform problem $Ru = f$ into a system of linear equations, whose solution approximates u. A family of ART methods can be developed based upon how one discretizes the problem and solves the resulting linear system.

The standard approach is to discretize the problem by square pixels. Let u be the density function of the image to be reconstructed, and assume it is defined in a square region of unit area. This unit square is subdivided into a grid of smaller elements, or pixels (from *picture elements*).and u is assumed to be constant throughout each pixel. Let the image be divided into n^2 square pixels of equal area, so that we are approximating the continuous solution u with an $n \times n$ array of numbers. These numbers will be the unknowns in the linear system of equations.

We define the geometry at which x-rays are passed through the image. Assume there is an array of N_1 detectors positioned to measure the intensity of the x-rays after they have passed through the image, and an array of N_1 x-ray sources positioned parallel to the detectors so that the path of an x-ray through the image is perpendicular to the detectors. Assume further that each detector measures the intensity of only one x-ray, and that the sources are positioned such that the x-rays

24

cover the entire image. Let the source/detector arrays be rotated about the image, stopping at M angles ϕ_1, ϕ_2, $\cdots$, ϕ_M. At each angle the array produces a sampled profile, yielding a set of M such profiles. The N_1 detectors on each of the M angles determines the geometry of the problem.

Each x-ray passing through the image defines an equation in the linear system. Figure 11 depicts the ith x-ray at angle ϕ_j passing through the image, which has beeen discretized into an $n \times n$ array of square pixels. The equation generated by this x-ray is given by

$$\sum_{I=1}^{n} \sum_{J=1}^{n} w_{IJ} x_{IJ} = f_{ij},$$

where f_{ij} is the measured intensity of the x-ray, the x_{IJ} are the unknown values for the pixels, and the w_{IJ} are weight factors which are non-zero only for those pixels through which the x-ray passed. Observe that

$$\sum_{I=1}^{n} \sum_{J=1}^{n} w_{IJ} x_{IJ} = f_{ij} \approx Ru(\rho_i, \phi_j) = \int_{L} u(x, y) ds,$$

i.e., the sum approximates the integral.

There are N_1 x-rays at each of M angles passing through the image, for a total of $N = N_1 \times M$ such equations. We can write the resulting linear system as $K\vec{x} = \vec{f}$, where $K : \mathcal{R}^{n^2} \to \mathcal{R}^N$.

The weight factors w_{IJ}, which are the entries of K, can be assigned in several ways, depending on assumptions made about the physical nature of x-rays. First, assume an x-ray has no width, so that its path through the image is a line. Then the
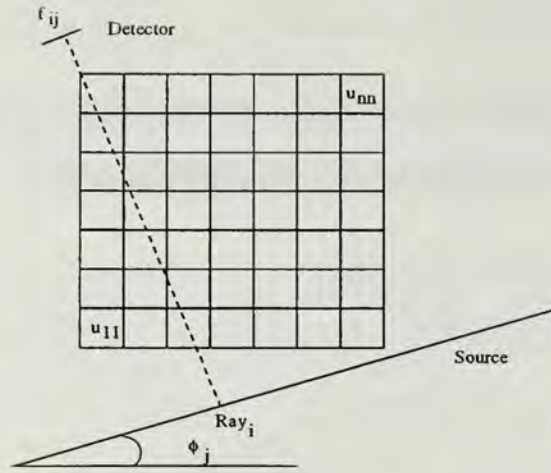
Figure 11. *ART geometry.*

weight factors can be assigned as

$$
w_{IJ} = \begin{cases} 1, & \text{x-ray passes through } IJ^{th} \text{ pixel} \\ \\ 0, & \text{otherwise} \end{cases}
$$

We call this approach the *zero-one* discretization. It is attractive in its simplicity, but it has several drawbacks. For example, if an x-ray passes through the center of a pixel, or just through the tip of a corner, the weight factor is still assigned a value of one, which intuitively seems inaccurate. Also, it is possible for a ray path to coincide with the border of two pixels, in which case we must decide to either assign both pixels a value of zero or both a value of one. Again, either decision seems inaccurate.

Some of these inaccuracies can be overcome by letting the weight factors be defined as the *lengths* of the ray paths through the pixels. This approach, which we call the *thin ray* discretization, corrects the problem of assigning equal weights to a pixel regardless of whether the x-ray passed through its center or just cut its corner.

26

This approach most accurately represents the line integrals of $[Ru]$. However, the possibility of x-rays coinciding with pixel boundaries still exists.

Another approach to assigning the weight factors can be developed by altering our assumption that x-rays have no width. Let the x-rays have width, so that a path through the image is a strip. Then the weight factors can be assigned as the area of the square pixel contained within the strip. We call this approach the *fat ray* discretization. It has a physical justification in that the detectors are actually a photgraphic plate subdivided by baffles into detection bins. Each bin will detect x-rays across its width, so the rays are modeled as strips.

No matter which of the discretizations is used, it should be clear from Figure 11 that the number of pixels intersected by an x-ray is small compared to the total number of pixels. Thus the resulting matrix will in general be sparse. Also, since one would desire as high a resolution image as possible without subjecting the patient to lethal doses of radiation, the number of x rays used is usually less than the number of pixels, producing a rectangular matrix and an underdetermined system of equations. A typical medical application could involve $n = N_1 = 512$ and $M = 180$. Thus the size of the problem is quite large as well, in this case involving a matrix with in excess of 2.4×10^{10} entries, of which less than one percent are nonzero. Figure 12 illustrates the sparsity pattern of such a matrix.

The size of this problem precludes a direct method of solution, so we look to iterative methods to solve the linear system. As the matrix is in general rectangular,
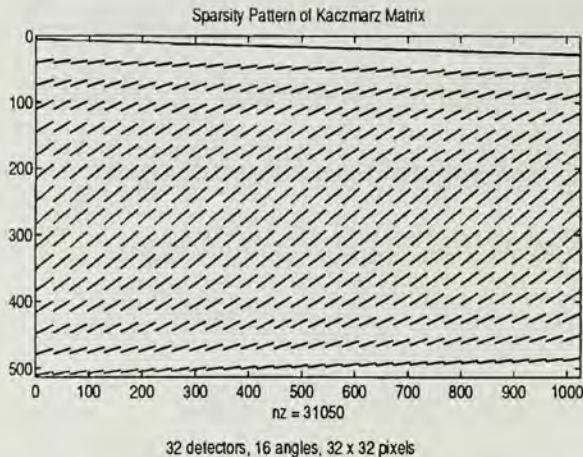
Figure 12. *ART matrix.*

and we have no guarantee of non-zero elements on the diagonal, the classical relaxation methods such as Jacobi and Gauss-Seidel are not appropriate here. The method of Kaczmarz [Ref. 17] can be applied to such problems, so we present it here, and then later use it for comparison.

B. THE METHOD OF KACZMARZ
1. Definition and Properties

Let $K : \mathcal{R}^{n^2} \to \mathcal{R}^N$ define the system of equations $K\vec{x} = \vec{f}$. Then given an initial approximation $\vec{x}_0 \in \mathcal{R}^{n^2}$, the method of Kaczmarz corrects the approximation by sequentially adding a multiple of each row of the matrix K to it. The desired multiple is that which causes the corresponding component of the residual, $\vec{r} = \vec{f} - K\vec{x}$, to vanish. One cycle through all N equations is called a *sweep*. Letting $\vec{w}_i$ be the ith standard basis vector, one sweep of Kaczmarz can be written as

For $i = 1, 2, \cdots, N$

Solve $\langle \vec{w}_i, K(\vec{x} + sK^T\vec{w}_i) - \vec{f} \rangle = 0$ for s

28

$$\text{Set } \vec{x} = \vec{x} + sK^T\vec{w_i}$$

Solving for s yields

$$\langle \vec{w_i}, K\vec{x} + sKK^T\vec{w_i} - \vec{f} \rangle = 0,$$

$$\langle \vec{w_i}, -\vec{r} + sKK^T\vec{w_i} \rangle = 0,$$

$$-\vec{w_i}^T\vec{r} + s\vec{w_i}^T KK^T\vec{w_i} = 0,$$

$$s\vec{w_i}^T KK^T\vec{w_i} = \vec{w_i}^T\vec{r},$$

$$s = \frac{\vec{r_i}}{||K^T\vec{w_i}||^2},$$

where $\vec{r_i}$ is the ith component of the residual vector $\vec{r}$.

We chose the method of Kaczmarz because of its convergence properties, which simply stated are that if a solution to the linear system exists, then Kaczmarz will converge to it. Formally, we cite the following convergence theorem [Ref. 14].

Theorem 3.1: *Let H_1 and H_2 be real Hilbert spaces, and let $R : H_1 \to H_2$ be a continuous linear operator. Let $f \in H_2$ be given. Assume that $Ru = f$ has a solution. If $u_0 \in Range(R^*)$, then the sequence u_k generated by the method of Kaczmarz converges to the solution of minimum norm as $k \to \infty$.*

If $\vec{f} \in Range(K)$, then the linear system $K\vec{x} = \vec{f}$ satisfies the hypothesis of the above theorem. The usual choice for $\vec{x_0}$ is the zero vector. It is not totally clear if the minimum norm solution is the *best* in terms of how the reconstructed image appears, but because we can find this solution and it is unique, this will be the solution we seek.

Geometrically, Kaczmarz's method acts by correcting the approxima-
tion in a direction orthogonal to the hyperplane defined by the equation being con-
sidered. Figure 13 shows the action of several iterations on a problem consisting of
two equations. Observe that convergence will be fast when the hyperplanes defined
by the equations are nearly orthogonal, and slow if they are nearly parallel.
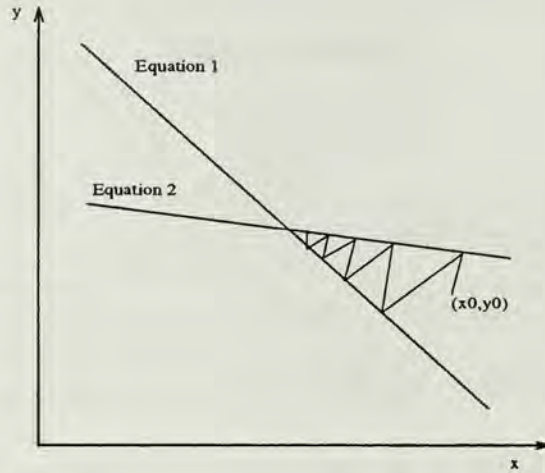
Figure 13. *Geometric interpretation of Kaczmarz's method.*

It is also interesting to consider a physical interpretation of the action
of Kaczmarz's method in terms of the reconstructed image. Starting with the zero
vector as an initial guess, the image is black. The action of Kaczmarz is to add a
multiple of each row of K to the solution, specifically a multiple of the element of
the residual corresponding to that row. Each row of K can be attributed to an x-ray
passed through the image, so the effect of the iteration is to spread a multiple of that
row back across the ray path through the image, assigning to each pixel along the
path that portion of the residual proportional to that pixel's contribution to the ray

path integral. That is, the jth correction is determined by the amount the current approximation fails to satisfy the jth equation, normalized by the area of that strip integral. Thus the action of Kaczmarz's method is a form of backprojection, which from Chapter II we know to be a primitive technique for image reconstruction in its own right.

One sweep of Kaczmarz's method involves an inner product, one scalar multiplication and one vector addition for each of the N equations, so it is an O($N \times n^2$) operation. This can be greatly reduced by exploiting the sparseness of K. For example, using the thin ray discretization the number of non-zero entries in any row of the matrix will not exceed $2n\sqrt{2}$. By working only with these nonzero entries, the amount of work involved can be reduced to O($N \times n$).

2.   Numerical Performance

The method of Kaczmarz is applied to several linear systems created at different x-ray geometries, and with assorted right sides generated both analytically and experimentally. Effectiveness is measured in terms of the 2-norm of the residual vector. These and all subsequent numerical computations are carried out using MATLAB Version 4.1 running on a SUN Sparc Station 10. In all cases, initial rapid convergence was followed by stalling, with the magnitude of the residual error well short of machine precision, which for the SUN is 2.22×10^{-16}. Figure 14 depicts graphs of the norm of the residual and the convergence factor plotted against iterations. Here the convergence factor at each iteration k is computed as the ratio of

31

the norm of the residuals after the $(k+1)^{st}$ sweep to the residual norm after the kth sweep. Note that after just a few sweeps, the convergence of the method has slowed significantly.
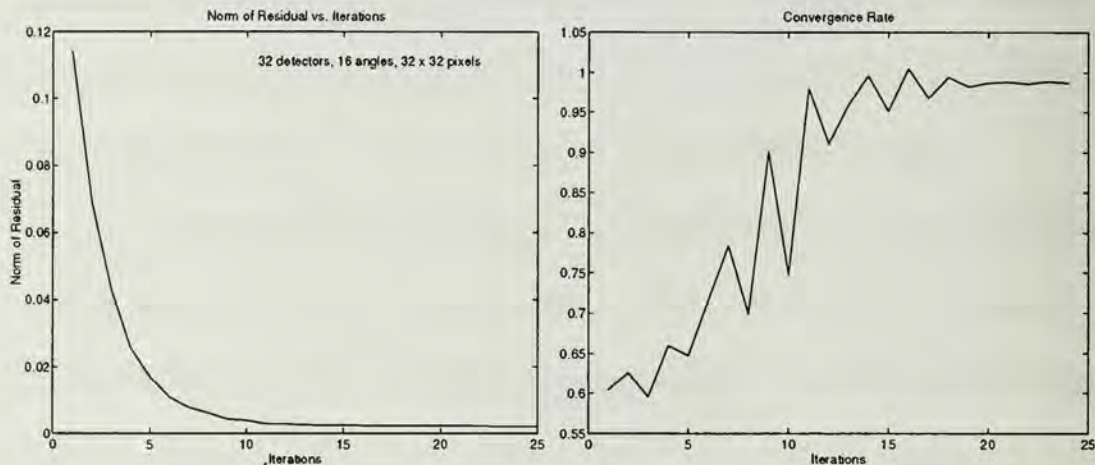
Figure 14. *Convergence of Kaczmarz's method.*

The geometry of the example problem is 32 detectors on each of 16 angles, with a square pixel discretization using a grid of 32 pixels × 32 pixels. Other geometries were examined, and this is a typical example. The image being reconstructed is a *brain* phantom, constructed by overlaying ellipses and rectangles of various sizes and grey levels inside the unit square, to simulate the cross section of the skull and brain. The data vector $\vec{f}$ was then created by *projecting* the image through multiplication by the matrix K. Note that while this data generation is artificial, it certainly assures us that $\vec{f} \in Range(K)$ and that infinitely many solutions exist. Figure 15 shows the "exact" and reconstructed images. In the reconstruction, all of the features are resolved to some degree. The white *skull* is quite clear, as are

32

the general shape, intensity and location of all the features contained within.
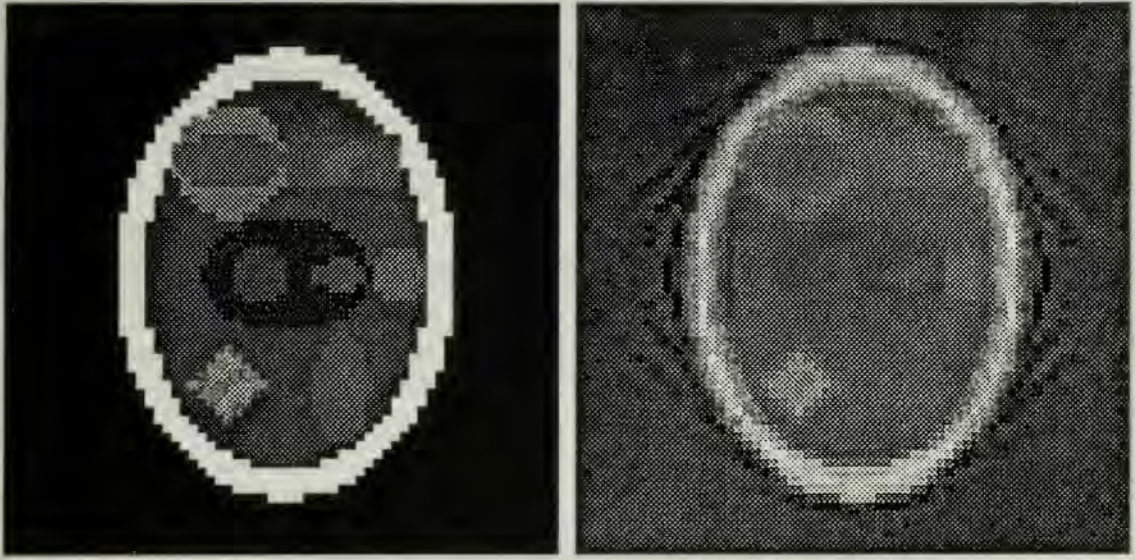
Figure 15. *Actual and reconstructed brain phantom images.*

In an attempt to explain the performance of the method, a thorough analysis of the matrix K is required. We begin that analysis with its singular value decomposition.

C.   ANALYSIS OF THE MATRIX K

The singular value decomposition (SVD) of the matrix K is

$$K = U\Sigma V^T, \tag{3.1}$$

where U and V are orthogonal matrices, and Σ is a diagonal matrix whose diagonal entries σ_i are the *singular values* of K. The columns of U and V are known as *left* and *right singular vectors*, respectively. Singular values are real and non-negative,

and we order them so that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \cdots 0.$$

The number of nonzero singular values $r \leq N$ equals the rank of K. Figure 16 shows the singular values of K generated at 32 detectors on each of 20 angles, with the image decomposed into a 32×32 array of square pixels using the fat ray discretization. We will refer to this particular geometry as the *standard test geometry.*Empirical evidence indicates that this matrix K is typical.
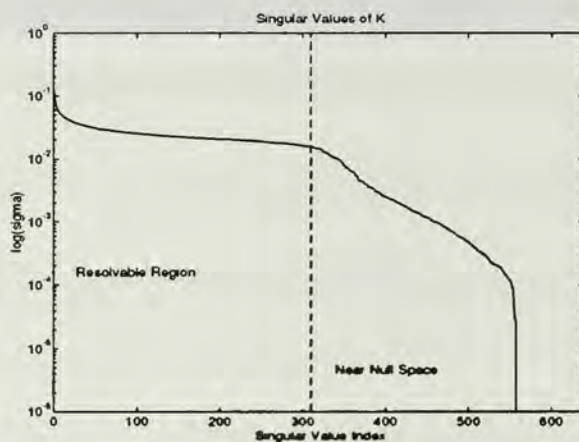
Figure 16. *Singular values of a typical matrix K.*

This figure contains a characteristic exhibited by the singular value spectrum of all such K matrices examined. That is, the spectrum can be separated into three disctinct regions, or bands, as shown in Figure 17. The first of these is the left portion of the spectrum, or *resolvable region*, where the singular values plot nearly horizontal. In the center is the second, a narrow region that we will call the *near null space*, shows marked decay of the singular values that ends abruptly as the singular

34

values drop off toward zero in the third region. The zero singular values correspond to the null space of K , implying that the matrix is rank deficient. In Figure 16, the resolvable region ranges from index 1 to about index 300, the near null space from index 301 to index 575, and those singular values with indices larger that 575 are in the null space.
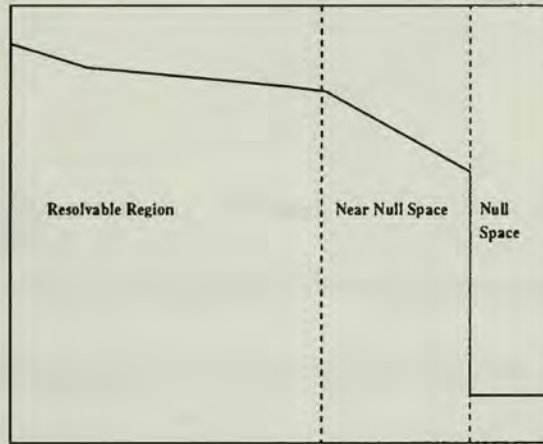
Figure 17. *The three bands of the singular value spectrum.*

Equation (3.1) can be rewritten as

$$KV = U\Sigma,$$

and if the columns are equated in the matrices on each side of this expression, the collection of linear systems

$$K\vec{v_i} = \sigma_i \vec{u_i}, \quad i = 1 : N \tag{3.2}$$

arises, where $\vec{u_i}$ and $\vec{v_i}$ denote the ith columns of U and V , respectively. If Kaczmarz's method is applied to these linear systems for various values of i , it is possible

35

to determine which singular values σ_i have singular vectors $\vec{v}_i$ that are slow to be reconstructed. The columns of V are linearly independent and form an orthonormal basis for $\mathcal{R}^{n^2}$, the space where the images live. Likewise, the columns of U form an orthonormal basis for $\mathcal{R}^N$, which is projection space. Thus any right side can be expressed uniquely as $\sum_{i=1}^{N} \alpha_i \vec{u}_i$, and any solution in the $Range(K)$ can be expressed as $\sum_{j=1}^{n^2} \beta_j \vec{v}_j$. Then by solving the SVD system

$$K\tilde{\vec{v}}_i = \sigma_i \vec{u}_i, \qquad (3.3)$$

whose solution is $\vec{v}_i$, for each i, it is possible to determine how well the solution component $\tilde{\vec{v}}_i$ can be recovered. The solutions should be close approximations to the corresponding singular vectors $\vec{v}_i$. The quality of an approximate solution can be analyzed by decomposing it into a linear combination of the singular vector basis as

$$\tilde{\vec{v}}_i = \sum_{j=1}^{N} \beta_j \vec{v}_j. \qquad (3.4)$$

The singular vector basis is orthonormal, and the β_j's can be computed as

$$\beta_j = \tilde{\vec{v}}_i^T \vec{v}_j, \quad \text{for } j = 1 : N. \qquad (3.5)$$

For the exact solution $\vec{v}_i$, we have

$$\beta_j = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases}$$

A plot of the coefficients for the decomposition of this solution would be a spike of magnitude one at index i. The following figures are plots of the absolute values of

36

the coefficients, $|\beta_j|$, of approximate solutions after 1 and 25 sweeps of Kaczmarz's method. Shown are the results for $i = 150$, a $\vec{v}_j$ the resolvable region, and for $i = 450$, a $\vec{v}_j$ in the near null space.
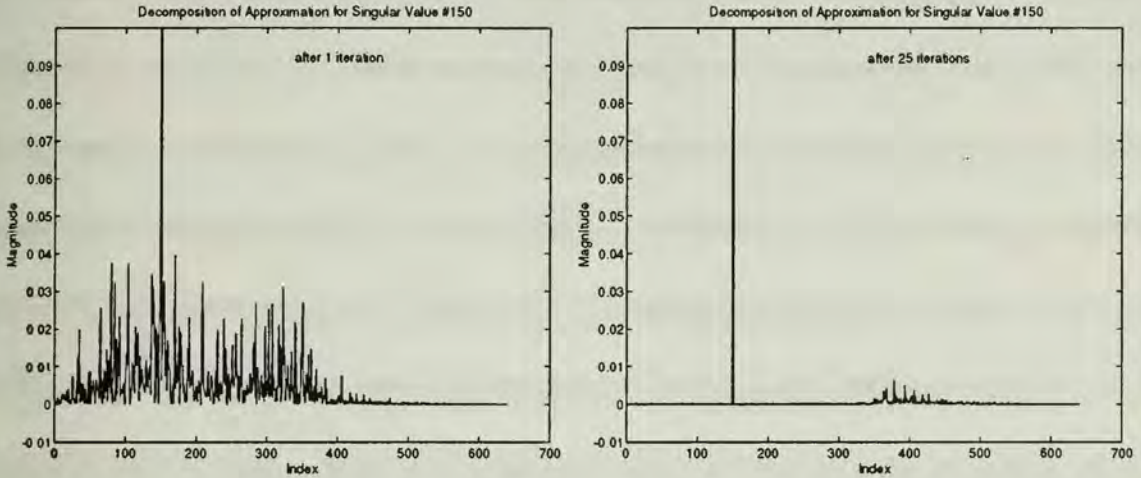
Figure 18. *Plot of the coefficients of the decomposition of an approximate solution from the resolvable region after 1 and 25 iterations. The plots are magnified to bring out detail.*
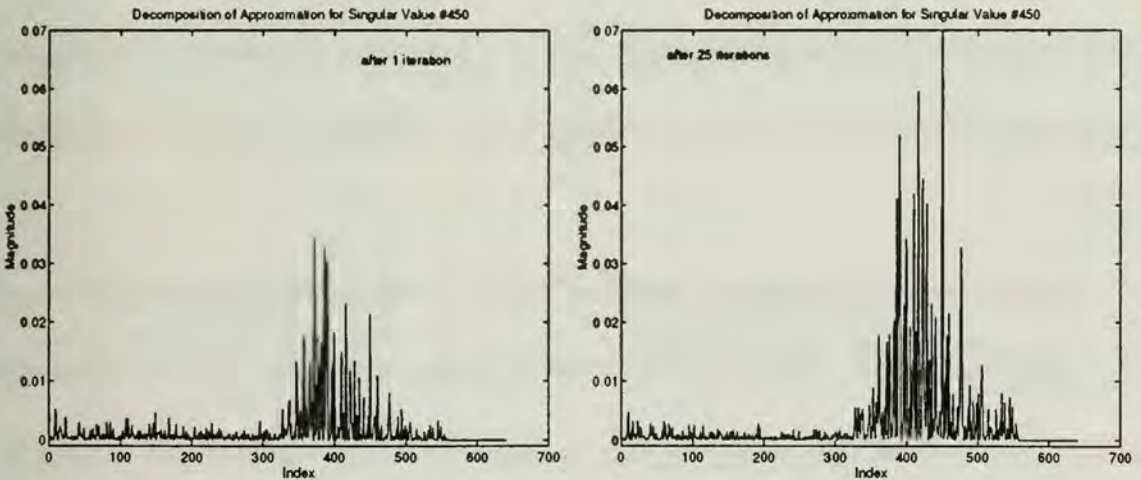
Figure 19. *Plot of the coefficients of the decomposition of an approximate solution from the near null space after 1 and 25 iterations.*

A qualitative interpretation of these plots follows. We observe from the experiments the components $\vec{v}_j$ for indices in the resolvable region are almost totally recovered. A spike of magnitude one located at the appropriate index is clearly present, along with a small amount of noise. These components do not adversely affect the performance of the iteration. On the other hand, $\vec{v}_j$ for indices associated with the near null space the components are only partially recovered. There is no easily recognizable spike, and significant noise is present. These components represent the *unrecoverable* , or *slow* components of the solution, and they cause the iteration to stall. It is for these reasons we name the regions resolvable and near null space. Finally, it should be noted that the iteration *mixes modes*,that is, introduces additional components of the singular value spectrum as noise into the approximation $\tilde{\vec{v}}_i$ that are not part of the exact solution $\vec{v}_i$. The mode mixing occurs in the near null space. This behavior has serious implications, in that we could have an exact solution that is defined entirely in the resolvable region, and the iteration will introduce components (through mode mixing) in the near null space that subsequently will be difficult to recover.

From our analysis thus far, if the solution has components in the near null space of K , then Kaczmarz's method will be slow in recovering them. Also, the iteration mixes modes, introducing components in the near null space that are not part of the actual solution. If the *width* of the near null space could somehow be controlled, we might be able to improve performance. We first considered how the geometry of the

38

problem influences the near null space. It is determined experimentally that, for a fixed number of detectors, the width of the near null space increases with the number of angles, (as does the dimension of the null space). Figure 20 illustrates the singular value spectrum for two matrices contructed from 32 detectors, using both 4 and 16 angles. The image space is decomposed into a 32 × 32 array of square pixels using the fat ray discretization. For the 4 angle geometry, the near null space is defined by indices 85 to 109, while for the 16 angle geometry it ranges from indices 300 to 449. This is an increase in the relative size of the near null space from 21% to 29% of the spectrum. Intuitively, one would expect the quality of the reconstruction to improve with a larger number of angles (hence more data). Thus a trade-off likely exists between accuaracy and performance.
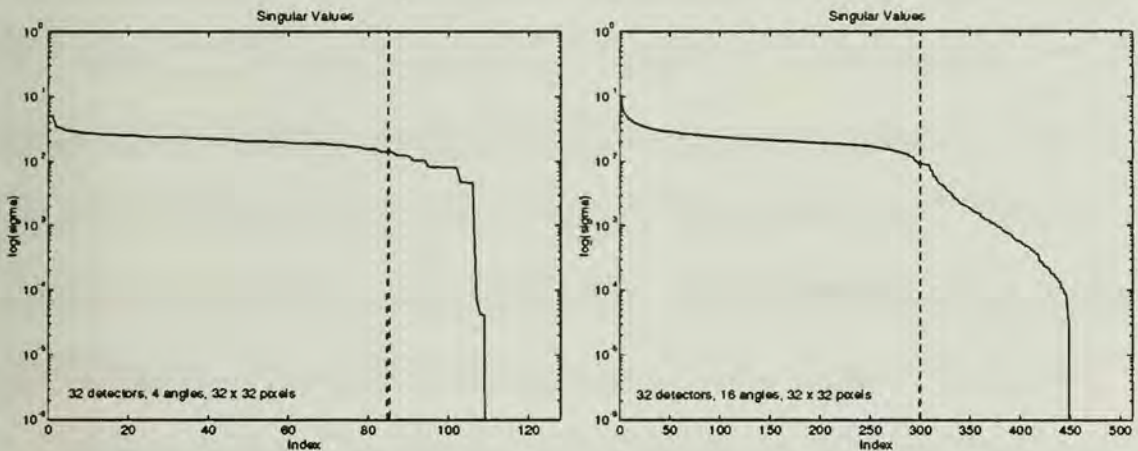
Figure 20. *Singular values for two different geometries.*

The nature of the Kaczmarz method itself will allow us to determine the source of the near null space components generated during the iteration. Recall that during a

sweep of Kaczmarz, we correct the approximation by adding a multiple of each row of the matrix K to the approximation. Since the rows of K are defined in the same space as is the image, that is, $\mathcal{R}^{n^2}$, we can decompose them in terms of the singular vector basis and discover where the slow components originate. An investigation along these lines reveals that strong near null space components are present in those rows of K corresponding to x-rays that nearly miss the image region. It should be pointed out that the square in which the image lives is the image region. The image may be zero throughout most of the image region, but the *value* of the image is immaterial. We are concerned with rays that nearly miss the image region. A ray that misses the image entirely produces a row of zeros in the matrix and is not used in the reconstruction. The rays adjacent to these are those are of interest to us. It is our conjecture that the rows of K corresponding to x-rays that nearly miss the image, i.e. adjacent to rows of all zeros corresponding to x-rays that entirely miss the image, are the major source of the slow components in the near null space. Figure 21 depicts a simple example of one angle of a fat ray discretization. Rays 1 and 8 produce rows of zeros as they entirely miss the image. Rays 2 and 7 are near misses and most likely correspond to rows in the matrix which contain near null space components. The remaining rays generate rows that most likely contain only components in the resolvable region.

We now return to the standard tets geometry of 32 detectors and 20 angles. Figure 22 depicts the spectral decomposition of a row corresponding to an x-ray passing through the center of the image (row 8), as well as that of a row corresponding
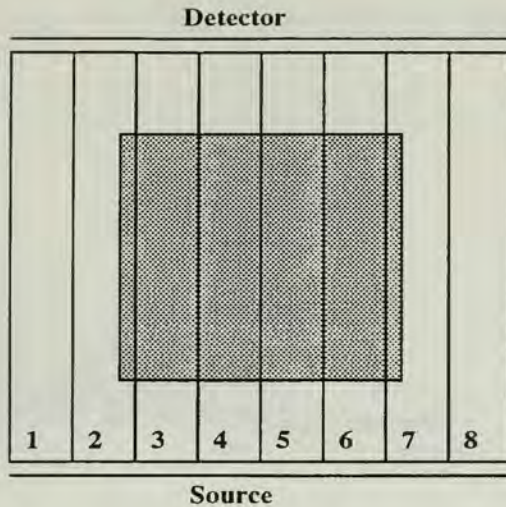
Figure 21. *One angle of a fat ray discretization.*

to a near miss (row 28), taken from a matrix with geometry 32 detectors, 4 angles, and 32×32 square pixels. Recall from Figure 20 above that for this geometry the near null space ranges from indices 85 to 109. The presence of near null space components is obvious in the second plot. One solution to this problem is to drop the near miss rows from the matrix, but doing so results in a geometry that does not completely cover the image. Another approach might be to keep the rows in K, but not include them in the Kaczmarz sweep. Another is to join near-miss strips with their interior neighbors. In all cases, inaccuracies might arise if significant parts of the image lie along its edges or in its corners.

Finally, we consider the spectral decomposition of the image we are trying to reconstruct. Let $u(x, y)$ be the image to be reconstructed, and assume we have discretized it into an $n \times n$ array of square pixels $\vec{x} \in \mathcal{R}^{n^2}$ so that the value in each square pixel is constant. This array is the *exact* solution that we are trying
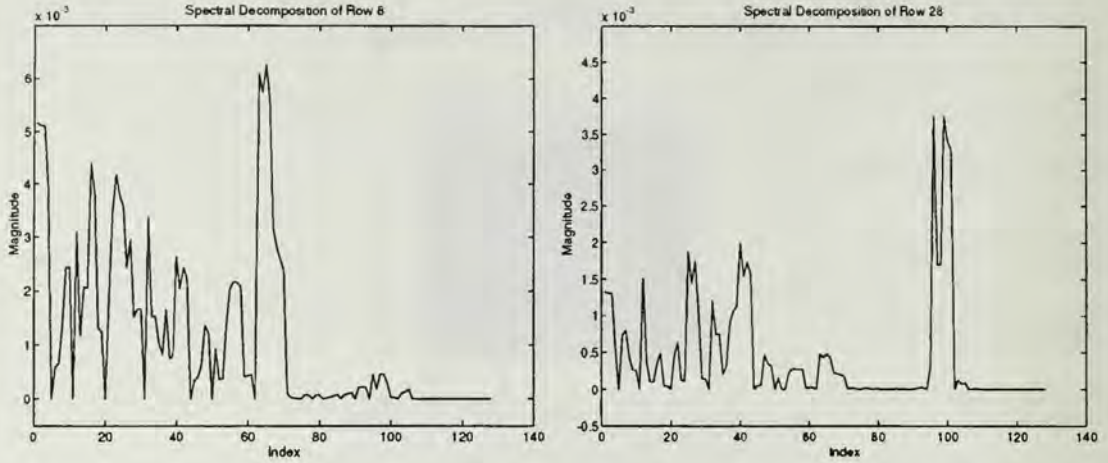
Figure 22. *Spectral decomposition of an interior and near miss row of K.*

to approximate. Based on experimental data. if we decompose $\vec{x}$ in terms of the singular vector basis of K,we find that the exact solution contains components in $NS(K)$. Figure 23 shows the spectrum of a phantom brain image using the SVD of a matrix with geometry 32 detectors, 8 angles and $32 \cdot 32$ pixels. The null space of this matrix is associated with indices 226 to 256. It is clear from the figure that the exact solution has components in the null space of the matrix. We know that the data vector $\vec{f}$ for an image is in the *Range(K)* by the way it is generated, i.e. projecting it with K, so inconsistency is not the source of the problem. These components in $NS(K)$ cannot be recovered by the iteration. so they form a portion of the error in the approximation that, essentially, can not be resolved.

We may summarize our analysis of the matrix K resulting from a square pixel discretization of the image reconstruction problem, noting that the analysis has revealed some significant drawbacks. First, the discretization results in a large, rect-

42
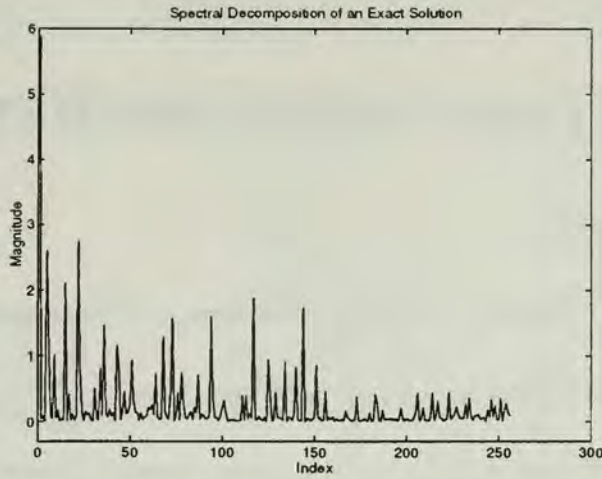
Figure 23. *Spectral decomposition of an exact solution.*

angular, sparse, singular matrix to which the classical iterative methods cannot be applied, so we use the method of Kaczmarz as our solver, which tends to stall after a few iterations. Further analysis using the SVD of K reveals the spectrum of the matrix can be separated into three bands, the resolvable region, near null space, and null space. We find that Kacamarz's method cannot easily resolve components in the near null space, and that it mixes modes, thereby introducing components in the near null space that subsequently cannot be recovered. It is also found that the exact solution itself, when decomposed in terms of the singular vector basis, has components in the null space that can not be recovered by the iteration. In the next chapter we develop a different discretization of the problem that may alleviate some of these problem areas.

43

IV. NATURAL PIXEL DISCRETIZATION

A. DERIVATION

Let $u(x,y)$ be the density function of the image to be reconstructed, and assume it is defined in a square region of unit area. Assume we have an array of detectors to capture the intensity of the x-rays after they have passed through the image. We do not require the detectors to be evenly spaced, but in general they are. Further, assume the x-ray sources are arrayed parallel to the detectors, so that the x-ray paths are perpendicular to the detectors. The sources are not point sources and the detectors are not point detectors, but have non-zero widths, so that an x-ray passing through the image defines a strip and its emerging intensity is entirely detected by the ray's corresponding detector. Finally, assume that the sources and detectors are positioned so that there is always total x-ray coverage of the image by the strips. Now, let the source-detector arrays be rotated through a set of M angles, and a profile measured at each angle. We again hope to reconstruct the image density u from this collection of M profiles.

Next, assume that all sources and detectors have a constant width so that each x-ray passing through the image defines a strip of constant width from source to detector, as depicted in Figure 24. The collection of strips at a given angle completely covers the image, and can be thought of as a set of *pixels* for that angle which are uniquely defined by the x-ray paths. For the M angles we have a collection of these

44

pixels that are discrete and overlapping. As they arise naturally as a result of the geometry used to x-ray the image, they are often refered to as *natural pixels* [Ref. 1].
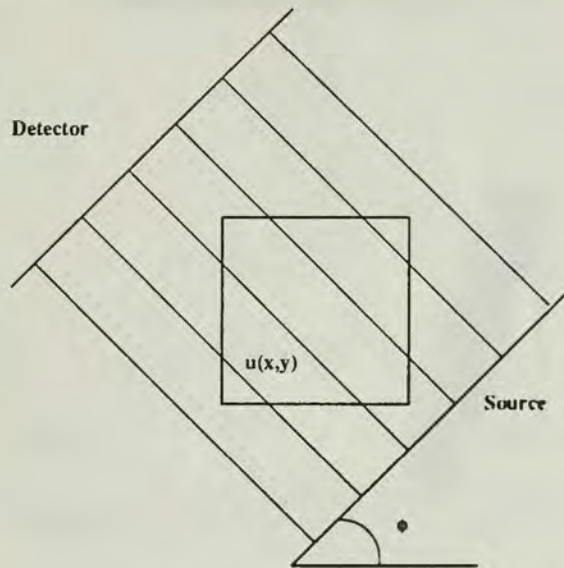
Figure 24. *X-ray strips completely covering the image at a fixed angle.*

Discretization of the problem requires the introduction of *characteristic strip functions* corresponding to the natural pixels. Define $N_1(m)$, *for* $m = 1 : M$, to be the number of sources/detectors for the mth profile whose x-rays pass throught the image. Then $N = \sum_{m=1}^{M} N_1(m)$ is the total number of x-rays passing through the image, and hence the total number of natural pixels, for a particular geometry. Let $\psi_k : \mathcal{R}^2 \to \mathcal{R}$, $1 \leq k \leq N$, be the kth characteristic strip function, where

$$\psi_k = \begin{cases} 1 & \text{for } (x, y) \in S_k \\ 0 & \text{elsewhere} \end{cases}$$

and S_k is the region of the kth strip within the image square. Figure 25 depicts one such characteristic strip function for a given angle ϕ. Note that the strip function ψ_k is nonzero only on the shaded portion, not the entire strip.
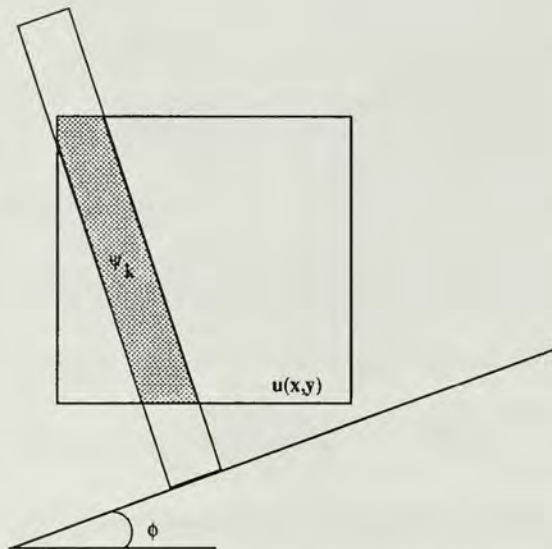
Figure 25. *A representative characteristic strip function.*

Define the operator $A : H \to \mathcal{R}^N$ by

$$Au = \begin{pmatrix} \langle \psi_1, u \rangle \\ \langle \psi_2, u \rangle \\ \vdots \\ \langle \psi_N, u \rangle \end{pmatrix}.$$

Here, $\langle \cdot, \cdot \rangle$ is the standard L_2 inner product and H is some appropriate Hilbert space in which the image u is defined.

The detectors measure x-ray intensity after the ray passes through the image, producing a data vector $\vec{f}$, which can be modeled as

$$Au = \begin{pmatrix} \langle \psi_1, u \rangle \\ \langle \psi_2, u \rangle \\ \vdots \\ \langle \psi_N, u \rangle \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}.$$

The system $Au = \vec{f}$ is underdetermined, so that if it is consistent then there exist infinitely many solutions. We must select one solution for the image, so we choose the minimum norm solution. This is given by $A^*\vec{\alpha}$ [Ref. 18], where $\vec{\alpha} \in \mathcal{R}^N$ solves the system

$$AA^*\vec{\alpha} = \vec{f}.$$

Once we have found the vector $\vec{\alpha}$, then the minimum norm solution is given by

$$u(x,y) = A^*\vec{\alpha}.$$

It is easy to see that the adjointoperator $A^* : \mathcal{R}^N \to H$ is given by

$$A^*\vec{\alpha} = [\psi_1\, \psi_2 \cdots \psi_N] \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \sum_{k=1}^{N} \alpha_k \psi_k(x,y) = u(x,y).$$

Hence, A^* can be viewed as a backprojection operator. It assigns a value α_j to each strip function ψ_j in the image. The strips overlap and the α_j values acculumate

47

additively in the intersections of the strips, ultimately producing a representation of the image. The image density u represented as a linear combination of these characteristic functions by

$$u(x,y) = \sum_{k=1}^{N} \alpha_k \psi_k(x,y) \tag{4.1}$$

defines a grid of *polygons*, on each of which u is constant. A typical grid of these polygons is depicted in Figure 26, which results from a geometry of 20 angles and 32 detectors per view.
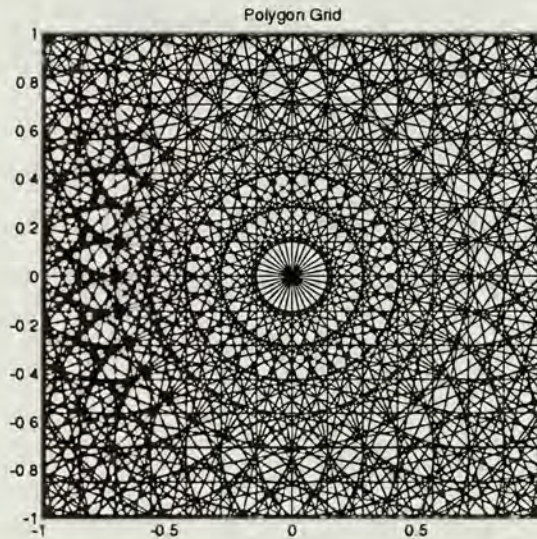
Figure 26. *A representative grid of polygons.*

Define the operator $B : \mathcal{R}^N \to \mathcal{R}^N$ as

$$B = AA^*.$$

Then finding the minimum norm solution given by $A^*\vec{\alpha}$ is equivalent to solving the linear system $B\vec{\alpha} = \vec{f}$. It should be noted that, unlike the case of overdetermined

48

systems, this formulation does not square the condition number of the operator [Ref. 19].

The entries in B can be calculated by substituting (4.1) for u in $Au = \vec{f}$, yielding

$$\int_{\mathcal{R}^2} \psi_j (\sum_{k=1}^{N} \alpha_k \psi_k) = f_j, \; j = 1, \cdots, N,$$

$$\sum_{k=1}^{N} \alpha_k \int_{\mathcal{R}^2} \psi_j \psi_k = f_j, \; j = 1, \cdots, N,$$

$$\sum_{k=1}^{N} \alpha_k \langle \psi_j, \psi_k \rangle = f_j, \; j = 1, \cdots, N.$$

Hence, the $(j, k)^{th}$ entry of B is given by

$$(\beta_{jk}) = \langle \psi_j, \psi_k \rangle.$$

The problem of reconstructing an image from samples of its profiles is now discretized by $B\vec{\alpha} = \vec{f}$, where $B = AA^*$ and the approximate solution to $Ru = f$ is $u = A^*\vec{\alpha}$.

The following simple example will serve to illustrate the discretization process. Assume we have two profiles, one with two detectors and the second with three detectors, oriented and numbered as in the figure below. Note the the image space is of unit area, so that the areas of 1 and 2 are $\frac{1}{2}$ and the areas of 4,5 and 6 are $\frac{1}{3}$.

The matrix B generated from this geometry will have entries corresponding to the areas of intersection of the strips, e.g. element (β_{23}) will be the area of intersection of strip 2 with strip 3, which is $\frac{1}{6}$. The complete matrix is
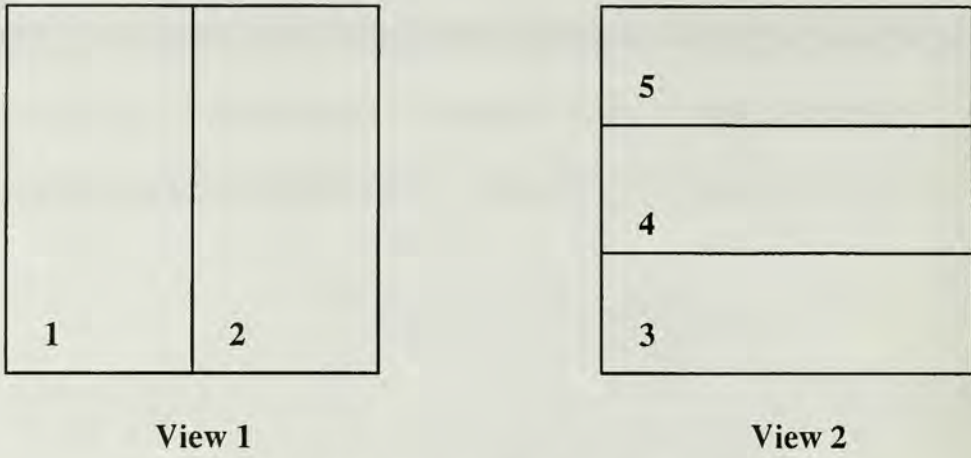
Figure 27. *Geometry for two profiles.*

$$B = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ 0 & \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & 0 & 0 \\ \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{6} & 0 & 0 & \frac{1}{3} \end{pmatrix}.$$

We must only solve the linear system resulting from this process for $\vec{\alpha}$ to reconstruct the image u.

Recall that the continuum problem we are trying to solve is $Ru = f$. In our discretization, we first approximate R with A , yielding $Au = \vec{f}$. We use a finite number of strip integrals taken at a finite number of angles around the image as an approximation to the Radon transform. Next, u is expanded as a linear combination of characteristic strip functions, yielding $AA^*\vec{\alpha} = B\vec{\alpha} = \vec{f}$. Thus a linkage exists between the continuum problem and our discretization.

We examine the strength of this linkage by examining the grid of polygons produced by superimposing the strip functions as depicted in Figure 26. Define Ω as the unit square containing the image, and let $g(x, y)$ be a continuous function defined on Ω. Let ℓ be the total number of polygons in the partition of Ω, generated by strips of constant width over equally spaced angles, and let a_i be the area of the ith polygon, so that $\sum_{i=1}^{\ell} a_i = 1$. Finally, let $\xi_i(x, y)$ be a characteristic function for the ith polygon, so that

$$
\xi_i(x, y) = \begin{cases} 1, & (x, y) \in \quad i^{th} \quad polygon \\ 0, & otherwise. \end{cases}
$$

Then we have the following result, partially attributed to Rhoden [Ref. 20].

Theorem 4.1: *For any $g(x, y)$ and any $\epsilon > 0$ there exists a strip discretization using ℓ strips and a function $f = \sum_{i=1}^{\ell} \alpha_i \xi_i(x, y)$ such that $||f - g||_R < \epsilon$, where*

$$
||u||_R^2 = \sum_{i=1}^{\ell} \int_{\Omega} |u(x, y) \xi_i(x, y)|^2 dx dy.
$$

Proof: Since $g(x, y)$ is continuous, there exists a δ for every ϵ such that $||(x_1, y_1) - (x_2, y_2)|| < \delta$ implies $|g(x_1, y_1) - g(x_2, y_2)| < \epsilon$. Let ϵ be given, and let δ_0 be the δ that implies the continuity conditiion. Let the strip width used in the discretization be $\frac{\delta_0}{\sqrt{2}}$, which in turn determines the number of polygons, which we denote ℓ_0. Then for two angles (which is the fewest allowable) the grid consists of squares with a maximum chord length of δ_0. Adding additional angles cannot increase the maximum chord length. Consider

$$
||f - g||_R^2 = \sum_{i=1}^{\ell_0} \int_{\Omega} |(f - g) \xi_i|^2 dx dy
$$

$$
= \sum_{i=1}^{\ell_0} \int_{\Omega} |f \xi_i - g \xi_i|^2 dx dy
$$

51

$$= \sum_{i=1}^{\ell_0} \int_\Omega |\sum_{j=1}^{\ell_0} \alpha_j \xi_j \xi_i - g\xi_i|^2 dxdy$$

$$= \sum_{i=1}^{\ell_0} \int_\Omega |\alpha_i \xi_i - g\xi_i|^2 dxdy$$

Since g(x,y) is continuous, it attains a maximum M_i and a minimum m_i on each polygon. Choose

$$\alpha_i = \frac{M_i + m_i}{2}.$$

Then, by the intermediate value theorem there exists a point (x_i, y_i) in each polygon so that $g(x_i, y_i) = \alpha_i$. Therefore we have

$$||(x, y) - (x_i, y_i)|| < \delta_0 \rightarrow |g(x, y) - \alpha_i| < \epsilon.$$

So

$$||f - g||_R^2 = \sum_{i=1}^{\ell_0} \int_\Omega |\alpha_i \xi_i - g\xi_i|^2 dxdy$$

$$< \sum_{i=1}^{\ell_0} \int_\Omega \epsilon^2 \xi_i dxdy$$

$$= \epsilon^2 \sum_{i=1}^{\ell_0} a_i$$

$$= \epsilon^2$$

∎

Hence we can approximate any continuous function arbitrarily closely on a polygon grid generated by strip pixels. Since the continuous functions are dense in L_2 [Ref. 21], this result implies that we can arbitrarily closely approximate any function defined on $L_2(\Omega)$ by any continuous function, which in turn can be approximated by a function defined on the polygon grid to arbitrary precision.

We hope to find that the linear system produced by discretizing the problem using natural pixels will have distinct advantages over that produced using a square

pixel discretization. Specifically, we seek a system that lends itself to other iterative solvers than Kaczmarz's method, one that requires less work to solve while still yielding an image of comparable quality, and one that might be approached with a multilevel method. A careful analysis of the matrix B will help us realize these advantages.

B. ANALYSIS OF THE MATRIX B

The matrix B is N × N with entries (β_{jk}) given by $\langle \psi_j, \psi_k \rangle$, for $j, k = 1 : N$. The quantity $\langle \psi_j, \psi_k \rangle$ is just the area of intersection of the jth and k^{th} x-ray strips. Figure 28 illustrates the geometry that would generate entry (β_{jk}) of the matrix.
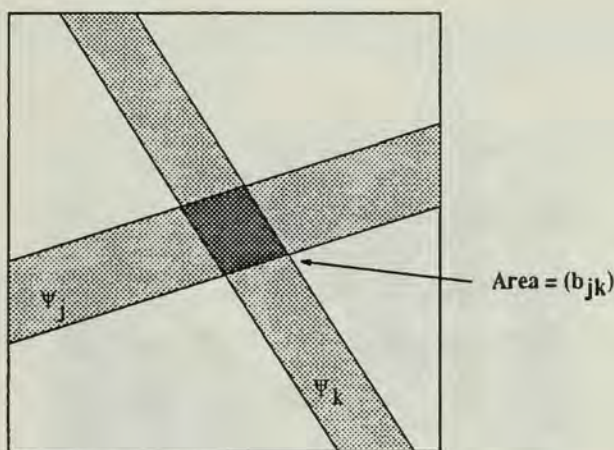
Figure 28. *Intersection of two x-rays inside the image.*

The nature of the non-zero entries of B gives the matrix a rich collection of properties that we can exploit, the first of which is symmetry.

Lemma 4.1: *B is symmetric and nonnegative.*

Proof: $B = (\beta_{ij}) = \langle\psi_i, \psi_j\rangle = \langle\psi_j, \psi_i\rangle = (\beta_{ji}) = B^T$. The non-zero entries of B represent areas of polygons, and as such cannot be negative. ∎

If B were positive definite, then any of the classical iterative methods could be applied to the linear system, with convergence guaranteed [Ref. 22]. We can show that B is positive semidefinite, and later use this fact to show that the Gauss-Seidel method when applied to the problem cannot diverge and in fact must converge.

Theorem 4.2: B *is positive semidefinite.*

Proof: Let $\vec{x} \in \mathcal{R}^N$ be any non-zero vector. Then we can write

$$\vec{x}^T B \vec{x} = x_1 \sum_{i=1}^N \beta_{1i} x_i + x_2 \sum_{i=1}^N \beta_{2i} x_i + \cdots + x_N \sum_{i=1}^N \beta_{Ni} x_i$$

$$= \sum_{j=1}^N [x_j \sum_{i=1}^N \beta_{ji} x_i] = \sum_{j=1}^N \sum_{i=1}^N \beta_{ji} x_j x_i.$$

But $\beta_{ji} = \langle\psi_j, \psi_i\rangle$, so

$$\vec{x}^T B \vec{x} = \sum_{j=1}^N \sum_{i=1}^N \langle\psi_j, \psi_i\rangle x_j x_i$$

$$= \sum_{j=1}^N \sum_{i=1}^N \langle x_j \psi_j, x_i \psi_i\rangle$$

$$= \left\langle \sum_{j=1}^N x_j \psi_j, \sum_{i=1}^N x_i \psi_i \right\rangle = \left\| \sum_{i=1}^N x_i \psi_i \right\|^2 \geq 0.$$

Therefore, B is positive semi-definite. ∎

B has a special block structure, and can be expressed as

$$B = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1M} \\ B_{21} & B_{22} & \cdots & B_{2M} \\ \vdots & \vdots & & \vdots \\ B_{M1} & B_{M2} & \cdots & B_{MM} \end{pmatrix},$$

where each block results from the intersections of all the rays within two given angles. If we define the angles used for the M profiles to be $\phi_1, \phi_2, \cdots \phi_M$, then block B_{ij} is formed by considering the intersections of the rays at angle ϕ_i with those at angle ϕ_j. The size of the block is the number of detectors at angle ϕ_i by the number of detectors at angle ϕ_j, or $N_1(i) \times N_1(j)$. Figure 29 illustrates the block structure of a typical matrix, resulting from a geometry of 8 angles and 16 detectors per view.
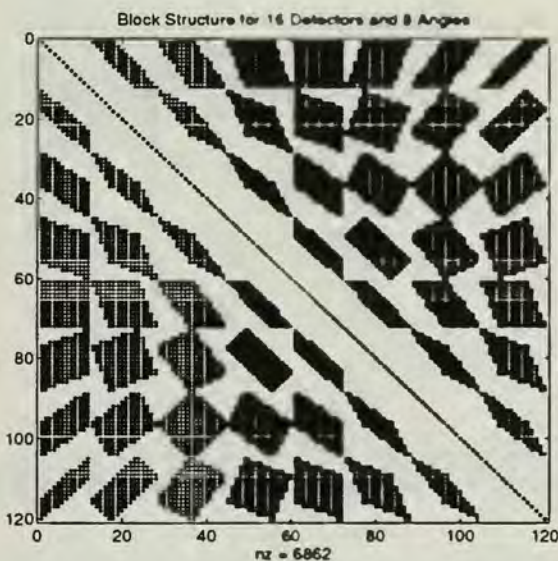
Figure 29. *The block structure of B. Non-zero elements are highlighted.*

When angle ϕ_i equals angle ϕ_j, the only time that the two rays will intersect is when they coincide. Therefore, the blocks B_{ii} are diagonal matrices whose entries

are the areas of the natural pixels at angle ϕ_i. The natural pixels from four detectors at a 45 degree angle are shown in Figure 30.
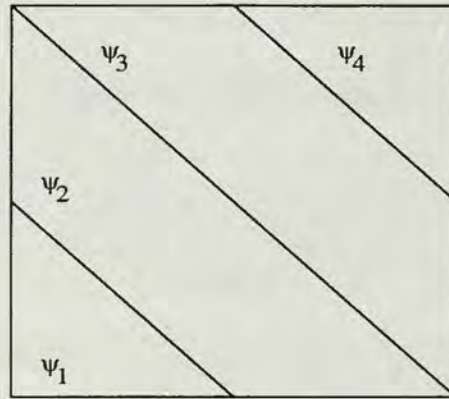
Four Natural Pixels

Figure 30. *Natural pixels from four detectors at angle* $\phi_i = 45$ *degrees.*

The resulting diagonal matrix is

$$B_{ii} = \begin{pmatrix} \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{3}{8} & 0 & 0 \\ 0 & 0 & \frac{3}{8} & 0 \\ 0 & 0 & 0 & \frac{1}{8} \end{pmatrix}.$$

Lemma 4.2: *The elements of the blocks of B exhibit the following summability properties:*

a) *The elements of any diagonal block B_{ii} sum to the area of the image.*

b) *Let $r_k = [\beta_{k1} \ \beta_{k2} \ \cdots \ \beta_{kN}]$ be the kth row of B, and suppose that the kth strip pixel occurs within the set of strips at angle ϕ_i. The elements of r_k in any off-diagonal block B_{ij} sum to b_{kk}, the value of the diagonal element of r_k.*

56

c) Let $c_k = [\beta_{1k}\ \beta_{2k}\ \cdots\ \beta_{Nk}]$ be the kth column of B, and suppose that the kth strip pixel occurs within the set of strips at angle ϕ_j. The elements of c_k in any off-diagonal block B_{ij} sum to b_{kk}, the value of the diagonal element of c_k.

Proof: At angle ϕ_i, diagonal block B_{ii} has entries $\beta_{kk} = \langle \psi_k, \psi_k \rangle = \int_S \psi_k^2(x,y)dxdy = \int_{S_k} dxdy$. Thus β_{kk} is the area of the strip corresponding to ψ_k. By definition, the rays for any angle completely cover the image. Combining these facts proves part a . To prove part b , consider how the elements of row k for the off-diagonal block B_{ij} corresponding to angle ϕ_i are constructed. The entries of row k for this block are the areas of intersection of strip k at angle ϕ_i with all the strips for angle ϕ_j. We know that for angle ϕ_j the rays must completely cover the image, so they must completely cover strip k as well. This geometry is illustrated in Figure 31. Therefore, the sum of these intersections, which are the elements of row k , must sum to the area of ray k , which is the diagonal element of row k . Since B is symmetric, part c follows. ∎
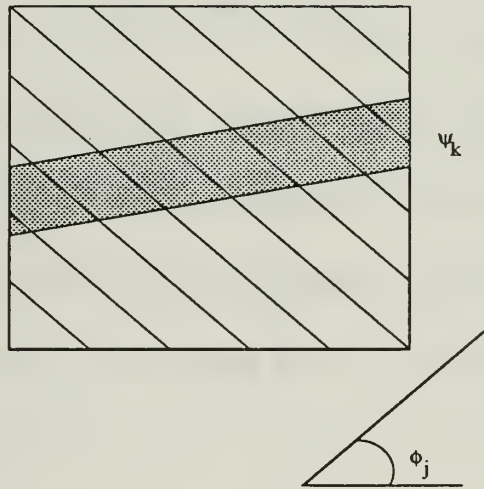
Figure 31. *Geometric interpretation of summability property.*

We know from Theorem 4.2 that B is positive semidefinite, so it can have zero eigenvalues. The summability properties discussed above give us an insight as to the *number* of zero eigenvalues. Consider any row of blocks B_{i1}, $B_{i2} \cdots$, B_{iM}, of B, and sum the indivdual rows into a new composite row. Since the elements of the kth row

or column of any off-diagonal block B_{ij} sum to the corresponding diagonal element b_{kk} of that row or column, this composite row will consist of the N diagonal elements of B. As this property holds for all M rows of blocks, B has *at least* a rank deficiency of *M-1*. We will ultimately show that the rank deficiency is *exactly M-1*, but first further analysis of the structure and properties of B is required.

Let $B \in \mathcal{R}^{N \times N}$ be the block matrix with square diagonal blocks of dimensions n_1, n_2, and n_3 so that $n_1 + n_2 + n_3 = N$, given by

$$
B = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix},
$$

and let $\vec{v} \in \mathcal{R}^N$ be given by

$$
\vec{v} = \begin{pmatrix} \vec{\alpha_1} \\ \vec{\alpha_2} \\ \vec{\alpha_3} \end{pmatrix},
$$

where

$$
\vec{\alpha_1} = \alpha_1 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \vec{\alpha_2} = \alpha_2 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \text{ and } \vec{\alpha_3} = \alpha_3 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},
$$

and dimensions of the vectors $\vec{\alpha_i}$ correspond to the dimensions of the blocks of B,i.e.

58

the length of $\vec{\alpha_i}$ is n_i. Then

$$
B\vec{v} = \begin{pmatrix} (B_{11}\vec{\alpha_1} + B_{12}\vec{\alpha_2} + B_{13}\vec{\alpha_3}) \\ (B_{21}\vec{\alpha_1} + B_{22}\vec{\alpha_2} + B_{23}\vec{\alpha_3}) \\ (B_{31}\vec{\alpha_1} + B_{32}\vec{\alpha_2} + B_{33}\vec{\alpha_3}) \end{pmatrix} .
$$

We wish to define vectors with the properties of $\vec{v}$ in the above example as being constant over each of the M angles used in the generation of the block matrix B. Formally:

Definition 4.1: A vector $\vec{v} \in \mathcal{R}^N$ is *constant by angle* with respect to the block matrix $B \in \mathcal{R}^{N \times N}$ generated over M angles if it consists of M subvectors of constant value α_1, α_2, $\cdots$, α_M corresponding to the block structure of B.

Consider a vector $\vec{v} \in \mathcal{R}^N$ that is constant by angle. If $\vec{v}$ represents the set of coefficients defining a reconstructed image in terms of its natural pixels, then for each of the M angles ϕ_i all of the strips covering the image have the same value α_i. Thus for each angle the contribution to the image is constant, and hence the image over all the angles is constant, with a value equal to the sum of the α_i's.

Writing B as

$$
B = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1N} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2N} \\ \vdots & \vdots & & \vdots \\ \beta_{N1} & \beta_{N2} & \cdots & \beta_{NN} \end{pmatrix} = \begin{pmatrix} \vec{b}_1^T \\ \vec{b}_2^T \\ \vdots \\ \vec{b}_N^T \end{pmatrix} ,
$$

we have the following lemma:

Lemma 4.3: *If $\vec{v} \in \mathcal{R}^N$ is constant by angle with respect to the block matrix B generated over M angles, then $\vec{b_i}^T \vec{v} = \beta_{ii}(\alpha_1 + \alpha_2 + \cdots + \alpha_M)$, where the α_i are the constants in the definition of constant by angle. If $\vec{z} \in \mathcal{R}^N$ is the vector of all ones, then $\vec{b_i}^T \vec{z} = (M)\beta_{ii}$.*

Proof:

$$
\begin{aligned}
\vec{b_i}^T \vec{v} &= \alpha_1 \beta_{i1} + \alpha_1 \beta_{i2} + \cdots + \alpha_1 \beta_{iN_1(1)} \\
&\quad + \alpha_2 \beta_{iN_1(1)+1} + \alpha_2 \beta_{iN_1(1)+2} + \cdots + \alpha_2 \beta_{iN_1(1)+N_1(2)} + \cdots \\
&\quad + \alpha_M \beta_{iN-N_1(M)+1} + \alpha_M \beta_{iN-N_1(M)+2} + \cdots + \alpha_M \beta_{iN} \\
&= \alpha_1 \sum_{k=1}^{N_1(1)} \beta_{ik} + \alpha_2 \sum_{k=N_1(1)+1}^{N_1(2)} \beta_{ik} + \cdots + \alpha_M \sum_{k=N-N_1(M)+1}^{N} \beta_{ik}
\end{aligned}
$$

which after applying Lemma 4.2b can be written as

$$
\begin{aligned}
\vec{b_i}^T \vec{v} &= \alpha_1 \beta_{ii} + \cdots + \alpha_M \beta_{ii} \\
&= \beta_{ii}(\alpha_1 + \alpha_2 + \cdots + \alpha_M).
\end{aligned}
$$

Since $\vec{z}$ is constant by angle with all the $\alpha_i = 1$, the result

$$
\vec{b_i}^T \vec{z} = (M)\beta_{ii}
$$

follows immediately. ∎

Armed with Definition 4.1 and Lemma 4.3, we are prepared to characterize the null space of B. Indeed, we can show that vectors in the null space $NS(B)$ are constant by angle and thus correspond to constant images. We accomplish this with the following theorem, a portion of which is due to Limber [Ref. 23].

Theorem 4.3: *$\vec{v} \in NS(B)$ if and only if $\vec{v}$ is constant by angle and $\sum_{j=1}^{M} \alpha_j = 0$.*

Proof: Let $\vec{v}$ be constant by angle, and let $\sum_{j=1}^{M} \alpha_j = 0$. Then, by Lemma 4.3,

$$
B\vec{v} = \begin{pmatrix} \beta_{11} \sum_{j=1}^{M} \alpha_j \\ \beta_{22} \sum_{j=1}^{M} \alpha_j \\ \vdots \\ \beta_{NN} \sum_{j=1}^{M} \alpha_j \end{pmatrix} = \vec{0}.
$$

Therefore $\vec{v} \in NS(B)$.

Now, let $\vec{v} \in NS(B)$. We first show that $NS(B) = NS(AA^*) = NS(A^*)$. So, assume $\vec{v} \in NS(A^*)$. Then $\vec{v} \in NS(AA^*)$ and $AA^*\vec{v} = \vec{0}$. If $\vec{v} \in NS(A^*)$, then $\vec{v} \in NS(AA^*)$ since $A(A^*\vec{v}) = A(0) = \vec{0}$. There are no $\vec{v} \notin NS(A^*)$ such that $AA^*\vec{v} = \vec{0}$, because if $AA^*\vec{v} = \vec{0}$, then $A^*\vec{v} \in Range(A^*)$. We know $Range(A^*) \perp NS(A)$ and therefore $A^*\vec{v} \notin NS(A))$. Then

$$A^*\vec{v} = [\psi_i(x,y) \ \psi_2(x,y) \ \cdots \ \psi_N(x,y)] \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = \sum_{i=1}^{N} v_i\psi_i(x,y) = 0.$$

Consider two adjacent strips S_k and S_{k+1} from the same profile. Now, select points $(x_1, y_1) \in S_k \cap \Omega$ and $(x_2, y_2) \in S_{k+1} \cap \Omega$, where Ω is the intersection of $(M-1)$ strips, one from each profile and none from the profile containing S_k and S_{k+1}. This selection can always be done. To see why, superimpose all of the x-ray path strips over the image at once. They subdivide the image into a collection of polygons. Since each profile completely covers the image, a point in the interior of any polygon is contained in one strip from each of the M profiles. A point on any edge (not a vertex) of any polygon will be contained in Ω, with the edge separating strips S_k and S_{k+1}. Moving a distance ϵ to either side and perpendicular to the edge will define the points (x_1, y_1) and (x_2, y_2). Figure 32 illustrates this geometrically.
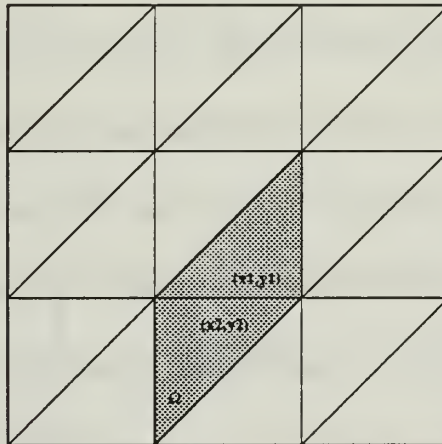
Figure 32. *Geometric representation of the proof of Theorem 4.3.*

Since $\vec{v} \in NS(A^*)$, we have

$$\sum_{i=1}^{N} v_i \psi_i(x_1, y_1) \; = \; \sum_{i=1}^{N} v_i \psi_i(x_2, y_2) \; = \; 0.$$

The $\psi_i(x, y)$ are characteristic functions, so we can rewrite the above sums as

$$\sum_{j \in \Omega} v_j + v_k \; = \; \sum_{j \in \Omega} v_j + v_{k+1} \; = \; 0,$$

which implies $v_k \; = \; v_{k+1}$. Therfore, $\vec{v}$ is constant by angle. Further, since $\sum_{j \in \Omega} v_j + v_k \; = \; 0$ and there is a v_j from each of the M profiles in the sum, the constants from the definition of constant by angle sum to zero. ∎

An immediate consequence of Theorem 4.3 is that we can show the dimension of $NS(B)$ is exactly *M-1*.

Theorem 4.4: *Let $B \in \mathcal{R}^{N \times N}$ be the block matrix generated over M angles as discussed above. Then the dimension of $NS(B) \; = \; M - 1$.*

Proof: Let $\vec{v} \in NS(B)$. Then $\vec{v}$ is constant by angle and $\sum_{i=1}^{M} \alpha_i \; = \; 0$. We can select α_1, α_2, $\cdots$, α_{M-1} arbitrarily, and then α_M is determined. Hence we have $M - 1$ degrees of freedom in the selection of the α_i's, which is the dimension of $NS(B)$. ∎

Since we have a matrix operator with a non-trivial null space, it is conceivable that in solving $B\vec{\alpha} \; = \; \vec{f}$, any solution $\vec{\alpha}$ may have components in $NS(B)$. As we are ultimately reconstructing an image, $u \; = \; A^* \vec{\alpha}$, the effect such null space components have on the appearance of the reconstructed image is of great importance. We will completely characterize $NS(B)$ by constructing a basis for it.

We can construct a basis $[\vec{q_1}, \vec{q_2}, \cdots, \vec{q}_{M-1}]$ for *NS(B)* in the following fashion. All of the $\vec{q_i}$ will be constant by angle, with the constants α_j for each vector defined as follows:

$$\vec{q}_j = \begin{pmatrix} \vec{\alpha}_1 \\ \vec{\alpha}_2 \\ \vdots \\ \vec{\alpha}_M \end{pmatrix}, \text{ where } \vec{\alpha}_i \in \mathcal{R}^{n_i},$$

and

$$\vec{\alpha}_i = \begin{cases} (1\ 1\ \cdots\ 1)^T, & i = 1 \\ (-1\ -1\ \cdots\ -1)^T, & i = j+1 \\ (0\ 0\ \cdots\ 0)^T, & \text{otherwise} \end{cases}$$

Each of these vectors is constant by angle, and $\sum_{j=1}^M \alpha_j = 0$ for each, by construction, so $\vec{q}_i \in NS(B)$. It is also apparent that the *M-1* vectors form a linearly independent set. Therefore they form a basis for *NS(B)*, so that any $\vec{v} \in NS(B)$ has such a representation. Let $\vec{v}$ be such a vector. It can be expressed as

$$\vec{v} = \sum_{j=1}^{M-1} \gamma_j \vec{q}_j.$$

This expression can be further expanded as

$$\vec{v} = \begin{pmatrix} (\gamma_1 + \gamma_2 + \cdots + \gamma_{M-1})\vec{\alpha}_1 \\ -\gamma_1 \vec{\alpha}_2 \\ -\gamma_2 \vec{\alpha}_3 \\ \vdots \\ -\gamma_{M-1}\vec{\alpha}_M \end{pmatrix},$$

63

where $\vec{\alpha}_i \in \mathcal{R}^{n_i}$ is given by

$$\vec{\alpha}_i = (1\ 1\ \cdots\ 1)^T.$$

It is clear that $\vec{v}$ is constant by angle, and that these constants sum to zero.

Any vector $\vec{v} \in \mathcal{R}^N$ that is constant by angle, (and particularly those in $NS(B)$), defines the coefficients of the natural pixel representation of a constant image. Such images, when displayed, appear as uniform shades of grey. We define a natural pixel with a value of zero to be *black* , and an image that displays to uniform black as *invisible*.Based on these definitions, we have the following:

Corollary 4.3: *If $\vec{v} \in NS(B)$, then the image defined by $u(x,y) = A^*\vec{v}$ is invisible.*

Proof: Since $\vec{v} \in NS(B)$, it it constant by angle, and the sum of these constants is zero. Now $u(x,y) = A^*\vec{v} = \sum_{j=1}^N v_j \psi_j$. Since $\vec{v}$ is constant by angle, for each angle i the natural pixels will all assume the same value to form a uniform gray sub-image in the image space. Since the constants that define each of these uniform grey sub-images sums to zero, the combination of the sub-images into the image $u(x,y)$ will be a uniform image in the image space with value zero, which by definition is invisible. ∎

These results are significant. Since components of the solution in the $NS(B)$ are invisible, we need not concern ourselves with them in the framework of reconstructing the image. Thus, if the iterative technique chosen to solve the system of equations $B\vec{\alpha} = \vec{f}$ excites components in $NS(B)$,as far as the display of the image is concerned we do not care. (Such components do not affect the residual calculation either).

Our goal was to develop a discretization that resulted in a linear system that could be solved with less work than the standard square pixel discretization, and

could produce comparable results. To that end, we now compare the two resulting linear systems, and address some other important issues.

C.   NATURAL VS. SQUARE PIXELS

In general, the matrix B generated by using natural pixels will be smaller than that generated using square pixels, for the same source/detector geometry. If we assume M angles and $N_1(m)$ detectors, $m = 1 : M$, then $N = \sum_{m=1}^{M} N_1(m)$ is the number of rays passing through the image, and B is an $N \times N$ matrix. Recall that for the square pixel discretization, we divide the image space into a square grid of n^2 pixels. The resulting matrix K is of size $N \times n^2$. Since smaller square pixels yield a higher resolution image, in general $n^2 > N$ and often $n^2 >> N$. Thus, the matrix B for a problem discretized by natural pixels is normally smaller than the matrix K for a problem discretized by square pixels, and may require less work to solve.

On the other hand, K is very sparse when compared to B. Recall that non-zero entries in a row of K correspond to pixels being intersected by a given x-ray. If the width of the ray is small, say on the order of the width of a pixel, then each row will contain no more than $2n$ non-zero entries. If this sparseness is exploited, then the work needed to solve the system can be greatly reduced.

Additionally, K is in general a rectangular matrix, and the only practical iterative method for solving the resulting system of equations is Kaczmarz's method [Ref. 17]. B,however, is square and symmetric and possesses all of the other properties discussed above. This gives us more selection when choosing an iterative method to

solve the accompanying system of equations.

One very significant drawback of discretizing by natural pixels is that the natural pixels do not map to the square pixel hardware of a computer screen easily. One way to display the image is now outlined. We must map the natural pixels to a rectangular grid that can then be illuminated on a computer screen. This seems at first a difficult task, but it can be easily accomplished through the use of a square pixel discretization matrix K created at the same geometry used to generate B .

The problem we are attempting to solve is $Ru = f$, where R is the Radon transform operator. Using the square pixel discretization, we discretize both the projection and image spaces and arrive at a linear system $K\vec{x} = \vec{f}$. When we discretize using natural pixels, only the projection space is discretized, and the density function $u(x, y)$ is expanded in terms of strip functions, yielding the system $B\vec{\alpha} = AA^*\vec{\alpha} = \vec{f}$. Now, if we view $\vec{x}$ as an approximation of $A^*\vec{\alpha}$, and K as an approximation for A , then we can write

$$\vec{x} = K^T \vec{\alpha}.$$

Thus, the natural pixels can be mapped to square pixels just by backprojecting $\vec{\alpha}$ with the K matrix.

We only require the geometries on number of detectors and angles to agree for both discretizations. The natural pixel problem is independent of the number of square pixels to which it is being mapped. Therefore, we can map to any resolution, as long as the x-ray geometries agree.

The cost involved to display the image is primarily due to building the matrix K, which can be excessive if high resolution is required. Fortunately, this is a one-time cost, for K can be repeatedly used to display different images generated at the same geometry. Another drawback is that K might be too large to fit in memory, so storage and file-handling procedures will be required when displaying the image that might not be necessary when working with the matrix B. Also, a different K is required for each geometry used to generate B, or for different resolutions at the same geometry.

We have developed the natural pixel discretization for the image reconstruction problem, and have shown that it produces a linear system whose matrix B is symmetric and in general smaller than that produced by the conventional square pixel discretization. An analysis of this matrix produced several interesting results. The rank of B is a function of the x-ray geometry used to generate it. The $NS(B)$ is characterized by vectors that are constant by angle with constants summing to zero, and the images they represent are invisible. We now concentrate on solving this linear system.

V. GAUSS-SEIDEL RELAXATION

There are two basic strategies for solving a linear system of equations $A\vec{x} = \vec{b}$, either directly or iteratively. The direct approach generally involves a factorization of the coefficient matrix A. Iterative methods, on the other hand, generate a sequence of approximate solutions $\vec{x}^{(k)}$ and only involve the matrix A for matrix-vector multiplications. If A is large, the direct approach could be impractical because one direct solve costs as much as many iteration sweeps, in terms of floating point operations. The effectiveness of an iterative method is determined by how fast the sequence of iterates $\vec{x}^{(k)}$ converges [Ref. 19]. In this chapter we will apply the *Gauss-Seidel* iterative method and analyze it in the context of our image reconstruction problem.

A. DERIVATION AND GENERAL PROPERTIES

We derive the Gauss-Seidel iterative method by considering how to solve the linear system of equations

$$Q\vec{x} = \vec{b},$$

where Q is an $N \times N$ matrix assumed to be nonsingular with nonzero diagonal entries. Then the formal solution is

$$\vec{x} = Q^{-1}\vec{b}.$$

Considering a splitting of the matrix Q into an upper triangular part U, a lower triangular part L, and a diagonal part D, so that

$$Q = D - L - U.$$

We can now write the system of equations as

$$Q\vec{x} = (D - L - U)\vec{x} = \vec{b},$$

and rearranging, arrive at

$$(D - L)\vec{x} = U\vec{x} + \vec{b}. \tag{5.1}$$

We convert (5.1) to an iterative form by introducing a superscript to the vector $\vec{x}$ representing its index in the sequence of approximations, as follows

$$(D - L)\vec{x}^{(k+1)} = U\vec{x}^{(k)} + \vec{b}, \quad k \geq 0. \tag{5.2}$$

Hence the $(k+1)^{st}$ approximation is generated from the kth approximation, and the process is started by providing an initial guess $\vec{x}^{(0)}$. Writing (5.2) in terms of the elements of the matrix Q produces

$$q_{ii}x_i^{(k+1)} - \sum_{j=1}^{i-1} q_{ij}x_j^{(k+1)} = \sum_{j=i+1}^{N} q_{ij}x_j^{(k)} + b_i, \quad 1 \leq i \leq N, \tag{5.3}$$

which can be further rearranged to yield

$$x_i^{(k+1)} = \frac{1}{q_{ii}} \left(b_i - \sum_{j=1}^{i-1} q_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{N} q_{ij}x_j^{(k)} \right), \quad 1 \leq i \leq N, \tag{5.4}$$

since the diagonal elements of A are nonzero. Equation (5.4) can be written in algorithm form as

For $i = 1, 2, \cdots, N$

$$x_i^{(k+1)} = \frac{1}{q_{ii}} \left(b_i - \sum_{j=1}^{i-1} q_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{N} q_{ij} x_j^{(k)} \right),$$

which defines one iteration of the Gauss-Seidel method. Recasting (5.4) once again in matrix notation, we get

$$\vec{x}^{(k+1)} = (D - L)^{-1} U \vec{x}^{(k)} + (D - L)^{-1} \vec{b}, \quad k \geq 0. \tag{5.5}$$

Defining $P_G = (D - L)^{-1} U$ and $\vec{c} = (D - L)^{-1} \vec{b}$, the Gauss-Seidel method (5.5) becomes

$$\vec{x}^{(k+1)} = P_G \vec{x}^{(k)} + \vec{c}, \quad k \geq 0.$$

The matrix P_G is known as the *Gauss-Seidel iteration matrix* .

Convergence of the Gauss-Seidel method can be analyzed in terms of the *error vector*, defined as

$$\vec{e}^{(k+1)} = \vec{x}^{(k+1)} - \vec{x}^*.$$

where $\vec{x}^*$ is the exact solution of the problem. Since $\vec{x}^*$ is the exact solution, then $\vec{x}^* = P_G \vec{x}^* + \vec{c}$, so that

$$
\begin{aligned}
\vec{e}^{(k+1)} &= \vec{x}^{(k+1)} - \vec{x}^* \\
&= P_G \vec{x}^{(k)} + \vec{c} - (P_G \vec{x}^* + \vec{c}) \\
&= P_G (\vec{x}^{(k)} - \vec{x}^*) \\
&= P_G \vec{e}^{(k)}.
\end{aligned}
$$

Thus we arrive at the relationship

$$\vec{e}^{(k+1)} = P_G \vec{e}^{(k)}. \tag{5.6}$$

70

Recursive applications of (5.6) yields

$$\vec{e}^{(k+1)} = P_G \vec{e}^{(k)} = (P_G)^2 \vec{e}^{(k-1)} = \cdots = (P_G)^{k+1} \vec{e}^{(0)}.$$

Thus, we get convergence if $(P_G)^{k+1} \to 0$ as $k \to \infty$, and this convergence is independent of the initial guess $\vec{x}^{(0)}$.

Define the *spectral radius* of $P_G = \rho(P_G)$ to be the magnitude of the largest eigenvalue of P_G. The following well-known theorems [Ref. 22] give conditions under which the Gauss-Seidel method is guaranteed to converge:

Theorem 5.1: $(P_G)^{k+1} \to 0$ *as* $k \to \infty$ *if and only if* $\rho(P_G) < 1$.

Theorem 5.2: *If* $A \in \mathcal{R}^{N \times N}$ *is symmetric and positive definite, and* P_G *is the Gauss-Seidel iteration matrix formed from A, then* $\rho(P_G) < 1$.

Thus, a condition that ensures convergence of the Gauss-Seidel method to is that the matrix defining the system of equations to be solved be symmetric and positive definite. The matrix B from the natural pixel discretization is symmetric and positive semi-definite, so, apparently, Gauss-Seidel applied to it is not guaranteed to converge. However, it will be shown that Gauss-Seidel is convergent for the matrix B, and why this must be so.

B.   CONVERGENCE ANALYSIS

First, we show that the Gauss-Seidel method cannot diverge for this problem. We introduce the following definitions:

71

Definition 5.1: The *energy inner product* can be defined as $(x, y) = \langle Gx, y \rangle$, where G is semidefinite.

Definition 5.2: The *energy norm* is defined as $|||x||| = (x, x)^{1/2}$.

The energy norm is actually a *semi-norm*, because $|||x||| = 0$ does not imply that $x = 0$. We will show that when measured with the energy norm, Gauss-Seidel cannot diverge.

Recall that (5.4) is

$$x_i^{(k+1)} = \frac{1}{q_{ii}} \left(b_i - \sum_{j=1}^{i-1} q_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(k)} \right), \quad 1 \le i \le N,$$

and after further manipulation we obtain

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{q_{ii}} \left(b_i - \sum_{j=1}^{i-1} q_{ij} x_j^{(k+1)} - \sum_{j=i}^{N} q_{ij} x_j^{(k)} \right). \tag{5.7}$$

Let

$$r_i = b_i - \sum_{j=1}^{i-1} q_{ij} x_j^{(k+1)} - q_{ii} x_i^{(k)} - \sum_{j=i+1}^{N} q_{ij} x_j^{(k)}, \quad 1 \le i \le N$$

be the ith component of the *current* residual vector $\vec{r}$ during the kth step of the iteration. That is, it is computed using both $\vec{x}^{(k+1)}$ and $\vec{x}^{(k)}$, and changes for each value of i. Substituting into (5.7) results in

$$x_i^{(k+1)} = x_i^{(k)} - \frac{r_i}{q_{ii}}, \quad 1 \le i \le N. \tag{5.8}$$

Let $\vec{w}_i$ to be the ith *standard basis vector* for $\mathcal{R}^{N \times N}$, that is, the vector with a 1 in the ith position and zeros in all other positions. Then we can express the diagonal elements of Q as

$$q_{ii} = \vec{w}_i^T Q \vec{w}_i = \langle Q \vec{w}_i, \vec{w}_i \rangle$$

and r_i as

$$r_i = \vec{r}^T \vec{w}_i = \langle \vec{r}, \vec{w}_i \rangle.$$

Writing (5.8) in terms of these inner products produces

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\langle \vec{r}, \vec{w}_i \rangle}{\langle Q\vec{w}_i, \vec{w}_i \rangle}, \quad 1 \le i \le N. \tag{5.9}$$

Only one component, the ith, of the approximation is being changed at each step, so that (5.9) can be written in terms of vectors as

$$\vec{x}^{(k+1)} \leftarrow \vec{x}^{(k)} - \frac{\langle \vec{r}, \vec{w}_i \rangle}{\langle Q\vec{w}_i, \vec{w}_i \rangle} \vec{w}_i, \quad 1 \le i \le N. \tag{5.10}$$

where the arrow indicates replacement, or overwriting. The above formulations are used to yield the following result:

Theorem 5.3: *The Gauss-Seidel method applied to $B\vec{\alpha} = \vec{f}$, where B is the matrix generated via discretization by natural pixels, is bounded in the energy norm.*

Proof: Define the *error vector* as

$$\vec{e}^{(k)} = \vec{\alpha}^* - \vec{\alpha}^{(k)}. \tag{5.11}$$

where $\vec{\alpha}^*$ is an exact solution, and note that

$$B\vec{e}^{(k)} = B(\vec{\alpha}^* - \vec{\alpha}^{(k)}) = B\vec{\alpha}^* - B\vec{\alpha}^{(k)} = \vec{f} - B\vec{\alpha}^{(k)} = \vec{r}. \tag{5.12}$$

Taking the energy norm of the error vector gives

$$|||\vec{e}^{(k+1)}|||^2 = \langle B\vec{e}^{(k+1)}, \vec{e}^{(k+1)} \rangle,$$

and applying (5.10), (5.11) and (5.12) to the right side produces

$$|||\vec{e}^{(k+1)}|||^2 = \left\langle B \left(\vec{e}^{(k)} - \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} \vec{w}_i \right), \left(\vec{e}^{(k)} - \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} \vec{w}_i \right) \right\rangle.$$

Some algebraic manipulations produce the following

$$|||\vec{e}^{(k+1)}|||^2 \;=\; \left\langle B\vec{e}^{(k)}, \left(\vec{e}^{(k)} - \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} \vec{w}_i \right) \right\rangle$$

$$- \left\langle \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} B\vec{w}_i, \left(\vec{e}^{(k)} - \frac{< B\vec{e}^{(k)}, \vec{e}_i >}{< B\vec{w}_i, \vec{w}_i >} \vec{w}_i \right) \right\rangle,$$

which can be expanded as

$$|||\vec{e}^{(k+1)}|||^2 \;=\; < B\vec{e}^{(k)}, \vec{e}^{(k)} > - \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} < B\vec{e}^{(k)}, \vec{w}_i >$$

$$- \frac{< B\vec{e}^{(k)}, \vec{w}_i >}{< B\vec{w}_i, \vec{w}_i >} < B\vec{w}_i, \vec{e}^{(k)} > + \frac{< B\vec{e}^{(k)}, \vec{w}_i >^2}{< B\vec{w}_i, \vec{w}_i >^2} < B\vec{w}_i, \vec{w}_i > .$$

Simplification results in

$$|||\vec{e}^{(k+1)}|||^2 \;=\; < B\vec{e}^{(k)}, \vec{e}^{(k)} > - 2\frac{< B\vec{e}^{(k)}, \vec{w}_i >^2}{< B\vec{w}_i, \vec{w}_i >} + \frac{< B\vec{e}^{(k)}, \vec{w}_i >^2}{< B\vec{w}_i, \vec{w}_i >}$$

$$= \; < B\vec{e}^{(k)}, \vec{e}^{(k)} > - \frac{< B\vec{e}^{(k)}, \vec{w}_i >^2}{< B\vec{w}_i, \vec{w}_i >}$$

$$= \; |||\vec{e}^{(k)}|||^2 - \frac{< B\vec{e}^{(k)}, \vec{w}_i >^2}{< B\vec{w}_i, \vec{w}_i >}.$$

Since B is positive semidefinite and

$$< B\vec{w}_i, \vec{w}_i > \;=\; b_{ii} > 0,$$

we have

$$|||\vec{e}^{(k+1)}|||^2 \leq |||\vec{e}^{(k)}|||^2.$$

Therefore, the norms remain bounded. ∎

This useful result assures us that the Gauss-Seidel iteration can be applied to $B\vec{\alpha} = \vec{f}$, and provided we use the energy norm to measure performance we need not fear a divergent process. To obtain a stronger (convergent) result, we show that under certain conditions Gauss-Seidel is equivalent to Kaczmarz's method, and that

74

the convergence of Kaczmarz established in Theorem 3.1 carries over to Gauss-Seidel applied to the natural pixel problem.

Consider applying Kaczmarz's method to $Au = \vec{f}$, where $A : H \to \mathcal{R}^N$, $u \in H$, $\vec{f} \in \mathcal{R}^N$, H is a real Hilbert space spanned by the constant-valued strip functions, and A is defined (see page 47) so that

$$Au = \begin{pmatrix} \langle \psi_1, u \rangle \\ \langle \psi_2, u \rangle \\ \vdots \\ \langle \psi_N, u \rangle \end{pmatrix}.$$

If $u^{(0)} \in \text{Range}(A^*)$, then the sequence $u^{(k)}$ generated by Kaczmarz's method converges to the minimum L_2 norm solution as $k \to \infty$. One sweep of Kaczmarz can be expressed as

Set $u = u^{(k)}$

For $i = 1, 2, \cdots, N$

Solve $< \vec{w}_i, A(u + sA^*\vec{w}_i) - \vec{f} >= 0$ for s

Set $u = u + sA^*\vec{w}_i$.

Set $u^{(k+1)} = u$

Solving for s in the above algorithm

$$\begin{aligned} 0 &= < \vec{w}_i, A(u + sA^*\vec{w}_i) - \vec{f} > \\ &= < \vec{w}_i, Au + sAA^*\vec{w}_i - \vec{f} > \end{aligned}$$

$$= \vec{w_i}^T A u + s \vec{w_i}^T A A^* \vec{w_i} - \vec{w_i}^T \vec{f}$$

$$= <u, \psi_i> + s <\psi_i, \psi_i> - f_i,$$

yielding

$$s = \frac{f_i - \langle \psi_i, u \rangle}{< \psi_i, \psi_i >}.$$

Thus, Kaczmarz's method becomes

Set $u = u^{(k)}$

For $i = 1, 2, \cdots, N$

$$u = u + \frac{f_i - \langle \psi_i, u \rangle}{\langle \psi_i, \psi_i \rangle} \psi_i.$$

Set $u^{(k+1)} = u$

Now, consider applying the Gauss-Seidel method to the natural pixel problem $B\vec{\alpha} = \vec{f}$, where $B = AA^*$ and $u = A^*\vec{\alpha}$. Note that $u \in \text{Range}(A^*)$. Gauss-Seidel on this problem can be written as

Set $\vec{\alpha} = \vec{\alpha}^{(k)}$

For $i = 1, 2, \cdots, N$

Solve $< \vec{w_i}, B(\vec{\alpha} + s\vec{w_i}) - \vec{f} > = 0$ for s

Set $\vec{\alpha} = \vec{\alpha} + s\vec{w_i}$.

Set $\vec{\alpha}^{(k+1)} = \vec{\alpha}$.

Again, we can solve for s in the above algorithm. Let

76

$$B = \begin{pmatrix} \vec{b}_1^{\,T} \\ \vec{b}_2^{\,T} \\ \vdots \\ \vec{b}_N^{\,T} \end{pmatrix}, \quad \text{where} \quad \vec{b}_i^{\,T} = (\beta_{i1} \; \beta_{i2} \; \cdots \; \beta_{iN}).$$

Then, solving for s , we obtain

$$
\begin{aligned}
0 &= \; < \vec{w}_i, B(\vec{\alpha} + s\vec{w}_i) - \vec{f} > \\
&= \; < \vec{w}_i, B\vec{\alpha} + sB\vec{w}_i - \vec{f} > \\
&= \; \vec{w}_i^{\,T} B\vec{\alpha} + s\vec{w}_i^{\,T} B\vec{w}_i - \vec{w}_i^{\,T} \vec{f} \\
&= \; \vec{b}_i^{\,T} \vec{\alpha} + s\beta_{ii} - f_i,
\end{aligned}
$$

yielding

$$s = \frac{f_i - \vec{b}_i^{\,T} \vec{\alpha}}{\beta_{ii}}.$$

But

$$\vec{b}_i^{\,T} \vec{\alpha} = \; < A^{\bullet}\vec{\alpha}, \psi_i > = \; < u, \psi_i > \quad \text{and} \quad \beta_{ii} = \; < \psi_i, \psi_i >,$$

so

$$s = \frac{f_i - \; < u, \psi_i >}{< \psi_i, \psi_i >},$$

and after substituting into the Gauss-Seidel algorithm we can write

$$\vec{\alpha} = \vec{\alpha} + \frac{f_i - \; < u, \psi_i >}{< \psi_i, \psi_i >} \vec{w}_i.$$

Multipliying both sides by A^* yields

$$A^*\vec{\alpha} = A^*\vec{\alpha} + \frac{f_i - <u,\psi_i>}{<\psi_i,\psi_i>}A^*\vec{w_i}.$$

But $A^*\vec{\alpha} = u$, so we arrive at

$$u = u + \frac{f_i - \langle\psi_i, u\rangle}{\langle\psi_i, \psi_i\rangle}\psi_i.$$

Therefore Gauss-Seidel applied to $B\vec{\alpha} = \vec{f}$ is equivalent to Kaczmarz applied to $Au = \vec{f}$ [Ref. 2]. This observation yields the following result:

Theorem 5.4: *If $\vec{\alpha}^{(0)} \in Range(B^T)$, then the sequence $\vec{\alpha}^{(k)}$ generated by the Gauss-Seidel method converges to $\vec{\hat{\alpha}}$ such that $\hat{u} = A^*\vec{\hat{\alpha}}$ is the minimum l_2 norm solution of $Au = \vec{f}$ as $k \to \infty$, provided such a solution exists.*

Proof: Assume $B\vec{\alpha} = \vec{f}$ is consistent. If $\vec{\alpha}^{(0)} \in Range(B^T)$, then

$$B\vec{\alpha^{(0)}} = AA^*\vec{\alpha}^{(0)},$$

so we can define $u^{(0)} = A^*\vec{\alpha}^{(0)} \in Range(A^*)$. By Theorem 3.1, we know that the sequence $u^{(k)}$ generated by Kaczmarz's method converges to the minimum L_2 norm solution, $\hat{u}$, of $Au = \vec{f}$ as $k \to \infty$. But Kaczmarz is equivalent to Gauss-Seidel under these conditions, so Gauss-Seidel must converge to some $\vec{\hat{\alpha}}$ such that $\hat{u} = A^*\vec{\hat{\alpha}}$. ∎

Next, we show that the problem $Au = \vec{f}$ can always be made to be consistent so that a solution exists, and that we can always produce an inital guess in the $Range(B^T)$. Consider a projection of the image at a given angle, that is, the collection of emergent x-ray intensities measured by the array of detectors. The natural pixel discretization assumes complete coverage of the image by x-rays at each angle. Noting

78

that the density of the image remains unchanged from angle to angle, and that the initial x-ray intensities are constant from angle to angle, then the total emergent x-ray intensities for each projection must be constant as well. This must be the case as the image will absorb the same amount of energy independent of the direction of that energy. We say that the data $\vec{f}$ is *compatible* with the image reconstruction problem when the total emergent x-ray intensities are constant for each angle, that is, the sum of the elements in the piece of $\vec{f}$ corresponding to each angle, is constant. Given this definition, we have the following

Theroem 5.5: *If the data $\vec{f}$ is compatible with the image reconstruction problem, then $Au = \vec{f}$ is consistent and has a solution.*

Proof: Let $\vec{f}$ be compatible. Then $\vec{f} \notin NS(B)$ because it cannot be constant by angle with the $\sum_{i=1}^{M} \alpha_i = 0$ and compatible at the same time. Therefore $\vec{f} \notin NS(A^*)$. Assume the sum of the elements in $\vec{f}$ corresponding to each angle sum to the constant C. Let $\vec{q_i}$ be the ith basis vector for NS(B) as constructed in Chapter IV, and consider $< \vec{f}, \vec{q_i} >$. If the inner product is computed by angle, the resulting sum is

$$< \vec{f}, \vec{q_i} > = C + 0 + 0 + \cdots + 0 - C + 0 + \cdots + 0 = 0,$$

where the negative entry is in the $(i+1)^{st}$ position. Therefore $\vec{f}$ is orthogonal to each basis vector of NS(B) and hence $\vec{f} \perp NS(B) = NS(A^*)$. Thus $\vec{f} \in Range(A)$. ∎

If for some reason the data we are given to reconstruct is not compatible due to measurement errors, we can *correct* the data and enforce compatibility by adding some constant ϵ_i to each of the elements of the ith view. Thus we can always ensure a consistent system, and theoretically Gauss-Seidel will converge.

Next, we use the concept of compatibilty to define an initial guess for the Gauss-Seidel iteration. Recall that if $\vec{f}$ is compatible, then the sum of its elements

over any angle is constant, call it C. Consider a vector that is constant by angle, where the constants are such that

$$C = \sum_{i=1}^{M} \alpha_i.$$

We define such a vector to be the *average grey* representation of the image whose data vector is $\vec{f}$. There is an infinite set of constants α_i that can define an average grey representation of a given image. We show that at least one of them defines a vector in the $Range(B^T)$.

Lemma 5.1: *Let $\vec{v} \in \mathcal{R}^N$ be an average grey representation of some image $u(x,y)$ with constants α_i such that*

$$\alpha_i = \frac{C}{M}$$

for all i. Then $\vec{v} \in Range(B^T)$.

Proof: Let $\bar{\alpha}$ be the arithmetic mean of the constants α_i in the definition of constant by angle. Now $\vec{v}$ can be written as

$$\vec{v} = \vec{v}_R + \vec{v}_N,$$

where $\vec{v}_R \in Range(B^T)$ and $\vec{v}_N \in NS(B)$. We claim this decomposition can be expressed as

$$\begin{pmatrix} \vec{\alpha_1} \\ \vec{\alpha_2} \\ \vdots \\ \vec{\alpha_M} \end{pmatrix} = \begin{pmatrix} \vec{\bar{\alpha}} \\ \vec{\bar{\alpha}} \\ \vdots \\ \vec{\bar{\alpha}} \end{pmatrix} + \begin{pmatrix} \vec{\alpha_1} - \vec{\bar{\alpha}} \\ \vec{\alpha_2} - \vec{\bar{\alpha}} \\ \vdots \\ \vec{\alpha_M} - \vec{\bar{\alpha}} \end{pmatrix},$$

where the quantities $\vec{\alpha_i}$ are constant vectors of length $N_1(i)$ with value α_i. A well known theorem from statistics [Ref. 24] states that $\sum_{i=1}^{N}(x_i - \bar{x}) = 0$. Therefore, by Theorem 4.3 the vector

$$\vec{v}_N = \begin{pmatrix} \vec{\alpha_1} - \vec{\bar{\alpha}} \\ \vec{\alpha_2} - \vec{\bar{\alpha}} \\ \vdots \\ \vec{\alpha_M} - \vec{\bar{\alpha}} \end{pmatrix}$$

80

is constant by angle and in NS(B). But we have

$$\bar{\alpha} = \frac{\sum_{i=1}^{M} \frac{C}{M}}{M} = \frac{C}{M}.$$

Therefore $\vec{v}_N = 0$ and $\vec{v} = \vec{v}_R$. Thus $\vec{v} \in Range(B^T)$. ∎

We have shown that the Gauss-Seidel method applied to the natural pixel problem is equivalent to the method of Kaczmarz applied to the square pixel problem, and given an appropriate initial guess Gauss-Seidel converges to the same minimum L_2 norm solution as Kaczmarz. Furthermore, we have shown conditions such that the problem is consistent and determined specifically what an appropriate initial guess should be. Analysis of the spectral properties of the Gauss-Seidel iteration matrix P_G is still required, to gain further insight as to why the method converges, and to analyze the rate of convergence.

C. SPECTRAL ANALYSIS

It has been shown thus far that the Gauss-Seidel method, when applied to the linear system derived from the natural pixel discretization, remains bounded in the energy semi-norm and that for certain starting vectors it converges to the minimum L_2 norm solution. The matrix B is only positive semi-definite, so Theorem 5.2 does not apply and the Gauss-Seidel iteration matrix P_G is not guaranteed to have a spectral radius $\rho(P_G) < 1$. Non-divergence has been established by Theorem 5.3, implying $\rho(P_G) \leq 1$. If an appropriate starting vector is given, Theorem 5.4 guarantees convergence, implying that the eigenvectors corresponding to eigenvalues

λ_i of P_G with $|\lambda_i| = 1$ do not affect the norm of the residuals. If such components affected the measure of the iterates, then the error expression

$$\vec{e}^{(k+1)} = (P_G)^{k+1}\vec{e}^{(0)}$$

would not decrease with k . We use this fact to arrive at the following result:

Theorem 5.6: *If $\vec{z}_i$ is an eigenvector of P_G, and if $\vec{z}_i \in NS(B)$, then the corresponding eigenvalue $\lambda_i = 1$.*

Proof: For the Gauss-Seidel method applied to the linear system

$$B\vec{\alpha} = \vec{f},$$

we have

$$B = D - U - L, \qquad P_G = (D - L)^{-1}U,$$

and

$$\vec{\alpha}^{(k+1)} = (D - L)^{-1}U\vec{\alpha}^{(k)} + (D - L)^{-1}\vec{f}.$$

Let an exact solution to the system be $\vec{\alpha}^*$. Then the iteration becomes

$$\vec{\alpha}^{(k+1)} = (D - L)^{-1}U\vec{\alpha}^{(k)} + (D - L)^{-1}B\vec{\alpha}^*.$$

Writing this expression in terms of the error vector yields

$$
\begin{aligned}
\vec{e}^{(k+1)} &= P_G\vec{e}^{(k)} \\
&= (D - L)^{-1}U\vec{e}^{(k)} \\
&= (D - L)^{-1}(D - L)\vec{e}^{(k)} - (D - L)^{-1}B\vec{e}^{(k)} \\
&= \vec{e}^{(k)} - (D - L)^{-1}B\vec{e}^{(k)}.
\end{aligned}
$$

Now, let $\vec{e}^{(k)} = \vec{z}_i \in NS(B)$. Then

$$\vec{e}^{(k+1)} = \vec{z}_i - (D - L)^{-1}B\vec{z}_i = \vec{z}_i.$$

But we know that $\vec{e}^{(k+1)} = P_G\vec{e}^{(k)}$ as well, so

$$\vec{e}^{(k+1)} = P_G\vec{e}^{(k)} = P_G\vec{z}_i = \lambda_i\vec{z}_i = \vec{z}_i = \vec{e}^{(k)}.$$

Therefore, $\lambda_i = 1$. ∎

We have learned that even though the spectral radius of the Gauss-Seidel iteration matrix for this problem may equal unity, the method still converges because the eigenvectors of P_G corresponding to eigenvalues of modulus one are in NS(B). Therefore, they do not contribute to the approximation. Theoretically, Gauss-Seidel should be able to solve the natural pixel problem. What remains to be done is actually apply Gauss-Seidel to several linear systems and analyze the resulting behavior.

D. BEHAVIOR OF GAUSS-SEIDEL APPLIED TO THE PROBLEM

Gauss-Seidel is applied to several linear systems generated at varying geometries, and with right-hand-side data created both analytically, and by projecting computer-generated images. In all cases, the overall behavior observed is of rapid initial convergence that eventually stalls out, just as occurs with the Kaczmarz iteration. Figure 33 depicts plots of both the norm of the residual and the convergence factor plotted against iterations for a typical problem. Here, the convergence factor is formed as the ratio of successive residual norms. The magnitude of the residual error in all cases is on the order of 10^{-3} or 10^{-4}, well short of machine precision. To explain why this behavior occurs, the spectrum of the matrix B is analyzed. Even though B is square, to parallel the analysis presented in Chapter III we use the singular value decomposition.

Recall the singular value decomposition (SVD) of B

$$B = U\Sigma V^T,$$

83
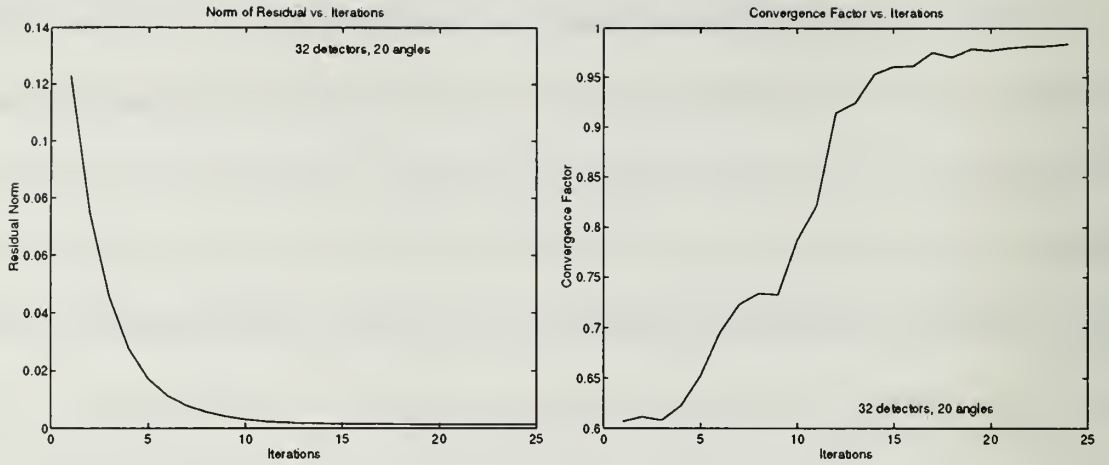
Figure 33. *Convergence of Gauss-Seidel on a typical problem.*

where U and V are orthogonal matrices, and Σ is a diagonal matrix whose diagonal entries σ_i are the *singular values* of B. The columns of U and V are known as *left* and *right singular vectors*, respectively. Singular values are real, nonnegative, and ordered such that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \cdots 0.$$

The number of non-zero singular values $r \leq N$ equals the rank of B. The SVD can be rewritten as

$$BV = U\Sigma,$$

and if the columns of this expression are compared, we arrive at a collection of linear systems

$$B\vec{v}_i = \sigma_i \vec{u}_i, \; 1 \leq i \leq N,$$

where $\vec{u}_i$ and $\vec{v}_i$ denote the ith columns of U and V, respectively. It is possible to determine which singular values have singular vectors $\vec{v}_i$ that are slow to converge.

84
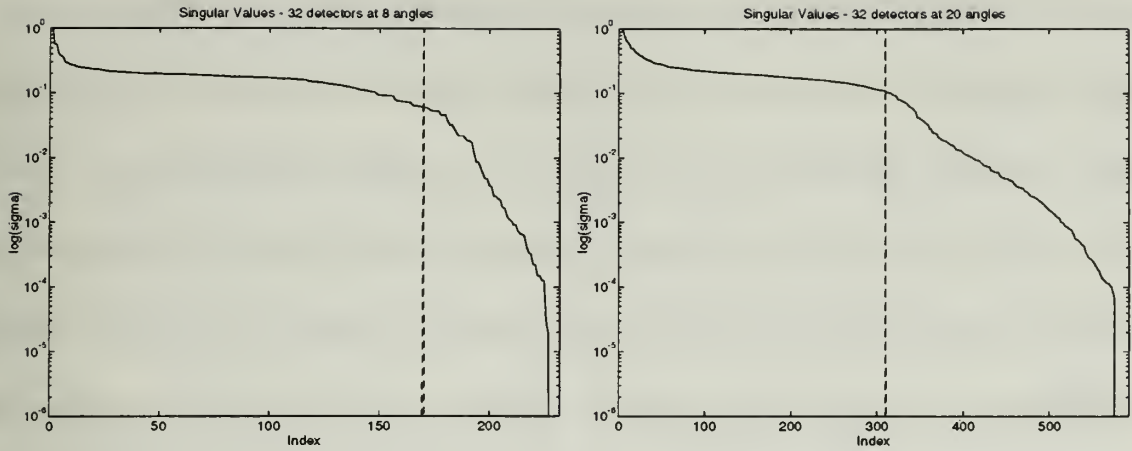
Figure 34. *Singular Values for 32 Detectors at 8 and 20 Angles.*

This is accomplished by using Gauss-Seidel to solve, for each i, the problem $B\vec{\alpha} = \sigma_i \vec{u}_i$, using zero as the initial guess. The columns of V form an orthonormal basis for $\mathcal{R}^N$, so any image can be expressed as a linear combination of the $\vec{v}_i$. If the image has components which are slow to converge, then the iteration will stall. Figure 34 depicts the spectrum of singular values for two geometries. One is constructed using 32 detectors at each of 8 evenly spaced angles, producing a matrix of size 232×232 and of rank 225. The second geometry is 32 detectors at each of 20 evenly spaced angles, resulting in a matrix of size 592×592 and of rank 573. It should be pointed out that for a geometry of 20 angles and 32 detectors per angle, one would think there would be 640 total x-rays producing a matrix B of size 640×640. However, only 592 x-rays actually pass through the image square, which explains the disparity in the size of the matrix. This fact holds for all geometries.

Notice that in both graphs, the magnitude of the singular values can be divided

into three *bands*.The first of these is the left portion of the plots where the curve is nearly horizontal, the second is in the center where the magnitude of the singular values noticeably decrease, and the third is to the right where the singular values drop suddenly to zero - the null space. As before, we will define the center band as the *near null space*,and the left band as the *resolvable region*.It will be observed that singular vectors corresponding to singular values in the near null space are the slow components of the image to converge.

Consider using Gauss-Seidel to solve the SVD system

$$B\tilde{\vec{v}}_i = \sigma_i \vec{u}_i,$$

whose exact solution is $\vec{v}_i$, for various values of i . The resulting solutions $\tilde{\vec{v}}_i$ should be good approximations for the corresponding $\vec{v}_i$. We can decompose the $\tilde{\vec{v}}_i$ into linear combinations of the singular vector basis as

$$\tilde{\vec{v}}_i = \sum_{j=1}^{N} \beta_j \vec{v}_j.$$

The singular vector basis is orthonormal, and the β_j's can be computed as

$$\beta_j = \tilde{\vec{v}}_i^T \vec{v}_j, \quad 1 \le j \le N.$$

Hence we can determine the components of $\tilde{\vec{v}}_i$ in the directions of each of the singular vectors $\vec{v}_j$. For the exact solution $\vec{v}_i$, we have

$$\beta_j = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases}$$

86
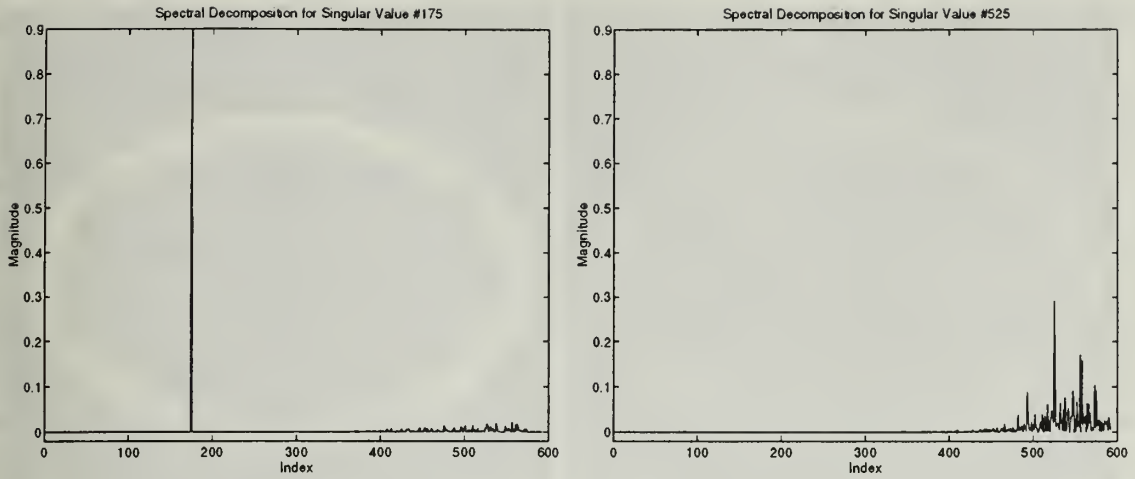
Figure 35. *Approximations for Singular Values in the Resolvable Region and Near Null Space.*

A plot of the β_j for the exact solution will be a spike of magnitude one at index i. The following figures are plots of the $|\beta_j|$ for some indices both in the resolvable region and in the near null space for a geometry of 32 detectors at 20 angles.

A qualitative interpretation of these plots follows. Three things are readily apparent. First, components in the resolvable region are almost totally recovered (Figure 35, left). A spike of magnitude one located at the appropriate index is clearly present, along with a small amount of noise. These components do not adversely affect the performance of the iteration. Next, components in the near null space are not recovered well (Figure 35, right). There is a partially recovered spike, and significant noise is present. These components represent the *unrecoverable,*or *slow,*components of the image that cause the iteration to stall. Finally, iterating with Gauss-Seidel, just as with Kaczmarz, *mixes modes,*that is, it introduces additional components of the singular vector spectrum as noise into the approximation $\tilde{\tilde{v}}_i$. The noise includes

components in the null space, but is predominantly composed of near null space components, that are not part of the actual solution $\vec{v}_i$.

This last observation is significant in that we can have an image that lives entirely in the resolvable region, apply Gauss-Seidel to reconstruct it, and excite components in the approximation that live in near null space and null space. The components in the null space are not a problem, as the images they generate have been shown to be invisible, and they do not affect the measurement of the residual. However, those in the near null space are a problem, for as seen above, they are slow to be recovered and virtually all of the residual error can be attributed to them.

In spite of these apparent shortcomings, Gauss-Seidel applied to the natural pixel discretized problem reconstructs images quite well. The following figures depict actual and reconstructed images for a brain phantom, ceated by superimposing a collection of ellipses and rectangles of varying grey levels on each other. The data vector $\vec{f}$ was then generated by *projecting* the image with Kaczmarz matrices of assorted geometries.

The behavior described above is not unique to the Gauss-Seidel method. Under the assumption that other types of iterative methods might not exhibit the behavior of Gauss-Seidel, several Lanczos-based methods were applied to the problem as well. In particular, the algorithms SYMMLQ and MINRES [Ref. 25] and several incomplete orthogonalization methods [Ref. 26] were applied to the image reconstruction problem. In all cases, convergence was initially rapid, followed by stalling. The recon-
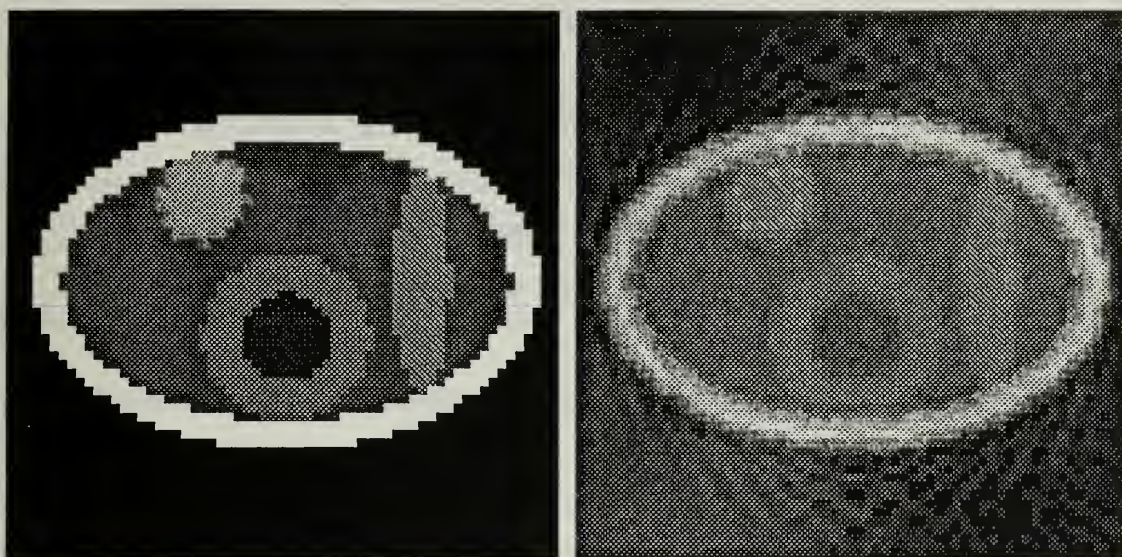
Figure 36. *Actual and Reconstructed Images - Brain Phantom*

structed images produced by these methods were no better than those produced by Gauss-Seidel. As all of these methods require more work per sweep than Gauss-Seidel and produce no better results, we will not examine them further.

We have conducted an analysis of the Gauss-Seidel method as applied to our image reconstruction problem. Although convergence has been established, there are certain components of the solution that are slow to be recovered. In the next chapter, a multilevel method will be developed that accelerates the convergence of the Gauss-Seidel iteration.

VI.   A MULTILEVEL APPROACH

A.   BASIC MULTILEVEL CONCEPTS

Multilevel methods were developed to overcome the numerical stalling of iterative methods for numerical partial differential equations. A fundamental principle of any multilevel methodology is that the amount of computational work should be proportional to the amount of real physical changes in the computed system [Ref. 27]. If it is not, e.g. if successive sweeps of an iterative method on a linear system produce smaller and smaller reductions in the error, then a more efficient method should exist to approach the problem. The image reconstruction problem exhibits such behavior. The situation occurs when there exists several solution components with different scales that conflict with each other. The answer could be a multilevel approach, which involves interactively employing several scales of discretization to resolve such conflicts, avoid stalling of the iteration, and eliminate computational waste. Before such a method can be designed, it is necessary to understand the basic methodology behind the approach and why it works. To that end, a *coarse grid correction scheme* as applied to a simple one-dimensional partial differential equation will be developed and analyzed.

1. Elements of a Multilevel Method

Consider the second-order differential equation for the steady-state heat distribution in a rod of uniform density that has the temperature fixed at both ends

$$-u''(x) - \sigma u(x) = f(x), \quad 0 < x < 1, \quad \sigma \geq 0$$

subject to the Dirichlet boundary conditions

$$u(0) = u(1) = 0.$$

The problem is discretized by breaking its domain into N equal subintervals of width h, which defines the node points $x_j = jh, \quad j = 0, 1, \cdots, N,$ and forms a grid which will be denoted as Ω^h. If we let v_j approximate $u(x_j)$, and if we approximate $u''(x_j)$ with a finite 2nd order difference, then the problem can be rewritten as

$$\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} + \sigma v_j = f(x_j), \quad v_0 = v_N = 0, \quad j = 1 : N - 1.$$

Here, we have converted an ordinary differential equation into a system of N-1 algebraic equations in N-1 unknowns, with the error in the approximation being of $O(h^2)$.

If we define

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{pmatrix}, \quad \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \end{pmatrix}, \quad \text{and } \vec{u} = \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_{N-1}) \end{pmatrix},$$

then we can express the problem as the linear system

$$A\vec{v} = \vec{f},$$

where

$$A = \frac{1}{h^2}\begin{pmatrix} 2 + h^2\sigma & -1 & & & \\ -1 & 2 + h^2\sigma & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 + h^2\sigma \end{pmatrix},$$

and whose solution $\vec{v}$ approximates $\vec{u} = u(\vec{x})$ at the grid points. It can be determined analytically that the eigenvectors of A are

$$\vec{z}_k = \begin{pmatrix} \sin(\frac{k\pi}{N}) \\ \sin(\frac{2k\pi}{N}) \\ \vdots \\ \sin(\frac{(N-1)k\pi}{N}) \end{pmatrix}, \quad k = 1 : N - 1.$$

Graphing $\vec{z}_k$, one finds that the graph is smooth for small values of k, and becomes increasingly oscillatory as k increases, as shown if Figure 37. Following the analysis of the previous chapter, we consider applying an iterative method to this linear system. We will use the weighted Jacobi method as a vehicle for this discussion. Letting

$$A = D - L - U,$$

92

then the weighted Jacobi iteration matrix is given by

$$P_J = (1 - \omega)I + \omega D^{-1}(L + U),$$

where $\omega \in \mathcal{R}$ is a weighting factor to be chosen. It can be shown that the eigenvectors of the matrix P_J are identical to the eigenvectors of the matrix A. The eigenvalues of P_J are

$$\lambda_k(P_J) = 1 - 2\omega \sin^2(\frac{k\pi}{2N}), \quad k = 1 : N - 1.$$
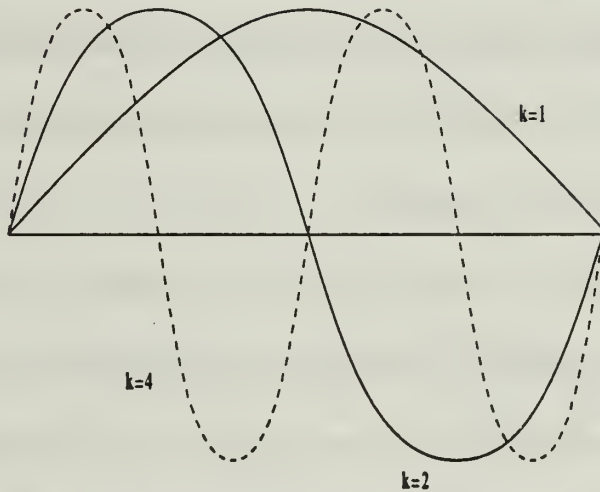
Figure 37. A mode at k=4 on grids of N=12 and N=6.

We define the *error vector* as

$$\vec{e} = \vec{v} - \vec{u}, \tag{6.1}$$

so that the initial error in the iteration is $\vec{e}^{(0)}$. This can be expanded in terms of the eigenvectors of P_J as

$$\vec{e}^{(0)} = \sum_{k=1}^{N-1} c_k \vec{z_k}, \quad c_k \in \mathcal{R}.$$

93

After n sweeps of the iteration, we have

$$\vec{e}^{(n)} = \sum_{k=1}^{N-1} c_k (P_J)^n \vec{z}_k = \sum_{k=1}^{N-1} c_k \lambda_k^n (P_J) \vec{z}_k.$$

Thus, after n sweeps the error has been reduced by a factor of $\lambda_k^n(P_J)$. This means that for large values of k, the error is reduced rapidly, while for small values of k it is not. We will call the eigenvectors with index $1 \leq k < \frac{N}{2}$ the low-frequency or *smooth* modes, and those with index $\frac{N}{2} \leq k \leq N-1$ the high-frequency or *oscillatory* modes. Our analysis shows that error components corresponding to smooth modes are slow to be eliminated. This behavior, while not as easily quantified, holds for the Gauss-Seidel method as well [Ref. 28]. The aim of a multilevel approach is to devise a way to address these slow components.

To develop the coarse grid correction scheme, we first must define a second set of $\frac{N}{2}$ grid points Ω^{2h}, obtained by selecting every other grid point from Ω^h. Note that Ω^{2h} is coarser that Ω^h, meaning that the grid spacing is wider. Now, assume the iterative method has been applied to the linear system on the original grid Ω^h, until only smooth error components remain. If we consider what the smooth error components look like on Ω^{2h}, as illustrated in the following Figure 38, we see that on the coarser grid the error becomes more oscillatory. On Ω^h mode 4 is one third of the way up the spectrum. However, on Ω^{2h} mode 4 is now two thirds of the way up the spectrum and, therefore, more oscillatory.

To be more precise, consider the kth mode on Ω^h evaluated at the even-numbered grid points, which is exactly Ω^{2h}. We can write

94

$$\vec{z}_k^h = \begin{pmatrix} \sin(\frac{2k\pi}{N}) \\ \sin(\frac{4k\pi}{N}) \\ \vdots \\ \sin(\frac{(N-2)k\pi}{N}) \end{pmatrix} = \begin{pmatrix} \sin(\frac{k\pi}{N/2}) \\ \sin(\frac{2k\pi}{N/2}) \\ \vdots \\ \sin(\frac{(N/2-1)k\pi}{N/2}) \end{pmatrix} = \vec{z}_k^{2h}, \quad 1 \leq k < \frac{N}{2}, \qquad (6.2)$$

where the superscripts denote the two grids. This implies that the kth mode on Ω^h

becomes the kth mode on Ω^{2h}, as long as $1 \leq k < \frac{N}{2}$. Thus in moving from the finer

grid to the coarser grid, a mode becomes more oscillatory, and as such relaxation

should be more effective. It is equally important to note that the smoothness of $\vec{e}$

after relaxation on Ω^h is what allows us to go to Ω^{2h} - only if $\vec{e}$ is smooth can it be

accurately represented on a coarser grid.
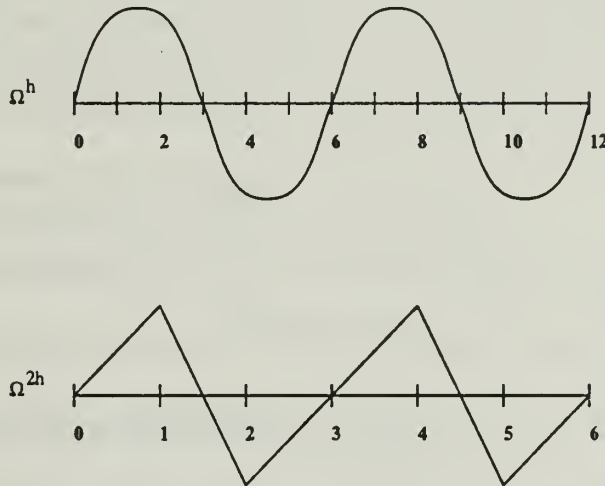
Figure 38. *A mode at k=4 on grids of N=12 and N=6.*

It can also be shown that if $k = \frac{N}{2}$, the mode becomes the zero vector

on Ω^{2h}, and if $\frac{N}{2} \leq k \leq N-1$, the kth mode on Ω^h becomes the $(N-k)^{th}$ mode

on Ω^{2h}. This last statement says that an oscillatory mode on the fine grid is *aliased*

smooth on a coarser grid, which implies that we do not want to move to the coarse grid if any oscillatory components are present in the error.

Let the *residual* be defined as

$$\vec{r} = \vec{f} - A\vec{v}, \tag{6.3}$$

and derive the *residual equation*

$$A\vec{e} = \vec{r}$$

by subtracting (6.3) from (6.1). It should be noted that relaxing on the original equation $A\vec{u} = \vec{f}$ with an arbitrary initial guess $\vec{v}$ is equivalent to relaxing on the residual equation $A\vec{e} = \vec{r}$ with an initial guess of the zero vector.

We can combine the above ideas into a multilevel method by relaxing on the fine grid until only smooth error components remain, then solving the residual equation for $\vec{e}$ on the coarse grid, and finally correcting the fine grid approximation $\vec{v}$ by that amount. There are two advantages of approaching a problem in this fashion. First, we can address the troublesome smooth error components on the coarse grid, and more importantly, by casting the problem on a coarser grid we reduce its size - in this example by 75%. In general, the reduction factor is $\frac{1}{2^D}$ in moving from Ω^h to Ω^{2h}, where D is the number of dimensions in the problem. This is the general basis of coarse grid correction, which can be expressed as the following procedure:

- Relax on $A\vec{u} = \vec{f}$ for $\vec{v}$ on Ω^h.
- Compute $\vec{r} = \vec{f} - A\vec{v}$
- Solve $A\vec{e} = \vec{r}$ on Ω^{2h}.

- Correct $\vec{v} \leftarrow \vec{v} + \vec{e}$ on Ω^h.

Some mechanisms must be developed for transfering vectors between the coarse and fine grids, as well as for representing the residual equation on the coarse grid. This will require some additional notation. Superscripts indicate the grid on which an expression is defined, e.g. $\vec{f}^{2h}$ is the right-hand-side represented on the coarse grid, while $\vec{v}^h$ is the approximate solution represented on the fine grid. In addition, the *intergrid transfer operators* $I_h^{2h} : \Omega^h \to \Omega^{2h}$ and $I_{2h}^h : \Omega^{2h} \to \Omega^h$ are defined, which serve to transfer quantities between the grids. The operation of transfering information from the coarse to the fine grid is called *interpolation* , and from the fine to the coarse grid is known as *restriction* . For the coarse grid correction procedure, we need to restrict the fine grid residual to the coarse grid, interpolate the coarse grid error correction to the fine grid, and represent the residual equation on both grids.

The interpolation operator is denoted by

$$I_{2h}^h \vec{v}^{2h} = \vec{v}^h,$$

and produces a fine grid vector whose entries are

$$v_{2j}^h = v_j^{2h} \quad \text{and} \quad v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}), \quad j = 0 : \frac{N}{2} - 1.$$

The even numbered grid points on Ω^h are exactly the Ω^{2h} grid points, while the odd numbered grid points on Ω^h are computed by averaging adjacent grid points from Ω^{2h}, that is, by linear interpolation. Figure 39 shows the action of I_{2h}^h.

97
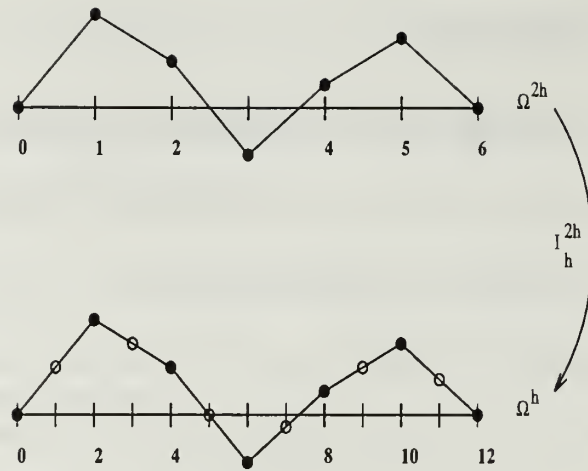
Figure 39. *Interpolation of a vector from* Ω^{2h} *to* Ω^h.

Interpolation acts on coarse grid vectors to produce fine grid vectors. It is critical to consider the smoothness of the vector being interpolated, as only smooth vectors on the coarse grid can be accurately represented on the fine grid. Figure 40 shows the interpolation of a smooth and oscillatory error vector to the fine grid. If the actual error is oscillatory on the fine grid, then interpolation will not accurately represent it.

The restriction operator takes vectors from the fine grid to the coarse grid as

$$I_h^{2h} \vec{v}^h = \vec{v}^{2h}.$$

There are several choices for restriction operators, of which we will use *full weighting,* which is defined as

$$v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h), \quad j = 1 : \frac{N}{2} - 1.$$

98
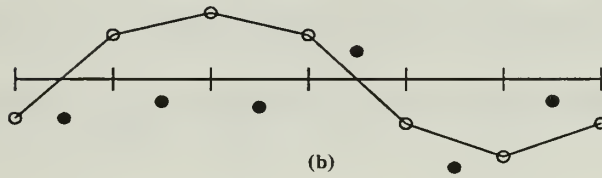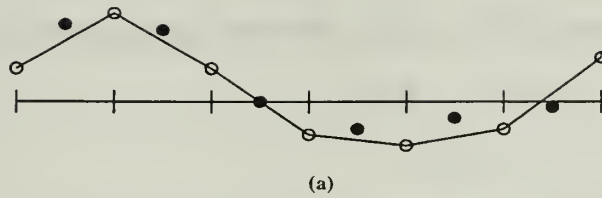
Figure 40. *(a) If the error (indicated by o and •) is smooth, an interpolant of the coarse grid error (indicated by o) should give a good representation. (b) If the error is oscillatory, an interpolant may give a poor representation.*

Values for coarse grid vectors are weighted averages of values at neighboring fine grid points. Figure 41 shows the action of I_h^{2h}, a very good example of why an oscillatory vector should not be restricted.
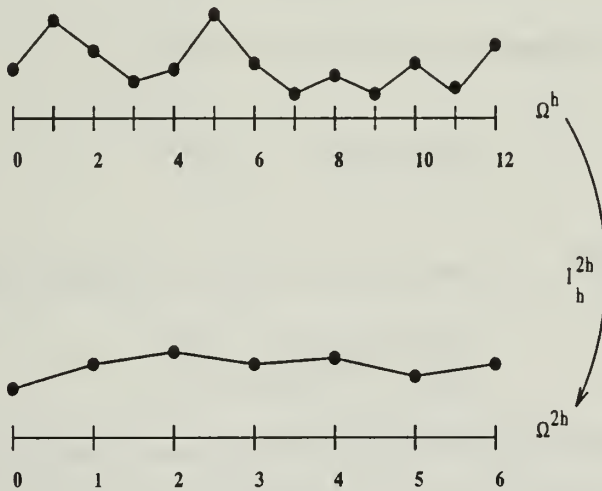
Figure 41. *Restriction of a fine grid vector to the coarse grid.*

Incorporating this new notation, we formally define the coarse grid correction scheme as

99

- Relax ν times on $A^h \vec{u}^h = \vec{f}^h$ on Ω^h with initial guess $\vec{v}^h$.

- Compute $\vec{r}^{2h} = I_h^{2h}(\vec{f}^h - A^h \vec{v}^h)$.

- Solve $A^{2h}\vec{e}^{2h} = \vec{r}^{2h}$ on Ω^{2h}.

- Correct the fine grid approximation $\vec{v}^h \leftarrow \vec{v}^h + I_{2h}^h \vec{e}^{2h}$.

Here, superscripts denote the grid where the quantity is defined, and the parameter ν represents the number of relaxation sweeps performed before moving to the coarse grid.

The quantity A^{2h}, the coarse grid version of A^h, remains to be defined. To this end, assume that the fine grid error $\vec{e}^h$ lies entirely in the $Range(I_{2h}^h)$. This means that for some $\vec{u}^{2h} \in \Omega^{2h}$, $\vec{e}^h = I_{2h}^h \vec{u}^{2h}$. Then we can write the residual equation as

$$A^h \vec{e}^h = A^h I_{2h}^h \vec{u}^{2h} = \vec{r}^h.$$

Applying the restriction operator to both sides yields

$$I_h^{2h} A^h I_{2h}^h \vec{u}^{2h} = I_h^{2h} \vec{r}^h,$$

or, denoting $I_h^{2h} \vec{r}^h$ by $\vec{r}^{2h}$,

$$(I_h^{2h} A^h I_{2h}^h)\vec{u}^{2h} = \vec{r}^{2h},$$

which has the same form as the residual equation, except that it is on the coarse grid. Thus it is reasonable to define

$$A^{2h} = I_h^{2h} A^h I_{2h}^h, \tag{6.4}$$

which is known as the *Galerkin condition* .

100

The intergrid transfer operators can be defined as matrices. Specifically

$$
I_{2h}^{h} = \frac{1}{2}
\begin{pmatrix}
1 & & & & \\
2 & & & & \\
1 & 1 & & & \\
& 2 & & & \\
& 1 & \ddots & 1 & \\
& & & 2 & \\
& & & 1 &
\end{pmatrix},
$$

and

$$
I_{h}^{2h} = \frac{1}{4}
\begin{pmatrix}
1 & 2 & 1 & & & & \\
& & 1 & 2 & 1 & & \\
& & & & \ddots & & \\
& & & & 1 & 2 & 1
\end{pmatrix}.
$$

The intergrid transfer operators satisfy the relationship

$$
I_{h}^{2h} = c(I_{2h}^{h})^{T}, \quad c \in \mathcal{R}. \tag{6.5}
$$

The relationships (6.4) and (6.5) together are known as the *variational properties* .
Satisfying the variational properties will further facilitate the analysis of the coarse
grid correction scheme.

Next, consider the action of the intergrid transfer operators on the modes of A^h, as given by (6.2). Full weighting applied to modes results in

$$I_h^{2h}\vec{z}_k^{\,h} = \cos^2\left(\frac{k\pi}{2N}\right)\vec{z}_k^{\,2h}, \quad 1 \le k \le \frac{N}{2},$$

and

$$I_h^{2h}\vec{z}_{N-k}^{\,h} = -\sin^2\left(\frac{k\pi}{2N}\right)\vec{z}_k^{\,2h}, \quad 1 \le k < \frac{N}{2}.$$

Here modes above $\frac{N}{2}$ are oscillatory, while those below $\frac{N}{2}$ are smooth. Thus, full weighting acting on the kth mode of A^h produces a multiple of the kth mode of A^{2h}. However, acting on the $(N-k)^{th}$ mode of A^h full weighting also produces a multiple of the kth mode of A^{2h} [Ref. 28]. So full weighting acting on an oscillatory mode will return a smooth mode. For this reason it is essential that we relax until only smooth modes remain before moving to the coarse grid.

The modes of A^{2h} are given by

$$\vec{z}_k^{\,2h} = \begin{pmatrix} \sin\left(\frac{k\pi}{N/4}\right) \\ \sin\left(\frac{2k\pi}{N/4}\right) \\ \vdots \\ \sin\left(\frac{(N/2-1)k\pi}{N/4}\right) \end{pmatrix}, \quad 1 \le k < \frac{N}{2}.$$

It can be shown /citebrig87 that the interpolation operator applied to these modes results in

$$I_{2h}^h\vec{z}_k^{\,2h} = \cos^2\left(\frac{k\pi}{2N}\right)\vec{z}_k^{\,h} - \sin^2\left(\frac{k\pi}{2N}\right)\vec{z}_{N-k}^{\,h}, \quad 1 \le k < \frac{N}{2}.$$

Thus, interpolation produces both smooth and oscillatory modes on the fine grid. Applied to the smoothest modes on Ω^{2h}, e.g., modes with $k \leq\leq \frac{N}{2}$, the magnitude of the multiplier of the smooth modes is of $O(1)$, while that of the oscillatory modes is of $O\left(\frac{k^2}{N^2}\right)$. Therfore, the result of interpolation is primarily a smooth mode on Ω^h.

With the knowledge of the action of the intergrid transfer operators, we return to the coase grid correction scheme, whose steps are

- Relax ν times on Ω^h using iterative method $P : \vec{v}^h \leftarrow P^\nu \vec{v}^h$,
- Full weight $\vec{r}^h$ to Ω^{2h}: $\vec{f}^{2h} \leftarrow I_h^{2h}(\vec{f}^h - A^h \vec{v}^h)$,
- Solve the residual equation exactly: $\vec{v}^{2h} = (A^{2h})^{-1} \vec{f}^{2h}$,
- Correct the approximation on Ω^h: $\vec{v}^h \leftarrow \vec{v}^h + I_{2h}^h \vec{v}^{2h}$.

The process may be written as a single operation, namely

$$\vec{v}^h \leftarrow P^\nu \vec{v}^h + I_{2h}^h (A^{2h})^{-1} I_h^{2h}(\vec{f}^h - A^h P^\nu \vec{v}^h).$$

The exact solution $\vec{u}^h$ is unaffected by coarse grid correction, so we have

$$\vec{u}^h \leftarrow P^\nu \vec{u}^h + I_{2h}^h (A^{2h})^{-1} I_h^{2h}(\vec{f}^h - A^h P^\nu \vec{u}^h).$$

Subtracting these two expressions yields

$$\vec{e}^h \leftarrow [I - I_{2h}^h (A^{2h})^{-1} I_h^{2h} A^h] P^\nu \vec{e}^h. \tag{6.6}$$

Denote the action of coarse grid correction as $CG : \Omega^h \to \Omega^h$, so that (6.6) can be rewritten as

$$\vec{e}^h \leftarrow CG \vec{e}^h.$$

We wish to determine the action of CG on the modes of A^h. Let $s_k = \sin^2(\frac{k\pi}{2N})$ and $c_k = \cos^2(\frac{k\pi}{2N})$. If we apply CG without relaxation, it can be shown that

$$CG\vec{z}_k^{\,h} = s_k \vec{z}_k^{\,h} + s_k \vec{z}_{N-k}^{\,h},$$

and

$$CG\vec{z}_{N-k}^{\,h} = c_k \vec{z}_k^{\,h} + c_k \vec{z}_{N-k}^{\,h}, \quad 1 \le k \le \frac{N}{2}.$$

Thus when CG is applied to either smooth or oscillatory modes, both smooth and oscillatory modes are produced. However, we must again look at the magnitudes of the resulting modes.

For $k << N$ we have

$$CG\vec{z}_k^{\,h} = O\left(\frac{k^2}{N^2}\right)\vec{z}_k^{\,h} + O\left(\frac{k^2}{N^2}\right)\vec{z}_{N-k}^{\,h},$$

and

$$CG\vec{z}_{N-k}^{\,h} = \left[1 - O\left(\frac{k^2}{N^2}\right)\right]\vec{z}_k^{\,h} + \left[1 - O\left(\frac{k^2}{N^2}\right)\right]\vec{z}_{N-k}^{\,h}.$$

Thus CG without relaxation acting on smooth modes produces both smooth and oscillatory modes of small magnitude, while, acting on oscillatory modes, produces both smooth and oscillatory modes of $O(1)$. To prevent using CG on oscillatory modes, relaxation is performed first to eliminate them. Then, after the fine grid approximation is corrected, more relaxation can be performed to eliminate those oscillatory modes excited by interpolation.

Now, the final version of the coarse grid correction scheme is

104

- Relax ν_1 times on $A^h \vec{u}^h = \vec{f}^h$ on Ω^h with initial guess $\vec{v}^h$.

- Compute $\vec{r}^{2h} = I_h^{2h}(\vec{f}^h - A^h \vec{v}^h)$.

- Solve $A^{2h} \vec{e}^{2h} = \vec{r}^{2h}$ on Ω^{2h}.

- Correct the fine grid approximation $\vec{v}^h \leftarrow \vec{v}^h + I_{2h}^h \vec{e}^{2h}$.

- Relax ν_2 times on $A^h \vec{u}^h = \vec{f}^h$ on Ω^h with initial guess $\vec{v}^h$.

The only unanswered question is how to *solve* the residual equation on Ω^{2h}. The answer is to think recursively. We keep transfering the problem to coarser and coarser grids until it is small enough to be solved easily with a direct method. The concept is illustrated in the following scheme.

Relax ν_1 times on $A^h \vec{u}^h = \vec{f}^h$ with initial guess $\vec{v}^h$.

Compute $\vec{f}^{2h} = I_h^{2h} \vec{r}^h$.

    Relax ν_1 times on $A^{2h} \vec{u}^{2h} = \vec{f}^{2h}$ with initial guess $\vec{v}^{2h} = 0$.

    Compute $\vec{f}^{4h} = I_{2h}^{4h} \vec{r}^{2h}$.

        ⋮

        Solve $A^{Lh} \vec{v}^{Lh} = \vec{f}^{Lh}$.

        ⋮

    Correct $\vec{v}^{2h} \leftarrow \vec{v}^{2h} + I_{4h}^{2h} \vec{v}^{4h}$.

    Relax ν_2 times on $A^{2h} \vec{u}^{2h} = \vec{f}^{2h}$ with initial guess $\vec{v}^{2h}$.

Correct $\vec{v}^h \leftarrow \vec{v}^h + I_{2h}^h \vec{v}^{2h}$.

Relax ν_2 times on $A^h \vec{u}^h = \vec{f}^h$ with initial guess $\vec{v}^h$.

The scheme telescopes down to the coarsest grid, which may be a single point, and works its way back up to the finest grid. Figure 42 shows the schedule of grids visited during the execution of the algorithm for six levels, which resembles the letter V. For this reason, the multilevel scheme is often referred to as a *V-cycle*.
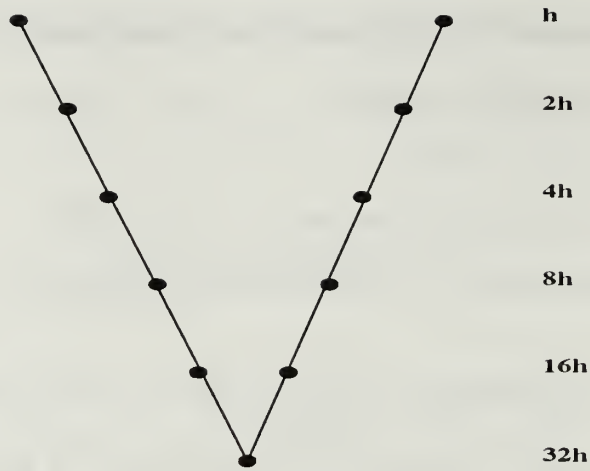
Figure 42. *Schedule of grids for a V-cycle.*

To summarize, all frequencies (excepting the smoothest) are eventually oscillatory on some grid where they are eliminated by relaxation. The smoothest frequencies are eliminated by the direct solve on the coarsest grid. When used these schemes are used together in the form of a V-cycle, *all* error components are eliminated. The intergrid transfer operations that make up CG are chosen to complement each other and the relaxation method being used. We want the restriction operator such that the smooth components of the error on Ω^h appear oscillatory transfered to Ω^{2h}, so that relaxation will be effective in eliminating them. Similarly, we want the interpolation operator to faithfully represent smooth components of the error on Ω^{2h} smooth when they are transfered to Ω^h. All of these operations must complement each other, or the multilevel method may not be as effective as it could be.

The coarse grid correction scheme derived for this model differential equation illustrates the workings of all the elements of a multilevel method. Unfor-

106

tunately, the image reconstruction problem is nothing like the model problem, and if we took the naive approach of applying the elementary multilevel approach outlined above numerous difficulties would arise. For example, what are the fine and coarse grids in this setting? What do restriction and interpolation represent in terms of the physics of the problem? Does restriction make a smooth error component appear more oscillatory in this setting? Can the error be accurately represented on the coarse grid? These and other questions must be addressed. Clearly, a different approach is needed.

To cast the image reconstruction problem in a multilevel setting, we will use the abstract *multilevel projection method,*or *PML,*approach. In PML, the problem is *discretized by projections,*and these projection operators in turn define the intergrid transfer operators and the appropriate relaxation method.

2.   Multilevel Projection Methods (PML)

The multilevel projection methodology, due to McCormick [Ref. 2], was developed so that a variety of problem types, not limited to elliptic partial differential equations, could be cast in a multilevel setting. PML is useful in that it provides a formalism that greatly eases the development of intergrid transfer operators and relaxation schemes that complement each other. The designer of the multilevel scheme must specify a set of subspaces, and the other components of the scheme are determined. This is significant, for most of the work involved in designing a multilevel scheme is taken up in choosing such components, which is a difficult process and

sometimes involves trial and error. If these operators are not matched, the multilevel scheme might not be effective.

In PML the problem is discretized by orthogonal projections, and the projection operators themselves lead to the correct choices for intergrid transfer operators and relaxation schemes. We now briefly describe the general principles of PML.

Let H_1 and H_2 be Hilbert spaces, let $L : H_1 \rightarrow H_2$, and let $f \in H_2$ be given, and consider a problem defined by

$$Lu = f, \quad u \in H_1.$$

Define $K(u) = Lu - f = 0$, where $K : H_1 \rightarrow H_2$. Then we write the problem as find $u \in H_1$ such that

$$K(u) = 0.$$

The problem is generally posed in this fashion so that it may be treated in equation (strong), variational, or weak form, although we will only consider equation form here.

One of the basic principles of the multilevel projection methodology is that discretization is accomplished by projections, a procedure that relates the continuum problem to a discrete problem on level h .

Let S^h be a finite-dimensional subspace of H_1, and let $P^{S^h} : H_1 \rightarrow S^h$ be an orthogonal projection of H_1 onto S^h. Similarly, let $P^{T^h} : H_2 \rightarrow T^h$ be an orthogonal projection of H_2 onto a finite-dimensional subspace T^h.

Assume also that mappings P_{S^h} and P_{T^h} exist from the level h spaces to the continuum spaces

$$P_{S^h} : S^h \to H_1, \quad S^h \subset H_1,$$

and

$$P_{T^h} : T^h \to H_2, \quad T^h \subset H_2.$$

These are generally identity operators, but we carry them for generality.

Here, S^h and T^h are subspaces corresponding to some discretization parameter h. That is, they could be the space spanned by a set of finite element basis functions on a grid with nodal spacing h; they could be continuum functions sampled on a grid with spacing h, or (for the image reconstruction problem) they could be spaces spanned by strips of width h.

Finally, assume that a similar set of orthogonal projections exist for level $2h$ subspaces S^{2h} and T^{2h}. Then the interlevel transfers between spaces at levels h and $2h$ are defined implicitly by finding operators I_h^{2h} and I_{2h}^h that make the diagram in Figure 43 commute.

That is, $I_h^{2h} : S^h \to S^{2h}$ and $I_{2h}^h : S^{2h} \to S^h$ are defined implicitly by

$$P_{S^{2h}} \;=\; P_{S^h} I_{2h}^h.$$

and

$$P^{S^{2h}} \;=\; I_h^{2h} P^{S^h}.$$

An analogous diagram exists for the H_2 subspaces T^h and T^{2h}.
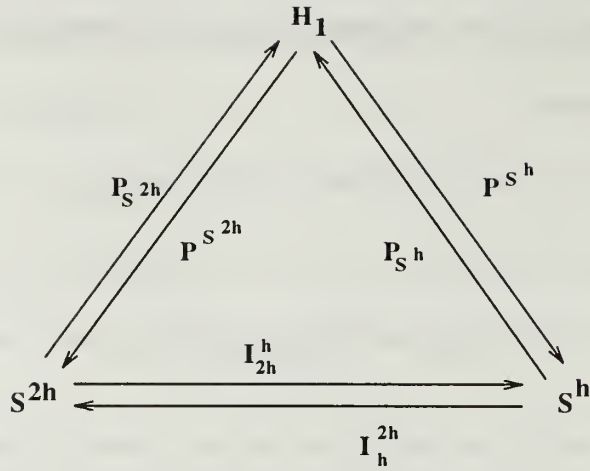
Figure 43. *Discretization/Coarsening Diagram for H_1.*

The process described above is known as *discretization by projections,* and it dictates what the coarse grid and intergrid transfer operators should be. Following this approach, the discretized problem becomes

$$P^{T^h} K (P^{S^h} u) = 0, \quad u \in H_1,$$

which we write as

$$K^h(u^h) = 0, \quad u^h \in S^h.$$

It is important to note that K^h is *defined* by the continuum operator K and the action of the projection operators, which is fundamental to PML. That is, $K^h \equiv P^{T^h} K P^{S^h}$.

For ease of development, it is usually assumed that the subspaces are *conforming* , with $S^{2h} \subset S^h$ and $T^{2h} \subset T^h$, and that the so-called *variational properties* are satisfied, that is

$$I_h^{2h} = c I_{2h}^h, \quad c \in \mathcal{R},$$

110

and

$$K^{2h} = I_h^{2h} K^h I_{2h}^h.$$

It is important that this discretized problem be *realizable,*or representable in a computational setting. The usual approach is to specify the subspaces S^h and T^h in terms of finite elements, choose a basis for each, and then rewrite the discrete problem in terms of the coefficients of the unknown u^h expanded in the basis for S^h.

The two main components of a multilevel method needed to solve this problem are relaxation and coarse grid corection. Relaxation will take the form of a generalized block Gauss-Seidel method. To develop the relaxation method, define the *block* subspaces S_ℓ^h, $1 \le \ell \le m$, such that

$$\sum_{\ell=1}^m S_\ell^h = S^h.$$

This is not necessarily a direct sum, although it may be. Therefore any element of S^h can be written as a (not necessarily unique) linear combination of the elements of S_ℓ^h, e.g.

$$u^h = \sum_{\ell=1}^m \alpha_\ell u_{(\ell)}^h, \quad \text{where} \quad u_{(\ell)}^h \in S_\ell^h.$$

We can define similar block subspaces for T^h. Relaxation applied to the discretized problem is then given by the steps

For $\ell = 1,2, ..., m$

Solve $P^{T_\ell^h} K^h (u^h + P^{S_\ell^h} u_{(\ell)}^h) = 0.$

111

Set $u^h \leftarrow u^h + u^h_{(\ell)}$.

Essentially, at each step we are seeking an element from the appropriate block sub-space such that after adding this element to the current approximation the projection of the residual onto the subspace vanishes. If the block subspaces S^h_ℓ are chosen to be the standard basis vectors, the relaxation method as defined here is just point Gauss-Seidel. Relaxation will be represented as

$$u^h \leftarrow G^h(u^h).$$

The coarse grid correction procedure is not difficult to define, as it involves an exact solution on level $2h$. The procedure is represented as

$$u^h \leftarrow C^h(u^h),$$

and is given by the steps

Solve $P^{T^{2h}} K^h(u^h + P^{S^{2h}} u^{2h}) = 0, \quad u^{2h} \in S^{2h}$.

Set $u^h = u^h + u^{2h}$.

Here we seek an element from the S^{2h} subspace that solves the residual equation on level $2h$. This element is then used to correct the level h approximation. We can combine coarse grid correction with relaxation to produce a two-level PML method that will be denoted as

$$u^h \leftarrow PML^h(u^h),$$

and is given by the steps

- $u^h \leftarrow G^h(u^h).$

112

- $u^h \leftarrow C^h(u^h)$.

As before, the exact solve on level $2h$ can be approximated by a recursive call $u^{2h} \leftarrow PML^{2h}(u^{2h})$, leading to a PML V-cycle.

B.   PML FORMULATION OF THE IMAGE RECONSTRUCTION PROBLEM

We can apply PML methodology to the image reconstruction problem to develop a multilevel method that is more effective than relaxation alone.

1.   Discretization by Projections

We begin by showing that the natural pixel discretization of Chapter IV is in fact a discretization by orthogonal projections. The continuum problem to be discretized is $Au = \vec{f}$, where $A : L_2(\Omega) \to \mathcal{R}^N$. We take $H_1 = L_2(\Omega)$ and $H_2 = \mathcal{R}^N$. The subspace $S^h = Range(A^*)$, the span of the characteristic strip functions. We take $T^h = \mathcal{R}^N$, which implies that $P^{T^h} = I_N$, the $N \times N$ identity matrix.

The orthogonal projection operator P^{S^h} *could* be explicitly calculated by applying the Gram-Schmidt process to the characteristic strip functions, so that

$$P^{S^h}u = \sum_{j=1}^{N} \langle u, \xi_j \rangle \xi_j,$$

where the ξ_j's are a set of orthogonal basis functions produced by Gram-Schmidt on the ψ_j's. However, the following result shows that it is not necessary to explicitly produce the orthogonal projection operator P^{S^h}.

Theorem 6.1: *For each angle ϕ_j, $1 \leq j \leq M$, let Ω be exactly partitioned into $N_1(j)$ parallel, non-overlapping strips, and let $N = \sum_{j=1}^{M} N_1(j)$. Number the strips from 1 to N and let $\psi_j^h(x,y)$ be the characteristic function of the jth strip. Let S^h be the subspace of the Hilbert space $H_1 = L_2(\Omega)$ spanned by the set*

$$\{\psi_j^h\}_{j=1}^{N}.$$

Then the matrix equation

$$B\vec{\alpha} = \vec{f}$$

is a discretization by orthogonal projection of the problem $Au = \vec{f}$.

Proof: Using the various subspaces as defined above, the discrete equation will be

$$I_N A P^{S^h} u = \vec{f},$$

where $P^{S^h} u$ is an orthogonal projection of $u(x,y)$ onto S^h, and therefore

$$P^{S^h} u = \sum_{j=1}^{N} \alpha_j \psi_j(x,y) = A^* \vec{\alpha}$$

for some $\vec{\alpha} \in \mathcal{R}^N$. Since P^{S^h} is an orthogonal projection, we must have $(u - P^{S^h} u) \perp \psi_j$ for every $\psi_j \in S^h$. Hence for $1 \leq j \leq N$,

$$
\begin{aligned}
0 &= \langle u - P^{S^h} u, \psi_j \rangle \\
&= \langle u - A^* \vec{\alpha}, \psi_j \rangle \\
&= \left\langle u - \sum_{k=1}^{N} \alpha_k \psi_k, \psi_j \right\rangle \\
&= \langle u, \psi_j \rangle - \sum_{k=1}^{N} \alpha_k \langle \psi_k, \psi_j \rangle.
\end{aligned}
$$

Therefore if $P^{S^h} u = A^* \vec{\alpha}$ is an orthogonal projection of $u(x,y)$ into S^h, then the vector $\vec{\alpha}$ must satisfy

$$
\begin{pmatrix}
< u, \psi_1 > \\
< u, \psi_2 > \\
< u, \psi_3 > \\
\vdots \\
< u, \psi_N >
\end{pmatrix}
= B\vec{\alpha}.
\tag{6.7}
$$

114

But the left side of (6.7) is just equal to Au, hence we conclude that $\vec{\alpha}$ must solve $Au = B\vec{\alpha}$, and the projection- discretized form of $Au = \vec{f}$ is just $B\vec{\alpha} = \vec{f}$. ∎

In this way our natural pixel discretization of the problem can formally be considered a discretization by projection in the PML sense, and we need not concern ourselves with explicitly forming the projection operator. With this discretization in hand, the remaining concepts of PML can be applied in a straightforward manner. We continue by defining the coarse subspaces S^{2h} and T^{2h} in a fashion that leads to a useful multilevel algorithm.

2.   Intergrid Transfer Operators

Let S^h be the span of the N characteristic strip functions ψ_j^h, where h is some parameter that indicates the level of the discretization. For example, h may indicate the width of the widest strip function at that level. Suppose that there is an even number of strip functions for each of the M angles, and that we number the functions from ψ_1^h to ψ_N^h in a way so that two adjacent strips on any view are always numbered consecutively. Then the subspace S^{2h} can be constructed as

$$S^{2h} = span\left\{\psi_k^{2h}\right\}_{k=1}^{N/2} \quad \text{where} \quad \psi_k^{2h} = \psi_{2k-1}^h + \psi_{2k}^h.$$

Each characteristic strip function in the coarse subspace is the union of two adjacent fine space strip functions. We define the coarse grid problem by thickening the x-rays. Using these coarse subspace strip functions and following the procedures of Chapter IV, we can define $(A^{2h})^* : \mathcal{R}^{N/2} \to S^{2h}$ by

$$(A^{2h})^*\vec{\alpha}^{2h} = \left(\psi_1^{2h} \; \psi_2^{2h} \; \psi_3^{2h} \; \cdots \; \psi_{N/2}^{2h}\right) \begin{pmatrix} \alpha_1^{2h} \\[6pt] \alpha_2^{2h} \\[6pt] \alpha_3^{2h} \\[6pt] \vdots \\[6pt] \alpha_{N/2}^{2h} \end{pmatrix},$$

which in turn defines $A^{2h} : S^{2h} \to \mathcal{R}^{N/2}$ by

$$A^{2h}u = \begin{pmatrix} <\psi_1^{2h}, u> \\[6pt] <\psi_2^{2h}, u> \\[6pt] <\psi_3^{2h}, u> \\[6pt] \vdots \\[6pt] <\psi_{N/2}^{2h}, u> \end{pmatrix}.$$

An application of Theorem 6.1 using these level $2h$ subspaces leads to the projection discretized coarse level problem $B^{2h}\vec{\alpha}^{2h} = \vec{f}^{2h}$, where $B^{2h} = A^{2h}(A^{2h})^*$ is an $(N/2) \times (N/2)$ matrix with entries $(b_{ij}^{2h}) = \langle \psi_i^{2h}, \psi_j^{2h} \rangle$.

Having found orthogonal projections into the S^h and S^{2h} subspaces, we next must determine intergrid transfer operators I_h^{2h} and I_{2h}^h as implicitly defined in Figure 43. The derivation is centered around the definitions of the coarse subspace strip functions ψ_k^{2h}.

Lemma 6.1: *Let the coarse subspace strip function ψ_k^{2h} be the union of two adjacent fine subspace strip functions given by $\psi_k^{2h} = \psi_{2k-1}^h + \psi_{2k}^h$, for $1 \leq$*

$k \leq N/2$. Then the operators $A^h : S^h \to T^h$ and $A^{2h} : S^{2h} \to T^{2h}$ are related by

$$A^{2h} = I_h^{2h} A^h$$

where I_h^{2h} is an $(N/2) \times (N)$ matrix given by

$$I_h^{2h} = \begin{pmatrix} 1 & 1 & & & & & \\ & & 1 & 1 & & & \\ & & & & 1 & 1 & \\ & & & & & \ddots & \ddots & \\ & & & & & & & 1 & 1 \end{pmatrix}.$$

Furthermore, the adjoint operators $(A^h)^* : T^h \to S^h$ and $(A^{2h})^* : T^{2h} \to S^{2h}$ are related by

$$(A^{2h})^* = (A^h)^* \left(I_h^{2h} \right)^T.$$

Proof: The kth component of the vector $A^{2h} u \in T^{2h}$ is related to the $(2k-1)^{st}$ and $2k^{th}$ components of $A^h u \in T^h$ by

$$\langle \psi_k^{2h}, u \rangle = \langle \psi_{2k-1}^h + \psi_{2k}^h, u \rangle = \langle \psi_{2k-1}^h, u \rangle + \langle \psi_{2k}^h, u \rangle = (1 \ \ 1) \begin{pmatrix} \langle \psi_{2k-1}^h, u \rangle \\ \langle \psi_{2k}^h, u \rangle \end{pmatrix}.$$

Now, partition the vector $A^h u$ into blocks consisting of pairs of adjacent entries and form the matrix I_h^{2h} by placing the block $(1 \ \ 1)$ in the $(2k-1)^{st}$ and $(2k)^{th}$ positions of the kth row of a $(N/2) \times (N)$ matrix of zeros, for $1 \leq k \leq N/2$. The matrix vector multiplication $I_h^{2h} A^h u$ produces $A^{2h} u$ and proves the first part of the lemma.

The second part of the lemma is arrived at by observing that

$$
\begin{aligned}
(A^{2h})^* \vec{\alpha}^{2h} &= \begin{pmatrix} \psi_1^{2h} & \psi_2^{2h} & \cdots & \psi_{N/2}^{2h} \end{pmatrix} \vec{\alpha}^{2h} \\
&= \begin{pmatrix} \psi_1^h + \psi_2^h & \psi_3^h + \psi_4^h & \cdots & \psi_{N/2-1}^h + \psi_{N/2}^h \end{pmatrix} \vec{\alpha}^{2h} \\
&= \begin{pmatrix} \psi_1^h & \psi_2^h & \psi_3^h & \psi_4^h & \cdots & \psi_{N/2-1}^h & \psi_{N/2}^h \end{pmatrix} \begin{pmatrix} 1 & & & \\ 1 & & & \\ & 1 & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & 1 \end{pmatrix} \vec{\alpha}^{2h} \\
&= (A^h)^* \left(I_h^{2h} \right)^T \vec{\alpha}^{2h}.
\end{aligned}
$$

The second part of the lemma verifies that the operator I_h^{2h} gives a consistent definition to the adjoint of the coarse space operator A^{2h}, showing that

$$(A^{2h})^* = (I_h^{2h} A^h)^* = (A^h)^* \left(I_h^{2h} \right)^T .$$

Thus we can define $I_{2h}^h = \left(I_h^{2h} \right)^T$. Finally, we show that the intergrid transfer operators as defined satisfy the variational properties.

Theorem 6.2: *The discrete operators satisfy the variational properties*

$$I_{2h}^h = c \left(I_h^{2h} \right)^T , \quad c \in \mathcal{R}$$

and

$$B^{2h} = I_h^{2h} B^h I_{2h}^h.$$

Proof: The first property is satisfied with $c = 1$, as was shown in the proof of Lemma 6.1. For the second property, consider

$$
\begin{aligned}
B^{2h} &= A^{2h}(A^{2h})^{\bullet} \\
&= I_h^{2h} A^h (A^h)^{\bullet} I_{2h}^h \\
&= I_h^{2h} B^h I_{2h}^h.
\end{aligned}
$$

3. Relaxation

A relaxation scheme can be developed, following the principles of PML, by partitioning the discrete spaces S^h and T^h into block subspaces

$$S^h = \sum_{\ell=1}^{m} S_\ell^h \quad \text{and} \quad T^h = \sum_{\ell=1}^{m} T_\ell^h.$$

118

One obvious choice of block subspaces is to let

$$S_\ell^h = span\left\{\psi_\ell^h\right\} \quad and \quad T_\ell^h = span\left\{\vec{w}_\ell^h\right\}, \quad \ell = 1:N.$$

Here, the $\vec{w}_\ell^h$ are the standard unit basis vectors for $\mathcal{R}^N$. Using these choices, we have the following

Lemma 6.2: *The PML relaxation scheme on $Au = \vec{f}$, using the above block subspaces, is implemented by performing point Gauss-Seidel iteration on the matrix equation $B\vec{\alpha} = \vec{f}$.*

Proof: The ℓth step of PML relaxation requires finding a value s satisfying

$$P_\ell^{T^h}\left(AP^{S^h}(u^h + s\psi_\ell^h) - \vec{f}^h\right) = 0$$

where u^h is the current approximation of the solution, which is then updated by

$$u^h \leftarrow u^h + s\psi_\ell^h.$$

Now $\psi_\ell^h = (A^h)^*\vec{w}_\ell^h$, and since $P^{S^h}u^h \in S^h$ we must have $P^{S^h}u^h = (A^h)^*\vec{\alpha}^h$ for some $\vec{\alpha}^h \in T^h$. Hence we need s to satisfy

$$0 = P_\ell^{T^h}\left(A^h((A^h)^*\vec{\alpha}^h + s(A^h)^*\vec{w}_\ell^h) - \vec{f}^h\right)$$
$$= P_\ell^{T^h}B^h(\vec{\alpha}^h + s\vec{w}_\ell^h) - f_\ell^h.$$

Realizing that the action of the projection P^{T^h} is nothing more than forming an inner product with $\vec{w}_\ell^h$, we seek an s such that

$$(\vec{w}_\ell^h)^T(B^h\vec{\alpha}^h + sB^h\vec{w}_\ell^h) = f_\ell^h.$$

The solution is given by

$$s = \frac{1}{b_{\ell\ell}}(f_\ell^h - \vec{b}_\ell^T\vec{\alpha}^h),$$

where $\vec{b}_\ell^T$ is the ℓth row of B^h. Therefore, the ℓth step of PML relaxation is

$$\vec{\alpha}^h \leftarrow \vec{\alpha}^h + \frac{1}{b_{\ell\ell}}(f_\ell^h - \vec{b}_\ell^T\vec{\alpha}^h),$$

which is precisely the correction of the ℓth step of Gauss-Seidel applied to $B^h\vec{\alpha}^h = \vec{f}^h$. ∎

4. Coarse Grid Correction

The final component required to complete our PML formulation is the coarse grid correction. For the problem $Au = \vec{f}$, it is defined as finding that element $u^{2h} \in S^{2h}$ which satisfies

$$P^{T^{2h}} \left(A(P^{S^h} u^h + P^{S^{2h}} u^{2h}) - \vec{f}^h \right) = 0, \tag{6.8}$$

where u^h is the current approximation of the solution in the fine space S^h. The correction is then given by

$$u^h \leftarrow u^h + u^{2h}.$$

We know that

$$P^{T^{2h}} A P^{S^h} u^h = I_h^{2h} P^{T^h} A P^{S^h} u^h = I_h^{2h} B^h \vec{\alpha}^h$$

where $(A^h)^* \vec{\alpha}^h$ represents $P^{S^h} u^h$. We also know that since $u^{2h} \in S^{2h}$, there exists a vector $\vec{\epsilon}^{2h} \in T^{2h}$ such that $u^{2h} = (A^{2h})^* \vec{\epsilon}^{2h}$. Hence $P^{T^{2h}} A P^{S^{2h}} u^{2h} = B^{2h} \vec{\epsilon}^{2h}$. Noting also that $P^{T^{2h}} \vec{f}^h = I_h^{2h} \vec{f}^h$, then (6.8) becomes

$$I_h^{2h} B^h \vec{\alpha}^h + B^{2h} \vec{\epsilon}^{2h} - I_h^{2h} \vec{f}^h = 0.$$

This establishes that under the PML methodology, the coarse grid correction scheme is equivalent to that used for conventional model problems. In other words, we have proved the following result.

Lemma 6.3: *Let $B^h \vec{\alpha}^h = \vec{f}^h$ be the discretization by projections of $Au = \vec{f}$ using the characteristic strip functions ψ_j^h and ψ_k^{2h}. Suppose that $(A^h)^* \vec{\alpha}^h$ is the representation of the current approximation u^h after relaxation. Then the PML coarse grid correction scheme is given by the steps*

120

1) Set $\vec{f}^{2h} = I_h^{2h}(\vec{f}^h - B^h \vec{\alpha}^h)$.

2) Solve the equation $B^{2h} \vec{e}^{\,2h} = \vec{f}^{2h}$.

3) Correct the approximation $\vec{\alpha}^h \leftarrow \vec{\alpha}^h + I_{2h}^h \vec{e}^{\,2h}$.

The complete PML two-level method is formed by combining relaxation and coarse grid correction.

Two-level PML method: $\vec{\alpha}^h \leftarrow PML(B^h, \vec{\alpha}^h, \vec{f}^h)$

1) Relax ν_1 times on $B^h \vec{\alpha}^h = \vec{f}^h$.

2) Set $\vec{f}^{2h} = I_h^{2h}(\vec{f}^h - B^h \vec{\alpha}^h)$.

3) Solve the equation $B^{2h} \vec{e}^{\,2h} = \vec{f}^{2h}$.

4) Correct the approximation $\vec{\alpha}^h \leftarrow \vec{\alpha}^h + I_{2h}^h \vec{e}^{\,2h}$.

5) Relax ν_2 times on $B^h \vec{\alpha}^h = \vec{f}^h$.

The additional ν_2 relaxation sweeps at step 5 are optional, but have been observed to improve performance in model problems, so are included here for generality. As discussed earlier, the exact solve at step 3 can be replaced by a recursive application of the entire process, so that the only time an exact solve is required is on the coarsest subspace. To realize such a recursion in the PML setting, define the coarser subspaces S^{jh}, for $j = 1, 2, \cdots$ by taking the characteristic strip functions that define the new subspace to be the pairwise joining of strip functions in the current subspace, just as was done to form S^{2h} from S^h. The coarsest level in this context is one thick x-ray that completely covers the image space at each angle. The resulting linear system would be diagonal and of size $M \times M$, and could be solved directly. The recursive version of the method, a PML V-cycle, is given by the following algorithm:

$$\text{PML V-cycle:} \quad \vec{\alpha}^h \leftarrow PMLV(B^h, \vec{\alpha}^h, \vec{f}^h)$$

1) Relax ν_1 times on $B^h \vec{\alpha}^h = \vec{f}^h$.

2) If at the coarsest level, go to 3. Otherwise

    a) $\vec{f}^{2h} = I_h^{2h}(\vec{f}^h - B^h \vec{\alpha}^h)$.

    b) $\vec{\alpha}^{2h} \leftarrow 0$.

    c) $\vec{\alpha}^{2h} \leftarrow PMLV(B^{2h}, \vec{\alpha}^{2h}, \vec{f}^{2h})$.

    d) $\vec{\alpha}^h \leftarrow \vec{\alpha}^h + I_{2h}^h \vec{\alpha}^{2h}$.

3) Relax ν_2 times on $B^h \vec{\alpha}^h = \vec{f}^h$.

5.   Convergence

We next look at the convergence properties of this multilevel method, and present a formal proof. It is of limited use as a convergence proof, as it depends on some constants that cannot be determined a priori. However, the result has some practical applications related to the performance of the multilevel method.

Consider comparing residual norms before and after a V-cycle is performed. This entails writing the algorithm in more detail, so that residuals can be examined at various steps within a cycle. It also involves placing *side conditions* on the relaxation scheme to measure its effectiveness. Ultimately, we desire the norm of the residual to be reduced by the scheme at each level in the V-cycle.

Define an *artificial* level $\Omega^{\frac{h}{2}}$. Let

$$\Omega^{\frac{h}{2}} = \Omega^h, \quad I_h^{\frac{h}{2}} = I_{\frac{h}{2}}^h = I, \quad B^{\frac{h}{2}} = B^h, \text{ and } r^{\frac{h}{2}} = r^h.$$

Note that this artificial level is identical to the finest grid in every respect. Then we have the following algorithm for k levels:

Algorithm $\text{MG}(j, r^{\frac{L}{2}}, \alpha_0^{\frac{L}{2}})$, where $L = 2^{j-1}h$, $j = 1:k$.

- Compute $r_0^L = I_{\frac{L}{2}}^L r^{\frac{L}{2}}$

- Relax ν_1 times on $B^L \alpha^L = r_0^L$ with initial guess α_0^L

- Compute $r_1^L = r_0^L - B^L \alpha^L$ where $|||r_1^L||| \leq \rho^L |||r^{\frac{L}{2}}|||$

- if $j < k$ (not the coarsest grid) then $e^{2L} \leftarrow \text{MG}(j+1, r_1^L, 0)$

- Correct $\alpha^L \leftarrow \alpha^L + I_{2L}^L e^{2L}$

- Compute $r_2^L = r_0^L - B^L \alpha^L$

- Relax ν_2 times on $B^L \alpha^L = r_0^L$

- Compute $r_3^L = r_0^L - B^L \alpha^L$ where $|||r_3^L||| \leq \epsilon^L |||r_2^L|||$

- Return α^L

The side conditions occur in steps 3 and 8 of the algorithm, and involve the constants ρ^L and ϵ^L. For convergence of the method, we require that these side conditions on the relaxation scheme be satisfied. For the first relaxation, on the way down into the V-cycle, the factor ρ^L is the amount the norm of the residual is reduced after relaxation on level L as compared to the residual before relaxation on level $\frac{L}{2}$. For the second (optional) relaxation, on the way up out of the V-cycle, the factor ϵ^L measures residual reduction before and after relaxation on level L. If this relaxation is not performed, then $\epsilon^L = 1$. Note that these factors are functions of L, and change as the algorithm moves from level to level. We also want the restriction operator satisfy the condition

$$|||(I - (I_L^{2L})^{-1} I_L^{2L})u||| \leq \delta^L |||u|||, \quad u \in \Omega^L.$$

Our restriction operator I_L^{2L} is not invertible, so we approximate it with its pseudo-inverse as

$$(I_L^{2L})^{-1} \approx (I_L^{2L})^\dagger = \frac{1}{2}I_{2L}^L,$$

and require this approximation to satisfy the condition instead. Before we give the main theorem, a lemma is required.

Lemma 6.4: *If the variational properties hold with $c = 1$, then the interpolation operator I_{2L}^L is norm-preserving in the energy norm.*

Proof:

$$
\begin{aligned}
|||v^{2L}||| &= \langle v^{2L}, B^{2L}v^{2L}\rangle \\
&= \langle v^{2L}, I_L^{2L}B^L I_{2L}^L v^{2L}\rangle \\
&= \langle (I_L^{2L})^T v^{2L}, B^L I_{2L}^L v^{2L}\rangle \\
&= \langle (I_{2L}^L v^{2L}), B^L(I_{2L}^L v^{2L})\rangle \\
&= |||I_{2L}^L v^{2L}|||
\end{aligned}
$$

∎

The next theorem follows the basic outline given in [Ref. 29], and uses the notation of Algorithm MG.

Theorem 6.3: *Let $r^{\frac{L_j}{2}}$ be the initial residual on level $\frac{L_j}{2}$ at some step, where $L_j = 2^{j-1}h$, $1 \le j \le k$, where k denotes the number of levels of the scheme. Let $(I_{\frac{L}{2}}^L)^{-1} : \Omega^L \to \Omega^{\frac{L}{2}}$ be approximated by its pseudoinverse $\frac{1}{2}I_{2L}^L$. Further, assume that there exists a $\delta^L \in \mathcal{R}$ such that*

$$|||(I - (I_L^{2L})^\dagger I_L^{2L})u||| \le \delta^L |||u|||, \quad u \in \Omega^L.$$

Define $E_1 = \epsilon^h \rho^h$ and $E_j = \epsilon^L \rho^L(\delta^L + E_{j-1})$, $j > 1$, where $L = 2^{j-1}h$. Then $|||(I_{\frac{L}{2}}^L)^\dagger r^{\frac{L}{2}}||| \le E_j |||r^{\frac{L}{2}}|||$.

Proof: The proof is by induction on j. For $j = 1$, we have

$$|||(I_{\frac{h}{2}}^h)^\dagger r_3^h||| = |||r_3^h||| \le \epsilon^h |||r_2^h|||,$$

124

but since there is only one level, $r_2^h = r_1^h$. So

$$|||(I_{\frac{h}{2}}^h)^\dagger r_3^h||| \leq \epsilon^h |||r_1^h||| \leq \epsilon^h \rho^h |||r_0^h||| = E_1 |||r^{\frac{h}{2}}|||.$$

Now, assume the claim holds for all levels $j < k$. Then

$$|||(I_{\frac{L}{2}}^L)^\dagger r_3^L||| = \frac{1}{2}|||I_L^{\frac{L}{2}} r_3^L||| = \frac{1}{2}|||r_3^L|||,$$

since the operator $I_L^{\frac{L}{2}}$ is norm preserving. Therefore

$$|||(I_{\frac{L}{2}}^L)^\dagger r_3^L||| = \frac{1}{2}|||r_3^L||| \leq \frac{1}{2}\epsilon^L|||r_2^L|||$$

$$= \frac{1}{2}\epsilon^L|||r_0^L - B^L(\alpha^L + I_{2L}^L e^{2L})|||$$

$$= \frac{1}{2}\epsilon^L|||r_1^L - B^L I_{2L}^L e^{2L}|||$$

$$= \frac{1}{2}\epsilon^L|||r_1^L - (I_L^{2L})^\dagger B^{2L} e^{2L}|||$$

$$= \frac{1}{2}\epsilon^L|||r_1^L - (I_L^{2L})^\dagger(I_L^{2L} r_1^L - r_3^{2L})|||$$

$$= \frac{1}{2}\epsilon^L|||(I - (I_L^{2L})^\dagger I_L^{2L})r_1^L + (I_L^{2L})^\dagger r_3^{2L}|||$$

$$\leq \frac{1}{2}\epsilon^L|||(I - (I_L^{2L})^\dagger I_L^{2L})r_1^L||| + \frac{1}{2}\epsilon^L|||(I_L^{2L})^\dagger r_3^{2L}|||$$

$$\leq \frac{1}{2}\epsilon^L \delta^L|||r_1^L||| + \frac{1}{2}\epsilon^L|||(I_L^{2L})^\dagger r_3^{2L}|||$$

$$\leq \frac{1}{2}\epsilon^L \delta^L|||r_1^L||| + \frac{1}{2}\epsilon^L E_{j-1}|||r_1^L|||$$

$$= \frac{1}{2}\epsilon^L(\delta^L + E_{j-1})|||r_1^L|||$$

$$\leq \frac{1}{2}\epsilon^L \rho^L(\delta^L + E_{j-1})|||r^{\frac{L}{2}}|||$$

$$= \frac{1}{2}E_j|||r^{\frac{L}{2}}|||$$

∎

The constants ϵ^L, ρ^L, δ^L and E_j indicate the performance of the multilevel routine. As long as $E_j < 1$, the method is converging. Unfortunately, there is no *a priori* way to determine the values of these constants, as they change from level to level and

cycle to cycle as the algorithm is executing. They can, however, be calculated and monitored during execution so that some idea of the performance of the method can be gained.

C. ANALYSIS OF PERFORMANCE

We compare the performance of Gauss-Seidel alone and the PMLV method by considering work required to reduce the norm of the residual. Let one sweep of Gauss Seidel on the finest level be defined as one work unit (WU), which is an $O(N^2)$ operation. The work required for one V-cycle can be computed in the following fashion. At each level, we perform $\nu_1 + \nu_2$ sweeps of Gauss-Seidel and compute a residual. Since computing the residual is an $O(N^2)$ operation, we let it be equivalent to the work of one sweep of Gauss-Seidel. As the problem is coarsened, the size of the matrix equation to be solved is reduced by a factor of 4 at each level. So for a V-cycle, the work required is

$$(\nu_1 + \nu_2 + 1)(1 + \frac{1}{4} + \frac{1}{16} + \cdots).$$

For these tests we use $\nu_1 = 2$ iterations going into the V-cycle and $\nu_2 = 1$ iteration coming out of it, so one V-cycle requires approximately $\frac{16}{3}$ WU. In all cases the problem is coarsened to the coarsest possible level, i.e. one ray per view for a problem of size $M \times M$ at the coarsest level. Figure 44 compares the performance of Gauss-Seidel to the PMLV algorithm for a problem of geometry 32 detectors over 20 angles. This performance is typical of that obtained from numerous experiments.

126
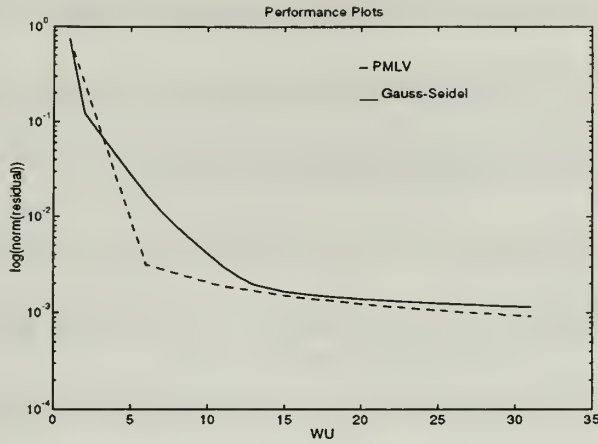
Figure 44. *Comparison of the performance of Gauss-Seidel and PMLV.*

It is clear from the figure that PMLV initially performs superior to Gauss-Seidel, but then this increase in performance stalls out, so that eventually both routines perform in about the same manner. However, examination of the slopes of the curves indicate that further iteration may favor PMLV. We return to the singular value decomposition to analyze this behavior. Consider the problem $B\vec{\alpha} = \vec{f}$, whose exact solution in the least squares sense is

$$\vec{\alpha} = B^{\dagger}\vec{f},$$

where $B^{\dagger}$ is the psuedo-inverse of B. In terms of the SVD, $\vec{\alpha}$ can be expressed as

$$\vec{\alpha} = \left[\sum_{i=1}^{r}\frac{1}{\sigma_i}\vec{v_i}\vec{u_i}^T\right]\vec{f} = \sum_{i=1}^{r}\frac{1}{\sigma_i}\vec{v_i} < \vec{u_i}, \vec{f} >,$$

where $r = \text{rank}(B)$. If there is *measurement noise* in the data we are given to reconstruct, so that instead of $\vec{f}$ we have $\vec{f} + \vec{\varepsilon}$, then solution components corresponding to small singular values will magnify this noise. Problems of this nature are referred to as *ill-posed* . Components in the near null space, i.e., those with small singular values,

127

are those components that are slow to to be recovered. Thus continued iteration after the procedure stalls in an attempt to recover these slow components has the potential to corrupt the solution with magnified noise [Ref. 30, 31].

Such problems require some form of regularization to prevent the ill-posedness from completely corrupting the approximation. One way to regularize the problem [Ref. 14] is to simply stop iterating when the algorithm begins to stall. An ad hoc approach to this is to measure the difference between successive residual norms, and stop iterating when a tolerance is achieved. Perhaps a better stopping criterion exists. Recall from Theorem 6.3 that the PMLV algorithm reduces the norm of the residual at each step by some factor E_j. This number can be computed as the algorithm is executing, and can then be used to monitor its performance. It has been experimentally observed that when the algorithm begins to stall, E_j becomes larger than one in magnitude. It could be postulated that a stopping criterion for the iteration is to monitor the magnitude of E_j, and then terminate execution when $E_j > 1$. For the above example, E_j became greater than one in magnitude after the third V-cycle, which coincides with the stalling of convergence in Figure 44.

The PMLV algorithm applied to the natural pixel discretized problem reconstructs images quite well. The following series of figures depicts actual and reconstructed images for two brain phantoms and a woman's face. Note that this last image is not the type of image for which the method is developed, but is included so that the reader unfamiliar with radiological imagery has something familiar to observe.

128
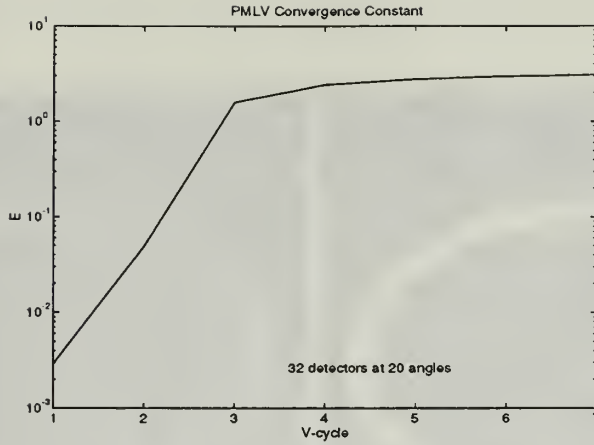
Figure 45. *Plot of the PMLV convergence constant E_j by sweep.*

In the next chapter, a multilevel fast adaptive composite method (FAC) approach will be investigated and applied to the *spotlight* CAT problem, which involves getting high resolution in one piece of a larger image without discretizing the global problem to that level of resolution. The natural pixel discretization approach will be taken throughout this development as well.
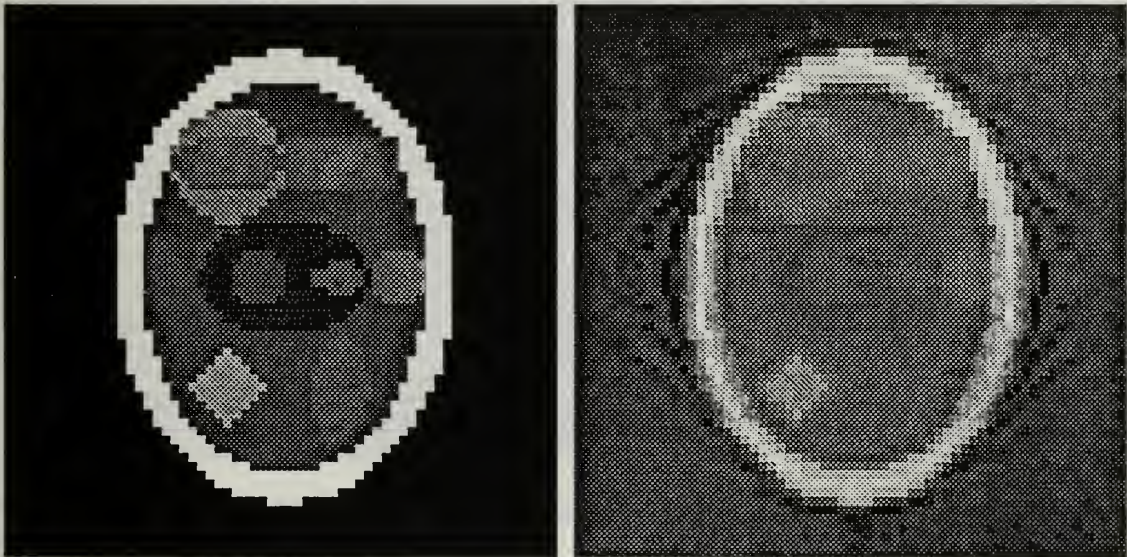
Figure 46. *Actual and Reconstructed Images - Brain Phantom 1.*

129
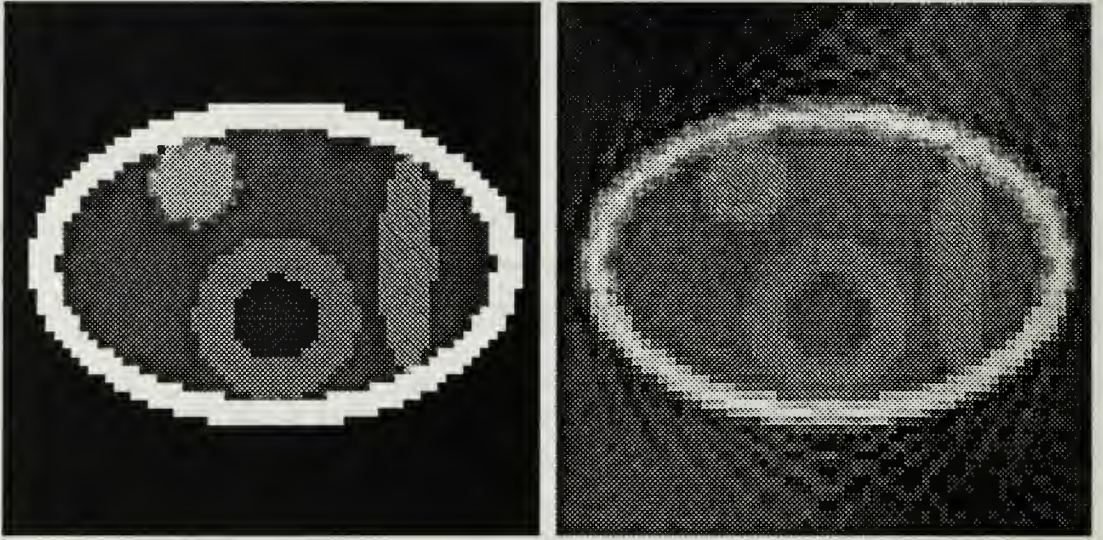
Figure 47. *Actual and Reconstructed Images - Brain Phantom 2.*

Figure 48. *Actual and Reconstructed Images - Face.*

VII. SPOTLIGHT COMPUTED TOMOGRAPHY

The image reconstruction techniques investigated to this point reconstruct the entire region of interest over which the data has been collected. In many cases, more detailed information may be required over a particular sub-area of the region. For example, a tumor might be suspected and the doctor wants a closer look. Collection of data over only the region containing the possible tumor result in inaccurate images, however, as only the sub-area is fully scanned. To overcome this problem, a multiresolution technique known as *spotlight* computed tomography (CT) can be employed. [Ref. 32, 33]

To utilize the spotlight technique, the sub-area is x-rayed at high resolution, while the remainder of the image is x-rayed at a lower resolution. The collection of high resolution data over only the sub-area reduces the size of the resulting linear system dramatically compared to uniform high resolution discretization. This reduced size in turn allows the problem to be solved in less time with fewer resources.

A. NATURAL PIXEL DISCRETIZATION

We again take the natural pixel approach to discretizing the problem. For ease of development, we initially restrict the refinement to two levels, and consider the general case later. Let $u(x, y)$ be the density function of the image to be reconstructed,

defined in a square region Ω of unit area. Let the sub-area to be *spotlighted* be defined in a square refinement region, Ω_R, that is contained within Ω. Assume that, following the methodology of Chapter IV, the image is x-rayed at coarse resolution over M angles with $N_1(j)$ source/detector pairs per angle, $1 \leq j \leq M$, for a total of $N = \sum_{j=1}^{M} N_1(j)$ x-rays that completely cover the image at each angle.

Refinement is achieved by x-raying Ω_R at a fine resolution. (However, it is assumed that all the data is collected at once. The x-ray data for the fine grid is not acquired at a later time). We also assume that these fine rays exactly partition the coarse rays over Ω_R, and that there are $N_2(j)$ such rays per angle, for a total of $P = \sum_{j=1}^{M} N_2(j)$ fine rays. Therefore the fine rays completely cover Ω_R at each angle. The following simple example will serve to illustrate this concept. Assume a geometry of three coarse rays over each of two angles, and that the refinement region is contained within the intersection of the center rays of each view, as shown in Figure 49 (a). Refinement is accomplished by dividing the center rays in half, as shown in Figure 49 (b).

Note that the coarse rays are global in nature, completely covering the image, while the fine rays are local, completely covering only Ω_R.

Once again, let the ray paths be thought of as natural pixels, and introduce characteristic strip functions corresponding to these pixels. Let ψ_j^{2h}, for $j = 1 : N$, be the jth coarse strip function, and ψ_k^h, $k = 1 : P$, be the the kth fine (refinement) strip function. Here we let the superscripts indicate which resolution is

Ω_R

(a)   (b)
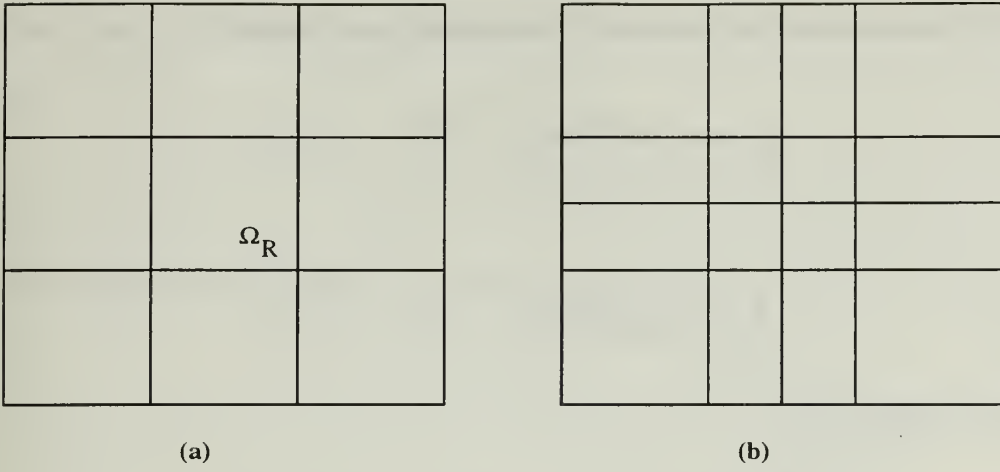
Figure 49. *An example of grid refinement.*

being considered, i.e. h denotes fine strips, $2h$ denotes coarse strips, and let $\underline{h}$ denote

a *composite* combination of both resolutions together.

Define the operator $A^{\underline{h}} : H \to \mathcal{R}^{N+P}$ by

$$
A^{\underline{h}} u = \begin{pmatrix} < \psi_1^{2h}, u > \\ < \psi_2^{2h}, u > \\ \vdots \\ < \psi_N^{2h}, u > \\ < \psi_1^{h}, u > \\ < \psi_2^{h}, u > \\ \vdots \\ < \psi_P^{h}, u > \end{pmatrix} = \begin{pmatrix} A^{2h} u^{2h} \\ A^{h} u^{h} \end{pmatrix},
$$

where $A^{2h} : H \to \mathcal{R}^N$ and $A^h : H \to \mathcal{R}^P$.

The data vectors are modeled by the integral operators

$$\int_{\mathcal{R}^2} u(x,y)\psi_j^{2h}dxdy = f_j^{2h}, \ 1 \le j \le N,$$

and

$$\int_{\mathcal{R}^2} u(x,y)\psi_k^h dxdy = f_k^h, \ 1 \le k \le P,$$

giving the system

$$A^{\underline{h}}u^{\underline{h}} = \begin{pmatrix} \int_{\mathcal{R}^2} u(x,y)\psi_1^{2h}dxdy \\ \int_{\mathcal{R}^2} u(x,y)\psi_2^{2h}dxdy \\ \vdots \\ \int_{\mathcal{R}^2} u(x,y)\psi_N^{2h}dxdy \\ \int_{\mathcal{R}^2} u(x,y)\psi_1^h dxdy \\ \int_{\mathcal{R}^2} u(x,y)\psi_2^h dxdy \\ \vdots \\ \int_{\mathcal{R}^2} u(x,y)\psi_P^h dxdy \end{pmatrix} = \begin{pmatrix} f_1^{2h} \\ f_2^{2h} \\ \vdots \\ f_N^{2h} \\ f_1^h \\ f_2^h \\ \vdots \\ f_P^h \end{pmatrix},$$

where the f_j^{2h} and f_k^h are the coarse and fine ray projection data for the image, respectively. Following the course of action outlined in Chapter IV, we seek the least squares solution to this system, which is given by $(A^{\underline{h}})^* \vec{\alpha}^{\underline{h}}$, where $\vec{\alpha}^{\underline{h}} \in \mathcal{R}^{N+P}$ defined as

$$\vec{\alpha}^{\underline{h}} = \begin{pmatrix} \vec{\alpha}^{2h} \\ \vec{\alpha}^h \end{pmatrix},$$

solves the system $A^{\underline{h}}(A^{\underline{h}})^*\vec{\alpha}^{\underline{h}} = \vec{f}^{\underline{h}}$. Once $\vec{\alpha}^{\underline{h}}$ is found, the least squares solution is given by

$$u(x,y) = (A^{\underline{h}})^*\vec{\alpha}^{\underline{h}}.$$

The operator $(A^{\underline{h}})^* : \mathcal{R}^{N+P} \to H$ is defined by

$$(A^{\underline{h}})^*\vec{\alpha}^{\underline{h}} = [\psi_1^{2h} \ \psi_2^{2h} \ \cdots \ \psi_N^{2h} \ \psi_1^h \ \psi_2^h \ \cdots \ \psi_P^h] \begin{pmatrix} \alpha_1^{2h} \\ \alpha_2^{2h} \\ \vdots \\ \alpha_N^{2h} \\ \alpha_1^h \\ \alpha_2^h \\ \vdots \\ \alpha_P^h \end{pmatrix}$$

$$= [(A^{2h})^* \ (A^h)^*] \begin{pmatrix} \vec{\alpha}^{2h} \\ \vec{\alpha}^h \end{pmatrix}$$

$$= \sum_{j=1}^N \alpha_j^{2h}\psi_j^{2h} + \sum_{k=1}^P \alpha_k^h\psi_k^h,$$

where $(A^{2h})^* : \mathcal{R}^N \to H$ and $(A^h)^* : \mathcal{R}^P \to H$ are defined in the expected way.

Thus the image density is represented as a linear combination of the characteristic strip functions by

$$u(x,y) = \sum_{j=1}^N \alpha_j^{2h}\psi_j^{2h} + \sum_{k=1}^P \alpha_k^h\psi_k^h. \tag{7.1}$$

135

Define

$$B^h = A^h(A^h)^*,$$

where $B^h : \mathcal{R}^{N+P} \to \mathcal{R}^{N+P}$. The entries in B^h can be calculated by substituting (7.1) for $u(x,y)$ in $A^h u^h = \vec{f}^h$, yielding

$$\int_{\mathcal{R}^2} (\sum_{i=1}^{N} \alpha_i^{2h} \psi_i^{2h} + \sum_{\ell=1}^{P} \alpha_\ell^h \psi_\ell^h) \psi_j^{2h} = f_j^{2h}, \quad j = 1 : N$$

$$\int_{\mathcal{R}^2} (\sum_{i=1}^{N} \alpha_i^{2h} \psi_i^{2h} + \sum_{\ell=1}^{P} \alpha_\ell^h \psi_\ell^h) \psi_k^h = f_k^h, \quad k = 1 : P,$$

which can be expanded as

$$\int_{\mathcal{R}^2} (\sum_{i=1}^{N} \alpha_i^{2h} \psi_i^{2h}) \psi_j^{2h} + \int_{\mathcal{R}^2} (\sum_{\ell=1}^{P} \alpha_\ell^h \psi_\ell^h) \psi_j^{2h} = f_j^{2h}, \quad j = 1 : N$$

$$\int_{\mathcal{R}^2} (\sum_{i=1}^{N} \alpha_i^{2h} \psi_i^{2h}) \psi_k^h + \int_{\mathcal{R}^2} (\sum_{\ell=1}^{P} \alpha_\ell^h \psi_\ell^h) \psi_k^h = f_k^h, \quad k = 1 : P,$$

finally yielding

$$\sum_{i=1}^{N} \alpha_i^{2h} < \psi_i^{2h}, \psi_j^{2h} > + \sum_{\ell=1}^{P} \alpha_\ell^h < \psi_\ell^h, \psi_j^{2h} > = f_j^{2h}, \quad j = 1 : N$$

$$\sum_{i=1}^{N} \alpha_i^{2h} < \psi_i^{2h}, \psi_k^h > + \sum_{\ell=1}^{P} \alpha_\ell^h < \psi_\ell^h, \psi_k^h > = f_k^h, \quad k = 1 : P.$$

This last expression is a block linear system, which can be written as

$$B^h \vec{\alpha}^h = \begin{pmatrix} B^{2h2h} & B^{2hh} \\ B^{h2h} & B^{hh} \end{pmatrix} \begin{pmatrix} \vec{\alpha}^{2h} \\ \vec{\alpha}^h \end{pmatrix} = \begin{pmatrix} \vec{f}^{2h} \\ \vec{f}^h \end{pmatrix}.$$

Thus $B^h \vec{\alpha}^h = \vec{f}^h$ is the natural pixel discretization of the spotlight CT problem $A^h u^h = \vec{f}^h$, where $u^h = (A^h)^* \vec{\alpha}^h$.

136

It is possible to refine the fine strip functions again in the same fashion, which allows for recursive refinement to as fine a level of discretization as is needed to resolve the image. Before investigating how to best solve this linear system, we first analyze the matrix $B^{\underline{h}}$, characterize its null space, and discuss some of its other interesting properties.

B.   PROPERTIES OF THE SYSTEM MATRIX

In our analysis of the matrix $B^{\underline{h}}$, much of the theory previously developed in Chapter IV will be directly applicable. The following is one such result, the proof of which follows directly those of Lemma 4.1 and Theorem 4.2.

Lemma 7.1: *The matrix $B^{\underline{h}}$ is non-negative, symmetric, and positive semi-definite.*

From above, we know that $B^{\underline{h}}$ can be written as

$$B^{\underline{h}} = \begin{pmatrix} B^{2h,2h} & B^{2h,h} \\ B^{h,2h} & B^{h,h} \end{pmatrix}.$$

and that each of these four blocks has a block structure of its own. The matrix B^{2h2h} is formed from the intersections of the coarse rays with themselves, and is exactly the matrix B analyzed in Chapter IV. B^{hh} is formed from the intersection of the fine rays with themselves. Like B^{2h2h}, it is block $M \times M$ with the diagonal blocks being diagonal matrices. However, it lacks the summability properties of B^{2h2h}. Finally, the off-diagonal block matrices $B^{2hh} = (B^{h2h})^{T}$ are formed by the intersections of

137
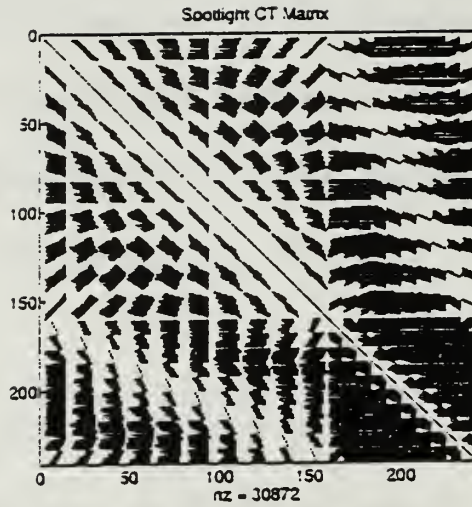
Figure 50. *Block structure of the Spotlight matrix.*

coarse rays with fine rays. For example, B^{hh} would have the form

$$B^{hh} = \begin{pmatrix} B^{hh}_{11} & B^{hh}_{12} & \cdots & B^{hh}_{1M} \\ B^{hh}_{21} & B^{hh}_{22} & \cdots & B^{hh}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ B^{hh}_{M1} & B^{hh}_{M2} & \cdots & B^{hh}_{MM} \end{pmatrix},$$

where B^{hh}_{ij} has elements corresponding to the areas of intersection of the fine rays at angle ϕ_i with the fine rays at angle ϕ_j. The other two blocks have a similar structure and similar interpretation. The sizes of the blocks are B^{2h2h}_{ij} is $N_1(i) \times N_1(j)$, B^{hh}_{ij} is $N_2(i) \times N_2(j)$, B^{h2h}_{ij} is $N_1(i) \times N_2(j)$, and B^{2hh}_{ij} is $N_2(i) \times N_1(j)$. Figure 50 illustrates the block structure of a typical matrix, in which only the nonzero entries appear in black.

138

The off-diagonal block matrices possess summability as given by

Lemma 7.2: *The off-diagonal blocks of $B^{\underline{h}}$ exhibit the following summability properties:*

a) *Let $\vec{r}_k$ be the kth row of $B^{\underline{h}}$. The elements of $\vec{r}_k$ in any block B_{ij}^{h2h} of B^{h2h} sum to the value of the corresponding diagonal element in the kth row of B^{hh}.*

b) *Let $\vec{r}_k$ be the kth row of $B^{\underline{h}}$. The sum of the elements of $\vec{r}_k$ in any block B_{ij}^{2hh} of B^{2hh} is equal to the sum of the elements in r_k in block B_{ij}^{2h2h} of B^{2h2h} corresponding to the coarse rays from which they were partitioned.*

c) *Let $\vec{r}_k$ be the kth row of $B^{\underline{h}}$. The sum of the elements of $\vec{r}_k$ in any block B_{ij}^{h2h} of B^{h2h} corresponding to coarse rays that are refined is equal to the sum of the elements in r_k in block B_{ij}^{hh} of B^{hh}.*

d) *The above three results hold for the columns as well.*

**Proof:** To prove part a, consider how the elements of row k for block B_{ij}^{h2h} corresponding to angle ϕ_i are formed. The entries of row k for this block are the areas of intersection of fine strip k with all of the coarse strips for angle ϕ_j. Since the coarse rays at angle ϕ_j must cover the entire image, they must cover fine strip k as well. To prove part b, note that the elements of r_k in B_{ij}^{2hh} are the areas of intersection of the fine strips at angle ϕ_j with coarse strip k from angle ϕ_i. The elements of r_k in B_{ij}^{2h2h} corresponding to the coarse rays at angle ϕ_j which are refined are their areas of intersection with coarse ray k at angle ϕ_i. Since the fine rays are an exact partition of these coarse rays, the sums must be equal. This is geometrically illustrated in Figure 51. For part c, again note that the elements of r_k in B_{ij}^{h2h} corresponding to coarse rays at angle ϕ_j that are refined are their areas of intersection with fine ray k from angle ϕ_i. The elements of r_k in B_{ij}^{hh} are the areas of intersection of the fine strips at angle ϕ_j with fine strip k at angle ϕ_i. Due to exact partitioning, these sums must be equal. This is geometrically illustrated in Figure 52. Part d follows from symmetry. ∎

Another summability property exists, based on how the refined rays are defined, that has a role in determining the rank of $B^{\underline{h}}$. Recall that by assumption the fine rays exactly partition the coarse rays. Refinement is carried out by selecting a
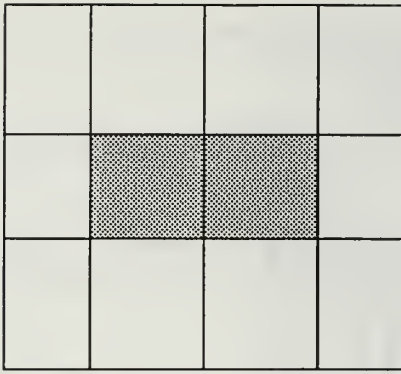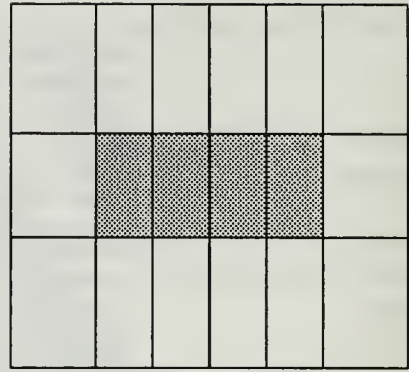
Figure 51. *Geometry of Lemma 7.2(b).*
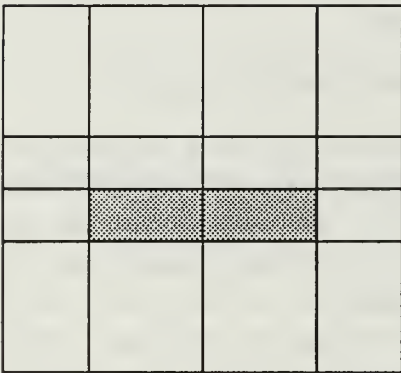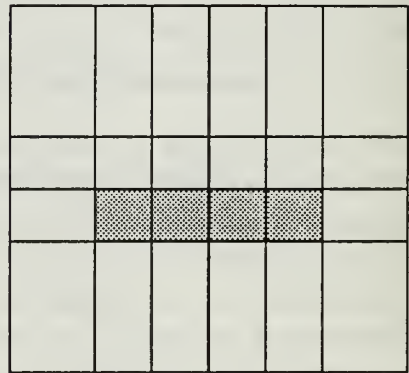
Figure 52. *Geometry of Lemma 7.2(c).*

coarse ray and dividing it into a number of *thinner* rays. Now, consider the kth row of the matrix $B^{\underline{h}}$. The entries in this row are determined by finding the areas of intersection of the kth ray path with all $N + P$ ray paths defined in the geometry of the problem, taken in sequence. This leads to the following theorem.

Theorem 7.1: *The sum of the rows of $B^{\underline{h}}$ corresponding to a set of fine rays that were formed by subdividing any one coarse ray equals that row of $B^{\underline{h}}$ corresponding to the coarse ray in question.*

140

Proof: Let ψ_k^{2h} be a coarse strip function which is refined into r fine strip functions, such that

$$\psi_k^{2h} = \sum_{i=j+1}^{j+r} \psi_i^h.$$

Let $\vec{b}_k^h$ be the kth row of $B^{\underline{h}}$, given by

$$\vec{b}_k^h = \begin{pmatrix} <\psi_k^{2h},\psi_1^{2h}> \\ <\psi_k^{2h},\psi_2^{2h}> \\ \vdots \\ <\psi_k^{2h},\psi_N^{2h}> \\ <\psi_k^{2h},\psi_1^h> \\ <\psi_k^{2h},\psi_2^h> \\ \vdots \\ <\psi_k^{2h},\psi_P^h> \end{pmatrix}^T = \begin{pmatrix} <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_1^{2h}> \\ <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_2^{2h}> \\ \vdots \\ <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_N^{2h}> \\ <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_1^h> \\ <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_2^h> \\ \vdots \\ <\sum_{i=j+1}^{j+r}\psi_i^h,\psi_P^h> \end{pmatrix}^T .$$

Expanding the sums yields

$$\vec{b}_k^h = \begin{pmatrix} <\psi_{j+1}^h,\psi_1^{2h}> + <\psi_{j+2}^h,\psi_1^{2h}> + \cdots + <\psi_{j+r}^h,\psi_1^{2h}> \\ <\psi_{j+1}^h,\psi_2^{2h}> + <\psi_{j+2}^h,\psi_2^{2h}> + \cdots + <\psi_{j+r}^h,\psi_2^{2h}> \\ \vdots \\ <\psi_{j+1}^h,\psi_N^{2h}> + <\psi_{j+2}^h,\psi_N^{2h}> + \cdots + <\psi_{j+r}^h,\psi_N^{2h}> \\ <\psi_{j+1}^h,\psi_1^h> + <\psi_{j+2}^h,\psi_1^h> + \cdots + <\psi_{j+r}^h,\psi_1^h> \\ <\psi_{j+1}^h,\psi_2^h> + <\psi_{j+2}^h,\psi_2^h> + \cdots + <\psi_{j+r}^h,\psi_2^h> \\ \vdots \\ <\psi_{j+1}^h,\psi_P^h> + <\psi_{j+2}^h,\psi_P^h> + \cdots + <\psi_{j+r}^h,\psi_P^h> \end{pmatrix}^T$$

$$= \vec{b}_{j+1}^h + \vec{b}_{j+2}^h + \cdots \vec{b}_{j+r}^h .$$

$\blacksquare$

An immediate implication of this result is the following

Corollary 7.1: *The rank of $B^{\underline{h}}$ can be no greater than $N + P$ less the number of coarse rays that are refined.*

Proof: Let m be the number of coarse rays that are refined. Each of these m rays leads to a dependent set of rows in the matrix $B^{\underline{h}}$, as per Theorem 7.1. Therfore the rank of $B^{\underline{h}}$ cannot exceed $N + P - m$. $\blacksquare$

These summability properties and some additional analysis will allow us to determine the rank of $B^{\underline{h}}$ and characterize its null space. Assume $\vec{v}^{\underline{h}} \in NS(B^{\underline{h}})$. Then

$$B^{\underline{h}}\vec{v}^{\underline{h}} = \begin{pmatrix} B^{2h2h} & B^{2hh} \\ B^{h2h} & B^{hh} \end{pmatrix} \begin{pmatrix} \vec{v}^{2h} \\ \vec{v}^h \end{pmatrix} = 0.$$

Matrix multiplication results in

$$B^{2h2h}\vec{v}^{2h} + B^{2hh}\vec{v}^h = 0 \tag{7.2}$$

$$B^{h2h}\vec{v}^{2h} + B^{hh}\vec{v}^h = 0 \tag{7.3}$$

We now investigate under what conditions (7.2) and (7.3) are satisfied. We saw from Theorem 4.3 that a vector is in the null space of the matrix $B = B^{2h2h}$ if and only if it is constant by angle with the constants summing to zero. That is, the image at each view is a shade of grey, and when all views are superimposed the result is a black, or invisible, image. The concept of a vector being constant by angle holds for the composite matrix $B^{\underline{h}}$ as well, with some modifications. Consider the characteristic strip functions for a refined image over one angle, as shown in Figure 53.

A vector $\vec{v}^{\underline{h}}$, where

$$\vec{v}^{\underline{h}} = \begin{pmatrix} \vec{v}^{2h} \\ \vec{v}^h \end{pmatrix},$$

142
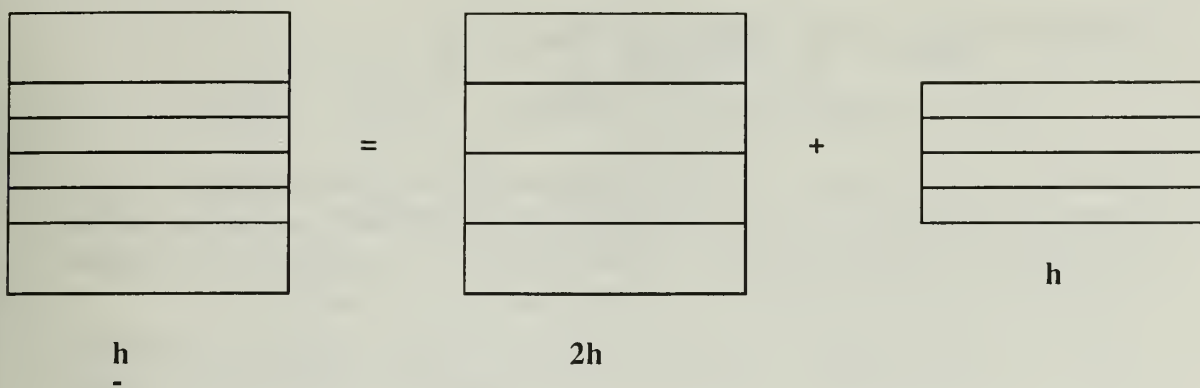
Figure 53. *Characteristic strip functions for a refined image over one angle.*

is constant by angle with respect to the matrix $B^{\underline{h}}$ correspnding to this geometry if

$$\vec{v}^{\underline{h}} = (\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_1 \ \gamma_1 \ \gamma_1 \ \gamma_2 \ \gamma_2)^T,$$

where

$$\alpha_1 = \alpha_2 + \gamma_1 = \alpha_3 + \gamma_2.$$

The contribution in the refinement region from a coarse strip, together with the contributions from its corresponding fine strips, equals the contributions from those coarse strips not in the refinement region. The overall result is that the composite collection of strips forms a uniform grey image.

We can generalize this idea for a geometry of M angles by requiring the composite subimage at each angle to be uniform grey. Define α_j to the constant for the unrefined strips from angle ϕ_j, $\alpha_j k$ for $k = 1 : m_j$ to be the values of the m_j coarse strips that cover the refinement region from angle ϕ_j, and $\gamma_j k$ for $k = 1 : m_j$ the values of the refinement strips at angle ϕ_j that partition the kth coarse strip in the refinement region.

Definition 7.1: Let $\vec{v}^{\underline{h}} = (\vec{v}^{2h} \ \vec{v}^h)^T$, where

$$\vec{v}^{2h} = (\vec{w}_1 \ \vec{w}_2 \ \cdots \ \vec{w}_M)^T \quad \text{and} \quad \vec{v}^h = (\vec{y}_1 \ \vec{y}_2 \ \cdots \ \vec{y}_M)^T.$$

Subvector $\vec{w}_i$ is of length $N_1(i)$ and subvector $\vec{y}_j$ is of length $N_2(j)$. Let m be the total number of coarse strips that are partitioned, and let m_j be the number of such strips at angle ϕ_j, so that $\sum_{j=1}^{M} m_j = m$. Then $\vec{v}^{\underline{h}}$ is *composite constant by angle* with respect to $B^{\underline{h}}$ generated over M angles if

$$\vec{w}_j = (\alpha_j \ \cdots \ \alpha_j \ \bar{\alpha}_{j1} \ \bar{\alpha}_{j2} \ \cdots \ \bar{\alpha}_{jm_j} \ \alpha_j \ \cdots \ \alpha_j)^T$$

and

$$\vec{y}_j = (\gamma_{j1} \ \cdots \ \gamma_{j1} \ \gamma_{j2} \ \cdots \ \gamma_{j2} \ \cdots \ \gamma_{jm_j} \ \cdots \ \gamma_{jm_j})^T,$$

subject to the constraints

$$\alpha_j = \bar{\alpha}_{ji} + \gamma_{ji}, \quad i = 1 : m_j.$$

Obviously, if $\vec{v}^{\underline{h}}$ is composite constant by angle, then the image it defines is a constant image with a value equal to the sum of the α_i's which define the M uniform grey subimages. Vectors in the $NS(B^{\underline{h}})$ are composite constant by angle and correspond to constant images, as will be shown in the following theorem.

Theorem 7.2: $\vec{v}^{\underline{h}} \in NS(B^{\underline{h}})$ *if and only if it is composite constant by angle with* $\sum_{i=1}^{M} \alpha_i = 0$.

Proof: Let $\vec{v}^{\underline{h}}$ be composite constant by angle with $\sum_{i=1}^{M} \alpha_i = 0$. We can write

$$B^{\underline{h}}\vec{v}^{\underline{h}} = \begin{pmatrix} B^{2h2h}\vec{v}^{2h} + B^{2hh}\vec{v}^h \\ B^{h2h}\vec{v}^{2h} + B^{hh}\vec{v}^h \end{pmatrix} = \begin{pmatrix} \vec{z}^{2h} \\ \vec{z}^h \end{pmatrix}.$$

Consider the contribution of $B^{\underline{h}}\vec{v}^{\underline{h}}$ corresponding to angle ϕ_j toward the ith component of $\vec{z}^{2h}$. This can be expressed as

$$\alpha_j \sum_{k \in \Omega^{2h}} \beta_{ik}^{2h2h} + \sum_{\substack{k=1 \\ \ell(k) \in \Omega^h}}^{m_j} \bar{\alpha}_{jk}\beta_{i\ell(k)}^{2h2h} + \sum_{k_1} \gamma_{j1}\beta_{ik_1}^{2hh} + \cdots + \sum_{k_{m_j}} \gamma_{jm_j}\beta_{ik_{m_j}}^{2hh},$$

which after applying Lemma 7.2(b)

$$= \alpha_j \sum_{k \in \Omega^{2h}} \beta_{ik}^{2h2h} + \sum_{\substack{k=1 \\ \ell(k) \in \Omega^h}}^{m_j} \bar{\alpha}_{jk} \beta_{i\ell(k)}^{2h2h} + \sum_{\substack{k=1 \\ \ell(k) \in \Omega^h}}^{m_j} \gamma_{jk} \beta_{i\ell(k)}^{2h2h}$$

$$= \alpha_j \sum_{k \in \Omega^{2h}} \beta_{ik}^{2h2h} + \alpha_j \sum_{\substack{k=1 \\ \ell(k) \in \Omega^h}}^{m_j} \beta_{i\ell(k)}^{2h2h}$$

$$= \beta_{ii} \alpha_j,$$

by Lemma 4.2(b) and the fact that $\vec{v}^{\underline{h}}$ is composite constant by angle. Since angle ϕ_j was arbitrarly chosen, we have

$$z_i^{2h} = \beta_{ii}(\alpha_1 + \alpha_2 + \cdots + \alpha_M) = 0.$$

Since row i was arbitrarily chosen as well, we have $\vec{z}^{2h} = 0$.

Now, consider the contribution of $B^{\underline{h}} \vec{v}^{\underline{h}}$ corresponding to angle ϕ_j toward the kth component of $\vec{z}^h$. This can be expressed as

$$\alpha_j \sum_{i \in \Omega^{2h}} \beta_{ki}^{h2h} + \sum_{\substack{i=1 \\ \ell(i) \in \Omega^h}}^{m_j} \bar{\alpha}_{ji} \beta_{k\ell(i)}^{h2h} + \sum_{i_1} \gamma_{j1} \beta_{ki_1}^{hh} + \cdots + \sum_{i_{m_j}} \gamma_{jm_j} \beta_{ki_{m_j}}^{hh},$$

which after applying Lemma 7.2(c)

$$= \alpha_j \sum_{i \in \Omega^{2h}} \beta_{ki}^{h2h} + \sum_{\substack{i=1 \\ \ell(i) \in \Omega^h}}^{m_j} \bar{\alpha}_{ji} \beta_{k\ell(i)}^{h2h} + \sum_{\substack{i=1 \\ \ell(i) \in \Omega^h}}^{m_j} \gamma_{ji} \beta_{k\ell(i)}^{h2h}$$

$$= \alpha_j \sum_{i \in \Omega^{2h}} \beta_{ki}^{h2h} + \alpha_j \sum_{\substack{i=1 \\ \ell(i) \in \Omega^h}}^{m_j} \beta_{k\ell(i)}^{h2h}$$

$$= \beta_{kk} \alpha_j,$$

by Lemma 7.2(a) and the fact that $\vec{v}^{\underline{h}}$ is composite constant by angle. Since angle ϕ_j was arbitrarly chosen, we have

$$z_k^h = \beta_{kk}(\alpha_1 + \alpha_2 + \cdots + \alpha_M) = 0.$$

Since row k was arbitrarily chosen as well, we have $\vec{z}^h = 0$. Therefore, $\vec{v}^{\underline{h}} \in NS(B^{\underline{hh}})$.

Now, assume $\vec{v}^{\underline{h}} \in NS(B^{\underline{h}})$. Then $\vec{v}^{\underline{h}} \in NS((A^{\underline{h}})^*)$ and

$$(A^{\underline{h}})^* \vec{v}^{\underline{h}} = \sum_{i=1}^{N} v_i^{2h} \psi_j^{2h} + \sum_{j=1}^{P} v_j^h \psi_j^h = 0.$$

145
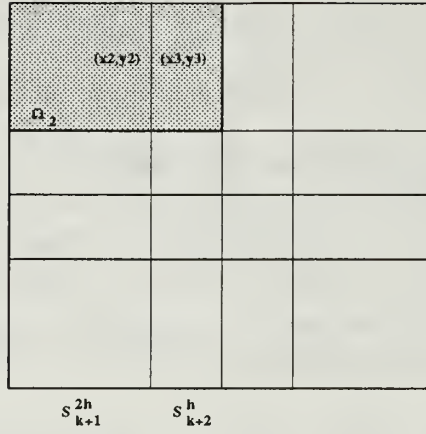
Figure 54. *Geometric illustration of Theorem 7.2.*

Consider three adjacent strips S_k^{2h}, S_{k+1}^{2h} and S_{k+2}^h from the same projection angle. Now, select points $(x_1, y_1) \in S_k^{2h} \cap \Omega_1$ and $(x_2, y_2) \in S_{k+1}^{2h} \cap \Omega_1$, where Ω_1 is the intersection of $(M-1)$ coarse grid strips, one from each profile and none from the profile containing S_k^{2h} and S_{k+1}^{2h}, and is not a part of the refinement region. This selection can always be made (see Figure 32). Therfore we have

$$\sum_{i=1}^{N} v_i^{2h} \psi_i^{2h}(x_1, y_1) + \sum_{j=1}^{P} v_j^h \psi_j^h(x_1, y_1) \; = \; \sum_{i=1}^{N} v_i^{2h} \psi_i^{2h}(x_2, y_2) + \sum_{j=1}^{P} v_j^h \psi_j^h(x_2, y_2) \; = \; 0.$$

Since the ψ_i^{2h} and ψ_j^h are characteristic functions, we can write

$$\sum_{i \in \Omega_1} v_i^{2h} + v_k^{2h} \; = \; \sum_{i \in \Omega_1} v_i^{2h} + v_{k+1}^{2h} \; = \; 0, \tag{7.4}$$

which implies that $v_k^{2h} = v_{k+1}^{2h}$. Therefore, outside the refinement region the coarse strips are constant across angles.

Now, select a point (x_3, y_3) such that $(x_2, y_2) \in S_{k+1}^{2h} \cap \Omega_2$ and $(x_3, y_3) \in S_{k+2}^h \cap \Omega_2$, where Ω_2 is the intersection of $(M-1)$ coarse grid strips, one from each profile and none from the profile containing S_{k+1}^{2h} and S_{k+2}^h. This selection can always be made. Superimpose all of the strips over the image at once, forming a grid of polygons. Since each coarse grid profile completely covers the image, a point in the interior of any polygon is contained in a strip from each of the M profiles. A point on the edge (not a vertex) of a polygon that separates a coarse grid and fine grid strip will be contained in Ω_2, with the edge separating strips S_{k+1}^{2h} and S_{k+2}^h. Moving a distance ϵ to either side and perpendicular to the edge will locate points (x_2, y_2) and (x_3, y_3). This geometry is illustrated in Figure 54.

146

Therefore we have

$$\sum_{i=1}^{N} v_i^{2h}\psi_i^{2h}(x_2,y_2)+\sum_{j=1}^{P} v_j^{h}\psi_j^{h}(x_2,y_2) \;=\; \sum_{i=1}^{N} v_i^{2h}\psi_i^{2h}(x_3,y_3)+\sum_{j=1}^{P} v_j^{h}\psi_j^{h}(x_3,y_3) \;=\; 0.$$

Since the ψ_i^{2h} and ψ_j^{h} are characteristic functions, we can write

$$\sum_{i\in\Omega_2} v_i^{2h} + v_{k+1}^{2h} \;=\; \sum_{i\in\Omega_2} v_i^{2h} + v_{k+2}^{2h} + v_{k+2}^{h} \;=\; 0,$$

which implies that $v_k^{2h} \;=\; v_{k+1}^{2h} \;=\; v_{k+2}^{2h} + v_{k+2}^{h}$. We can repeat this argument for the next fine strip that is adjacent to S_{k+2}^{h}, finding that its value added to the value of the coarse strip it partitions must also equal v_k^{2h}. Proceeding in this fashion across all fine strips for the selected angle will result in a view whose composite strips functions all have the same value of v_k^{2h}. Since the angle was arbitrarily selected, $\vec{v}^{h}$ is composite constant by angle. From (7.4) we have $\sum_{i\in\Omega_1} v_i^{2h} + v_k^{2h} \;=\; 0$. Since there is a v_j^{2h} from each of the M profile contained in this expression, the constants must sum to zero. ∎

An immediate consequence of this theorem is that images in the $NS(B^{h})$ are invisible. Another consequence is the following theorem, which relates the rank of B^{h} to the geometry used to x-ray the image.

Theorem 7.3: *Let $B^{h} \in \mathcal{R}^{N+P}$ be the composite natural pixel discretized matrix formed at M angles, and assume that the refinement region is covered by a total of m coarse strip functions subdivided into fine strip functions as outlined in the above discussion. Then the rank of B^{h} is $N+P-(M+m-1)$.*

Proof: Consider the degrees of freedom in selecting the values for the strip functions. For the views, the first $(M-1)$ values can be arbitrarily chosen, after which the final value is determined so that the M values sum to zero. The values of the m coarse strip functions which are refined are arbitrary as well, for a total of $M + m - 1$ degrees of freedom. Hence the rank of B^{h} is $N + P - (M + m - 1)$. ∎

The above results can be used to construct a basis for $NS(B^{h})$. A null space vector, in terms of Definition 7.1, must satisfy

$$\vec{w}_j \;=\; (\alpha_j \;\cdots\; \alpha_j \;\; \bar{\alpha}_{j1} \;\; \bar{\alpha}_{j2} \;\cdots\; \bar{\alpha}_{jm_j} \;\; \alpha_j \;\cdots\; \alpha_j)^T$$

and

$$\vec{y}_j = (\gamma_{j1} \cdots \gamma_{j1} \ \gamma_{j2} \cdots \gamma_{j2} \cdots \gamma_{jm_j} \cdots \gamma_{jm_j})^T,$$

subject to the constraints

$$\alpha_j = \bar{\alpha}_{ji} + \gamma_{ji}, \quad i = 1 : m_j$$

and

$$\sum_{j=1}^{M} \alpha_j = 0.$$

Given this form of a null space vector, the basis is constructed as follows. The first $M - 1$ basis vectors are obtained by setting $\vec{v}^{2h}$ equal to the $M - 1$ basis vectors of B, while maintaining $\vec{v}^h = 0$. This has the effect of letting all the $\beta_{jk} = 0$ and all the $\bar{\alpha}_{jk} = \alpha_j$. To construct the remaining m basis vectors, start with the $M - 1$ just constructed and for $k = 1 : m$ let $\bar{\alpha}_{jk} = 0$ and let $\beta_{jk} = \alpha_j$ in the appropriate places. We illustrate this procedure with a simple example. Consider three coarse strips over two angles, with the center strip of each view refined by splitting it in half. The basis is

$$\vec{q}_1 = (1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 0 \ 0 \ 0 \ 0)^T$$

$$\vec{q}_2 = (1 \ 0 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 0 \ 0)^T$$

$$\vec{q}_3 = (1 \ 1 \ 1 \ -1 \ 0 \ -1 \ 0 \ 0 \ -1 \ -1)^T$$

Let us now consider the matrix B^{hh}, which represent only the fine strip functions. This region is depicted for four views in the figure below. Note that the refined

148
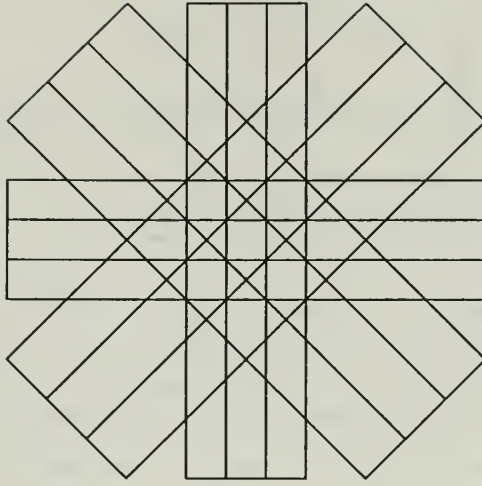
Figure 55. *Refined natural pixels for four angles.*

rays do not cover the whole image. Only the refined region Ω_R is totally covered by fine strip functions at each angle.

Now, for a vector $\vec{v}^h$ to be in the $NS(B^{hh})$, the refined natural pixels when overlayed must form a zero image. We know that for this to occur in Ω_R, $\vec{v}^h$ must be constant by angle with the constants summing to zero. This fact provides the next result.

Theorem 7.4: B^{hh} *is of full rank.*

Proof: Assume there exists a non-zero vector $\vec{v}^h \in \mathcal{R}^P$ such that $B^{hh}\vec{v}^h = 0$. Then $\vec{v}$ must be constant by angle with $\sum_{i=1}^{M} \alpha_i = 0$. If all the fine strips are superimposed over the image at once, they will divide it into a collection of polygons. By assumption, only the refinement region Ω_R is completely covered by strips at each angle. Therefore, there must exist a polygon P_k that borders Ω_R, and in particular is adjacent to some polygon $P_i \in \Omega_R$, but is not formed from any of the strips at angle ϕ_j. Since $\vec{v}^h$ is constant by angle with constants summing to zero, then for these two polygons we must have

$$\sum_{\ell=1}^{M-1} \alpha_\ell = \sum_{\ell=1}^{M-1} \alpha_\ell + \alpha_j = 0,$$

149
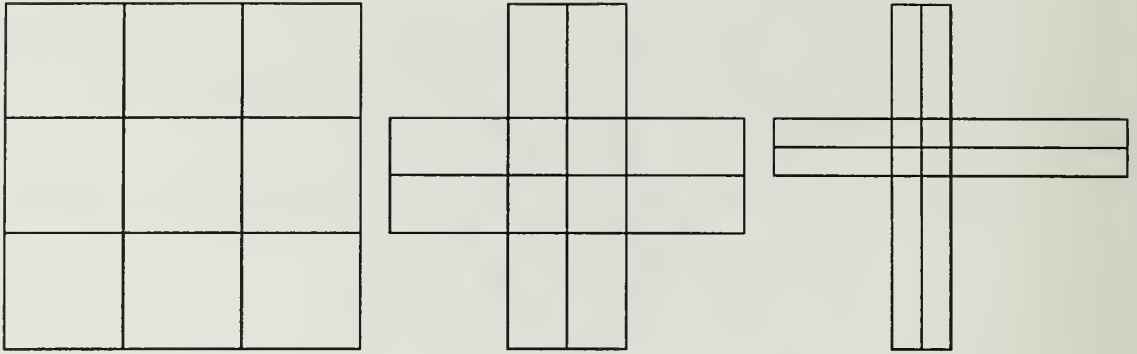
Figure 56. *The strip functions for a two-level refinement.*

which implies that $\alpha_j = 0$. Since ϕ_j was arbitrarily chosen, all the α_j must be zero. Therefore $\vec{v}^h = 0$, which is a contradiction. Hence B^{hh} must be of full rank. ∎

An immediate consequence of this theorem, when combined with (7.3) is

Corollary 7.4: *If $\vec{v}^{2h} \in NS(B^{2h2h})$, then $\vec{v}^{2h} \in NS(B^{h2h})$.*

All of the preceding results are derived from a spotlight CT problem involving one level of refinement. Most will generalize to multiple refinement levels. For example, assume that a portion of the original refinement region is itself refined, producing two levels of refinement. The strip functions for two angles of such a two level refinement are shown in the Figure 56.

Letting the global coarse level be denoted as $4h$, the first refinement level as $2h$, and the finest level as h, then the composite system matrix $B^{\underline{h}}$ will have the form

150

$$B^{\underline{h}} = \begin{pmatrix} B^{4h4h} & B^{4h2h} & B^{4hh} \\ B^{2h4h} & B^{2h2h} & B^{2hh} \\ B^{h4h} & B^{h2h} & B^{hh} \end{pmatrix}.$$

This matrix is also non-negative, symmetric and positive semi-definite. Each of the nine blocks comprising it have themselves a $M \times M$ block structure, where M is the number of angles used in x-raying the image. The same summability properties are present as well. That is, the level h rows of the matrix formed from refining a level $2h$ strip function, when summed, will equal that row corrsponding to the level $2h$ strip function in question. Likewise, level $2h$ rows will sum to equal the level $4h$ rows they are refined from. This allows an extension of Theorem 7.3, which we believe can be proved using identical arguments.

Conjecture: *Let $B^{\underline{h}} \in \mathcal{R}^{N+P+Q}$ be the composite natural pixel discretized matrix formed at M angles, and assume that the first refinement region is covered by a total of m_1 level $4h$ strip functions subdivided into level $2h$ strip functions, and assume the second refinement region is covered by a total of m_2 level $2h$ strip functions subdivided into level h strip functions, as outlined above. Then the rank of $B^{\underline{h}}$ is $N + P + Q - (M + m_1 + m_2 - 1)$.*

This recursive refinement can be expanded to as many levels as required, in the same fashion as the second level was added.

From our analysis, we see that the composite matrix $B^{\underline{h}}$ retains many of the properties of the unrefined matrix. Additionally, we have determined its rank and characterized its null space. In the next section, we propose a multilevel solution

technique for $B^{\underline{h}}\vec{\alpha}^{\underline{h}} = \vec{f}^{\underline{h}}$ that will ultimately be equivalent to the well-known Fast Adaptive Composite (FAC) method.

C.   MULTILEVEL APPROACH

The spotlight CT problem is a composite grid problem, in which an operator equation $Lu = f$ must be solved on some composite grid $\Omega^{\underline{h}}$ comprised of a global coarse grid Ω^{2h} and a local fine grid Ω^h (which itself could be a composite grid, allowing for recursive refinement). Fast Adaptive Composite grid methods (FAC) were developed to utilize multilevel technology to solve such composite problems in an efficient manner [Ref. 34, 3].

FAC methods are characterized by their use of a composite grid, which is the union of regular grids of various sizes. The problem is discretized and solved on the non-uniform composite grid, but all of the actual computations occur on the uniform subgrids. This provides the advantage of using existing uniform grid solvers, while at the same time allowing for effective resoltion of local areas of interest. For this reason, FAC is preferable to just solving the system of composite equations.

There are three main features that allow FAC to handle grid refinement problems successfully. First, the composite grid is the union of a sequence of nested uniform grids, which simplifies the data structure needed to represent it. Second, almost all computation is restricted to these uniform grids. Finally, the use of multilevel processing to correct coarse grid approximations with fine grid residuals through the use of overlapping grids and interpolation at the grid interfaces allows for effective

152

intergrid communication.

As with all multilevel methods, it is necessary for quantities to be accurately represented on the various grids, and intergrid transfer operators must exist to transfer these quantities between grids. FAC is no exception. Letting superscripts denote the grid on which a quantity is defined, the composite grid equation becomes

$$L^{\underline{h}} h^{\underline{h}} = f^{\underline{h}}.$$

Likewise, the equations $L^{2h} u^{2h} = f^{2h}$ and $L^h u^h = f^h$ denote the problem restricted to the global coarse grid and the local fine grid, respectively. Assume that intergrid transfer operators $I_{\underline{h}}^{2h} : \Omega^{\underline{h}} \to \Omega^{2h}$, $I_{\underline{h}}^h : \Omega^{\underline{h}} \to \Omega^h$, $I_{2h}^{\underline{h}} : \Omega^{2h} \to \Omega^{\underline{h}}$, and $I_h^{\underline{h}} : \Omega^h \to \Omega^{\underline{h}}$ exist to transfer quantities between grids. The details involved in deriving these representations and operators may be very cumbersome. In-depth treatments can be found in [Ref. 35, 34, 3]. Once all of these components are in place, FAC is given by the following steps.

- Set $r^{2h} = I_{\underline{h}}^{2h}(f^{\underline{h}} - L^{\underline{h}} u^{\underline{h}})$
- Solve $e^{2h} = (L^{2h})^{-1} r^{2h}$
- Correct $u^{\underline{h}} \leftarrow u^{\underline{h}} + I_{2h}^{\underline{h}} e^{2h}$
- Set $r^h = I_{\underline{h}}^h(f^{\underline{h}} - L^{\underline{h}} u^{\underline{h}})$
- Solve $e^h = (L^h)^{-1} r^h$
- Correct $u^{\underline{h}} \leftarrow u^{\underline{h}} + I_h^{\underline{h}} e^h$

In general, FAC first solves the restriction of the composite residual equation to the global coarse grid, using this solution to correct the composite grid approximation.

The restriction of the composite residual equation to the local fine grid is then solved and the solution is used to correct the composite grid approximation. Although formally the procedure calls for exact solvers on both the coarse and fine grids, in practice iterative methods or other multilevel solvers are used.

We now show that FAC applied to the spotlight CT problem is formally equivalent to a *block Gauss-Seidel* formulation of the same problem. Note that a composite grid element $u^{\underline{h}} \in \mathcal{R}^{N+P}$ given by

$$u^{\underline{h}} = \begin{pmatrix} u^{2h} \\ \\ u^h \end{pmatrix}$$

can be decomposed as

$$u^{\underline{h}} = I^h_{2h} u^{2h} + I^h_h u^h. \tag{7.5}$$

Intergrid transfer operators that satisfy (7.5) are given by

$$I^h_{2h} = \begin{pmatrix} I_N \\ \\ 0 \end{pmatrix} \quad \text{and} \quad I^h_h = \begin{pmatrix} 0 \\ \\ I_P \end{pmatrix}.$$

where I_N and I_P are identity operators of the appropriate sizes. It is significant to note that FAC generally does not have such simple intergrid transfer operators. The simplicity in our case is a direct result of the discretization by natural pixels, and the fact that we require refinement to be an exact partition of coarse rays.

Now, consider the FAC scheme applied to $B^{\underline{h}} u^{\underline{h}} = f^{\underline{h}}$. Initially, we compute the residual of the composite problem restricted to the global coarse grid as

154

$$r^{2h} = I_{\underline{h}}^{2h}(f^{\underline{h}} - B^{\underline{h}}u^{\underline{h}})$$

$$= I_{\underline{h}}^{2h}\left(\begin{pmatrix} f^{2h} \\ f^h \end{pmatrix} - \begin{pmatrix} B^{2h2h} & B^{2hh} \\ B^{h2h} & B^{hh} \end{pmatrix}\begin{pmatrix} u^{2h} \\ u^h \end{pmatrix}\right)$$

$$= f^{2h} - B^{2h2h}u^{2h} - B^{2hh}u^h,$$

which implies that

$$I_{\underline{h}}^{2h} = (I_N \ 0).$$

Next, the coarse grid residual error is computed as $e^{2h} = \left(B^{2h2h}\right)^{-1} r^{2h}$. Note that this is a *formal* treatment, as B^{2h2h} is singular. The current approximation $u^{\underline{h}}$ is then corrected as

$$u^{\underline{h}} \leftarrow u^{\underline{h}} + I_{2h}^{\underline{h}}e^{2h},$$

which after applying (7.5) becomes

$$u^{\underline{h}} \leftarrow I_h^{\underline{h}}u^h + I_{2h}^{\underline{h}}(u^{2h} + e^{2h}). \tag{7.6}$$

We now compute the residual of the composite problem restricted to the local fine grid as

$$r^h = I_{\underline{h}}^h(f^{\underline{h}} - B^{\underline{h}}u^{\underline{h}})$$

$$= I_{\underline{h}}^{2h}\left(\begin{pmatrix} f^{2h} \\ f^h \end{pmatrix} - \begin{pmatrix} B^{2h2h} & B^{2hh} \\ B^{h2h} & B^{hh} \end{pmatrix}\begin{pmatrix} u^{2h} \\ u^h \end{pmatrix}\right)$$

$$= f^h - B^{h2h}u^{2h} - B^{hh}u^h,$$

155

which implies that

$$I_{\underline{h}}^h = (0 \quad I_P).$$

The fine grid residual error is computed as $e^h = \left(B^{hh}\right)^{-1} r^h$, and the current approximation $u^{\underline{h}}$ is again corrected as

$$u^{\underline{h}} \leftarrow u^{\underline{h}} + I_h^{\underline{h}} e^h,$$

which after applying (7.5) and (7.6) becomes

$$u^{\underline{h}} \leftarrow I_h^{\underline{h}}(u^h + e^h) + I_{2h}^{\underline{h}}(u^{2h} + e^{2h}). \tag{7.7}$$

Now, we show that the block Gauss-Seidel formulation of $B^{\underline{h}} u^{\underline{h}} = f^{\underline{h}}$ results in (7.7) as well. The spotlight CT problem can be expressed as

$$\begin{pmatrix} B^{2h2h} & B^{2hh} \\ B^{h2h} & B^{hh} \end{pmatrix} \begin{pmatrix} u^{2h} \\ u^h \end{pmatrix} = \begin{pmatrix} f^{2h} \\ f^h \end{pmatrix}.$$

which when solved for u^{2h} and u^h yields the one sweep block Gauss-Seidel scheme

- Set $u^{2h} \leftarrow (B^{2h2h})^{-1}(f^{2h} - B^{2hh}u^h)$
- Set $u^h \leftarrow (B^{hh})^{-1}(f^h - B^{h2h}u^{2h})$

This scheme can be rearranged to produce the desired results in the following fashion. For the global coarse grid we have

$$u^{2h} \leftarrow (B^{2h2h})^{-1}(f^{2h} - B^{2hh}u^h) \tag{7.8}$$

$$= (B^{2h2h})^{-1}(r^{2h} + B^{2h2h}u^{2h}) \tag{7.9}$$

$$= e^{2h} + u^{2h} \tag{7.10}$$

156

Proceeding in the same manner with the local fine grid yields

$$u^h \quad \leftarrow \quad (B^{hh})^{-1}(f^{2h} - B^{h2h}u^{2h}) \tag{7.11}$$

$$= \quad (B^{hh})^{-1}(r^h + B^{hh}u^h) \tag{7.12}$$

$$= \quad e^h + u^h \tag{7.13}$$

The argument is completed by applying the relationship (7.5) to the expressions in (7.10) and (7.13) above, yielding

$$u^{\underline{h}} \leftarrow I_{2h}^{h}u^{2h} + I_{h}^{\underline{h}}u^h \quad = \quad I_{2h}^{\underline{h}}(e^{2h} + u^{2h}) + I_{h}^{\underline{h}}(e^h + u^h).$$

This shows that FAC is formally equivalent to block Gauss-Seidel on the spotlight CT problem.

The methods are formally equivalent, because we already know that the block matrix B^{2h2h} is singular. In practice, to utilize the block Gauss-Seidel approach we solve each block system in turn with an iterative method, or a multilevel method such as PMLV.

A considerable body of theoretical results exists for two-level FAC [Ref. 3]. This theory requires that the variational properties be satisfied, that quantities be measured using the energy norm, and that the operator $B^{\underline{h}}$ be positive definite. Hence B^{2h2h} and B^{hh} are non-singular. Under these assumptions, there exist convergence factors for FAC, given in terms of the spectral radii of combinations of B^{2h2h} and B^{hh} with the intergrid transfer operators. These convergence factors exist for the case when an exact solver is used, and for the case when relaxtion is used to approximate

the exact solver. The latter case is also a function of the relaxation scheme being used.

The equivalence of FAC and the block Gauss-Seidel scheme would allow this theory to be directly applied, except for the fact that our operator $B^{\underline{h}}$ is not positive definite. Even for such problems, FAC theory may apply in certain circumstances. McCormick [Ref. 3] states that

Theorem 7.5: *If $B^{\underline{h}}$ is positive semidefinite and*
$$NS(B^{\underline{h}}) \subset (I_{2h}^{h} NS(B^{2h2h})) \cap (I_{h}^{h} NS(B^{hh})), \tag{7.14}$$
then existing FAC convergence theory is applicable.

Unfortunately, (7.14) is not satisfied for the spotlight CT problem, as B^{hh} is of full rank and we know that $B^{\underline{h}}$ is rank deficient.

Even without this theory, numerical results are promising. In the reconstructions that follow, a geometry of 32 detectors over 20 angles is used for the global coarse grid. The refinement region is located in the center of the image, as depicted in Figure 57 below.

Sixteen coarse strip functions at each angle are refined by splitting them in half. The composite grid image is then reconstructed by using the spotlight (FAC) method developed above.

The first two examples are reconstructions of the two brain phantoms used in prior numerical experiments. (See Figures 45 and 46). In each case, the global coarse grid representation on the left lacks detail. The geometric objects within the *brain*

Figure 57. *Location of refinement region.*
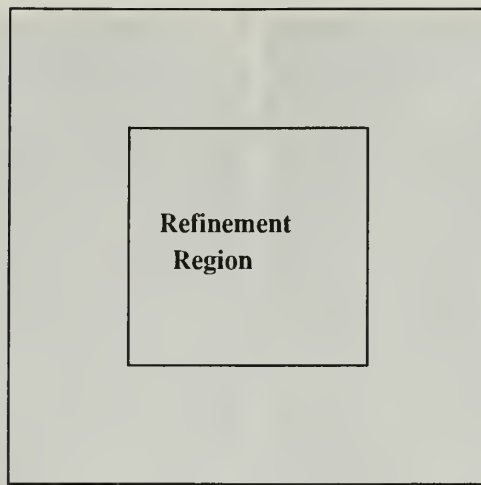
are difficult to identify. The local fine grid representation on the right, at twice the resolution, provides better detail about the center region of each brain.

In the third example, the real value of spotlight CT is demonstrated. A brain phantom containing a small *tumor* is depicted in Figure 60. Note that the tumor is
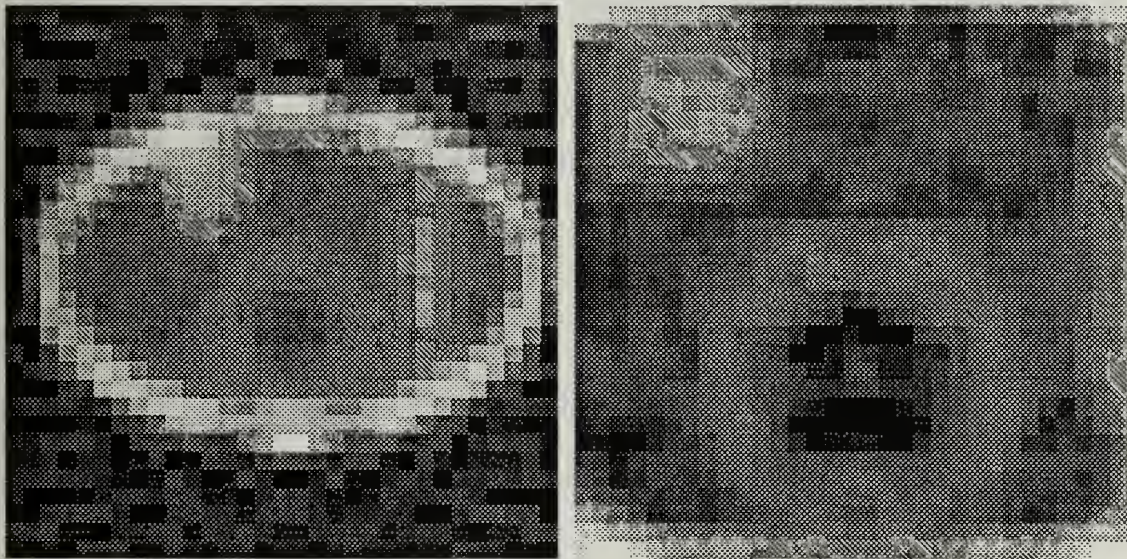
Figure 58. *Global and refined reconstructions - example 1.*
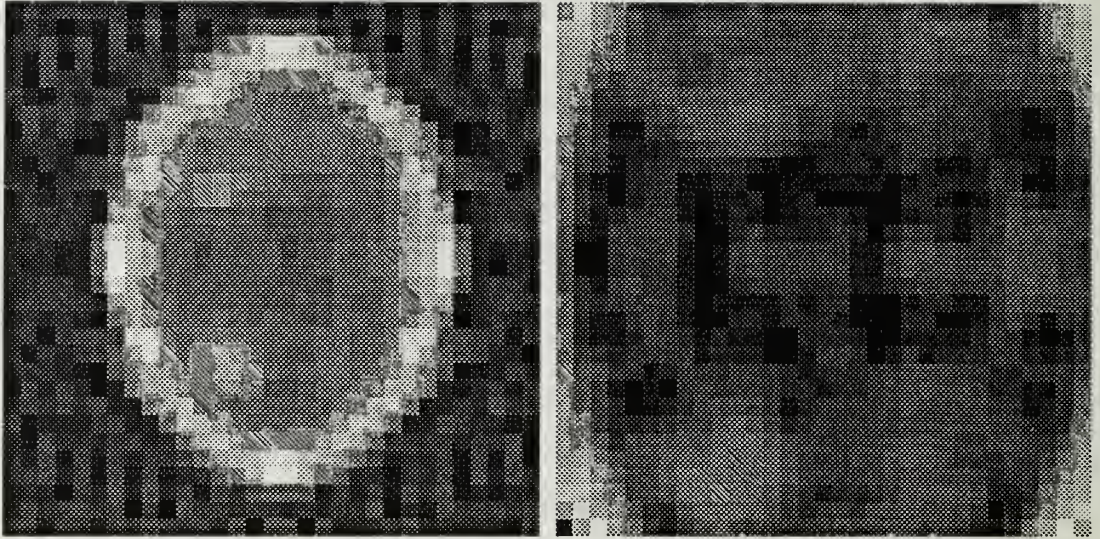
Figure 59. *Global and refined reconstructions - example 2.*

sufficiently small that it is not expected that we see it on the global coarse reconstruction. We reconstruct this image using the spotlight technique, and the reconstruction is shown in Figure 61. In the global coarse reconstruction to the left, the tumor is not apparent. However, the higher resolution spotlight image on the right clearly shows the presence of the tumor.

The natural pixel discretization of the spotlight CT problem allows a high resolution reconstruction of a portion of an image at a lower cost than that required if the entire image is discretized at a fine resolution. When solved in block Gauss-Seidel form, further savings may be realized by using the PMLV routine as a solver.
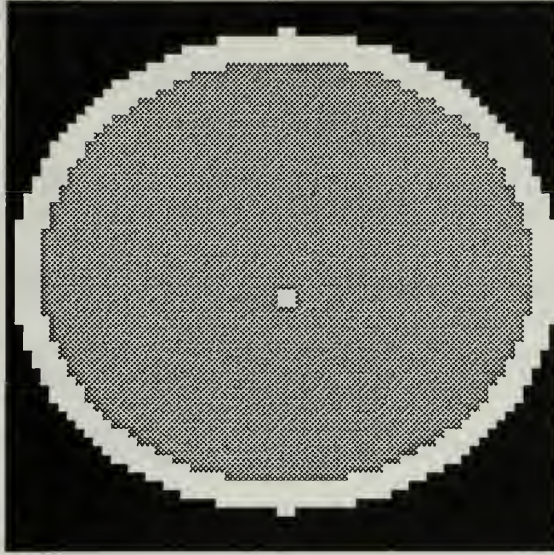
Figure 60. *Actual image - example 3.*
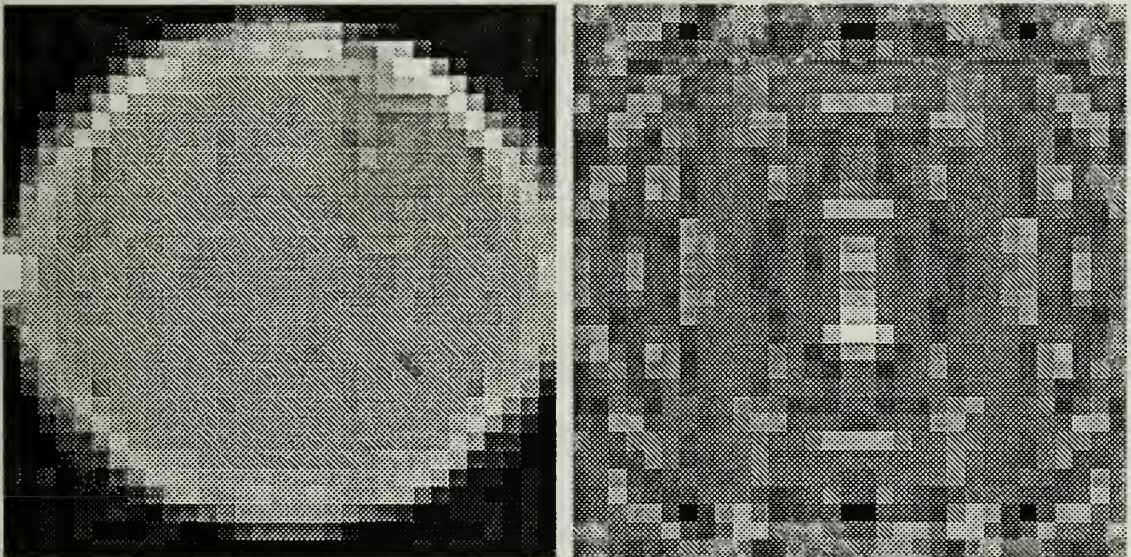
Figure 61. *Global and refined reconstructions - example 3.*

VIII.   CONCLUSIONS

A.   GOALS OF THE RESEARCH

The primary goal of our research was to introduce multilevel technology into the solution of this problem. Intermediate objectives included the development of the mathematics of natural pixels more fully, development of spotlight CT, and *building a better mousetrap* - development of a multilevel algorithm that is competitive with state-of-the-art image reconstruction methods. We achieved all of these goals but the last one, and laid some foundations for future research.

B.   STANDARD ART

We focus our efforts on the algebraic reconstruction technique (ART), one of several solution techniques commonly used to solve this problem. For a preliminary foundation, the standard ART approach is examined, in which the problem is discretized using square pixels and the resulting linear system of equations is solved using the method of Kaczmarz. We learn why convergence of Kaczmarz stalls after several iterations on the large rectangular system produced by this dicretization. Using the Singular Value Decompostion (SVD) of the system matrix, it is determined that the singular value spectrum can be separated into three bands - a resolvable region, a near null space, and a null space. Numerical tests show that solution components in the resolvable region of the spectrum can be recovered during the iteration, while those

in the near null space cannot. It is also learned that Kaczmarz mixes modes during iteration, i.e. singular value modes over the entire spectrum are excited, including those modes in the near null space, which cannot be eliminated later.

C. NATURAL PIXEL DISCRETIZATION

We adopt the natural pixel discretization, in which the image is discretized into strips corresponding to the x-ray paths through it, on each of which the image is assumed constant. The resulting system matrix is symmetric and positive semi-definite, which allows for a wider variety of relaxation methods that can be used to solve the linear system. It is also in general smaller than the matrix produced through the square pixel discretization. We take advantage of the symmetry and reduced matrix size and solve the problem faster, while still reconstructing high-quality images. A detailed linear algebraic examination of the matrix is conducted, producing several useful results. The rank of the matrix is completely determined by the geometry used to x-ray the image, and the null space of the matrix is characterized by vectors with easily recognized properties. We construct a basis for the null space of a general matrix show that images corresponding to such vectors are invisible and thus do not affect the quality of the reconstruction. While the idea of natural pixels did not originate in this work, the analysis of the matrix properties goes well beyond anything previously done, and the spectral analysis is entirely new.

D. GAUSS-SEIDEL ITERATION

The Gauss-Seidel method is considered for solving the linear system. We show that when measured with the energy norm Gauss-Seidel cannot diverge on this problem, and with the correct initial starting vector must converge to the minimum norm solution. We examine this issue further, and learn that the eigenvectors of the Gauss-Seidel iteration matrix associated with eigenvalues of modulus one are in the null space of the system matrix. This fact explains why convergence is ensured for the semidefinite system. We find that Gauss-Seidel applied to this problem stalls after a few iteration sweeps. Using the techniques developed earlier, the performance of the iteration is analyzed. As with the Kaczmarz matrix, the natural pixel system matrix is found to have a spectrum that separates into a resolvable region, near null space and null space. Gauss-Seidel exhibits behavior similar to Kaczmarz in that it cannot resolve components in the near null space, and it mixes modes. Such analysis has not been done before in this setting, and helps us understand why the iteration stalls.

E. MULTILEVEL METHODS

This type of behavior is often exhibited by Gauss-Seidel when applied to systems resulting from the discretization of PDE's as well. It is well-known that multilevel methods improve the performance of relaxation methods such as Gauss-Seidel in the PDE setting [Ref. 28]. We seek a similar increase in performance in the image reconstruction setting.

We formally cast the natural pixel discretized image reconstruction problem in

a multilevel setting using the Multilevel Projection Methodology (PML). The natural pixel discretization is shown to be a discretization by orthogonal projections, which subsequently induces interspace transfer operators, a coarse grid correction scheme and a relaxation scheme. This is the first time PML is formally applied to a problem not arising from partial differential equations, and represents an expansion in the types of problems that can be approached with multilevel techniques. A PML V-cycle algorithm is developed, and its convergence properties established. Numerical results show that PML initially converges faster than Gauss-Seidel alone on the problem, and then it stalls as well. While the *better mousetrap*, a fully competetive multilevel algorithm, did not emerge, nevertheless the PML method can solve the problem cheaper and faster than either Gauss-Seidel alone or the standard Kaczmarz approach, while producing reconstructions of comparable quality. This represents an improvement, although not a revolution, in the state-of-the-art of the algebraic image reconstruction problem.

F. SPOTLIGHT CT

Finally, we consider the Spotlight CT problem, where high resolution is desired for only a portion of the image. We discretize the problem using natural pixels on multiple levels of resolution, which has not previously been attempted. The resulting composite grid avoids the high cost of discretizing the whole problem on a fine level. We hope to solve the resulting composite linear system efficiently. An analysis of the system again reveals a rich collection of properties. As in the one level case, the rank

165

rank of the system matrix is a function of the geometry used to generate it. The null space is once again characterized by vectors with easily recognizable properties, and these vectors represent invisible images. We cast the composite linear system in a block form that can be solved using a block Gauss-Seidel scheme, and this approach is formally shown to be equivalent to the multilevel Fast Adaptive Composite (FAC) method. This formulation allows us to solve the problem on uniform grids using the techniques developed earlier, instead of having to solve the composite system. Numerical results for two levels yield high quality reconstructions.

G. FUTURE RESEARCH

Directions for future research would concentrate in the area of Spotlight CT. Because this area is so new and unexplored, we believe that it is here our results have the most promise of making a positive contribution. Currently, we construct the system matrix in a piecemeal fashion, one block at a time. This process could be automated, and the discretization could be taken to three or more levels. Theoretically, using this approach an image could be resolved to as fine a level as desired.

Concurrent research on the image reconstruction problem paralleling our own work concentrated on a similar discretization, but with the image region defined as that region in the intersection of all the views, not restricted to the square. With this approach, it is claimed that the resulting system matrix can be represented by a small fraction of its elements [Ref. 23]. This discretization could be incorporated into the spotlight CT problem as well.

166

Finally, the open question of finding a competitive algorithm based on multi-level technology remains. This would be a great accomplishment, and warrants continued investigation. Some possible avenues to pursue include coarsening the problem by angles, or a combination of angles and detectors. These are but a few of the areas that appear ideal for additional research.

APPENDIX. DETAILS ON GENERATING THE MATRIX B

The entries of the matrix B are the areas of intersection of the strips defined by the x-rays. There are N such strips, each of which defines a row in the matrix. To determine the N entries for the ith row, we must compute the areas of intersection of the ith strip with all N strips in turn. We assume the image is contained in the unit square, so we are only concerned with the intersections of strips that lie within this square.

The calculations proceed as follows. A coordinate system is imposed on the image square, with the origin at the center of the square. There are two cases to consider - the strips are parallel so that if they intersect the area is just the area of the entire strip, or the strips are not parallel, in which case they intersect in the form of a parallelogram. In the former case, the area of the strip is calculated based on how it intersects the square. There are six ways this can occur, i.e. it contains a corner of the square, it contains two corners, it intersects opposite sides of the square, etc.

If the strips are not parallel, then we compute the coordinates of the vertices of the parallelogram that is formed from their intersection. Next, the number of vertices that lie within the image square is determined. If the number is zero or four, the calculations to find the area are trivial. If not, then we must determine which

.

vertices are in the square and whether or not the intersection contains the corner of the square. The area is then calculated based on these determinations. The number of unique ways the parallelogram can intersect the border of the square are far too numerous to list here. Symmetry is exploited to reduce the number of calculations by half, that is, the area of strip i with strip j is the same as the area of intersection of strip j with strip i, so the calculation need only be performed once.

A brief outline of the algorithm follows:

```
Input: m = # of angles
       n1 = # of detectors per angle

Output: B, an N x N matrix, where N = m*n1

initialize B=0
for i=1:m
  for j=1:n1
    index=(i-1)*n1+j
    for k=1:m
      for l=1:n1
        index1=(k-1)*n1+l
        if index1 >= index          {exploit symmetry}
          if i=k                     {rays are parallel}
            compute area
          else                       {rays intersect}
            compute coordinates of vertices
            determine which vertices are in square
            compute area
          end{else}
          B(index,index1)=B(index1,index)=area
        end{if}
      end{for}
    end{for}
  end{for}
end{for}
```

REFERENCES

[1] Michael H. Buonocore, William R. Brody, and Albert Macovski. A natural pixel decomposition for two-dimensional image reconstruction. *IEEE Transactions on Biomedical Engineering*, 28:69–78, 1981.

[2] Stephen F. McCormick. *Multilevel Projection Methods for Partial Differential Equations*. SIAM, Philadelphia, 1992.

[3] Stephen F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*. SIAM, Philadelphia, 1989.

[4] Gabor T. Herman. *Topics in Applied Physics: Image Reconstruction from Projections*. Springer-Verlag, Berlin, 1979.

[5] Gabor T. Herman. *Image Reconstruction from Projections*. Academic Press, Orlando, 1980.

[6] Alfred K. Louis and Frank Natterer. Mathematical problems of computerized tomography. *Proceedings of the IEEE*, 71:379–389, 1983.

[7] N.T. Gaarder and Gabor T. Herman. Algorithms for reproducing objects from their x-rays. *Computer Graphics and Image Processing*, 1:97–106, 1972.

[8] Gabor T. Herman. Two direct methods for reconstructing pictures from their projections: A comparative study. *Computer Graphics and Image Processing*, 1:123–144, 1972.

[9] Gabor T. Herman and Stuart W. Rowland. Three methods for reconstructing objects from x-rays: A comparative study. *Computer Graphics and Image Processing*, 2:151–178, 1973.

[10] A.M. Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of Applied Physics*, 34:2722–2727, 1963.

[11] E. Segre. *Nuclei and Particles*. Benjamin, Reading, MA, 1977.

[12] Stanley R. Deans. *The Radon Transform and Some of its Applications*. Krieger, Malabar, FL, 1993.

[13] Van E. Henson. Fourier Methods of Image Reconstruction. PhD Thesis, University of Colorado at Denver, 1990.

[14] Frank Natterer. *The Mathematics of Computerized Tomography.* John Wiley and Sons, Stuttgart, 1986.

[15] Henry Stark. *Image Recovery: Theory and Application.* Academic Press, San Diego, 1987.

[16] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging.* IEEE Press, New York, 1988.

[17] Kunio Tanabe. Projection method for solving a singular system of linear equations and its applications. *Numer. Math.*, 17:203–214, 1971.

[18] Philip E. Gill, Walter Murray, and Margaret H. White. *Numerical Linear Algebra and Optimization.* Addison-Wesley, Redwood City, CA, 1991.

[19] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* John Hopkins, Baltimore, 1989.

[20] Chris Rhoden. Personal Communications, 1994.

[21] Ivar Stakgold. *Green's Functions and Boundary Value Problems.* John Wiley and Sons, New York, 1979.

[22] Richard S. Varga. *Matrix Iterative Analysis.* Prentice Hall, Englewood Cliffs, 1962.

[23] Mark A. Limber. Personal Communications, 1994.

[24] John Neter, William Wasserman, and G.A. Whitmore. *Applied Statistics.* Allyn and Bacon, Needham Heights, MA, 1993.

[25] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[26] Youcef Saad. Practical use of some Krylov subspace methods for solving indefinite and nonsymmetric linear systems. *SIAM J. Sci. Stat. Compu.*, 5:203–228, 1984.

[27] Achi Brandt. Multigrid Techniques: 1984 Guide. Lecture notes for the computational fluid dynamics lecture series, von-Karmen Institute, Rhodes-Siant-Genese, Belgium, 1984.

[28] William L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, 1987.

[29] Craig C. Douglas and Jim Douglas Jr. A unified convergence theory for abstract multigrid or multilevel algorithms, serial and parallel. *SIAM J. Numer. Anal.*, 30:136–158, 1993.

[30] Tin-Sun Pan and Andrew E. Yagle. Numerical study of multigrid implementations of some iterative image reconstruction algorithms. *IEEE Transactions on Medical Imaging*, 10:572–588, 1991.

[31] A.K. Louis. Medical imaging: State of the art and future development. *Inverse Problems*, 8:709–738, 1992.

[32] Jeffrey W. Eberhard and Robert J. Plemmons. Image Reconstruction Algorithms for 3D CT. General Electric Corporate Research and Development Proposal, Schenectady, NY, 1993.

[33] Stephen F. McCormick. Personal Communications, 1993.

[34] S.F. McCormick and J. Thomas. The fast adaptive composite grid method (fac) for elliptic boundary value problems. *Math. Comp.*, 46:439–456, 1986.

[35] J. Mandel and S.F. McCormick. *Iterative Solution of Elliptic Equations with Refinement: The Model Multilevel Case, in Domain Decomposition Methods, T.F. Chan et. al., ed.* SIAM, Philadelphia, 1987.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, VA 22304-6145

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Prof. William Gragg, Code MA/Gr   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

4. Prof. Beny Neta, Code MA/Nd   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

5. Prof. Van Henson, Code MA/Hv   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

6. Prof. Yuh-Jeng Lee, Code CS/Le   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

7. Prof. Monique Fargues, Code EC/Fa   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

8. Prof. Richard Franke, Code MA/Fe   1
   Naval Postgraduate School
   Monterey, CA 93943-5002

9. MAJ Bruce Robinson   4
   2801 Dalewood Drive
   Columbus, GA 31907

10. Mark A. Limber   1
    CECM, Department of Mathematics
    Simon Frasier University
    Burnaby, BC V5A 1S6 CANADA

11. Prof. Stephen F. McCormick   1
     Program in Applied Mathematics
     University of Colorado
     Boulder, CO 80303

12. Dr. Richard Lau   1
     Office of Naval Research
     800 N. Quincy Street
     Arlington, VA 22217