ALGORITHMS FOR RECONSTRUCTION
OF PARTIALLY KNOWN, BAND LIMITED
FOURIER TRANSFORM PAIRS FROM
NOISY DATA

RGB Associates, Inc.

Sponsored by
Defense Advanced Research Projects Agency (DOD)
ARPA Order No. 3655

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441
This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-84-72 has been reviewed and is approved for publication.

APPROVED: 
RICHARD J. MICHALAK
Project Engineer

APPROVED: 
LAWRENCE J. HILLEBRAND, Colonel, USAF
Chief, Surveillance Division

FOR THE COMMANDER: 
JOHN A. RITZ
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCSE) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.
ALGORITHMS FOR RECONSTRUCTION OF PARTIALLY KNOWN, BAND LIMITED FOURIER TRANSFORM PAIRS FROM NOISY DATA

Richard Barakat

Contractor: RGB Associates, Inc.
Contract Number: F30602-82-C-0146
Effective Date of Contract: 4 August 1982
Contract Expiration Date: 30 April 1983
Short Title of Work: Phase Deconvolution Studies
Program Code Number: 3E20
Period of Work Covered: Aug 82 - 30 Apr 83

Principal Investigator: Richard Barakat
(617) 358-4675

Project Engineer: Dr. Richard J. Michalak
(315) 330-3143

Approved for public release; distribution unlimited

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Dr. R.J. Michalak (OCSE), Griffiss AFB NY 13441 under Contract F30602-82-C-0146
This report summarizes detailed mathematical investigations, both analytical and numerical, on recovery of partially known Fourier transforms with emphasis in the context of optical diffraction theory.
Block 11 (Continued)

PAIRS FROM NOISY DATA
GENERAL INTRODUCTION

A recurring problem in many fields, especially diffraction optics, is the reconstruction of a Fourier transform pair \( g, G \) from partial data on either or both functions. Considerable effort has been expended in the development of algorithms for its solution; although there have been some successes, the problem has generally proved to be difficult. Of particular importance to the RADC effort is the retrieval of wavefront aberrations from the measured point spread function of an optical system.

Discussions with several investigators who have employed their own algorithms to these problems have indicated a sort of hit-or-miss attitude with respect to their behavior in various situations. Sometimes the particular algorithm works and sometimes it fails when the data are noisy. With the possible exception of Youla's recent study, there are really no serious attempts to understand the stability, rate of convergence, etc. with respect to noise in the measurements.

The present contract effort was devoted to the development and mathematical understanding of new algorithms based upon numerical functional analysis which are robust with respect to noisy data. The basic material is contained in the two sections entitled:

1. Algorithms for reconstruction of partially known, bandlimited Fourier transform pairs from noisy data: 1, the prototypical linear problem.
II. Algorithms for reconstruction of partially known, bandlimited Fourier transform pairs from noisy data: II, the nonlinear problem of phase retrieval.

Both sections are very mathematical and employ mathematics not commonly encountered by optical physicist and engineers. For this reason a summarizing section has been included; it is the first section in the report and is entitled

Algorithms for reconstruction of partially known, bandlimited Fourier transform pairs from noisy data.
ALGORITHMS FOR RECONSTRUCTION OF PARTIALLY KNOWN, BANDLIMITED FOURIER TRANSFORM PAIRS FROM NOISY DATA
ABSTRACT

This paper is a summary of more detailed mathematical work by the author on recovery of partially known Fourier transforms. These problems of inversion of the finite Fourier transform and of phase retrieval are known to be ill-posed. We draw a distinction in the resultant ill-conditioning of the problems between global ill-conditioning (due to the existence of multiple exact solutions) and local ill-conditioning (due to the existence of large neighborhoods of the true solution, all of whose members are indistinguishable from the true solution if the data is noisy). We then develop extensions of known algorithms that attempt to reduce at least the effects of local ill-conditioning on numerical solutions by using the idea of filtered singular value decomposition, and present some numerical examples of the use of those algorithms. (The author-supplied keywords include: ...)}
1. INTRODUCTION

A recurring problem in many fields (especially diffraction optics, electron microscopy, and X-ray diffraction) is the reconstruction of a Fourier transform pair \( g, G \) from partial data on either or both functions. Considerable effort has been expended in the development of algorithms for its solution; although there have been some successes, the problem has generally proved to be difficult.

The canonical examples for such reconstructions are:

**Example 1, Extrapolation of Band-Limited Signals.** Given a noisy measurement \( \tilde{g} \) of \( g \) on an interval \( A = [a_1, a_2] \) and the knowledge that \( G \) vanishes outside the bounded interval \( B = [b_1, b_2] \), reconstruct \( g \) and \( G \) on the entire real line.

Example 1 is the archetypical linear problem in transform reconstruction. A number of algorithms have been proposed for its solution, either by iterative means: Gerschberg and Saxton [1], Papoulis [2], Youla [3] or by direct means: Cadzow [4], Sabri and Steenaart [5].

**Example 2, The Phase Problem.** Given a noisy measurement \( m \) of \( |g| \) on an interval \( A \) and the knowledge that \( G \) vanishes outside the bounded interval \( B \), reconstruct \( g \) and \( G \) on the entire real line.

Example 2 has been of theoretical interest for some time; see for example Burge, Fiddy, Greenway, and Ross [6]; however in this most general form it has proved intractable. Numerical solutions obtained in particular cases have done so by (sensibly) incorporating further knowledge of \( g \) and \( G \). Two
such examples are:

*Example 2a, The Two Moduli Problem.* Given noisy measurements $m$ and $n$ of $|g|$ on $A$ and $|G|$ on $B$, and the knowledge that $G$ vanishes identically outside of $B$, reconstruct $g,G$ over the entire real line.

*Example 2b, The Phase Problem with Nonnegativity Constraints.* Given a noisy measurement $m$ of $|g|$ on $A$ and the knowledge that $G$ is nonnegative over $B$ and vanishes identically outside $B$ reconstruct $g,G$ over the real line.

Gerschberg and Saxton [1] developed a widely used algorithm for Example 2a; almost all the iterative algorithms for the solution of the general reconstruction problem are suitably modified versions of this particular case. The Gerschberg-Saxton algorithm (hereafter denoted as the GS algorithm) in its general form is essentially the steepest descent algorithm with unit step length and so is first order [7]. In [8], the GS algorithm is successfully applied to a problem of the type in Example 2b. An alternative second order algorithm, based on Newton's method, has been proposed by Barakat and Newsam [9].

The canonical examples presented are simple in form, nevertheless they contain the salient features that make other, more complicated, problems intractable. The font of all difficulties is that the reconstruction problem is ill-posed, and badly ill-posed at that. The original definition of a well posed problem is due to Hadamard [10].
**Definition.** A problem is well posed if the solution

1. exists
2. is unique
3. depends continuously on the data.

If a problem violates any of these conditions, it is ill-posed. Each of the model problems violates at least one of these criteria. The literature, e.g. [11-13] has focussed on violations of uniqueness, however it is our contention that violation of condition 3 is the cause of the most of the problems encountered in numerical solutions; in particular it accounts for the extreme sensitivity of such solutions to small perturbations in the data.

The purpose of this paper is to summarize for the optical community the detailed analysis of transform reconstructions developed in [14-16]. These three references contain a detailed mathematical treatment of ill-conditioning in transform recovery problems with emphasis on the implications for numerical algorithms. The theory is dimension independent, although the supporting numerical calculations are restricted to one-dimensional problems. We hope to present two-dimensional calculations in the near future.

In order to present the results the following notation will be used. If $D \subseteq \mathbb{R}^N$ then $L^2(D)$ denotes the space of square integrable, complex valued functions over $D$ with norm $\| \cdot \|$ and inner product $(\cdot, \cdot)$. If $T$ is any set in $L^2(\mathbb{R}^N)$ then the projection $P_T$ onto $T$ is defined by

$$y = P_T x \iff y \in T \quad \text{and} \quad \| y - x \| = \inf_{z \in T} \| z - x \| \quad (1.1)$$
In the special case when $T \in L^2(D)$; $P$ will be abbreviated to $P_D$, $P_D$ has the form

$$G(\hat{\omega}) \quad \hat{\omega} \in D$$

otherwise

$$= 0$$

(1.2)

The Fourier transform $\mathcal{F} : L^2(\mathbb{R}^N) \to L^2(\mathbb{R}^N)$ is defined to be

$$g(\hat{\nu}) = (\mathcal{F} G)(\hat{\omega}) = \int_{-\infty}^{\infty} e^{2\pi i \hat{\nu} \cdot \hat{\omega}} G(\hat{\omega}) \, d\hat{\omega}.$$  

(1.3)

If $D$ and $E$ are bounded subsets of $\mathbb{R}^N$ then the operator $P_D \mathcal{F} P_E$ will be called a finite Fourier transform (fFT). Finally the interval $[-c,c]$ shall be denoted by $\mathcal{C}$; the projection $P_{\mathcal{C}}$ shall be abbreviated to $P_{\mathcal{C}}$, and the fFT $P_{\mathcal{C}} \mathcal{F} P_{\mathcal{C}}$ is represented by $\mathcal{F}_{\mathcal{C}}$.

The next section of the paper presents a survey of known results on example 1, which may be shown to be equivalent to inversion of the operator $\mathcal{F}_{\mathcal{C}}$ where $c = \frac{1}{2} \left( (a_1 - a_2) (b_1 - b_2) \right)^{1/2}$. Of particular importance is the relation between the ill-conditioning of problem 1 and the singular value decomposition (SVD) of $\mathcal{F}_{\mathcal{C}}$; and how the two ideas are combined in inversion by filtered SVD. In Section 3 local ill-conditioning of the nonlinear phase retrieval problems is described in terms of that of the fFT; and it is contrasted with the global ill-conditioning due to the possibility of multiple exact solutions. Section 4 presents a simple generalization of the GS algorithm, tailored to overcome some of this ill-conditioning; together with a review of its convergence properties. In Section 5 extensions of this
algorithm are discussed, Section 6 contains a brief survey of numerical results (in the context of optical diffraction theory) on the relative behavior of these algorithms, followed by some concluding remarks.
2. THE LINEAR PROBLEM

The extrapolation problem of example 1 requires for its solution inversion of the linear integral equation

\[ P_A \mathcal{F}_P G = P_A g \quad (2.1) \]

Since \( A \) and \( B \) are bounded sets \( P_A \mathcal{F}_P \) is a compact linear integral operator and Eq. (2.1) is a Fredholm integral equation of the first kind. Inversion of such an equation is the prototypical linear ill-posed problem.

The ill-posed nature of the problem is exposed in the construction of the solution using the singular value decomposition (SVD) of \( P_A \mathcal{F}_P \). As outlined in Baker [17], a compact linear operator has an SVD \( \{ \phi_i, \sigma_i, \psi_i \}_{i=1}^{\infty} \) consisting of functions \( \phi_i \) and \( \psi_i \) and nonnegative real numbers \( \sigma_i \) with the properties that

i. \( \{ \phi_i \}_{i=1}^{\infty} \) and \( \{ \psi_i \}_{i=1}^{\infty} \) are complete sets of orthonormal functions for \( L^2(A) \) and \( L^2(B) \), respectively.

ii. \( \sigma_i > \sigma_{i+1} \) and \( \lim_{i \to \infty} \sigma_i = 0 \).

iii. \( P_A \mathcal{F}_P \psi_i = \sigma_i \phi_i \) \quad (2.2)

Expansion of \( G \) and \( P_A g \) as sums of singular functions

\[ G = \sum_i b_i \psi_i \quad \quad P_A g = \sum_i \sigma_i a_i \phi_i \quad (2.3) \]

gives a formal solution to Eq. (2.1) by equating coefficients, i.e.,

\[ P_A \mathcal{F}_P G = P_A g \iff \sigma_i b_i = a_i \quad (2.4) \]
This solution illustrates the ill-conditioning in the problem. For instance if \( \tilde{g} \) is a perturbation of \( g \) such that

\[
P_A \tilde{g} = \sum \tilde{a}_i \phi_i \quad \text{with} \quad \sum |\tilde{a}_i - a_i|^2 < \epsilon^2
\]

then it is possible that the error occurs in a high frequency (large \( i \)) component of \( P_A g \), so that a perturbation of size \( \epsilon \) is induced in \( G \). For fixed \( \epsilon \) this perturbation grows arbitrarily large as \( i \to \infty \). Thus Eq. (2.1) fails condition 3 of Hadamard's definition with respect to data perturbation in \( g \). A similar argument, but one that is rarely made, shows that the equation is also ill-posed with respect to changes in the model, i.e. perturbations of the operator \( P_A P_B \).

The above arguments show that the whole solution \( G \) cannot be recovered to within any specified accuracy given uncertainty in the data. Therefore the natural question to ask next is: "How large a part of the solution can be recovered to within a desired accuracy in the presence of noise?" A partial answer lies in the idea of the essential dimension \( N(\delta, \epsilon_1, \epsilon_2) \) of the problem; loosely speaking this is the maximum number of parameters in a description of the solution that can be determined to within an accuracy \( \delta \), given errors \( \epsilon_1 \) in \( P_A g \) and \( \epsilon_2 \) in \( P_A P_B \). (The latter error includes discretization and roundoff errors as well as those arising from an imperfect mathematical model of the real world.)

A more precise definition of the essential dimension as the maximum dimension of any subspace \( U \) for which the associated projection \( P_U G \) of the solution can be accurately calculated [16,18] shows that it may be
expressed in terms of the singular values. The optimum subspace is the span
of the first \( N(\varepsilon_1, \varepsilon_2) \) singular functions. This analysis suggests that
Eq. (2.1) be solved by filtered SVD. This algorithm was introduced by Hanson
\cite{19} in which \( G \) is expressed as the sum

\[
G = \sum_i p(a_i, \sigma_i, \gamma) \psi_i
\]  

(2.6)

where \( p \) is a filter function with the general form

\[
\mathcal{P}(a, \sigma, \gamma) \sim \sigma^{-1} \quad \text{for large } \sigma
\]

\[
\sim 0 \quad \text{for small } \sigma
\]  

(2.7)

and \( \gamma \) is a parameter that incorporates knowledge of data errors and the
desired solution accuracy.

Filters come in many forms. For instance, if the error in \( g \) is such
that \( \| P_A g - P_A \tilde{g} \| < \varepsilon_1 \), the model error \( \varepsilon_2 \) is negligible and the projection
\( P_U G \) is to be determined to within an accuracy \( \delta \) (i.e. \( \| P_U G - P_U \tilde{G} \| < \delta \)); then
the filter associated with the essential dimension is

\[
\mathcal{P}(a, \sigma, \gamma) = a \sigma^{-1} \quad \text{if } \sigma > \gamma
\]

\[
= 0 \quad \sigma < \gamma
\]  

(2.8)

where \( \gamma = \varepsilon_1 \sigma^{-1} \). This cutoff filter is a special case \((q+\infty)\) of the class
of filters

\[
p(a, \sigma, \gamma) = \frac{\sigma^q}{\sigma^{q+1} + \gamma^{q+1}}
\]  

(2.9)

The utility of the theory of the essential dimension lies in its ability
to predict the size and general form of components of the solution that may be
accurately recovered in the presence of noise, before numerical calculations
are undertaken. Such predictions are therefore most useful in deciding on
the size and form of an appropriate discretization for use in numerical
solutions. The theory is easily applied to transform recovery problems due to
the large body of knowledge about the SVD of the finite Fourier transform
(fFT) developed by Landau, Pollak, Slepian and Walom [20-26]. The next
theorem gives a brief summary of those results that are most useful in the
present problem.

**Theorem 1.** i. If $D$ and $E$ are bounded subsets of $\mathbb{R}^N$ with volumes
$|D|$ and $|E|$, and surface areas $|\partial D|$ and $|\partial E|$, then, for large $c$, the
number $n(c, \alpha)$ of singular values of $P_D \mathcal{F} P_E$ greater than $\alpha$ is given
approximately by

$$
n(c, \alpha) \sim |D| \cdot |E| c^{2N-2} \cdot \gamma |\partial D| |\partial E| c^{2N-2} \cdot \log(\alpha^{-1} - 1) \log c + o(c^{2N-3})$$

(2.10)

where $cD$ is the set $\{cd : d \in D\}$ and $\gamma$ is a constant independent of $c, D$
and $E$.

ii. In one dimension let $\sigma_n$ be the $n$-th singular value of $\mathcal{F}$. Then
if $b$ is fixed, $c$ arbitrary and $n$ is determined by

$$
n = \left[4c^2 + \frac{2b^2}{\pi} \log(2c \sqrt{2\pi})\right]$$

(2.11)

where $[\alpha]$ denotes the nearest integer to $\alpha$, then

$$\lim_{n \to \infty} \sigma_n = (1 + e^{b})^{-1/2}$$

(2.12)
iii. The singular functions of $\mathcal{F}_c$ are the eigenfunctions of the
Sturm-Liouville equation

$$(1-t^2)\phi'' + (\lambda - 4\pi^2 c^2 t^2)\phi = 0$$

(2.13)

where solutions are required to be uniformly bounded over the entire real line.

The theorem indicates that the singular values $\sigma_n$ of $P_D\mathcal{F}_E$ have a
steplike distribution; $\sigma_n \sim 1$ for small $n$, $\sigma_n$ decays exponentially for large
$n$, and the change from $\sigma_n \sim 1$ to exponential decay occurs over an interval of
width proportional to $c^{2N-2}\log c$ centered on $c^2|D|\cdot|E|$. This in turn
implies that the essential dimension is approximately $c^2|D|\cdot|E|$ and is
almost independent of noise: The exponential decay of $\sigma_n$ implies that the
essential dimension $N = N(\delta, \varepsilon_1, \varepsilon_2)$ is determined by an equation of the form
$e^{-N} \sim \varepsilon_1 \delta^{-1}$ so that the noise level $\varepsilon_1$ must be reduced by a multiplicative
factor to give an additive increase in $N$.

A second consequence is that the low order singular functions are solu-
tions to a Sturm-Liouville problem and therefore are analytic and slowly
varying. Thus they will be well approximated by a discretization based on
smooth functions. This was experimentally verified in [14], where a number
of different discretizations of $\mathcal{F}_c$ were calculated for varying values of $c$.
The results indicated that discretizations based on Gaussian quadrature or
Galerkin approximations using Legendre polynomials required only $N + O(\log N)$
parameters to accurately approximate the first $N$ singular functions of $\mathcal{F}_c$.
In contrast Galerkin approximations based on piecewise constant or trigono-
metric functions appeared to require at least $aN$ parameters to achieve the
same accuracy, where \( a \geq 3 \). The difference may be explained by noting that the expansion of an analytic function on \([-c,c]\) in terms of Legendre polynomials will have rapidly decaying coefficients; whereas an expansion in terms of piecewise continuous functions will converge only slowly. Moreover although trigonometric functions are themselves smooth, they do not approximate the original function but rather a periodic extension of it outside of \([-c,c]\). This extension is likely to have discontinuities at the endpoints \( \pm c \), so that its Fourier series is slowly convergent. This was observed in approximations of the singular functions of \( \mathcal{F}_c \) where they had the worst performance of the discretizations examined; therefore their use is not recommended in transform recovery.

To conclude the section an example of solution of problem 1 by filtered SVD is presented, see [14] for the details and other examples. The equation

\[
\mathcal{F}_c G = (P_c g)(v) + \varepsilon \mu(v)
\]  

(2.14)

was solved numerically, where

\[
g(v) = 2 \left( \frac{\sin \pi v}{\pi v} \right)^2 .
\]  

(2.15)

This is the point spread function of an infocus, aberration free slit aperture. The presence of noise was simulated by the term \( \varepsilon \mu(v) \) with \( \mu(v) \) a random variable uniformly distributed over \([-1,1]\) and \( \varepsilon \) a control of the magnitude of the noise. The parameter values \( c = 1 \) and \( \varepsilon = .03 \) were chosen, a Galerkin approximation based on 80 piecewise constant functions was used to discretize the problem, and the resulting finite system was solved by filtered SVD using the filter of Eq. (2.9) with \( q = 2 \) and \( \gamma = .01 \).
Figure 1 shows two approximate solutions $\tilde{G}$ calculated from noise data along with the true solution for noiseless data

$$G(\omega) = 2(1 - |\omega|)$$

(2.16)

Figure 2 shows the extrapolation $\mathcal{F}\tilde{G}$ of one particular perturbation $P_\epsilon \tilde{G}$. Because of the symmetry of the test functions each graph is for negative values of the argument only.

The graphs show that $\tilde{G}$ is a good approximation to the true solution $G$, except at the origin where the smooth singular functions cannot reconstruct the discontinuity in slope. Since this discontinuity dominates the far field behavior of $\mathcal{F}G$, the extrapolation is not as accurate as the reconstruction.
3. SOURCES OF ILL-CONDITIONING IN PHASE RETRIEVAL

The previous discussion of ill-conditioning in the linear problem gives new insight into why the nonlinear problem of phase retrieval is ill-posed. Previous examinations of the problem have concentrated on showing that the problem is ill-posed due to violations of conditions 1 and 2 of Hadamard's. That it is also ill-posed due to violations of condition 3, and the implications such violations have for numerical solutions, has not been noted previously to [15]. We therefore present a brief summary of all three possible sources of ill-conditioning and their effects on the behavior of algorithms for numerical solution of such problems.

Violations of condition 1 are not in themselves important for the following reason. Phase retrieval is a model of a real world phenomenon known to exist. Therefore failure of the model to have a solution does not imply that the real physical quantity does not exist; so nonexistence must be due either to an inaccurate model or to noisy data. Both of these possibilities are simply extreme examples of discontinuous dependence of the solution on the data; therefore violations of condition 1 are subsumed under violations of condition 3.

Nonuniqueness, however, is an important source of ill-conditioning. The precise form of all possible solutions to the one-dimensional (1-D) phase problem appears to have been first determined by Akutowicz [27,28] and has been independently rediscovered by a number of other authors, e.g. [29,30]. Their results imply that the phase problem in 1-D has uncountably many
solutions. In [31], see also [15], we show that these results may be extended
to two (and higher) dimensions to give necessary conditions on the form of
multiple solutions for higher dimensions; these conditions imply that multiple
solutions are significantly less likely than in 1-D.

The 1-D results depend on noting that any solution \( g(v) \) is an analytic
function of exponential growth: this follows from the fact that \( g \) is the
transform of a function \( G \) with bounded support and the Paley-Wiener theorem.
Therefore \( g(v) \) has a Hadamard factorization [32] of the form

\[
g(v) = |g(0)| e^{i(\alpha + 2\pi)} \prod_{k=1}^{\infty} \left( 1 - \frac{v}{v_k} \right)
\]  

(3.1)

where \( \alpha \) and \( \beta \) are real constants and \( \{v_k\}_{k=1}^{\infty} \) are the countably many
zeroes of \( g \). Then any other solution must be of the form

\[
\tilde{g}(v) = e^{i(\tilde{\alpha} + 2\pi)} B(v)g(v) \quad \tilde{\alpha}, \tilde{\beta} \in \mathbb{R}
\]  

(3.2)

where \( B(v) \) is a finite or infinite product of Blaschke factors, i.e.

\[
B(v) = \prod_{k=1}^{\infty} B_k(v) \quad \text{where} \quad B_k(v) = \frac{v - v^*}{v - v_k}.
\]  

(3.3)

Furthermore if \( \tilde{\beta} = 0 \) then any \( \tilde{g} \) given by Eq. (3.2) is indeed a solution.
Thus alternative exact solutions are basically generated by "flipping" zeroes
of \( g(v) \) to their complex conjugates. Since there are an infinite number of
zeroes there are also an infinite number of exact solutions to a 1-D phase
retrieval problem.
If the exact solutions were few and well separated, and the phase retrieval problem well-posed in a neighborhood of each zero, then any standard numerical algorithm would perform satisfactorily. However, as noted by Napier [33], any $N$ zeroes may be flipped or not flipped in $2^N$ different combinations, giving $2^N$ different solutions. Therefore there is a very large number of possible solutions, in fact an uncountable infinity of such solutions. Furthermore, for reasonable functions $G(\omega)$, any infinite product $B(z)$ of Blaschke factors will converge, i.e. $B(z) = \lim_{N \to \infty} B_N(z)$ where $B_N(z)$ is a product of $N$ Blaschke factors. Since each finite product corresponds to a possible solution, it follows that the set of solutions has limit points. Any numerical algorithm will have great difficulty in the neighborhood of such points.

However the situation improves markedly in $N \geq 2$ dimensions. An extension of the arguments for $N = 1$ shows that the zeroes of any alternative solution $\tilde{g}$ are the zeroes, or complex conjugates of the zeroes, of $g$. But, as $g$ is an entire function of $N$ complex variables, its zeroes form an analytic set $X$ of dimension $N - 1$ [34]. This set $X$ is essentially the union of $M$ connected, $N-1$ dimensional, analytic manifolds $X_i$. Therefore, if part of a manifold $X_i$ is flipped to form the zeroes of $\tilde{g}$, all of $X_i$ must be flipped to ensure that the zeroes of $\tilde{g}$ also form an analytic set $\tilde{X}$; so $\tilde{X}$ will be of the form $\bigcup_{k=1}^{P} \bigcup_{i_k=1}^{M-P} \bigcup_{\ell=1}^{1} X_i$. But in two or more dimensions the set $X$ is likely to be irreducible, i.e. $M = 1$, in the same way that almost all polynomials of two or more variables are irreducible. Therefore at most two possible solutions with zeroes $X$ and $X^*$ can be
formed, and the solution is essentially unique.

Therefore, in the simplest model problem, ill-conditioning due to non-uniqueness is likely to be a severe problem in one-dimensional problems, or in problems that are essentially one-dimensional (e.g. those with radial symmetry considered in [35]); but in higher dimensions it should have significantly less effect. The situation is less well understood if side conditions are imposed. For instance if both $g$ and $G$ are analytic then the solution of example 2a is essentially unique, but for arbitrary $G$, nonunique solutions have been constructed (see [11] for a review). Even less is known about the effect of positivity.

However, it should be noted that for all three problems there can be parasitic solutions due to symmetries, etc. For example, if $G$ vanishes outside of the interval $dI$ where $d < c$ then translations $\tilde{G}$ of $G$ will still be within $cI$ and will have transforms $\tilde{g}$ which have the same modulus as $g$ over $cI$.

Finally we show that example 2 is locally ill-conditioned in that it violates condition 3. The phase retrieval problem may be recast as requiring the solution of the nonlinear equation

$$\mathcal{D}(G) = m \quad \mathcal{D}: L^2(B) \rightarrow L^2(A)$$

where $\mathcal{D}$ is the composite operator $L^1 P_A \mathcal{F}_B$ and

$$\mathcal{D}_1(g) = |g| \quad \mathcal{D}_1: L^2(A) \rightarrow L^2(A)$$

Since $\mathcal{D}_1$ is a bounded continuous operator and $P_A \mathcal{F}_B$ is compact, $\mathcal{D}$ is
compact. This implies that for any \( \varepsilon > 0 \) an infinite sequence of functions \( \{G_i\} \) can be found, such that

\[
\|G_i - G_j\| > 1 - \delta_i; \quad \text{but} \quad \|\mathcal{L}(G_i) - \mathcal{L}(G_j)\| < \varepsilon .
\]  

(3.6)

Nonuniqueness led to global ill-conditioning, in that there are regions in which many exact solutions exist. Compactness leads to local ill-conditioning, in that in the neighborhood of an exact solution there are directions \( H \) in which a change in the solution \( G \) induces a negligibly small change in the observation \( m \). Therefore although \( \mathcal{L}(G) \) and \( \mathcal{L}(G + H) \) are distinct in theory in practice, with the presence of measurement noise, they are indistinguishable.

This local ill-conditioning may be partially quantified by noting that compactness is due to the operator \( PA \mathcal{F} P_B \) and that inversion of \( \mathcal{L} \) involves inversion of \( PA \mathcal{F} P_B \). Therefore the theory of the essential dimension and the idea of filtered inversion outlined in the previous section suggest that the essential dimension of the phase problem (the number of complex parameters that may be accurately determined) is approximately bounded above by

\[ N = (a_2 - a_1) \cdot (b_1 - b_2), \]

is relatively independent of the noise level in \( m \), and that the solution space should be restricted by filtering to the span of the first \( N \) singular functions of \( PA \mathcal{F} P_B \). Furthermore the discretization chosen for the problem should accurately approximate these singular values.

The local ill-conditioning of examples 2a and 2b is less well understood. However it is reasonable to suppose that the total amount of information available in all the constraints is less than the maximum amount of
information in each constraint considered separately. Therefore, if $G$ is represented by $2P$ real parameters, in example 2a $P$ of these are determined by knowledge of $|G|$, and up to $2N$ by knowledge of $|g|$, giving an upper bound on the essential dimension of $P + 2N$. In example 2b the upper bound is $P + N$; in this case the condition that $G$ be real places symmetry constraints on $g$, effectively halving the amount of information available in $m$. Again these results suggest that numerical solutions be constructed in a similar fashion as solutions to the linear problem.
4. ITERATED PROJECTION ALGORITHMS

The previous sections showed that transform recovery problems are locally ill-conditioned, which confirms the practical experience of a number of authors [1,8,36] who noted very slow convergence rates of the Gerschberg-Saxton (GS) algorithm. Therefore, in order to modify the algorithm to cope with this ill-conditioning, we place GS in a more general setting by viewing it as a special case of finding a common intersection point of a collection of sets. In formal language, it is:

Given the sets \( \{T_i\}_{i=1}^{M} \) with associated projections \( P_i = P_{T_i} \), find \( G \) such that \( G \in \bigcap_{i=1}^{M} T_i \).

Gubin, Polyak and Raik [37] have proposed an iterative algorithm for the solution of such problems in which at the \( n \)-th stage \( G_{n+1} \) is generated by

\[
G_{n+1} = P_i G_n \quad \text{where} \quad i = (n-1) \mod M + 1 \quad (4.1)
\]

and proved that under certain conditions, such as the convexity of the sets \( T_i \), the iterates converged to a common intersection point if one existed. A survey of this and similar algorithms appearing in the Russian literature is given by Censor and Herman in [38]. If the set \( S \) is defined to be

\[
S = \{ g \in L^2(\mathbb{R}) : g(v) = \tilde{g}(v) \ \text{for} \ v \in A \} \quad (4.2)
\]

for example 1, or

\[
S = \{ g \in L^2(\mathbb{R}) : |g(v)| = m(v) \ \text{for} \ v \in A \} \quad (4.3)
\]

for example 2, and the sets \( T_1 \) and \( T_M \) defined as
\[ T_1 = \{ g \in L^2(\mathbb{R}) : G = \mathcal{F}^{-1}g, \ g \in S \} \] (4.4)

\[ T_M = \{ g \in L^2(\mathbb{R}) : G(\omega) = 0 \text{ for } \omega \notin B \} \] (4.5)

then, as the Fourier transform preserves \( L^2 \) norms

\[ h = P_S g \iff H = \mathcal{F}^{-1}h = P_1(\mathcal{F}^{-1}g) = P_1G \] (4.6)

and GS applied to examples 1 and 2 is recognizable as the iterated projection algorithm of Eq. (4.1) with \( M = 2 \).

The advantages of the iterated projection algorithm are: First that it allows easy incorporation of extra constraints such as those in examples 2a and 2b by setting \( M = 3 \) and adding either the set

\[ T_2 = \{ g \in L^2(\mathbb{R}) : G(\omega) \geq 0 \text{ for } \omega \in B \} \] (4.7)

or

\[ T_2 = \{ g \in L^2(\mathbb{R}) : |G(\omega)| = n(\omega) \text{ for } \omega \in B \} \] (4.8)

Second that in transform recovery problems the projections \( P_1 \) may be very easily computed. The disadvantage is that, as stated, the algorithm is sensitive to local ill-conditioning. Figure 3 shows two instances of the effects of ill-conditioning: in the first the sets intersect at a very acute angles so that the projections are very slowly convergent, and in the second the presence of noise has perturbed the two sets so that an intersection point does not exist. These possibilities, and the additional fact that \( T_1 \) is not convex in phase retrieval problems, imply that the iterated projection algorithm will either be very slowly convergent or fail to converge at all.
In order to escape these difficulties we propose that the original problem be replaced by

Find $G$ to minimize $F(G) = \sum_{i=1}^{M} \|G - P_i G\|^2$.

Obviously $F(G) \geq 0$ and $F(G) = 0$ iff $G$ is a common intersection point, but even if such a point does not exist due to perturbations of the sets by noise in the data, the $	ilde{G}$ that minimizes $F(G)$ is an acceptable pseudo-solution to the problem. We also propose the following extension of the iterative projection algorithm for minimization of $F(G)$ in which at the $n$-th iteration

$$H_{n+1} = \frac{1}{(M-1)} \sum_{i=1}^{M-1} (P_i G_n - G_n)$$

$$G_{n+1} = P_M (G_n + \lambda_n H_n)$$

(4.9)

where

$$\lambda_n \in (0, 2) \text{ if } T_M \text{ convex}$$

$$\in (0, 1) \text{ otherwise}$$

(4.10)

This algorithm will be termed the restricted projection (RP) algorithm from here on.

RP has a number of useful features, in particular it produces constantly decreasing residuals as does GS [1], which are summarized in the following Theorem, proved in [15].

Theorem 4.1. If the projections $P_i G$ are unique and continuous in $G$ then at the $n$-th iteration the iterates $G_n$ of Eq. (4.9) satisfy
\[ F(G_{n+1}) < F(G_n) \quad \text{or} \quad G_{n+k} = G_n \quad \forall k > 1. \quad (4.11) \]

Furthermore if \( \bar{G} \) is a limit point of the iterates then \( \bar{G} \) is a fixed point of the iteration.

RP can also be recast as one of the standard optimization algorithms as the next theorem from [15] shows.

**Theorem 4.2.** The gradient \( \nabla F(G) \) of \( F(G) \) is

\[ \nabla F(G) = 2 \sum_{i=1}^{M} (G - P_i G). \quad (4.12) \]

Therefore if \( T_M = L^2(\mathbb{R}) \) then RP is the standard steepest descent algorithm with variable steplength \( \lambda \in (0, \frac{1}{M-1}) \).

We have presented the algorithm in a form in which iterates are restricted to the particular set \( T_M \). This includes the unrestricted case \( (T_M = L^2(\mathbb{R})) \), but also allows from knowledge about the solution \( G \) to be reimposed at each iteration after less well-known requirements are satisfied by moving in the search direction. However, the chief benefit of the restriction in transform recovery problems with the set \( T_M \) of Eq. (4.5) is that the resulting algorithms are efficient. That is at each iteration they require function values of \( G \) and \( \mathcal{F}G \) only across the intervals \( A \) and \( B \). Efficiency has not always been achieved, for instance the version of GS proposed by Papoulis [2] for inversion of the FFT requires knowledge of \( G_n \) across the entire line at each iteration.

To demonstrate that transform recovery problems are efficient we first note that if \( T_M \) is a linear subspace then the projection \( P_M \) is linear and
so that the search direction is independent for $\lambda$, and for any $\lambda$ the new iterate is still in $T_M$. Now for convenience let $A = B = cI$ so that $P_M = P_C$, then

$$P_{c} P_{c} P_{c} G = P_{c} F_{S} P_{c} G$$

$$P_{sg}$$

is easily shown to be

$$m(v) e^{i \arg g(v)}, \quad v \in cI$$

so that

$$P_{sg} = g(v), \quad \text{otherwise}$$

$$P_{s} y = g - P_{c} g + P_{c} P_{s} P_{c} g$$

and

$$P_{c} P_{c} G = G - F_{C} F_{C} G + F_{C} P_{S} F_{C} G$$

Moreover for example 2a

$$(P_{c} P_{2} G)(w) = \frac{n(w) e^{i \arg G(w)}}{n(w)}$$

and for example 2b

$$(P_{c} P_{2} G)(w) = (\text{Re} G)(w), \quad \text{if } (\text{Re} G)(w) > 0 \text{ and } w \in cI$$

It is obvious that calculation of $P_{c} P_{c} P_{c} G$ requires only values of $G$ and $g = F_{C} G$ on $cI$, so that $F_{C}$ is efficient.
We conclude this section by noting that the relationship between RP and gradient method opens up new possibilities for improvement of RP. In particular, as noted in Eq. (4.13), if $T_M$ is linear then the search direction is independent of $\lambda$, therefore it is possible to do a line search in the direction of $H_n$. By Eq. (4.12)

$$
\frac{d}{d\lambda} F(G + \lambda H_n) = \nabla F(G + \lambda H_n, H_n)
$$

$$
= 2 \sum_{i=1}^{M-1} (G_n + \lambda H_n - P_i(G_n + \lambda H_n), H_n) . \quad (4.20)
$$

Since calculation of $F(G)$ requires values of $P_i G$, calculation of $F(G_n + \lambda H_n)$ at any point in a line search gives sufficient information for calculation of the gradient at that point. Therefore a line search algorithm using derivative information, such as Powell's cubic line search algorithm [39], may be used for the same cost as a standard quadratic line search that uses values of $F(G)$ only. Since in phase retrieval $T_M = L^2(cI)$, cubic line searches may be profitably employed.
5. **ALGORITHMS BASED ON AFFINE APPROXIMATIONS**

The previous section showed that the GS algorithm could be extended to an algorithm RP that mitigated some of the effects of ill-conditioning in transform recovery. However the extension does not remove all of these effects as, in particular, it performs no filtering to restrict the solution to a well-posed solution set. Moreover as RP is a variant of steepest descent, it is only of first order and therefore will not perform well even on some well-posed problems for the same reasons that gradient algorithms perform poorly on some standard optimization problems. Therefore we seek a solution to both these problems by proposing a new class of iterative algorithms based on more accurate affine approximations to the sets $T_i$ or the functions $F(G)$ and $P_i G$. At each iteration these algorithms require the solution of an ill-posed linear subproblem similar to that discussed in Section 2; this may be done using filtered SVD thus further reducing ill-conditioning in phase retrieval.

We start with a simple example of how such algorithms may be constructed. In RP the search direction $H_n$ may be viewed as being obtained by replacing the sets $T_i$ by point approximations $P_i G$ at the $n$-th iteration, and then solving the subproblem

$$\text{Minimize } F_n(G) = \sum_{i=1}^{M} \| G - P_i G \|_2^2. \tag{5.1}$$

In particular the set $S$ is approximated by the point $P_S G$. However the only restrictions on functions in $S$ is that they have value $\tilde{g}$ or modulus
m over the interval A; outside of A they may take on arbitrary values. Therefore, if A = B = cl, the affine subspace

\[ S_n = \{ h : h = P_c S_n + (g - P_c g), g \in S \} \]  

(5.2)

contains the point \( P_n g \) and is contained in the set S. If it is used to replace the point approximation \( P_n G \) to \( T_n \) by \( T_n = \mathcal{F}^{-1} S_n \) at the n-th iteration, then the new subproblem to be solved is

\[ \text{minimize } F_n(G) = \| G - P_n T_n G \|^2 + \sum_{i=2}^{M-1} \| G - P_n G \|^2. \]  

(5.3)

The minimum \( L_n \) satisfies the normal equations

\[ (\mathcal{F}_c \mathcal{J}_c + (M-2)\mathcal{J}) L = \mathcal{F}_c P_n g + \sum_{i=2}^{M-1} P_i G \]  

(5.4)

giving a search direction \( H_n = L_n - G_n \). If \( M > 2 \) then this subproblem is well-posed as \( (\mathcal{F}_c \mathcal{J}_c + (M-1)\mathcal{J}) \) has a bounded inverse; but if \( M = 2 \) then the problem is inversion of the FFT \( \mathcal{F}_c \) recast as a linear least squares problem. This is an ill-posed problem best solved by filtered SVD as described in Section 2; but as the same linear operator appears at each iteration the SVD need be calculated only once.

A more accurate approximation arises from replacing the projections \( P_i G \) by the linear approximation

\[ P_i G \sim P_i G_n + \mathcal{H}_i(G_n)(G - G_n) \]  

(5.5)

at the n-th iteration, where \( \mathcal{H}_i \) is the Frechet derivative of the operator \( P_i \). We assume that \( \mathcal{H}_i \) exists and is a bounded linear operator; conditions
on the sets $T_i$ that would guarantee these properties are quite complicated (e.g. [40]) and are beyond the scope of this work. However if $\mathcal{H}_1$ possesses these properties then it is symmetric, $\mathcal{I} - \mathcal{H}_1$ is the Hessian (i.e. the second Frechet derivative) of the function $F_1(G) = \|G - P_1 G\|^2$ and for every $G$

$$(\mathcal{I} - \mathcal{H}_1(G))(G - P_1 G) = 0 \quad \text{(5.6)}$$

If this approximation is used at the $n$-th iteration the corresponding subproblem to be solved is

$$\text{minimize } F_n(G) = \sum_{i=1}^{M-1} \|G - P_i G - \mathcal{H}_1(G_n)(G - G_n)\|^2 \quad \text{(5.7)}$$

The function $L$ that solves this problem is by definition the least squares solution to the block system

$$\begin{bmatrix}
\mathcal{I} - \mathcal{H}_1(G_n) \\
\vdots \\
\mathcal{I} - \mathcal{H}_M(G_n)
\end{bmatrix} L = -\begin{bmatrix}
G_n - P_1 G_n \\
\vdots \\
G_n - P_{M-1} G_n
\end{bmatrix} \quad \text{(5.8)}$$

Since the original transform recovery problem was ill-posed, this block system is also usually ill-posed and should be inverted by filtered SVD. However the cost of calculation of the SVD of the entire block matrix at each iteration would very expensive. Therefore we propose that the SVD of each block $\mathcal{I} - \mathcal{H}_1(G_n)$ be calculated and filtered separately to give an approximate filtering of the whole matrix. As we shall see later in the section such filtering can be done relatively cheaply.
The third set of algorithms described here are based on the following affine approximation of \( F(G) \) by

\[
F(G) \approx F(G_n) + \nabla F(G_n) (G - G_n) + \frac{1}{2} (G - G_n) \nabla^2 F(G_n) (G - G_n)
\]  

(5.9)

From Eqs. (4.11) and (5.5),

\[
\nabla F(G) = 2 \sum_{i=1}^{M} (G - P_i G)
\]

\[
\nabla^2 F(G) = 2 \sum_{i=1}^{M} (I - H_i (G))
\]

(5.10)

At each iteration this approximation gives the subproblem

\[
\text{minimize } F_n(G) = F(G_n) + \nabla F(G_n) (G - G_n) + \frac{1}{2} (G - G_n) \nabla^2 F(G_n) (G - G_n)
\]

(5.11)

which, if \( \nabla^2 F(G_n) \) is positive definite, has as its unique minimum the solution \( L_n \) of Newton's equation

\[
\nabla^2 F(G_n) L = -\nabla F(G_n)
\]

(5.12)

giving the standard Newton search direction \( H_n = L_n - G_n \). However, for phase retrieval problems, in order that the resulting algorithm be efficient in the sense of the previous section we replace Eq. (5.12) by the restricted equation

\[
P_c \nabla^2 F(G_n) P_c L = -P_c \nabla F(G_n)
\]

(5.13)

so that values of iterates are required only over the interval \( cI \).
Equations (5.12) and (5.13) are normally ill-posed and therefore best solved by filtered SVD. As the FFT, with its exponentially decaying singular values, underlies the Hessian $\nabla^2 F(G)$ we propose a cutoff filter of the form

$$f(\lambda) = \begin{cases} \lambda^{-1} & \text{if } |\lambda| \geq \varepsilon_3 \\ 0 & \text{otherwise} \end{cases}$$

(5.14)

where $\varepsilon_3$ is dependent on the noise levels and desired accuracy. This should produce a search direction $H_n$ in which $F(G)$ should vary moderately rapidly as directions corresponding to small eigenvalues, and thus slowly varying $F$, have been filtered out. However the resulting direction is not necessarily a descent direction, as for some $G$, $\nabla^2 F(G)$ will have negative eigenvalues; but this may be corrected by use of the filter

$$f(\lambda) = \begin{cases} \lambda^{-1} & \text{if } \lambda \geq \varepsilon_3 \\ 0 & \text{otherwise} \end{cases}$$

(5.15)

Newton's algorithm in which Eq. (5.13) is inverted with the filter of Eq. (5.14) shall be denoted by $F_N$; if the filter of Eq. (5.15) is used it will be denoted by $F_{NP}$.

Since $\nabla^2 F(G_n)$ changes with each iteration $F_N$ and $F_{NP}$ incur considerable costs in calculation of a new SVD at each iteration. Although some savings are possible by using the old SVD as a first approximation to the new SVD with optional iterative refinement, we instead sought to reduce Eq. (5.13) to the block form of Eq. (5.8) so that block filtering could be applied. This may be done by noting that if $I-H_i$ is a symmetric positive
definite square root \((I - \mathcal{K}_1)^{1/2}\) which by Eq. (5.6) satisfies

\[
(I - \mathcal{K}_1(G))(G - P_i G) = G - P_i G.
\]  

(5.16)

Therefore Eq. (5.13)

\[
\left[ \sum_{i=1}^{M-1} P_c (I - \mathcal{K}_1(G_n)) P_c \right] L = - \left[ \sum_{i=1}^{M-1} (G_n - P_i P_c G_n) \right]
\]

(5.17)

may be rewritten as

\[
\left[ \sum_{i=1}^{M-1} P_c (I - \mathcal{K}_1(G_n))^{1/2} P_c \right] L = - \left[ \sum_{i=1}^{M-1} P_c (I - \mathcal{K}_1(G_n))^{1/2} (G_n - P_i G_n) \right]
\]

(5.18)

which in turn is recognizable as the normal equations for the least squares solution of the block system

\[
\begin{bmatrix}
(I - \mathcal{K}_1(G_n))^{1/2} P_c \\
\vdots \\
(I - \mathcal{K}_{M-1}(G_n))^{1/2} P_c
\end{bmatrix} L = \begin{bmatrix}
G_n - P_i P_c G_n \\
\vdots \\
G_n - P_c P_{M-1} G_n
\end{bmatrix}
\]

(5.19)

Block filtering was chosen because of the simple form of the blocks in Eq. (5.19) in phase retrieval problems. If the operators \(P_i\) are viewed as acting on the pair of real functions \(\frac{Re H(\omega)}{Im H(\omega)}\) instead of on the complex valued function \(H(\omega)\) then simple calculations show that \(I - \mathcal{K}_2(G)\) may be represented as a block diagonal operator whose \(2 \times 2\) diagonal blocks, are indexed by the variable \(\omega\). For example 2a
\[
\begin{bmatrix}
\sin \theta & \cos \theta \\
-cos \theta & \sin \theta
\end{bmatrix}
\begin{bmatrix}
1 - \frac{n(\omega)}{|G(\omega)|} & 0 \\
0 & 1
\end{bmatrix}
\]

\[(\mathcal{J} - \mathcal{H}_2(G))(\omega) = \begin{bmatrix}
\sin \theta & -\cos \theta \\
\cos \theta & \sin \theta
\end{bmatrix}
\]

where \( \theta = \text{arg } G(\omega) \)

and for example 2b

\[
\begin{bmatrix}
0 & 0 \\
0 & 1
\end{bmatrix}
\]

if \( \text{Re } G(\omega) > 0 \) and \( \omega \in \mathbb{C} \)

\[(\mathcal{J} - \mathcal{H}_2(G))(\omega) = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]

otherwise

The spectrum of each operator \( \mathcal{J} - \mathcal{H}_2 \) is now recognizable as the union over \( \omega \) of the spectrum of each block. For example 2b the block spectra consists only of the set \( \{0, 1\} \) so \( \mathcal{J} - \mathcal{H}_2 \) is its own square root. For example 2a the spectrum of \( \mathcal{J} - \mathcal{H}_2 \) takes on a range of values, some of which may be small or negative. Consequently \( (\mathcal{J} - \mathcal{H}_2)^{1/2} \) is either ill-conditioned, or not well defined (i.e. has imaginary eigenvalues) or is both. Therefore a filter with one of the following forms is used

\[
f(\lambda) = \frac{\lambda^{1/2}}{|\lambda|^{1/2}} \text{ sign } \lambda, \quad \text{if } |\lambda| \geq \varepsilon_3
\]

\[
= 0, \quad \text{otherwise}
\]

\[
f(\lambda) = \lambda^{1/2}, \quad \text{if } \lambda \geq \varepsilon_3
\]

\[
= 0, \quad \text{otherwise}
\]

(5.22)

(5.23)

To replace the entry \( 1 - \frac{n(\omega)}{|G(\omega)|} \) in Eq. (5.20) by \( f(1 - \frac{n(\omega)}{|G(\omega)|}) \). Finally for either problem the original block \( (\mathcal{J} - \mathcal{H}_2(G))p_c \) is reduced in size by...
eliminating from the block those equations for \((\text{Re } H)(\omega)\) or \(\sin \theta(\text{Re } H)(\omega) - \cos \theta(\text{Im } H)(\omega)\) that correspond to eigenvalues that have been filtered to zero.

From Eq. (4.15)

\[
P_c (\mathcal{I} - \mathcal{H}_1(G)) P_c = \mathcal{F}_c (\mathcal{I} - \mathcal{H}_S(g)) \mathcal{F}_c
\]  

(5.24)

where \(g = \mathcal{G}\) and \(\mathcal{I} - \mathcal{H}_S(g)\) is the block diagonal operator

\[
(\mathcal{I} - \mathcal{H}_S(g)) = \begin{bmatrix}
\sin \phi & \cos \phi \\
-\cos \phi & \sin \phi
\end{bmatrix} \begin{bmatrix}
1 - \frac{m(v)}{|g(v)|} & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\sin \phi & -\cos \phi \\
\cos \phi & \sin \phi
\end{bmatrix}
\]  

(5.25)

where \(\phi = \arg g(v)\).

\(P_c (\mathcal{I} - \mathcal{H}_1(G)) P_c\) does not possess an obvious symmetric square root however Eq. (5.24) suggests the asymmetric square root \((\mathcal{I} - \mathcal{H}_S(g))^{1/2} \mathcal{F}_c\). This composite operator may be approximately filtered by filtering each component;

\((\mathcal{I} - \mathcal{H}_S(g))^{1/2}\) may be filtered as was \((\mathcal{I} - \mathcal{H}_2)\) in example 2a and \(\mathcal{F}_c\) may be filtered after calculation of its SVD as described in Section 2.

Since the first filtering requires a trivial calculation at each iteration, and \(\mathcal{F}_c\) is independent of the iterates, this filtering may again be done cheaply.

Therefore, after rows corresponding to filtered elements have been eliminated, in phase retrieval problems we are left with a reduced block system

\[
\begin{bmatrix}
(\mathcal{I} - \mathcal{H}_S(g))^{1/2} & (\mathcal{F}_c) f
\end{bmatrix} L = \begin{bmatrix}
P_c P_S \mathcal{F} G_c \\
P_c G_n - G_n
\end{bmatrix}_f
\]  

(5.26)

33
to solve in a least squares sense. This can be done using a standard routine such as LLSQF in the IMSL library, or further advantage may be taken of sparsity within the system. If Eq. (5.26) is solved at each iteration with the filter of Eq. (5.22) resulting algorithm is termed SQFN; if the filter of Eq. (5.23) is used the algorithm is SQFNP.
6. SOME ILLUSTRATIVE NUMERICAL RESULTS

To conclude the paper we give a brief account of some numerical solutions to examples 2, 2a and 2b using the algorithms proposed in the previous sections. Reference is made to our two previous papers for detailed numerical computations of a wide selection of problems. The measure of performance of the algorithms was taken to be the number of iterations required to reduce the function $F(G_n)$ to below a prescribed value. The ill-posed nature of phase retrieval problems implies that this is not the best of measures in that a small value $F(G_n)$ does not necessarily imply that $G_n$ is close to the true solution $G$. However without knowledge of the true solution we have none better. Therefore, because of this ill-conditioning and the limited resources for numerical computation at our disposal, the results presented here are intended as a guide to the behavior of the algorithms on real problems rather than a firm prediction of their likely performance.

We begin with an account of the mechanics of the computation. First is the discretization: it was determined by the requirements that it gives an accurate, efficient approximation to $S_c$, and that the discrete projections be easily calculated. Therefore a discretization based on $N$ point Gaussian quadrature was chosen; the numerical experiments mentioned in Section 2 showed its efficiency and its pointwise nature allows easy evaluation of projections. The discrete problem thus involves determination of vectors $\hat{g}, \hat{G} \in \mathbb{C}^N$ whose elements $g_k, G_k$ are to be approximations to the function values $g(\rho_k), G(\rho_k)$ at the abscissae $\rho_k$ of the $N$-point Gaussian quadrature.
rule on \( cI \). To this end, vectors \( \hat{m}, \hat{n} \in \mathbb{R}^N \) are formed with components \( m_k = m(\rho_k), \ n_\alpha = n(\rho_\alpha) \) where \( m(\nu) \) and \( n(\omega) \) are the known moduli. Then matrices \( \hat{W}, \hat{F} \in \mathbb{C}^{N \times N} \) are constructed with \( \hat{W} \) being a diagonal matrix whose \( k \)-th entry is the weight \( w_k \) of the quadrature rule, and \( \hat{F} \) having entries \( F_{k\ell} = e^{2\pi i \omega_{k\ell}} \). The discretized example problems are now.

Find vectors \( \hat{g}, \hat{G} \) such that \( \hat{g} = \hat{F} \hat{W} \hat{G} \) and

Example 2: \( |g_k| = m_k \)

Example 2a: \( |g_k| = m_k, \ |G_k| = n_\alpha \)

Example 2b: \( |g_k| = m_k, \ G_\alpha \geq 0 \).

In order to test the algorithms, particular examples of each model problem were chosen. Of particular interest, in-as-far as this paper is concerned, is the test function

\[
G(\omega) = e^{i2\pi W(\omega)}
\]

(6.1)

\[
W(\omega) = W_3 S_3(\omega) + W_4 S_4(\omega)
\]

\[
= W_3 \left[ \frac{(\omega)}{c}^3 - \frac{3}{5} \frac{(\omega)}{c} \right] + W_4 \left[ \frac{(\omega)}{c}^4 - \frac{6}{7} \frac{(\omega)}{c}^2 \right]
\]

(6.2)

\( G(\omega) \) is the pupil function of a slit aperture having unit amplitude over the exit pupil and suffering from wavefront aberrations of optimum balanced coma \( S_3 \) and optimum balanced spherical aberration \( S_4 \), where \( W_3/\lambda \) and \( W_4/\lambda \) are the dimensionless aberration strengths. \( S_3 \) and \( S_4 \) are the slit aperture versions [41] of the Zernike polynomials. This test function was used previously in [9]. In reference [9], the inversion assumed that the amplitude distribution over the exit pupil was unity; in the present case...
neither the amplitude distribution or wavefront are known a priori. The numerical calculations summarized here were carried out for $W_3 = W_4 = 3\lambda/8$.

We now outline the basic structure of all the numerical test runs before considering various points in detail. All tests consisted of the four essential steps:

1. Choose an initial guess $G_0$.
2. Given iterates $G_n$ and $G_n$ compute a search direction $\hat{H}_n$.
3. Calculate a steplength $\lambda_n$ and new iterates

$$G_{n+1} = G_n + \lambda_n \hat{H}_n, \quad G_{n+1} = F(WG_n + G_n)$$

4. Iterate steps 2 and 3 until the convergence criteria are satisfied.

The convergence criteria used in all calculations were

$$F(G_n) < 10^{-5} \quad \text{or} \quad \sum_{k=0}^{2} \| \lambda_n \hat{H}_n \| < 2 \times 10^{-3}$$

(together with an upper limit $N_{\text{max}}$ on the number of iterations. The sum of the last three steplengths, rather than $\| \lambda_n \hat{H}_n \|$ alone, was chosen as in ill-conditioned minimization problems "stop-start" behavior is often noticed.

That is a large step often followed by one or two small steps, after which another large step is taken. This feature was often observed in the use of affine approximation algorithms, differences in magnitude of successive steplengths by factors greater than 100 occurred fairly frequently.

Three different forms of initial guesses were used:

1. $G_0 = 0$,
2. $G_0 = (1+\varepsilon_1)(1+\varepsilon_1)^{-1}G_0$, 

3. \( G_{0k} = 10^{-3} r_k \).

Guess 2 is a damped perturbation of the true solution \( \hat{G}^i \) with \( \epsilon_1 \) representing the noise level. For convenience we shall express this level as a percentage, e.g. \( \epsilon_1 = .2 \) will be described as \( \epsilon_1 = 20\% \) noise. The variable \( r_k \) is a random complex variable with modulus uniformly distributed over \([0,1]\) and phase uniformly distributed over \([0,2\pi]\). Guess 3 represents a small totally random perturbation about the origin; again for convenience such guesses shall be denoted by \( \epsilon_1 = 100\% \).

The first results reported are those on determination of an optimal choice of \( \lambda_n \). Three possibilities were considered:

1. \( \lambda_n^1 = 1 \).
2. \( \lambda_n^2 = r_n \) where \( r_n \) is a random variable uniformly distributed over \([0,2]\).
3. \( \lambda_n^3 \) is the approximate minimum of \( F(\hat{G}_n + \lambda_n \hat{H}_n) \) as a function of \( \lambda_n \) determined by Powell's cubic line search algorithm [39] with the following convergence criteria on the iterates \( \lambda_n^k \):

\[
\left| \frac{(\lambda_n^3 - \lambda_n^2)}{\lambda_n^3} \right| \cdot \left| \frac{F(\hat{G}_n + \lambda_n^3 \hat{H}_n)}{F(\hat{G}_n)} \right| < .02
\]

\[
\left| \frac{\nabla F(\hat{G}_n + \lambda_n^3 \hat{H}_n)}{\nabla F(\hat{G}_n)} \right| \cdot \left| \frac{F(\hat{G}_n + \lambda_n^3 \hat{H}_n)}{F(\hat{G}_n)} \right| < .02
\]

\( k < 5 \quad (6.4) \)

The performances of \( \lambda_n^i \) were compared by running each possibility on each model problem with the appropriate test functions using RP. The parameters
c = 2, $N = 40$ and $N_{\text{max}} = 50$ were chosen and each problem was started with three different initial guesses with $\epsilon_1 = 20\%, 60\%$ and $100\%$.

The results were remarkably uniform over all test cases of model problem, test function and initial guess. Choices $\lambda_n^1$ and $\lambda_n^2$ performed almost identically with $\lambda_n^3$ just under a factor of 2 better. Almost always only one extra function evaluation was required for $\lambda_n^3$. This extra computation almost exactly balances the savings in the reduced number of iterations so that all three choices incurred the same computational cost in reduction of $F(\hat{G}_n)$ to a specified value. However the greater flexibility of $\lambda_n^3$ led to its adoption in all subsequent calculations.

We next attempted to estimate the local ill-conditioning in the problem by adding a small perturbation of size $\epsilon_2$ to the data and starting the algorithm at the true solution $\hat{G}_i$ of the unperturbed problem. The results were inconclusive; although the algorithms terminated at a vector $\hat{G}_i$ such that $\|\hat{G}_i - G_i\|/\|G_i\| \approx \epsilon_2$, this was due to the conditions $F(\hat{G}_i) < 10^{-5}$ or $n > N_{\text{max}}$ being satisfied, and not due to convergence of the iterates which appeared to cycle around some fixed point.

Further results showed that the problem was globally ill-conditioned in that a number of differing functions $\hat{G}_i$ were found starting from a random guess ($\epsilon_1 = 100\%$) such that $F(\hat{G}_i) \sim 10^{-4}$ for example 2, or $F(\hat{G}_i) \sim 10^{-2}$ for examples 2a and 2b, but with $\|\hat{G}_i - G_i\|/\|G_i\| \sim 1$. This suggests that the surface $\{(F(G), G): G \in \Omega\}$ is very rugged, as expected from the results in Section 3 on existence of multiple solution in one dimension. The results also showed that the iterates determined as approximate solutions to example
2a and 2b were definitely more acceptable as solutions than those for example 2, even though as measured by $F(G)$ they were worse by a factor of 10 or more.

A good initial guess is of great help. If one is not available then it is tempting to start with guess i, however some analysis [9] has shown that if the iterates have some symmetries, then such symmetries will be preserved by the algorithm regardless of the form of the true solution. Thus the highly symmetric choice of $G_0 = \hat{0}$ is to be avoided.

An attempt to estimate the effects of ill-conditioning due to the presence of $F_{c}$ was made by restricting iterates in $F_{P}$ to the span of the first few singular functions of $F_{c}$. When done for $c=2$ the resulting iterates reduced $F(\hat{G}_{n})$ at a slower rate, gave final iterates $\hat{G}^{i}$ for which $\|\hat{G}^{i} - G^{i}\|$ was approximately the same as in the unfiltered algorithm, and still failed to terminate due to the convergence of successive iterates but rather ended due to the satisfaction of one of the other two conditions. For this value of $c$ there are approximately nineteen significantly nonzero singular values; the results indicate that even over this subspace the function $F(G)$ is widely varying due to the nonlinearity and existence of multiple solutions.

It therefore seems likely that a severe restriction, e.g. to the span of the first five singular functions, is necessary to provide an easily solved problem.

The numerical results presented are some typical examples of the behavior of the algorithms on model problems 2 and 2a, with parameter pairs $(c,N)$ of $(1.5,22)$, $(2,32)$ and $(2,40)$ and test function $G$, Eq. (6.1). For numerical
results concerning model problem 2b, see [15]. Tables 1 and 2 first give the average number of iterations \( \hat{n} \) and average final value \( F(G_n) \) of RP when iterated to convergence on several different initial guesses, each perturbed by an amount \( \epsilon_1 \) from the true solution. The remaining entries are the ratio \( \hat{n}/\bar{n} \), where \( \bar{n} \) is the number of iterations required by the remaining algorithms to reduce \( F(G'_n) \) to below \( F(G_n) \). Cases where the algorithm on trial failed to reduce \( F(G'_n) \) to within 100 \( F(G_n) \) are denoted by a *. Figures 4-7 show 4 typical final iterates reached by these algorithm for varying problems and values \( \epsilon_1 \).

The filter parameter \( \epsilon_3 \) of Eq. (5.14), (5.15), (5.22) and (5.23) at the \( n \)-th iteration was calculated from the quantity

\[
\epsilon = \max\{\min\{4\|\lambda_{n-1}H_{n-1}\| - 0.25, 0.2, 0.002\}\} . \tag{6.5}
\]

For FN and FNP, \( \epsilon_3 \) was taken to be \( \rho \); for SQFN and SQFNP, \( \epsilon_3 = \rho/3 \). This expression was calculated by trial and error and, although by no means the last word, at least has the property of giving search direction \( H_n \) such that the associated step length \( \lambda_n \) almost always lay in the interval [1.1,1.5] and \( \lambda_n \sim 1 \) when close to the true solution.
7. CONCLUSION

The ill-posed nature of phase retrieval induced too much variation in numerical calculations to allow the drawing of quantitative judgements from these results, but some qualitative remarks can be made. For these one-dimensional problems RF was clearly the least expensive in terms of total computational cost to reach a desired objective, and it is difficult to see how any other algorithm can be improved through taking advantage of sparse Hessians, etc. to seriously challenge RF for this position. However SQFNP was the most robust in producing acceptable iterates from almost any starting point, and although it did not display to the same degree the apparent quadratic convergence of FN and FNP when close to the true solution, this could possibly be remedied by a better choice of filter.

Efforts to estimate the role of local ill-conditioning and filtering in phase retrieval problems were largely frustrated by the effects of global ill-conditioning due to the existence of multiple solutions. This makes one dimensional problems unsolvable for practical purposes, but it is expected that in higher dimensional problems of real interest, with fewer possible exact solutions, that local ill-conditioning will become dominant. Hopefully this may be removed by filtering and quadratically convergent algorithms will have a wider domain of convergence, thus becoming competitive with the present first order iterative schemes.
Table 1. Relative performance of algorithms on example 2 with test function $G$, Eq. (6.1).

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>$N$</th>
<th>FN</th>
<th>FNP</th>
<th>SQFN</th>
<th>SQFNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>50</td>
<td>1.9 x 10^{-5}</td>
<td>1.5</td>
<td>1.0</td>
<td>.40</td>
</tr>
<tr>
<td>32</td>
<td>1.5 x 10^{-5}</td>
<td>20</td>
<td>*</td>
<td>.825</td>
<td>.875</td>
</tr>
<tr>
<td>40</td>
<td>3 x 10^{-5}</td>
<td>40</td>
<td>*</td>
<td>1.0</td>
<td>.35</td>
</tr>
<tr>
<td>22</td>
<td>5 x 10^{-5}</td>
<td>50</td>
<td>1.5</td>
<td>.30</td>
<td>.60</td>
</tr>
<tr>
<td>32</td>
<td>4 x 10^{-4}</td>
<td>20</td>
<td>1.6</td>
<td>.45</td>
<td>.35</td>
</tr>
<tr>
<td>40</td>
<td>8 x 10^{-4}</td>
<td>40</td>
<td>.50</td>
<td>.50</td>
<td>.425</td>
</tr>
</tbody>
</table>
Table 2. Relative performance of algorithms on example 2a with test function G, Eq. (6.1).

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>N</th>
<th>$F_N$</th>
<th>$F_{NP}$</th>
<th>$SQF_N$</th>
<th>$SQF_{NP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>22</td>
<td>$9 \times 10^{-6}$</td>
<td>23</td>
<td>.20</td>
<td>.20</td>
</tr>
<tr>
<td>60</td>
<td>22</td>
<td>$5 \times 10^{-4}$</td>
<td>50</td>
<td>.50</td>
<td>.30</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>$3 \times 10^{-4}$</td>
<td>20</td>
<td>2.0</td>
<td>.45</td>
</tr>
<tr>
<td>100</td>
<td>22</td>
<td>$2 \times 10^{-3}$</td>
<td>50</td>
<td>3.0</td>
<td>.80</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>$6 \times 10^{-2}$</td>
<td>20</td>
<td>*</td>
<td>1.0</td>
</tr>
</tbody>
</table>
REFERENCES


Fig. 1. Two sample realizations (— — and —) of the reconstruction of
$G(\omega)$, Eq. (2.16), in the presence of 3% noise in $S_c G$, Eq. (2.14).
Fig. 2. Extrapolation of $g(v)$, Eq. (2.15), see open circles, corresponding to \ldots in Fig. 1.
Fig. 3. Two examples of poor convergence of the alternating projection algorithm induced by ill-conditioning.
Fig. 4. Solution $\hat{G}$ to example 2 reached from an initial guess $\hat{G}_0$ with $\varepsilon_1 = 20\%$: — exact phase from Eq. (6.1); ○ reconstructed phase;

--- exact modulus from Eq. (6.1); △ reconstructed modulus.
Fig. 5. Solution $\hat{G}$ to example 2 reached from an initial guess $\hat{G}_0$ with $\varepsilon_1 = 60\%$: — exact phase from Eq. (6.1); ○ reconstructed phase; —— exact modulus from Eq. (6.1); △ reconstructed modulus.
Fig. 6. Solution $\hat{G}$ to example 2a reached from an initial guess $\hat{G}_0$ with $\varepsilon_1 = 60\%$: — exact phase from Eq. (6.1); ◦ reconstructed phase; ---- exact modulus from Eq. (6.1); △ reconstructed modulus.
Fig. 7. Solution $\hat{G}$ to example 2a reached from an initial guess $\hat{G}_0$ with $\epsilon_1 = 100\%$: — exact phase from Eq. (6.1); ○ reconstructed phase; --- exact modulus from Eq. (6.1); △ reconstructed modulus.
ALGORITHMS FOR RECONSTRUCTION OF PARTIALLY KNOWN, BANDLIMITED FOURIER TRANSFORM PAIRS FROM NOISY DATA:

I. THE PROTOTYPICAL LINEAR PROBLEM
ABSTRACT

Problems involving reconstruction of partially known, bandlimited Fourier transform pairs from noisy data are now regularly encountered in a wide variety of scientific and technical areas. This paper is the first in a series of studies of algorithms for their solution. These studies focus on the algorithmic structure with respect to the dominant feature of such problems, that they are ill-posed. The algorithms developed here are for the linear prototype problem; namely extrapolation of a bandlimited function known over a finite interval. The problem is cast as the inversion of a linear operator, the finite Fourier transform. Criteria can be deduced from a knowledge of the spectrum of this operator for the suitability of extrapolation algorithms. These criteria are used to evaluate existing algorithms (such as the Gerschberg-Saxton) as well as our new algorithm based upon inversion of a Galerkin approximation to the operator using singular value decomposition. Numerical results on the relative merits of different discretizations in our algorithm, and on its success in extrapolation in two examples (with optical diffraction interpretations) with noisy data are presented.
1. INTRODUCTION

A recurring problem in many fields is the reconstruction of a Fourier transform pair $g$, $G$ from partial data on either or both functions. Considerable effort has been expended in the development of algorithms for its solution; although there have been some successes, the problem has generally proved to be difficult. In a series of papers, of which this is the first, we aim to detail the features that cause difficulty for numerical calculations and to construct algorithms that take explicit account of such features. It is not possible to remove such difficulties, but if they are ignored in the construction of algorithms they invariably surface later in the solution in a more inconvenient form.

The canonical examples for such reconstructions are:

Example 1, Extrapolation of Band-Limited Signals. Given a noisy measurement of $g$ on an interval $[a_1, a_2]$ and the knowledge that $G$ vanishes outside the bounded interval $[b_1, b_2]$, reconstruct $g$ and $G$ on the entire real line.

Example 1 is the archetypical linear problem in transform reconstruction. A number of algorithms have been proposed for its solution, either by iterative means: Gerschberg and Saxton [1], Papoulis [2], Youla [3] or by direct means: Cadzow [4], Sabri and Steenaart [5].

Example 2, The Phase Problem. Given a noisy measurement of $|g|$ on an interval $[a_1, a_2]$ and the knowledge that $G$ vanishes outside the bounded interval $[b_1, b_2]$, reconstruct $g$ and $G$ on the entire real line.
Example 2 has been of theoretical interest for some time; see for example Burge, Fiddy, Greenway, and Ross [6]; however in this most general form it has proved intractable. Numerical solutions obtained in particular cases have done so by (sensibly) incorporating further knowledge of \( g \) and \( G \). Two such examples are:

Example 2a, The Two Moduli Problem. Given noisy measurements of \( |g| \) on \([a_1, a_2]\), \( |G| \) on \([b_1, b_2]\), and the knowledge that \( G \) vanishes identically outside of \([b_1, b_2]\), reconstruct \( g, G \) over the entire real line.

Example 2b, The Phase Problem with Nonnegativity Constraints. Given noisy measurements of \( |g| \) on \([a_1, a_2]\) and the knowledge that \( G \) is nonnegative over \([b_1, b_2]\) and vanishes identically outside \([b_1, b_2]\), reconstruct \( g, G \) over the real line.

Gerschberg and Saxton [1] developed a widely used algorithm for Example 2a; almost all the iterative algorithms for the solution of the general reconstruction problem are suitably modified versions of this particular case. The Gerschberg-Saxton algorithm (hereafter denoted as the GS algorithm) in its general form is essentially the steepest descent algorithm with unit step length and so is first order [7]. An alternative second order algorithm, based on Newton's method, has been proposed by Barakat and Newsam [8]. In [9], the GS algorithm is successfully applied to a problem of the type in Example 2b.

The canonical examples presented are simple in form, nevertheless they contain the salient features that make other, more complicated, problems intractable. The font of all difficulties is that the reconstruction problem is ill posed, and badly ill posed at that. The original definition of a well posed problem is due to Hadamard [10].
Definition. A problem is well posed if the solution

1. exists
2. is unique
3. depends continuously on the data.

If a problem violates any of these conditions, it is ill posed.

Example 1 satisfies only condition 2; therefore it is ill posed. To see that it fails condition 1, it suffices to note that $g$ is the transform of a function with bounded support so that it is analytic by the Paley-Wiener theorem [11]. Any perturbation of the true $g$ by noise that is not analytic produces a perturbed problem with no solution. Analyticity does imply condition 2, but also that condition 3 will fail very badly. This is due to the natural error metric for noise being the standard energy norm, which is of no use as a measure of analyticity. Therefore noise that is arbitrarily small in the error metric will produce arbitrarily large changes in the solution, or even cause it not to exist. This aspect of the problem is discussed in section 2.

Example 2 fails all three conditions; condition 1 for the same reason as Example 2. Akutowicz [12,13] shows the existence of multiple solutions for the phase problem both with and without nonnegativity constraints. It is not known whether the two moduli problem has multiple solutions, except for some trivial multiplicities due to symmetry and constant phase factors. As for condition 3, Example 2 can be locally linearized into Example 1, so it also fails this condition.
Many researchers do not seem to be aware of the full implications of these failures for numerical algorithms for approximate solutions. Gerschberg and Saxton have been careful to show that their algorithm is always error decreasing and to give examples of behavior using noisy data. Chapman [14] modified their algorithm to make even greater use of known noise levels. However the overall approach is still on an \textit{ad hoc} basis, noise cannot be explicitly handled in a satisfactory fashion in an iterative algorithm. Papoulis [2] gives error bounds for his iterative method, but not a way of incorporating them efficiently into calculations. It will be shown in this paper that direct methods do allow development of robust algorithms. The two direct methods proposed in Cadzow [3], Sabri and Steenaart [4] call for inversion of ill conditioned matrices formed with no concern for optimizing the condition number. The optics literature contains many references to "the number of degrees of freedom" [15,16].

However, the natural corollary has not been noted: numerical solutions to this problem should contain discretizations of approximately this size and no significant improvement in results can be obtained by using greater numbers of points, approximating functions, etc. Ignorance of this point has lead to unnecessarily long calculations in the belief that more points give greater accuracy.

This paper aims to provide a detailed study of Example 1 in the language of numerical analysis, and to use this knowledge to construct numerical algorithms for its solution in a rational manner. Section 2 contains an exposition of the structure of the finite Fourier transform and the linear operator associated with Example 1. Using this structure in conjunction with the theory of singular value decomposition, the solution is decomposed into
two parts; the first of which is finite dimensional and contains significant information in the presence of noise, while the second contains no trustworthy information in the presence of noise. In section 3, numerical algorithms are proposed that efficiently identify these two components, in addition a list of desirable features of possible discretizations is given. Section 4 contains an exposition of iterative methods, and a comparison of their merits with those of the direct methods of section 3. Finally section 5 contains a discussion of some particular discretizations with numerical calculations showing their robustness in the presence of noise.

A clear exposition of the linear problem is important not only in its own right, because as mentioned above the local structure of nonlinear reconstructions can be approximated by linearization in the neighborhood. The results of section 2 on this linearization show that the nonlinear reconstructions are essentially problems over finite dimensional manifolds. If Newton's method is used to carry out these reconstructions, an accurate efficient way of identifying the tangent plane to this manifold is needed. Reference [8] shows that singular value decomposition can accomplish this task. We believe that the results of this paper show that Newton's method coupled with careful discretizations for nonlinear reconstructions can be competitive with the widely used iterative methods.
2. THEORY OF THE MODEL LINEAR PROBLEM

This section contains an analysis of the theory of the linear reconstruction problem discussed in the previous section. A formal description is:

Given a function \( \hat{g}(v) \) measured on \( v \in A \equiv [a_1, a_2] \) that is known to be in the Fourier transform of a function \( G(\omega) \) which has support contained in \( B \equiv [b_1, b_2] \), extend \( \hat{g}(v) \) to a function \( g(v) \) defined on the entire real axis.

For convenience we introduce the following notation.

**Definition 1.** Let \( G(\omega) \in L^2 \), the Fourier transform \( g(v) \) of \( G(\omega) \) is defined as

\[
g(v) = \int_{-\infty}^{\infty} e^{2\pi i \omega v} G(\omega) d\omega \tag{2.1}
\]

and denoted by

\[
g = \mathcal{F}G
\tag{2.2}
\]

The inverse transform is denoted by

\[
G = \mathcal{F}^{-1}g
\tag{2.3}
\]

**Definition 2.** Let \( S \) be a subset of the real line. The projection operator on \( L^2 \) associated with \( S \) is denoted by \( P_S \) and defined by

\[
(P_S f)(v) = f(v) \quad v \in S
\]

\[
= 0 \quad v \notin S
\tag{2.4}
\]
In this notation, the model problem is: Given a measurement \( \hat{g} \), find \( g \) and \( G \) such that

\[
\hat{g} = P_B F_P G, \quad g = F_P G \tag{2.5}
\]

The problem can be symmetrized using simple translations and scaling to read:

Given a measurement \( \hat{g} \), find \( g \), \( G \) such that

\[
\hat{g} = P_C F_P G, \quad g = F_P G \tag{2.6}
\]

where

\[
C = [-c,c], \quad c \equiv \frac{1}{2} \left[ (a_2 - a_1)(b_2 - b_1) \right]^{1/2} \tag{2.7}
\]

To solve Equation 2.6, we need to convert the linear operator \( P_C F_P C \). As noted in the previous section this is an ill posed problem.

By the Paley-Wiener theorem [11], \( \hat{g}(v) \), the Fourier transform of a function with bounded support is analytic; consequently it has a unique extension \( g(v) \). Therefore the model problem is one of analytic continuation which is known to be highly sensitive to measurement error in \( \hat{g}(v) \), see [1,8]. In order to make quantitative judgements on the effects of noise in Equation 2.6, we shall use the concept of singular value decomposition. We assume the reader to be familiar with this technique applied to either infinite dimensional compact linear operators or to finite dimensional matrices; a discussion of the first case can be found in Baker [17], of the second in Stewart [18].

In a series of papers, Slepian et al. [19-21], the structure of \( P_C F_P C \) was fully explored. The key results are repeated here:

1) \( P_C F_P C \) is a normal compact operator, and

\[
\| P_C F_P C \|_2 \leq 1 \tag{2.8}
\]
2) The eigenfunctions of $P_c \mathcal{A} P_c$ can be taken as real. They are then the prolate spheriodal wavefunctions here denoted by $\phi_n(v, c)$, making explicit the dependence on $c$.

3) The eigenvalues $\lambda_n(c)$ satisfy

$$\lambda_n(c) = e^{i\pi n/2} |\lambda_n(c)|$$

$$|\lambda_n| > |\lambda_{n+1}| > 0$$

4) The singular values are denoted by $\sigma_n(c)$ with

$$\sigma_n(c) \equiv |\lambda_n(c)|$$

They behave as follows

a. For a fixed $c$ and increasing $n$

$$\sigma_n(c) \sim (c^2/n)^{2n}$$

b. For a fixed $n$, there exist constants $\alpha_n, \beta_n$ such that for increasing $c$

$$\sigma_n(c) \sim 1 - \alpha_n e^{-\beta_n c^2}$$

Since $P_c \mathcal{A} P_c$ is compact, it has a singular value decomposition $U, \Sigma, V$ where $U$ and $V$ are complete sets of orthonormal functions, and $\Sigma$ is the decreasing sequence of nonnegative numbers defined in Equation 2.10.

Since $P_c \mathcal{A} P_c$ is normal
\[ U = \{ u_n \phi_n (\omega, c) \}_{n=0}^\infty \]  
\[ V = \{ v_n \phi_n (\omega, c) \}_{n=0}^\infty \]  

(2.13)

where \( u_n \) and \( v_n \) are constants such that \(|u_n| = |v_n| = 1\). With this machinery, the solution to the problem can theoretically be determined by expanding \( \hat{g} \) and \( G \) as

\[ \hat{g}(v) = \sum_{n=0}^{\infty} a_n v_n \phi_n (v, c) \]  
\[ G(\omega) = \sum_{n=0}^{\infty} b_n u_n \phi_n (\omega, c) \]  

(2.14)

(2.15)

The coefficients \( a_n \) are determined directly from the known \( \hat{g} \) and the \( b_n \) from the relation

\[ \hat{g} = P_c \mathcal{F} P_c G \]  

(2.16)

the final result is

\[ b_n = \frac{a_n}{\sigma_n} \]  

(2.17)

In this representation, the degree of ill conditioning of Equation 2.6 (i.e. the measure of how ill posed Equation 2.6 is) as well as the effect of noise can be determined from Equation 2.17 and the behavior of \( \sigma_n \). Intuitively since \( \sigma_n \to 0 \), by Equation 2.12 a small error in the coefficients \( a_n \) of a high frequency (large \( n \)) component \( a_n v_n \phi_n (v, c) \) of \( g(v) \) will induce a large error in the coefficient \( b_n \), given by Equation 2.17, of the corresponding component of \( G(\omega) \). A more precise statement is
Lemma 1  Suppose the relative error (in the $L^2$ norm) on the measured $g$ is $\varepsilon$, and the maximum absolute error that will be tolerated in $G$ is $\delta$. Then there is a decomposition of $G$ into two components $G_1$ and $G_2$, in which $G_1$ is always accurate within an error $\delta$ and $G_2$ cannot be guaranteed to be this accurate.

Proof: Let $G$ be decomposed as the sum in Equation 2.15. For a fixed $N$, the worst possible absolute error in the first $N$ components of $G$ is induced by assuming that $a_1, \ldots, a_{N-1}$ are precisely known and that $a_1 = 1, a_2 = \ldots = a_{N-1} = 0, a_N$ is in error by $\varepsilon$ and $a_i = 0$ for $i > N$. Then

$$b_1 = 1/\sigma_N, \quad b_2 = \ldots = b_{N-1} = 0, \quad b_i = 0 \text{ for } i > N$$

$$b_N = \varepsilon/\sigma_N = b_1 (\varepsilon \sigma_1/\sigma_N)$$

(2.18)

Thus the absolute error induced is $(\varepsilon \sigma_1/\sigma_N)$. If $N$ is chosen so that

$$(\varepsilon \sigma_1/\sigma_N) < \delta \leq (\varepsilon \sigma_1/\sigma_{N+1})$$

(2.19)

then the first $N$ components of $G$ are always known to within an absolute error $\delta$, but the remaining components are not that trustworthy.

But as the decomposition allows a direct observation of the effects of noise, so also does it allow direct action to reduce these effects. A filter $f(\sigma, \varepsilon)$ dependent on the noise levels $\varepsilon, \delta$ is introduced so that $G$ is estimated by $G'$

$$G'(\omega) = \sum_{n=0}^{\infty} a_n f(\sigma_n) \phi_n(\omega, c)$$

(2.20)

where
\[ f(n) + (\sigma_n)^{-1} \quad \text{as} \quad \sigma_n \to 1 \]
\[ \to 0 \quad \text{as} \quad \sigma_n \to 0 \]

(2.21)

In order to construct a sensible filter for the general problem, we first note the following conclusions from Equations 2.8 - 2.12.

a. \( P \mathcal{F} P \) is badly ill conditioned because \( \sigma_n(c) \) exhibits exponential decay in \( n \)

b. For a given \( c \), the singular values are approximately distributed as a step function in that there exists an \( N(c) \) such that

\[ n < N(c) \Rightarrow \sigma_n(c) \sim 1 \]
\[ n > N(c) \Rightarrow \sigma_n(c) \sim 0 \]

(2.22)

c. For a given \( c \) and noise levels \( \varepsilon, \delta \) let \( N(\varepsilon, \delta, c) \) be the \( N \) of Equation 2.19. Then \( N(\varepsilon, \delta, c) \) is bounded by Equation 2.12. Furthermore from conclusions a and b, \( N(\varepsilon, \delta, c) \) is relatively insensitive to \( \varepsilon \) and \( \delta \) (i.e. for almost all \( \varepsilon \) and \( \delta \) there exists a \( p \ll 1 \) such that

\[ N(\varepsilon, \delta, c) - 1 \leq N(\varepsilon, p\delta, c) \leq N(\varepsilon, \delta, c) \]
\[ \leq N(p\varepsilon, \delta, c) \leq N(\varepsilon, \delta, c) + 1 \]

(2.23)
The general choice of filters is an art, however here we make
the particular choice

\[ f(\sigma, \epsilon) = \frac{1}{\sigma}, \quad \sigma > \epsilon/\delta \]

\[ = 0, \quad \sigma < \epsilon/\delta \]  \hspace{1cm} (2.24)

relying on conclusions \( b \) and \( c \) above. This filter effectively truncates
the series in Equation 2.20 leaving only the trustworthy component \( G \), of
Lemma 1. Since the filter is insensitive to \( \epsilon \) and \( \delta \), the number of
elements in the truncated sum of Equation 2.20 is insensitive to noise levels
It corresponds to the "essential number of degrees of freedom" of the model
problem mentioned in section 1. We stress that this number is relatively
noise independent (i.e. a function of \( c \) with weak dependence on \( \epsilon \) and \( \delta \))
and that it represents the number of components of the solution that are of
guaranteed accuracy.

Based on these results, the following algorithm is proposed. For
a given \( c, \epsilon, \) and \( \delta \) calculate \( N(c, \epsilon, \delta) \). If it is sufficiently small,
accurate numerical approximations to \( \sigma_n, \phi_n(v, c) \) are calculated for
\( n \leq N(c, \epsilon, \delta) \) and \( G \) is estimated using Equation 2.20. This procedure is
optimal in that it focuses on accurately calculating components of \( \hat{g} \) that
are significant in estimating \( G \) while ignoring components that are useless
for estimation.
3. DISCRETIZATION AND ASSOCIATED THEORY

In this section we consider a general class of finite dimensional approximations to \( g, G \) and weigh the merits of different approximations. The discretizations considered are derived using Galerkin's method, they are based upon the natural error metric \( \| \cdot \|_2 \) for the model problem. We choose two sets of linearly independent functions on \( L^2[-c,c] \)

\[
\{ \psi_k \}_{k=1}^K, \{ \theta_\lambda \}_{\lambda=1}^L; \quad \| \psi_k \|_2 = \| \theta_\lambda \|_2 = 1
\] (3.1)

and introduce

**Definition 3.** Let \( S_K \) denote the subspace spanned by \( \{ \psi_k \}_{k=1}^K \), \( S_L \) the subspace spanned by \( \{ \theta_\lambda \}_{\lambda=1}^L \); let \( P_K \) denote the projection operator from \( L^2 \) onto \( S_K \), \( P_L \) the projection operator from \( L^2 \) on \( S_L \).

We generate approximations

\[
\hat{g}_K \in S_K + \hat{g}, \quad G_L \in S_L + G
\] (3.2)

by requiring that \( \hat{g}_K, G_L \) minimize

\[
\| \hat{g} - h \|_2, \quad h \in S_K
\] (3.3)

\[
\| \hat{g} - P_K P_L g^H \|_2, \quad H \in S_L
\] (3.4)

Since \( S_K, S_L \) are finite dimensional, such approximations always exist. By expanding \( \hat{g}_K, G_L \) in terms of the basis functions

\[
\hat{g}_K = \sum_{k=1}^K b_k \psi_k, \quad G_L = \sum_{\lambda=1}^L a_\lambda \theta_\lambda
\] (3.5)
and defining the quantities

\[ c_k = \langle \psi_k, \hat{\theta} \rangle, \quad B_{ij} = \langle \psi_i, \psi_j \rangle \]  
\[ A_{ij} = \langle \theta_i, \theta_j \rangle, \quad D_{kl} = \langle \psi_k^P, \Phi_P \theta_l \rangle \]  

then Equations 3.3 and 3.4 can be rewritten as discrete equations for the vectors \( \hat{a} \equiv (a_1, \ldots, a_L)^+ \), \( \hat{b} = (b_1, \ldots, b_K)^+ \).

We note that

\[ \hat{c}_K \text{ minimizes Equation 3.3 } \Rightarrow \hat{B} \hat{b} = \hat{c} \]  
\[ \hat{G}_L \text{ minimizes Equation 3.4 } \Rightarrow \hat{a} \text{ minimizes } \]  
\[ \| \hat{B}^{-1} (\hat{c} - \hat{B} \hat{a}) \| \quad \hat{a} \in R^L \]  

where the matrices \( \hat{B}, \hat{D} \) are defined in Equations 3.6 and 3.7. For typographic convenience, the subscript 2 will be omitted from \( \| \cdot \| \).

In this construction, there is considerable freedom of choice for the subspaces \( S_K, S_L \) and their bases; it is therefore appropriate to adopt criteria by which a particular choice can be judged. We list six criteria for evaluation of approximations and for each criterion list features or bases that produce an approximation giving good results. We can then seek a "best fit" approximation to these features.

1. **Uniqueness of approximation:** Because \( \| \cdot \| \) is a strictly convex norm \( \hat{c}_K \) is unique. If \( K < L \) then \( G_L \) is not unique; if \( K \geq L \) then \( G_L \) is unique unless \( P_C \Phi_P S_L \) contains components orthogonal to \( S_K \).
Alignments of this form are pathological; the practical problem is the occurrence of nearly orthogonal components. These components will contribute to ill conditioning in $D$, however both the pathology and the ill conditioning can be avoided by choosing $\psi_k$ so that

$$P_c \mathcal{F} P_c S_k \subset S_k$$

(3.9)

Because the true solution is unique there appears to be no gain in constructing nonunique approximations. Henceforth, we assume that $K > L$ and that $G_L$ is unique.

2. Accuracy of approximations: An obvious requirement for any sequence of approximations $\hat{F}_K$, $G_L$ is that

$$\lim_{K \to \infty} \| \hat{g} - \hat{g}_K \| = 0$$

(3.10)

and if $g$ is noiseless that

$$\lim_{K,L \to \infty} \| G - G_L \| = 0$$

(3.11)

Necessary conditions for Equations 3.10 and 3.11 to hold are that the sequences \( \{ \psi_k \}_{k=1}^{\infty}, \{ \theta_{L'} \}_{l'=1}^{\infty} \) be complete sets in $L^2[-c,c]$. These conditions are also sufficient for Equation 3.10, but not for Equation 3.11 because the inverse of $P_c \mathcal{F} P_c$ is unbounded.

The important issue for numerical calculations is the rate of convergence of such approximations. Rather than attempt direct estimates of $\| G - G_L \|$ we consider a more easily computed quantity. The approximating subspaces chosen implicitly define an associated finite dimensional operator
The quantity $\|R_{KL} - P_c \mathcal{S} P_c\|$ can then serve as a measure of the accuracy of the approximation.

We therefore seek functions $\psi_k, \theta_k$ such that

$$\hat{\psi}_k, \hat{\theta}_k : G_L \to \hat{\psi}, \hat{\theta}$$

and that minimize this error metric. The following proposition provides a partial answer

**Proposition 1:** If $K = L = N$ and $\psi_k, \theta_k$ are chosen so that $\psi_k(v) = \theta_k(v) = \phi_k(v)$ (the $k$-th eigenfunction of $P_c \mathcal{S} P_c$), then the Galerkin approximation generated by Equations 3.3, 3.4 satisfy Equations 3.10, 3.11 as $N \to \infty$. Furthermore for any other subspaces $S_{K'}, S_{L'}$, with $K', L' \leq N$,

$$\|R_{NN} - P_c \mathcal{S} P_c\| \leq \|R_{K'L'} - P_c \mathcal{S} P_c\|$$

The proof of this and the succeeding two propositions can also be found in [ ]. Thus the eigenfunctions are an "optimal" basis for construction of approximations under this criterion.

3. **Conditioning of $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{D}}$:** The previous criterion aids in distinguishing suitable approximating subspaces $S_K, S_L$. One must now make a choice of bases within these subspaces. It is possible to choose $\psi_k, \theta_k$ so that $\tilde{\mathbf{D}}$ is well conditioned, but this is an illusory gain because the matrices $\tilde{\mathbf{B}}$ and/or $\tilde{\mathbf{A}}$ will then be ill conditioned. Therefore calculations of $\hat{\psi}_k(v), G_L(\omega)$ made by using the representations in Equation 3.5 will be
error prone. Since the most informative measure of ill conditioning in a
metric, the condition number, is not easily calculated; we use a less informe-
tive, but more easily manipulated, measure for comparison of different
bases, the determinant.

Proposition 2: Let $S_K$, $S_L$ be fixed subspaces with orthonormal
bases $\{\psi_k\}_{k=1}^K$, $\{\theta_{\ell}\}_{\ell=1}^L$. If $\{\psi'_{k}\}_{k=1}^K$, $\{\theta'_{\ell}\}_{\ell=1}^L$ are any other bases
for $S_K$, $S_L$, and the corresponding matrices $\tilde{A}$, $\tilde{A}'$, $\tilde{B}$, $\tilde{B}'$, $\tilde{D}$, $\tilde{D}'$ are defined
by Equations 3.5, then

$$|\det(\tilde{B}'\tilde{D}'\tilde{A}')| \leq |\det(\tilde{B}\tilde{D}\tilde{A})| = |\det(\tilde{D})|.$$ \hspace{1cm} (3.15)

Proposition 3: Let $K = L = N$, $\psi_k = \theta_k = \phi_k$. Then for any other
subspaces $S_{K'}$, $S_{L'}$ with $K'$, $L' \geq N$

$$|\det \tilde{D}'| \leq |\det \tilde{D}|.$$ \hspace{1cm} (3.16)

These two propositions demonstrate that under this measure of ill conditioning,
orthonormal bases should be chosen and that the best choice of bases is the eigen-
functions.

4. Desirable special results: Particular choices of bases
yield interesting additional results. If both bases are orthonormal, then
the spectral properties of $\tilde{D}$ are identical with those of $\tilde{R}_{KL}$, which in
turn approximate those of $P_c \tilde{P}_c$. If $K = L$ and $\psi_k, \theta_k$ satisfy

$$\psi_k = P_c \tilde{P}_c \theta_k$$ \hspace{1cm} (3.17)

then the Galerkin approximation is also a least squares solution, i.e. $G_L$
iminizes
\[ \| \hat{g} - P_c \mathcal{H} \| \quad H \in S_L \quad (3.18) \]

Both these results are obtained if \( K = L = N, \quad \psi_k = \theta_k = \phi_k \).

5. Presence of noise: The above criterion are concerned only with properties of the approximations. The important property of the data, that it contains noise, must be considered. In section 2 we showed that it forced a division of the solution into trustworthy and untrustworthy components. The approximations therefore must echo this division as accurately as possible.

6. Ease of computation: In light of the above discussion the obvious choice for a basis are the eigenfunctions, but we do not consider their use as there is no known simple numerical algorithm for their calculation. To keep computing costs down \( \psi_k, \theta_k \) must have a simple closed form representation. Furthermore the functions \( \mathcal{H}P_c \theta_k \) should also be available in a closed form so that calculation of an approximate interpolation \( g_L \) to \( g \)

\[ g_L = \mathcal{H}P_c \theta_L \quad (3.19) \]

is cheap compared to numerical evaluation. If possible the inner products \( (\psi_k, P_c \mathcal{H} \theta_k) \) should also be found explicitly.

Taking all these issues into consideration we seek basis of simply calculated functions \( \{ \rho_k \}_{k=1}^{\infty} \) which are approximations to the eigenfunctions \( \phi_k \); or at least possess, to some degree, the properties discussed in the criteria above. In particular the basis should be complete, orthonormal, and have the property that each eigenfunction can be expressed as a rapidly convergent series in \( \rho_k \).
Given such a basis, we now describe an algorithm that for a given observation \( \hat{\gamma} \) and noise levels \( \varepsilon, \delta \) produces a Galerkin approximation \( G_N \) to a component of \( G \) that is accurate up to an error \( \delta \), as in Lemma 1.

1. \( N, K(N) \) and the subspace \( S_{K(N)} \) are defined by
   a. \( N \) is the \( N(\varepsilon, \delta, c) \) of Equation 2.19
   b. \( S_{K(N)} = \text{span} \{ \phi_k \}_{k=1}^{K(N)} \)
   c. \( K(N) \) is the least integer such that the finite dimensional operator \( R_{KL} \) associated with \( K=L=K(N) \),
      \( S_K = S_L = S_N \) satisfies
      \[ \| R_{K(N)K(N)} \|_{P_c F_c} < \min(\sigma_N - \frac{\varepsilon}{\delta}, \varepsilon) \] (3.20)

2. The \( \tilde{c}, \tilde{\delta} \) of Equation 3.5 are formed for this choice of basis and the singular value decomposition [18]
   \[ \tilde{\delta} = \tilde{\Upsilon} \tilde{\Lambda} \tilde{\Upsilon}^T \] (3.21)
calculated.

3. The subspaces \( S_K = S_L = S_N \) from which the Galerkin approximation \( G_N \) will be formed by Equation 3.4 are the subspaces of \( S_{K(N)} \), being the span of the first \( N \) eigenfunctions of \( R_{K(N)K(N)} \), i.e.
   \[ \psi_k(v) = \theta_k(v) = \sum_{i=1}^{K(N)} v_{ik} \rho_i(v), \quad k = 1, \ldots, N \] (3.22)
4. The approximant $\hat{g}_N$ to $\hat{g}$ from $S_N$ can be formed from the approximation $\hat{g}_{K(N)}$ to $\hat{g}$ in $S_{K(N)}$ (i.e. the vector $\tilde{c}$).

Moreover $G_N$ can be formed directly from $\tilde{c}$ as

$$G_N(\omega) = \sum_{k=1}^{K(N)} a_k c_k(\omega)$$

with

$$\tilde{a} = \tilde{V} \Sigma^+ \tilde{U}^+ \tilde{c}$$

where

$$\Sigma^+_{kl} = \Sigma_{kl}^{-1}, \text{ if } \Sigma_{kl} > \frac{\epsilon}{\delta}$$

$$= 0, \text{ otherwise}$$

5. The approximate extrapolation $g_N$ to $g$ is now defined as

$$g_N(\nu) = \sum_{k=1}^{K(N)} a_k \mathcal{F}_c \rho_k(\nu)$$

The algorithm may also be used to solve the inversion of noiseless observations $\hat{g}$ by choosing sequences $\epsilon_k$, $\delta_k$ such that $\epsilon_k/\delta_k \to 0$ and calculating for each $\epsilon_k$, $\delta_k$ an approximation $G_{N(k)}$ by the above algorithm. Then in the limit $G_{N(k)}$ converges to the true $G$.

The construction of an algorithm for the inversion of Equation 2.6 is now complete. The salient features are: use of the singular value decomposition of $P_c \mathcal{F}_c P_c$, prior estimation of the number of degrees of freedom $N(\epsilon, \delta, c)$ present at given noise levels, and construction of a Galerkin approximation from a predetermined set of functions of a simple closed form that approximate the first $N(\epsilon, \delta, c)$ eigenfunctions.
A final note: \( g \) cannot be approximated directly from the subspace spanned by \( \{ \mathcal{F}_v^\alpha \}_{v,k=1}^{K(N)} \), as is often done with

\[
\mathcal{F}_c^\alpha_k(v) = \frac{\sin(v + k\pi/c)}{(v + k\pi/c)}
\]  

(3.27)

The untrustworthy components of the basis must first be filtered out by singular value decomposition before the extrapolation is done.
4. ITERATIVE SOLUTIONS AND ASSOCIATED THEORY

In contrast to the direct algorithm of the previous section we now consider iterative algorithms and integrate them into the body of theory already built up. We start with the original iterative algorithm, that of Gerschberg and Saxton [1]; a two-step iteration that updates the approximation in both physical and frequency space.

The formal statement of the algorithm applied to the model problem is:

a. Initial conditions

\[ g_0 = G_0 = 0 \]  

(4.1)

b. Update

\[ g_{n+1} = P_c \mathcal{F}_c G_n + \hat{g} \]  

(4.2)

\[ G_{n+1} = P_c \mathcal{F}_c^{-1} g_{n+1} \]  

(4.3)

where \( \mathcal{C} \) is the set \( \mathbb{R} - \mathcal{C} \). An alternative, but more conventional, form is easily obtained. Concatenating the iterations above yields

\[ G_{n+1} = P_c \mathcal{F}_c^{-1} (P_c \mathcal{F}_c G_n + \hat{g}) = (I - P_c \mathcal{F}_c^{-1} P_c \mathcal{F}_c) G_n + P_c \mathcal{F}_c^{-1} P_c \hat{g} \]  

(4.4)

upon noting that

\[ P_c G_n = G_n \quad , \quad P_c \hat{g} = \hat{g} \]  

(4.5)

\[ P_c + P_c = I \quad (\text{the identity operator}) \]  

(4.6)
Equation 4.4 has a computational advantage over Equations 4.2, 4.3 because it requires evaluation of functions in the bounded set $C$, instead of the infinite domain $\overline{C}$, during the iteration.

Equation 4.4 is recognizable as the simplest form of iterative solution to the normal equations associated with Equation 2.6, that is

$$P_c \mathcal{F}^{-1} P_c \mathcal{F} P G = P_c \mathcal{F}^{-1} P_c \mathcal{H}$$

(4.7)

The normal equations are not usually formed for ill conditioned systems since they have even poorer conditioning; if $P_c \mathcal{F} P_c$ has singular values $\sigma_n$, then $P_c \mathcal{F}^{-1} P_c \mathcal{F} P_c$ has singular values $\sigma_n^2$. However since the singular values display the step like behavior, quoted in Equation 2.23, for almost all significantly non zero singular values $\sigma_n \approx \sigma_n^2 \approx 1$. Consequently little is lost by consideration of Equation 4.7.

In the presence of noise, the iteration based on the update Equation 4.4 is usually modified so that either it stops when

$$\| G_n - G_{n+1} \| \leq \delta$$

(4.8)

where $\delta$ depends on the noise level; or the damped update

$$G_{n+1} = (1 - \lambda) G_n - P_c \mathcal{F}^{-1} P_c \mathcal{F} P G_n + P_c \mathcal{F}^{-1} P_c \mathcal{H}$$

(4.9)

is used, where $\lambda > 0$ depends on the noise level. The update, Equation 4.9, if iterated to completion corresponds to a choice of filter

$$g(\sigma) = \frac{\sigma}{\sigma^2 + \lambda}$$

(4.10)
in Equation 2.20. However iteration to convergence is impossible in practice, and as yet there is no good theory linking termination after \( N \) steps with accuracy of approximation or noise levels. Therefore the clear insights singular value decomposition gave in modifying direct algorithms to take account of noise have no analog for iterative algorithms.

There is a large literature on alternative algorithms for iterative solutions of Equations 2.6 and 4.7, eg. [22]. The most attractive of these is the conjugate gradient method. This algorithm uses the residuals

\[
r_n = p_c \hat{F}_c^{-1} p_c \hat{G} - p_c \hat{F}_c^{-1} p_c \hat{F}_c p_c G_n
\]  

(4.11)

in the following iteration scheme

a. Initial conditions

\[
p_0 = G_0 = 0
\]  

(4.12)

\[
 r_0 = p_c \hat{F}_c^{-1} p_c \hat{G}
\]  

(4.13)

b. Update

\[
 G_{n+1} = G_n + \alpha_n p_n
\]  

(4.14)

\[
 p_{n+1} = r_{n+1} + \beta_n p_n
\]  

(4.15)

where

\[
 \alpha_n = \frac{\|r_n\|^2}{\|p_c \hat{F}_c p_c p_n\|^2}, \quad \beta_n = \frac{\|r_{n+1}\|^2}{\|r_n\|^2}
\]  

(4.16)

The conjugate gradient method stands midway between direct and iterative methods since it can be shown that the iterates \( G_N \) are also
Galerkin approximations for Equation 4.7 formed by taking \( K = L = N \), 
\[ S_K = S_L = \text{span} \{ r_k \}_{k=1}^N. \]
Although these approximations satisfy many of the criteria of section 3 (the \( r_k \) are orthogonal, \( G_N \rightarrow G \) if \( G \) exists), an a priori estimate of amount of computation to achieve a desired accuracy cannot be made, nor can the iteration be modified in the presence of noise as is done for the direct methods.

If the iterative algorithms are run on a digital computer, a basis is implicitly used to represent the iterates. The only reason for not directly using this basis to form Galerkin approximations is cost. With present day computers, direct solutions are faster than iterative solutions to linear systems for all but the largest scale problems. This fact, combined with the theoretical results on the effective finite dimension of the model problem and a need for careful treatment of noise lead us to the conclusion that iterative algorithms are not the method of choice for this problem.
5. ILLUSTRATIVE NUMERICAL CALCULATIONS

As an illustration of the results obtainable from the previous analysis, we present numerical solutions to two particular problems. The first is that of discriminating between possible bases for representation of \( g, G \). The second is reconstruction of two particular \( G(\omega) \) of interest in diffraction optics from noisy observations \( \hat{g}(v) \) using the algorithm of Section 3.

Three commonly used bases for reconstruction are:

\[
\rho^1_k(v) = \alpha_k P_k(v/c) \quad , \quad k = 0,1,...,N \quad (5.1)
\]

\[
\rho^2_k(v) = \beta_k \cos(k\pi v/2c) \quad , \quad k = \text{even} \quad k = 0,1,...,N \quad (5.2)
\]

\[
= \beta_k \sin((k+1)\pi v/2c) \quad , \quad k = \text{odd}
\]

\[
\rho^3_k(v) = \gamma_k \quad , \quad v \in \left[ \frac{2ck - (N+1)c}{(N+1)}, \frac{2c(k+1) - (N+1)c}{(N+1)} \right], \quad k = 0,1,...,N
\]

\[
= 0 \quad , \quad \text{elsewhere} \quad (5.3)
\]

Here \( P_k(x) \) is the \( k \)th Legendre polynomial, and \( \alpha_k, \beta_k, \gamma_k \) are constants chosen so that the various \( \rho_k \) are orthonormal. All three bases may be seen as the initial segment of a complete basis for \( L^2[-c,c] \). Van Buren [23] used \( \rho^1_k \) to give accurate representations of the prolate spheroidal wave functions \( \phi_k \), the \( \rho^2_k \) correspond to the sinc function basis of Equation 3.27.
used widely in optics \cite{24,25}, and the $\rho_k^3$ are the familiar piecewise constant functions of numerical analysis (often used implicitly in algorithms based on sampled point values of $\hat{g}(v)$).

If a basis $\{\rho_k\}$ is used to generate approximate reconstructions via the algorithm of section 3, condition 1.c provides a natural merit function for the basis.

1. For fixed $\varepsilon$ and $c$, the merit of the basis is the size of the least integer $K \equiv K(\varepsilon,c)$ such that

$$\|R_{K\varepsilon} - P_c \mathcal{H} P_c\| < \varepsilon$$

(5.4)

However, as estimation of the norm in Equation 5.4 is too expensive for repeated calculation, we choose a slightly different merit function.

2. For fixed $\varepsilon$, $c$ the merit of the basis is the size of the least integer $L \equiv L(\varepsilon,c)$ such that

$$|\sigma_n - s_n| < \varepsilon$$

for all $n$

(5.5)

where $s_n$ are the singular values of $R_{LL}$.

Since the $\rho_k$ are orthonormal, the $s_n$ are also the singular values of the matrix $\tilde{D}$. This, together with knowledge of $\sigma_n$, allows calculation of $L(\varepsilon,c)$ for varying $\rho_k$, $\varepsilon$ and $c$. 

86
To rank the above bases, their $L(E, c)$ were calculated for $c = .5, 1.0, 1.5, 2.0$ and $\varepsilon = .002, .02$. The results appear in Tables 1 and 2. $N(E, c)$ denotes the number of $\sigma_n$ greater than $\varepsilon$; $L^i(E, c)$, the $L(E, c)$ of basis $\{p^i_k\}$. Because it is overly costly to consider every value of $N$, the table entries give an interval containing $N$ rather than the exact result. The singular value decompositions of the matrices $D$ were evaluated using the subroutine LSDVF from the IMSL library.

The tables indicate that $\{p^1_k\}$ has by far the highest merit rating. However, the complexity of programming and calculating the Legendre polynomials and their finite Fourier transforms the spherical Bessel functions $j_n(y)$, [26]

$$\int_{-1}^{1} p_k(x) e^{iyx} \, dx = 2i^k j_k(y) \quad (5.6)$$

affects this rating. We feel their use is worthwhile only for large $c$ or high accuracy computations. The $p^2_k$ are, surprisingly, the worst of the basis functions; it requires a very large number of such functions to approximate closely the higher order eigenfunctions $\phi_{2k+1}$. We believe this is due to $\sin[(k+1)\pi v/2c]$ vanishing at the endpoints $v = \mp c$; however, $\phi_{2k+1}(\pm c) \neq 0$ consequently the $\phi^2_{2k+1}$ produce poor approximations in this region. The same problem appears to a lesser degree in approximating even eigenfunctions $\phi_{2k}$; at the endpoints the derivatives of $\cos(k\pi v/c)$ vanish whereas $d\phi_{2k}(\pm c)/dv \neq 0$. The low rating of the $p^2_k$ marks them as a poor choice for reconstruction and imply the sinc functions should not be used for extrapolation.

The basis of choice appears to be $\{p^3_k\}$, combining adequate approximation power with ease of computation. The basis also has natural extensions.
to other problems. For greater accuracy, higher order splines may be preferable in use to \( \{ \rho_k \} \); for reconstruction problems in higher dimensions over arbitrarily shaped support sets \( A, B \) the basis extends to give the usual finite element type approximations.

These results on Galerkin approximations serve as a guide to construction of good discretizations based on pointwise quadrature rules. As an illustration we present some results on a discretization of Equation (2.5) on Gaussian quadrature. Let \( \{ p_i \}_{i=1}^N \) be the abscissae of the \( N \) point Gaussian quadrature scheme on \([-c,c]\) with associated weights \( \{ \omega_i \}_{i=1}^N \). Then Equation (2.5) may be approximated by

\[
\hat{g} = \hat{F} \hat{W} \hat{G}
\]  

(5.7)

where \( \hat{F} \) is an \( N \times N \) matrix with entries

\[
\hat{F}_{kl} = e^{2\pi i p_k p_l / c},
\]  

(5.8)

\( \hat{W} \) a diagonal matrix with entries \( \hat{W}_{kk} = \omega_k \) and \( \hat{g} \) and \( \hat{G} \) are vectors whose \( k \)-th entries are (hopefully) good approximations to \( g(p_k) \) and \( G(p_k) \).

However in this particular discretization the Euclidean norms of the vectors \( \hat{g} \) and \( \hat{G} \) bear little relation to the \( L^2 \) norms of \( g(v) \) and \( G(\omega) \) and calculations show that the singular values of \( \hat{F} \hat{W} \) are not good approximations of those of \( \hat{F} \hat{G} \). Therefore we replace Equation (5.7) by the system

\[
(\hat{W}^{1/2} \hat{g}) = (\hat{W}^{1/2} \hat{F} \hat{W}^{1/2} \hat{G}) (\hat{W}^{1/2} \hat{G})
\]  

(5.9)

Now the Euclidean norms of \( \hat{g} = \hat{W}^{1/2} \hat{g} \) and \( \hat{G} = \hat{W}^{1/2} \hat{G} \) should coincide with
the $L^2$ norms of $g(v)$ and $G(\omega)$ and the singular values of
$F \hat{=}(W^{1/2} \cdot W)^{1/2}$ be close to those of $P_c \mathcal{F}P_c$.

We have calculated the $L(\varepsilon, c)$ of this approximation for the same
values of $\varepsilon$ and $c$ used above, the results appear in Table 3. It appears
that Gaussian quadrature is on a par with Galerkin approximations by Legendre
polynomials as an efficient discretization; this is only to be expected given
the role Legendre polynomials play in Gaussian quadrature. However in general
Galerkin approximations seem preferable since they must be used anyway to
choose among the many possible discretizations associated with a given
quadrature rule, and to perform interpolation within the interval or extra-
polation beyond it.

To test the actual inversion of noisy data, the algorithm of section 3,
with $\{\rho^2_k\}$ as basis functions, was used to invert the following data
corrupted with random noise:

$$g_1(v) = \frac{\sin 2\pi v}{\pi v}, \quad |v| \leq 1$$

(5.10)

$$g_2(v) = 2\left(\frac{\sin \pi v}{\pi v}\right)^2, \quad |v| \leq 1$$

(5.11)

These functions can be interpreted as the point spread functions ($\equiv$ diffraction
images) caused by an aberration-free slit aperture in coherent light and in-
coherent light respectively. The true solutions to the inversion problem are:

$$G_1(\omega) = 1, \quad |\omega| \leq 1$$

(5.12)

$$= 0, \quad |\omega| > 1$$
\[ G_2(\omega) = 2(1 - |\omega|), \quad |\omega| < 1 \]
\[ = 0 \quad \quad \quad |\omega| > 1 \quad \quad \quad (5.13) \]

The G's are the corresponding optical transfer functions in the diffraction imagery interpretation. In both cases \( c = 1 \). The calculated inversions were compared to the true \( G_1(\omega) \) on \( |\omega| < 1 \), and the approximate extrapolations of Equation (3.26) were compared to the true \( q_1(v) \) on \( |v| < 4 \).

We first applied the algorithm to noiseless data, choosing \( \delta = 1, \epsilon = 10^{-2} \) and \( K(N) = 40 \) for \( \hat{g}_1 \), \( K(N) = 80 \) for \( \hat{g}_2 \) (in both cases \( \| P_{K(N)} K(N) - P_{C} \|_F < \epsilon \)). The resulting approximations to \( G_1(\omega) \) coincides with the true \( G_1(\omega) \) to well within graphical accuracy. The approximate extrapolation to \( q_1(v) \) also coincided to the true \( q_1(v) \) to within graphical accuracy. The approximate extrapolation to \( q_2(v) \) is plotted in Figure 1, the solid line is the true \( q_2(v) \).

To simulate the presence of noise in the data, whenever a value of \( g(v) \) was required in the numerical integrations used to calculate the components \( c_k = (g, \rho_k) \), a random perturbation was added. That is

\[ g_\epsilon(v) = g(v) + \epsilon \mu \max_{v \in C} |\hat{g}(v)| \quad (5.14) \]

was used in place of \( g(v) \). The constant \( \epsilon \) denotes the noise level, \( \mu \) is a random variable uniformly distributed over \([-1,1]\). For this artificial noise, the error level \( \epsilon \) of the algorithm was taken to be the \( \epsilon \) in Equation (2.8). Since we had no prior expectations on the accuracy of the inversions, the error tolerance \( \delta \) of the algorithm was taken to be unity.
To test the robustness of the algorithm, reconstruction of data with different random perturbations of different noise levels were made. The same $K(N)$ as in the noiseless case were used. Table 4 lists four reconstructions of $G_1(\omega)$ from data with 1% (i.e., $\epsilon = .01$) noise level; there is approximately a 5% maximum deviation from the true value. Figure 2 shows four reconstructions of $G_1(\omega)$ from data with a 3% noise level; there is now a 25% maximum deviation from the true value. Reconstructions from data with 5% noise levels were still recognizable; but at 10% noise levels the error (measured in the energy norm) in the reconstruction exceeded 50%. Figures 3 and 4 show (see open circles) the extrapolation of two sample reconstructions from Figure 1. For comparison the true $g_1(v)$ is also shown as the solid line. Because $G_1$ and $g_1$ are even functions of their arguments, the graphs have been plotted only for the negative values of the respective arguments.

Figure 5 and 6 show four realizations of $G_2(\omega)$ from data with 3% noise level. Extrapolations of three of these reconstructions appear in Figures 7-9, with the true $g_2(v)$ shown as the solid line. Although the reconstructions of $G_2(\omega)$ appear to be more accurate than those of $G_1(\omega)$, the extrapolation to $g_2(v)$ is less accurate than those to $g_1(v)$. This is due to the discontinuity in slope of $G_2(\omega)$ at the origin. Since the approximations to $G_2(\omega)$ are sums of small numbers of smooth eigenfunctions, they do not approximate $G_2(0)$ well as the graphs indicate. This low frequency error in the $\omega$ domain is then transformed into high frequency errors in the $v$ domain, resulting in errors in the extrapolation.
The analysis used in defining the "trustworthy" component of \( G \), and a trustworthy Galerkin approximation, is worst case in nature. It is therefore reasonable to supply the algorithm with smaller \( \varepsilon \) or larger \( \delta \) than a priori noise levels suggest (as was done here) or use a different filter in Equation (3.25). However this cannot be made precise without some statistical statements on expected errors.
ACKNOWLEDGMENTS

We are indebted to Professor D.G.M. Anderson for helpful criticisms of this work. Barakat was supported in part by RADC, Newsam was supported in part by NSF.
REFERENCES


TABLE 1: Distribution of singular values, and required basis size for desired accuracy when \( \epsilon = .02 \)

<table>
<thead>
<tr>
<th>( c )</th>
<th>( N(\epsilon,c) )</th>
<th>( L^1(\epsilon,c) )</th>
<th>( L^2(\epsilon,c) )</th>
<th>( L^3(\epsilon,c) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>3</td>
<td>3</td>
<td>20(-30)</td>
<td>3(-10)</td>
</tr>
<tr>
<td>1.0</td>
<td>7</td>
<td>7(-9)</td>
<td>40(-50)</td>
<td>10(-20)</td>
</tr>
<tr>
<td>1.5</td>
<td>13</td>
<td>13(-15)</td>
<td>&gt;80</td>
<td>20(-40)</td>
</tr>
<tr>
<td>2.0</td>
<td>20</td>
<td>23(-25)</td>
<td>&gt;80</td>
<td>40(-80)</td>
</tr>
</tbody>
</table>
TABLE 2: Distribution of singular values, and required basis size for desired accuracy when $\varepsilon = .002$

<table>
<thead>
<tr>
<th>$c$</th>
<th>$N(\varepsilon, c)$</th>
<th>$L^1(\varepsilon, c)$</th>
<th>$L^2(\varepsilon, c)$</th>
<th>$L^3(\varepsilon, c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>4</td>
<td>5 - 7</td>
<td>&gt;80</td>
<td>10 - 20</td>
</tr>
<tr>
<td>1.0</td>
<td>9</td>
<td>9 - 11</td>
<td>&gt;80</td>
<td>40 - 80</td>
</tr>
<tr>
<td>1.5</td>
<td>15</td>
<td>19 - 23</td>
<td>&gt;80</td>
<td>&gt;80</td>
</tr>
<tr>
<td>2.0</td>
<td>22</td>
<td>31 - 35</td>
<td>&gt;80</td>
<td>&gt;80</td>
</tr>
</tbody>
</table>
TABLE 3. Distribution of $L(\epsilon, c)$ for approximations based on Gaussian quadrature.

<table>
<thead>
<tr>
<th>$c$</th>
<th>$L(0.02, c)$</th>
<th>$L(0.002, c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>4-8</td>
<td>4-8</td>
</tr>
<tr>
<td>1.0</td>
<td>8-12</td>
<td>8-12</td>
</tr>
<tr>
<td>1.5</td>
<td>14-18</td>
<td>18-22</td>
</tr>
<tr>
<td>2.0</td>
<td>30-34</td>
<td>30-34</td>
</tr>
</tbody>
</table>
TABLE 4: Four Sample Realizations of Reconstructions with 1% Noise.

<table>
<thead>
<tr>
<th>$\omega$</th>
<th>$G(\omega)$</th>
<th>$G(\omega)$</th>
<th>$G(\omega)$</th>
<th>$G(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.0000</td>
<td>0.983</td>
<td>0.973</td>
<td>1.099</td>
<td>0.907</td>
</tr>
<tr>
<td>-0.9474</td>
<td>1.009</td>
<td>1.010</td>
<td>1.043</td>
<td>1.005</td>
</tr>
<tr>
<td>-0.8947</td>
<td>1.013</td>
<td>1.021</td>
<td>0.995</td>
<td>1.046</td>
</tr>
<tr>
<td>-0.8421</td>
<td>1.004</td>
<td>1.017</td>
<td>0.962</td>
<td>1.050</td>
</tr>
<tr>
<td>-0.7895</td>
<td>0.991</td>
<td>1.007</td>
<td>0.944</td>
<td>1.035</td>
</tr>
<tr>
<td>-0.7368</td>
<td>0.980</td>
<td>0.997</td>
<td>0.942</td>
<td>1.013</td>
</tr>
<tr>
<td>-0.6842</td>
<td>0.974</td>
<td>0.990</td>
<td>0.952</td>
<td>0.993</td>
</tr>
<tr>
<td>-0.6316</td>
<td>0.975</td>
<td>0.989</td>
<td>0.971</td>
<td>0.980</td>
</tr>
<tr>
<td>-0.5789</td>
<td>0.980</td>
<td>0.993</td>
<td>0.995</td>
<td>0.976</td>
</tr>
<tr>
<td>-0.5263</td>
<td>0.988</td>
<td>1.000</td>
<td>1.017</td>
<td>0.979</td>
</tr>
<tr>
<td>-0.4737</td>
<td>0.999</td>
<td>1.008</td>
<td>1.035</td>
<td>0.988</td>
</tr>
<tr>
<td>-0.4211</td>
<td>1.007</td>
<td>1.015</td>
<td>1.046</td>
<td>0.998</td>
</tr>
<tr>
<td>-0.3684</td>
<td>1.013</td>
<td>1.019</td>
<td>1.048</td>
<td>1.007</td>
</tr>
<tr>
<td>-0.3158</td>
<td>1.004</td>
<td>1.019</td>
<td>1.042</td>
<td>1.012</td>
</tr>
<tr>
<td>-0.2632</td>
<td>1.011</td>
<td>1.015</td>
<td>1.028</td>
<td>1.014</td>
</tr>
<tr>
<td>-0.2105</td>
<td>1.005</td>
<td>1.008</td>
<td>1.008</td>
<td>1.012</td>
</tr>
<tr>
<td>-0.1579</td>
<td>0.997</td>
<td>0.999</td>
<td>0.987</td>
<td>1.007</td>
</tr>
<tr>
<td>-0.1053</td>
<td>0.988</td>
<td>0.990</td>
<td>0.968</td>
<td>1.001</td>
</tr>
<tr>
<td>-0.0526</td>
<td>0.982</td>
<td>0.983</td>
<td>0.953</td>
<td>0.996</td>
</tr>
<tr>
<td>0</td>
<td>0.978</td>
<td>0.979</td>
<td>0.945</td>
<td>0.993</td>
</tr>
</tbody>
</table>
Figure 1: Extrapolation of \( g_2(v) \), see open circles, corresponding to a noiseless situation.
Figure 2: Four sample realizations of the reconstruction of $G_1(\omega)$ in the presence of 3% noise in $\hat{g}(v)$. 
Figure 3: Sample realization extrapolation of \( g_1(v) \) corresponding to solid line in upper half of Figure 2.
Figure 4: Sample realization extrapolation of $g_1(v)$ corresponding to dashed line in upper half of Figure 2.
Figure 5: Two sample realizations (--- and ---) of the reconstruction of $G_2(\omega)$ in the presence of 3% noise in $\bar{G}(v)$. 
Figure 6: Two sample realizations (— • — and ——) of the reconstruction of $G_2(\omega)$ in the presence of 3% noise in $\hat{g}(v)$. 
Figure 7: Sample realization extrapolation of $g_2(v)$ corresponding to --- line in Figure 5.
Figure 8: Sample realization extrapolation of $g_2(v)$ corresponding to $-\cdots-\cdots$ line in Figure 5.
Figure 9: Sample realization extrapolation of $g_2(v)$ corresponding to \_\_\_\_\_\_\_\_ line in Figure 6.
ALGORITHMS FOR RECONSTRUCTION OF PARTIALLY KNOWN, 
BANDLIMITED FOURIER TRANSFORM PAIRS FROM NOISY DATA:

II THE NONLINEAR PROBLEM OF PHASE RETRIEVAL
ABSTRACT

Phase retrieval problems are ill-posed, however previous analysis has focused upon global ill conditioning due to the existence of multiple exact solutions. In this paper we consider the effects of local ill conditioning due to the presence of large infinite dimensional neighborhoods of any exact solution where members are all possible solutions provided there is any uncertainty in the data. The form of such neighborhoods can be characterized by viewing phase retrieval as a nonlinear extension of the linear problem of inversion of the finite Fourier transform considered in the previous companion to the present work. In particular, we are able to estimate the essential dimension of phase retrieval problems, i.e. the number of parameters in a solution representation that can be determined accurately given specified error levels in the data. Based on these results a modification of the widely used Gerschberg-Saxton algorithm is proposed, analyzed, and then used as a basis for development of more sophisticated algorithms. Numerical results are presented on the performance of these algorithms on one-dimensional problems. The results indicate that although the algorithms may be tuned to overcome local ill conditioning, good solutions in one dimension are still difficult to find numerically because of global ill conditioning. However, the material in the Appendix indicates that in higher dimensions global ill conditioning is considerably reduced so that the algorithms should be effective in such problems.
1. INTRODUCTION

In our companion paper [1], we introduced the problem of numerical reconstruction of a partially known, bandlimited Fourier transform pair \( g, G \) and considered in detail the model linear problem:

Given values over a finite interval \([a_1, a_2]\) of a function \( g(v) \) known to be the Fourier transform of a function \( G(\omega) \) with support contained in the finite interval \([b_1, b_2]\), find \( G(M) \).

The paper focused on the ill posed nature of this problem and its effects on numerical solution, an algorithm (filtered singular-value decomposition) was presented that could be specially tailored to cope with these effects. In this paper we consider the associated model nonlinear problem of phase retrieval.

The model problem in phase retrieval is:

Given value \( m(v) \) over a finite interval \([a_1, a_2]\) of the modulus of a function \( g(v) \) known to be the Fourier transform of a function \( G(\omega) \) with support contained in the finite interval \([b_1, b_2]\) and the knowledge that \( G \) is a member of some set \( B \), find \( g(v) \) and \( G(\omega) \).

The set \( B \) represents available prior knowledge on \( G \). In this paper we study three particular cases: no prior knowledge, \( G \) is nonnegative, and \( |G(\omega)| = n(\omega) \) is known over \([b_1, b_2]\). Again our focus is on the ill posed nature of phase retrieval and the resulting implications for numerical reconstructions. In particular we generalize the iterated projection algorithm of Gerschberg and Saxton [2] in such a fashion so that it may be identified
as a first order, gradient type optimization algorithm. We then propose second order, Newton type extensions of the algorithm and show that the linear problem arising at each step has behaviour dominated by the behaviour of the linear problem [1].

The structure of the paper is as follows. Section 2 sets out the notation employed in the paper and lists the particular model problems used in illustrative numerical calculations. Section 3 contains a discussion of the ill conditioning of phase retrieval, focusing on the two causes: possible multiple solutions and ill conditioning of the underlying linear operator (the finite Fourier transform). In the Appendix we show that although in one dimension multiplicity of solutions presents a serious problem, in two or more dimensions its occurrence is pathologically rare. Section 4 is devoted to a generalization of the Gerschberg-Saxton algorithm through restatement of the phase retrieval problem as one of finding the closest point to several, possibly disjoint, sets. Two particular algorithms are developed that use iterated projections; their convergence properties are investigated and their identification with the well known steepest descent algorithm is established. In Section 5 a number of second order algorithms are developed from those of Section 4. Each algorithm requires the solution of an ill posed linear problem at each iteration; we propose that this solution be found by filtered singular value decomposition. In order to save on the cost of an expensive decomposition calculation at each step, we indicate how a partial filtering can be obtained using a block decomposition of the linear system and a precalculated singular value decomposition of the
finite Fourier transform (fFT) that is constant across iterations. Finally, Section 6 contains numerical results showing the application of these algorithms to each of the model phase retrieval problem. Tables showing the relative performance of the various algorithms and graphs of the approximate solutions generated are presented and discussed.

The results indicate that in one dimension the multiplicity of solutions induces poor global conditioning so that the significant improvement of second order methods over first order methods is restricted to small local neighborhoods of the solution. Therefore the simplest algorithms appear to be the most cost effective, however we feel that in higher dimensions, with the increased likelihood of unique solutions, that higher order methods will come into their own.
2. SOME MODEL PROBLEMS IN PHASE RETRIEVAL

The analysis of this paper is set in the function space $L^2(\mathbb{R})$ with the standard inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. The Fourier transform $\mathcal{F}: L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$ is here defined to be

$$g(v) = (\mathcal{F}g)(\omega) = \int_{-\infty}^{\infty} e^{2\pi i v \omega} G(\omega) \, d\omega.$$  

(2.1)

Subsets of $L^2(\mathbb{R})$ will be denoted by $A_i$ or $B_j$, and the projection operator $P_A: L^2(\mathbb{R}) \rightarrow A$ onto a particular set $A$ is defined by

$$P_A f = g \iff \| f - g \| = \min_{h \in A} \| f - h \|$$

(2.2)

$P_A f$ is termed the projection of $f$ on $A$, projections are assumed to exist and be unique.

The interval $[-1,1]$ will be denoted by $I$ and $[-c,c]$ by $cI$; $cI$ will also be used to denote the subset of $L^2(\mathbb{R})$ of all functions whose support is contained in $[-c,c]$. The associated projection operator is therefore

$$(P_{cI} f)(v) = f(v), \quad \text{if } v \in [-c,c]$$

$$= 0, \quad \text{if otherwise}.$$  

(2.3)

It will be abbreviated to $P_c$ for typographic convenience.

Three model problems, variations of phase retrieval that occur in physical problems, will be used as numerical illustrations in the remaining sections. They are:
I. Given values $m(v)$ over the interval $[a_1, a_2]$ of the modulus of a function $g(v)$ known to be the Fourier transform of a function $G(\omega)$ whose support is contained in the interval $[b_1, b_2]$, find $g(v)$ and $G(\omega)$. Some relevant references are [2,3].

II. Given values $m(v)$ over the interval $[a_1, a_2]$ of the modulus of a function $g(v)$ known to be the Fourier transform of a real, nonnegative function $G(\omega)$ whose support is contained in the interval $[b_1, b_2]$, find $g(v)$ and $G(\omega)$. Some relevant references are [4,5].

III. Given values $m(v)$ over the interval $[a_1, a_2]$ of the modulus of a function $g(v)$ known to be the Fourier transform of a function $G(\omega)$ whose support is contained in the interval $[b_1, b_2]$ and whose modulus $n(\omega)$ is given over $[b_1, b_2]$, find $g(v)$ and $G(\omega)$. Some relevant references are [6,7].

These two moduli $m(v)$ and $n(\omega)$ may only be known to within some accuracy $\epsilon$, that is measurements $\tilde{m}$ and $\tilde{n}$ are available such that

$$\|m - \tilde{m}\|, \quad \|n - \tilde{n}\| \leq \epsilon .$$

(2.4)

In this setting all problems and errors are invariant under translation and scaling; so, as in the previous paper [1], the sets $[a_1, a_2], [b_1, b_2]$ can be transformed into $cI$ where

$$c = \frac{1}{2} \left[ (a_2 - a_1)(b_2 - b_1) \right]^{1/2} .$$

(2.5)

Upon defining the sets
\[
A \equiv \{ g : \|(P_v g)(v)\| = (P_v m)(v) \} \quad (2.6a)
\]
\[
B_1 \equiv cI \quad (2.6b)
\]
\[
B_2 \equiv cI \cap \{ G : G \geq 0 \} \quad (2.6c)
\]
\[
B_3 \equiv cI \cap \{ G : \|(P_v G)(\omega)\| = (P_v n)(\omega) \} \quad (2.6d)
\]

the i-th problem \((i = 1, 2, 3)\) can be rewritten as:

Find \((g, G)\) such that \(g = G, g \in A, G \in B_1\) . \quad (2.7)
3. THE ILL-POSED NATURE OF PHASE RETRIEVAL

In the previous paper we indicated that the phase retrieval problem is, in general, ill-posed in that it fails to satisfy Hadamard's definition of a well-posed problem (the definition is repeated here for convenience).

Definition: A problem is well-posed if the solution
a) exists
b) is unique
c) depends continuously on the data.
If any of these conditions are violated it is ill-posed.

We now consider the ill-posed and ill conditioned nature of the phase retrieval problem in more detail with particular regard as to the consequences for numerical solutions. We wish to show that it is a failure of condition c that is the main source of difficulty.

To illustrate the point we briefly review the prototypical linear problem and its solution as described in the previous paper, the inversion of a compact linear operator by use of its filtered singular value decomposition (SVD). Let \( H \) be a separable Hilbert space and \( \mathcal{K}:H \rightarrow H \) be a compact linear operator with an SVD\( \{ \phi_i, \sigma_i, \psi_i \} \). If \( g \) is an arbitrary member of \( H \) with the expansion

\[
g = \sum_{i=0}^{\infty} a_i \psi_i , \quad \sum_{i=0}^{\infty} |a_i|^2 < \infty
\]  

(3.1)

then the equation \( \mathcal{K}g = g \) has the formal solution
The solution $G$ is in $H$ if and only if 
$$\sum_{i=0}^{\infty} \left| \frac{a_i}{\sigma_i} \right|^2 < \infty.$$ 
Because the singular values $\sigma_i$ tend to zero this is a stronger condition on the coefficients $a_i$ than Eq. (3.1), so a solution does not exist for every right hand side $g$. Thus the problem is ill-posed with respect to existence of solutions. The solution is unique if and only if $\mathcal{H}$ has a non-trivial null space, which is true if and only if $\sigma_i > 0 \ \forall i$. The solution does not depend continuously on the data. If a small perturbation $\varepsilon$ is made in a high frequency component $a_i$ of $g$ then a perturbation $(\varepsilon/\sigma_i)$ is induced in $G$; this can be made as large as desired by increasing $i$. Thus for any given $\varepsilon$, $\delta$ and solution pair $g$, $G$ there exists an infinite dimensional set of solutions $\tilde{g}$, $\tilde{G}$ such that

$$\mathcal{H} \tilde{g} = \tilde{g}, \quad \|g - \tilde{g}\| < \varepsilon, \quad \|G - \tilde{G}\| > \delta .$$

It is violation of condition $c$ that makes numerical inversion of compact operators so difficult. In problems arising from physical systems existence of a solution is a prerequisite for making the measurements; failure of the subsequent mathematical model to have a solution is usually due to measurement errors in $g$ or $\mathcal{H}$. Thus non-existence is really a consequence of discontinuous dependence and unimportant in its own right. Likewise the existence of multiple exact solutions is not in itself a problem. In most cases the nullspace can be predetermined theoretically and the operator restricted to the complement of this space. If the restricted
operator had a bounded inverse, numerical inversion would present no further difficulties.

Solution by filtered SVD directly addresses the problem of discontinuous dependence on the data by identifying the infinite dimensional subspace $S$ over which $\mathcal{H}$ produces distinct (but very small) variations; then restricting the approximate solutions to the complementary subspace $S^\perp$ over which $\mathcal{H}^{-1}$ is uniformly continuous. This decomposition, effected by the filter, corresponds to choosing the first $N$ terms in a generalized Fourier expansion of the solution $G$; as such it has a wide variety of uses and interpretations.

In the nonlinear problem of phase retrieval the traditional investigations of ill conditioning and ill positioning have concentrated on the questions of existence and uniqueness [3]. Necessary and sufficient conditions that $m$ and $g$ must satisfy for existence of a solution $G$ are given by the Paley-Wiener theorem. For one-dimensional retrieval problems the form and existence of multiple solutions is a consequence of the Hadamard factorization theorem (a result first derived by Akutowicz [8, 9]. An extension of this theorem gives necessary conditions for existence of multiple solutions in two-dimensional problems. A detailed exposition appears in Appendix A.

Although questions of existence and uniqueness have some influence on the behavior of numerical algorithms, as with the linear problem discontinuous dependence on the data dominates. Existence or non-existence of solutions is again just a simple consequence of discontinuous dependence. If multiple solutions exist but are uniformly separated, their existence will influence
the initial iterations of any algorithm but the final behavior will be determined by the question of whether the problem is well posed in a neighborhood of each solution or not. As we shall show the phase problem is never well posed.

However existence of multiple solutions may be a stronger contributor to ill conditioning than indicated above. As shown in the first section of Appendix A, in the one-dimensional problem there may exist an infinite sequence of exact solutions \( G_N \) with a limit \( G_\infty \), corresponding to a sequence of finite products of Blaschke factors and the limiting infinite product. In this case the solutions are not uniformly separated and any algorithm will have difficulties in the neighborhood of \( G_\infty \). Fortunately, for the reasons outlined in Appendix A, in two-dimensional problems multiple solutions appear to be rare and the limiting behavior described above unlikely in the extreme, unless symmetry considerations reduce the problem to an essentially one-dimensional form.

Proof of discontinuous dependence on the data is difficult for an arbitrary nonlinear equation \( \mathcal{L}(G) = g \). The most widely used criterion is that the Frechet derivative \( D(G) \) of \( \mathcal{L} \) at the point \( G \) be a compact linear operator. The phase retrieval problem may be formally stated as that of finding a solution pair to the equation

\[
\mathcal{L}(G) \equiv |P_c \mathcal{F}_c G| \equiv (\mathcal{L}_1 \circ P_c \mathcal{F}_c)(G) = |P_c g| = m
\]  

(3.4)

where the nonlinear operator \( \mathcal{L}_1 \) is defined by

\[
\mathcal{L}_1(g) \equiv |g|
\]  

(3.5)
so that \( \mathcal{L} \) is the composition \((\circ)\) of \( \mathcal{L}_1 \) and the compact linear operator \( \mathcal{P}_c \mathcal{F}_c \). The Fréchet derivative \( D_1(g) \) of \( \mathcal{L}_1(g) \) is the bounded linear operator defined by

\[
D_1(g)h = \text{Re}(g^*h)/|g|.
\]

(3.6)

The Fréchet derivative \( D(G) \) of \( \mathcal{L}(G) \) is the operator \( D_1(G) \circ \mathcal{P}_c \mathcal{F}_c \); since this is the composition of a bounded operator with a compact operator it follows that \( D(G) \) is compact and the phase retrieval ill-posed.

Since \( D(G) \) varies with \( G \) the decomposition by filtering of the underlying space into a subspace \( S(G) \), over which \( D(G) \) is slowly varying, and its complement \( S(G) \), is not a global decomposition. Instead \( S(G) \) and \( S(G) \) vary with \( G \) and serve as tangent planes in the definition of manifolds \( \mathcal{M} \) and \( \mathcal{M} \) over which \( \mathcal{L}(G) \) is slowly varying or \( \mathcal{L}^{-1} \) is uniformly continuous. However as \( \mathcal{M} \) is not linear, generation of an approximate solution \( \tilde{G} \in \mathcal{M} \) does not convey the information that is contained in generation of an approximate solution from a linear subspace \( S(G) \); and is also a much harder problem.

The usual approach taken to overcome this problem, and to deal with measurement noise, is to reduce the ill posed problem to a well posed problem by regularization; that is a parameter \( \lambda > 0 \) and functional \( \Omega(G) \) are chosen and the regularized solution \( G_\lambda \) found by minimization of

\[
\| \mathcal{L}(G) - y \| + \lambda \Omega(G)
\]

(3.7)

is taken as an approximation to \( G \). As stated the parameter \( \lambda \) and functional \( \Omega \) do not appear to depend on measurement noise or the problem.
form in any obvious way, but the following theorem due to Tikhonov [10] elucidates their relationship.

**Theorem 1:** Under certain mild conditions on \( \mathcal{L}, \Omega, G \) and \( g \), then

\[
G_\lambda \text{ minimizes } \| \mathcal{L}(G) - g \| + \lambda \Omega(G)
\]

if and only if

a) there exists \( \delta_\lambda > 0 \) such that

\[
G_\lambda \text{ minimizes } \| \mathcal{L}(G) - g \| \text{ subject to } \Omega(G) \leq \delta_\lambda
\]

b) there exists \( \epsilon_\lambda > 0 \) such that

\[
G_\lambda \text{ minimizes } \Omega(G) \text{ subject to } \| \mathcal{L}(G) - g \| \leq \epsilon_\lambda.
\]

This approach is explicit or implicit in many of the algorithms presented in the literature [11].

Solution by regularization does not identify an approximating subspace, or even a submanifold, such as that found by filtering; settling rather for the (less informative) identification of sets \( A_\epsilon \equiv \{ G : \| L(G) - g \| \leq \epsilon \} \) and \( A_\delta \equiv \{ G : \Omega(G) \leq \delta \} \) that contain the solution \( G \). However the phase problem does have a natural decomposition of the solution space into subspaces rather than submanifolds, encouraging solution by filtered SVD rather than by regularization. This follows from the form of \( D(G) \). Upon definition of the subspace

\[
T \equiv \{ h : \arg h = \arg g \}
\]

it is easily shown that

\[
D_1(g) h = e^{-i(\arg g)} h \quad h \in T
\]

\[
= 0 \quad h \in T^\perp
\]
Consequently $D_1(g)$ has a well defined null space $T^\perp$ and a bounded inverse on the complement $T$. Therefore the ill conditioning of $D(G)$ is essentially due to the operator $P_F P_G$ studied in the previous paper. This suggests that the global decomposition of $H$ into linear subspaces $S$ and $S^\perp$ used there in inversion of $P_F P_G g = g$ by filtered SVD will also be appropriate in the nonlinear problem. This idea is further developed in Section 6.
4. ITERATIVE METHODS

The formulation of phase retrieval as the search for a common intersection point, given in Equation (2.7), suggests use of successive projection algorithms for its solution. The simple structure of the projection operators $P_A$ and $P_{B_i}$ for the sets $A$ and $B_i$ appearing in Eq. (2.7) indicates that this class of algorithms will be computationally efficient. Such algorithms are presented for the general problem in [12] and were first applied to phase retrieval by Gerschberg and Saxton [7,13]. However these algorithms were originally derived and analyzed under conditions such as existence of an intersection point, so they should not be applied directly to the phase problem due to the presence of ill conditioning. We therefore propose and discuss modifications more suited to the ill-posed nature of this problem.

4.1 The Generic Algorithm

Let $B_i$ be a collection of $M$ sets with associated projection operators $P_i$. Since ill conditioning implies that $\bigcap_{i=1}^{M} B_i$ can be empty, we attempt to find an $x$ that is closest to all sets $B_i$ and which reduces to a common intersection point if one exists. Therefore we seek to minimize

$$F(x) = \sum_{i=1}^{M} \|x - P_i x\|^2$$  \hspace{1cm} (4.1)

The simplest iterative algorithms for minimization of $F(x)$ can be considered as pointwise approximation. For an arbitrary point $x$ the points $b_i = P_i x$ are point approximations to sets $B_i$ and induce an approximation
\[ G(z, x) = \sum_{i=1}^{M} \|z - b_i\|^2 \] (4.2)

to \( F(x) \). The algorithm then constructs a sequence \( x_n \), where \( x_{n+1} \) is determined from \( x_n \) after minimization of \( G(z, x_n) \).

**Lemma 1:** \( G(z, x) \) has a unique minimum at \( y = M^{-1} \sum_{i=1}^{M} P_i x \) and if \( y \neq z \) then \( G(z + \lambda(y - z), x) < G(z, x) \) \( \forall \lambda \in (0, 2) \).

**Proof:**

\[
G(z, x) = \sum_{i=1}^{M} \|b_i\|^2 - 2 \sum_{i=1}^{M} (b_i, z) + M\|z\|^2
\geq \sum_{i=1}^{M} \|b_i\|^2 - 2M \left( \sum_{i=1}^{M} \frac{b_i}{M} \right) \cdot \|z\| + M\|z\|^2
\]

The quadratic in \( \|z\| \) is minimized at \( \|z\| = \|y\| \), and the inequality is strict unless \( z = y \). Moreover

\[
G(z + \lambda(y - z), x) = \sum_{i=1}^{M} \|b_i - z\|^2 - 2\lambda \sum_{i=1}^{M} (y - z, b_i - z) + \lambda^2 M\|y - z\|^2
\]

\[
= \sum_{i=1}^{M} \|b_i - z\|^2 - (2 - \lambda)\lambda M\|y - z\|^2
\]

\[
< \sum_{i=1}^{M} \|b_i - z\|^2 = G(z, x)
\]

since \( z \neq y \) and \( \lambda \in (0, 2) \) imply that \( \lambda(2-\lambda)\|y-z\|^2 > 0 \).

**Proposition 1:** The sequence \( \{x_n\} \) defined by the iteration

126
\[ y_n = \frac{1}{M} \sum_{i=1}^{M} p_i x_n \]  

(4.5)

\[ x_{n+1} = x_n + \lambda_n (y_n - x_n), \quad \lambda_n \in (0, 2) \]

satisfies

\[ F(x_{n+1}) < F(x_n) \quad \text{or} \quad x_{n+1} = x_n . \]  

(4.6)

If the projection operators are single valued then \( x_{n+1} = x_n \) implies that \( x_{n+k} = x_n \) \( \forall k \). If the sequence \( x_n \) has a limit point \( x \) at which the projections are continuous and

\[ 0 < \lim \inf \lambda_n \leq \lim \sup \lambda_n < 2 \]

then \( x \) is a fixed point of the iteration.

Proof:

\[ F(x_{n+1}) = \sum_{i=1}^{M} \| x_{n+1} - p_i x_{n+1} \| \leq \sum_{i=1}^{M} \| x_n - p_i x_n \|^2 \]

\[ = G(x_n + \lambda_n (y_n - x_n), x_n) \]

\[ \leq G(x_n, x_n) = F(x_n) . \]  

(4.7)

From Lemma 1 the last inequality is strict if \( x_n \neq y_n \). Therefore

\[ F(x_{n+1}) = F(x_n) \] if and only if \( y_n = x_n \) if and only if \( x_{n+1} = x_n \) implies \( y_{n+1} = y_n \), consequently \( x_{n+k} = x_n \) for all \( k \). Let \( \bar{x} \) be a limit point of \( x_n \), if \( \bar{x} \) is not a fixed point then \( \| y - \bar{x} \| = \epsilon > 0 \). By continuity of \( \| \cdot \| \) and \( p_i \) at \( \bar{x} \), then \( \lim x_n = \bar{x} \) implies

a) \( \lim F(x_{nk}) = F(\lim x_{nk}) = F(\bar{x}) \)

b) \( \lim y_{nk} = \bar{y} . \)
Now choose $x_N$ sufficiently close to $\tilde{x}$ such that

$$|F(x_N) - F(\tilde{x})| < \frac{1}{4} \alpha^2 M \varepsilon$$

(4.8)

and

$$\|y_N - x_N\|^2 \geq \frac{\varepsilon}{2}$$

(4.9)

where

$$\alpha = \min\{\liminf_{n} \lambda_n, 2 - \limsup_{n} \lambda_n\}.$$  

(4.10)

These inequalities imply

$$F(x_{N+1}) \leq G(x_{N+1}, x_N) = G(x_N, x_N) - \lambda (2 - \lambda_n) M \|y_N - x_N\|^2$$

$$\leq F(x_N) - \frac{1}{2} \alpha^2 M \varepsilon$$

$$< F(x)$$

(4.11)

But as $F(x_n)$ is always decreasing, this implies the contradiction

$$\lim F(x_{n_k}) < F(\tilde{x}).$$

Existence of limit points and convergence of the algorithm from an arbitrary $x_0$ cannot be guaranteed without further rather restrictive global conditions such as compactness of the sets $B_i$ and continuity of the projections $P_i$. However examples showing failure of convergence after violation of these conditions are somewhat pathological; in practice, although the conditions may not be met, the algorithm almost always converges. Apparent failures are usually traced to ill conditioning not to violation or near violations of these conditions, therefore we assume henceforth the existence of limit points, leaving determination of sufficient conditions for this
existence to the analyst, and turn instead to an alternative characterization of these limit points and the algorithm itself.

4.2 Successive Projections as Steepest Descent Algorithms

To demonstrate that the generic algorithm of Proposition 1 is the standard steepest descent algorithm applied to $F(x)$ we need the following definition: Let $f: H \rightarrow \mathbb{R}$ be a real valued function on a Hilbert space $H$. Then the gradient $\nabla f(x) \in H$ of $f$ at $x$ is $y$ if and only if $y$ is the unique vector satisfying

$$\lim_{\|z\| \to 0} \frac{|f(x+z) - f(x) - (z,y)|}{\|z\|} = 0 .$$

(4.12)

The gradient $\nabla F(x)$ is the sum of the gradients of $\|x - P_A x\|^2$. We next require

Proposition 2. If there exists a $\lambda > 1$ such that

$$P_A (P_A x + \lambda (x - P_A x)) = P_A x$$

(4.13)

and $P_A x$ is single valued, then

$$\nabla (\|x - P_A x\|^2) = 2(x - P_A x).$$

(4.14)

Proof: It suffices to show that for some constant $c$ the following inequality holds

$$\|y - P_A y\|^2 - \|x - P_A x\|^2 - 2(x - P_A x, y - x) \leq c \|x - y\|^2 .$$

(4.15)

We first note the pair of inequalities.
\[ \| y - P_A y \|^2 \leq \| y - P_A x \|^2 = \| y - P_A y \|^2 + 2(y - P_A y, P_A y - P_A x) + \| P_A x - P_A y \|^2 \]

\[ \| y - P_A x \|^2 \geq \| y - P_A y \|^2 = \| y - P_A x \|^2 + 2(y - P_A x, P_A x - P_A y) + \| P_A x - P_A y \|^2 \quad (4.16) \]

and the resultant inequalities

\[ 2(y - P_A y, P_A y - P_A x) \geq -\| P_A y - P_A x \|^2 \quad (4.17) \]

\[ 2(y - P_A x, P_A x - P_A y) \leq -\| P_A y - P_A x \|^2 \quad (4.18) \]

We also require the equality

\[ \| y - P_A y \|^2 - \| x - P_A x \|^2 = 2(y - x, x - P_A x) \]
\[ = \| x - y \|^2 + \| P_A x - P_A y \|^2 + 2(y - x, P_A x - P_A y) + 2(x - P_A x, P_A x - P_A y) \quad (4.19) \]

We next bound \( \| P_A x - P_A y \| \) in terms of \( \| x - y \| \). By the hypothesis of the proposition

\[ \| P_A x + \lambda(x - P_A x) - P_A y \|^2 \geq \| P_A x + \lambda(x - P_A x) - P_A y \|^2 \]
\[ = \lambda^2 \| x - P_A x \|^2 \quad (4.20) \]

which in turn implies

\[ \| P_A x - P_A y \|^2 \geq 2\lambda(P_A x - x, P_A x - P_A y) \quad (4.21) \]

Now use of Equations (4.18) and (4.21) gives

\[ \| P_A x - P_A y \|^2 \leq 2(P_A x - y, P_A x - P_A y) \]
\[ = 2(P_A x - x, P_A x - P_A y) + 2(x - y, P_A x - P_A y) \]
\[ \leq \frac{1}{\lambda} \| P_A x - P_A y \|^2 + 2\| x - y \| \| P_A x - P_A y \| \quad (4.22) \]

Consequently

130
\[
\|P_A^{-1} x - P_A^{-1} y\| \leq \frac{2\lambda}{\lambda - 1} \|x - y\| \tag{4.23}
\]

which is the sought-for result. To establish the lower bound on the inequality in Eq. (4.15), reverse \(x\) and \(y\) in Eqs. (4.16) and (4.17), substitute the new Eq. (4.17) into Eq. (4.19) and use Eq. (4.23) to give

\[
\|y - P_A y\|^2 - \|x - P_A x\|^2 - 2(y - x, x - P_A x) \geq -\frac{\lambda + 1}{\lambda - 1} \|x - y\|^2 . \tag{4.24}
\]

For the upper bound, we note that the left hand side of Eq. (4.15) can be written as

\[
-\|y - x\|^2 - \|P_A y - P_A x\|^2 - 2(y - x, P_A y - P_A x) - 2(y - P_A y, P_A y - P_A x) - 2(y - P_A y, P_A y - P_A x)
\]

\[
- 2(x - y, y - x - (P_A y - P_A x)) \tag{4.25}
\]

upon using Eq. (4.19) with \(x\) and \(y\) reversed. Further substitution of Eqs. (4.17) and (4.23) yields

\[
\|y - P_A y\|^2 - \|x - P_A x\|^2 - 2(y - x, x - P_A x)
\]

\[
\leq -\|y - x\|^2 - \|P_A y - P_A x\|^2 + 2\|x - y\| \|P_A y - P_A x\| \tag{4.26}
\]

\[
+ \|P_A y - P_A x\|^2 + 2\|x - y\| \|y - P_A y\| \|P_A y - P_A x\| \leq \frac{9\lambda - 1}{\lambda - 1} \|x - y\|^2
\]

which completes the proof.

This result is sufficient to show that

\[
\nabla F(x) = 2 \sum_{i=1}^M (x - P_i x) = 2M(x - y) . \tag{4.27}
\]

Hence the directions \(y_n - x_n\) used in the algorithm of Proposition 1 are steepest descent directions for \(F(x)\). Furthermore the result that, at a limit point \(\tilde{x}, \tilde{y} = \tilde{x}\) shows that \(\nabla F(\tilde{x}) = 0\) so \(\tilde{x}\) is by definition a stationary point of \(F(x)\).
4.3 The Restricted Projection Algorithm

In many reconstruction problems considerations such as computational complexity and firm restrictions on the functions \( g \) and \( G \) argue for the iterates being restricted to one particular set, \( B_M \), say. We now show that the natural modification of the generic algorithm still produces a sequence with decreasing function values whose limit points are stationary points of \( F(x) \), with \( x \) restricted to \( B_M \).

**Proposition 3.** If the projections \( P_i \) are continuous and unique then the sequence \( x_n \in B_M \) defined by

\[
y_n = \frac{1}{(M-1)} \sum_{i=1}^{M-1} P_i x_n
\]

\[
x_{n+1} = P_M (x_n + \lambda (y_n - x_n)), \quad \lambda \in (0,1)
\]

satisfies \( F(x_{n+1}) < F(x_n) \) or \( x_{n+k} = x_n \) for all \( k \). Furthermore if \( \tilde{x} \) is a limit point of \( x_n \), then \( \nabla F(\tilde{x}) \) is normal to the set \( B_M \).

**Proof:** The first step is to establish a bound on \( (x_{n+1} - x_n, y_n - x_n) \). Let

\[
z_n = x_n + \lambda (y_n - x_n)
\]

then

\[
\| x_{n+1} - z_n \|^2 = \| x_{n+1} - x_n \|^2 + \| x_n - z_n \|^2 + 2(x_{n+1} - x_n, x_n - z_n)
\]

and

\[
2(x_{n+1} - x_n, x_n - z_n) = 2\lambda (x_{n+1} - x_n, y_n - x_n)
\]

\[
= \| x_{n+1} - x_n \|^2 + \| x_n - z_n \|^2 - \| x_{n+1} - z_n \|^2.
\]
Since \( P_M x_n = x_{n+1} \) we have that
\[
2(x_{n+1} - x_n', y_n - x_n) \geq \frac{1}{\lambda_n} \| x_{n+1} - x_n \|^2.
\] (4.32)

Therefore
\[
F(x_{n+1}) \leq \sum_{i=1}^{M-1} \| P_i x_n - x_{n+1} \|^2
\]
\[
= \sum_{i=1}^{M-1} \| P_i x_n - x_n \|^2 + 2 \sum_{i=1}^{M-1} (P_i x_n - x_n, x_n - x_{n+1}) + (M-1) \| x_{n+1} - x_n \|^2
\]
\[
= F(x_n) + (M-1)(x_n - x_{n+1}, 2y_n - x_n - x_{n+1})
\]
\[
= F(x_n) + (M-1)[2(x_n - x_{n+1}, y_n - x_n) + \| x_n - x_{n+1} \|^2]
\]
\[
\leq F(x_n) + (M-1) \left( 1 - \frac{1}{\lambda_n} \right) \| x_{n+1} - x_n \|^2
\] (4.33)

where the last line follows from Eq. (4.32); consequently

\[
F(x_{n+1}) \leq F(x_n)
\] (4.34)

since \( \lambda_n < 1 \). Furthermore the inequality is strict unless \( x_{n+1} = x_n \). Uniqueness of projections ensures that \( x_{n+1} = x_n \) implies that \( x_{n+k} = x_n \) \( \forall k \). Let \( \tilde{x} \) be a limit point of \( x_n \), then continuity of \( P_M \) implies that \( \tilde{x} \in B_M \) and

\[
\nabla F(\tilde{x}) = \sum_{i=1}^{M-1} 2(\tilde{x} - P_i \tilde{x}) = 2(M-1)(\tilde{x} - \tilde{y}).
\] (4.35)

Since \( \tilde{x} \) is a fixed point of the iteration (the same arguments as used in the proof of Proposition 1 shown this) then \( P_M \tilde{y} = \tilde{x} \) and so the sphere \( S \equiv \{ z : \| z - \tilde{y} \| < \| \tilde{x} - \tilde{y} \| \} \) must satisfy \( S \cap B_M = \phi \). But the sphere has a tangent plane at \( \tilde{x} \), so \( B_M \) also has a tangent plane there; since \( \tilde{y} - \tilde{x} \) is the normal to this tangent plane then \( \tilde{y} - \tilde{x} \equiv \nabla F(\tilde{x})/2(M-1) \) is normal to \( B_M \).
If the set $B_M$ is convex, then the condition $\lambda_n \in (0,1)$ can be replaced by $\lambda_n \in (0,2)$. If $B_M$ is a linear subspace then $P_M$ is a linear map, so for any $\lambda$

$$P_M(x_n + \lambda(y_n - x_n)) = P_M x_n + \lambda P_M(y_n - x_n) \in B_M.$$  \hfill(4.36)

Thus a unique search direction $P_M(y_n - x_n)$ is defined regardless of $\lambda$. A line search can be used to find the minimum of $F(P_M x_n + \lambda P_M(y_n - x_n))$ as a function of $\lambda$.

4.4 Restricted Projections in Phase Retrieval

The restricted projection algorithm is especially suitable for the phase retrieval problem for two reasons. The first reason stems from the requirement that $G(\omega) \in cI$; this set is a linear subspace and so is a natural choice for the set $B_M$ of the previous subsection. The second reason is that such a choice is efficient. The problem as stated gives information on $g$ and $G$ only over the sets $cI$, we will show that the restricted projection algorithm with $B_M = cI$ requires knowledge of $g_n$ and $G_n$ at each iteration only over $cI$, even if the solution $g(v)$ is required over the entire real line. With the restricted projection algorithm, only the values $P_{c^n} g_n$ are used until the iterates converge. At this point $P_{c^n} g$ can be extrapolated to the whole line. The existence of efficient algorithms for phase retrieval has not always been realized, several authors, [14], have proposed schemes requiring storage of values of the iterates from outside the interval $cI$.

To show the efficiency of the algorithm, we present the details of the restricted projections arising in model problems I-III. Upon defining the set
$\mathcal{F}^{-1} A = \{ G : G = \mathcal{F}^{-1} g, g \in A \}$, it follows that $g \in A$ if and only if $\mathcal{F}^{-1} g \in \mathcal{F}^{-1} A$. Since the Fourier transform preserves $L^2$ norms, $P^{-1} G$ is the closest point in $\mathcal{F}^{-1} A$ to $G$ if and only if $\mathcal{F}^{-1} G$ is the closest point in $A$ to $g$, therefore

$$P^{-1} G = \mathcal{F}^{-1} P_A g.$$  \hspace{1cm} (4.37)

$P_A g$ is easily shown to be

$$(P_A g)(v) = m(v)e^{i \arg g(v)}, \quad v \in cI$$  \hspace{1cm} (4.38)

$$= g(v), \quad \text{otherwise}$$

Therefore, after noting that $P c P_A c P_c$ and restricting $G$ to $cI$,

$$P_A g = g - P_c g + P_P P_A c g$$  \hspace{1cm} (4.39)

and

$$P c P^{-1} A c = P c \mathcal{F}^{-1} (F G - P_c F G + P_P P_A c F G)$$

$$= G - P_c \mathcal{F}^{-1} p c F G + P_c \mathcal{F}^{-1} P P_A c F G.$$  \hspace{1cm} (4.40)

As the remaining sets $B_i$ are by definition contained in $cI$, it follows that $P c P B_i = P B_i$. For the record these projections are

$$P B_1 G = P B_1$$  \hspace{1cm} (4.41)

$$(P B_2 G)(\omega) = (\Re G)(\omega), \quad \text{if} \quad (\Re G)(\omega) > 0, \omega \in cI$$

$$= 0, \quad \text{otherwise}$$  \hspace{1cm} (4.42)

$$(P B_3 G)(\omega) = n(\omega)e^{i \arg G(\omega)}, \quad \text{if} \quad \omega \in cI$$

$$= 0, \quad \text{otherwise}.$$  \hspace{1cm} (4.43)

135
For typographic convenience we will let $P^{-1}_A + P_1$, $P_B + P_2$ and $P_B + P_3$; also we will denote the restricted projection algorithm using these projections and the set $B_M = cI$ by RP. Therefore the iterates $x_n$ become iterates $G_n \in cI$. Further we shall denote the search directions $y_n - x_n$ by $H_n$. For RP, $H_n$ is given by

$$H_n = \frac{1}{M} \sum_{i=1}^{M} (P_i P_i G_n - G_n) \quad (4.44)$$

The calculation of $H_n$ requires values of $g_n$ and $G_n$ only over $cI$ after use of Eq. (4.40) for $P_{c1}$.  

4.5 Ill Conditioning and Line Searches

The geometric view of the RP algorithm developed in this section sheds new light on some of the problems encountered in previous use of projection algorithms, [7], in particular that of constant but small decrements in $F(G_n)$. This may be due to local ill conditioning of the problem; with the geometric interpretation that the sets $F^{-1}A$ and $B_i$ are nearly parallel at $G_n$, either intersecting at a very acute angle or failing to intersect at all. Figure 1 shows simple two-dimensional examples of such problems. In both cases shown in Fig. 1, $H_n$ is very small so that if $\lambda_n \in (0,2)$ then $G_n$ and $G_{n+1}$ are nearly coincident even though $G_n$ is far from the true minimum.

The relation established between projection methods and steepest descent methods opens up a wide range of algorithms already developed for gradient methods. In particular, we note that calculation of $F(G_n)$ requires evaluation of $P_{c1} G_n$ so we may simultaneously calculate...
\[ \frac{d}{d\lambda} F(G_n + \lambda H_n) = (\nabla F(G_n + \lambda H_n), H_n) \]

\[ = 2 \sum_{i=1}^{M} (G_n + \lambda H_n - P_i (G_n + \lambda H_n), H_n) \]

\[ = 2 \sum_{i=1}^{M} (G_n + \lambda H_n - P_{ci} (G_n + \lambda H_n), H_n) \quad (4.45) \]

because

\[ H_n = P_{ci} H_n, \quad (G, P_{ci} H) = (P_{ci} G, H) \quad (4.46) \]

Therefore a quadratic [15] or cubic [16] line search routine will improve convergence beyond simple variation of \( \lambda_n \) in (0,2).
5. SECOND ORDER ITERATIVE METHODS

The iterative algorithms for minimization of \( F(G) \) discussed in the previous section were developed from first order truncations, either approximating \( F(G) \) by its gradient \( \nabla F(G) \) or taking pointwise approximations to the sets \( A \) and \( B \). In a similar vein we now construct second order algorithms by approximations involving the Hessian \( H(G) \) of \( F(G) \) and affine approximations to the sets. A succession of such algorithms is constructed, increasing in accuracy and in complexity up to the standard Newton's method for minimization of \( F(G) \). In each algorithm a linear subproblem must be solved at each stage; typically the subproblem is ill-conditioned and requires a filtered inversion. The geometric viewpoint shows that the system can be taken to be a linear least squares problem whose solution is the closest point to a collection of affine subspaces approximating the sets \( A \) and \( B \). We then demonstrate that in this new form the system may be preconditioned so that a filtered inversion is possible without having to calculate an expensive singular value decomposition at each stage.

Unlike pointwise approximations, the affine approximations to the sets are not necessarily contained within the sets, therefore we cannot guarantee that \( F(G_n + \lambda H_n) < F(G_n) \) for \( \lambda \in (0, 2) \). However, it is often possible to show that the search directions \( H_n \) generated at each iteration are descent directions, i.e. \( (d/d\lambda) F(G_n + \lambda H_n)_{\lambda=0} < 0 \). In the interests of brevity only the form of the algorithms is presented here, proof of these and other similar results are left to the reader.
5.1 The Partial Affine Algorithm

As an introduction to the use of affine approximations we present a simple extension of RP. In RP at the n-th iteration the set $A$ is approximated by the point $P_{A_n}$ so that the restricted projection $P_{A_n}G$ is given by Eq. (4.40). However membership of $g$ in $A$ is determined only by the modulus values of $g$ across $c_1$; outside of this interval there are no restrictions. Therefore the affine subspace

$$A_n \equiv \{ h : h = P_{A_n}g + (g - P_{A_n}g), \ g \in L^2(\mathbb{R}) \} \quad (5.1)$$

contains the point $P_{A_n}g$ but is contained in $A$. So for any function $G \in cI$

$$\|G - P_{A_n}G\| \leq \|G - P_{A_n}^{-1}P_{A_n}g\| \quad (5.2)$$

Use of $A_n$ and the points $P_{A_n}G$ as approximations to sets gives an approximation $F_n$ to $F$ such that

$$F_n(G) = \|P_{A_n}g - P_{A_n}G\|^2 + \sum_{i=2}^{M} \|P_{G_i}G - G\|^2 \quad (5.3)$$

$$F(G) \leq F_n(G), \quad F(G) = F_n(G)$$

$F_n(G)$ is minimized at the point $L_n$ satisfying the normal equations

$$(P_{A_n}^{-1}P_{A_n} + (M-1)I)L_n = P_{A_n}^{-1}P_{A_n}g + \sum_{i=2}^{M} P_{G_i}G \quad (5.4)$$

giving a search direction $H_n = L_n - G_n$. If $M > 1$, Eq. (5.4) can be solved by any regular linear equations package since the eigenvalues of
are all greater than unity and less than \( M \). If \( M = 1 \), then minimization of \( F_n(G) \) reduces to the least squares problem

\[
\min_{G \in \mathcal{C}} \| P \mathcal{F} G - P c A_n \|^2
\]  

which is the focus of the previous paper. In this case the ill conditioning of the finite Fourier transform \( P \mathcal{F} \) forces the use of a filtered approximation \( \tilde{L}_n \) to \( L_n \) in which \( \tilde{L}_n \) is the projection of \( L_n \) onto the span of the first few eigenfunctions of \( P \mathcal{F} \). Since the same linear operator appears at each iteration, the eigendecomposition need be calculated only once. As the range of the projection remains unchanged, an extra global restriction on the iterates to this span is implicitly imposed.

Henceforth the algorithm with search direction given by Eq. (5.4) will be termed the PA algorithm.

5.2 The Gauss-Newton Algorithm

Having introduced the geometric viewpoint of algorithms as based upon set approximations, we now look at some of the standard nonlinear least squares algorithms in this context. For solution of the model problem

\[
\min_{\hat{x} \in \mathbb{R}^N} F(\hat{x}) = \min_{\hat{x} \in \mathbb{R}^N} \sum_{i=1}^{M} f_i^2(\hat{x})
\]  

a popular choice of iterative algorithm is the Gauss-Newton method [17] in which at the \( n \)-th iteration the search direction \( \hat{z}_n \) is given by the least squares minimum norm solution to the linear system
\[ \hat{J}(\hat{x}_n) \hat{s}_n = -\hat{f}(\hat{x}_n) \]  

(5.7)

where \( \hat{f}(\hat{x}_n) \) is the vector in \( \mathbb{R}^M \) with components \( f_i(\hat{x}_n) \) and \( \hat{J}(\hat{x}_n) \) is the \( M \times N \) matrix with entries

\[ \hat{J}_{ij}(\hat{x}_n) = \frac{\partial f_i(\hat{x}_n)}{\partial x_j} \]  

(5.8)

In the present paper \( x \in G \subseteq L^2(\mathbb{R}) \) and \( f_i(x) = \|x - P_i x\| \); the gradient of \( f_i(x) \) as defined by Eq. (4.12) is

\[ (\nabla f_i)(x) = (x - P_i x)/\|x - P_i x\| \]  

(5.9)

so that the Jacobian \( J \) is now an operator from \( L^2(\mathbb{R}) \) into \( \mathbb{R}^M \) defined by

\[ [J(x)z]_i = (x - P_i x, z)/\|x - P_i x\| \]  

(5.10)

Therefore \( z_n \) is now the least squares, minimum norm solution to the underdetermined system, Eq. (5.7), with \( J(x_n) \) replacing \( \hat{J}(\hat{x}_n) \); such a \( z_n \) gives a descent direction for \( F(x) \).

We next elucidate the affine approximation implicit in this algorithm.

Since \( P_i x_n \) is the unique closest point to \( x_n \) from \( B_i \), it follows that if \( S_i(x_n) \) is the sphere

\[ S_i(x_n) = \{ z : \|x_n - z\| \leq \|x_n - P_i x_n\| \} \]  

(5.11)

then \( B_i \cap S_i(x_n) = P_i x_n \). At \( P_i x_n \) the sphere has a well defined tangent plane

\[ T_i(x_n) = \{ P_i x_n + z : (x_n - P_i x_n, z) = ((\nabla f_i)(x_n), z) = 0 \} \]  

(5.12)
This must also be locally a supporting hyperplane for $B_i$ at $P_i x_n$; Fig. 2 shows a simple two dimensional example. $T_i(x_n)$ can be taken as an affine approximation to $B_i$ at the $n$-th iteration, in which case the least squares, minimum norm solution to Eq. (5.4) is the closest point in $\bigcap_{i=1}^{M} T_i(x_n)$ to $x_n$. Since $T_i(x_n)$ has codimension one, this intersection is always nonempty, being in fact an infinite dimensional linear subspace.

Although we bring this algorithm to the readers attention we did not implement it for two reasons. The first is that the subspaces $T_i(x_n)$ are poor approximations to the sets $B_i$, containing no further information than that already available in the projections $P_i x_n$. Second for our model problems $M \leq 3$, so that Eq. (5.7) is of very small dimensions. Therefore the search directions $z_n$ are not likely to be very different from those of RP or PA.

5.3 Newton’s Algorithm

The standard second order Newton’s algorithm for Eq. (5.6) is based on the approximation of $F(z)$ near $x = x_n$ by a truncated Taylor series expansion

$$ F(x) \approx F(x_n) + VF(x_n)(x-x_n) + \frac{1}{2} (x-x_n) V^2F(x_n)(x-x_n) \quad (5.13) $$

If $V^2F(x_n)$ is a positive definite matrix, then the approximation is minimized at the point $y_n = z_n + x_n$, where $z_n$ is the solution of

$$ V^2F(x_n)z = -VF(x_n) \quad (5.14) $$
In phase retrieval problems this linear system is ill conditioned and the Hessian often has negative eigenvalues. Therefore to ensure that a descent direction is chosen for which Eq. (5.13) is a good approximation, an eigendecomposition of the Hessian must be calculated and a filtered solution $z_n$ used. This approach must be taken if Eq. (5.14) is solved directly. However, as we will show, a geometric interpretation allows the equation to be rewritten in a form in which filtering can be performed without the cost of an eigendecomposition at each iteration.

We begin by noting that the quantities $\nabla F(x_n)$ and $\nabla^2 F(x_n)$ are given by

$$\nabla F(x_n) = 2 \sum_{i=1}^{M} (x_n - P_i x_n)$$

$$\nabla^2 F(x_n) = 2 \sum_{i=1}^{M} (I - H_i(x_n))$$

(5.15)

(5.16)

where the operators $H_i(x_n)$ are the Hessians of the functions $f_i(x) = \|x - P_i x\|^2$ evaluated at $x_n$. They are defined by

$$H_i(x_n) : L^2(\Omega) \rightarrow L^2(\Omega)$$

$$H_i(x_n)y = \lim_{\alpha \to 0} \frac{P_i (x + \alpha y) - P_i (x)}{\alpha}$$

(5.17)

Conditions on the set $B_i$ that guarantee existence and boundedness of the operator $H_i$ are quite complicated [18]. For the purposes of this paper we shall assume that $H_i(x_n)$ exists. The only further properties we require are that $H_i$ is symmetric and that
\[ \mathcal{H}_i(x_n)(x - P_i x_n) = 0 \] \hspace{1cm} (5.18)

The Hessian implicitly defines a first order approximation to the projection operator \( P_i \) at \( x_n \) by
\[ P_i x \approx P_i x_n + \mathcal{H}_i(x_n)(x - x_n) \] \hspace{1cm} (5.19)

which in turn defines an associated affine approximation \( U_i(x_n) \) to the set \( B_i \) at \( P_i x_n \)
\[ U_i(x_n) \equiv \{ P_i x_n + z : z \in \text{Range } \mathcal{H}_i(x_n) \} \] \hspace{1cm} (5.20)

\( U_i(x_n) \) has more structure than the previous approximation \( T_i(x_n) \) because \( \mathcal{H}_i(x_n) \) is not just a projection operator with range \( U_i(x_n) - P_i x_n \) but is also a contraction (or expansion) mapping about \( P_i x_n \) depending on whether \( B_i \) is locally convex (or concave) in particular directions at \( P_i x_n \). Figure 3 shows a two-dimensional example of \( U_i(x_n) \) for \( B_i \) a sphere; \( \mathcal{H}_i(x) \) is obviously a contraction mapping. Therefore at the n-th iteration we choose a search direction \( z_n = y_n - x_n \) where \( y_n \) minimizes the quadratic approximation
\[ F(y) \approx \sum_{i=1}^{M} \| y - P_i x_n - \mathcal{H}_i(x_n)(y - x_n) \|^2 \] \hspace{1cm} (5.21)

to \( F(x) \). The geometric view of \( y_n \) is that it is the "closest point" to the collection of subspaces \( U_i(x_n) \) with the distance measured, not in the standard \( L^2 \) metric, but as a sum of new metrics. Each new metric reflects the degree of curvature of the set \( B_i \) at \( P_i x_n \), and therefore the accuracy of the affine approximation to \( B_i \) at \( P_i x_n \).
The direction $z_n$ of Eq. (5.21) is equivalently described as the least squares solution to the block system

$$
\begin{pmatrix}
I - H_i(x_n) & -(x_n - P_i x_n) \\
\vdots & \vdots \\
I - H_M(x_n) & -(x_n - P_M x_n)
\end{pmatrix}
\begin{pmatrix}
z \\
\vdots \\
z
\end{pmatrix}
= 
\begin{pmatrix}
-(x_n - P_i x_n) \\
\vdots \\
-(x_n - P_M x_n)
\end{pmatrix} \quad (5.22)
$$

Equations (5.22) and (5.14) are not the same, so that the search directions $z_n$ are different. However, the equations are sufficiently close in form to show how Eq. (5.14) can be recast in a block form more suitable for numerical computation. To accomplish this, we note that the normal equations for Eq. (5.22) are

$$
\left( \sum_{i=1}^{M} (I - H_i(x_n))^2 \right) z = - \sum_{i=1}^{M} (I - H_i(x_n))(x_n - P_i x_n)
= - \sum_{i=1}^{M} (x_n - P_i x_n) \quad (5.23)
$$

after use of Eq. (5.18). If the operators $I - H_i(x_n)$ are positive definite, then they possess a positive definite square root $(I - H_i(x_n))^{1/2}$ which by Eq. (5.18) satisfies

$$
(I - H_i(x_n))^{1/2}(x_n - P_i x_n) = x_n - P_i x_n \quad (5.24)
$$

Therefore Eq. (5.14) can be rewritten as

$$
\left[ \sum_{i=1}^{M} ((I - H_i(x_n))^{1/2})^2 \right] z = - \sum_{i=1}^{M} (I - H_i(x_n))^{1/2}(x_n - P_i x_n) \quad (5.25)
$$

145
which in turn is recognizable as the normal equations associated with the
block system

\[
\begin{bmatrix}
(I - \mathcal{K}_1(x_n))^{1/2} & - (x_n - P_1 x_n) \\
\vdots & \vdots \\
(I - \mathcal{K}_M(x_n))^{1/2} & - (x_n - P_M x_n)
\end{bmatrix}
\begin{bmatrix}
\hat{z} \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\vdots \\
\vdots
\end{bmatrix}.
\]  

(5.26)

Equation (5.26) has a geometric interpretation, its solution is the solution
to an approximate minimization of \( F(x) \) similar to that of Eq. (5.21) using
the same affine approximation \( U_i(x_n) \) to \( B_i \) but different metrics.

5.4 Ill Conditioning and Filtering

We noted at the end of Section 4 that line searches could overcome some
simple cases of ill conditioning. We now consider more complicated cases and
ways to alleviate them.

The first comes from the fact that \( L^2(cI) \) is not one-dimensional, as
portrayed in Fig. 1, so that the valleys of the surface \( T = \{ (F(G), G) : G \in cI \} \)
will not typically be straight but rather will be curving through space in
much the same fashion as the classical test case for optimization algorithms,
the Rosenbrock function [16]. Experience has shown that second order
methods significantly outperform gradient methods in such examples, but that
such features can easily be sufficiently ill conditioned to defeat even second
order algorithms.

The second arises from the existence of the subspace \( S \) described in
Section 3 which is, for practical purposes, the null space of \( P_c \). If a
significant component of the search direction lies within this subspace then
a relatively large step can be taken for a very small decrement in $F(G)$.
This may lead to "zig-zags" in which steps are taken backwards and forwards
through $S$ so that the iterates do not appear to converge and yet $F(G_n)$ is
decreasing, albeit very slowly. The presence of $S$ also implies that the
Hessian $\nabla^2 F(G_n)$ will have very small eigenvalues corresponding to directions
in $S$ (particularly in Hessians from model problem I) so that Eq. (5.14) is
difficult to solve numerically.

These problems can be overcome by calculation of the eigendecomposition
of $\nabla^2 F(G_n)$, filtering the eigenvalues (which are real since the Hessian is
symmetric) by choice of a cutoff parameter $\epsilon_3$ and use of the filter

$$f(\sigma) = \sigma, \quad \text{if } |\sigma| \geq \epsilon_3$$
$$= 0, \quad \text{otherwise}, \quad (5.27)$$

then finding the minimum norm least squares solution $z_n$ of the resulting
filtered version of Eq. (5.14). This scheme produces a search direction in
which $F(G)$ should vary moderately rapidly because directions corresponding
to small eigenvalues, and thus slowly varying $F(G)$, have been filtered out.
We denote such a filtered Newtonian algorithm by $FN$. The resulting direction
$z_n$ is not necessarily a descent direction ($\nabla^2 F(G_n)$ may have large negative
eigenvalues) so we shall also consider a variant of $FN$ using the filter

$$f(\sigma) = \sigma, \quad \text{if } \sigma \geq \epsilon_3$$
$$= 0, \quad \text{otherwise}. \quad (5.28)$$

The resulting algorithm is termed $FNP$. 147
The problem of global ill conditioning, in particular the presence of multiple minima due to possible nonuniqueness of the phase retrieval problem, cannot be addressed by similar modifications to those undertaken for local ill conditioning simply because the algorithms are derived from local analysis.

5.5 Block Filtering

Having shown the need for filtering, we now show that Eq. (5.14) can be solved by a filtered solution of Eq. (5.26) in which each block is prefiltered and a standard least squares algorithm then applied (avoiding the cost of a singular value decomposition of the block matrix). We begin by noting that if our algorithm is to be efficient then we must solve

\[ P_c \nabla^2 F(G_n) H = -P_c \nabla F(G_n) \quad (5.29) \]

rather than Eq. (5.14), so that values only over the interval \( cI \) are used. (For this reason, Eq. (5.29) rather than Eq. (5.14) is also used in FN and FNP). We also note that the Hessians are not complex valued linear operators since it will be obvious that for an arbitrary complex number \( \alpha \)

\[ \mathcal{H}_1(G_n)(\alpha H) \neq \alpha(\mathcal{H}_1(G_n)H) \quad (5.30) \]

Rather they are linear operators on the pair of real functions \((\text{Re } H)(\omega)\) and \((\text{Im } H)(\omega)\).

The ability to do block prefiltering will depend upon the special form of the Hessian in Eq. (5.26). For the operators \( \mathcal{H}_2 \) and \( \mathcal{H}_3 \) associated with \( P_2 \) and \( P_3 \) it is obvious that \( P_c \mathcal{H}_2 = \mathcal{H}_1 \). In addition, simple calculations establish
\[ \mathcal{H}_2(G_n) = \begin{cases} (\text{Re } H)(\omega) & \text{if } (\text{Re } G_n)(\omega) > 0 \\ 0 & \omega \in cI \end{cases} \]

and

\[ \mathcal{H}_3(G_n) = \begin{cases} (\text{Re } H)(\omega) & \text{if } (\text{Re } G_n)(\omega) > 0 \\ n(\omega) & \omega \in cI \end{cases} \]

\[ \mathcal{H}_3(G_n) = \begin{cases} (\text{Re } H)(\omega) & \text{if } (\text{Re } G_n)(\omega) > 0 \\ \sin^2 \theta & \text{if } n(\omega) > 0 \\ -\sin \theta \cos \theta & \text{otherwise} \end{cases} \]

where

\[ \theta = \arg G_n(\omega), \quad \omega \in cI. \]

The operators \( I - \mathcal{H}_i \) can be represented by \( 2 \times 2 \) block diagonal operators; at each point \( \omega \) the blocks are

\[ [I - \mathcal{H}_2(G_n)](\omega) = \begin{cases} 0 & \text{if } (\text{Re } G_n)(\omega) > 0 \\ 1 & \text{otherwise} \end{cases} \]

\[ [I - \mathcal{H}_3(G_n)](\omega) = \begin{cases} \sin \theta & \cos \theta & 1 - \frac{n(\omega)}{G_n(\omega)} & 0 \\ -\cos \theta & \sin \theta & 0 & 1 \end{cases} \]

The spectrum of each operator \( I - \mathcal{H}_i \) is now seen to be simply the union over \( \omega \) of the spectrum of each block. For \( I - \mathcal{H}_2 \) this gives a spectrum.
consisting only of the points \{0,1\}, so \(I - \mathcal{H}_2\) is its own square root. Therefore the only prefiltering that needs to be done to \((I - \mathcal{H}_2)^{1/2}\) is to eliminate all equations for \((\text{Re } H)(\omega)\) that correspond to zeros in the block spectra. For \(I - \mathcal{H}_3\), the spectrum will take on a range of values. Some of these values may be small or negative if for some \(\omega\) the quantity 
\(1 - n(\omega)/|G_n(\omega)|\) is small or negative. Consequently \((I - \mathcal{H}_3)^{1/2}\) either is not well defined (i.e. has imaginary eigenvalues) or is ill conditioned, or is both. To prefilter the block \((I - \mathcal{H}_3)^{1/2}\) we therefore choose a parameter \(\varepsilon_3 > 0\) and one of the following filters:

\[
f(\sigma) = \frac{1}{2} \begin{cases} \frac{1}{2} \text{sgn } \sigma, & \text{if } |\sigma| \geq \varepsilon_3 \\ 0, & \text{otherwise} \end{cases} \quad (5.36)
\]

\[
f(\sigma) = \frac{1}{2} \begin{cases} \frac{1}{2}, & \text{if } \sigma \geq \varepsilon_3 \\ 0, & \text{otherwise} \end{cases} \quad (5.37)
\]

The prefiltered block is then defined as

\[
(I - \mathcal{H}_3(G_n))^{1/2}(\omega) = \begin{bmatrix}
\sin \theta & \cos \theta \\
-cos \theta & \sin \theta
\end{bmatrix}
\begin{bmatrix}
f\left(1 - \frac{n(\omega)}{|G_n(\omega)|}\right) & 0 & \sin \theta & -\cos \theta \\
0 & 1 & \cos \theta & \sin \theta
\end{bmatrix}.
\]

It is reduced in size by eliminating those equations for \(\sin \theta(\text{Re } H)(\omega) - \cos \theta(\text{Im } H)(\omega)\) that are associated with elements in the spectrum that are filtered out.

From Eq. (4.39) \(\mathcal{H}_1\) has the form
\[ \mathcal{H}_1(G_n) = I - \tilde{F}_{P_c}^{-1} \mathcal{P}_c + \tilde{F}_{P_c}^{-1} \mathcal{P}_c \mathcal{H}_A(g_n) \mathcal{P}_c \mathcal{P}_c \]  

so that

\[ (I - \mathcal{H}_A(g_n))(G_n) = \mathcal{P}_c \tilde{F}_{P_c}^{-1} (I - \mathcal{H}_A(g_n)) \mathcal{P}_c \mathcal{P}_c \]  

where \( (I - \mathcal{H}_A(g_n)) \) is in block diagonal form over \( cI \):

\[ \begin{bmatrix} [I - \mathcal{H}_A(g_n)](v) = \begin{vmatrix} \sin \phi & \cos \phi \\ -\cos \phi & \sin \phi \end{vmatrix} \begin{bmatrix} 1 - \frac{m(v)}{g_n(v)} \\ 0 \end{bmatrix} = \begin{vmatrix} \sin \phi & -\cos \phi \\ \cos \phi & \sin \phi \end{vmatrix} \end{bmatrix} \]  

with

\[ \phi \equiv \arg g_n(v), \quad v \in cI \]  

Unfortunately \( I - P \mathcal{H}_1 \) does not possess an obvious symmetric square root. However, an asymmetric root can be found by noting that

\[ (I - P \mathcal{H}_1(G_n)) = G_n - P \mathcal{P}_c \mathcal{P}_c \]  

is the normal equation associated with the least square problem

\[ \min_H \| (I - \mathcal{H}_A(g_n))^{1/2} \mathcal{P}_c \mathcal{P}_c \mathcal{H} - P \mathcal{P}_c \mathcal{P}_c \mathcal{P}_c \mathcal{P}_c \| \]  

so that we may take for the block \( (I - P \mathcal{H}_1)^{1/2} \) the composite operator \( (I - \mathcal{H}_A)^{1/2}(P \mathcal{P}_c) \) and prefilter each component. The filtered component \( (I - \mathcal{H}_A)^{1/2}(P \mathcal{P}_c) \) is calculated in the same fashion as \( (I - \mathcal{H}_3)^{1/2} \); the component \( P \mathcal{P}_c \) requires further manipulations before prefiltering. Let \( P \mathcal{P}_c \) have an eigendecomposition \( U \bar{U} U^+ \), then as \( U \) is a unitary operator the solution \( H_n \) to a prefiltered Eq. (5.26) is given by \( H_n = U L_n \), where \( L_n \) is the least
squares solution to

\[
\begin{bmatrix}
(I - \mathcal{H}_A^n(f))^{1/2} U \\
(I - \mathcal{H}_I^n(f))^{1/2} U
\end{bmatrix}
\begin{bmatrix}
L \\
\Sigma
\end{bmatrix} =
\begin{bmatrix}
P & P & P & P \\
G & c & A & \Sigma_n
\end{bmatrix}
\begin{bmatrix}
P & G - G \\
1 & n & n & n
\end{bmatrix}
\]

(5.45)

where \( \Sigma \) may be filtered using Eqs. (5.36) or (5.37).

The reason for choosing to solve for \( L_n \) instead of \( H_n \) lies in the choice of the pointwise discretization used in numerical calculations and in the form of the IMSL subroutine LLSQF used to solve the discretized Eq. (5.45). This subroutine uses an adaptive QR algorithm, at each stage a column is chosen from the block matrix and incorporated in the QR factorization calculated at the previous stage. A running estimate is kept on the condition number of \( R \), if this estimate exceeds a use defined parameter, ATOL, then the routine halts and computes a minimum norm least squares solution using the most recent QR factorization and also sets components in \( L \) corresponding to unused columns to zero. For a solution \( H \) this corresponds to setting point values of \( H(\omega) \) to zero; for \( L \) it is setting components of \( L \) in an eigenfunction expansion to zero. Our analysis of ill conditioning indicates that such components are likely to be zero anyway, so we believe Eq. (5.45) to be the preferred form for solution.

We shall term the algorithm based on Eq. (5.45) with the filter given by Eq. (5.24) the SQFN algorithm, and that based on Eq. (5.45) with the filter given by Eq. (5.37) the SQFNP algorithm. SQFN incorporates more information about the Hessian at each step, but SQFNP gives a guaranteed descent direction \( H_n \) at each step.
6. NUMERICAL RESULTS

We now present results for the solution of the discretized model problems using the various algorithms so far proposed and show the discrete approximate solutions $G$ generated. The main measure of performance used in this section will be the number of iterations $\hat{n}$ required by the algorithm to reduce the function $F(G_n)$ to below a prescribed value. Since we are interested more in the comparative than absolute behaviour of the algorithms we take as the benchmark case the RP algorithm and consider the quantity $\hat{n}/\bar{n}$ where $\bar{n}$ is the number of iterations required by RP to reduce $F(\hat{G}_n)$ to below a prescribed value.

The ill conditioned nature of the problem means that this is not the best of measures. Difficulties are encountered through behaviour such as the erratic decrease of $F(\hat{G}_n)$; often a succession of iterates produce little change in $F(\hat{G}_n)$ and then a sudden decrease is recorded. In some algorithms the iterates may converge to a particular value $\tilde{G}$ whereas in others they fail to converge so that $F(\hat{G}_n)$ and $F(\tilde{G})$ can be considerably different. For these reasons the results reported here are intended as a guide to the behaviour of algorithms on real problems rather than a prediction. However, poor though the measure is, there is none other available for use that does not require prior knowledge of the solution.

6.1 Discretization, Test Functions and Algorithms

The choice of discretization in these calculations was determined by the condition that it give an accurate, efficient approximation to the
finite Fourier transform and that the discretized projections be easily calculated. Therefore a discretization based on $N$ point Gaussian quadrature was chosen; the previous paper shows its accuracy and efficiency and its pointwise nature allows easy evaluation of projections. The finite dimensional problem now involves vectors $\hat{g}, \hat{G} \in \mathbb{C}^N$ whose elements $g_k, G_k$ are to be approximations to the function values $g(\rho_k), G(\rho_k)$ at the abscissae $\rho_k$ of the $N$ point Gaussian quadrature rule on $\mathcal{C}_L$. Vectors $\hat{m}, \hat{n} \in \mathbb{R}^N$ are formed with components $m_k = m(\rho_k), n_k = n(\rho_k)$ where $m(w)$ and $n(w)$ are the known moduli. Matrices $\hat{W}, \hat{F} \in \mathbb{C}^{N \times N}$ are constructed with $\hat{W}$ being a diagonal matrix whose $k$-th entry is the weight $\omega_k$ of the quadrature rule and with $\hat{F}$ having entries $F_{k,l} = \frac{1}{2\pi \rho_k \rho_l}$. This construction gives as the discretized model problems

Find vectors $\hat{g}$ and $\hat{G}$ such that $\hat{g} = \hat{F}\hat{W}\hat{G}$ and

I: $|g_k| = m_k$

II: $|g_k| = m_k, G_k \geq 0$

III: $|g_k| = m_k, |G_k| = n_k$

where $\hat{m}$ and $\hat{n}$ are specified.

Since the metric for all analysis so far has been $\| \cdot \|_2$, the numerical calculations were done on a rescaled version of the equation $\hat{g} = \hat{F}\hat{G}$,

$$\left(\hat{W}^{1/2}\hat{g}\right) = \left(\hat{W}^{1/2}\hat{F}\hat{W}^{1/2}\right)\left(\hat{W}^{1/2}\hat{G}\right) \quad (6.1)$$

With this rescaling the Euclidean norms of the vectors $\hat{W}^{1/2}\hat{G}$ and $\hat{W}^{1/2}\hat{g}$ are now good approximations to the norms $\|G\|_2$ and $\|g\|_2$, and $\hat{W}^{1/2}\hat{F}\hat{W}^{1/2}$ is a good approximation to $P_c \mathcal{F}' P_c$ in the operator norm induced by $\| \cdot \|_2$. 

154
The two test functions chosen for the model problems were

\[ G^1(\omega) = e^{\omega}, \quad G^2(\omega) = e^{\frac{3}{4} \pi i \left[ P_3\left(\frac{\omega}{c}\right) + P_4\left(\frac{\omega}{c}\right) \right]} \]  

(6.2)

where \( P_i(\omega) \) is the \( i \)-th monic Legendre polynomial. The Fourier transform \( g^1(v) \) of \( G^1(\omega) \) is

\[ g^1(v) = 2 \frac{\sin(2\pi v-i)c}{(2\pi v-i)}, \]  

(6.3)

\( g^2(v) \) has no closed form and was calculated numerically. Both functions were used as test cases for model problem I, \( G^1 \) and \( g^1 \) as a test case for problem II and \( G^2 \) and \( g^2 \) as a test case for problem III. The discrete approximations to \( G^i(\omega) \) and \( g^i(v) \) are denoted by \( \hat{G}^i \) and \( \hat{g}^i \).

All numerical tests used the following basic algorithm

1: Choose an initial guess \( G_0 \)

2: Given iterates \( \hat{g}_n \) and \( \hat{G}_n \) compute a search direction \( \hat{H}_n \)

3: Calculate a step length \( \lambda_n \) and new iterates

\[ \hat{G}_{n+1} = \hat{G}_n + \lambda_n \hat{H}_n, \quad \hat{g}_{n+1} = \hat{g}_n + \lambda_n \hat{H}_n \]

4: Iterate steps 2 and 3 until the convergence criteria are satisfied.

The convergence criteria used in all calculations were

\[ F(\hat{G}_n) < 10^{-5} \quad \text{or} \quad \sum_{k=0}^{2} \| \lambda_{n-k} \hat{H}_{n-k} \| < 2 \times 10^{-3} \]  

(6.4)

together with an upper limit \( N_{\text{max}} \) on the number of iterations. The sum of the last three steplengths, rather than \( \| \lambda_n \hat{H}_n \| \) alone, was chosen as in ill conditioned minimization problems "stop-start" behaviour is often noticed.
That is a large step often followed by one or two small steps after which another large step is taken. This feature was often observed in the use of any second order algorithm, differences in magnitude of successive step lengths by factors greater than 100 occurred fairly frequently.

Three different forms of initial guess were used

1. \( \hat{G}_0 = 0 \)
2. \( G_0 \hat{\xi} = (1-\xi_1)(1+\xi_1 r) \hat{G}_0 \)
3. \( G_0 \hat{\xi} = 10^{-3} r \hat{\xi} \)

Guess 2 is a damped perturbation of the true solution \( \hat{G}_0 \) with \( \xi_1 \) representing the noise level. For convenience we shall express this level as a percentage, e.g. \( \xi_1 = 0.2 \) will be expressed as \( \xi_1 = 20\% \) noise. The variable \( r \) is a random complex variable with modulus uniformly distributed over \([0,1]\) and phase uniformly distributed over \([0,2\pi]\). Guess 3 represents a small totally random perturbation about the origin, again for convenience such guesses shall be denoted by \( \xi_1 = 100\% \).

6.2 Choice of Step Lengths \( \lambda_n \)

The first results reported are those on determination of an optimal choice of \( \lambda_n \). Three possibilities were considered

1. \( \lambda_n^1 = 1 \)
2. \( \lambda_n^2 = r_n \) where \( r_n \) is a random variable uniformly distributed over \([0,2]\)
3. $\lambda^3_n$ is the approximate minimum of $F(\hat{G}_n + \lambda_n \hat{H}_n)$ as a function of $\lambda$ determined by Powell's cubic line search algorithm [16] with the following convergence criteria on the iterates $\lambda^3_n$

$$\left| \frac{(k-1)\lambda^3_n - k\lambda^3_n}{k\lambda^3_n} \right| \frac{F(\hat{G}_n + k\lambda^3_n \hat{H}_n)}{F(\hat{G}_n)} < 0.02$$

$$\left| \frac{\nabla F(\hat{G}_n + k\lambda^3_n \hat{H}_n)}{\nabla F(\hat{G}_n)} \right| \frac{F(\hat{G}_n + k\lambda^3_n \hat{H}_n)}{F(\hat{G}_n)} < 0.02 \quad k \leq 5$$

The performances of $\lambda^i_n$ were compared by running each possibility on each model problem with the appropriate test functions using the RP algorithm. The parameters $c = 2$, $N = 40$ and $N_{\text{max}} = 50$ were chosen and each problem was started with three different initial guesses with $\epsilon_1 = 20\%$, 60\% and 100\%.

The results were remarkably uniform over all test cases of model problem, test function and initial guess. Choices $\lambda^1_n$ and $\lambda^2_n$ performed almost identically with $\lambda^3_n$ just under a factor of 2 better. Almost always only one extra function evaluation was required for $\lambda^3_n$, i.e. the convergence criteria of Equation (6.5) were satisfied at $k = 1$. This extra computation almost exactly balances the savings in the reduced number of iterations so that all three choices incurred the same computational cost in reduction of $F(\hat{G}_n)$ to a specified value. However the greater flexibility of $\lambda^3_n$ led to its adoption in all subsequent calculations.

The convexity of the set $C_I$ implies that $F(\hat{G}_n + \lambda \hat{H}_n) < F(\hat{G}_n)$ for all $\lambda \in (0, 2)$, the computations showed that this inequality still frequently held
for $\lambda \in (2,3)$ but almost always failed beyond this. Almost all the values of $\lambda_n^3$ lay in the interval $[1,2.5]$ so in all subsequent line searches an initial guess of $\lambda_n^3 = 4$ was used. As an experiment further runs were made with the fixed choice $\lambda_n = 2$ but these compared very poorly with $\lambda_n^1$.

6.3 Initial Guesses and Ill Conditioning

We now present some results on the ill conditioning of the model problems. The first set are from efforts to estimate local ill conditioning through the effects of small perturbations of the data $m(v)$. They were obtained from model problems ã-III with the appropriate test functions, parameters $c = 2$, $N = 40$ and $N_{\text{max}} = 40$ and using the RP algorithm. Random relative errors of size $\epsilon_2$ were induced in the vector $\hat{m}$ representing the data, then the algorithm was started at the true solution $\hat{G}$ of the unperturbed problem. Table 1 gives some typical results for two perturbations $\epsilon_2 = 1\%$ and $\epsilon_2 = 5\%$. The data appearing is the percentage relative error in $\tilde{G}$ $(100.\|\hat{G} - \hat{G}^i\|/\|\hat{G}\|)$ and the value $F(\tilde{G})$ where $\tilde{G}$ is the approximate solution to the perturbed problem reached by the algorithm.

The data indicate that the problems are locally well conditioned in that there exists a possible nearby solution to the perturbed problem to which the iterates tended and whose distance from the solution of the unperturbed problem is of the same order as the perturbation. We cannot assert that the solution does exist for in almost all cases the algorithm terminated with $F(G_n) < 10^{-5}$ or $n > N_{\text{max}}$. Termination due to convergence of successive
iterates did not occur although they always appeared to be circling about some common point. This problem was encountered throughout these test computations, however we delay further discussion until the presentation of results on filtering.

The second set of results indicates the global ill conditioning of the problem, particularly with respect to the natural measure \( F(\hat{G}) \). Figures 4-6 represent three functions \( \hat{G} \) found as solutions to model problem I with test function \( G^2 \) with values \( F(\hat{G}) \) of \( 8 \times 10^{-5} \), \( 7 \times 10^{-5} \) and \( 8 \times 10^{-4} \), respectively. Figures 7-9 show the \( \hat{G} \)'s obtained by repeating the runs for model problem III holding all other factors (algorithm, initial guess, etc.) constant. The values \( F(\hat{G}) \) were \( 2 \times 10^{-2} \), \( 3 \times 10^{-2} \) and \( 7 \times 10^{-2} \), respectively. Although all functions \( \hat{G} \) resemble \( \hat{G}^2 \) to some degree the wide variance among them suggests that the surface \( T = \{(F(G),G): G \in cI\} \) is very rugged. Furthermore although the functions in Figures 7-9 appear graphically to be closer to \( \hat{G}^2 \) than those in Figures 4-6 the function \( F(\hat{G}) \) holds them to be substantially further away.

In general the solutions \( \hat{G} \) to model problems II and III were, as suggested by these graphs, definitely more acceptable as solutions to the phase retrieval problem than solutions to model problem I; nevertheless the values \( F(\hat{G}) \) for II and III were almost always a factor of 10 to \( 10^3 \) greater than those of I. A tentative ranking of the problems in order of decreasing ill conditioning would be I (with \( \hat{G}^1 \)) \( \succ \) II \( \succ \) I (with \( \hat{G}^2 \)) \( \succ \) III. As expected more information gives better solutions, however we were not sure to what degree the particular choice of \( \hat{G}^1 \) with its large discontinuity at \( \omega = c \)
influenced this ordering.

We conclude this subsection with a brief mention of an anomaly that lead to use of initial guess 3 rather than guess 1. As noted in a previous paper [19] it is possible to show that if iterates possess certain symmetries then almost all the algorithms preserve these symmetries. In particular with the highly symmetric choice of $G_0 = 0$ some form of symmetry was always preserved yet the algorithm often converged to reasonable pseudo-solutions. One such example for model problem I with $N = 40$ and $c = 2$ is shown in Figure 10, it displays a symmetric solution $\tilde{G}$ for which $F(G) < 5 \times 10^{-5}$. Therefore, if no a priori knowledge is available for $\hat{G}$, a small random perturbation is a better initial guess than zero.

6.4 Filtered PR and PA Algorithms

Having determined the problem to be globally ill conditioned we sought to produce a better conditioned global problem by filtering out the local ill conditioning of the finite Fourier transform discussed previously. This was done by projecting the search directions $\hat{H}_n$ (and therefore the iterates $\hat{G}_n$) on to the span of the first $P$ eigenfunctions of $P_c \mathcal{F}_P$ whose associated eigenvalues had magnitude greater than a prescribed cutoff $\epsilon_3$. As a test this filter was applied to iterates in the RP algorithm producing the filtered restricted projection algorithm (FRP) and the algorithms compared on all model problems with parameters $c = 2$, $N = 40$ and $N_{\text{max}} = 40$. Two choices of cutoff $\epsilon_3 = .9$ and $\epsilon_3 = .25$ were considered, these values gave projections on to subspaces of dimensions 15 and 18, respectively.
As measured by $\mathbf{F}(\mathbf{G})$, FRP performed very poorly compared to RP for $\varepsilon_3 = .9$. At best, 1.5 times as many iterations were needed to reduce $\mathbf{F}(\hat{\mathbf{G}}_n)$ to comparable values; very often FRP failed in that it ran the full $N_{\text{max}}$ iterations reaching a point $\hat{\mathbf{G}}$ such that $\mathbf{F}(\hat{\mathbf{G}}) > 100\mathbf{F}(\hat{\mathbf{G}})$, where $\hat{\mathbf{G}}$ was the final iterate of RP under the same conditions. FRP substantially improved with $\varepsilon_3 = .25$, now averaging 1.5 times as many iterations, but still occasionally failing in the manner described above. However, if measured by the distance between final iterates the algorithms appeared to be more equal, $\|\hat{\mathbf{G}} - \tilde{\mathbf{G}}\|$ was usually less than .25 for all algorithms.

The disappointing result was that FRP did not display improved convergence properties over RP with regard to the size of $\sum_{k=0}^{\infty} \|\mathbf{H}_{n-k}\|$. Iterates still appeared to wander through the subspace without much effect on $\mathbf{F}(\mathbf{G}_n)$. It seems that a fifteen dimensional subspace is still too large and quite severe restrictions ($P \approx 5$) must be used to ensure convergence to a minimum.

These results were borne out by the results and comparison of the PA and RP algorithms over all model problems for parameter pairs $(\mathbf{t}, N)$ of (2,40), (2,32) and (1.5,22) and $N_{\text{max}} = 40$. A variety of filters were used but even for the best, using the cutoff of Equation (6.6), the results had a large variance and averaged out so that PA still required about the same number of iterations to achieve the same improvement in $\mathbf{F}(\mathbf{G}_n)$. Again termination was due to reduction of $\mathbf{F}(\hat{\mathbf{G}})$ or $n \gg N_{\text{max}}$ and not to convergence of successive iterates. One reason for this is that for the parameter values $\mathbf{c}$ under consideration almost all significantly nonzero eigenvalues of $\mathbf{P}_c \mathbf{R}_P \mathbf{c}$ have magnitude $\sim 1$ so that the filtered inverse $(\mathbf{P}_c \mathbf{R}_P \mathbf{c})^{-1}$ and the Hermitian
agree on most functions, therefore the search directions produced by the algorithms are usually quite close. It was noticed that the relative performance of PA did appear to improve with decreasing $c$, averaging about $0.9$ for $c=1.5$.

6.5 Relative Performance of Second Order Algorithms

We now present results on the relative performance of algorithms FN, FNP, SQFN and SQFNP compared to RP. They are not comprehensive; considerations such as the number of parameters at the user's disposal, failure of the IMSL subroutines on certain equations and computational cost prevented this. The algorithms were run on each model problem with the parameter pairs $(c,N)$ being $(2,40)$, $(2,32)$ and $(1.5,22)$, $N_{\text{max}}$ ranging from 15 (for computationally costly algorithms) to 50 and with perturbations in the initial guess of $\varepsilon_1 = 20$, 60 and 100. The cutoff parameters $\varepsilon_3$ in Equations (5.26), (5.27), (5.34) and (5.35) used at the n-th iteration, were calculated from quantity

$$\rho = \max\{\min\{4.2\|\lambda_{n-1}\hat{N}_{n-1}\| - .024, .2\}, .002\}.$$  \hspace{1cm} (6.6)

For FN and FNP $\varepsilon_3 = \rho$, for SQFN and SQFNP $\varepsilon_3 = \rho/3$. The IMSL subroutine EIGRS was used to calculate the eigendecomposition of $\nabla^2 F(\hat{G}_n)$, the subroutine LLSQF to calculate the minimum norm least squares solution of Equation (5.42), with the upper bound $\text{ATOL}$ on the condition number of $R$ in the QR factorization specified to be $\varepsilon_3/10$. 

162
The results in Tables 2-5 give first the average performance of RP on each problem in terms of the final value \( F(\hat{G}) \) and the number of iterates \( \hat{n} \) needed to reach this. The remaining entries are the ratios \( \hat{n} / \bar{n} \) where \( \bar{n} \) is the average number of iterations needed by the alternative algorithm to reduce \( F(\hat{G}) \) to below \( F(G) \). In some cases the algorithm on trial failed to reduce \( F(\hat{G}) \) to below 100\% \( F(G) \), such cases are marked by a *. In other cases the alternative algorithm converged but with \( F(\hat{G}) \) larger than \( F(\hat{G}) \) in which case the \( \hat{n} \) of the ratio was altered to the number of iterations required by RP to reduce \( F(\hat{G}) \) to below \( F(\hat{G}) \).

The remaining tables show the effect of (and thus the need for) filtering. Table 6 contains pairs giving the range of the number of eigenvalues preserved at each iteration by the filter in FN and FNP. The final table for SQFN and SQFNP shows two pairs, the upper giving the range of the number of rows in the filtered block equation (5.40), the lower pair indicates the range of the number of columns used by LLSQF. Typically the larger figures in the range occur as \( F(\hat{G}) \) decreases and \( \hat{G} \) converges, however the "stop-start" behaviour of the iterates often caused these measures of filtering to also jump about.

Finally we present graphs (Figures 11-22) of final iterates for RP with parameters \( c=2 \) and \( N=40 \) for each model problem and \( \epsilon_1 = 20\%, 60\% \) and 100\%. As noted previously final iterates varied much less than final values of \( F(\hat{G}) \) and the graphs are typical of all final iterates for these problems.

Some points worth noting on observed algorithm behaviour are given. RP hardly ever halted due to convergence of the sequence \( \hat{G}_n \), iterates exhibited
steady "zig-zagging" with accompanying small but steady decreases in
\( F(\hat{G}_n) \) after an initial burst of reduction in the first 10 or so iterates.
The higher order algorithms did give convergent iterates on a number of
occasions but some limit points were not true minima since at \( \hat{G}_n \) \( \nabla F(\hat{G}_n) \neq 0 \)
but \( \nabla F(\hat{G}_n) \) lay in the null space of the filtered Hessian.

The cutoff parameter given by Equation (6.6) was derived after some
trials and errors and is certainly problem specific. However it performed
satisfactorily for FN, FNP, SQFN and SQFNP in that it generated \( \lambda_n \) that
almost always lay in the interval \([.1,1.5]\), and for FN and FNP started close
to the true solution the expected value of \( \lambda_n \sim 1 \) was always observed.
Moreover it was noted that RP gave the best initial decreases in \( F(\hat{G}_n) \) when
started with \( \varepsilon_1 = 100\% \), so in all algorithms the final search direction \( \hat{H}_n \)
was taken to be a weighted linear combination of the \( \hat{H}_n^1 \) determined by the
particular algorithm and the \( \hat{H}_n^2 \) of RP, i.e.

\[
\hat{H}_n = \hat{H}_n^1 \left( \frac{\varepsilon_1}{1+\varepsilon_3} \right) + \hat{H}_n^2 \left( \frac{\varepsilon_3}{1+\varepsilon_3} \right).
\] (6.7)
7. CONCLUSIONS

The ill posed nature of phase retrieval induces enough variance in the data to prevent the drawing of quantitative conclusions, however qualitative results are apparent. As expected FN and FNP show the best performance when iterates are close to the true solution \( \hat{G}^i \), with FNP displaying greater robustness in regions further away. Close to the solution SQFN and SQFNP do not improve upon RP, but they give the best results when started from a random initial guess. Since SQFN, SQFNP, FN and FNP are all filtered variants of the same basic algorithm it should be possible by a judicious choice of filters to combine the best features of all in one algorithm; to do so requires further study of the block filterings of subsection 5.5.

As measured in terms of the computational cost to reduce \( F(\hat{G}_n) \) to a certain value RP appears to be the certain winner and it is difficult to see how any other algorithm can be improved enough to compete with it. Of the second order algorithms SQFNP seems to be the most efficient, requiring a matrix with \( 0.8 \) the number of rows as SQFN to give approximately the same results. Furthermore the cost can probably be substantially reduced by taking advantage of the sparsity and special structure of the blocks in Equation (5.42).

Tables 6 and 7 confirm that the essential dimension of model problem I is twice the number \( P \) of significantly nonzero eigenvalues of \( \mathcal{P}_c \mathcal{H}_c \). The essential dimension of II and III is harder to estimate although a reasonable upper bound would be \( 2P + N \). Certainly for \( N = 22 \) and 32 the problems
appear to be almost well posed. The tables also indicate that II may be more (locally) ill conditioned than III although this may be attributable to the form of the test functions.

The oscillatory nature of the graphed solutions suggests that further filtering is necessary and that local filtering to the degree done here is insufficient to give a well posed problem. Three options for further conditioning are:

1. Use of a more restrictive local filter

2. Regularization of $F(G)$ by addition of a penalty function e.g.,

$$F_R(G) = F(G) + \alpha \| \frac{dc}{d\omega} \|_2^2$$

(7.1)

3. Use of the algorithm above followed by smoothing of the numerical solutions $\hat{G}^*$. Each option has its attractions: with use of 1 the iterates would be restricted to the span of the first 5 to 10 singular functions, these are known to be smooth and slowly varying so a linear combination would also be reasonably well behaved. Note that this also appears if the interval size $c$ is reduced to around .75 or smaller, thus having less information may actually give a better solution. For 2 a suitably chosen penalty function can easily be accommodated within the theory of algorithms based on projections developed here. Finally visual inspection of the graphs shown indicates that the functions $\hat{G}^*$ are often very perturbed but are basically similar to the true solution $\hat{G}^1$ so that option 3 will often give an acceptable final solution.
APPENDIX A. ON THE EXISTENCE OF MULTIPLE SOLUTIONS TO THE 2-D PHASE RECOVERY PROBLEM

The two-dimensional phase retrieval problem can be stated as:

Let A and B be bounded subsets of $\mathbb{R}^2$. Given the information that $g(z_1, z_2)$ is the Fourier transform of a function $G(\omega_1, \omega_2)$ with support contained in B and the values $m(x_1, x_2) = |g(x_1, x_2)|$ on A, find the phase of $g$ on A and reconstruct $G$.

The phase retrieval problem does not necessarily have a unique solution. The aim of this appendix is to derive from a given solution pair $g$ and $G$ necessary conditions for the existence (and characterization) of alternative solution pairs $h$ and $H$. An intuitive start for such an investigation is the simple result that if $f(z)$ is an analytic function of the complex variable $z$ then so is $f^*(z^*)$ and $f(z)$ and $f^*(z^*)$ have the same modulus for real $z$. This suggests that if a solution $g$ can be factored into a product of analytic functions $g_1$ and $g_2$ then the entire function of two complex variables $h(z_1, z_2) \equiv g_1(z_1, z_2)g_2^*(z_1^*, z_2^*)$ is also a possible solution.

The main result of this appendix is to show that all possible alternative solutions must be of this simple form.

A.1 One-Dimensional Results

The following results from the theory of functions of a single complex variable are required.

Theorem A.1: Paley-Wiener Theorem [20]. Let $B \equiv [b_1, b_2]$ be a bounded interval in $\mathbb{R}$. Then for any $G \in L^2(\mathbb{R})$, $G \neq 0$ on $B$, the transform
\[ g(z) = \int_{b_1}^{b_2} e^{izu} G(u) \, du \quad (A.1) \]

is an entire function and there exist constants \( \alpha \) and \( \beta \) such that

\[ |g(z)| < \begin{cases} 
\alpha e^{-b_1 \text{Im } z} & \text{if Im } z > 0 \\
\beta e^{b_2 \text{Im } z} & \text{if Im } z < 0
\end{cases} \quad (A.2) \]

The next result is the fundamental theorem providing the necessary machinery to characterize all possible solutions to the phase problem both in one and two dimensions. Although independently derived by many authors [3,21], it appears to have been first stated by Akutowicz [8,9]. We state the result as originally presented there.

**Theorem A.2.** Let \( \mathcal{C} \) be the class of all functions \( g \in L^2(\mathbb{R}) \) satisfying

1. \( |g(x)| = m(x) \neq 0 \quad \forall x \in \mathbb{R} \)
2. \( g = \mathcal{G} G \) where support of \( G \) is contained in a bounded interval \( B \) of \( \mathbb{R} \).

Then any two functions \( g, h \in \mathcal{C} \) are related by equations of the form

\[ h(z) = e^{i(\alpha + \beta z)} B(z) g(z) \quad (A.3) \]

where the \( z_\lambda^* \) form some subset of the zeros of \( g(z) \). The function \( (z-z_\lambda^*)/(z-z_\lambda) \) is termed a Blaschke factor.
Lemma A.1: A necessary and sufficient condition for the infinite product \( B(z) \) to converge is that [9]

\[
\sum_{k=1}^{\infty} \frac{|\text{Im} z_k|}{1+|z_k|^2} < \infty \tag{A.4}
\]

A sufficient condition for the convergence of the infinite sum is that \( G(u) \) have only a finite number of jump discontinuities over \( B \).

It is easily shown that if \( G \) has support in an interval \( B \) and if \( h(z) = B(z)g(z) \), then \( H(u) = (\mathcal{F}^{-1} h)(u) \) also has support in \( B \). Therefore combining theorems A.1, A.2 and lemma A.1 gives the following statement on existence of multiple solutions to the 1-D phase retrieval problem.

Theorem A.3: Let \( A \) and \( B \) be bounded intervals in \( \mathbb{R} \) with a modulus \( m(x) \) specified over \( A \) and a solution pair \( g \) and \( G \) be given to the corresponding 1-D phase problem. Then if \( m(x) \) has an extension to the entire real line such that \( m(x) > 0 \) and \( G(u) \) has only finite jump discontinuities over \( B \) as well as being nonzero in neighborhoods of the endpoints of \( B \), then all other solution pairs \( h, H \) are given by

\[
h(z) = e^{i\alpha} B(z)g(z) \tag{A.5}
\]

\[
H(u) = (\mathcal{F}^{-1} h)(u) \tag{A.6}
\]

where \( B(z) \) is any finite or infinite product of Blaschke factors and \( \alpha \in \mathbb{R} \).

The conditions of Theorem A.3 imply that a solution \( g(z) \) has a Hadamard factorization
where $a, b \in \mathbb{R}$, which may be rewritten as

$$g(z) = \left| g(0) \right| e^{i(\alpha + \beta z)} \prod_{\ell=1}^{\infty} \left( 1 - \frac{z}{z_{\ell}} \right)$$

and

$$g(z) = \left[ \left| g(0) \right| e^{i(\alpha + \beta z)} \prod_{\ell \in \Lambda} \left( 1 - \frac{z}{z_{\ell}} \right) \right] \prod_{\ell \in (\mathbb{N} - \Lambda)} \left( 1 - \frac{z}{z_{\ell}} \right)$$

$$\equiv g_{1}(z)g_{2}(z)$$

where $\Lambda$ is a subset of the natural numbers $\mathbb{N}$. Theorem A.3 thus states that any solution $h(z)$ has the form

$$h(z) = e^{i\gamma} g_{1}(z)g_{2}^{*}(z^{*}), \quad \gamma \in \mathbb{R}$$

i.e. that all possible solutions are in one to one correspondence with all possible factorizations of $g(z)$.

### A.2 Extension to Two Dimensions

Conditions that multiple solutions to the 2-D phase retrieval problem must satisfy can be deduced from the 1-D results. To begin, suppose that the problem as stated has a solution pair $g(z_{1}, z_{2}), G(\omega_{1}, \omega_{2})$ where $z = x + iy$ and $\omega = u + iv$ denote variables in the transform and physical domains, such that $G(u_{1}, u_{2})$ has only a finite number of jump discontinuities over $B$. Then

**Lemma A.2**: $g(z_{1}, z_{2})$ is an entire function of the complex variables $z_{1}, z_{2}$ of exponential growth.
Proof. After defining the quantities

\[ u_1^+ = \max\{u_1 : (u_1, u_2) \in B\}, \quad u_2^+ (u_1) = \max\{u_2 : (u_1, u_2) \in B\} \]

\[ u_1^- = \min\{u_1 : (u_1, u_2) \in B\}, \quad u_2^- (u_1) = \min\{u_2 : (u_1, u_2) \in B\} \]

g(z_1, z_2) can be expressed as

\[ g(z_1, z_2) = \int_{u_1^-}^{u_1^+} \int_{u_2^-}^{u_2^+ (u_1)} du_1 e^{iz_1 u_1} du_2 e^{iz_2 u_2} G(u_1, u_2). \quad (A.10) \]

Writing \( g(z_1, z_2) \) as \( g_2 (z_1) \) to indicate that \( g(z_1, z_2) \) is to be considered as a function of \( z_1 \) only with \( z_2 \) fixed gives

\[ g_2 (z_1) = \int_{u_1^-}^{u_1^+} \bar{G}(u_1, z_2) e^{-iz_1 u_1} du_1. \quad (A.11) \]

Therefore by Theorem A.1 \( g_2 (z_1) \) is an entire function of \( z_1 \) of exponential growth. A similar procedure shows that \( g_1 (z_2) \) is an entire function of \( z_2 \).

An immediate consequence of this lemma is that if a solution exists then the modulus \( m(x_1, x_2) \) has an analytic extension to all of \( \mathbb{R}^2 \), which, under the assumption that \( m(x_1, x_2) > 0 \ \forall (x_1, x_2) \), is unique.

Now let \( h(z_1, z_2), H(\omega_1, \omega_2) \) be any other solution pair to this problem; then \( h_2 (x_1) \) and \( g_2 (x_1) \) must have the same modulus \( m_2 (x_1) \) over the set \( B_{x_2} = \{x_1 : (x_1, x_2) \in B\} \); i.e. \( h_2 (z_1) \) and \( g_2 (z_1) \) are both solutions to \( \mathbb{R}^2 \)-phase retrieval problem. Therefore by Theorem A.2, Lemma A.1 and the above assumptions on \( g, G \) and \( m \) we have that
where $\alpha(x_2)$ and $\beta(x_2)$ are constants dependent on $x_2$ only and $B(z)$ is an infinite product of Blashke factors formed from the zeros $\rho_k(x_2)$ of $g_{x_2}(z_1)$. Thus if $\eta_k(x_2)$ are the zeros of $h_{x_2}(z_1)$ and the sets $X_{x_2}$, $Y_{x_2}$ are defined by

$$X_{x_2} = \bigcup_{\ell=1}^{\infty} \{\rho_k(x_2), x_2\}, \quad Y_{x_2} = \bigcup_{\ell=1}^{\infty} \{\eta_k(x_2), x_2\} \quad (A.13)$$

it follows that

$$Y_{x_2} \subseteq X_{x_2} \cup X^*, \quad X_{x_2} \subseteq Y_{x_2} \cup Y^* \quad (A.14)$$

Let $X$ and $Y$ be the sets of zeros of $g$ and $h$ respectively. It is known [22] that the zeros of a function of $n$ complex variables form an analytic set of dimension $(n-1)$ which in turn is the union of analytic manifolds of dimension $(n-1)$ and a set of singular points of dimension at most $(n-2)$. Therefore for almost all $x_2$ the points $({\rho}_k(x_2), x_2)$ and $({\eta}_k(x_2), x_2)$ are members of analytic submanifolds of $X$ and $Y$ respectively; that is for each $k$ there exist maps

$$\phi_k : \mathbb{C} \rightarrow X \quad \phi_k(x_2) = (\rho_k(x_2), x_2) \quad (A.15)$$

$$\psi_k : \mathbb{C} \rightarrow Y \quad \psi_k(x_2) = (\eta_k(x_2), x_2)$$

such that for almost all $x_2$, $\phi_k$ and $\psi_k$ are analytic in a neighborhood $N_k(x_2)$ of $x_2$. Furthermore for almost all $x_2$, the maps
\{\phi_k, \psi_k\}_{k=1}^{\infty}$ are analytic in neighborhoods $N_k(x_2)$ of $x_2$ (note the dependence of $N_k(x_2)$ on $k$).

We now suppose that the zeros $\rho_k(x_2)$ are well separated, i.e.

$$\rho_k(x_2) \neq \rho_{k'}(x_2) \quad \text{or} \quad \rho_k^*(x_2) \quad \forall k, l \ k \neq l \quad (A.16)$$

Since by (A.12) $y_k(x_2) = \rho_k(x_2)$ or $\rho_k^*(x_2)$ the analyticity of $\phi_k$ and $\psi_k$ imply that

$$y_k(z) = \rho_k(z) \quad \text{or} \quad \rho_k^*(z) \quad \forall z \in N_k(x_2). \quad (A.17)$$

Therefore if $X_1 \subseteq X$ and $Y_1 \subseteq Y$ are the analytic extensions of the neighborhoods of $x_2$ and $x_2$ to analytic manifolds then

$$X_1 \subseteq Y_1 \cup Y_1^* \quad \text{and} \quad Y_1 \subseteq X_1 \cup X_1^* \quad \quad (A.18)$$

If $Y_1 \neq X_1$ then there exists a point $y \in Y_1$ and an associated neighborhood $N(y) \subseteq Y_1$ such that $N(y) \subseteq X_1^* - X_1$. By analytic continuation $N(y)$ can be extended to an analytic manifold $Y_2$ such that $Y_2 \subseteq X_1^*$ and $Y_2 \subseteq Y_1$. If $Y_2 = Y_1$ then $Y_1 = X_1^*$ otherwise $Y_1$ must be decomposable into two analytic submanifolds $Y_2$ and $Y_3$ such that $Y_2 \subseteq X_1^*$ and $Y_3 \subseteq X_1^*$ or $X_1$.

Thus, apart from alternative solutions generated by varying $\alpha(x_2)$ and $\beta(x_2)$, a necessary condition that an alternative solution $h$ to the 2-D phase problem must satisfy is that some of its zeros be the complex conjugates of those of the original solution. Although this is the same mechanism by which an infinite number of alternative solutions to the 1-D phase problem are generated, the zeros must now satisfy the condition that they form a union of one-dimensional analytic manifolds as opposed to a union of zero
dimensional manifolds, that is a collection of connected analytic line segments as opposed to a collection of isolated points. If an alternative solution exists then either the whole manifold $X$ has been "flipped" to its conjugate, or it has been "torn" and only partially flipped. The connected nature of $X_1$ implies that the existence of "dotted lines" along which tears may be made is very unlikely; this compares to the isolated points in the 1-D problem, each of which may be flipped independently of the other.

Given this condition the form of an alternative solution may be determined. Let $X_1$ be decomposable into submanifolds $X_2, X_3$ and define

$$X_1, x_2 \equiv X_1 \cap \{(z_1, x_2): x_2 \text{ fixed, } z_1 \in \mathbb{C}\}.$$  \hspace{1cm} (A.19)

then the function $g_{x_2}(z_1)$ may be written as the product

$$g_{x_2}(z_1) = g_{1, x_2}(z_1)g_{2, x_2}(z_1)$$ \hspace{1cm} (A.20)

where

$$g_{2, x_2}(z_1) = \prod_{(\rho_k(x_2), x_2) \in X_2, x_2} \left(1 - \frac{z_1}{\rho_k(x_2)}\right).$$ \hspace{1cm} (A.21)

By (A.12) $h_{x_2}(z_1)$ may be written as

$$h_{x_2}(z_1) = g_{1, x_2}(z_1)g^*_{2, x_2}(z^*_{1}).$$ \hspace{1cm} (A.22)

If $h(z_1, z_2)$ exists it is the analytic extension of $h_{x_2}(z_1)$, therefore

$$h(z_1, z_2) = g_1(z_1, z_2)g^*_2(z^*_1, z^*_2).$$

We have not been able to show that this necessary condition for alternative solutions is also sufficient; i.e. given submanifolds $X_2, X_3$ and
the decomposition of Eq. (A.20) that the $h_{x_2}(z_1)$ of Eq. (A.22) may be analytically continued to a function $h(z_1,z_2)$. One source of trouble is the dependence of $N_k(x_2)$ on $k$; it is possible that for every $x_2$, $\prod_{k=1}^{\infty} N_k(x_2) = \emptyset$.

Then although each zero $(\rho_k(x_2),x_2)$ is analytic in a neighborhood of $x_2$, there does not exist a neighborhood over which all zeros are uniformly analytic, and therefore a neighborhood over which the product of Eq. (A.21) is provably analytic.

If the cardinalities of $X_1$ and $X_2$ are finite, e.g. $g(z_1,z_2)$ is a polynomial, then sufficiency can be shown. In the polynomial case decomposability of $X_1$ into $X_2$ and $X_3$ is equivalent to a factorization of the polynomial. However almost all polynomials of two variables are irreducible so that such a factorization and decomposition does not exist, therefore alternative solutions do not exist. Irreducibility extends to general functions of two variables with infinite sets of zeros, so that exact alternative solutions are most unlikely in 2-D phase retrieval. This result on polynomials and its implications is also presented in [23].

A.3 The Support of Alternative Solutions

In the previous subsection conditions that alternative solutions $g$ and $h$ must satisfy in order that $|g(x_1,x_2)| = |h(x_1,x_2)|$ were derived. It remains to derive necessary conditions on $g$ and $h$ so that $(\text{support } G) = (\text{support } H)$. The first is that $\beta(x_2) \equiv 0$ in Eq. (A.12). This follows by noting that if $G(u_1^+,u_2^+)$ is nonzero in neighborhoods of points $(u_1^+,u_2^+), (u_1^-,u_2^-) \in B$ then the function $\tilde{G}(u_1,x_2)$ of Eq. (A.11) will be nonzero in
neighborhoods of \( u_1 = u_1^+ \) and \( u_1 = u_1^- \) for almost all \( x_2 \). So by Theorem A.3
\[ h_{x_2}(z_1) \] is the transform of a function \( \tilde{H}(u_1, x_2) \) with support in \((u_1^-, u_1^+)\) if and only if \( B(x_2) \equiv 0 \).

A second condition follows from noting that the boundedness of the set \( B \) implies that \( h(z_1, z_2) \) is of exponential growth in \( z_2 \), so that \( \alpha(x_2) \) must only be a linear function of \( x_2 \). Summarizing these results and those of the previous section gives the next theorem.

**Theorem A.4:** Let \( g, G \) be a solution pair to the 2-D phase retrieval problem. Then any other solution pair \( h, H \) must have the form
\[ h(z_1, z_2) = e^{i(\alpha_1 + \alpha_2 z_2)} g_1(z_1, z_2)^* g_2(z_1, z_2)^* \] (A.23)
where \( g_1 g_2 \) is a factorization of \( g \).

We have been unable to complement these necessary conditions for equality of support with sufficient conditions equivalent to those for the 1-D problem. The difficulty seems to lie in determining the role of the geometry of \( B \); we give two examples

1. The first example concerns convexity and is taken from Huisier and Torn [24]. Let \( g, G \) be a solution pair, then after the change of variables to the new orthogonal coordinate systems \((s_1, s_2), (t_1, t_2)\) with
\[ s_1 = u_1 \cos \psi + u_2 \sin \psi \quad t_1 = x_1 \cos \psi + x_2 \sin \psi \]
\[ s_2 = -u_1 \sin \psi + u_2 \cos \psi \quad t_2 = -x_1 \sin \psi + x_2 \cos \psi \] (A.24)
and definition of the quantities
\[ s_2^+(s_1) = \max \{ s_2 : (s_1, s_2) \in B \} \]
\[ s_2^-(s_1) = \min \{ s_2 : (s_1, s_2) \in B \} \]
\[ s_1^+(\psi) = \max \{ s_1 : (s_1, s_2) \in B \} \]
\[ s_1^-(\psi) = \min \{ s_1 : (s_1, s_2) \in B \} \]  
(A.25)

the relationship \( g = \mathcal{F} G \) may be rewritten as

\[ g(t_1, t_2) = \int_{s_1^-(\psi)}^{s_1^+(\psi)} ds_1 e^{s_1 t_1} \int_{s_2^-(s_1)}^{s_2^+(s_1)} ds_2 e^{s_2 t_2} G(s_1, s_2). \]  
(A.26)

For fixed \( t_2 \) the growth rate in \( g_{t_2}(t_1) \) is determined by \( s_1^+(\psi) \) and \( s_1^-(\psi) \). Knowing these values for all \( \psi \) is equivalent to knowing all supporting hyperplanes for the set \( B \), which by duality arguments from linear algebra is equivalent to knowing the convex hull of \( B \). If \( h \) and \( H \) is any other solution pair then \( h_{t_2}(t_1) \) must have the same growth as \( g_{t_2}(t_1) \) otherwise \( H \) has support outside of the convex hull of \( B \).

If \( g \) has a factorization \( g_1 g_2 \) such that the growth of \( g_2 \) is always dominated by that of \( g_1 \) (e.g. \( g_2 \) is a polynomial) then the alternative function

\[ h(z_1, z_2) = g_1(z_1, z_2) g_2^*(z_1^*, z_2^*) \]  
(A.27)

has the same modulus as \( g \) and support in the convex hull of \( B \). If \( B \) is convex then \( h \) is an alternative solution, if \( B \) is not convex then it is possible that the support of \( H \) is not \( B \) even though still in the convex hull.
2. Let $g, G$ be a solution pair, then it is trivial to show that the inverse transform of $g^*(z_1^*, z_2^*)$ is $G^*(-\omega_1, -\omega_2)$ which has support $-B$. So a sufficient condition that $g^*(z_1^*, z_2^*)$ be an alternative solution is that $B = -B$, i.e., $B$ is invariant under rotation by $180^\circ$.

Example 1 suggests that convexity of $B$ is necessary for existence of an alternative solution and taken with example 2 suggests that for a factorization $g$ into $g_1 g_2$ and an alternative solution $h$ of Eq. (A.27) then $B$ must have symmetries linked in some fashion to those directions in which growth of $g_2$ dominates $g_1$.

A.4 Conclusions

Nonuniqueness in the phase retrieval problem in two dimensions appears to depend on two conditions: (1) that the zero space of $g$ be decomposable into a union of several submanifolds, (2) that $B$ possesses a suitable combination of convexity and symmetry. Both conditions will, in general, be difficult to satisfy compared to the 1-D phase retrieval problem. Only in the case of symmetries that effectively reduce $g(z_1, z_2)$ to a function of one variable (e.g., the possession of radial symmetry investigated in [25]) will the manifold have an infinite decomposition as appears in the 1-D problem. In most cases it will be indivisible. Likewise the general two-dimensional bounded set has considerably more degrees of freedom than the one-dimensional bounded set, the interval, consequently it has far fewer symmetries. Therefore in general the 2-D phase retrieval problem will have a unique solution if one exists.
REFERENCES

1. R. Barakat and G. Newsam, "Algorithms for reconstruction of partially known, bandlimited Fourier transform pairs from noisy data: I the prototypical linear problem." Submitted for publication to J. Integral Eqs.


Table 1. Sensitivity of solution $\hat{G}$ to a relative perturbation of $\varepsilon_2$ in the data.

<table>
<thead>
<tr>
<th>Model Problem</th>
<th>Test Function $\hat{G}$</th>
<th>Percentage error in $\hat{G}$</th>
<th>$F(G)$</th>
<th>Percentage error in $\hat{G}$</th>
<th>$F(G)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$\hat{G}^1$</td>
<td>1%</td>
<td>$1 \times 10^{-5}$</td>
<td>9%</td>
<td>$6 \times 10^{-4}$</td>
</tr>
<tr>
<td>I</td>
<td>$\hat{G}^2$</td>
<td>2%</td>
<td>$9 \times 10^{-6}$</td>
<td>10%</td>
<td>$3 \times 10^{-5}$</td>
</tr>
<tr>
<td>II</td>
<td>$\hat{G}^1$</td>
<td>.75%</td>
<td>$8 \times 10^{-4}$</td>
<td>4%</td>
<td>$4 \times 10^{-2}$</td>
</tr>
<tr>
<td>III</td>
<td>$\hat{G}^2$</td>
<td>1.5%</td>
<td>$2 \times 10^{-4}$</td>
<td>13%</td>
<td>$3 \times 10^{-3}$</td>
</tr>
</tbody>
</table>
Table 2. Relative performance of algorithms on model problem I with test function $G^1$.

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>N</th>
<th>$10^{-6}$</th>
<th>RP</th>
<th>FN</th>
<th>FNP</th>
<th>SQFN</th>
<th>SQFNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>20</td>
<td>$8 \times 10^{-6}$</td>
<td>6</td>
<td>.325</td>
<td>.325</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>22</td>
<td>32</td>
<td>$9 \times 10^{-6}$</td>
<td>14</td>
<td>.275</td>
<td>.275</td>
<td>.775</td>
<td>1.3</td>
</tr>
<tr>
<td>22</td>
<td>40</td>
<td>$5 \times 10^{-6}$</td>
<td>5</td>
<td>.80</td>
<td>.80</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>22</td>
<td>60</td>
<td>$9 \times 10^{-6}$</td>
<td>34</td>
<td>.20</td>
<td>.10</td>
<td>.275</td>
<td>.40</td>
</tr>
<tr>
<td>60</td>
<td>32</td>
<td>$1 \times 10^{-5}$</td>
<td>7</td>
<td>.575</td>
<td>.575</td>
<td>2.0</td>
<td>1.4</td>
</tr>
<tr>
<td>60</td>
<td>40</td>
<td>$9 \times 10^{-6}$</td>
<td>10</td>
<td>.40</td>
<td>.40</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>22</td>
<td>100</td>
<td>$5 \times 10^{-4}$</td>
<td>50</td>
<td>*</td>
<td>.65</td>
<td>.45</td>
<td>.50</td>
</tr>
<tr>
<td>22</td>
<td>32</td>
<td>$7 \times 10^{-4}$</td>
<td>30</td>
<td>.65</td>
<td>.45</td>
<td>.40</td>
<td>.40</td>
</tr>
<tr>
<td>22</td>
<td>40</td>
<td>$1.4 \times 10^{-3}$</td>
<td>40</td>
<td>*</td>
<td>.20</td>
<td>.30</td>
<td>.30</td>
</tr>
</tbody>
</table>
Table 3. Relative performance of algorithms on model problem I with test function $G^2$

<table>
<thead>
<tr>
<th>$\varepsilon_1$</th>
<th>N</th>
<th>FN</th>
<th>FNP</th>
<th>SQFN</th>
<th>SQFNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>1.9 $\times 10^{-5}$</td>
<td>50</td>
<td>1.5</td>
<td>1.0</td>
<td>.40</td>
</tr>
<tr>
<td>60</td>
<td>1.5 $\times 10^{-5}$</td>
<td>20</td>
<td>*</td>
<td>.825</td>
<td>.875</td>
</tr>
<tr>
<td>40</td>
<td>3 $\times 10^{-5}$</td>
<td>40</td>
<td>*</td>
<td>1.0</td>
<td>.35</td>
</tr>
<tr>
<td>100</td>
<td>5 $\times 10^{-5}$</td>
<td>50</td>
<td>1.5</td>
<td>.30</td>
<td>.60</td>
</tr>
<tr>
<td>40</td>
<td>4 $\times 10^{-4}$</td>
<td>20</td>
<td>1.6</td>
<td>.45</td>
<td>.35</td>
</tr>
<tr>
<td>100</td>
<td>8 $\times 10^{-4}$</td>
<td>40</td>
<td>.50</td>
<td>.50</td>
<td>.425</td>
</tr>
</tbody>
</table>

184
Table 4. Relative performance of algorithms on model problem II.

<table>
<thead>
<tr>
<th>$\varepsilon_1$</th>
<th>N</th>
<th>$9 \times 10^{-6}$</th>
<th>FN</th>
<th>FNP</th>
<th>SQFN</th>
<th>SQFNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>40</td>
<td>9</td>
<td>.50</td>
<td>.50</td>
<td>-</td>
<td>1.0</td>
</tr>
<tr>
<td>60</td>
<td>40</td>
<td>9</td>
<td>.50</td>
<td>.50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100</td>
<td>40</td>
<td>$1 \times 10^{-3}$</td>
<td>.25</td>
<td>.20</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 5. Relative performance of algorithms on model problem III.

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>N</th>
<th>$9 \times 10^{-6}$</th>
<th>FN</th>
<th>FNP</th>
<th>SQFN</th>
<th>SQFNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>22</td>
<td>23</td>
<td>.20</td>
<td>.20</td>
<td>.50</td>
<td>.50</td>
</tr>
<tr>
<td>60</td>
<td>22</td>
<td>$5 \times 10^{-4}$</td>
<td>.50</td>
<td>.30</td>
<td>.35</td>
<td>.45</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>$3 \times 10^{-4}$</td>
<td>2.0</td>
<td>.45</td>
<td>.50</td>
<td>.60</td>
</tr>
<tr>
<td>100</td>
<td>22</td>
<td>$2 \times 10^{-3}$</td>
<td>3.0</td>
<td>.80</td>
<td>.325</td>
<td>.325</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>$6 \times 10^{-2}$</td>
<td>*</td>
<td>1.0</td>
<td>.50</td>
<td>.40</td>
</tr>
</tbody>
</table>
Table 6. The range of nonzero eigenvalues in algorithms FN and FNP. Note that for \( c = 1.5, 2 \) the finite Fourier transform has at most 15 and 22 eigenvalues respectively with magnitudes greater than .002.

<table>
<thead>
<tr>
<th>N</th>
<th>22</th>
<th>22</th>
<th>22</th>
<th>32</th>
<th>32</th>
<th>40</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon_1 )</td>
<td>20</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>( G^1 ) FN</td>
<td>18-19</td>
<td>18-25</td>
<td>14-28</td>
<td>30</td>
<td>24-39</td>
<td>30-34</td>
<td>25-37</td>
</tr>
<tr>
<td>( G^1 ) FNP</td>
<td>16-18</td>
<td>17-18</td>
<td>14-21</td>
<td>28</td>
<td>23-28</td>
<td>30-33</td>
<td>23-37</td>
</tr>
<tr>
<td>Model Problem I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( G^2 ) FNP</td>
<td>17</td>
<td>14-24</td>
<td>16-23</td>
<td>24-29</td>
<td>25-30</td>
<td>24-26</td>
<td>28-29</td>
</tr>
<tr>
<td>Model Problem II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>55-57</td>
<td>54-60</td>
</tr>
<tr>
<td>FNP</td>
<td>30-31</td>
<td>30-33</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>55-57</td>
<td>55-62</td>
</tr>
<tr>
<td>Model Problem III</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>33-44</td>
<td>33-44</td>
<td>33-44</td>
<td>47-64</td>
<td>57-63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FNP</td>
<td>31-37</td>
<td>31-42</td>
<td>31-42</td>
<td>46-61</td>
<td>52-58</td>
<td>55-75</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 7. The range of matrix dimensions and effective dimensions of the block matrices in algorithms SQFN and SQFNP. The rows marked A contain the range of block matrix row sizes; the rows marked B contain the range of the number of columns used by the routine LLSQF.

<table>
<thead>
<tr>
<th></th>
<th>G^1 SQFN</th>
<th></th>
<th>G^1 SQFNP</th>
<th></th>
<th>MODEL PROBLEM I</th>
<th></th>
<th>G^2 SQFN</th>
<th></th>
<th>G^2 SQFNP</th>
<th></th>
<th>MODEL PROBLEM II</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>32</td>
<td>32</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\varepsilon_1)</td>
<td>20</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G^1</td>
<td>SQFN</td>
<td>A</td>
<td>40-44</td>
<td>30-44</td>
<td>32-44</td>
<td>36-64</td>
<td>44-64</td>
<td>49-80</td>
<td>54-80</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>12-23</td>
<td>12-17</td>
<td>9-22</td>
<td>9-25</td>
<td>12-26</td>
<td>12-29</td>
<td>13-29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G^1</td>
<td>SQFNP</td>
<td>A</td>
<td>23-32</td>
<td>28-36</td>
<td>27-37</td>
<td>36-50</td>
<td>38-51</td>
<td>46-61</td>
<td>47-62</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>10-18</td>
<td>11-17</td>
<td>9-24</td>
<td>14-20</td>
<td>13-28</td>
<td>12-26</td>
<td>15-32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEL PROBLEM I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G^2</td>
<td>SQFN</td>
<td>A</td>
<td>28-44</td>
<td>36-44</td>
<td>32-44</td>
<td>46-64</td>
<td>53-63</td>
<td>72-80</td>
<td>75-80</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>10-22</td>
<td>13-21</td>
<td>10-23</td>
<td>8-27</td>
<td>12-25</td>
<td>16-33</td>
<td>23-33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G^2</td>
<td>SQFNP</td>
<td>A</td>
<td>26-29</td>
<td>30-35</td>
<td>29-36</td>
<td>45-53</td>
<td>41-51</td>
<td>54-60</td>
<td>58-65</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>14-19</td>
<td>10-16</td>
<td>10-20</td>
<td>18-24</td>
<td>11-28</td>
<td>11-27</td>
<td>20-30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEL PROBLEM II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQFN</td>
<td>A</td>
<td>87-88</td>
<td>79-88</td>
<td>72-88</td>
<td>91-128</td>
<td>120-126</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>36-44</td>
<td>25-44</td>
<td>18-44</td>
<td>31-64</td>
<td>33-43</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQFNP</td>
<td>A</td>
<td>53-65</td>
<td>58-77</td>
<td>61-73</td>
<td>85-104</td>
<td>102-107</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>23-37</td>
<td>17-44</td>
<td>18-44</td>
<td>31-64</td>
<td>30-41</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FIGURE LEGENDS

Fig. 1. Two examples of poor convergence of the alternating projection algorithm induced by ill conditioning.

Fig. 2. The tangent plane $T_i(x)$ to a set $B_i$ at the point $P_i(x)$. $T_i(x)$ locally separates $B_i$ and $S_i(x)$.

Fig. 3. An example of the form of the Hessian operator $H_i(x)$ associated with a sphere $B_i$.

Fig. 4. A solution to model problem I, test function $G^2$ with $F(G) = 8 \cdot 10^{-5}$:
- $\Delta$ reconstructed modulus, $\circ$ reconstructed phase, $---$ exact modulus,
- $-$ exact phase.

Fig. 5. A solution to model problem I, test function $G^2$ with $F(G) = 7 \cdot 10^{-5}$:
- $\Delta$ reconstructed modulus, $\circ$ reconstructed phase, $---$ exact modulus,
- $-$ exact phase.

Fig. 6. A solution to model problem I, test function $G^2$ with $F(G) = 8 \cdot 10^{-4}$:
- $\Delta$ reconstructed modulus, $\circ$ reconstructed phase, $---$ exact modulus,
- $-$ exact phase.

Fig. 7. A solution to model problem III, test function $G^2$ with $F(G) = 2 \cdot 10^{-2}$:
- $\Delta$ reconstructed modulus, $\circ$ reconstructed phase, $---$ exact modulus,
- $-$ exact phase.

Fig. 8. A solution to model problem III, test function $G^2$ with $F(G) = 3 \cdot 10^{-2}$:
- $\Delta$ reconstructed modulus, $\circ$ reconstructed phase, $---$ exact modulus,
- $-$ exact phase.
Fig. 9. A solution to model problem III, test function $G^2$ with $F(G) = 7 \cdot 10^{-2}$: $\triangle$ reconstructed modulus, $\circ$ reconstructed phase, $----$ exact modulus, $---$ exact phase.

Fig. 10. An example of preservation of symmetry, a solution to model problem I, test function $G^1$ with $G_0 = 0$.

Fig. 11. Sample realization solution to model problem I, test function $G^1$, $\varepsilon = 20\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.

Fig. 12. Sample realization solution to model problem I, test function $G^1$, $\varepsilon = 60\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.

Fig. 13. Sample realization solution to model problem I, test function $G^1$, $\varepsilon = 100\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.

Fig. 14. Sample realization solution to model problem III, test function $G^1$, $\varepsilon = 20\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.

Fig. 15. Sample realization solution to model problem III, test function $G^1$, $\varepsilon = 60\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.

Fig. 16. Sample realization solution to model problem III, test function $G^1$, $\varepsilon = 100\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $----$ exact real component, $---$ exact imaginary component.
Fig. 17. Sample realization solution to model problem I, test function $G^2$, 
$\varepsilon = 20\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.

Fig. 18. Sample realization solution to model problem I, test function $G^2$, 
$\varepsilon = 60\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.

Fig. 19. Sample realization solution to model problem I, test function $G^2$, 
$\varepsilon = 100\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.

Fig. 20. Sample realization solution to model problem III, test function $G^2$, 
$\varepsilon = 20\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.

Fig. 21. Sample realization solution to model problem III, test function $G^2$, 
$\varepsilon = 60\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.

Fig. 22. Sample realization solution to model problem III, test function $G^2$, 
$\varepsilon = 100\%$: O reconstructed phase, \( \Delta \) reconstructed modulus, 
---- exact modulus, ----- exact phase.
Fig. 1. Two examples of poor convergence of the alternating projection algorithm induced by ill conditioning.
Fig. 2. The tangent plane $T_i(x)$ to a set $B_i$ at the point $P_i(x)$. $T_i(x)$
locally separates $B_i$ and $S_i(x)$. 
Fig. 3. An example of the form of the Hessian operator $\mathcal{H}_i(x)$ associated with a sphere $B_i$. 
Fig. 4. A solution to model problem I, test function $\tilde{G}^2$ with $F(\tilde{G}) = 8 \cdot 10^{-5}$:

- $\triangle$ reconstructed modulus,
- $\circ$ reconstructed phase,
- --- exact modulus,
- --- exact phase.
Fig. 5. A solution to model problem I, test function $\tilde{G}^2$ with $P(\tilde{G}) = 7 \cdot 10^{-5}$:

- $\triangle$ reconstructed modulus,
- $\circ$ reconstructed phase,
- $-$ exact modulus,
- $-$ exact phase.
Fig. 6. A solution to model problem I, test function $\tilde{G}^2$ with $F(\tilde{G}) = 8 \cdot 10^{-4}$:

- $\Delta$ reconstructed modulus,
- $\circ$ reconstructed phase,
- ---- exact modulus,
- ----- exact phase.
Fig. 7. A solution to model problem III, test function $G^2$ with $F(G) = 2 \times 10^{-2}$:

- $\Delta$ reconstructed modulus,
- $\circ$ reconstructed phase,
- $\ldots\ldots$ exact modulus,
- $\ldots\ldots$ exact phase.
Fig. 8. A solution to model problem III, test function $\frac{3}{G^2}$ with $F(\frac{3}{G}) = 3 \cdot 10^{-2}$:

$\triangle$ reconstructed modulus, $\circ$ reconstructed phase, $\ldots$ exact modulus,

$-$ exact phase.
Fig. 9. A solution to model problem III, test function $G^2$ with

$F(3) = 7 \times 10^{-2}$: $\Delta$ reconstructed modulus, $\circ$ reconstructed phase,

--- exact modulus, --- exact phase.
Fig. 10. An example of preservation of symmetry, a solution to model problem I, test function $G^1$ with $\hat{G}_0 = 0$. 
Fig. 11. Sample realization solution to model problem I, test function $\tilde{G}$, $\varepsilon = 20\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $-$ exact real component, $-$--$-$ exact imaginary component.
Fig. 12. Sample realization solution to model problem I, test function $\tilde{G}^1$, $\varepsilon = 60\%$: $\triangle$ reconstructed real component, $\circ$ reconstructed imaginary component, $-$--$-$--$-$ exact real component, $---$--$-$ exact imaginary component.
Fig. 13. Sample realization solution to model problem I, test function $\tilde{C}_1$,

$\varepsilon = 100\%$: $\Delta$ reconstructed real component, $\bigcirc$ reconstructed imaginary component, $\text{---}$ exact real component, $\text{-----}$ exact imaginary component.
Fig. 14. Sample realization solution to model problem III, test function $\tilde{G}^1$, $\varepsilon = 20\%$: △ reconstructed real component, ○ reconstructed imaginary component, - - - - exact real component, ---- exact imaginary component.
Fig. 15. Sample realization solution to model problem III, test function $c^1$.

$\epsilon = 60\%$: $\Delta$ reconstructed real component, $\circ$ reconstructed imaginary component, $\ldots$ exact real component, $\ldots\ldots$ exact imaginary component.
Fig. 16. Sample realization solution to model problem III, test function \( G \), \( \epsilon = 100\% \): \( \Delta \) reconstructed real component, \( \circ \) reconstructed imaginary component, \( \ldots \) exact real component, \( \ldots \ldots \) exact imaginary component.
Fig. 17. Sample realization solution to model problem I, test function $G^2$, $\varepsilon = 20\%$: $\circ$ reconstructed phase, $\Delta$ reconstructed modulus, ——— exact modulus, ——— exact phase.
Fig. 18. Sample realization solution to model problem I, test function $\frac{3^2}{G}$, $\epsilon = 60\%$: ○ reconstructed phase, △ reconstructed modulus,

--- exact modulus, —— exact phase.
Fig. 19. Sample realization solution to model problem I, test function $g^2$.

$\varepsilon = 100\%$: ○ reconstructed phase, △ reconstructed modulus,

---- exact modulus, —— exact phase.
Fig. 20. Sample realization solution to model problem III, test function $\frac{\xi^2}{\eta}$, $\xi = 20\%$: ○ reconstructed phase, Δ reconstructed modulus, 
--- exact modulus, —— exact phase.
Fig. 21. Sample realization solution to model problem III, test function $G^2$,
$c = 60\%$: ○ reconstructed phase, △ reconstructed modulus,
----- exact modulus, —— exact phase.
Fig. 22. Sample realization solution to model problem III, test function $G^2$, $\varepsilon = 100\%$: ○ reconstructed phase, △ reconstructed modulus,
--- exact modulus, —— exact phase.
## Distribution List

<table>
<thead>
<tr>
<th>Addresses</th>
<th>Number of Copies</th>
</tr>
</thead>
</table>
| Dr. RJ Michalak  
RADC/OCSE | 2 |
| RADC/TSTD  
GRIFFISS AFB NY 13441 | 1 |
| RADC/DAP  
GRIFFISS AFB NY 13441 | 2 |
| Administrator  
DEF TECH INF CTR  
DTIC-DDA  
CAMERON STA BG 5  
ALEXANDRIA VA 22314 | 12 |
| ERIM  
POBox 8618  
Ann Arbor MI 48107 | 1 |
| ERIM  
Attn: Dr. S Robinson  
POBox 8618  
Ann Arbor MI 48107 | 1 |
MRJ Inc.
Attn: Dr K Robinson
1 Blake St.
Needham MA 02192

Itek Corp.
Attn: E. Galat
Optical Systems Division
10 Maguire Rd.
Lexington MA 02173

DARPA/STD
Attn: Lt. Col. Bell
1400 Wilson Blvd.
Arlington VA 22209

AFWL/AARAA
Attn: Dr L Skelnik
Kirtland AFB 87117

Aerospace
Attn: Dr V Manajan
MS M4/979
2250 E. El Segundo Blvd.
El Segundo CA 90245

Lockheed
Attn: Dr S Williams
2251 Hanover St.
Palo Alto CA 94304

Hughes
Attn: Dr R Withrington
Build. E1 MS F/124
El Segundo CA 90245
Attn: Dr D Fried
FOBox 446
Placentia CA 92670

IBM Corp.
Attn: Dr E Silvertooth
C/O Jeanette Miller (Security Officer)
3155 W. Rosecrans Ave Hawthorne CA 90250

Riverside Research
Attn: HALO Library, Mr R Passett
1701 N Fort Meyer Dr
Arlington VA 22209

CARPA/DEO
Attn: R Strunce
1400 Wilson Blvd.
Arlington VA 22209

ERIM
Attn: Dr S Robinson
POBox 8618
Ann Arbor MI 48107

MRJ Inc.
Attn: Dr K Robinson
71 Blake St.
Needham MA 02192

Itek Corp.
Attn: E. Galat
Optical Systems Division
10 Maguire Rd.
Lexington MA 02173

DARPA/STO
Attn: Lt. Col. Bell
1400 Wilson Blvd.
Arlington VA 22209

DL-3
AFWL/ARAA
Attn: Dr L Skolnik
Kirtland AFB 87117

Aerospace
Attn: Dr V Mahajan
MS M4/978
2350 E. El Segundo Blvd.
El Segundo CA 90245

Lockheed
Attn: Dr S Williams
3251 Hanover St.
Palo Alto CA 94304

Hughes
Attn: Dr R Withrington
Build. E1 MS F/124
El Segundo CA 90245

tOSC
Attn: Dr D Fried
POBox 446
Placentia CA 92670

3DM Corp.
Attn: Dr E Silvertooth
C/O Jeanette Miller (Security Officer)
5155 W. Rosecrans Ave Hawthorne CA 90250

Riverside Research
Attn: HALO Library, Mr R Passett
1701 N Fort Meyer Dr
Arlington VA 22209

CRA/DEO
Attn: R Strunce
1400 Wilson Blvd.
Arlington VA 22209

DL-4
Eikonix Corp.
Attn: Dr R Gonsalves
23 Crosby Rd.
Bedford MA 01730

NYIT
Institute for Optics
Prof M Halioua
Old Westbury NY 11568
MISSION

of

Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.