

Phase retrieval algorithms: a comparison

J. R. Fienup

Iterative algorithms for phase retrieval from intensity data are compared to gradient search methods. Both the problem of phase retrieval from two intensity measurements (in electron microscopy or wave front sensing) and the problem of phase retrieval from a single intensity measurement plus a non-negativity constraint (in astronomy) are considered, with emphasis on the latter. It is shown that both the error-reduction algorithm for the problem of a single intensity measurement and the Gerchberg-Saxton algorithm for the problem of two intensity measurements converge. The error-reduction algorithm is also shown to be closely related to the steepest-descent method. Other algorithms, including the input-output algorithm and the conjugate-gradient method, are shown to converge in practice much faster than the error-reduction algorithm. Examples are shown.

I. Introduction

In electron microscopy, wave front sensing, astronomy, crystallography, and in other fields one often wishes to recover phase, although only intensity measurements are made. One is usually interested in determining an object, $f(x)$, which is related to its Fourier transform, $F(u)$, by

$$\begin{aligned} F(u) &= |F(u)| \exp[i\psi(u)] = \mathcal{F}[f(x)] \\ &= \int_{-\infty}^{\infty} f(x) \exp(-i2\pi u \cdot x) dx, \end{aligned} \quad (1)$$

where x is an M -dimensional spatial coordinate, and u is an M -dimensional spatial frequency coordinate. For the majority of interesting problems $M = 2$. In practice one deals with sampled data in the computer, where for the 2-D case, assuming square arrays, $u = (u_1, u_2)$ and $x = (x_1, x_2)$, where u_1, u_2, x_1 , and $x_2 = 0, 1, 2, \dots, N-1$. Then one uses the discrete Fourier transform (DFT)

$$F(u) = \sum_{x=0}^{N-1} f(x) \exp(-i2\pi u \cdot x/N) \quad (2)$$

and its inverse

$$f(x) = N^{-2} \sum_{u=0}^{N-1} F(u) \exp(i2\pi u \cdot x/N), \quad (3)$$

which are, of course, computed using the fast Fourier transform (FFT) method.

For the problem of recovering phase from two intensity measurements, as in electron microscopy and in wave front sensing,

$$f(x) = |f(x)| \exp[i\eta(x)] \quad (4)$$

is complex valued, and one wishes to recover $\psi(u)$ or equivalently recover $\eta(x)$ from measurements of both

$|F(u)|$ and $|f(x)|$. For the problem of recovering phase from a single intensity measurement, as in image recovery from speckle interferometry data in astronomy and from structure factors in crystallography, one wishes to recover $\psi(u)$ or equivalently recover $f(x)$ given a measurement of $|F(u)|$ and the constraint that $f(x)$ be real and non-negative,

$$f(x) \geq 0. \quad (5)$$

A particularly successful approach to solving these problems is the use of the Gerchberg-Saxton algorithm^{1,2} and related algorithms.³⁻⁷ Reference 7 lists a large number of different problems which have been solved by these algorithms. These algorithms involve iterative Fourier transformation back and forth between the object and Fourier domains and application of the measured data or known constraints in each domain.

In what follows a generalized Gerchberg-Saxton algorithm, referred to as the error-reduction algorithm, and its convergence properties are reviewed (Sec. II), and it is shown to be equivalent to the steepest-descent gradient search method (Sec. III). Alternative gradient search methods (Sec. IV) and iterative Fourier transform algorithms (Sec. V) are described and are shown to converge much faster than the error-reduction algorithm for the problem of a single intensity measurement (Sec. VI). Some practical considerations are discussed in Sec. VII. A typical reconstruction experiment is shown in Sec. VIII, and the major conclusions are summarized in Sec. IX.

II. Error-Reduction Algorithm

The Gerchberg-Saxton algorithm was originally invented in connection with the problem of reconstructing phase from two intensity measurements^{1,2} (and for synthesizing phase codes given intensity constraints in each of two domains^{8,9}). The algorithm consists of the following four simple steps: (1) Fourier transform an estimate of the object; (2) replace the modulus of the resulting computed Fourier transform with the measured Fourier modulus to form an estimate of the Fourier transform; (3) inverse Fourier transform the esti-

The author is with Environmental Research Institute of Michigan, Radar & Optics Division, P.O. Box 8618, Ann Arbor, Michigan 48107.

Received 19 February 1982.

0003-6935/82/152758-12\$01.00/0.

© 1982 Optical Society of America.

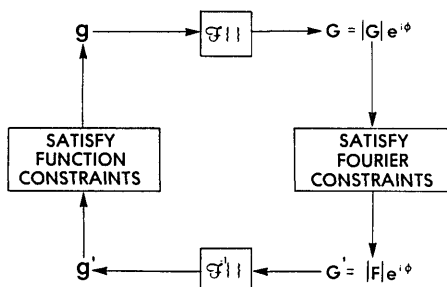


Fig. 1. Block diagram of the error-reduction (Gerchberg-Saxton) algorithm.

mate of the Fourier transform; and (4) replace the modulus of the resulting computed image with the measured object modulus to form a new estimate of the object. In equations this is, for the k th iteration,

$$G_k(u) = |G_k(u)| \exp[i\phi_k(u)] = \mathcal{F}[g_k(x)], \quad (6)$$

$$G'_k(u) = |F(u)| \exp[i\phi_k(u)], \quad (7)$$

$$g'_k(x) = |g'_k(x)| \exp[i\theta'_k(x)] = \mathcal{F}^{-1}[G'_k(u)], \quad (8)$$

$$g_{k+1}(x) = |f(x)| \exp[i\theta_{k+1}(x)] = |f(x)| \exp[i\theta'_k(x)], \quad (9)$$

where g_k , θ_k , G'_k , and ϕ_k are estimates of f , η , F , and ψ , respectively. Here and throughout this paper functions represented by uppercase letters are the Fourier transforms of the functions represented by the corresponding lowercase letters.

As depicted in Fig. 1 the Gerchberg-Saxton algorithm is easily generalized to a large class of problems.^{6,7} The generalized Gerchberg-Saxton algorithm can be used for any problem in which partial constraints (in the form of measured data or information known *a priori*) are known in each of two domains, usually the object (or image) and Fourier domains. One simply transforms back and forth between the two domains, satisfying the constraints in one before returning to the other. This generalization of the Gerchberg-Saxton algorithm will be referred to as the error-reduction algorithm since, as will be shown below, the error decreases at each iteration.

For the most general problem the error-reduction algorithm consists of the following four steps: (1) Fourier transform $g_k(x)$, an estimate of $f(x)$; (2) make the minimum changes in $G_k(u)$, the resulting computed Fourier transform, which allow it to satisfy the Fourier-domain constraints to form $G'_k(u)$, an estimate of $F(u)$; (3) inverse Fourier transform $G'_k(u)$; and (4) make the minimum changes in $g'_k(x)$, the resulting computed image, which allow it to satisfy the object-domain constraints to form $g_{k+1}(x)$, a new estimate of the object. In particular, for the problem of a single intensity measurement (as in astronomy) the first three steps are identical to the first three steps of the Gerchberg-Saxton algorithm, Eqs. (6)–(8), and the fourth step is given by

$$g_{k+1}(x) = \begin{cases} g'_k(x), & x \notin \gamma, \\ 0, & x \in \gamma, \end{cases} \quad (10)$$

where γ is the set of points at which $g'_k(x)$ violates the

object-domain constraints, i.e., wherever $g'_k(x)$ is negative or (optionally) where it exceeds the known diameter of the object. The diameter of the object can be computed since it is just half of the diameter of the autocorrelation function, which is the inverse Fourier transform of the squared Fourier modulus. (However, in two dimensions the exact support of the object cannot in general be determined uniquely from the support of its autocorrelation,¹⁰ and so the diameter constraint cannot be applied very tightly.)

The iterations continue until the computed Fourier transform satisfies the Fourier-domain constraints or the computed image satisfies the object-domain constraints; then one has found a solution, a Fourier transform pair that satisfies all the constraints in both domains. The convergence of the algorithm can be monitored by computing the squared error. In the Fourier domain the squared error is the sum of the squares of the amounts by which $G_k(u)$, the computed Fourier transform, violates the Fourier-domain constraints. Since $G'_k(u)$ was formed from $G_k(u)$ by making the minimum changes to satisfy the Fourier-domain constraints, the squared error can be expressed as

$$B_k = E_{Fk}^2 = N^{-2} \sum_u |G_k(u) - G'_k(u)|^2, \quad (11)$$

which, for both problems being considered, can be expressed as

$$B_k = E_{Fk}^2 = N^{-2} \sum_u [|G_k(u)| - |F(u)|]^2. \quad (12)$$

In this section the symbol E_{Fk}^2 is used to distinguish it from the object-domain error E_{0k}^2 described below. For economy of notation in the section of this paper dealing with the gradient methods, the symbol B_k is used instead of E_{Fk}^2 . The symbol B (with the subscript k deleted) is given by Eq. (11) with G and G' replacing G_k and G'_k , respectively.

Similarly, for the error-reduction algorithm the squared error in the object domain can be expressed as

$$E_{0k}^2 = \sum_x |g_{k+1}(x) - g'_k(x)|^2, \quad (13)$$

which for the problem of two intensity measurements can be expressed as

$$E_{0k}^2 = \sum_x [|f(x)| - |g'_k(x)|]^2 \quad (14)$$

and for the problem of a single intensity measurement can be expressed as

$$E_{0k}^2 = \sum_{x \in \gamma} |g'_k(x)|^2, \quad (15)$$

where γ is defined as in Eq. (10). The asymmetry in the use of the N^{-2} factor above was chosen because of the similar asymmetry in the definition of the discrete Fourier transform in Eqs. (2) and (3). When the squared error is zero, a solution has been found.

In the following the error-reduction algorithm is shown to converge, and this convergence property holds for all the applications of the error-reduction algorithm

(not just the problems of recovering phase from single or two intensity measurements).

For the general problem, at the k th iteration the squared error in the Fourier domain is given by Eq. (11). By Parseval's theorem¹¹

$$\begin{aligned} E_{Fk}^2 &= N^{-2} \sum_u |G_k(u) - G'_k(u)|^2 \\ &= \sum_x |g_k(x) - g'_k(x)|^2. \end{aligned} \quad (16)$$

Now compare this with Eq. (13), the error in the object domain. Both $g_k(x)$ and $g_{k+1}(x)$ by definition satisfy the object-domain constraints. Also at any point x , by definition $g_{k+1}(x)$ is the nearest value to $g'_k(x)$ that satisfies the object-domain constraints. Therefore, at all points x

$$|g_{k+1}(x) - g'_k(x)| \leq |g_k(x) - g'_k(x)|, \quad (17)$$

and, therefore, from Eqs. (13) and (16)

$$E_0^2 \leq E_{Fk}^2. \quad (18)$$

Similarly, by Parseval's theorem

$$\begin{aligned} E_{0k}^2 &= \sum_x |g_{k+1}(x) - g'_k(x)|^2 \\ &= N^{-2} \sum_u |G_{k+1}(u) - G'_k(u)|^2. \end{aligned} \quad (19)$$

Since both $G'_k(u)$ and $G'_{k+1}(u)$ satisfy the Fourier-domain constraints, and at any point u , by definition $G'_{k+1}(u)$ is the nearest value to $G_{k+1}(u)$ that satisfies the Fourier-domain constraints, then

$$|G_{k+1}(u) - G'_{k+1}(u)| \leq |G_{k+1}(u) - G'_k(u)|. \quad (20)$$

Therefore, from Eqs. (11) and (19),

$$E_{F,k+1}^2 \leq E_{0k}^2, \quad (21)$$

and combining this with Eq. (18) gives the desired result

$$E_{F,k+1}^2 \leq E_{0k}^2 \leq E_{Fk}^2. \quad (22)$$

That is, the error can only decrease (or stay the same) at each iteration.

In practice, the error-reduction algorithm usually decreases the error rapidly for the first few iterations but much more slowly for later iterations.^{1,2,9,12} The speed of convergence also depends on the type of constraints imposed. Convergence seems to be reasonably fast for the problem of two intensity measurements but painfully slow for the problem of a single intensity measurement. Figure 2 shows an example of the error as a function of the number of iterations of the error-reduction algorithm for the problem of a single intensity measurement. Shown is the normalized rms error, i.e., the square root of E_{Fk}^2 divided by $\sum_u |F(u)|^2$. Both the error and the iteration number are shown on logarithmic scales. The error decreases rapidly during the first thirty iterations but then reaches a plateau at a level of 0.16, decreasing very slowly. After seventy iterations the error again starts to decrease until it reaches a second plateau at the level of 0.02, at which point the error decreases extremely slowly. After 2000 iterations the

error once again starts to decrease until it reaches a third plateau at the level of 0.003, at which point the decrease in the error is negligible.

The occurrences of plateaus during which convergence is extremely slow seem to occur, more often than not, for both applications,^{1,9,12} and in the past it has been observed that, with persistence, one can go beyond the plateau region and again make rapid progress toward a solution.^{1,12} However, as shown in the example of Fig. 2 for the problem of a single intensity measurement, the number of iterations required for convergence of the error-reduction algorithm can be extremely large, making that algorithm unsatisfactory for that application. Fortunately, as will be shown in later sections, other related algorithms converge much faster, reconstructing a recognizable image in twenty or thirty iterations and completing the reconstruction in under one-hundred iterations, which takes <2 min on a Floating Point System AP-120B array processor for array sizes of 128×128 .

III. Steepest-Descent Method

An alternative approach to solving the phase-retrieval problems is to employ one of the gradient search methods. In this section it is shown that one such method, the steepest-descent method, is closely related to the error-reduction algorithm for the problem of reconstructing phase from a single intensity measurement. The relationship between the steepest-descent method and the Gerchberg-Saxton algorithm is also discussed.

An example of how a gradient search method would be used for this problem follows. One can define $B = E_F^2$, the squared error in the Fourier domain given by Eq. (12), as the error metric which one seeks to minimize

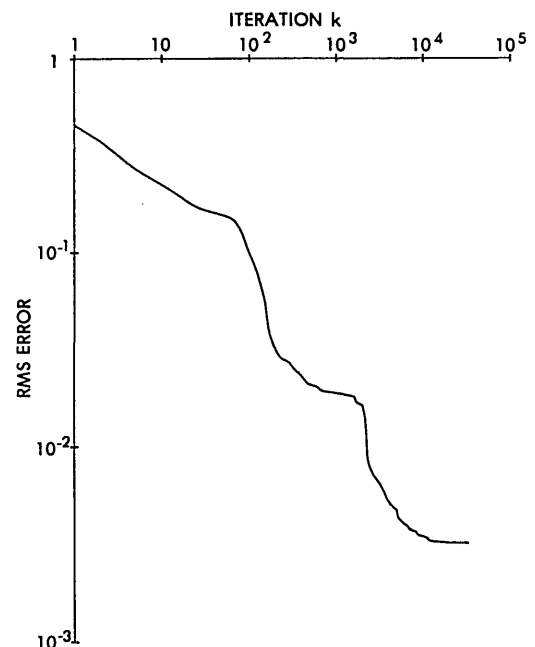


Fig. 2. RMS error vs the number of iterations for the problem of phase retrieval from a single intensity measurement using the error-reduction algorithm.

by varying a set of parameters. Here the N^2 values of $g(x)$, the estimate of $f(x)$, are treated as N^2 independent parameters. Starting at a given point, $g_k(x)$, in the N^2 -dimensional parameter space, one would reduce the error by computing the partial derivatives of B with respect to each of the points $g(x)$ (N^2 partial derivatives) forming the gradient and then move from $g_k(x)$ in a direction opposite that of the gradient to a new point $g'_k(x)$. One would then form a new estimate, $g_{k+1}(x)$, from $g'_k(x)$ by forcing the object-domain constraints to be satisfied. This would be done iteratively until a minimum (it is to be hoped a global minimum) is found. That is, one minimizes the error, B , as a function of the N^2 parameters, $g(x)$, subject to the object-domain constraints.

Ordinarily the computation of the N^2 partial derivatives would be a very lengthy task since each evaluation of B involves an $N \times N$ discrete Fourier transform. However, for the problems considered here the computation can be greatly reduced as described below.

First, consider the problem of a single intensity measurement. The partial derivative of B with respect to a value at a given point, $g(x)$, [if $g(x)$ is assumed to be real since $f(x)$ is real], using Eq. (12), is

$$\partial_g B \equiv \frac{\partial B}{\partial g(x)} = 2N^{-2} \sum_u [|G(u)| - |F(u)|] \frac{\partial |G(u)|}{\partial g(x)}. \quad (23)$$

Later the notation $\partial_g B_k$ will be used to denote $\partial_g B$ evaluated at $g(x) = g_k(x)$. Since

$$\begin{aligned} \frac{\partial |G(u)|}{\partial g(x)} &= \frac{\partial}{\partial g(x)} \sum_y g(y) \exp[-i2\pi u \cdot y/N], \\ &= \exp[-i2\pi u \cdot x/N], \end{aligned} \quad (24)$$

then

$$\begin{aligned} \frac{\partial |G(u)|}{\partial g(x)} &= \frac{\partial [|G(u)|^2]^{1/2}}{\partial g(x)} = \frac{1}{2|G(u)|} \frac{\partial |G(u)|^2}{\partial g(x)}, \\ &= \frac{G(u) \exp[i2\pi u \cdot x/N] + G^*(u) \exp[-i2\pi u \cdot x/N]}{2|G(u)|}. \end{aligned} \quad (25)$$

Therefore, Eq. (23) becomes

$$\begin{aligned} \partial_g B &= N^{-2} \sum_u [G(u) - |F(u)|G(u)/|G(u)|] \exp[i2\pi u \cdot x/N] \\ &\quad + N^{-2} \sum_u [G^*(u) - |F(u)|G^*(u)/|G(u)|] \\ &\quad \times \exp[-i2\pi u \cdot x/N]. \end{aligned} \quad (26)$$

Using Eqs. (6) and (7) to define $G'(u)$ as

$$G'(u) = |F(u)|G(u)/|G(u)|, \quad (27)$$

and noting that Eq. (26) is in the form of a discrete Fourier transform, it can be reduced to

$$\partial_g B = 2[g(x) - g'(x)], \quad (28)$$

where $g'(x)$ is defined by Eq. (8), and the fact that $g(x)$ and $g'(x)$ are real valued has been used. [Note that since $\partial g'(y)/\partial g(x) \neq 0$, it is not true that Eq. (28) follows immediately from Eq. (16).] From Eq. (28) it is evident that the entire gradient, consisting of the N^2 partial derivatives, can be computed very simply by Fourier transforming $g(x)$, applying Eq. (27), inverse Fourier transforming to arrive at $g'(x)$, subtracting $g'(x)$ from

$g(x)$, and multiplying by a constant. In fact, the computation of $g'(x)$ is identical to the first three steps of the error-reduction algorithm.

The optimum step size to take in the direction of the gradient can be determined by forming a first-order Taylor series expansion of B as a function of $g(x)$ about the point $g_k(x)$,

$$B \simeq B_k + \sum_x \partial_g B_k [g(x) - g_k(x)]. \quad (29)$$

This first-order expansion of B is equal to zero at $g(x) = g'_k(x)$ given by

$$g'_k(x) - g_k(x) = - \frac{B_k \partial_g B_k}{\sum_y (\partial_g B_k)^2}, \quad (30)$$

which can be easily verified by inserting Eq. (30) into Eq. (29). Since by Eqs. (28) and (16)

$$\sum_y (\partial_g B_k)^2 = 4 \sum_y [g_k(y) - g'_k(y)]^2 = 4B_k, \quad (31)$$

Eq. (30) becomes

$$g'_k(x) - g_k(x) = -(1/4) \partial_g B_k = (1/2) [g'_k(x) - g_k(x)]. \quad (32)$$

However, since B is quadratic in $g(x)$, the linear approximation above can be expected to predict a step size half as large as the optimum.² Therefore, one should use the double-length step,

$$g'_k(x) - g_k(x) = [g'_k(x) - g_k(x)]$$

or

$$g'_k(x) = g'_k(x). \quad (33)$$

In fact, since $|G'_k(u)| = |F(u)|$, moving to $g'_k(x)$ reduces the error, Eq. (12), to exactly zero. As a final step in one iteration the new estimate should be made to satisfy the object-domain constraints, which is accomplished by using Eq. (10).

Comparing this new estimate with that of the error-reduction algorithm described in Sec. II, it is seen that they are identical. That is, the error-reduction iterative Fourier transform algorithm can be looked on as a rapid method of implementing a double-length step steepest-descent method.

Although the steepest-descent method is identical to the error-reduction algorithm for the problem of a single intensity measurement, the connection is not so close for the problem of two intensity measurements as explored in the Appendix. In that case the error is minimized with respect to the phase estimate $\theta(x)$, and the derivative of the error does move one in a direction approximately toward the Gerchberg-Saxton phase, $\theta'(x)$, of Eq. (9). However, according to Eq. (A4), the direction is somewhat different from that of the Gerchberg-Saxton algorithm; and the step size, according to Eqs. (A12) and (A16), is considerably larger than that of the Gerchberg-Saxton algorithm. Experimental results using the steepest-descent method for the problem of two intensity measurements are shown in Ref. 2.

The relationship between a gradient search method and the error-reduction algorithm for a problem in digital filter design is discussed in Ref. 13.

IV. Other Gradient Search Methods

As shown in Sec. III for the phase problem of a single intensity measurement the steepest-descent method is equivalent to the error-reduction algorithm. And as described in Sec. II, although there is a convergence proof for the error-reduction algorithm, in practice it converges very slowly for the problem of a single intensity measurement. Slow convergence of the steepest-descent (also known as the optimum-gradient) method has also been observed for other applications as well.¹⁴ In this section some other gradient search methods are briefly described, and in Sec. VI it will be shown that in practice they converge much faster than the steepest-descent method for this problem.

Recall from Eq. (30) that the steepest-descent method moves from the point $g_k(x)$ in parameter space to the point

$$g'_k(x) = g_k(x) - h_k \partial_g B_k, \quad (34)$$

where h_k , the step size, is a positive constant, and the gradient is given by Eq. (28). For many applications one would search along the direction of $-\partial_g B_k$, evaluating B repeatedly until the value of h_k that minimizes B is found; then from that point one would recompute the gradient and go off in a new direction. For this application, however, since an $N \times N$ Fourier transform is required to evaluate B and only one more Fourier transform is required to recompute the gradient, one may as well recompute the gradient at each step. After each step of this or the other gradient methods described later one must then satisfy the object-domain constraints to form the new estimate as was done in Eq. (10),

$$g_{k+1}(x) = \begin{cases} g'_k(x), & x \notin \gamma, \\ 0, & x \in \gamma, \end{cases} \quad (35)$$

where set γ is defined as in Eq. (10). In Sec. III the optimum double-length step value of h_k was shown to be unity, for which the steepest-descent method is equivalent to the error-reduction algorithm. In fact, $h_k = 1$ leads to a point $g'_k(x) = g_k(x)$ at which $B = 0$. This is not a solution, however, unless $g'_k(x)$ satisfies the object domain constraints. With this in mind, other values of h_k are better in practice as will be shown in Sec. VI.

A useful block diagram representation of this and other gradient methods is shown in Fig. 3, which emphasizes the similarity of gradient methods to the error-reduction algorithm. The first three steps of the error-reduction algorithm, resulting in the computation of $g'(x)$, do most of the work of computing the gradient. The final step of satisfying the object-domain constraints is common to gradient methods and the error-reduction algorithm. Therefore, one can think of the error-reduction algorithm as a special case of a more general class of gradient methods. For the error-reduction algorithm (or the double-length step steepest-descent method) it just happens that $g'_k(x) = g_k(x)$.

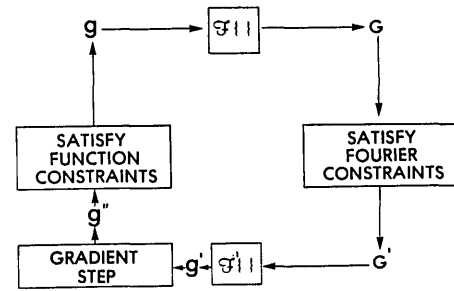


Fig. 3. Block diagram for the gradient-search methods using the method of Fourier transforms to compute the gradient.

A gradient search method superior to the steepest-descent method is the conjugate-gradient method. For that method Eq. (34) is replaced by¹⁴

$$g''_k(x) = g'_k(x) + h_k D_k(x), \quad (36)$$

where the direction $D_k(x)$ is given by

$$D_k(x) = -(\frac{1}{2}) \partial_g B_k + \left[\frac{\sum_y (\partial_g B_k)^2}{\sum_y (\partial_g B_{k-1})^2} \right] D_{k-1}(x), \quad (37)$$

which, using Eqs. (28) and (31), can be written as

$$D_k(x) = g'_k(x) - g_k(x) + (B_k/B_{k-1}) D_{k-1}(x), \quad (38)$$

where one would start the first iteration with $D_1(x) = g'_1(x) - g_1(x)$. After using Eq. (36) one would employ Eq. (35) to form the new estimate as indicated in Fig. 3.

Numerous variations on these gradient methods are possible. For example, one could argue that from one iteration to the next the solution is going in the following direction:

$$D_k(x) = g'_k(x) - g'_{k-1}(x). \quad (39)$$

Since the step in that direction may be too small, a better point to go to would be

$$g''_k(x) = g'_k(x) + h_k [g'_k(x) - g'_{k-1}(x)], \quad (40)$$

where the parameter h_k controls the step size. In Eq. (40) one jumps from the point $g'_k(x)$ rather than from $g_k(x)$ since presumably $g'_k(x)$ is closer to the solution than $g_k(x)$. After using Eq. (40) one would employ Eq. (35) to form the new estimate.

A method that does not seem as practical for this problem is that of (damped) least squares (or Newton-Raphson).¹⁴⁻¹⁶ Since each iteration of a least-squares method involves the inversion of an N^2 by N^2 matrix, a large number of iterations of one of the gradient methods or of one of the iterative Fourier transform methods described in Sec. V could be performed in the same time it takes to do a single iteration of least squares. Furthermore, as has been discussed above, one can readily find a point $g'_k(x) = g_k(x)$ at which the error B is equal to zero, and so a more sophisticated (and more difficult) method, such as least squares, of finding such a point is not warranted.

The problem here is that one is constantly running into the object-domain constraints on $g(x)$. An approach that would be superior to the ones considered

here would be one that minimizes the Fourier-domain error while inherently satisfying the object-domain constraints, or one that minimizes an error metric that combines the Fourier- and object-domain constraints. An example of the former is the use of a gradient search method for the problem of two intensity measurements; by searching over $\theta(x)$ one automatically satisfies the object-domain constraints that they have a given modulus $|f(x)|$. Something along these lines would be very useful for the problem of a single intensity measurement; clearly, more could be done in this area.

V. Input-Output Algorithm

A solution to the problem of the slow convergence of the error-reduction algorithm has been the input-output algorithm, which has proved to converge faster for both the problem of two intensity measurements^{6,17} and the problem of a single intensity measurement.^{4,5} The input-output algorithm differs from the error-reduction algorithm only in the object-domain operation. The first three operations—Fourier transforming $g(x)$, satisfying the Fourier-domain constraints, and inverse Fourier transforming the result—are the same for both algorithms. If grouped together as indicated in Fig. 4, those three operations can be thought of as a nonlinear system having an input $g(x)$ and an output $g'(x)$. The useful property of this system is that its output is always an image having a Fourier transform that satisfies the Fourier-domain constraints. Therefore, if the output also satisfies the object-domain constraints, it is a solution to the problem. Unlike the error-reduction algorithm and the gradient methods, the input $g(x)$ no longer must be thought of as the current best estimate of the object; instead, it can be thought of as the driving function for the next output, $g'(x)$. The input $g(x)$ does not necessarily satisfy the object-domain constraints. This viewpoint allows one a great deal of flexibility and inventiveness in selecting the next input, and allows for the invention of algorithms that converge more rapidly to a solution. The input-output algorithm, then, is actually a class of algorithms as will be described below.

As described elsewhere,^{6,7,17} it has been found that a small change of the input results in a change of the

output in the same general direction as the change of the input. More precisely, for a small change of the input the expected value of the corresponding change of the output is a constant α times the change of the input. Since additional nonlinear terms also appear in the output, the change of the output due to a particular change of the input cannot be predicted exactly. Nevertheless, by appropriate changes of the input, the output can be pushed in the general direction desired. If a change $\Delta g(x)$ is desired in the output, a logical choice of the change of the input to achieve that change of the output would be $\beta \Delta g(x)$, where β is a constant ideally equal to α^{-1} .

For the problem of phase retrieval from a single intensity measurement the desired change of the output is

$$\Delta g_k(x) = \begin{cases} 0, & x \notin \gamma, \\ -g'_k(x), & x \in \gamma, \end{cases} \quad (41)$$

where γ is the set of points at which $g'_k(x)$ violates the object-domain constraints. That is, where the constraints are satisfied, one does not require a change of the output; but where the constraints are violated, the desired change of the output, in order to have it satisfy the object-domain constraints, is one that drives it to a value of zero (and, therefore, the desired change is the negative of the output at those points). Therefore, a logical choice for the next input is

$$\begin{aligned} g_{k+1}(x) &= g_k(x) + \beta \Delta g_k(x) \\ &= \begin{cases} g_k(x), & x \notin \gamma, \\ g_k(x) - \beta g'_k(x), & x \in \gamma. \end{cases} \end{aligned} \quad (42)$$

We will refer to the use of Eq. (42) as the basic input-output algorithm.

An interesting property of the nonlinear system (consisting of the set of three steps mentioned above) is that if an output g' is used as an input, its output will be itself. Since the Fourier transform of g' already satisfies the Fourier-domain constraints, g' is unaffected as it goes through the system. Therefore, irrespective of what input actually resulted in the output g' , the output g' can be considered to have resulted from itself as an input. From this point of view another logical choice for the next input is

$$\begin{aligned} g_{k+1}(x) &= g'_k(x) + \beta \Delta g_k(x) \\ &= \begin{cases} g'_k(x), & x \notin \gamma, \\ g'_k(x) - \beta g'_k(x), & x \in \gamma. \end{cases} \end{aligned} \quad (43)$$

We will refer to the use of Eq. (43) as the output-output algorithm.

Note that if $\beta = 1$ in Eq. (43), the output-output algorithm reduces to the error-reduction algorithm of Eq. (10). Since the optimum value of β is usually not unity, the error-reduction algorithm can be looked on as a suboptimal version of a more general approach.

Still another method of choosing the next input which was investigated is a combination of the upper line of Eq. (43) with the lower line of Eq. (42):

$$g_{k+1}(x) = \begin{cases} g'_k(x), & x \notin \gamma, \\ g_k(x) - \beta g'_k(x), & x \in \gamma. \end{cases} \quad (44)$$

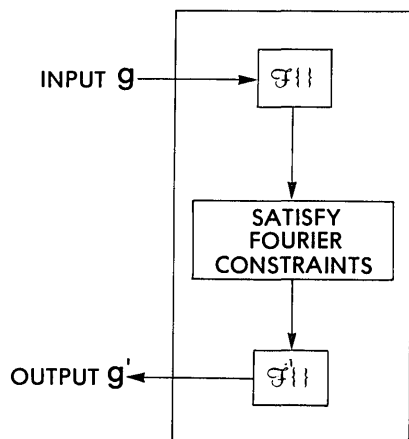


Fig. 4. Block diagram of the system for the input-output concept.

We will refer to the use of Eq. (44) as the hybrid input-output algorithm. The hybrid input-output algorithm is an attempt to avoid a stagnation problem that tends to occur with the output-output algorithm. The output-output algorithm often works itself into a situation in which the output on successive iterations does not change despite being far from a solution. For the hybrid input-output algorithm, on the other hand, if at a given value of x the output remains negative for more than one iteration, the corresponding point in the input continues to grow larger and larger until eventually that output value must go non-negative.

For the input-output algorithms the error E_F is usually meaningless since the input $g_k(x)$ is no longer an estimate of the object. Then the meaningful error is the object-domain error E_0 given by Eq. (15).

For the problem of phase retrieval from two intensity measurements the desired change of the output takes a different form, and it is described elsewhere^{6,7,17} in the context of a computer holography synthesis problem involving intensity constraints in each of the two domains.

VI. Experimental Comparison of Phase-Retrieval Algorithms

In this section the gradient search and input-output algorithms are compared for the problem of phase retrieval from a single intensity measurement by using them all on the same Fourier modulus data and with the same starting input. For each approach several different values of the algorithm parameter (h or β) were tried. The principal problem with the error-reduction algorithm is that it tends to stagnate after a few iterations. For this reason the starting point for the iterations was chosen to be a partially reconstructed image on which the error-reduction algorithm was making slow progress. Figure 5 shows a plot of E_0 , the rms error, vs the number of iterations beyond this starting point using the error-reduction algorithm. Starting at 0.071275, E_0 decreased slowly but steadily to 0.067470 after ten iterations and to 0.063373 after nineteen iterations. In this paper, all values of E_0 are normalized by dividing by the square root of $\sum [g'(x)]^2$, the total image energy. The object for the experiment described in this section is a digitized photograph of a satellite in a field of view of 128×128 pixels, and its Fourier modulus is noise-free.

The algorithms were compared by performing ten iterations of each algorithm, followed by nine iterations of the error-reduction algorithm (a total of nineteen iterations) using the same starting input for each. During the first ten iterations the value of the algorithm parameter β or h was held constant. The reason that each algorithm was followed by nine iterations of the error-reduction algorithm is as follows. In many cases it has been observed that definite progress (i.e., improved visual quality of the output image) is being made with an input-output algorithm even though E_0 decreases very little or even increases with each iteration. The relationship between E_0 and the visual image quality is not fully understood, although, of course, one would expect a high degree of correlation between the

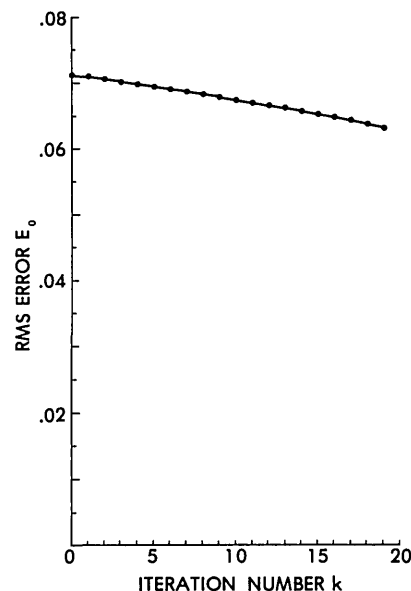


Fig. 5. RMS error vs the number of iterations using the error-reduction algorithm.

two. For those cases for which the visual quality improves while E_0 does not it was found that, if one then performs a few (say five or ten) iterations of the error-reduction algorithm, the visual quality of the output image changes very little, but E_0 decreases rapidly until it becomes more consistent with the visual image quality. Therefore, to gauge the true progress of an input-output algorithm using the value of E_0 , a few iterations of the error-reduction algorithm are performed after the iterations of the input-output algorithm. For this reason, for all cases the ten iterations of the algorithm being tested were followed by nine iterations of the error-reduction algorithm in order to make a fair comparison.

Plots of E_0 after the set of nineteen iterations described above (ten iterations followed by nine iterations of the error-reduction algorithm) are shown in Fig. 6 as a function of the algorithm parameter for the various algorithms. Note that both the steepest-descent method with $h = 1.0$ and the output-output algorithm with $\beta = 1.0$ are equivalent to the error-reduction algorithm at $E_0 = 0.063373$, both circled in Fig. 6. Comparing these plots it is seen that the algorithm which most reduced the error after the set of nineteen iterations is the hybrid input-output algorithm with a value of β equal to about unity.

For each algorithm the use of a small algorithm parameter (β or h) leads to a steady but slow decline of E_0 . Increasing the value of the parameter increases the speed at which E_0 decreases until one reaches a point where the parameter is too large and the algorithm becomes unstable. The instability of the algorithm for larger values of the algorithm parameter makes possible more than one local minimum in the plots of E_0 vs the algorithm parameter.

For all the algorithms, keeping h or β fixed for all iterations is not the best possible strategy, particularly for the gradient methods. At the point at which the

error-reduction algorithm is converging slowly the gradient is small, and one must then use a large value of h to make rapid progress. However, after using a large value of h for a few iterations, one moves to a point where the gradient is much larger. Then one should use a smaller value of h to avoid algorithm instability. If a method for adaptively choosing h at each iteration was devised, one would expect the gradient methods to perform considerably better than the results shown here using a fixed value of h .

Examples of an alternative to using a fixed value of h and β are shown in Fig. 7. For the first ten iterations of each algorithm the indicated value of h or β was used for iterations $k = 1, 3, 5, 7$, and 9 , and the error-reduction algorithm ($h = 1$ for the steepest-descent and conjugate-gradient) was used for iterations $k = 2, 4, 6, 8$, and 10 . The iterations using the error-reduction algorithm help to stabilize the algorithm by moving toward a point where the gradient is smaller. Comparing Figs. 6 and 7, it is seen that with this alternative strategy the optimum value of each algorithm parameter is considerably larger than the optimum value when the parameter is kept constant. At the optimum value of the algorithm parameter the alternative strategy gave better results (a lower value of E_0 at the end of the sequence of nineteen iterations) than those shown in Fig. 6 for the basic input-output and for the output-output algorithms; the two strategies were comparable for the steepest-descent method; and for the hybrid input-output algorithm the alternative strategy gave poorer results than those shown in Fig. 6.

Curve E in Fig. 7 shows the result with the algorithm of Eq. (40) using the alternative strategy described

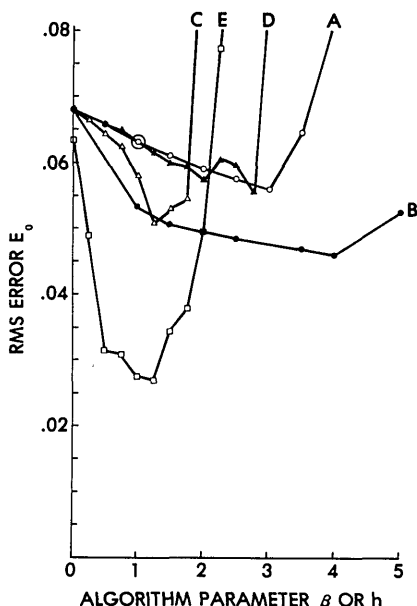


Fig. 6. RMS error after a fixed number of iterations vs the algorithm parameter. Curve A: steepest-descent method (O); B: conjugate-gradient method (●); C: basic input-output algorithm (Δ); D: output-output algorithm (\blacktriangle); E: hybrid input-output algorithm (\square). The result using the error-reduction algorithm is indicated by a large circle.

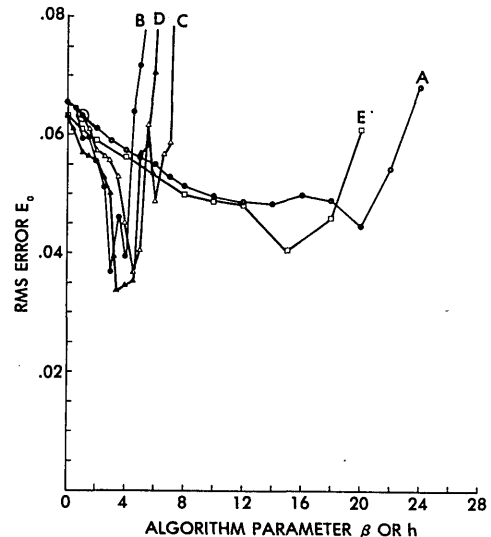


Fig. 7. RMS error after a fixed number of iterations (using the alternative strategy) vs the algorithm parameter. Curve A: steepest-descent method (O); B: basic input-output algorithm (●); C: output-output algorithm (Δ); D: hybrid input-output algorithm (\blacktriangle); E: the algorithm of Eq. (40) (\square).

above. This and numerous other variations on these algorithms can be used with varying degrees of success.

The results shown in Figs. 6 and 7 were for a particular set of Fourier modulus data for a particular stage of reconstruction and for a particular number of iterations, and the results in other circumstances could be significantly different. At the optimum values of the algorithm parameter in each case the algorithm parameter was large enough to make the algorithm somewhat unstable, and so substantially different results could be obtained if relatively small changes in starting point, algorithm parameter, or number of iterations were made. In general, slower but steadier progress is made if an algorithm parameter is used that is somewhat smaller than the optimum according to Figs. 6 and 7. These results do serve to show trends that can be expected to apply in a wider range of circumstances. Further development is needed to determine the best approach for the general case. As of this writing the most successful strategy has been to alternate between several (10–30) iterations of the hybrid input-output algorithm and a few (5–10) iterations of the error-reduction algorithm.

Figure 8 shows E_0 vs the number of iterations past the starting point for the hybrid input-output algorithm with $\beta = 1$ (curve B) and for the error-reduction algorithm (curve A, repeated from Fig. 5). Curve B1 shows the results for the set of nineteen iterations described above (ten iterations of the hybrid input-output algorithm followed by nine iterations of the error-reduction algorithm). Curve B2 shows the results of twenty iterations of the hybrid input-output algorithm followed by a few iterations of the error-reduction algorithm. The instability of the hybrid input-output algorithm is seen in curve B, in which E_0 increases from 0.071275 to 0.137707 during the first four iterations. By the end

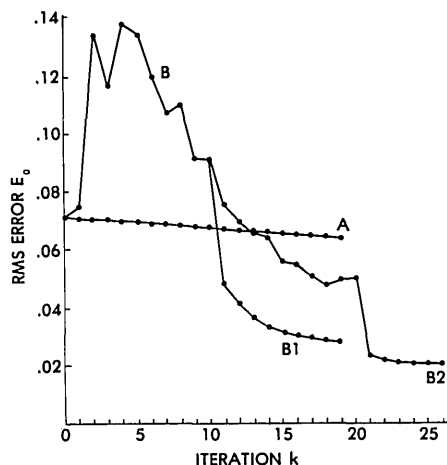


Fig. 8. RMS error vs the number of iterations for the error-reduction algorithm (curve A), and for the hybrid input-output algorithm (curve B—see text).

of ten iterations E_0 decreases to 0.091176, still worse than the starting point, although the appearance of the image is improved from the starting point. At the eleventh iteration of curve B1, the first iteration of the error-reduction algorithm, E_0 drops sharply to a value of 0.047244, although the appearance of the output image changes little from that of the tenth iteration. If after the tenth iteration the hybrid input-output iterations are continued for ten more iterations (curve B2), E_0 continues to decrease to well below the level of E_0 for the error-reduction algorithm alone (curve A). Similar to the case of B1, after the twentieth iteration of the hybrid input-output algorithm (curve B2), when a few iterations of the error-reduction algorithm were performed, again E_0 fell rapidly to a level consistent with the improved image quality that was present. This is the characteristic of the input-output algorithms that mandated the use of a few iterations of the error-reduction algorithm to make a fair comparison. For the input-output algorithms the image quality is often better than what one would infer from the value of E_0 .

VII. Diameter Constraint, Starting Input, and Stripes

A side issue pertinent to all algorithms is how one defines the diameter of the object for the purpose of applying the diameter constraint in the object domain. For the problem of phase retrieval from a single intensity measurement the object is usually of finite extent and on a dark (zero value) background. Bounds on the support of the object (the set of points over which the object is nonzero) can then be determined from the support of the autocorrelation, which, being the inverse Fourier transform of the square of the Fourier modulus, can be computed from the given data. As shown in Ref. 10, for extended objects the support of the autocorrelation usually does not uniquely define the support of the object. Nevertheless, reasonably tight bounds can be made on the support of the object. Locator sets¹⁰ can be defined that contain all possible solutions. One can, therefore, define a region (a mask) outside of which

the output image is constrained to be zero. That is, the set γ defined in Eq. (10) includes all points outside the mask for which $g'_k(x) \neq 0$. One need not use this information for the algorithm to converge, but it is desirable to do so since using this additional information speeds up the convergence of the algorithm.

A problem often occurs with the diameter constraint even though the mask region is correctly defined. If the partially reconstructed image $g'_k(x)$ is not centered in the mask region, in applying the diameter constraint one might inadvertently be trying to chop off (truncate) one part of the object, which usually results in stagnation of the algorithm. For this reason it is usually advantageous to define the mask as being somewhat larger than the extent of the object.

We have found a good strategy for choosing the mask: For the first several iterations, define a smaller mask which very tightly constrains the object. This helps to speed up the convergence of the algorithm initially, but slows it down for later iterations when the problem mentioned above becomes more significant. Then, for later iterations use a larger mask which ensures that none of the solution is being truncated by the mask. Logical choices for masks are any of the locator sets.¹⁰

Faster convergence can be expected if the starting input $g_1(x)$ is closer to the solution. A good starting input is formed as follows. Compute the autocorrelation function and then demagnify it by a factor of 2 (save only every other pixel in both dimensions). Then threshold the demagnified autocorrelation at a value which is a small fraction of its peak, setting it equal to zero wherever it is below the threshold value. Finally, replace each value above the threshold with a sample of a random variable uniformly distributed between zero and unity. The result is a random (unbiased) starting input having approximately the same size and shape as the original object.

A curious phenomenon often occurs for the problem of phase retrieval from a single-intensity measurement. The phase-retrieval algorithm often stagnates at a local minimum characterized by a pattern of stripes across the image.^{18,19} In most cases the stripes are barely noticeable and are of low contrast, superimposed on an otherwise excellent reconstructed image. In some cases the stripes are of high enough contrast to be objectionable, although they still permit the object to be recognized. The cause of this phenomenon is not well understood, but it is thought that it is an algorithm convergence problem rather than a uniqueness problem¹⁹ (it is at a local, not a global, minimum of E_0). A method of avoiding this phenomenon is presently being sought, although it fortunately is not much of a problem in most cases.

VIII. Image Reconstruction Example

An example of a computer experiment using the iterative reconstruction algorithm for the problem of phase retrieval from a single intensity measurement is shown in Fig. 9. In this example a realistic stimulation was performed to arrive at the kind of noisy Fourier

modulus data that would be provided by stellar speckle interferometry,²⁰ including the effects of atmospheric turbulence and photon noise.¹⁸

An undegraded object, a digitized photograph of a satellite shown in Fig. 9(a), was convolved with 156 different point spread functions to produce 156 different blurred images. Each point spread function represented a different realization of the effects of the turbulent atmosphere. The blurred images were then subjected to a Poisson noise process to simulate the effects of photon noise. For this example there were ~300,000 photons/degraded image (or of the order of 100 photons/pixel over the extent of the object), which is realistic for objects of this type when imaged through a telescope of 1.2-m diam. Two of the resulting 156 degraded images are shown in Figs. 9(b) and 9(c). The degraded images were then processed by Labeyrie's²⁰ method as modified by Goodman and Belsher.²¹ The estimate of the modulus of the Fourier transform of the object is given by¹⁸

$$|\hat{F}(u)| = W(u) \left[\frac{\sum_{m=1}^M |I_m(u)|^2 - N_p}{\sum_{m=M+1}^{2M} |S_m(u)|^2} \right]^{1/2}, \quad (45)$$

where $I_m(u)$ is the Fourier transform of the m th degraded image, N_p is the total number of photons detected (it is subtracted to compensate for a certain noise bias term that arises in the power spectrum due to photon noise²¹), $S_m(u)$ is the Fourier transform of the m th point spread function (to provide compensation for the MTF of the speckle interferometry process), and the weighting factor $W(u)$ is the MTF due to the telescope

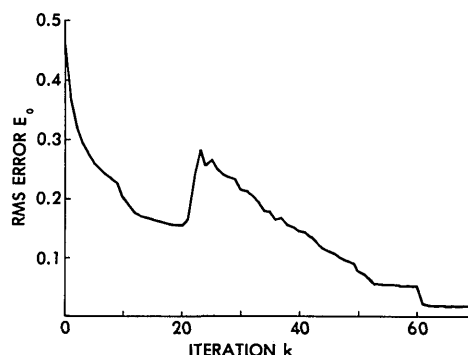


Fig. 10. RMS error vs the number of iterations.

aperture. In practice the denominator of this expression would be obtained by making measurements on a reference star through an atmosphere having the same statistics as that which blurred the images or by using a model of the effects of atmospheric turbulence. $W(u)$ was included to restore the natural MTF due to the telescope aperture which was removed by the denominator of the equation above. Figure 9(d) shows the resulting Fourier modulus estimate.

The object was reconstructed (or equivalently, the Fourier phase was retrieved) using the hybrid input-output algorithm alternately with the error-reduction algorithm. The result, shown in Fig. 9(e), agrees very well with the original object shown in Fig. 9(a), despite the noise present in the Fourier modulus data. Good reconstructed images were also obtained when only one-tenth as many photons were assumed to be available.¹⁸

Figure 10 shows E_0 vs the number of iterations for this reconstruction. The starting input used was the randomized demagnified autocorrelation described in Sec. VII, using a threshold value of 0.004 times the peak of the autocorrelation. For the first ten iterations the error-reduction algorithm was used, and the mask defining the diameter constraint was chosen to be the region over which the autocorrelation function, spatially demagnified by a factor of 2, exceeded 0.004 of its maximum value (providing a fairly tight diameter constraint). For iterations 11–20 the error-reduction algorithm was used, and the mask for these and the remaining iterations was chosen to be a square of length 64 pixels, which is larger than the actual object extent of ~60 × 40 pixels (imbedded in an array of 128 × 128 pixels). The error decreased suddenly at the tenth iteration since some positive-valued points that were inside the second mask but outside the first mask were no longer counted as contributing to E_0 . By the twentieth iteration the error-reduction algorithm was converging very slowly. For iterations 21–60 the hybrid input-output algorithm, with β equal to one, was used. At first E_0 increased sharply (although the output image appeared no worse than at iteration 20) but then decreased fairly rapidly until stagnating at $E_0 \approx 0.05$ at about iteration 55. For iterations 61–70 the error-reduction algorithm was used, for which E_0 dropped suddenly from 0.05 to 0.02, although the visual ap-

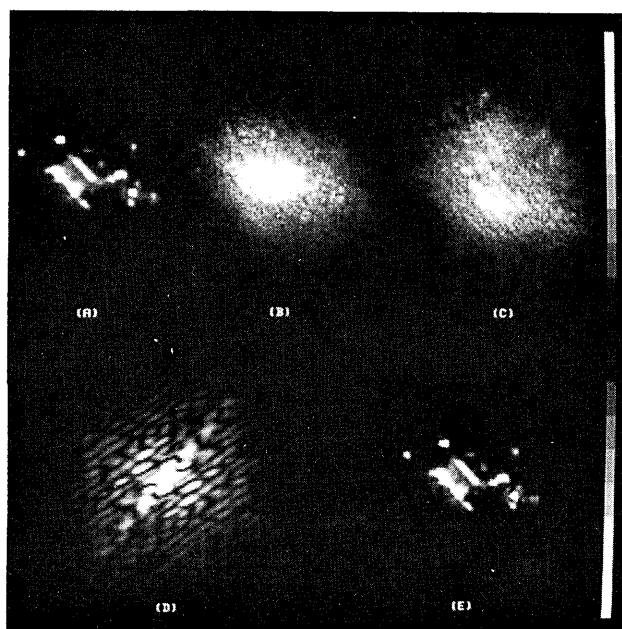


Fig. 9. Image reconstruction experiments. (a) Undegraded object; (b), (c) examples of degraded images simulated to include the effects of atmospheric turbulence and photon noise; (d) Fourier modulus estimate computed from the degraded images; (e) image reconstructed using the iterative algorithm.

pearance of the reconstructed image remained the same as for iteration number 60.

This final value of E_0 is comparable to 0.03, the normalized rms error of the Fourier modulus estimate itself.¹⁸ It is impossible to reduce E_0 to zero since the noise in the Fourier modulus estimate results in an inconsistency of the non-negativity constraint and the Fourier modulus estimate. This inconsistency can be seen from the fact that the autocorrelation estimate computed from the Fourier modulus estimate has areas of negative value.

Reconstruction experiments do not always proceed as smoothly as the one described above. When stagnation occurs before E_0 is reduced to a level consistent with the error in the Fourier modulus data, often the best strategy is to restart the algorithm using a different set of random numbers as the initial input. One must also be careful in the way one chooses the mask as discussed in Sec. VII. Finally, it is always advisable to reconstruct the image two or three times from the same Fourier modulus data using different starting inputs each time. If the same image is reconstructed each time, one would have confidence that the solution is unique.

IX. Comments and Conclusions

For the problem of phase retrieval from a single-intensity measurement it has been shown that the error-reduction algorithm (the Gerchberg-Saxton algorithm applied to this problem) is equivalent to the steepest-descent method with a double-length step. Furthermore, it was shown that the error-reduction algorithm converges in the sense that the error monotonically decreases. However, in practice the error-reduction algorithm converges very slowly; several other algorithms converge much faster, including other gradient search algorithms which utilize the method of Fourier transforms for rapidly computing the gradient of the error and the input-output family of iterative Fourier transform algorithms. Of the algorithms investigated so far, the hybrid input-output algorithm has converged the fastest. Nevertheless, the gradient-search algorithms also converge reasonably fast, and they deserve further development. The performance of any given algorithm can vary depending on the stage of reconstruction and on what is being reconstructed. Although a practical phase-retrieval algorithm is in hand, there is still an element of trial and error involved.

For the problem of phase retrieval from two intensity measurements, the steepest-descent method and the Gerchberg-Saxton algorithm were compared and were found to share some similarities. When manipulating just the object's phase, the object-domain constraint, that the object's modulus equals the measured modulus, is automatically satisfied. Therefore, the gradient-search methods seem inherently better suited to the problem of two intensity measurements than to the problem of a single intensity measurement.

Not considered here has been the question of the uniqueness of solutions to either of the two phase-retrieval problems. There has been considerable con-

troversy surrounding this question, both for the problem of two intensity measurements^{2,22,23} and for the problem of a single intensity measurement.²⁴⁻²⁸ For the latter problem, the experimental reconstruction results shown here and elsewhere^{4,5,10,19} suggest that for complicated 2-D objects the solution is usually unique in practice, even when the Fourier modulus data are corrupted by a considerable amount of noise.¹⁹

This research was supported by the Air Force Office of Scientific Research under contract F49620-80-C-0006.

Appendix

In this Appendix the relationship between the steepest-descent method for the problem of two intensity measurements and the Gerchberg-Saxton algorithm is explored.

For the problem of two intensity measurements, the error B is minimized with respect to the phase estimates $\theta(x)$. The partial derivative of B with respect to the value of a given phase $\theta(x)$ is

$$\partial_{\theta} B = \frac{\partial B}{\partial \theta(x)} = 2N^{-2} \sum_u [|G(u)| - |F(u)|] \frac{\partial |G(u)|}{\partial \theta(x)}. \quad (\text{A1})$$

Since

$$\begin{aligned} \frac{\partial |G(u)|}{\partial \theta(x)} &= \frac{\partial}{\partial \theta(x)} \sum_y |f(y)| \exp[i\theta(y)] \exp[-i2\pi u \cdot y/N] \\ &= i|f(x)| \exp[i\theta(x)] \exp[-i2\pi u \cdot x/N] \\ &= ig(x) \exp[-i2\pi u \cdot x/N], \end{aligned} \quad (\text{A2})$$

then

$$\frac{\partial |G(u)|}{\partial \theta(x)} = \frac{G(u)(-i)g^*(x) \exp[i2\pi u \cdot x/N] + \text{c.c.}}{2|G(u)|}, \quad (\text{A3})$$

where c.c. indicates the complex conjugate of what precedes it. Using Eq. (27), Eq. (A1) becomes²

$$\begin{aligned} \partial_{\theta} B &= ig^*(x)[g'(x) - g(x)] + \text{c.c.} \\ &= ig^*(x)g'(x) + \text{c.c.} = -2 \text{Im}[g^*(x)g'(x)] \\ &= -2|f(x)||g'(x)|\sin[\theta'(x) - \theta(x)], \end{aligned} \quad (\text{A4})$$

where $\theta'(x)$ is the phase of $g'(x)$. Therefore, the gradient can be easily computed using two Fourier transforms (the first three steps of the Gerchberg-Saxton algorithm) plus evaluating Eq. (A4). Analogous to the linear approximation of Eq. (29) is

$$B \simeq B_k + \sum_x \partial_{\theta} B_k [\theta(x) - \theta_k(x)], \quad (\text{A5})$$

where $\partial_{\theta} B_k$ is $\partial_{\theta} B$ evaluated at $\theta(x) = \theta_k(x)$, the phase of $g_k(x)$. This linear expansion of B is equal to zero for

$$\theta(x) - \theta_k(x) = - \frac{B_k \partial_{\theta} B_k}{\sum_y (\partial_{\theta} B_k)^2}, \quad (\text{A6})$$

which can be easily verified by inserting Eq. (A6) into Eq. (A5). By Eq. (A4),

$$\begin{aligned} \sum_y (\partial_{\theta} B_k)^2 &= 4 \sum_y |f(y)|^2 |g'_k(y)|^2 \\ &\quad \times \sin^2[\theta'_k(y) - \theta_k(y)], \end{aligned} \quad (\text{A7})$$

and so Eq. (A6) becomes

$$\theta(x) - \theta_k(x) = \frac{B_k |f(x)g'_k(x)| \sin[\theta'_k(x) - \theta_k(x)]}{2 \sum_y |f(y)|^2 |g'_k(y)|^2 \sin^2[\theta'_k(y) - \theta_k(y)]}. \quad (\text{A8})$$

Now consider the situation after several iterations have been performed and the error is decreasing slowly. Then for most values of x ,

$$\sin[\theta'_k(x) - \theta_k(x)] \simeq \theta'_k(x) - \theta_k(x). \quad (\text{A9})$$

Then from Eq. (A8) it is seen that $\theta(x) - \theta_k(x)$, which is the change of $\theta_k(x)$ indicated by the steepest-descent method, is approximately proportional to $\theta'_k(x) - \theta_k(x)$. Recall from Eq. (9) that $\theta'_k(x)$ is the new phase that would be given by the Gerchberg-Saxton algorithm. Thus, for the problem of two intensity measurements the steepest-descent method gives a new phase that is going in the same general direction as the Gerchberg-Saxton algorithm but uses a different step size. For the steepest-descent method the direction differs somewhat from $\theta'_k(x) - \theta_k(x)$ due to the weighting factor $|f(x)g'_k(x)|$, which varies with x .

To obtain some notion of the step size for the steepest-descent method, continue to consider the situation where the error is decreasing slowly and Eq. (A9) applies. Then, using Eq. (16) one can approximate

$$\begin{aligned} B_k &= \sum_x |f(x)| \exp[i\theta_k(x)] - |g'_k(x)| \exp[i\theta'_k(x)]|^2 \\ &= \sum_x [|f(x)|^2 + |g'_k(x)|^2 - 2|f(x)g'_k(x)| \cos[\theta'_k(x) - \theta_k(x)]] \\ &\simeq \sum_x [|f(x)| - |g'_k(x)|]^2 + \sum_x |f(x)g'_k(x)| \\ &\quad \times \sin^2[\theta'_k(x) - \theta_k(x)]. \quad (\text{A10}) \end{aligned}$$

To further simplify things, consider the special case for which $f(x) = f_a$, a constant independent of x . Furthermore, if B is small, for most values of x ,

$$||f(x)| - |g'_k(x)|| \ll |f(x)|. \quad (\text{A11})$$

Under these circumstances, inserting Eqs. (A9) and (A10) into (A8) yields

$$\theta(x) - \theta_k(x) \simeq \frac{1}{2} \left(1 + \frac{B_k^r}{B_k^t} \right) [\theta'_k(x) - \theta_k(x)], \quad (\text{A12})$$

where

$$B_k^r = \sum_x [|f(x)| - |g'_k(x)|]^2, \quad (\text{A13})$$

$$B_k^t = \sum_x |g'_k(x)|^2 [\theta'_k(x) - \theta_k(x)]^2 \quad (\text{A14})$$

are the respective radial and tangential components of the error $B_k \simeq B_k^r + B_k^t$. The radial component B^r is the object-domain squared error of the modulus of $|g'_k(x)|$ and is equal to E_{0k}^2 according to Eq. (14). When the error is decreasing slowly,

$$B_k^t \simeq B_k - B_k^r = E_{pk}^2 - E_{0k}^2 \ll E_{0k}^2 = B_k^r. \quad (\text{A15})$$

Therefore, under these circumstances the coefficient of Eq. (A12),

$$\frac{1}{2} \left[1 + \frac{B_k^r}{B_k^t} \right] \gg 1. \quad (\text{A16})$$

That is, when the Gerchberg-Saxton algorithm is converging slowly, the steepest-descent method indicates that a much larger step size be used. Note that if one uses the double-length step steepest-descent method, the factor of $1/2$ in Eqs. (A8), (A12), and (A16) is replaced by unity.

References

1. R. W. Gerchberg and W. O. Saxton, *Optik* **35**, 237 (1972).
2. W. O. Saxton, *Computer Techniques for Image Processing in Electron Microscopy* (Academic, New York, 1978).
3. R. A. Gonsalves, *J. Opt. Soc. Am.* **66**, 961 (1976).
4. J. R. Fienup, *Opt. Lett.* **3**, 27 (1978).
5. J. R. Fienup, *Opt. Eng.* **18**, 529 (1979).
6. J. R. Fienup, *Opt. Eng.* **19**, 297 (1980).
7. J. R. Fienup, "Reconstruction and Synthesis Applications of an Iterative Algorithm," in *Transformations in Optical Signal Processing*, W. T. Rhodes, J. R. Fienup, and B. E. A. Saleh, Eds. (Society of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1982), to be published.
8. P. M. Hirsch, J. A. Jordan, Jr., and L. B. Lesem, "Method of Making an Object-Dependent Diffuser," U.S. Patent 3,619,022 (9 Nov., 1971).
9. N. C. Gallagher and B. Liu, *Appl. Opt.* **12**, 2328 (1973).
10. J. R. Fienup, T. R. Crimmins, and W. Holsztynski, *J. Opt. Soc. Am.* **72**, 610 (1982).
11. R. N. Bracewell, *The Fourier Transform and Its Applications* (McGraw-Hill, New York, 1965).
12. R. H. Boucher, *Proc. Soc. Photo-Opt. Instrum. Eng.* **231**, 130 (1980).
13. M. T. Manry and J. K. Aggarwal, *IEEE Trans. Circuits Syst. CAS-23*, 185 (1976).
14. D. P. Feder, *Appl. Opt.* **2**, 1209 (1963).
15. D. R. Buchele, *Appl. Opt.* **7**, 2433 (1968).
16. B. R. Frieden and D. G. Currie, *J. Opt. Soc. Am.* **66**, 1111 (1976) (Abstract).
17. J. R. Fienup, "Improved Synthesis and Computational Methods for Computer-Generated Holograms," Ph.D. Thesis, Stanford University, May, 1975 (University Microfilms No. 75-25523), Chap. 5.
18. G. B. Feldkamp and J. R. Fienup, *Proc. Soc. Photo-Opt. Instrum. Eng.* **231**, 84 (1980).
19. J. R. Fienup, "Fourier Modulus Image Construction," Report RADCR-TR-81-63 (1981).
20. A. Labeyrie, *Astron. Astrophys.* **6**, 85 (1970); D. Y. Gezari, A. Labeyrie, and R. V. Stachnik, *Astrophys. J. Lett.* **173**, L1 (1972).
21. J. W. Goodman and J. F. Belsher, *Proc. Soc. Photo-Opt. Instrum. Eng.* **75**, 141 (1976).
22. A. M. J. Huizer, A. J. J. Drenth, and H. A. Ferwerda, *Optik* **45**, 303 (1976); A. M. J. Huizer and H. A. Ferwerda, *Optik* **46**, 407 (1976).
23. A. J. Devaney and R. Chidlaw, *J. Opt. Soc. Am.* **68**, 1352 (1978).
24. Yu. M. Bruck and L. G. Sodin, *Opt. Commun.* **30**, 304 (1979).
25. W. Lawton, *Proc. Soc. Photo-Opt. Instrum. Eng.* **231**, 94 (1980).
26. A. M. J. Huizer and P. Van Toorn, *Opt. Lett.* **5**, 499 (1980).
27. T. R. Crimmins and J. R. Fienup, *J. Opt. Soc. Am.* **71**, 1026 (1981).
28. J. R. Fienup, "Image Reconstruction for Stellar Interferometry," in *Current Trends in Optics*, F. T. Arecchi and F. R. Aussenegg, Eds. (Taylor & Francis, London, 1981), pp. 95-102.