OPTIK
Vol. 35 (No. 2)
237-246 (1972)

A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures

By *R. W. Gerchberg* and *W. O. Saxton*

Cavendish Laboratory, Cambridge, England

Received 29 November 1971

Abstract

An algorithm is presented for the rapid solution of the phase of the complete wave function whose intensity in the diffraction and imaging planes of an imaging system are known. A proof is given showing that a defined error between the Thimated function and the correct function must decrease as the algorithm iterates. este problem of uniqueness is discussed and results are presented demonstrating the power of the method.

Inhalt

Ein praktischer Algorithmus zur Berechnung der Phase aus den Intensitäten in Beugungs- und Bildebene. Ein Algorithmus zur schnellen Lösung der Aufgabe wird mitgeteilt. Wie eine Probe zeigt, nimmt eine definierte Abweichung zwischen der errechneten und der wirklichen Funktion ab, wenn der Algorithmus iterativ angewendet wird. Das Problem der Eindeutigkeit wird diskutiert. Ergebnisse demonstrieren die Leistungsfähigkeit der Methode.

In 1948, *Gabor* (1948) proposed an experimental method for determining the phase function across a wave front. Basically the method involved the addition of a reference wave to the wave of interest in the recording plane. The resulting hologram recorded was a series of intensity fringes which contained enough information to reconstruct the complete wave function of interest. However, in many real situations the method has been cumbersome and impractical to employ. It is interesting that *Gabor* originally proposed the method in connection with electron microscopy and that even now it has not been used profitably in that field.

In subsequent years, methods were proposed for inferring the complete wave function in imaging experiments from intensity recordings which did not employ reference waves (see for example *Hoppe* (1970), *Schiske* (1968), *Erickson* and *Klug* (1970)). These methods have involved linear approximations and hence have had validity only in the limit of small phase and/or amplitude deviations across the wave front of interest. For the most part, they have also suffered from an excess of computation and have not gained wide acceptance.

Gerchberg and *Saxton* (1971) recently proposed that in many imaging

experiments, intensity recordings of wave fronts can be made conveniently in both the imaging and diffraction planes. Sets of quadratic equations were developed which defined the phase function across a wave in terms of its intensity in the image and diffraction planes. This method of analysis was not limited to small phase deviations, but again it required a large amount of computation.

In this paper we put forward a rapid computational method for determining the complete wave function (amplitudes and phases) from intensity recordings in the image and diffraction planes. The method depends on there being a Fourier Transform relation between the waves in these two planes and hence constrains the degree of temporal and/or spatial coherency of the wave. However, without a required degree of coherency in the experimental situation, it would be profitless to solve for a phase function anyway. The method should prove to be generally useful in electron microscopy, and under certain conditions even in ordinary light photography. It is also felt that it may have exciting implications for crystallography where only the X-ray diffraction pattern may be measured. Thus, if one has a crystal which may be reasonably inferred to be a phase object to X-ray illumination, its image would be contrastless and this bit of information plus the X-ray diffraction pattern would be sufficient to solve the "phase problem". Sophisticated methods for phase determination such as heavy atom replacement could be replaced or at least supplemented by the method to be described here.

The Algorithm

The basic algorithm is an iterative procedure which is shown schematically as Fig. 1. The input data to the algorithm are the amplitudes of the sampled image and diffraction plane intensity pictures measured. The amplitudes are proportional to the square roots of the measured intensities. The two sets of data are accessed once per complete iteration and hence require computer storage.

To begin, a random number generator is used to generate an array of random numbers between pi and minus pi which serve as the initial estimate of the phases corresponding to the sampled image amplitudes. These then are multiplied by the respective sampled image amplitudes and the Fourier Transform of this synthesized complex discrete function is done by means of the Fast Fourier Transform (FFT) algorithm of *Cooley* and *Tukey* (1965). The phases of the discrete complex function resulting from this transformation are calculated and combined with the corresponding sampled diffraction plane amplitude function. This function is then Fourier Transformed, the phases of the sample points computed and combined with the sampled image amplitude function to form a new estimate of the complex sampled image plane function and the process is repeated.

In what follows, we will discuss the problems of sampling and uniqueness and we will show by a simple proof that the squared error between the

amplitudes of the discrete functionus generated by the Fourier Transform operation and the discrete set of amplitudes derived from the measured intensities in the corresponding image or diffraction plane must decrease or at worst remain constant with each iteration. We will also display some pertinent results achieved on modelled problems. However, at this point some general remarks about the algorithm are in order.

To start the algorithm, we use a random number generator to arrive at a set of phase angles from a uniform distribution density between pi and minus pi. This is not necessary in every case and indeed there is every reason to suppose that an educated guess at the correct phase distribution would lessen the computation time required for the process to achieve an acceptable squared error. However, one initial phase distribution which will cause the algorithm to fail, is to have all phases equal to a constant when the intensity pattern in both the image and diffraction plane is centrosymmetric. This is because the phase distribution will not change under Fourier Transformation in this case. If one studies the algorithm as it is shown in Fig. 1, it may be realized that basically the mechanism relies on the fact that a change in the amplitude distribution alone in one domain of the Fourier Transform will result in changing both the amplitude and phase distributions in the opposite domain.

The fact that the algorithm is rapid computationally rests on using the Fast Fourier Transform (FFT) algorithm of *Cooley* and *Tukey* to do the transformations. This algorithm has reduced the time required to compute finite Fourier transforms by the fraction $(\log_2 N)/N$, where N is the number of points undergoing transformation. To give some idea of the time required to solve the phase problem on a large modern digital computer (ICL-ATLAS 2), we have been able to solve for the phase of a picture on a 32 by 32 grid (1024 points) in under 80 seconds. Thirty seven Fourier transforms were required during this time to reduce the squared error from 73.0 to 0.01.

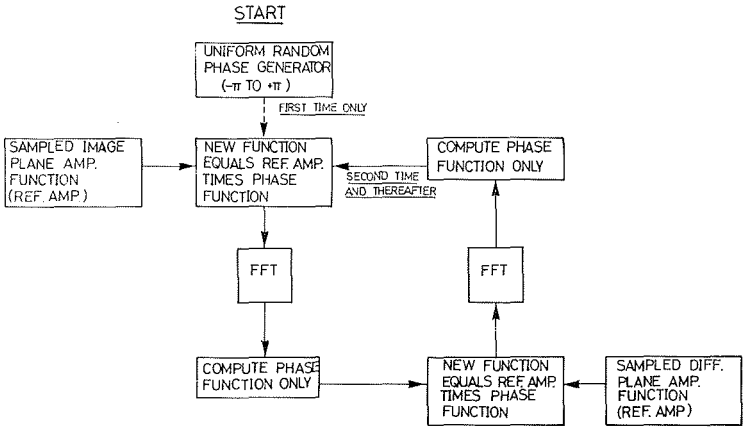


Fig. 1. Schematic drawing of phase determining algorithm.

The computer storage required depends of course on the density of points necessary for adequate sampling of the picture. In the example just given, 2048 words of storage were needed, to store the function generated by the FFT block (see Fig. 1) because the numbers are complex. Another 2048 words were required to store the reference amplitudes in the image and diffraction planes. Therefore, the total required storage was 4096 words.

The Uniqueness Problem

The information that is used to solve the phase problem is the measured intensity distribution in both the image and diffraction planes of the image forming system. But to start, it is clear that the solution on this basis will not be unique. By adding a constant but arbitrary phase to any function whose intensity in the image and diffraction planes is the same as the measured intensities, we generate a new function whose intensities in both planes will also be the same as those which were measured. Thus only relative phases of the solution functions are meaningful. Another case of inherent ambiguity occurs when both intensity distributions (diffraction and image planes) are centrosymmetric. The complex conjugate function to any solution found in this case will also be a possible solution. There may be more inherent ambiguities but thus far these are the only two that we have encountered in the trials we have completed. In most applications, these ambiguities are tolerable.

Proof that the Algorithm Error Must Decrease

We define the squared error as the sum of the squared differences between the amplitudes of points in either the image or diffraction plane (based on intensity measurements) and the amplitudes of those points calculated by the FFT in our algorithm (see Fig. 1). When the squared error is zero, we have found the correct phases of the points in both planes and we have a solution to the phase problem. Referring to Fig. 1, the energy (defined as the sum of the squared amplitudes of the finite discrete function) of the function which undergoes transformation is the same for either image or diffraction plane data. This is a consequence of Parseval's theorem and the fact that the image function is the Fourier Transform of the diffraction plane function. Also by Parseval's theorem, the energies of the two functions immediately on either side of the FFT block in Fig. 1 are equal. Now consider the operation of the algorithm at two sample points, one from the diffraction plane and one from the image plane. The situation is shown in Fig. 2. The reference amplitudes of the two points, based on the measured data are t_1 and t_2. The FFT of the diffraction plane function estimate (immediately after the FFT block in Fig. 1) yields the vector $\bar{g}_1$ at the image plane point. The algorithm corrects this vector in amplitude but retains its phase by adding the vector $\bar{c}_1$ to $\bar{g}_1$ to yield $\bar{h}_1$ the new image point estimate. All points in the image plane are similarly corrected and the function is transformed to yield $\bar{h}_2$ in the diffrac-

tion plane. Fig. 2b shows an arbitrary point in the diffraction plane. The $\bar{h}_2$ vector is the sum of $\bar{g}_2$ and $\bar{c}_2$ (the transforms of $\bar{g}_1$ and $\bar{c}_1$ respectively). By Parseval's theorem the sum of the squared amplitudes of $\bar{c}_1$ in domain 1 must equal the sum of the squared amplitudes of $\bar{c}_2$ in domain 2. But:

$$\sum_{\text{all points}} |c_1|^2 \triangleq \text{squared error}$$

and from Fig. 2 it is clear that the correction vector $\bar{d}_2$ at each point must be less than or at most equal to $\bar{c}_2$ in amplitude. Hence the squared error must decrease or remain constant with each pass through the FFT in the algorithm.

The Squared Error Limit

It is worthwhile to study and understand the action of the algorithm through circle point figures like those in Fig. 2. One thing that becomes clear is that for a finite limit to the squared error to exist, the algorithm must approach or reach a situation where the correction function ($\bar{c}$) is colinear with the estimated function ($\bar{h}$) in both domains 1 and 2 (see Fig. 2). Thus the estimate ($\bar{h}$) and the correction ($\bar{c}$) would have the same phase functions in both domains 1 and 2 at every point. This is a difficult condition to impose on two distinct functions and yet it has been achieved in every instance where we have forced the algorithm to fail. The squared error appeared to approach a finite limit other than zero at the same time that the phase function of the estimate appeared to approach a limiting function as well. There may be some characteristic way that the estimate approaches this limiting condition.

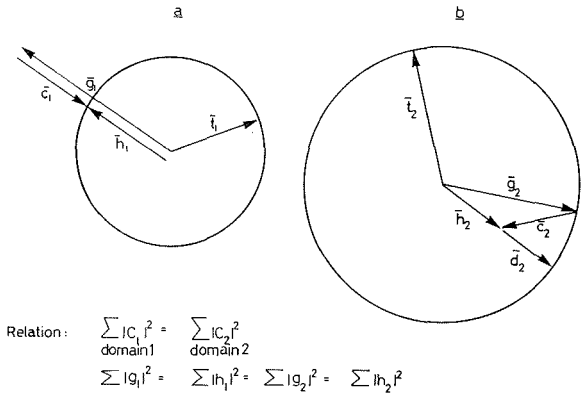


Fig. 2. The action of the algorithm on a) an arbitrary point in the image plane and b) an arbitrary point in the diffraction plane. "t_1" ("t_2") is the measured amplitude at the arbitrary image (diffraction) plane point. After FFT g_1 is the vector at point in "a". "c_1" is correction vector added. "h_1" is new estimate. "h_1" is then transformed to yield "b". In "b" d_2 is correction vector added to h_2 to form new estimate which then transformed to complete one iteration or loop of algorithm.

We have not examined this question closely but we can say that in the trials we have run, no obvious characteristic stands out. This is plain in the example of an incorrect function generated by the algorithm and displayed as Fig. 3.

Sampling Considerations

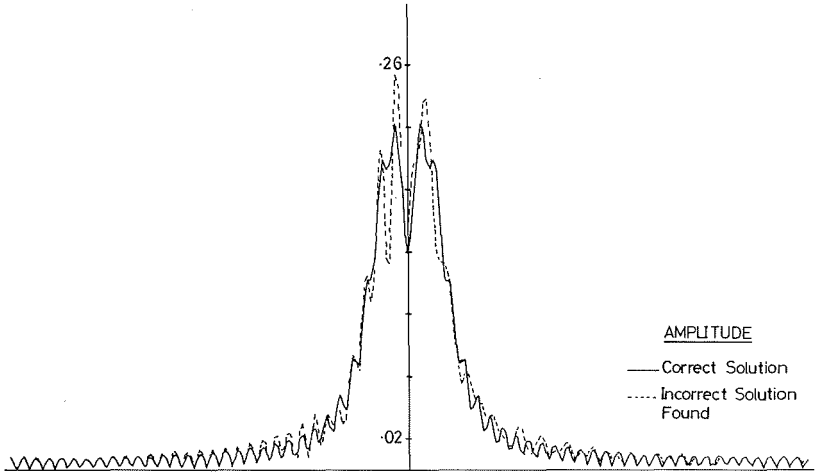
The only way that the algorithm has been forced to fail thus far is by inadequate sampling in either of the two domains. If a continuous function is sampled at a rate which is too "slow" to adequately portray its spectrum, the algorithm fails. At this point, we would not attempt to set minimum sampling rate limits. The problem stems from the fact that sampling occurs in both domains and the duration of the function must be infinite in at least one domain. At least one of the domains will present the function in a distorted way. The severity of this effect appears to vary with the function. It was a surprise to find that in the case of the "chirp function" (see Fig. 3) rect (2t) exp ($i30\pi t^2$), the sampling increment $t = 1/128$, was too large. That sampling increment corresponded to a Shannon-Whittaker sampling bandwidth containing better than 99% of the function energy. This has not been the case for every test function we have run. However, in every instance of our testing, the algorithm has succeeded in achieving a valid solution with the square error limit appearing to be zero when the function sampling was taken at a sufficiently high rate.

Typical Test Results

Fig. 3 shows the FFT of the function rect (2t) $\exp(i30\pi t^2)$ and the incorrect answer that the algorithm appeared to limit on. Fig. 3a compares the amplitudes of the two functions and Fig. 3b compares the phases. For purposes of illustration, only the results on this family of functions are given. Aside from this trial, which failed because of improper sampling, the other results which will be discussed yielded curves which were valid solutions of the phase problem. The history of this test leading to the results of Fig. 3 is shown in Table I.

Table II shows the trial history of the same function but this time the sampling increment was halved. The squared error was recorded only to three decimal places and went from a normalized value of 0.538 to 0.000 in approximately 65 loops of the algorithm as shown in Fig. 1. The amplitude of the function found was of course virtually identical to the correct one. Except for a constant phase shift and the fact that the function was complex conjugate to the one used to create the image, it was virtually perfect. It was noted previously that functions which, along with their transforms, have centrosymmetric amplitude distributions as this class does, can produce valid solution functions which are complex conjugates.

Table III shows the history of a solution run on a problem with slightly incompatible pictures in the diffraction and image planes. The algorithm was used on the amplitudes of the function rect(2t) $\exp(i15\pi t^2)$. The amplitudes of the FFT of this function were modified by suppressing all those amplitudes



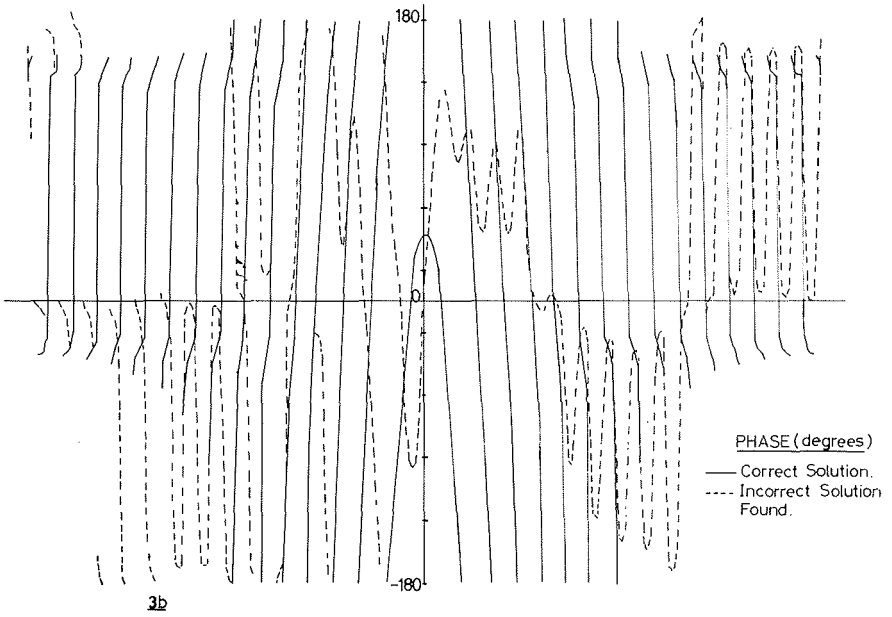


Fig. 3. Point plot of a) amplitude and b) phases of FFT of sampled function rect (2t) exp ($i30\pi t^2$). The incorrect solution found by the algorithm is compared to the correct solution. Details of this trial are recorded in Table I.

Table I

Function: rect (2t)exp(i30πt^2)

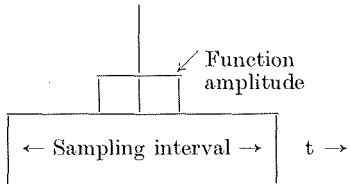
Sampling increment: $\Delta t = 1/128$

Number of sampling points: 256

Result: Algorithm failed

Number of times FFT called: 172

Squared error history							
1	.634	26	.030	51	.014	16	.013
2	.190	21	.029	52	.014	11	.013
3	.121	28	.029	53	.014	18	.012
4	.086	29	.028	54	.014		
5	.066	30	.028	55	.014		Goes no lower
6	.055	31	.021	56	.013		
7	.048	32	.026	57	.013		
8	.044	33	.025	58	.013		
9	.041	34	.023	59	.013		
10	.040	35	.022	60	.013		
11	.039	36	.020	61	.013		
12	.038	37	.019	62	.013		
13	.031	38	.018	63	.013		
14	.036	39	.017	64	.013		
15	.036	40	.016	65	.013		
16	.035	41	.015	66	.013		
17	.034	42	.015	61	.013		
18	.034	43	.015	68	.013		
19	.033	44	.015	69	.013		
20	.032	45	.014	70	.013		
21	.032	46	.014	71	.013		
22	.031	47	.014	72	.013		
23	.031	48	.014	73	.013		
24	.031	49	.014	74	.013		
25	.030	50	.014	75	.013		



Total function energy (same basis as history): 0.50

less than 1/30th the maximum. Thus a situation was simulated in which the dynamic range of the recording medium was in the neighbourhood of 30 DB limited by noise at the low end. The results were very good indeed. The error decreased from 0.639 to 0.003 in 31 loops of the algorithm. The energy of the function in the two domains is not the same and the error limits at this value. The phases of the points in the solution found are quite closely correct ($\pm 2°$) for corresponding amplitudes greater than 10% the maximum. The phase error becomes as high as $\pm 10°$ at some points of lesser amplitude.

Table II

Function: rect (2t)exp(i30πt^2)

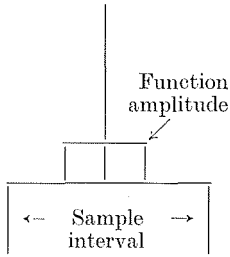
Sampling increment: $\Delta t = 1/256$

Number of Sampling points: 512

Result: Correct solution

Number of times FFT called: 130

Squared error history											
1	.538	26	.025	51	.012	76	.009	101	.006	126	.001
2	.146	27	.024	52	.012	77	.008	102	.006	127	.001
3	.110	28	.023	53	.012	78	.008	103	.005	128	.001
4	.091	29	.022	54	.012	79	.008	104	.005	129	.001
5	.081	30	.021	55	.011	80	.008	105	.005	130	.000
6	.015	31	.020	56	.011	81	.008	106	.004		
7	.010	32	.019	57	.011	82	.008	107	.004		
8	.065	33	.018	58	.011	83	.008	108	.004		
9	.061	34	.018	59	.011	84	.008	109	.003		
10	.057	35	.017	60	.010	85	.008	110	.003		
11	.054	36	.017	61	.010	86	.008	111	.003		
12	.051	37	.016	62	.010	87	.008	112	.003		
13	.049	38	.016	63	.010	88	.008	113	.003		
14	.047	39	.015	64	.010	89	.008	114	.002		
15	.045	40	.015	65	.010	90	.008	115	.002		
16	.043	41	.015	66	.010	91	.007	116	.002		
17	.040	42	.015	67	.009	92	.007	117	.002		
18	.038	43	.014	68	.009	93	.007	118	.002		
19	.036	44	.014	69	.009	94	.007	119	.002		
20	.034	45	.014	70	.009	95	.007	120	.002		
21	.032	46	.013	71	.009	96	.007	121	.002		
22	.031	47	.013	72	.009	97	.007	122	.001		
23	.029	48	.013	73	.009	98	.007	123	.001		
24	.028	49	.013	74	.009	99	.007	124	.001		
25	.026	50	.012	75	.009	100	.006	125	.001		



Total function energy (same basis as history): 0.50

Perusal of the solution histories indicates that the progress of the algorithm toward a solution becomes less rapid as the error decreases. In this respect these results are not characteristic of all the trials we have made. Some functions have actually moved toward zero squared error at a geometric rate with a factor of around 0.5.

R. W. Gerchberg acknowledges the support of the United States National Institutes of Health by way of Fellowship NIH(1 F03 GM49953-01). *W.O. Saxton* acknowledges the support of the Science Research Council.

Table III

Function: rect (2t)exp(i15πt^2) in domain 1, all values less than 3% maximum are set to zero in domain 2

Sampling increment: Δt = 1/128

Number of sampling points: 256

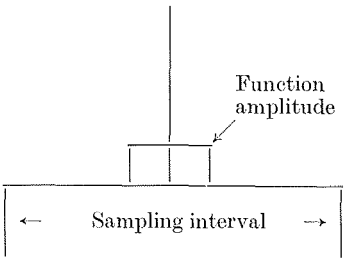
Result: Acceptable solution (see text)

Number of times FFT called: 356

Squared error history

1	.639	26	.009	51	.004
2	.083	27	.009	52	.004
3	.048	28	.009	53	.004
4	.031	29	.008	54	.004
5	.024	30	.008	55	.004
6	.021	31	.008	56	.004
7	.019	32	.008	57	.004
8	.018	33	.007	58	.004
9	.017	34	.007	59	.004
10	.017	35	.007	60	.004
11	.016	36	.007	61	.004
12	.015	37	.006	62	.004
13	.015	38	.006	63	.004
14	.014	39	.006	64	.003
15	.013	40	.006		
16	.013	41	.005		
17	.012	42	.005		
18	.011	43	.005		
19	.011	44	.005		
20	.011	45	.005		
21	.011	46	.005		
22	.010	47	.004		
23	.010	48	.004		
24	.010	49	.004		
25	.010	50	.004		

Goes no lower



Total function energy (same basis as history): 0.50

References

[1] *J. W. Cooley* and *J. W. Tukey*, Mathematics of Computation *19* (1965) 297.
[2] *H. Erickson* and *A. Klug*, Berichte der Bunsen-Gesellschaft *74* (1970) 1129.
[3] *D. Gabor*, Nature *161* (1948) 777.
[4] *R. Gerchberg* and *W. Saxton*, Optik *34* (1971) 275.
[5] *W. Hoppe*, Acta Cryst. *A 26* (1970) 414.
[6] *P. Schiske*, 4th European Conference on Electron Microscopy Rome (1968).