NASA CR- 166743

FINAL REPORT

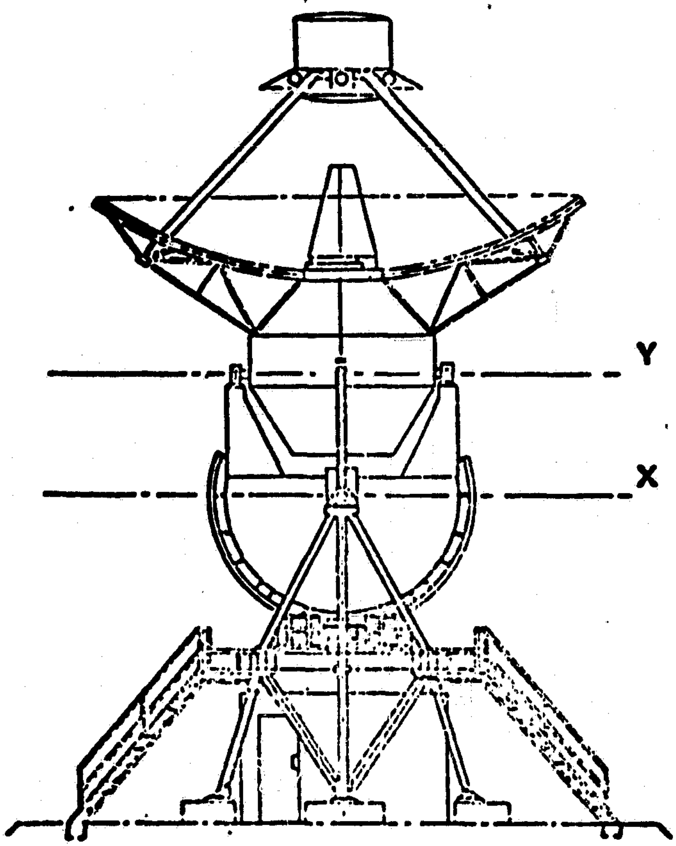
STUDY OF HIGH SPEED COMPLEX NUMBER ALGORITHMS

Contract No. NAS5-25994

National Aeronautics and Space Administration

Goddard Space Flight Center

Greenbelt, Maryland



Rodney Heisler

School of Engineering

Walla Walla College

College Place, Washington

(NASA-CR-166743) STUDY OF HIGH SPEED COMPLEX NUMBER ALGORITHMS Final Report (Walla Walla College, College Place) 44 p HC A03/MF A01 CSCL 09B

N82-18914

Unclas
G3/61 08346

I. INTRODUCTION

Basic to the design of reflecting antenna systems is the need to compute the far-field radiation pattern for a given antenna geometry, feed placement, and feed radiation characteristic. A simple, yet fast solution to this classic problem has proven to be a formidable challenge. This is especially true for very large reflectors with extensive current distributions and off-focus feeds with their associated asymmetry. These problems are made troublesome because of their large computational requirements.

Recent interest in this problem [1-4] has resulted in the new application of powerful mathematical techniques to produce a more time-efficient means of computing the radiation intensity. Galindo-Israel and Mittra [1] have reformulated the two-dimensional radiation integral to a double Fourier integral "representation." The Fourier integral is evaluated by expanding the integrand in a double summation of circular functions and modified Jacobi polynomials. The result is a rapidly convergent series representation of the radiation intensity, the zeroth term being $J_1(x)/x$, the pattern for a uniform, cophase aperture. Mittra, et al. [2] extend the work of the first paper to offset paraboloid reflectors and introduce a more rigorous corrective technique to compensate for the non-Fourier structure of the radiation integral.

This paper presents a new method which evaluates the radiation integral on the antenna surface. The algorithm is computationally efficient and produces the far-field radiation pattern along planer cuts at any angle

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

ILLUSTRATIONS

PRECEDING PAGE BLANK NOT FILMED

ILLUSTRATIONS (Continued)

I. INTRODUCTION

Basic to the design of reflecting antenna systems is the need to compute the far-field radiation pattern for a given antenna geometry, feed placement, and feed radiation characteristic. A simple, yet fast solution to this classic problem has proven to be a formidable challenge. This is especially true for very large reflectors with extensive current distributions and off-focus feeds with their associated asymmetry. These problems are made troublesome because of their large computational requirements.

Recent interest in this problem [1-4] has resulted in the new application of powerful mathematical techniques to produce a more time-efficient means of computing the radiation intensity. Galindo-Israel and Mittra [1] have reformulated the two-dimensional radiation integral to a double Fourier integral "representation." The Fourier integral is evaluated by expanding the integrand in a double summation of circular functions and modified Jacobi polynomials. The result is a rapidly convergent series representation of the radiation intensity, the zeroth term being $J_1(x)/x$, the pattern for a uniform, cophase aperture. Mittra, et al. [2] extend the work of the first paper to offset paraboloid reflectors and introduce a more rigorous corrective technique to compensate for the non-Fourier structure of the radiation integral.

This paper presents a new method which evaluates the radiation integral on the antenna surface. The algorithm is computationally efficient and produces the far-field radiation pattern along planer cuts at any angle

ϕ through the three-dimensional pattern. The physical optics approximation is used to compute the induced surface current which is the input to the algorithm. The method is developed for focal-point and translated feeds and is easily extended to offset antennas and arbitrary surfaces.

On the premise that simplification and efficiency may come from first generalizing, the radiation integral is reformulated to three dimensions. The result is shown to be a triple Fourier integral. To evaluate this integral, an extremely fast algorithm is introduced which evaluates, along planes of constant ϕ, a subset of the total three-dimensional Fourier transform results. No approximations are made other than those normally associated with digitization and the discrete Fourier transform (DFT). To further reduce the computation time, the Winograd Fourier transform algorithm (WFT) is used in place of the standard radix-2 FFT when a DFT is called for in the algorithm. The any ϕ angle feature of the program is implimented using a technique similar to one used in computerized tomography (cross-sectional x-rays) [5].

While other methods exist for evaluating the radiation integral, this new theory brings a fast, simple and direct approach. Due to its speed it is especially useful for very large asymmetric antennas.

II. THE THREE-DIMENSIONAL RADIATION INTEGRAL AS A FOURIER TRANSFORM

The far-field radiation intensity of a volume distribution of current may be expressed in terms of the three-dimensional radiation integral

$$\vec{E}(\theta,\phi) = \iiint \vec{K}(r',\theta',\phi')e^{-jk(r'-\vec{r}'\cdot\hat{R})}\, dv' \qquad (1)$$

The geometry for (1) is given in Figure 1. In formulating this equation, it is assumed that the phase center is at the origin, the observation point is far from the current distribution and the currents are bounded in a region dimensionally small compared to R.

The $-\vec{r}'\cdot\hat{R}$ ($\hat{R}$ a unit vector) term is the distance from the source point to a plane through the origin and normal to $\hat{R}$. Hence, the exponent $k(r'-\vec{r}'\cdot\hat{R})$ is the total phase delay modulo $2\pi R$.

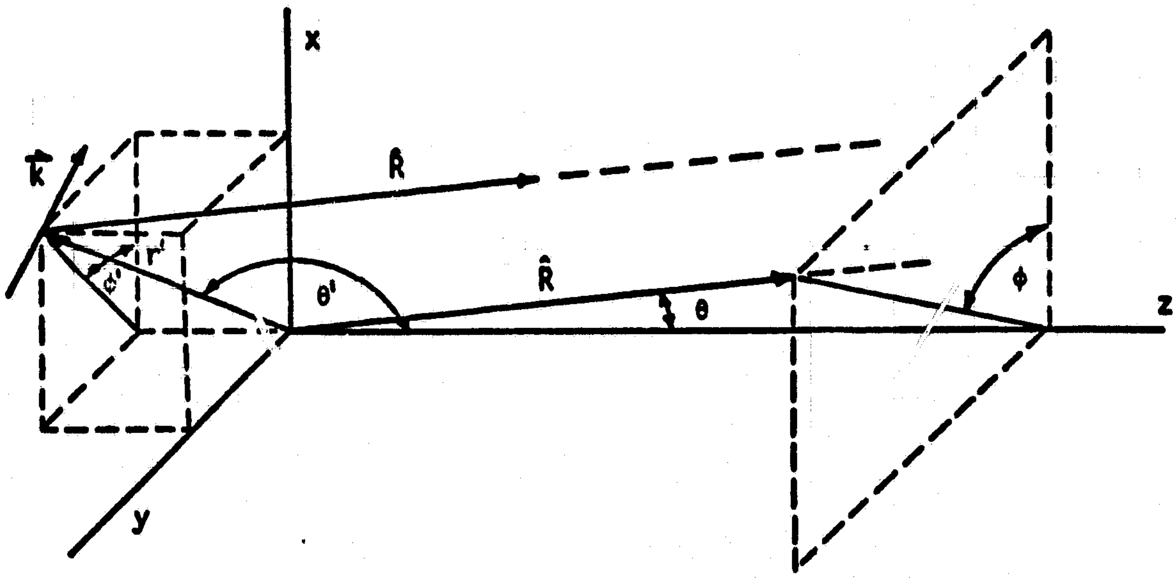


Figure 1. Geometry for the radiation integral

Using the direction cosines u, v, and w, we can write

$$\vec{r}'\cdot\hat{R} = r'_z w + r'_x u + r'_y v$$

$$= r'\cos\theta'\cos\theta + r'\cos\phi'\sin\theta'\cos\phi\sin\theta + r'\sin\phi'\sin\theta'\sin\phi\sin\theta$$

$$= r'\cos\theta'\cos\theta + r'\sin\theta'\sin\theta\cos(\phi-\phi')$$

The radiation integral of (1) may now be rewritten as

$$\vec{E}(\theta,\phi) = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \int_{r=0}^{\infty} \vec{K}(r',\theta',\phi')e^{-jkr'}e^{jkr'\cos\theta'\cos\theta} \times e^{jkr'[\sin\theta'\sin\theta\cos(\phi-\phi')]}r'^2\sin\theta' dr' d\theta' d\phi' \tag{2}$$

Equation (2) is a three-dimensional Fourier transform in spherical coordinates as presented by Bracewell [6]. An important corollary to this observation is that the far-field radiation pattern may be computed as the three-dimensional Fourier transform of a volume current distribution. Techniques will be presented for efficiently evaluating equation (2).

If the current exists only on a surface within the volume, the volume current function may be expressed in terms of a surface current as

$$\vec{K}(r',\theta',\phi') = \vec{J}(\theta',\phi')\delta(r'-\rho)$$

where $\rho = \rho(\theta',\phi')$ defines the geometry of the surface. Substituting this expression into (2) and integrating with respect to r' (with the aid of the sifting property of the Dirac function) we arrive at

$$\vec{E}(\theta,\phi) = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \vec{J}(\theta',\phi')e^{-jk\rho}e^{jk\rho\cos\theta'\cos\theta} \times e^{jk\rho[\sin\theta'\sin\theta\cos(\phi-\phi')]}\rho^2\sin\theta' d\theta' d\phi' \tag{3}$$

Equation (3) agrees with equation (5) of Galindo-Israel [1]. To phrase it in exactly the same form requires two modifications. First, the phase center must be moved from the origin to a point defined by the feed position vector $\vec{\varepsilon}$. Secondly, the integration must be transferred from the reflector surface to the aperture plane. The latter is accomplished by use of the "Jacobian of the transformation" between the reflector surface and the projected aperture. A single component of $\vec{J}(\theta',\phi')$, say $J_x(\theta',\phi')$, may be transformed to an "equivalent" aperture distribution over the circular disk of the aperture. This results in the transformation

$$J_x(\theta',\phi')\rho^2\sin\theta' d\theta' d\phi' = J_x(\theta',\phi')\sqrt{1 + (dz/ds)^2}\, s ds d\phi'$$
$$= f(s,\phi') s ds d\phi'$$

$f(s,\phi')$ is the equivalent aperture distribution and is not to be confused with the physical distribution on the aperture such as might be derived approximately by the ray technique.

Equation (3) now becomes

$$E(\theta,\phi) = \int_{\phi=0}^{2\pi} \int_{s=0}^{a} f(s,\phi') e^{-jk\rho} e^{jk\rho\cos\theta'\cos\theta} \times e^{jk\rho[\sin\theta'\sin\theta\cos(\phi-\phi')]} s ds d\phi' \tag{4}$$

which is the same as (5) in Galindo-Israel [1] except for a difference in phase center. Equation (4) is not a Fourier transform as $\exp(jk\rho \cos\theta'\cos\theta)$ is a function of both source and observation coordinates.

III. AN EFFICIENT ALGORITHM FOR THE THREE-DIMENSIONAL DISCRETE FOURIER TRANSFORM

The three-dimensional Fourier transform of a volume source is defined by Bracewell [6] as

$$F(u,v,w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y,z) e^{-j2\pi(ux + vy + wz)} dx dy dz \tag{5}$$

where u, v, and w are the direction cosines

$$u = \sin\theta\cos\phi$$
$$v = \sin\theta\sin\phi \tag{6}$$
$$w = \cos\theta$$

Note that while there appear to be three independent variables in this transform, i.e., the frequencies u, v, and w, there are in fact only two geometric variables, θ and ϕ. These may be solved for as

$$\theta = \sin^{-1}\sqrt{u^2 + v^2} = \cos^{-1}w$$
$$\phi = \tan^{-1}v/u \tag{7}$$

Similarly, the three-dimensional discrete Fourier transform (DFT) is given by the triple summation

$$F(k_3,k_2,k_1) = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1}\sum_{n_3=0}^{N_3-1} f(n_3,n_2,n_1)e^{-j\frac{2\pi}{N_3}n_3k_3}e^{-j\frac{2\pi}{N_1}n_1k_1}e^{-j\frac{2\pi}{N_2}n_2k_2} \tag{8}$$

for $k_1 = 0, 1, 2, \ldots, N_1-1$

$k_2 = 0, 1, 2, \ldots, N_2-1$

$k_3 = 0, 1, 2, \ldots, N_3-1$

This procedure may be broken into three operations by the use of intermediate results. Consider the intermediate transforms

$$g(k_3,n_2,n_1) = \sum_{n_3=0}^{N_3-1} f(n_3,n_2,n_1)e^{-j\frac{2\pi}{N_3}n_3k_3} \tag{9}$$

$n_1 = 0, 1, 2, \ldots, N_1-1$

$n_2 = 0, 1, 2, \ldots, N_2-1$

$k_3 = 0, 1, 2, \ldots, N_3-1$

$$h(k_3,n_2,k_1) = \sum_{n_1=0}^{N_1-1} g(k_3,n_2,n_1)e^{-j\frac{2\pi}{N_1}n_1k_1} \tag{10}$$

$k_1 = 0, 1, 2, \ldots, N_1-1$

$n_2 = 0, 1, 2, \ldots, N_2-1$

$k_3 = 0, 1, 2, \ldots, N_3-1$

From these the DFT is found to be

$$F(k_3,k_2,k_1) = \sum_{n_2=0}^{N_2-1} h(k_3,n_2,k_1)e^{-j\frac{2\pi}{N_2}n_2k_2} \tag{11}$$

$$k_1 = 0,\ 1,\ 2,\ .\ .\ .\ .\ ,\ N_1-1$$

$$k_2 = 0,\ 1,\ 2,\ .\ .\ .\ .\ ,\ N_2-1$$

$$k_3 = 0,\ 1,\ 2,\ .\ .\ .\ .\ ,\ N_3-1$$

Comparing the frequency variables of the Fourier integral and the DFT, we note the correspondence

$$u \sim \frac{k_1}{N_1T_1} \quad , \quad v \sim \frac{k_2}{N_2T_2} \quad , \quad w \sim \frac{k_3}{N_3T_3} \tag{12}$$

where each T is the sampling period for the respective axes. Combining (12) and (7), we arrive at expressions for θ and ϕ in terms of the frequency indices of the DFT

$$\theta = \sin^{-1}\sqrt{\left(\frac{k_1}{N_1T_1}\right)^2 + \left(\frac{k_2}{N_2T_2}\right)^2} = \cos^{-1}\left(\frac{k_3}{N_3T_3}\right)$$

$$\phi = \tan^{-1}\left(\frac{k_2}{N_2T_2} \cdot \frac{N_1T_1}{k_1}\right) \tag{13}$$

A direct evaluation of the DFT in (8) is impractical, even for modest transform lengths. Nor does use of an FFT algorithm provide sufficient improvement. However, the computation time may be reduced by orders of magnitude in the case where one is interested in a limited amount of the total DFT information. It is possible in this way to compute a two-dimensional planer cut through the far-field radiation pattern with no compromise in accuracy. Once the algorithm is developed, it may be extended to include an any angle planer cut.

The simplification is begun by selecting $k_1 = 0$. From equation (13) this requires that $\phi = 90^\circ$. Next, the value of k_2 is chosen. From (13) this determines the polar angle $\theta = \sin^{-1}(k_2/N_2T_2)$ and forces k_3 to a specific value given by

$$\left(\frac{k_3}{N_3T_3}\right)^2 + \left(\frac{k_2}{N_2T_2}\right)^2 = 1 \tag{14}$$

Clearly, there are going to be digitization problems in evaluating (14) for k_3 based on a selected integer value of k_2. This works to our advantage in decreasing the computation time.

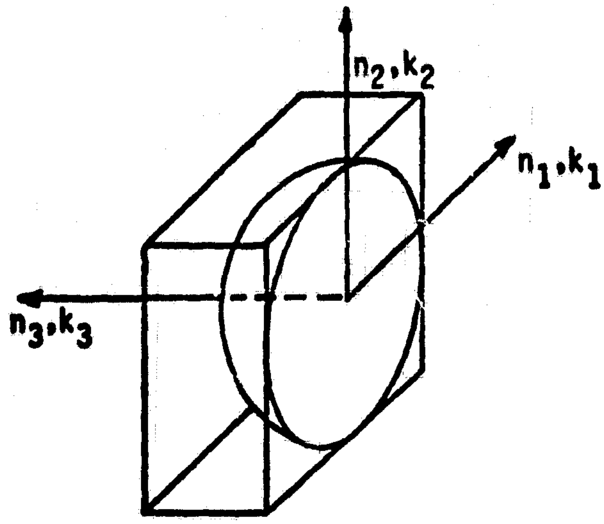


Figure 2. Antenna Geometry for the DFT

The summation in equation (9) may now be reduced. k_3 is a determined single value and along the n_3 dimension (see Figure 2) there is only one non-zero data element for each (n_1,n_2) pair. Hence (9) becomes

$$g(k_3,n_2,n_1) = f(n_3,n_2,n_1)e^{-j\frac{2\pi}{N_3}n_3k_3} \tag{15}$$

$n_1 = 0, 1, 2, \ldots, N_1-1$

$n_2 = 0, 1, 2, \ldots, N_2-1$

k_3 a constant

Equation (15) describes a coalescing of the data to a planer array and hence a reduction to two-dimensions as shown in Figure 3. Each point in the n_1, n_2 plane corresponds to a $g(k_3, n_2, n_1)$. The $f(n_3, n_2, n_1)$, (i.e. the current density) data will be present in polar form so that the complex multiplications indicated here are simple additions of $\frac{2\pi}{N_3} n_3 k_3$ to the phase of $f(n_3, n_2, n_1)$. Furthermore, the phase term $\frac{2\pi}{N_3} n_3 k_3$ need be calculated only once for each n_3 and recalled as a linear array variable for the individual phase additions.

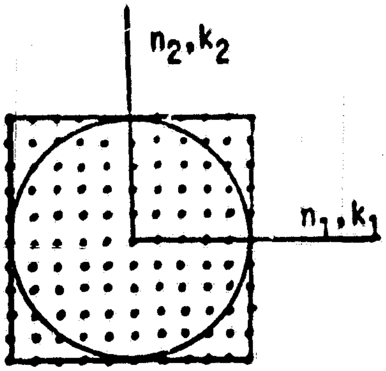


Figure 3. The coalesced two-dimensional array

With the selection of $k_1 = 0$ comes the simplification of equation (10) to

$$h(k_3, n_2, 0) = \sum_{n_1=0}^{N_1-1} g(k_3, n_2, n_1) \qquad (16)$$

$n_2 = 0, 1, 2, \ldots, N_2 - 1$

k_3 a constant

Equation (16) describes the coalescing of a two-dimensional array to one dimension. The $g(k_3, n_2, n_1)$ data along each row are summed to form the linear array $h(k_3, n_2, 0)$ in n_2.

The final operation is described by equation (11)

$$F(k_3,k_2,0)=\sum_{n_2=0}^{N_2-1} h(k_3,n_2,0)e^{-j\frac{2\pi}{N_2}n_2k_2} \tag{17}$$

$$k_2 = N_2T_2\sin\theta$$

$$k_3 = N_3T_3\sqrt{1-(k_2/N_2T_2)^2}$$

Note that (17) is not a DFT since the sum is performed for only one value of k_2. It is in fact only one of the N_2 harmonic constituents of the DFT corresponding to one value of θ in the spectrum. In practice, one may want to compute the DFT of $h(k_3,n_2,0)$, as digitization problems in evaluating k_3 make (17) applicable for many values of θ (or k_2) based on a single value of k_3.

Generally, it is not necessary to execute the above process for more than several values of k_3. For NT = (2520) (1/8) in equation (14), k_3 = 315 for k_2 in the range of zero to 17. This requires that for θ between zero and 3.1°, the generation of the $h(k_3,n_2,0)$ one-dimensional array by coalescing to a plane and thence to a line need be done only once. Radiation data at θ increments within this bound are found by evaluating (17) for k_2 between zero and 17. This may be done by computing the FFT or WFT [7] of $h(k_3,n_2,0)$ and using only the first 18 transform results. Subsequent integer values of k_3 yield θ bounds of 3.1° to 5.5°, 5.5° to 7.1°, etc.

The three-dimensional DFT produces a spectrum in the three variables k_1, k_2, and k_3. Such a thing is difficult to represent graphically and fortunately, in this case it is not necessary as $k_1 = 0$. What results, then, is a two-dimensional spectrum or a surface in k_2 and k_3. Only a portion of this surface is of interest; that section defined by the elliptic relationship between k_2 and k_3 in equation (14). It is the spectrum along this elliptic arc (see Figure 4) that describes the far-field radiation pattern for a constant ϕ cut.

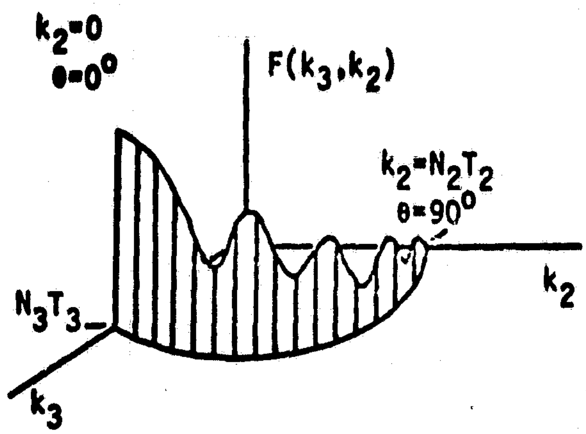


Figure 4. Two-dimensional spectrum for $(\frac{k_2}{N_2T_2})^2 + (\frac{k_3}{N_3T_3})^2 = 1$

The DFT described in (8) is a "one-sided" transformation, i.e., it operates on data sequences in positive time or position only. Furthermore, the DFT is cyclic and requires that the input data be periodic. The first condition leads to a necessary format of data input. The second requires a large number of zeros to be embedded in the data to adequately isolate the antenna from distant images regularly spaced three-dimensionally about it. The geometric association of data to the (n_1,n_2,n_3) indices is demonstrated in Figure 5. The location of the origin for the n_1 and n_2 axes is along the center line of the antenna and is a clear extension from two-dimensional DFT theory. There is no obvious point of symmetry along the n_3 axis to fix the three-dimensional origin. Consequently, it would be nice to discover that the choice did not affect the DFT results and could be made arbitrarily. It will be shown that this is true for the amplitude results, but not the phase.

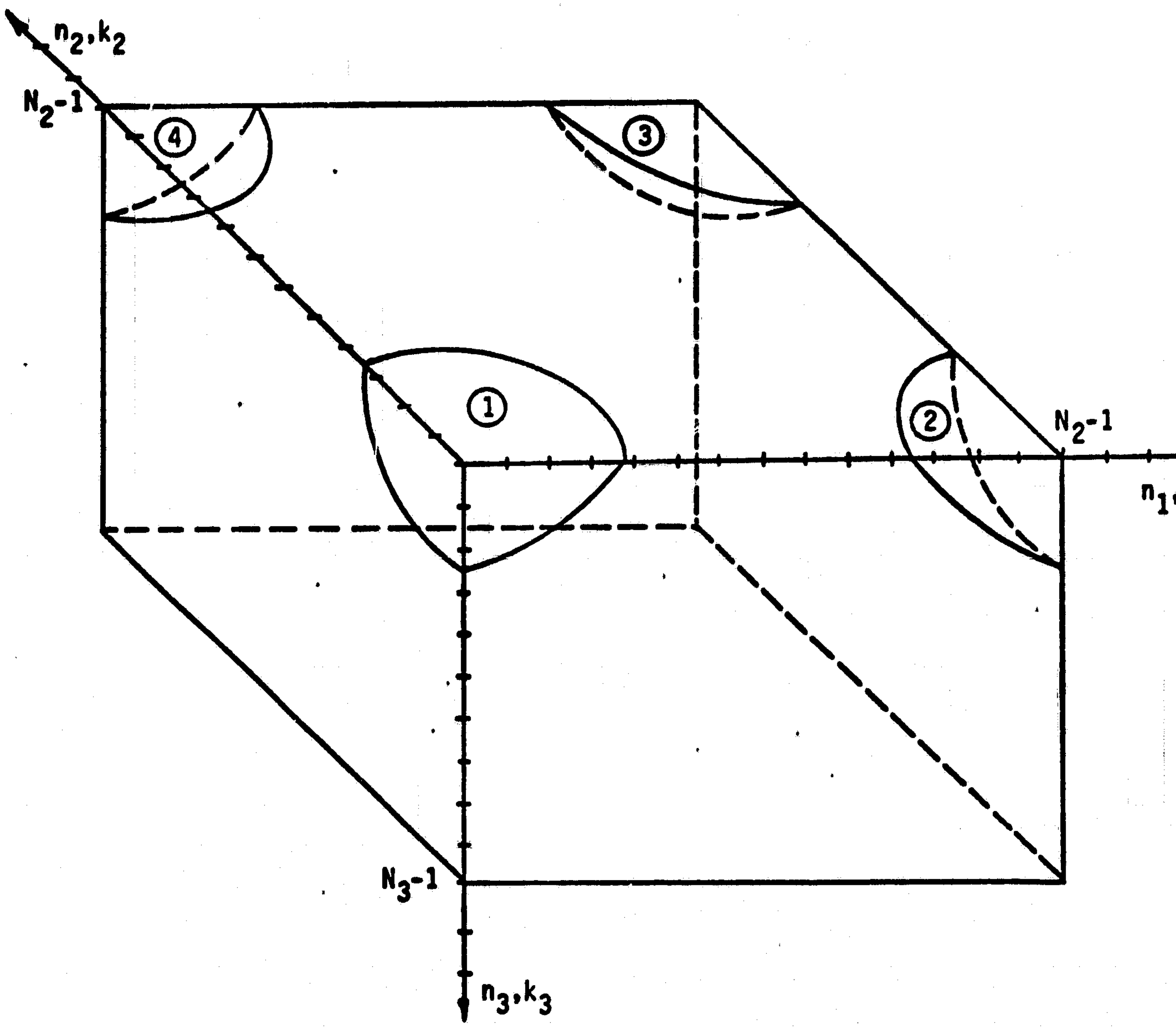


Figure 5. Data input format for the three-dimensional DFT

Equation (15) contains the n_3 coordinate dependence of the DFT. A shift in the origin along the n_3 axis by an integer amount γ will alter the phase term to

$$e^{-j\frac{2\pi}{N_3}k_3(n_3+\gamma)} = e^{-j\frac{2\pi}{N_3}k_3\gamma}\ e^{-j\frac{2\pi}{N_3}k_3 n_3}$$

$$= e^{-j\beta}\ e^{-j\frac{2\pi}{N_3}k_3 n_3}$$

When this data is coalesced to a line, the resulting one-dimensional data vector will have a constant phase shift as seen from equation (16)

$$\sum_{n_1=0}^{N_1-1} f(n_3,n_2,n_1)e^{-j\frac{2\pi}{N_3}n_3k_3}e^{-j\beta}$$

$$= e^{-j\beta}\sum_{n_1=0}^{N_1-1} g(k_3,n_2,n_1)$$

$$= e^{-j\beta}h(k_3,n_2,k_1)$$

Hence, a constant phase shift, β, will be present in all the DFT terms, but the amplitude results remain unaltered. A constant phase shift is certainly not a problem. However, difficulty does arise when the algorithm is repeated for other values of k_3. Recall that repeated application is necessary in order to sweep through a broad range in θ and that the number of separate values of k_3 is determined by this θ range (three applications for θ out to 7.1°). Clearly, β is a function of k_3 and the separate composite ranges will each be shifted by a different constant phase. If phase information is important, it is an easy matter to compute these phase terms and subtract them from the results.

Finally, the direction of the n_3 coordinate is important. The Fourier transform may be formulated with either a plus or minus exponential phase term in the integrand. The n_3 coordinate orientation shown in Figures 2 and 5 is consistent with the negative phase form and when used with this definition produces correct transform results.

IV. AN ANY ANGLE PATTERN CUT

The selection of $k_1 = 0$ in the previous section was significant in that it allowed us to simplify the three-dimensional DFT by coalescing the planer data to a linear array. However, it also limited our consideration to the $\phi = 90^\circ$ plane. It is imperative that we be able to examine the radiation intensity along other planer cuts through the pattern. This may be done using a projection technique similar to that used in computerized tomography [5] and multiangular scanning in gases [8].

Equation (16) indicates that the planer data (see Figure 3) be summed along rows of constant n_2 to produce a one-dimensional data vector along the n_2 (ϕ=90°) axis. Following the same procedure, the data may be projected to a diagonal line n_2' (see Figure 6) at an angle ϕ to the n_1 axis. Conceptually, this is equivalent to a rotation of the coordinate system about the n_3 axis. The resulting data vector is then operated on by equation (17) to produce a planer cut at the new angle ϕ.

From Figure 6, the complex data are summed along grid tubes (of width T) perpendicular to the diagonal axis. Looking along these tubes, one observes that the data points do not lie in the centers of the new grid squares and that a few squares contain no date points, while others contain two. The first difficulty is related to quantitization error which is always present when converting a continuous function to discrete data. The assumption that an interval of a continuous function may be

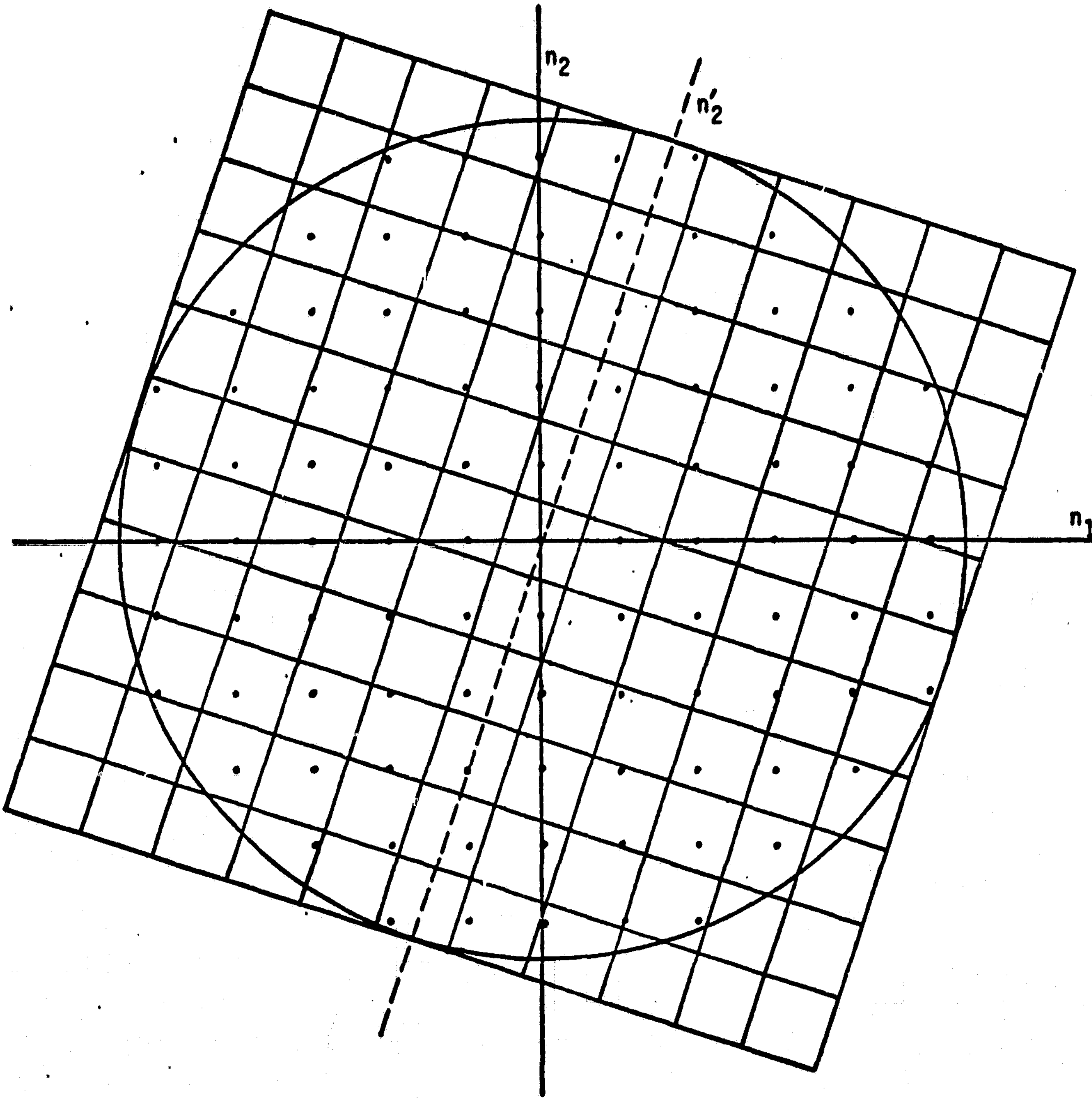


Figure 6. Coalescing to a diagonal line for an any angle cut

represented by a constant value is a first order type of approximation; that the constant is not the value of the function at the center point is second order.

As for the second difficulty, the concern here is that the sum along each grid tube be essentially the same as if resulting from a regular grid with one and only one value associated with each square. This will be true if the sampling rate is sufficient and if the sums are adjusted to account for any irregularity in the number of data in each tube.

V. CALCULATION OF THE INDUCED ANTENNA CURRENTS

The physical optics approximation was used to compute the antenna surface currents. This comes from applying the magnetic boundary conditions

$$\vec{J} \approx 2\hat{n} \times \vec{H} \tag{18}$$

Generally, this is considered sufficiently accurate for studies of the main beam and several side lobes. It is only necessary to determine the unit vector ($\hat{n}$) normal to the surface and the magnetic field ($\vec{H}$). The currents are derived for a paraboloid reflector with a $(-\cos\theta')^n$ offset feed and a feed displacement $\vec{\rho}_\varepsilon = (x_\varepsilon, y_\varepsilon, z_\varepsilon)$. Assuming a unit electric polarization in the y-direction, a $1/\rho'$ space divergence of the field and a $e^{-jk\rho'}$ phase delay, we determine

$$\begin{aligned}
J_x &= \frac{-x'y/2f}{\sqrt{1 + (\sigma/2f)^2}\sqrt{x'^2 + z'^2}}(-\cos\theta')^n \frac{e^{-jk\rho'}}{\rho'} \\
J_y &= \frac{-z' + x'x/2f}{\sqrt{1 + (\sigma/2f)^2}\sqrt{x'^2 + z'^2}}(-\cos\theta')^n \frac{e^{-jk\rho'}}{\rho'} \qquad (19) \\
J_z &= \frac{yz'/2f}{\sqrt{1 + (\sigma/2f)^2}\sqrt{x'^2 + z'^2}}(-\cos\theta')^n \frac{e^{-jk\rho'}}{\rho'}
\end{aligned}$$

The geometric variables for (19) are defined in Figure 7.

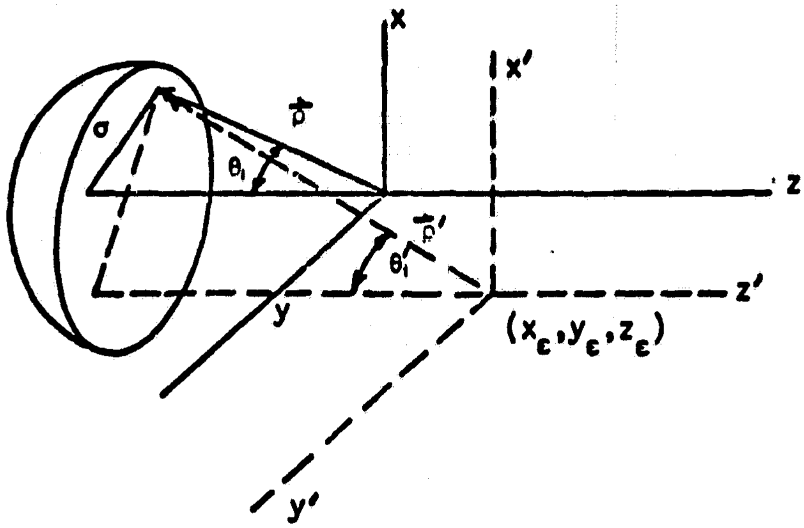


Figure 7. Geometry of focal-point and translated feeds.

VI. COMPUTATIONAL RESULTS

Performance of the new three-dimensional algorithm is verified by comparing computed results with other published work. A driver program was used to generate, on the paraboloid, a surface current of constant amplitude. Furthermore, a phase advance $e^{j2\pi n_3/T_3}$ was assigned to the current at each point so that the field at the aperture would be both uniform and cophase. This provided a test of the new algorithm with classical theory. The computed result was the expected $J_1(x)/x$ (see Figure 8).

Radiation patterns were computed for currents derived by the physical-optics approximation. The reflector is characterized by an $f/D = 0.5$, and the feed pattern as a circularly symmetric $(-\cos\theta')^n$. The value of n was chosen to produce a -10dB taper to the edge of the reflector. Figures 9(a)

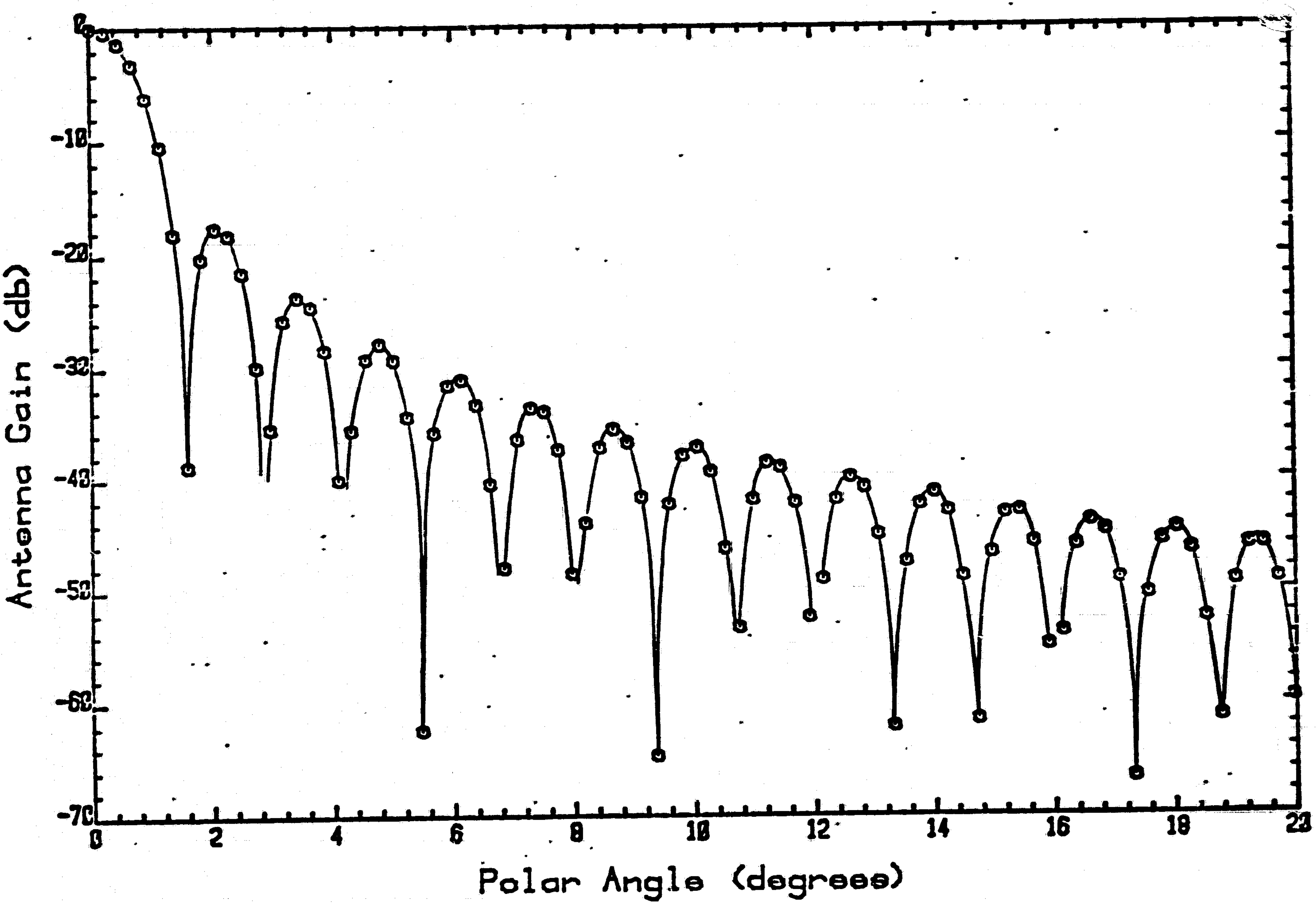


Figure 8. Radiation pattern for a uniform, cophase aperture by the three-dimensional DFT

and 10(a) present radiation patterns in the $\phi = 90^\circ$ plane for a focal-point feed and an offset feed ($y_\varepsilon = -1.25\lambda$). The phase results are given in Figures 9(b) and 10(b). These results may be compared with Figures 3 and 4 of Galindo-Israel and Mittra [1] where amplitude and phase patterns were computed for similar antenna parameters using the Jacobi polynomial method. Careful examination reveals strong agreement between the two methods.

A more rigorous test of the algorithm is to translate the feed 1.25λ along a line bisecting the -x and -y axes and compute a $\phi = 45^\circ$ cut through the pattern. In this case, the feed displacement parameters are $x_\varepsilon = y_\varepsilon = -0.883883\lambda$ and $z_\varepsilon = 0$. The resulting pattern should be essentially the same (for $\theta<10^\circ$) as computed for $y_\varepsilon = -1.25\lambda$, $x_\varepsilon = z_\varepsilon = 0$, and $\phi = 90^\circ$. Excellent agreement is clearly seen in Figure 11. Due to polarization, the $\phi = 45^\circ$ results fall increasingly further below the $\phi = 90^\circ$ results as θ becomes larger.

Figure 12 demonstrates the flexibility of this new algorithm. For a feed offset $y_\varepsilon = -1.25\lambda$, and $x_\varepsilon = z_\varepsilon = 0$, pattern cuts are computed for $\phi = 90^\circ$, 60°, 30°, and 0°. Repeated application to other ϕ angles can adequately describe the three-dimensional nature of the radiation pattern.

By experience, the new three-dimensional algorithm was found to be computationally very fast. Timing was done on the Goddard Space Flight Center's IBM 360/91 using the interval timer available in the system library. With a very close $\lambda/8$ sampling, the cpu run time per pattern (for the general case) was approximately 25 seconds. A $\lambda/4$ sampling was found to be sufficient to study the main beam and several side lobes (see Figure 13), and ran in a mere seven seconds. Circularly symmetric current sheets may be treated as a special case and consequently, run times reduced by about 75 percent. For the cpu times cited, most of the time is

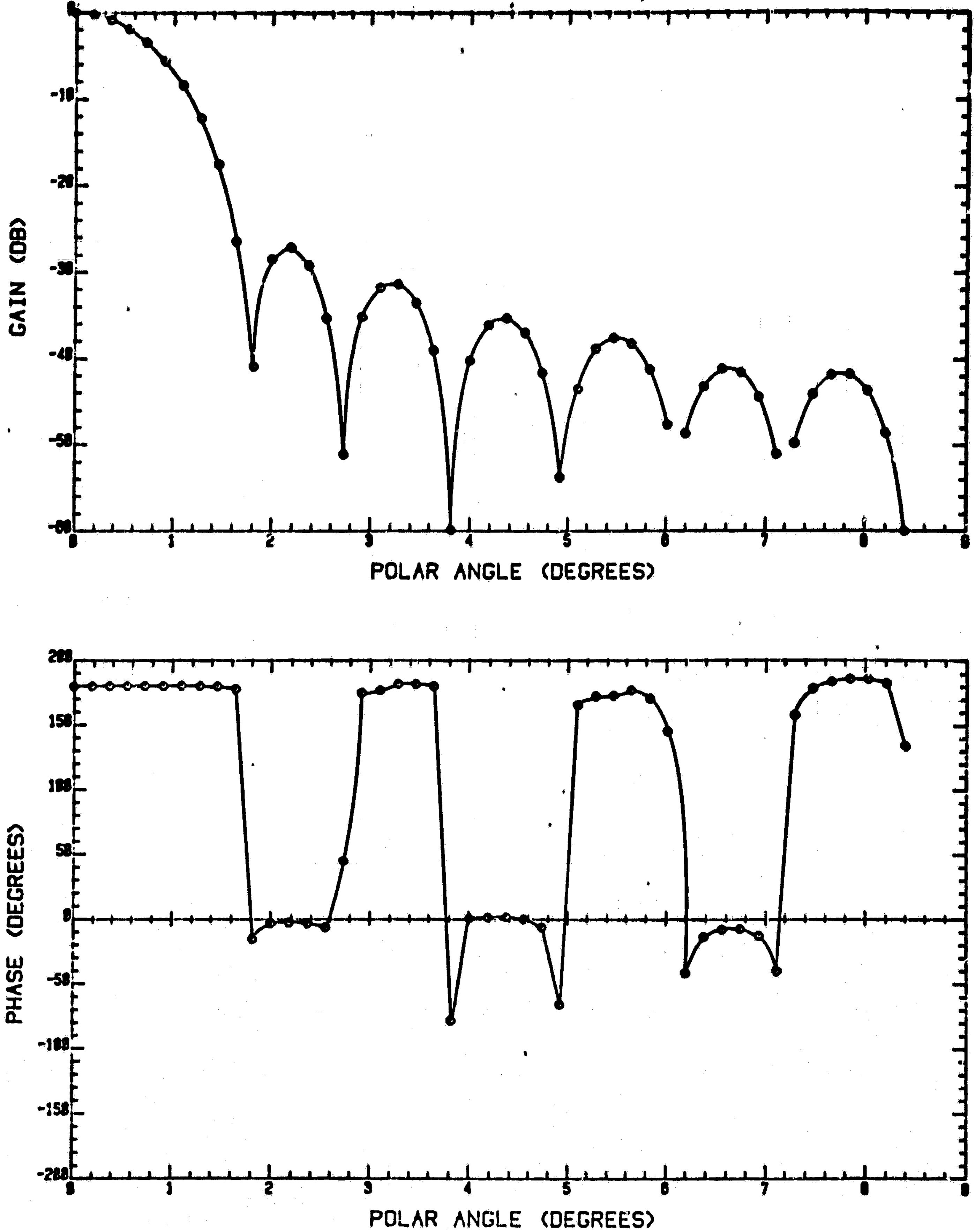


Figure 9. Gain and phase patterns for a focal-point feed. $D/\lambda = 50$, $f/D = 0.5$, $T = \lambda/8$ and $N = 2520$.

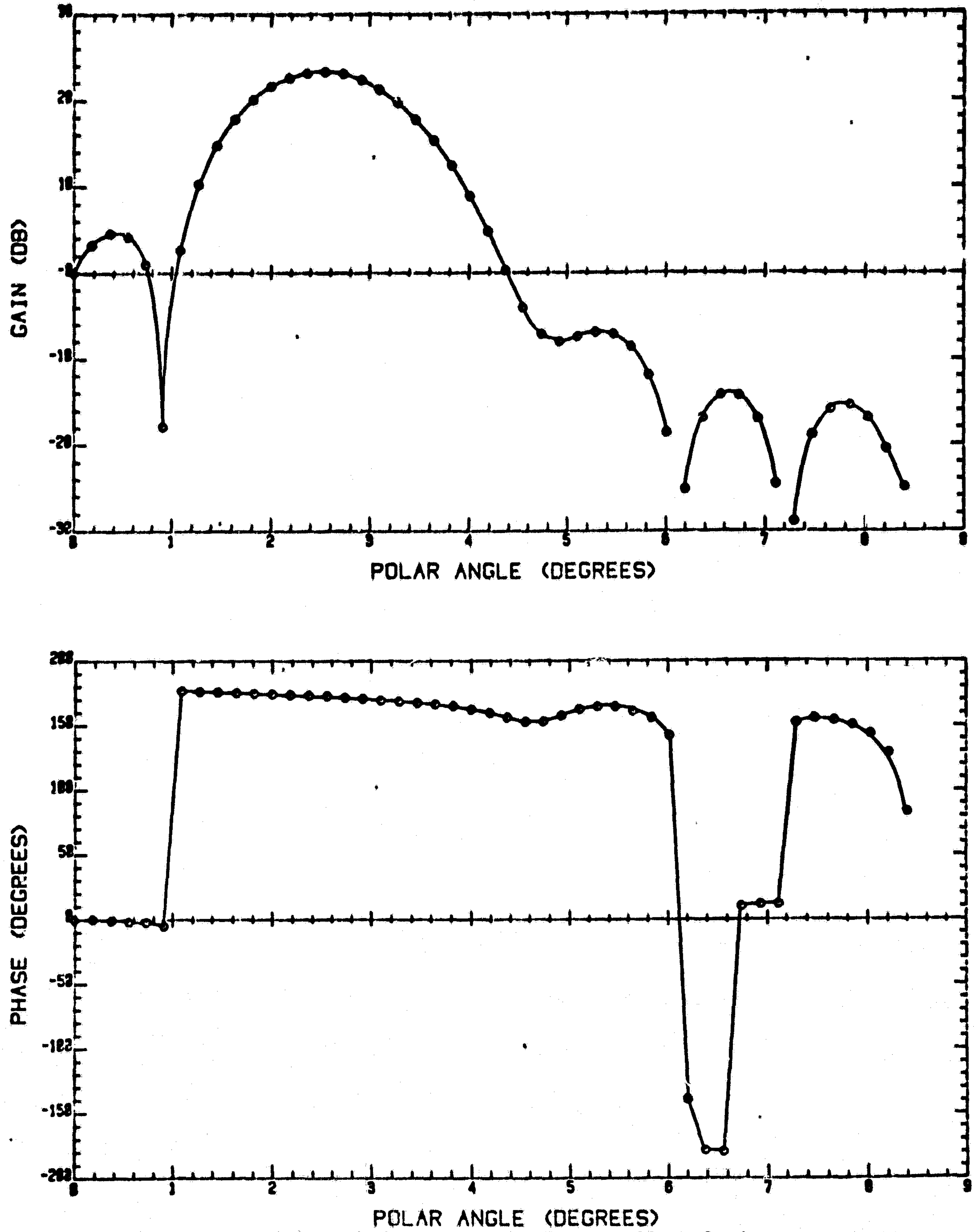


Figure 10. Gain and phase patterns for an offset feed. $y_c = -1.25\lambda$, $x_c = z_c = 0$, $D/\lambda = 50$, $f/D = 0.5$, $T = \lambda/8$ and $N = 2520$.

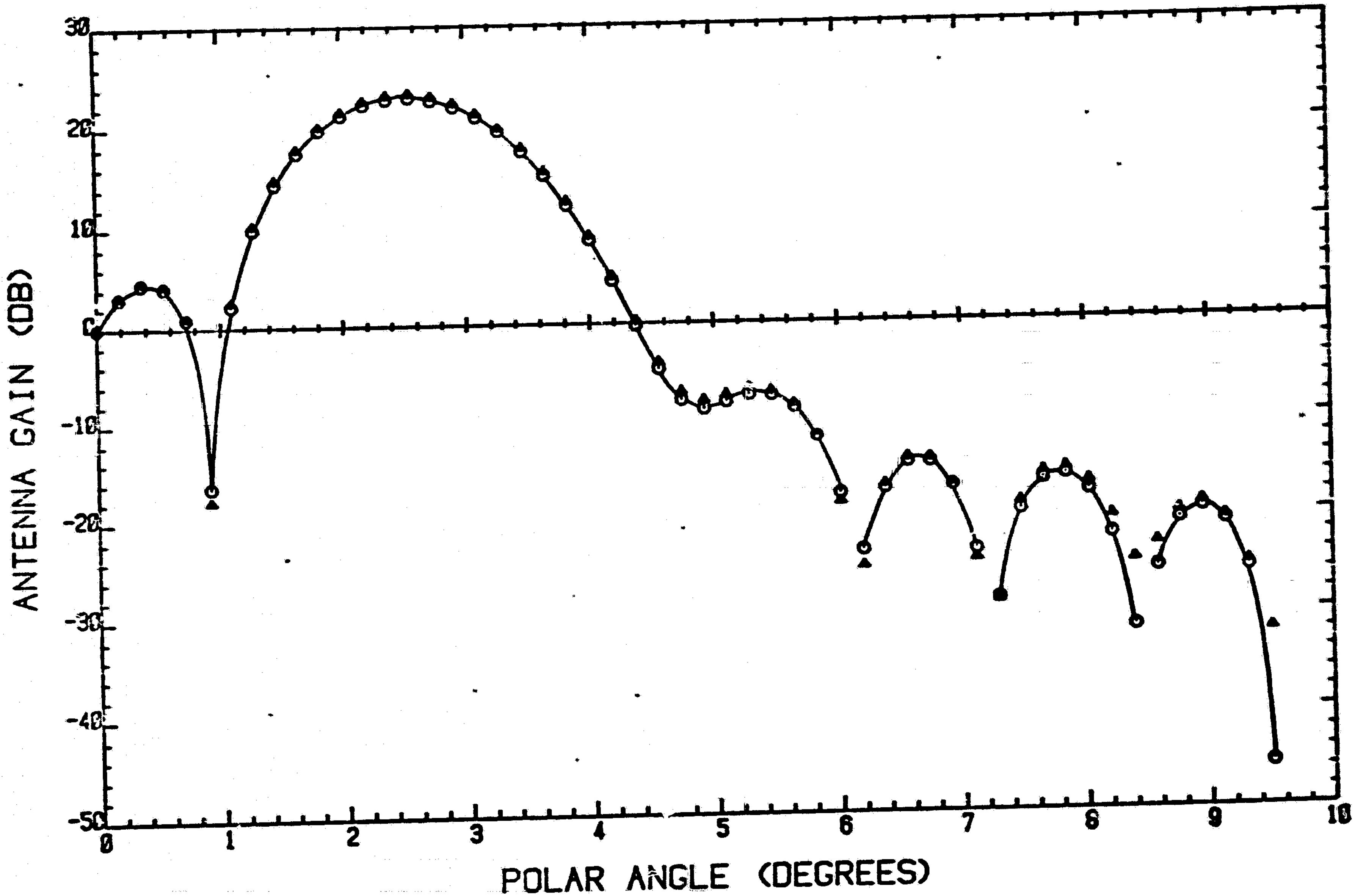


Figure 11. Diagonal axis radiation pattern for a diagonally offset feed. ⊙—⊙ $x_\varepsilon = y_\varepsilon = -0.883883\lambda$, $z_\varepsilon = 0$ and $\phi = 45^0$. △ $y_\varepsilon = -1.25\lambda$, $x_\varepsilon = z_\varepsilon = 0$, and $\phi = 90^0$.

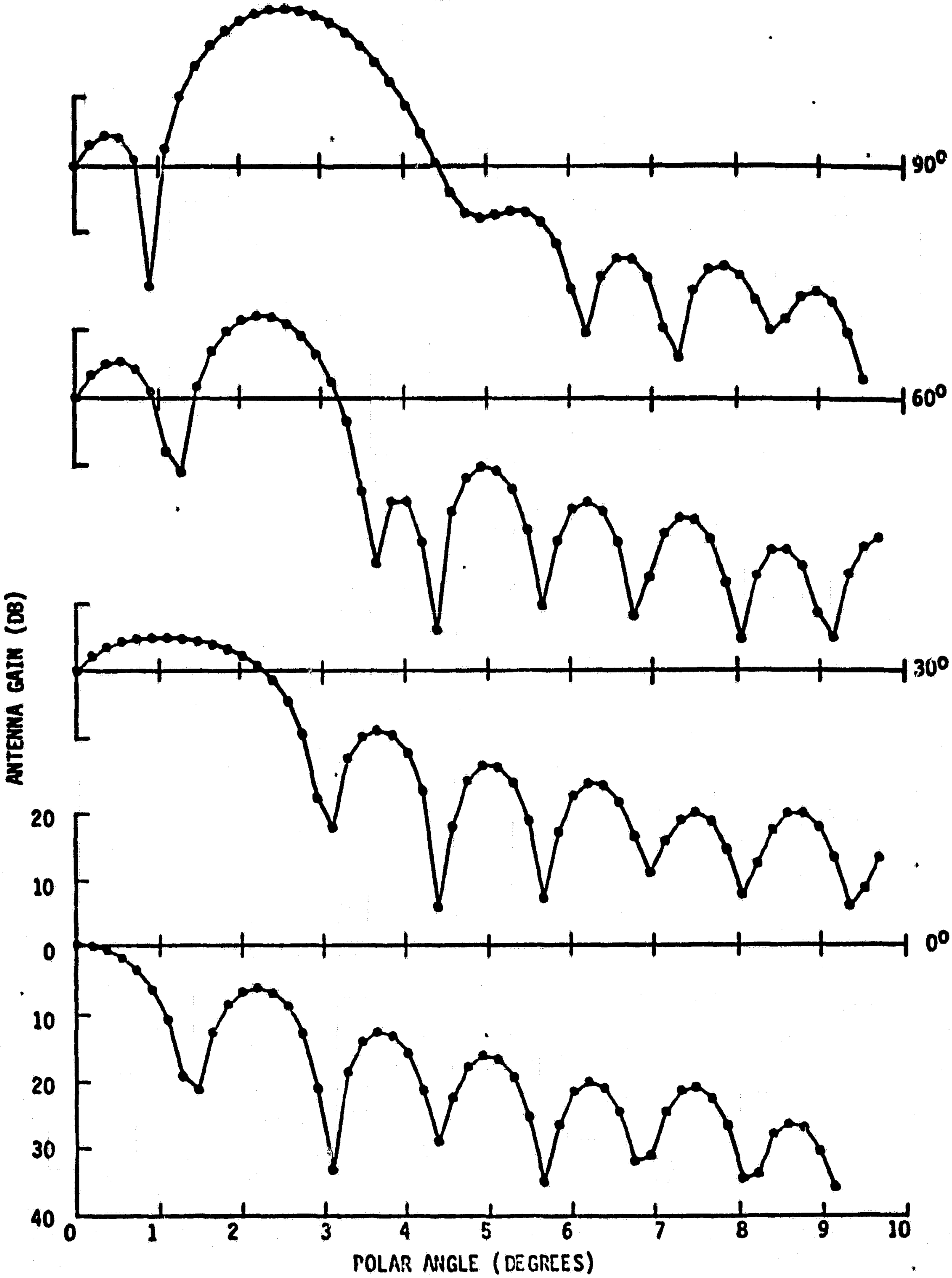


Figure 12. Selected sectional cuts through the radiation pattern. $y_\varepsilon = -1.25\lambda$, $x_\varepsilon = z_\varepsilon = 0$, $D/\lambda = 50$, $f/D = 0.5$, $T = \lambda/8$ and $N = 2520$.

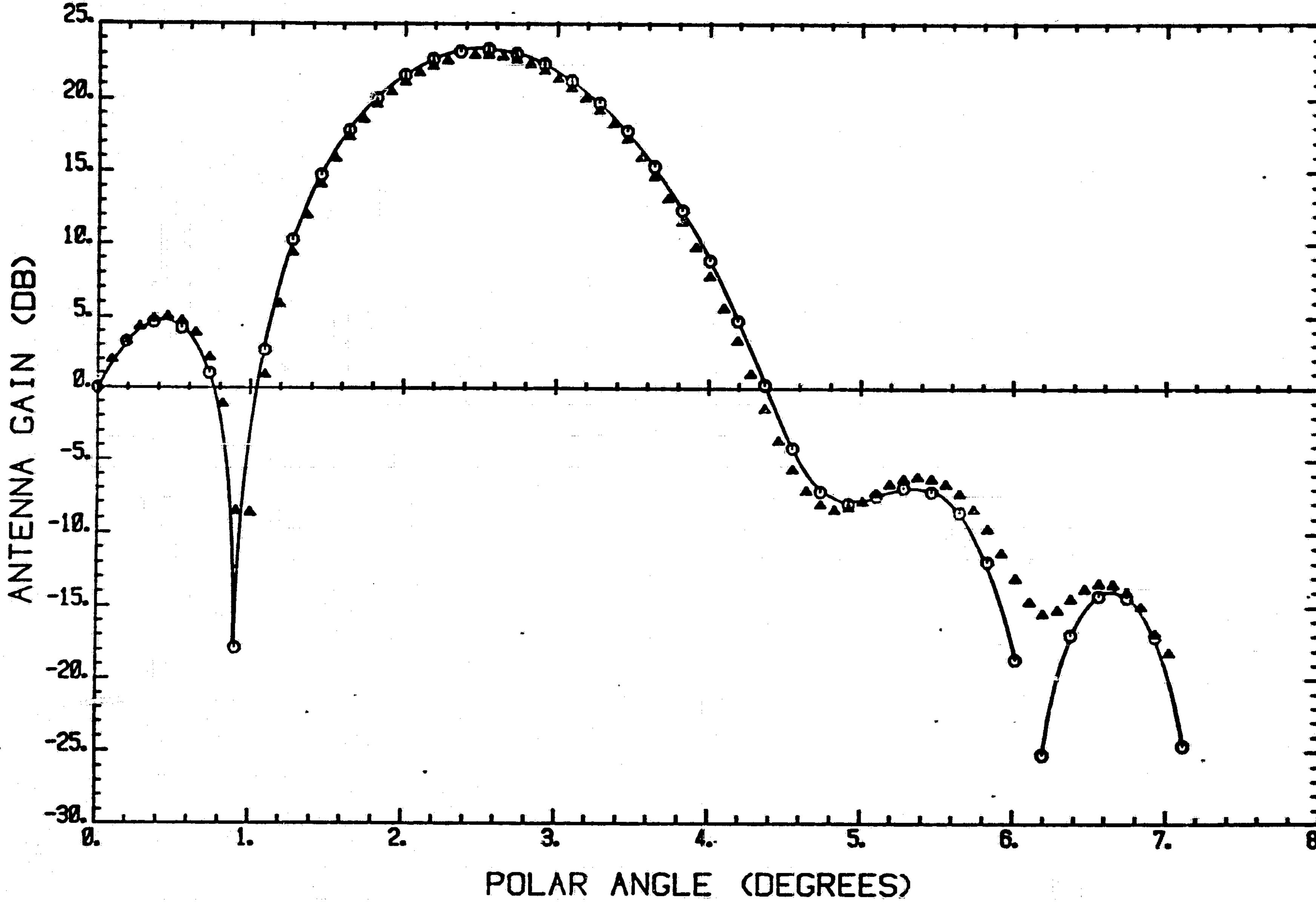


Figure 13. Comparison of λ/8 and λ/4 sampling for an offset feed problem.
⊙—⊙ T = λ/8; △ T = λ/4.

consumed in computing the induced currents. The partial three-dimensional DFT results are computed in much less than half of these times.

VII. CONCLUSIONS

A method of evaluating the radiation integral on the curved surface of a reflecting antenna has been presented. The result is a two-dimensional radiation cross-section along a planer cut at any angle ϕ through the far-field pattern. This section is produced by evaluating the radiation integral via a three-dimensional Fourier transform. A unique feature of this method is a new algorithm for evaluating a subset of the total three-dimensional DFT results. The algorithm is extremely fast so that the computer time required to produce a radiation pattern is primarily determined by the computation of the antenna currents.

High quality gain and phase results have been computed for a paraboloid reflector with translate feed. However, the method is easily extended to offset antenna systems and reflectors of arbitrary shapes. The new method provides a direct but fast approach to the analysis of large asymmetric reflector antennas.

ACKNOWLEDGMENT

This work was supported by NASA Contract No. NAS5-25994. The author wishes to thank R. F. Schmidt of the Goddard Space Flight Center for many helpful discussions.

REFERENCES

1. V. Galindo-Israel and R. Mittra, "A New Series Representation for the Radiation Integral with Application to Reflector Antennas," IEEE Trans. Antennas Propagation, Vol. AP-25, No. 5, September 1977.

2. R. Mittra, Y. Rahmat-Samii, V. Galindo-Israel, and R. Norman, "An Efficient Technique for the Computation of Vector Secondary Patterns of Offset Paraboloid Reflectors," IEEE Trans. Antennas Propagation, Vol. AP-27, No. 3, May 1979.

3. O. M. Bucci, G. Franceschetti, and G. D'Elia, "Fast Analysis of Large Antennas--A New Computational Philosophy," IEEE Trans. Antennas Propagation, Vol. AP-28, No. 3, May 1980.

4. Y. Rahmat-Samii and V. Galindo-Israel, "Shaped Reflector Antenna Analysis using the Jacobi-Bessel Series," IEEE Trans. Antennas Propagation, Vol. AP-28, No. 4, July 1980.

5. W. Swindall and H. H. Barrett, "Computerized Tomography: Taking Sectional X-Rays," Physics Today, December 1977.

6. R. N. Bracewell, The Fourier Transform and Its Applications, New York:McGraw-Hill, 1978, p. 251, 252.

7. R. Heisler, "Far-Field Radiation Patterns of Aperture Antennas by the Winograd Fourier Transform Algorithm," Goddard Space Flight Center Report No. 159911, October 1978.

8. P. J. Emmerman, R. Goulard, R. J. Santoro, and H. G. Semerjian, "Multiangular Absorption Diagnostics of a Turbulent Argon-Methane Jet," J. Energy, Vol. 4, No. 2, March-April 1980.

```
C   THIS PROGRAM COMPUTES THE SURFACE CURRENTS ON A PARABOLIC
C   REFLECTING ANTENNA AS PRODUCED BY AN OFFSET COSINE TO THE NTH
C   FEED AND THEN THE FAR-FIELD PATTERN BY MEANS OF A 3-D DFT.
C   THE INPUTS ARE:
C   D=DIAMETER IN WAVELENGTHS
C   FD=F/D RATIO
C   ANLD= ANGLE (IN DEG) OF THE CUT THROUGH THE PATTERN
C   XE = X FEED OFFSET IN WAVELENGTHS
C   YE = Y FEED OFFSET IN WAVELENGTHS
C   ZE = Z FEED OFFSET IN WAVELENGTHS
C   K4 IS A PARAMETER THAT SPECIFIES THE THATA LIMIT OF THE PATTERN
      REAL*8 A,B,SUMRR,SUMRI,RATIO,C12
      DIMENSION LIMITX(5040)
      COMMON/DATA/SUMRR(5040),SUMRI(5040)
      COMMON/PAR1/M1,M2,M3,M4,NFT,KOUT
      COMMON/PAR2/ANL,K3,C12
      COMMON/PAR3/I,II,X,Y,XE,YE,ZE,Y2,SIGMA2,F,KLIMIT,PI,
     1PHASE3(100),ICOUNT(5040)
      COMMON/DFT/A(5040),B(5040)
      COMMON N,T
      CALL RSTIOO(ICPU)
      READ(5,100) D,FD,ANLD,XE,YE,ZE,K4
  100 FORMAT(6F10.3,I4)
      T=0.125
      XE=XE/T
      YE=YE/T
      ZE=ZE/T
      N1=D/T+1
      K3=315
      WRITE(6,150)
  150 FORMAT(1H1,15X,25HFAR FIELD ANTENNA PATTERN/7X,5HANGLE,4X,
     27HMAG(DB),5X,5HPHASE)
    1 N=2520
      M1=7
      M2=9
      M3=5
      M4=8
      NFT=4
      KOUT=1459
      RAD=1.745329E-2
      PI=3.141597
      ANL=ANLD*RAD
      F=(N1-1)*FD
      KLIMIT=INT((N1-1)/(16.0*FD)+0.5)+1
      DO 3 K=1,KLIMIT
      PHASE3(K)=FLOAT(K-1)*FLOAT(K3)/N
    3 CONTINUE
      DO 5 I=1,N
      SUMRR(I)=0.0D0
      SUMRI(I)=0.0D0
      ICOUNT(I)=0
      LIMITX(I)=0
    5 CONTINUE
      LIMTY1=(1+N1)/2
      LIMTY2=N+1-(N1-1)/2
      DO 200 I=1,N
      IF(I.LE.LIMTY1) GO TO 10
      IF(I.GE.LIMTY2) GO TO 50
```

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      I=LIMTY2-1
      GO TO 200
   10 Y=I-1.0
      Y2=Y*Y
      ARG=(LIMTY1-1.0)**2-Y2
      TX1=SQRT(ARG)+1.0
      LIMTX1=TX1+0.5
      LIMTX2=N+2-LIMTX1
      LIMITX(I)=LIMTX1*2-1
C INPUT DATA FOR QUADRANT ONE
      DO 20 J=1,LIMTX1
      X=J-1.0
      IF(J.EQ.1) GAMA=90.0*RAD
      IF(J.GT.1) GAMA=ATAN(Y/X)
      PSI=ANL-GAMA
      SIGMA2=X*X+Y2
      SIGMA=SQRT(SIGMA2)
      PROJAN=SIGMA*COS(PSI)
      II=INT(PROJAN+0.5)+1
      IF(PROJAN.LT.-0.5) II=N+1+INT(PROJAN-0.5)
      CALL KCOMP
   20 CONTINUE
C INPUT DATA FOR QUADRANT TWO
      IF(LIMTX2.GT.N) GO TO 200
      DO 30 J=LIMTX2,N
      X=-(N-J+1.0)
      IF(I.EQ.1) GAMA=90.0*RAD
      IF(I.GT.1) GAMA=ATAN(-X/Y)
      PSI=90.0*RAD-ANL+GAMA
      SIGMA2=X*X+Y2
      SIGMA=SQRT(SIGMA2)
      PROJAN=SIGMA*COS(PSI)
      II=INT(PROJAN+0.5)+1
      IF(PROJAN.LT.-0.5) II=N+1+INT(PROJAN-0.5)
      CALL KCOMP
   30 CONTINUE
      GO TO 200
   50 Y=-(N+1.0-I)
      Y2=Y*Y
      ARG=(LIMTY1-1.0)**2-Y2
      TX1=SQRT(ARG)+1.0
      LIMTX1=TX1+0.5
      LIMTX2=N+2-LIMTX1
      LIMITX(I)=LIMTX1*2-1
C INPUT DATA FOR QUADRANT FOUR
      DO 60 J=1,LIMTX1
      X=J-1.0
      IF(J.EQ.1) GAMA=90.0*RAD
      IF(J.GT.1) GAMA=ATAN(-Y/X)
      PSI=ANL+GAMA
      SIGMA2=X*X+Y2
      SIGMA=SQRT(SIGMA2)
      PROJAN=SIGMA*COS(PSI)
      II=INT(PROJAN+0.5)+1
      IF(PROJAN.LT.-0.5) II=N+INT(PROJAN-0.5)+1
      CALL KCOMP
   60 CONTINUE
C INPUT DATA FOR QUADRANT THREE
      IF(LIMTX2.GT.N) GO TO 200
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 70 J=LIMTX2,N
      X=-(N-J+1.0)
      GAMA=ATAN(Y/X)
      PSI=ANL-GAMA
      SIGMA2=X*X+Y2
      SIGMA=SQRT(SIGMA2)
      PROJAN=SIGMA*COS(PSI)
      II=N-INT(PROJAN+0.5)+1
      IF(PROJAN.LT.0.5) II=INT(-PROJAN+0.5)+1
      CALL KCOMP
   70 CONTINUE
  200 CONTINUE
      DO 250 I=1,N
      IF(ICOUNT(I).EQ.0) GO TO 250
      RATIO=DFLOAT(LIMITX(I))/DFLOAT(ICOUNT(I))
      SUMRR(I)=SUMRR(I)*RATIO
      SUMRI(I)=SUMRI(I)*RATIO
  250 CONTINUE
      CALL GOODFT(N)
      CALL PATOUT
      K3=K3-1
      IF(K3.GE.K4) GO TO 1
      CALL RMT100(ICPU2)
      ICPU3=ICPU-ICPU2
      WRITE(6,160) ICPU3
  160 FORMAT(1H1,5HICPU=,I10)
      STOP
      END
      SUBROUTINE KCOMP
      REAL*8 SUMRR,SUMRI
      COMMON/DATA/SUMRR(5040),SUMRI(5040)
      COMMON/PAR3/I,II,X,Y,XE,YE,ZE,Y2,SIGMA2,F,KLIMIT,PI,
     1PHASE3(100),ICOUNT(5040)
      COMMON N,T
      F2=2.0*F
      K=KLIMIT-INT(SIGMA2/(4.0*F)+0.5)
      XP=X-XE
      YP=Y-YE
      ZP=F-KLIMIT+K+ZE
      XP2=XP*XP
      YP2=YP*YP
      ZP2=ZP*ZP
      RHOP=SQRT(XP2+YP2+ZP2)
      COTHAP=ZP/RHOP
      ARG2=SIGMA2/(F2*F2)
      DENOM=SQRT((1.0+ARG2)*(XP2+ZP2))
      RNUM=ZP+XP*X/F2
      XMAG=RNUM*(COTHAP**2.2638)/(DENOM*RHOP*T)
      XPHASE=-2.0*PI*(PHASE3(K)+RHOP*T)
      XRE=XMAG*COS(XPHASE)
      XIM=XMAG*SIN(XPHASE)
      SUMRR(II)=SUMRR(II)+XRE
      SUMRI(II)=SUMRI(II)+XIM
      ICOUNT(II)=ICOUNT(II)+1
      RETURN
      END
      SUBROUTINE GOODFT(N)
C   THE SUBROUTINE GOODFT COMPUTES A LENGTH N DFT OF THE INPUT DATA WHICH IS
```

```
C   TWO VECTORS, XR THE REAL PART AND XI THE IMAGINARY PART. BOTH XR AND XI
C   LENGTH N VECTORS. THE LENGTH OF THE DFT, N, MUST BE A PRODUCT OF AT MOST
C   FOUR MUTUALLY PRIME FACTORS. THE POSSIBLE FACTORS ARE 2,3,4,5,7,8,9 AND
C   THESE FACTORS ARE M1, M2, M3, AND M4. IF THE FOUR FACTORS ARE NOT ALL US
C   THE UNUSED FACTORS ARE SET EQUAL TO 1. FOR EXAMPLE WITH N=30, WE HAVE
C   M1=5, M2=3, M3=2, AND M4=1. THE FACTORS OF ONE MUST BE THE LAST OF THE
C   THE NUMBER OF NONUNITY FACTORS IS NFT. KOUT IS AN OUTPUT INDEXING CONSTA
C   WHICH IS PRECOMPUTED. KOUT=(K1+K2+K3+K4)MOD N WHERE K1=M2*M3*M4,
C   K2=M1*M3*M4, K3=M1*M2*M4, K4=M1*M2*M3, AND K2=0 IF M2=1, K3=0 IF M3=1,
C   K4=0 IF M4=1. FOR EXAMPLE, N=30, K1=6, K2=10, K3=15, K4=0 AND KOUT=31
C   =1. THE TRANSFORMED RESULTS ARE STORED IN TWO LENGTH N VECTORS, A AND B.
C   CONTAINS THE REAL PART AND B CONTAINS THE IMAGINARY PART OF THE RESULTS.
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/DATA/XR(5040),XI(5040)
      COMMON/DFT/A(5040),B(5040)
      COMMON/PAR1/M1,M2,M3,M4,NFT,KOUT
      DIMENSION UR(16),UI(16),I(16)
      REAL*8  MR1,MR2,MR3,MR4,MR5,MR6,MR7,MR8,MR9,MR10,MR11,MR12,MR13
      REAL*8  MR14,MR15,MR16,MR17,MR18,MR19,MR20,MR21,MR22,MR23,MR24
      REAL*8  MR25,MR26,MR27,MR28,MR29,MR30,MI1,MI2,MI3,MI4,MI5,MI6,MI7
      REAL*8  MI8,MI9,MI10,MI11,MI12,MI13,MI14,MI15,MI16,MI17,MI18,MI19
      REAL*8  MI20,MI21,MI22,MI23,MI24,MI25,MI26,MI27,MI28,MI29,MI30
      NF=NFT
C   ORDER FACTORS FOR TRANSFORMS OF LENGTH M1
      MM1=M1
      MM2=M2
      MM3=M3
      MM4=M4
      GO TO 20
   10 GO TO(12,13,14),NF
C   ORDER FACTORS FOR TRANSFORMS OF LENGTH M2
   12 MM1=M2
      MM2=M1
      MM3=M3
      MM4=M4
      GO TO 20
C   ORDER FACTORS FOR TRANSFORMS OF LENGTH M3
   13 MM1=M3
      MM2=M1
      MM3=M2
      MM4=M4
      GO TO 20
C   ORDER FACTORS FOR TRANSFORMS OF LENGTH M4
   14 MM1=M4
      MM2=M1
      MM3=M2
      MM4=M3
C   INDEXING INITIALIZATION FOR THE TRANSFORMS
   20 N2=0
      N3=0
      N4=0
      K1=MM2*MM3*MM4
      K2=MM1*MM3*MM4
      K3=MM1*MM2*MM4
      K4=MM1*MM2*MM3
      I(1)=0
C   INPUT INDEXING ALONG ONE DIMENSION
   21 DO 22 J=2,MM1
      I(J)=I(J-1)+K1
      IF(I(J).LT.N) GO TO 22
      I(J)=I(J)-N
   22 CONTINUE
C   TRANSFERRING DATA TO TEMPORARY VECTORS UR AND UI
```

```
   30 DO 31 J=1,MM1
      IJ=I(J)+1
      UR(J)=XR(IJ)
   31 UI(J)=XI(IJ)
C  TRANSFORM UR,UI
      GO TO(50,200,300,400,500,50,700,800,900,50,50,50,50,50,50,1600),MM
     21
C  PLACE RESULTS OF TRANSFORM BACK IN XR AND XI
   40 DO 41 J=1,M11
      IJ=I(J)+1
      XR(IJ)=UR(J)
   41 XI(IJ)=UI(J)
C  TESTING FOR COMPLETION OF THIS FACTOR'S TRANSFORMS
      IF(N2.NE.MM2-1) GO TO 51
      N2=0
      IF(N3.NE.MM3-1) GO TO 52
      N3=0
      IF(N4.NE.MM4-1) GO TO 53
   50 NF=NF-1
      IF(NF.EQ.0) GO TO 1000
      GO TO 10
C  INPUT INDEXING ALONG OTHER DIMENSIONS
   51 N2=N2+1
      DO 54 J=1,MM1
      I(J)=I(J)+K2
      IF(I(J).LT.N) GO TO 54
      I(J)=I(J)-N
   54 CONTINUE
      GO TO 30
   52 N3=N3+1
      I(1)=K3*N3+K4*N4
      IF(I(1).LT.N) GO TO 21
      I(1)=I(1)-N
      GO TO 21
   53 N4=N4+1
      I(1)=K4*N4
      GO TO 21
C  UNSCRAMBLING TRANSFORM RESULTS
 1000 II=1
      J=1
      GO TO 1001
 1002 IF(J.GT.N) GO TO 1003
      II=II+KOUT
 1004 IF(II.LE.N) GO TO 1001
      II=II-N
      GO TO 1004
 1001 A(J)=XR(II)
      B(J)=-XI(II)
      J=J+1
      GO TO 1002
C  2 POINT TRANSFORM
  200 URX=UR(1)+UR(2)
      UIX=UI(1)+UI(2)
      UR(2)=UR(1)-UR(2)
      UI(2)=UI(1)-UI(2)
      UR(1)=URX
      UI(1)=UIX
      GO TO 40
C  3 POINT TRANSFORM
  300 AR=UR(2)+UR(3)
      AI=UI(2)+UI(3)
      MR1=-1.500*AR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      MI1=-1.5D0*AI
      MR2=0.8660254038D0*(UR(2)-UR(3))
      MI2=0.8660254038D0*(UI(2)-UI(3))
      UR(1)=AR+UR(1)
      UI(1)=AI+UI(1)
      MR1=UR(1)+MR1
      MI1=UI(1)+MI1
      UR(2)=MR1-MI2
      UI(2)=MI1+MR2
      UR(3)=MR1+MI2
      UI(3)=MI1-MR2
      GO TO 40
C   4 POINT TRANSFORM
  400 AR1=UR(1)+UR(3)
      AI1=UI(1)+UI(3)
      AR2=UR(1)-UR(3)
      AI2=UI(1)-UI(3)
      AR3=UR(2)+UR(4)
      AI3=UI(2)+UI(4)
      AR4=UR(2)-UR(4)
      AI4=UI(2)-UI(4)
      UR(1)=AR1+AR3
      UI(1)=AI1+AI3
      UR(2)=AR2-AI4
      UI(2)=AI2+AR4
      UR(3)=AR1-AR3
      UI(3)=AI1-AI3
      UR(4)=AR2+AI4
      UI(4)=AI2-AR4
      GO TO 40
C   5 POINT TRANSFORM
  500 AR1=UR(2)+UR(5)
      AI1=UI(2)+UI(5)
      AR2=UR(2)-UR(5)
      AI2=UI(2)-UI(5)
      AR3=UR(3)+UR(4)
      AI3=UI(3)+UI(4)
      AR4=UR(3)-UR(4)
      AI4=UI(3)-UI(4)
      AR5=AR1+AR3
      AI5=AI1+AI3
      MR1=0.9510565163D0*(AR2+AR4)
      MI1=0.9510565163D0*(AI2+AI4)
      MR2=1.538841769D0*AR2
      MI2=1.538841769D0*AI2
      MR3=0.3632712640D0*AR4
      MI3=0.3632712640D0*AI4
      MR4=0.5590169944D0*(AR1-AR3)
      MI4=0.5590169944D0*(AI1-AI3)
      MR5=-1.25D0*AR5
      MI5=-1.25D0*AI5
      UR(1)=UR(1)+AR5
      UI(1)=UI(1)+AI5
      MR5=UR(1)+MR5
      MI5=UI(1)+MI5
      AR1=MR5+MR4
      AI1=MI5+MI4
      AR2=MR5-MR4
      AI2=MI5-MI4
```

```
      AR3=MR1-MR3
      AI3=MI1-MI3
      AR4=MR1-MR2
      AI4=MI1-MI2
      UR(2)=AR1-AI3
      UI(2)=AI1+AR3
      UR(3)=AR2+AI4
      UI(3)=AI2-AR4
      UR(4)=AR2-AI4
      UI(4)=AI2+AR4
      UR(5)=AR1+AI3
      UI(5)=AI1-AR3
      GO TO 40
C   7 POINT TRANSFORM
  700 AR1=UR(2)+UR(7)
      AI1=UI(2)+UI(7)
      AR2=UR(2)-UR(7)
      AI2=UI(2)-UI(7)
      AR3=UR(3)+UR(6)
      AI3=UI(3)+UI(6)
      AR4=UR(3)-UR(6)
      AI4=UI(3)-UI(6)
      AR5=UR(4)+UR(5)
      AI5=UI(4)+UI(5)
      AR6=UR(4)-UR(5)
      AI6=UI(4)-UI(5)
      AR7=AR1+AR3+AR5
      AI7=AI1+AI3+AI5
      MR1=-1.1666666667D0*AR7
      MI1=-1.1666666667D0*AI7
      MR2=0.7901564688D0*(AR1-AR5)
      MI2=0.7901564688D0*(AI1-AI5)
      MR3=0.055854268D0*(AR5-AR3)
      MI3=0.055854268D0*(AI5-AI3)
      MR4=0.734302201D0*(AR3-AR1)
      MI4=0.734302201D0*(AI3-AI1)
      MR5=0.440958552D0*(AR2+AR4-AR6)
      MI5=0.440958552D0*(AI2+AI4-AI6)
      MR6=0.340872931D0*(AR2+AR6)
      MI6=0.340872931D0*(AI2+AI6)
      MR7=-0.533969361D0*(-AR6-AR4)
      MI7=-0.533969361D0*(-AI6-AI4)
      MR8=0.874842291D0*(AR4-AR2)
      MI8=0.874842291D0*(AI4-AI2)
      UR(1)=UR(1)+AR7
      UI(1)=UI(1)+AI7
      AR1=UR(1)+MR1
      AI1=UI(1)+MI1
      AR2=AR1+MR2+MR3
      AI2=AI1+MI2+MI3
      AR3=AR1-MR2-MR4
      AI3=AI1-MI2-MI4
      AR4=AR1-MR3+MR4
      AI4=AI1-MI3+MI4
      AR5=MR5+MR6+MR7
      AI5=MI5+MI6+MI7
      AR6=MR5-MR6-MR8
      AI6=MI5-MI6-MI8
```

```
      AR7=MR5-MR7+MR8
      AI7=MI5-MI7+MI8
      UR(2)=AR2-AI5
      UI(2)=AI2+AR5
      UR(3)=AR3-AI6
      UI(3)=AI3+AR6
      UR(4)=AR4+AI7
      UI(4)=AI4-AR7
      UR(5)=AR4-AI7
      UI(5)=AI4+AR7
      UR(6)=AR3+AI6
      UI(6)=AI3-AR6
      UR(7)=AR2+AI5
      UI(7)=AI2-AR5
      GO TO 40
C  8 POINT TRANSFORM
  800 AR1=UR(2)-UR(8)
      AI1=UI(2)-UI(8)
      AR2=UR(2)+UR(8)
      AI2=UI(2)+UI(8)
      AR3=UR(4)-UR(6)
      AI3=UI(4)-UI(6)
      AR4=UR(4)+UR(6)
      AI4=UI(4)+UI(6)
      AR5=UR(1)-UR(5)
      AI5=UI(1)-UI(5)
      AR6=UR(1)+UR(5)
      AI6=UI(1)+UI(5)
      AR7=UR(3)-UR(7)
      AI7=UI(3)-UI(7)
      AR8=UR(3)+UR(7)
      AI8=UI(3)+UI(7)
      MR1=0.7071067812D0*(AR1+AR3)
      MI1=0.7071067812D0*(AI1+AI3)
      MR2=0.7071067812D0*(AR2-AR4)
      MI2=0.7071067812D0*(AI2-AI4)
      MR3=AR2+AR4
      MI3=AI2+AI4
      MR4=AR6+AR8
      MI4=AI6+AI8
      MR5=AR6-AR8
      MI5=AI6-AI8
      MR6=AR1-AR3
      MI6=AI1-AI3
      MR7=AR5+MR2
      MI7=AI5+MI2
      MR8=AR5-MR2
      MI8=AI5-MI2
      MR9=AR7+MR1
      MI9=AI7+MI1
      MR10=AR7-MR1
      MI10=AI7-MI1
      UR(1)=MR4+MR3
      UI(1)=MI4+MI3
      UR(2)=MR7-MI9
      UI(2)=MI7+MR9
      UR(3)=MR5-MI6
      UI(3)=MI5+MR6
      UR(4)=MR8+MI10
```

```
      UI(4)=MI8-MR10
      UR(5)=MR4-MR3
      UI(5)=MI4-MI3
      UR(6)=MR8-MI10
      UI(6)=MI8+MR10
      UR(7)=MR5+MI6
      UI(7)=MI5-MR6
      UR(8)=MR7+MI9
      UI(8)=MI7-MR9
      GO TO 40
C   9 POINT TRANSFORM
  900 AR1=UR(2)+UR(9)
      AI1=UI(2)+UI(9)
      AR2=UR(2)-UR(9)
      AI2=UI(2)-UI(9)
      AR3=UR(3)+UR(8)
      AI3=UI(3)+UI(8)
      AR4=UR(3)-UR(8)
      AI4=UI(3)-UI(8)
      AR5=UR(5)+UR(6)
      AI5=UI(5)+UI(6)
      AR6=UR(5)-UR(6)
      AI6=UI(5)-UI(6)
      AR7=UR(4)+UR(7)
      AI7=UI(4)+UI(7)
      AR8=UR(4)-UR(7)
      AI8=UI(4)-UI(7)
      AR=AR1+AR3+AR5
      AI=AI1+AI3+AI5
      MR1=-0.5D0*AR7
      MI1=-0.5D0*AI7
      MR2=0.8660254038D0*AR8
      MI2=0.8660254038D0*AI8
      MR3=0.19746542D0*(-AR1+AR5)
      MI3=0.19746542D0*(-AI1+AI5)
      MR4=0.56857902D0*(AR1-AR3)
      MI4=0.56857902D0*(AI1-AI3)
      MR5=0.3711136D0*(-AR3+AR5)
      MI5=0.3711136D0*(-AI3+AI5)
      MR6=0.54253179D0*(AR2-AR6)
      MI6=0.54253179D0*(AI2-AI6)
      MR7=0.10025579D0*(AR2+AR4)
      MI7=0.10025579D0*(AI2+AI4)
      MR8=0.44227597D0*(-AR4-AR6)
      MI8=0.44227597D0*(-AI4-AI6)
      MR9=-1.5D0*AR
      MI9=-1.5D0*AI
      MR10=0.8660254038D0*(AR2-AR4+AR6)
      MI10=0.8660254038D0*(AI2-AI4+AI6)
      AR1=UR(1)+MR1
      AI1=UI(1)+MI1
      UR(1)=AR+AR7+UR(1)
      UI(1)=AI+AI7+UI(1)
      AR=UR(1)+MR9
      AI=UI(1)+MI9
      AR2=MR4-MR5
      AI2=MI4-MI5
      AR3=MR3+MR4
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      AI3=MI3+MI5
      AR6=MR7-MR8
      AI6=MI7-MI8
      AR5=MR6-MR7
      AI5=MI6-MI7
      AR6=AR2-MR5-MR3+AR1
      AI6=AI2-MI5-MI3+AI1
      AR7=AR3+MR3+MR5+AR1
      AI7=AI3+MI3+MI5+AI1
      AR8=-AR3-AR2+AR1
      AI8=-AI3-AI2+AI1
      MR1=MR6-MR8
      MI1=MI6-MI8
      MR3=AR4+MR1+MR2
      MI3=AI4+MI1+MI2
      MR4=AR5+MR1-MR2
      MI4=AI5+MI1-MI2
      MR5=AR5-AR4+MR2
      MI5=AI5-AI4+MI2
      UR(2)=AR6-MI3
      UI(2)=AI6+MR3
      UR(3)=AR7-MI4
      UI(3)=AI7+MR4
      UR(4)=AR-MI10
      UI(4)=AI+MR10
      UR(5)=AR8-MI5
      UI(5)=AI8+MR5
      UR(6)=AR8+MI5
      UI(6)=AI8-MR5
      UR(7)=AR+MI10
      UI(7)=AI-MR10
      UR(8)=AR7+MI4
      UI(8)=AI7-MR4
      UR(9)=AR6+MI3
      UI(9)=AI6-MR3
      GO TO 40
C 16 POINT TRANSFORM
 1600 AR1=UR(1)+UR(9)
      AI1=UI(1)+UI(9)
      AR2=UR(5)+UR(13)
      AI2=UI(5)+UI(13)
      AR3=UR(3)+UR(11)
      AI3=UI(3)+UI(11)
      AR4=UR(3)-UR(11)
      AI4=UI(3)-UI(11)
      AR5=UR(7)+UR(15)
      AI5=UI(7)+UI(15)
      AR6=UR(7)-UR(15)
      AI6=UI(7)-UI(15)
      AR7=UR(2)+UR(10)
      AI7=UI(2)+UI(10)
      AR8=UR(2)-UR(10)
      AI8=UI(2)-UI(10)
      AR9=UR(4)+UR(12)
      AI9=UI(4)+UI(12)
      AR10=UR(4)-UR(12)
      AI10=UI(4)-UI(12)
      AR11=UR(6)+UR(14)
      AI11=UI(6)+UI(14)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
AR12=UR(6)-UR(14)
AI12=UI(6)-UI(14)
AR13=UR(8)+UR(16)
AI13=UI(8)+UI(16)
AR14=UR(8)-UR(16)
AI14=UI(8)-UI(16)
AR15=AR1+AR2
AI15=AI1+AI2
AR16=AR3+AR5
AI16=AI3+AI5
AR17=AR15+AR16
AI17=AI15+AI16
AR18=AR7+AR11
AI18=AI7+AI11
AR19=AR7-AR11
AI19=AI7-AI11
AR20=AR9+AR13
AI20=AI9+AI13
AR21=AR9-AR13
AI21=AI9-AI13
AR22=AR18+AR20
AI22=AI18+AI20
AR23=AR8+AR14
AI23=AI8+AI14
AR24=AR8-AR14
AI24=AI8-AI14
AR25=AR10+AR12
AI25=AI10+AI12
AR26=AR12-AR10
AI26=AI12-AI10
AR31=UR(1)-UR(9)
AI31=UI(1)-UI(9)
UR(1)=AR17+AR22
UI(1)=AI17+AI22
UR(9)=AR17-AR22
UI(9)=AI17-AI22
AR29=AR15-AR16
AI29=AI15-AI16
AR30=AR1-AR2
AI30=AI1-AI2
BR1=0.7071067812D0*(AR19-AR21)
BI1=0.7071067812D0*(AI19-AI21)
BR2=0.7071067812D0*(AR4-AR6)
BI2=0.7071067812D0*(AI4-AI6)
BR3=0.3826834324D0*(AR24+AR26)
BI3=0.3826834324D0*(AI24+AI26)
BR4=1.306562965D0*AR24
BI4=1.306562965D0*AI24
BR5=-0.5411961001D0*AR26
BI5=-0.5411961001D0*AI26
AR32=AR18-AR20
AI32=-AI18+AI20
AR33=AR3-AR5
AI33=-AI3+AI5
AR34=UR(5)-UR(13)
AI34=-UI(5)+UI(13)
BR6=0.7071067812D0*(AR19+AR21)
BI6=-0.7071067812D0*(AI19+AI21)
```

```
MR7=0.7071067812D0*(AR4+AR6)
MI7=-0.7071067812D0*(AI4+AI6)
MR8=0.9238795325D0*(AR23+AR25)
MI8=-0.9238795325D0*(AI23+AI25)
MR9=-0.5411961001D0*AR23
MI9=0.5411961001D0*AI23
MR10=1.306562965D0*AR25
MI10=-1.306562965D0*AI25
MR11=AR30+MR1
MI11=AI20+MI1
MR12=AR30-MR1
MI12=AI20-MI1
MR13=AR33+MR6
MI13=AI33+MI6
MR14=MR6-AR33
MI14=MI6-AI33
MR15=AR31+MR2
MI15=AI31+MI2
MR16=AR31-MR2
MI16=AI31-MI2
MR17=MR4-MR3
MI17=MI4-MI3
MR18=MR5-MR3
MI18=MI5-MI3
MR19=MR15+MR17
MI19=MI15+MI17
MR20=MR15-MR17
MI20=MI15-MI17
MR21=MR16+MR18
MI21=MI16+MI18
MR22=MR16-MR18
MI22=MI16-MI18
MR23=AR34+MR7
MI23=AI34+MI7
MR24=AR34-MR7
MI24=AI34-MI7
MR25=MR8+MR9
MI25=MI8+MI9
MR26=MR8-MR10
MI26=MI8-MI10
MR27=MR23+MR25
MI27=MI23+MI25
MR28=MR23-MR25
MI28=MI23-MI25
MR29=MR24+MR26
MI29=MI24+MI26
MR30=MR24-MR26
MI30=MI24-MI26
UR(2)=MR19+MI27
UI(2)=MI19+MR27
UR(3)=MR11+MI13
UI(3)=MI11+MR13
UR(4)=MR22-MI30
UI(4)=MI22-MR30
UR(5)=AR29+AI32
UI(5)=AI29+AR32
UR(6)=MR21+MI29
UI(6)=MI21+MR29
UR(7)=MR12+MI14
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      UI(7)=AI12+AR14
      UR(9)=AR20-AI28
      UI(9)=AI20-AR28
      UR(10)=AR20+AI28
      UI(10)=AI20+AR28
      UR(11)=AR12-AI14
      UI(11)=AI12-AR14
      UR(12)=AR21-AI29
      UI(12)=AI21-AR29
      UR(13)=AR29-AI32
      UI(13)=AI29-AR32
      UR(14)=AR22+AI30
      UI(14)=AI22+AR30
      UR(15)=AR11-AI13
      UI(15)=AI11-AR13
      UR(16)=AR19-AI27
      UI(16)=AI19-AR27
      GO TO 40
 1003 RETURN
      END
      SUBROUTINE RATOUT
      REAL*8 C12.C2.A.B.CDB.PHA
      COMMON/DFT/A(5040).B(5040)
      COMMON/PAR2/SOL.K3.C12
      COMMON N.T
      INTEGER START.STOP
      SANG2=SIN(ANG)**2
      DEG=57.29578
      IF(K3.NE.315) GO TO 101
      C12=A(1)**2+B(1)**2
      START=1
      STOP=18
      GO TO 200
  101 IF(K3.NE.314) GO TO 102
      START=19
      STOP=31
      GO TO 200
  102 IF(K3.NE.313) GO TO 103
      START=32
      STOP=40
      GO TO 200
  103 IF(K3.NE.312) GO TO 104
      START=41
      STOP=47
      GO TO 200
  104 IF(K3.NE.311) GO TO 105
      START=48
      STOP=54
      GO TO 200
  105 IF(K3.NE.310) GO TO 80
      START=55
      STOP=59
  200 DO 50 I=START.STOP
      S=(I-1.0)/(N*T)
      PHA=DATAN(B(I)/A(I))*DEG
      IF(B(I).LT.0.0) PHA=PHA+180.0
      IF(S.GE.1.0) GO TO 60
      SZ=1.0-S*S
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      TS=S/SQRT(S2)
      ANG=ATAN(TS)
      ANGD=[illegible]
      C2=B(1)**2+A(1)**2
      C2=C2*(1.0-SA*L2*SIN(ANG)**2)/(COS(ANG)**2)
      CDB=10.0*ALOG10(C2/C12)
      WRITE(6,70) ANGD,CDB,PHA
50    CONTINUE
[illegible]
70    FORMAT(3X,F10.4,F10.4,F10.4)
80    RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY