FLUSH-MOUNTED DIELECTRIC-LOADED AXIAL SLOT ON CIRCULAR CYLINDER

J. H. Richmond

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

TECHNICAL REPORT 2902-17

Grant Number NGL 36-008-138

June 1974

(NASA-CR-138752) FLUSH-MOUNTED DIELECTRIC-LOADED AXIAL SLOT ON CIRCULAR CYLINDER (Ohio State Univ.) 34 p HC $4.75 CSCL 20N
N74-27619
Unclas
G3/07 42480

National Aeronautics and Space Administration
Langley Research Center
Hampton, Va. 23365

NOTICES

FLUSH-MOUNTED DIELECTRIC-LOADED AXIAL SLOT ON CIRCULAR CYLINDER

J. H. Richmond

TECHNICAL REPORT 2902-17

Grant Number NGL 36-008-138

June 1974

National Aeronautics and Space Administration
Langley Research Center
Hampton, Va. 23365

ABSTRACT

This report presents the theory, computer program and numerical results for an axial slot antenna on a circular cylinder.

CONTENTS

I. INTRODUCTION

We consider an axial slot antenna on a perfectly conducting circular cylinder. The cylinder is partially coated with a dielectric layer, and the antenna radiates through this flush-mounted window. The motivation for this study is to determine the effects of a high-temperature dielectric layer on the performance of antennas mounted on a space shuttle.

For an axial slot antenna on a circular cylinder completely coated with a dielectric layer, the admittance and patterns have been investigated by Knop[1], Fante[2], and Croswell, Westrick and Knop[3]. Our analysis has some similarity to that of Billingsley and Sinclair[4] for scattering by circular-sector cylinders.

The following sections define the problem and present the theory, computer programs and some numerical results.

II. THEORY

Consider an axial slot antenna on a perfectly conducting circular cylinder as illustrated in Fig. 1. The inner aperture has radius "a" and

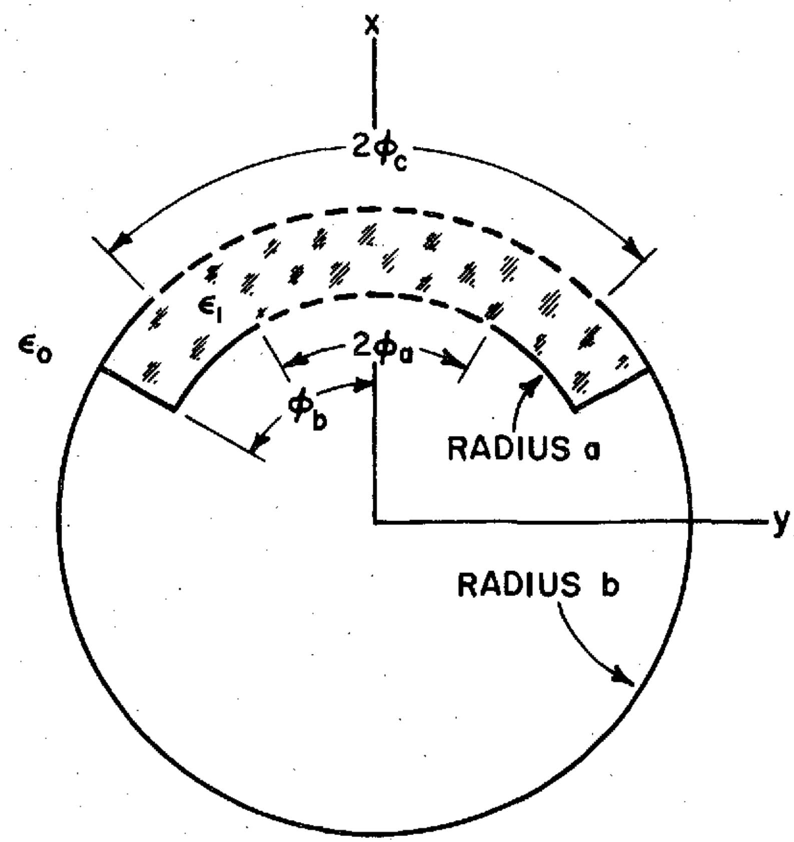

Fig. 1. An axial-slot antenna radiates through a flush-mounted dielectric window in a conducting circular cylinder.

half-angle ϕ_a. The outer aperture has radius b and half-angle ϕ_c. The exterior medium is free space. The inner slot radiates through a flush-mounted homogeneous dielectric window with permittivity ε_1, permeability μ_1, inner radius a, outer radius b and half-angle ϕ_b. The metallic flange prevents the dielectric window from falling out. This cylindrical structure has infinite length, and its axis coincides with the z axis. We consider a time-harmonic excitation with the time dependence $e^{j\omega t}$ understood, and the fields have no z dependence. This report considers the TE polarization in which the non-zero field components are E_ρ, E_ϕ and H_z. Given an even field distribution E_ϕ over the inner aperture, the objective is to determine the aperture admittance, gain and far-field pattern of this antenna. Our solution employs cylindrical-mode expansions and Galerkin's method.

The field in region I (the dielectric window) is

$$E_\rho^I = \frac{j\eta_1}{k_1\rho} \sum_k \nu \, [c_k \, J_\nu(\rho) + d_k \, N_\nu(\rho)] \sin\nu\phi \tag{1}$$

$$E_\phi^I = j\eta_1 \sum_k [c_k \, J_\nu'(\rho) + d_k \, N_\nu'(\rho)] \cos\nu\phi \tag{2}$$

$$H_z^I = \sum_k [c_k \, J_\nu(\rho) + d_k \, N_\nu(\rho)] \cos\nu\phi \tag{3}$$

$$k_1 = \omega\sqrt{\mu_1 \, \varepsilon_1} \tag{4}$$

$$\eta_1 = \sqrt{\mu_1/\varepsilon_1} \tag{5}$$

$$\nu = k\pi/\phi_b \tag{6}$$

where the integer k runs from zero to infinity and (ρ,ϕ,z) are the cylindrical coordinates. (In this report the symbols $J_\nu(\rho)$ and $N_\nu(\rho)$ denote the Bessel and Neumann functions with order ν and argument $k_1\rho$.) This field satisfies the source-free version of Maxwell's equations in region I. From Eqs. (1) and (6), tangential E vanishes at the perfectly conducting surfaces at $\phi = \pm\,\phi_b$. The expansion constants c_k and d_k are to be determined from the boundary conditions.

The voltage across the inner aperture is

(7) $$V = 2a \int_0^{\phi_a} E \, d\phi$$

where E denotes $E_\phi(a,\phi)$. (We assume the aperture field E is a specified even function of ϕ.) The external admittance of the inner aperture is

(8) $$Y = \frac{2a}{VV^*} \int_0^{\phi_a} E^* \, H_z^I(a,\phi) \, d\phi \ .$$

The boundary condition at $\rho = a$ is

(9) $$E_\phi^I = \begin{cases} E & \text{for} \quad 0 < \phi < \phi_a \\ 0 & \text{for} \ \phi_a < \phi < \phi_b \end{cases} \ .$$

From Eqs. (2) and (9) with Fourier analysis,

(10) $$j\eta_1 \ \phi_b \ [c_k \ J_\nu'(a) + d_k \ N_\nu'(a)] = e_k \ G_k$$

(11) $$G_k = \int_0^{\phi_a} E \cos\nu\phi \ d\phi$$

where $e_0 = 1$ and $e_k = 2$ for $k = 1, 2, 3 \cdots$

From Eqs. (3) and (8), the external admittance (per unit length of cylinder) of the inner aperture is

(12) $$Y = \frac{2a}{VV^*} \sum_k [c_k \ J_\nu(a) + d_k \ N_\nu(a)] \ G_k^* \ .$$

The field in region II (the exterior free-space region) is

(13) $$E_\rho^{II} = \frac{j\eta_0}{k_0\rho} \sum_i \ i \ a_i \ H_i(\rho) \sin i\phi$$

(14) $$E_\phi^{II} = j\eta_0 \sum_i \ a_i \ H_i'(\rho) \cos i\phi$$

(15) $$H_z^{II} = \sum_i a_i \, H_i(\rho) \cos i\phi$$

(16) $$k_o = \omega\sqrt{\mu_o \, \varepsilon_o}$$

(17) $$\eta_o = \sqrt{\mu_o/\varepsilon_o}$$

where the integer i runs from zero to infinity. (In this report the symbol $H_i(\rho)$ denotes the Hankel function with order i and argument $k_o\rho$ and the superscript (2) is understood. The argument $k_1\rho$ will not be encountered with the Hankel function.) This field satisfies the radiation conditions and the source-free version of Maxwell's equations.

To complete the solution, it remains only to enforce the boundary conditions at $\rho = b$. The rigorous solution involves an infinite system of simultaneous linear equations. We desire an accurate approximation involving a finite system of simultaneous linear equations. To develop a solution of this type, we expand the field in the outer aperture (at $\rho = b$) as follows:

(18) $$E_\phi = \sum_n b_n \cos(n\pi\phi/\phi_c) \qquad \text{for } 0 < \phi < \phi_c$$

where n runs from zero to N. If the constants b_n were known, the remaining constants (a_i, c_k and d_k) could be determined. In this sense the b_n are independent unknowns, and the others are dependent. When the simultaneous linear equations are written as a matrix equation, the square matrix will be symmetric if the b_n are chosen as the independent quantities.

From Eq. (2) and the boundary condition on E_ϕ^I at $\rho = b$,

(19) $$j\eta_1 \sum_k [c_k \, J_\nu'(b) + d_k \, N_\nu'(b)] \cos\nu\phi = \begin{cases} E_\phi & \text{for } 0 < \phi < \phi_c \\ 0 & \text{for } \phi_c < \phi < \phi_b \end{cases}$$

where E_ϕ is defined by Eq. (18). Multiplying both sides of Eq. (19) by $\cos\nu\phi$ and integrating over the range $0 < \phi < \phi_b$ yields

(20) $$j\eta_1 \, \phi_b \, [c_k \, J_\nu'(b) + d_k \, N_\nu'(b)] = e_k \sum_n b_n \, F_{kn}$$

$$F_{kn} = \int_0^{\phi_c} \cos(k\pi\phi/\phi_b)\, \cos(n\pi\phi/\phi_c)\, d\phi \quad . \tag{21}$$

From Eqs. (10) and (20),

$$c_k = P_k \left[G_k\, N_\nu'(b) - N_\nu'(a) \sum_n b_n F_{kn} \right] \tag{22}$$

$$d_k = P_k \left[-G_k\, J_\nu'(b) + J_\nu'(a) \sum_n b_n F_{kn} \right] \tag{23}$$

$$P_k = \frac{e_k}{j\eta_1\, \phi_b [J_\nu'(a)\, N_\nu'(b) - J_\nu'(b)\, N_\nu'(a)]} \quad . \tag{24}$$

From Eq. (14) and the boundary condition on E_ϕ^{II} at $\rho = b$,

$$j\eta_o \sum_i a_i\, H_i'(b) \cos i\phi = \begin{cases} E_\phi & \text{for} \quad 0 < \phi < \phi_c \\ 0 & \text{for} \quad \phi_c < \phi < \pi \end{cases} \tag{25}$$

where E_ϕ is defined by Eq. (18). Multiplying both sides of Eq. (25) by $\cos i\phi$ and integrating over the range $0 < \phi < \pi$ yields:

$$a_i = \frac{e_i}{j\pi\eta_o H_i'(b)} \sum_n b_n G_{in} \tag{26}$$

$$G_{in} = \int_0^{\phi_c} \cos(i\phi)\, \cos(n\pi\phi/\phi_c)\, d\phi \quad . \tag{27}$$

Equations (22)-(26) show explicitly that a knowledge of the constants b_n is sufficient to determine all the other constants.

At this point we have used the boundary conditions on E_ϕ to relate a_i, c_k and d_k to b_n. The next step is to use the boundary condition on H_z to generate a system of simultaneous linear equations for the constants b_n. From Eqs. (3) and (15) and continuity of tangential H across the outer aperture (at $\rho = b$):

$$(28) \qquad \sum_i a_i H_i(b) \cos i\phi = \sum_k [c_k J_\nu(b) + d_k N_\nu(b)] \cos\nu\phi$$

where ϕ ranges from zero to ϕ_c. In Eq. (28), multiplying both sides by $\cos(m\pi\phi/\phi_c)$ and integrating from $\phi = 0$ to $\phi = \phi_c$ yields

$$(29) \qquad \sum_i a_i H_i(b) G_{im} = \sum_k [c_k J_\nu(b) + d_k N_\nu(b)] F_{km}.$$

In matching H_z across the aperture, we selected the same weighting function $\cos(m\pi\phi/\phi_c)$ also used as a basis function in Eq. (18). This is the distinctive feature of Galerkin's method. If Eqs. (22), (23) and (26) are used to eliminate a_i, c_k and d_k, Eq. (29) yields:

$$(30) \qquad \sum_n Z_{mn} b_n = V_m \quad \text{with } m = 0, 1, 2, \cdots N$$

$$(31) \qquad V_m = \frac{2 j\eta_1 \phi_b}{k_1 b \phi_c} \sum_k P_k G_k F_{km}$$

$$(32) \qquad Z_{mn} = \frac{\phi_b}{\phi_c} \left[\frac{\eta_1}{\eta_0} \sum_i \frac{e_i H_i(b) G_{im} G_{in}}{H_i'(b)} + \frac{\pi}{\phi_b} \sum_k e_k R_k F_{km} F_{kn} \right]$$

$$(33) \qquad R_k = \frac{J_\nu(b) N_\nu'(a) - J_\nu'(a) N_\nu(b)}{J_\nu'(a) N_\nu'(b) - J_\nu'(b) N_\nu'(a)} \quad .$$

Equation (30) is recognized as a system of simultaneous linear equations. In the summation, n runs from zero to N. Equation (30) can also be written as a matrix equation. The symmetry of the square matrix Z_{mn} is obvious in Eq. (32).

The matrix equation is solved with a digital computer to obtain numerical values for b_n. Then Eqs. (22), (23) and (26) are employed to determine c_k, d_k and a_i. The aperture admittance is obtained from Eq. (12). The far-field pattern is obtained from Eq. (15) as follows:

$$(34) \qquad H_z = e^{-jk\rho} \sqrt{\frac{2j}{\pi k\rho}} \sum_i a_i j^i \cos i\phi \ .$$

The power gain is calculated as follows:

$$(35) \qquad G_p(\phi) = \frac{2\pi\rho\eta_0 \; |H_z(\rho,\phi)|^2}{VV^*G}$$

where the aperture voltage V is given by Eq. (7) and the aperture conductance G is the real part of Y in Eq. (12).

III. NUMERICAL RESULTS

Figures 2-6 illustrate the far-field patterns of an axial slot antenna radiating through a lossless window with dielectric constant of 1, 1.2, 2, 3 and 4. Figures 2 and 3 compare the calculated patterns with experimental measurements performed at NASA Langley. Measurements are not available for the other cases. In this sequence of figures all parameters of the slot, window and cylinder are fixed except the dielectric constant. The electric field distribution is uniform across the inner aperture.

All the patterns are reasonably smooth except in Fig. 5 where the pattern breaks up into many lobes with deep nulls. With a dielectric constant of 3, this anomalous type of pattern is observed when the aperture half-angle is ϕ_b = 13.8, 14.8, 15.8, 16.8°, etc. When ϕ_b differs from one of these critical angles by more than 0.1 degrees, the pattern becomes smooth again. At each critical angle, the aperture width is an integral number of wavelengths for the lowest-order surface wave. This surface-wave resonance phenomenon is less pronounced with a dielectric constant of 1.2 but is observed when ϕ_b = 14.6°. The effect may be reduced with a lossy dielectric window or by reducing the reflection coefficient at the edges of the aperture.

In these figures, the calculations are based on a two-dimensional model with an infinitely long axial slot. A case of greater interest is a half-wave axial slot in a long cylinder. The effects of surface-wave resonance will be reduced with a slot of finite length.

In generating the data for Fig. 5, the execution time was 50 seconds on a Datacraft 6024/3 computer. The solution involved a system of 20 simultaneous linear equations (N = 19 in Eqs. (18) and (30)). The infinite series with index i (in Eqs. (32) and (34)) were truncated after 148 terms, and the series with index k (in Eqs. (12), (31) and (32)) were truncated after 20 terms. The calculated aperture admittance was 0.372 + j 0.122 mhos/wavelength. Identical results were obtained with N = 18 and 19, but N = 15 proved inadequate.

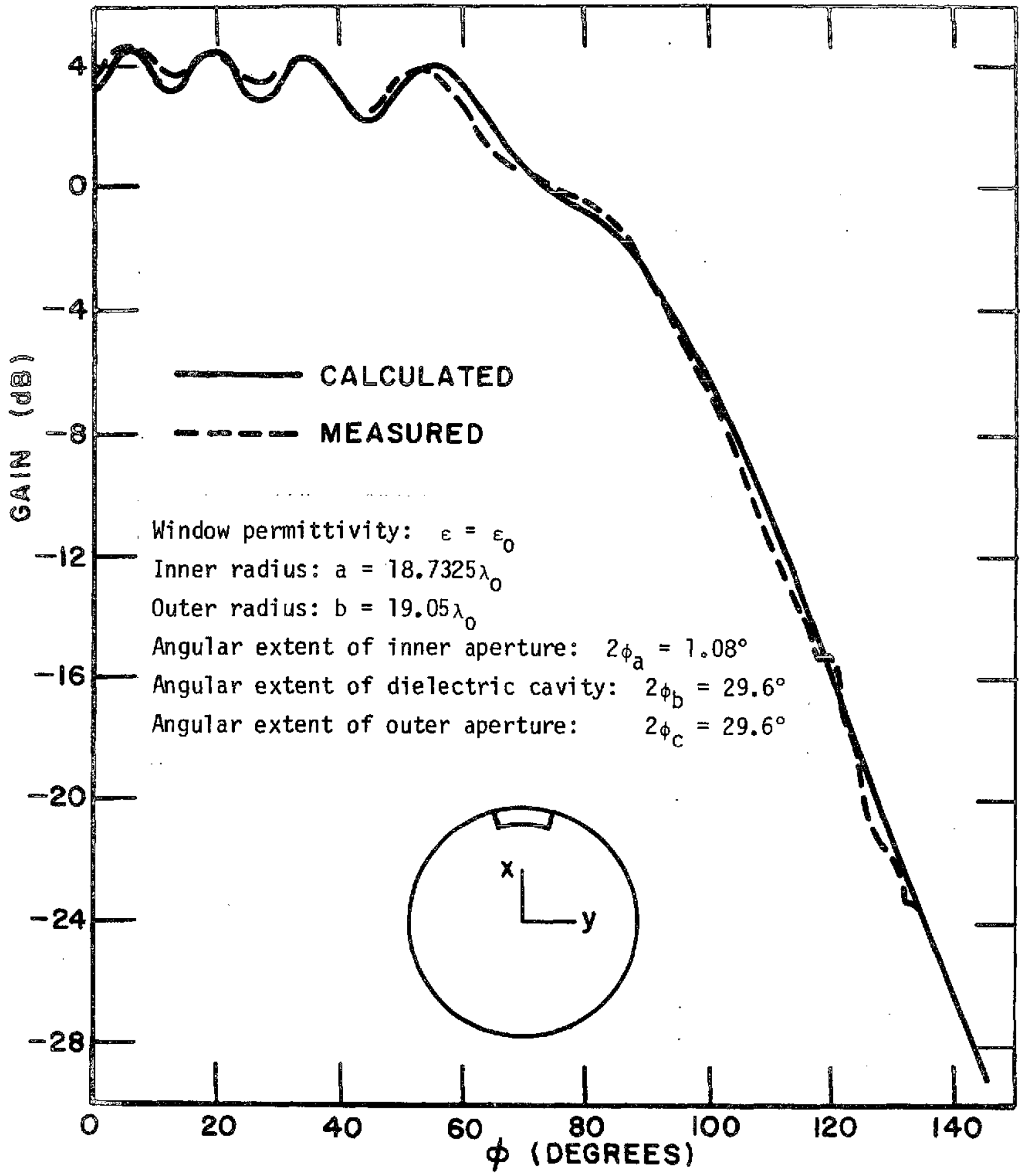

Fig. 2. Far-field pattern with $\varepsilon_r = 1$.

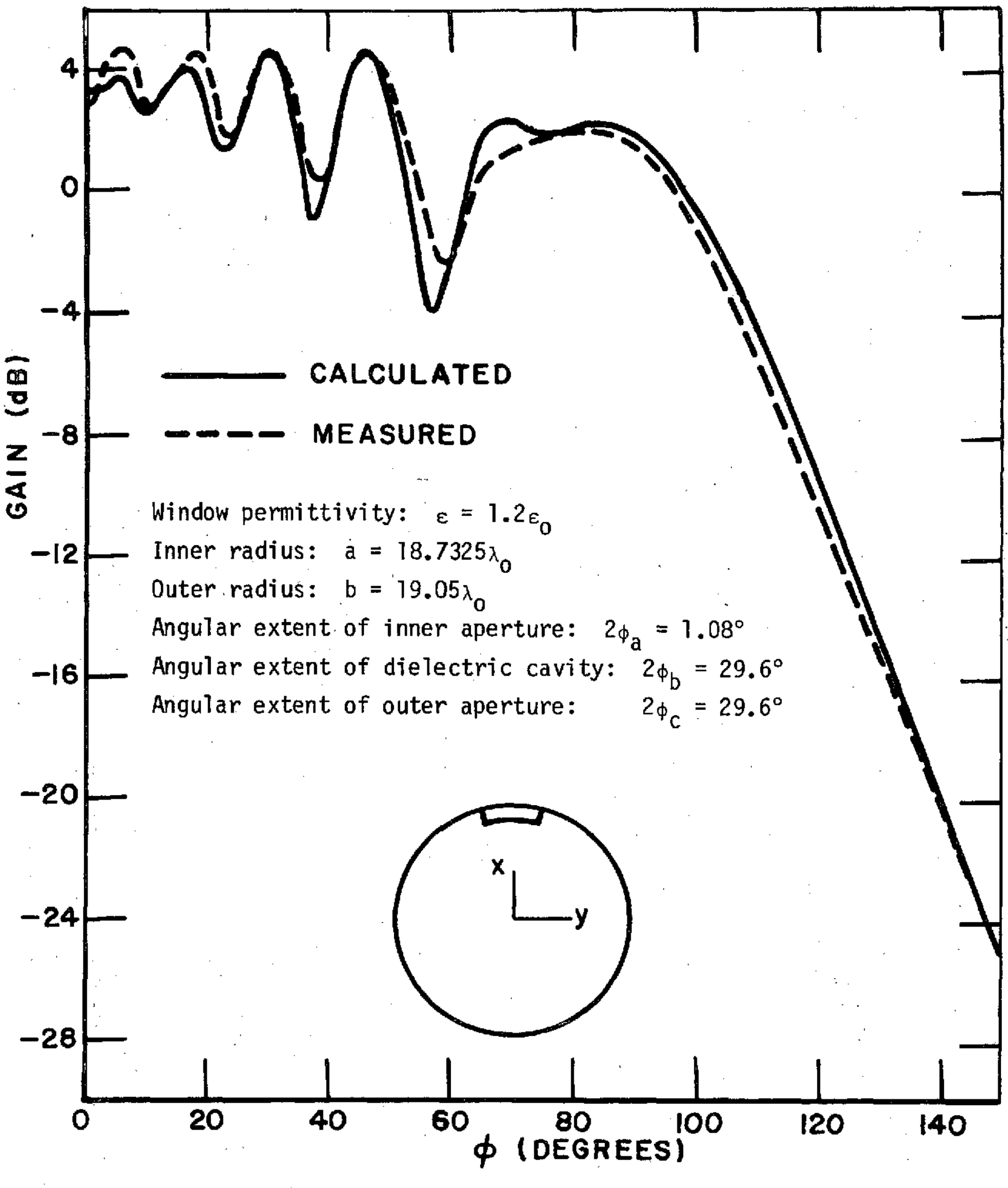

Fig. 3. Far-field pattern with $\varepsilon_r = 1.2$.

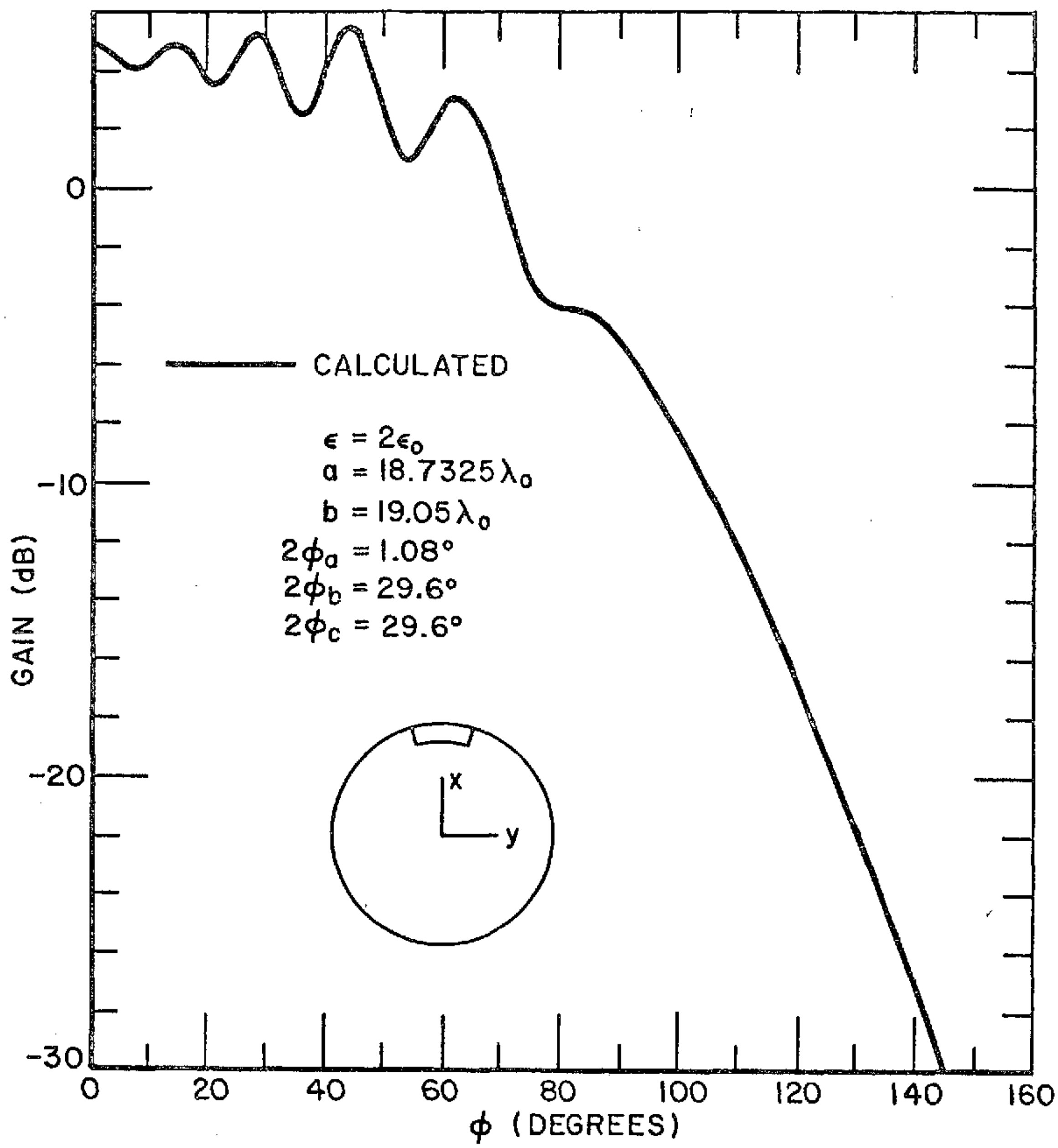

Fig. 4. Far-field pattern with $\varepsilon_r = 2$.

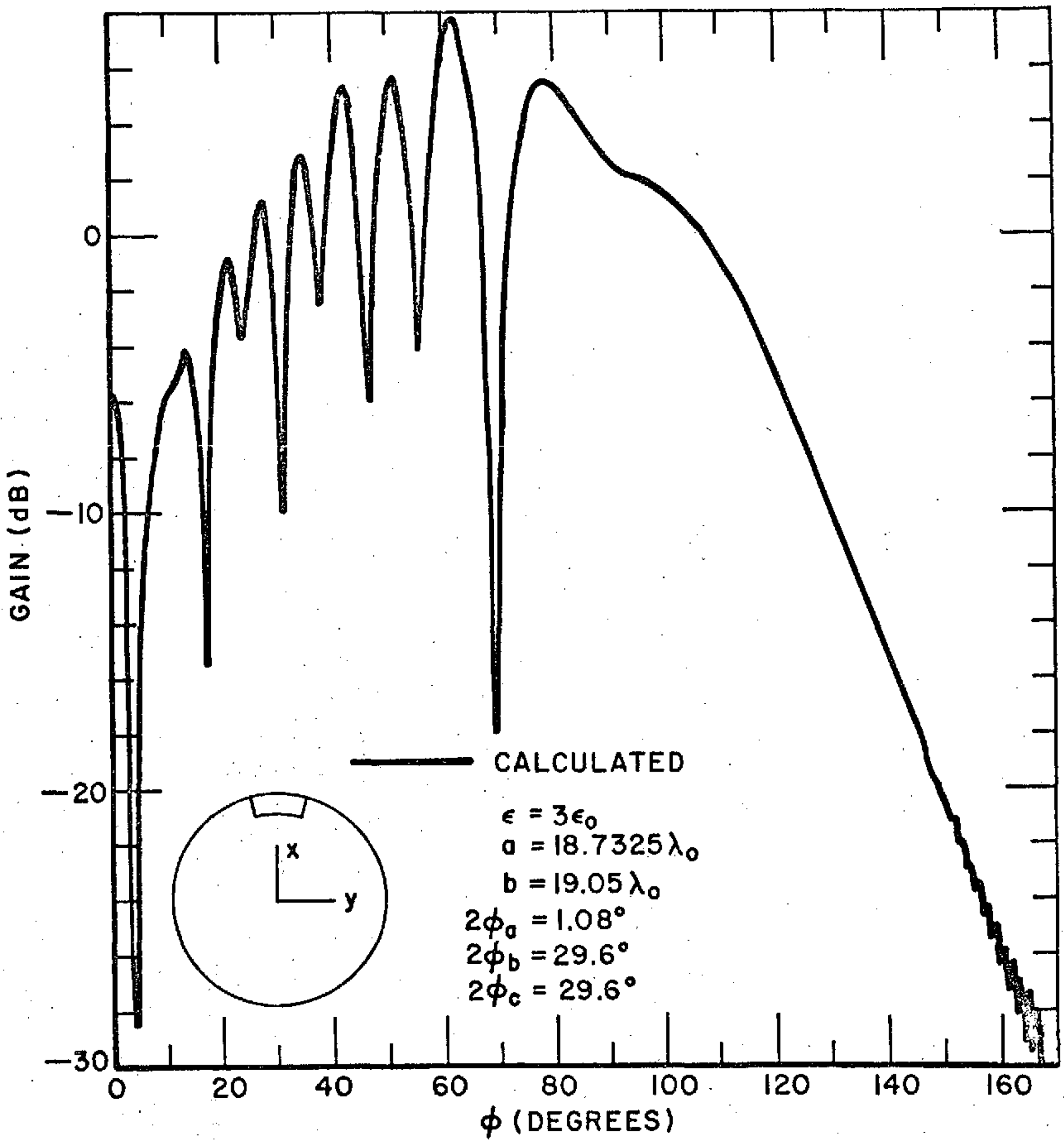

Fig. 5. Far-field pattern with ε_r = 3.

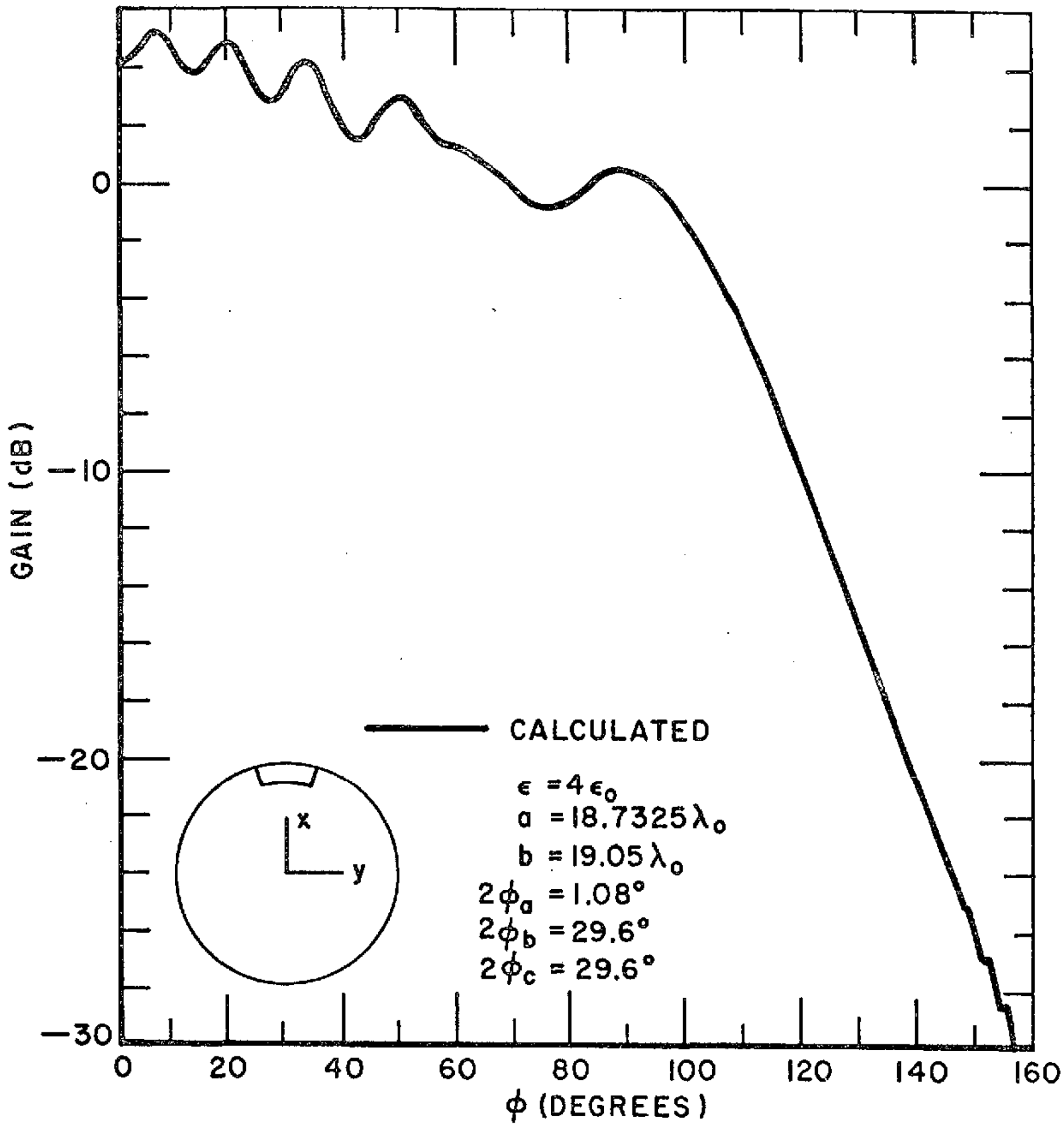

Fig. 6. Far-field pattern with $\varepsilon_r = 4$.

IV. SUMMARY AND CONCLUSIONS

This report develops the theoretical formulation for a TE axial slot antenna radiating through a dielectric window in a circular cylinder. Numerical results are presented for the far-field patterns, and it is noted that the calculations show excellent agreement with experimental measurements. The computer program is presented in the Appendices.

The solution is based on Galerkin's method. Simultaneous linear equations are generated in which the unknown quantities are the coefficients in a Fourier-series expansion for the electric field in the outer aperture. The formulation is rapidly convergent, and the computer program is quite efficient.

REFERENCES

1. Knop, C.M., "External Admittance of an Axial Slot on a Dielectric Coated Metal Cylinder," Radio Science, Vol. 3, No. 8, August 1968, pp. 803-817.

2. Fante, R.L., "Calculation of the Admittance, Isolation, and Radiation Pattern of Slots on an Infinite Cylinder Covered by an Inhomogeneous Lossy Plasma," Radio Science, Vol. 6, No. 3, March 1971, pp. 421-428.

3. Croswell, W.F., Westrick, G.C. and Knop, C.M., "Computations of the Aperture Admittance of an Axial Slot on a Dielectric Coated Cylinder," IEEE Trans., Vol. AP-20, January 1972, pp. 89-92.

4. Billingsley, J.B. and Sinclair, G., "Numerical Solutions to Electromagnetic Scattering from Strips, Finite Wedges, and Notched Circular Cylinders," Canadian Journal of Physics, Vol. 44, 1966, pp. 3217-3225.

5. Goldstein, M. and Thaler, R.M., "Recurrence Techniques for the Calculation of Bessel Functions," MTAC (Mathematics of Computation) Vol. 13, April, 1959, pp. 102-108.

6. Goldstein, M. and Thaler, R.M., "Bessel Functions for Large Arguments," MTAC (Mathematics of Computation) Vol. 12, January 1958, pp. 18-26.

APPENDIX I
THE MAIN COMPUTER PROGRAM

The MAIN computer program is listed in Fig. 7. In this program E_ϕ is uniform across the inner aperture and the dielectric window is lossless. Following the format statements, the dimensions are indicated for the subscripted quantities as follows:

IDC	dimension of B and V
IDH	dimension of A, BHR, BB, YY, BP, YP, SNC and SGA
IDJ	dimension of C, D, SGC, RBES, BEN, AJJ, etc
IDZ	dimension of Z.

The input data are programmed at statement 20 with the following definitions:

AL	inner radius a/λ
BL	outer radius b/λ
ER	dielectric constant ϵ_1/ϵ_0
DPH	angular increment for far-field pattern calculations
PHA	ϕ_a in degrees
PHB	ϕ_b in degrees
PHC	ϕ_c in degrees

where λ denotes the wavelength in free space.

At statement 30, subroutine BESSI is called for the Bessel and Neumann functions and their derivatives. This subroutine also determines the number of terms (denoted by KK) to be employed in the summations on k in Eqs. (12), (31) and (32). The last call to BESSI determines the number of terms (denoted by II) to be employed in the summations on i in Eqs. (32) and (34). If KK is equal to IDJ, the dimension IDJ should be increased. If II is equal to IDH, the dimension IDH should be increased. II should exceed IMIN, and KK should exceed KMIN. NN denotes the number of simultaneous linear equations and the number of terms to be employed in the summations on n in Eqs. (22), (23) and (26). NN should exceed NMIN.

For a uniform aperture distribution with V = 1 volt, Eqs. (7) and (11) yield

$$(36) \qquad G_k = \frac{\sin(\nu\phi_a)}{2a\nu\phi_a} \quad .$$

In the computer program, SGA(K) denotes $2aG_k$. Subroutine GLJ calculates F_{kn}/ϕ_c where F_{kn} is defined by Eq. (21). Subroutine GNJ calculates

G_{in}/ϕ_c where G_{in} is defined by Eq. (27). Some of the symbols used in the program are defined as follows:

A	a_i
B	b_n/k_o
C	c_k
D	d_k
V	V_m/k_o
AK	$k_o a$
BK	$k_o b$
AK1	$k_1 a$
BK1	$k_1 b$
BEN	denominator in Eq. (33)
BHR	$H_i(b)/H_i'(b)$
ETA	η_o
ETA1	η_1
GNU	ν
RBES	$-R_k$
SGC	$(\sin\nu\phi_c)/(\nu\phi_c)$
SNC	$(\sin i\phi_c)/(i\phi_c)$
SJN	first summation in Eq. (32)
SJL	second summation in Eq. (32)
Y11	aperture admittance Y
Z(L)	Z_{mn}

In statement 130, subroutine SQROT is called to solve the system of simultaneous linear equations. Then the expansion coefficients a_i, c_k and d_k are calculated from Eqs. (22), (23) and (26). The aperture admittance is calculated at statement 360 with Eq. (12). Finally, the gain is calculated with Eqs. (34) and (35).

```
C     TE AXIAL SLOT IN PERFECTLY CONDUCTING CIRCULAR CYLINDER.
C     SLOT RADIATES THROUGH DIELECTRIC WINDOW.
C     PROGRAM BY JACK H RICHMOND, OHIO STATE UNIVERSITY.
      COMPLEX V(30),B(30),Z(465),C(150),D(150),A(400),BHR(400)
      COMPLEX CQ,HZ,SUMN,SUL,SJN,Y11
      DIMENSION BB(400),YY(400),BP(400),YP(400),SNC(400),SGA(400)
      DIMENSION SGC(150),RBES(150),BEN(150)
      DIMENSION AJJ(150),AYY(150),AJP(150),AYP(150)
      DIMENSION BJ1(150),BY1(150),BJP1(150),BYP1(150)
      EQUIVALENCE (B,V),(BB,SNC),(YY,SGA),(BJ1,SGC),(BY1,RBES)
      DATA ETA,PI,TP/376.727,3.14159,6.28318/
    2 FORMAT(1X,8F15.6)
    4 FORMAT(1X,12I10)
    5 FORMAT(1H0)
      IDC=30
      IDH=400
      IDJ=150
      IDZ=465
   20 AL=18.7325
      BL=19.05
      ER=3.
      DPH=2.
      PHA=0.54
      PHB=14.8
      PHC=PHB
      IF(PHA.GT.PHB)PHA=PHB
      IF(PHC.GT.PHB)PHC=PHB
      N=.5+(SQRT(1.+8.*IDZ)-1.)/2.
      IF(N.LT.IDC)IDC=N
      NN=IDC
      SQR=SQRT(ER)
      ETA1=ETA/SQR
      TL=BL-AL
      WRITE(6,2)AL,BL,TL,ER,PHA,PHB,PHC
      WRITE(6,5)
      AK=TP*AL
      BK=TP*BL
      AK1=AK*SQR
      BK1=BK*SQR
      PHAR=.0174533*PHA
      PHBR=.0174533*PHB
      PHCR=.0174533*PHC
      GNU=PI/PHBR
   30 CALL BESSI(AK1,GNU,AJJ,AYY,AJP,AYP,IDJ,IDH,KK ,BB,YY)
      CALL BESSI(BK1,GNU,BJ1,BY1,BJP1,BYP1,IDJ,IDH,LL ,BB,YY)
      CALL BESSI(BK,1.,BB,YY,BP,YP,IDH,IDH,II,BB,YY)
      IF(LL.LT.KK)KK=LL
      KMIN=BK*PHB/180.
      IMIN=BK
      NMIN=BK*PHC/180.
      NMAX=.5+KK*PHC/PHB
      IF(NN.GT.IDC)NN=IDC
      IF(NN.GT.NMAX)NN=NMAX
      IF(NN.LT.1)NN=1
      WRITE(6,4)IMIN,II,KMIN,KK,NMIN,NN
      WRITE(6,5)
      SUMN=(.0,.0)
      BHR(1)=CMPLX(BB(1),-YY(1))/CMPLX(BP(1),-YP(1))
      DO 60 I=2,II
      BHR(I)=CMPLX(BB(I),-YY(I))/CMPLX(BP(I),-YP(I))
      N=I-1
      SC=SIN(N*PHCR)/(N*PHCR)
```

Fig. 7. The MAIN computer program.

```
      SNC(I)=SC
60    SUMN=SUMN+BHR(I)*SC*SC
      RUM1=AJP(1)*BY1(1)-BJ1(1)*AYP(1)
      REN1=AJP(1)*BYP1(1)-BJP1(1)*AYP(1)
      SUML=.0
      SUMP=.0
      DO 70 K=2,KK
      BEN(K)=AJP(K)*BYP1(K)-BJP1(K)*AYP(K)
      RBES(K)=(AJP(K)*BY1(K)-BJ1(K)*AYP(K))/BEN(K)
      GNU=(K-1.)*PI/PHBR
      SC=SIN(GNU*PHCR)/(GNU*PHCR)
      SGC(K)=SC
      SGA(K)=SIN(GNU*PHAR)/(GNU*PHAR)
      SUMP=SUMP+SGA(K)*SC/BEN(K)
70    SUML=SUML+RBES(K)*SC*SC
      Z(1)=ETA1*PHBR*(.5*BHR(1)+SUMN)/(ETA*PI)-.5*RUM1/REN1-SUML
      Z(1)=PHCR*Z(1)
      V (1)=CMPLX((1./REN1+2.*SUMP)/(AK*BK1),0.)
      IF(NN.EQ.1)GO TO 130
      DO 120 N=1,NN
      MA=2
      IF(N.GT.2)MA=N
      DO 120 M=MA,NN
      SJN=(.0,.0)
      DO 100 I=2,II
      CALL GNJ(I,M,PHCR,SNC,GIM)
      GIN=SNC(I)
      IF(N.GT.1)CALL GNJ(I,N,PHCR,SNC,GIN)
100   SJN=SJN+BHR(I)*GIM*GIN
      SJL=.0
      CJR=.0
      DO 110 K=2,KK
      CALL GLJ(K,M,PHBR,PHCR,SGC,FKM)
      FKN=SGC(K)
      IF(N.GT.1)CALL GLJ(K,N,PHBR,PHCR,SGC,FKN)
      IF(N.EQ.1)CJR=CJR+FKM*SGA(K)/BEN(K)
110   SJL=SJL+RBES(K)*FKM*FKN
      L=(N-1)*NN-(N*N-N)/2+M
      Z(L)=(ETA1*PHBR*SJN/(ETA*PI)-SJL)*PHCR
120   IF(N.EQ.1)V(M)=CMPLX(2.*CJR/(AK*BK1),0.)
130   CALL SOROT(Z,V,0,1,NN)
      A(1)=-(.0,1.)*PHCR*B(1)/(ETA*PI*CMPLX(BP(1),-YP(1)))
      DO 300 I=2,II
      SC=SNC(I)
      SUL=B(1)*SC
      IF(NN.EQ.1)GO TO 300
      DO 290 N=2,NN
      CALL GNJ(I,N,PHCR,SNC,GIN)
290   SUL=SUL+B(N)*GIN
300   A(I)=-2.*PHCR*(.0,1.)*SUL/(ETA*PI*CMPLX(BP(I),-YP(I)))
      C(1)=(.0,1.)*(2.*AL*PHCR*B(1)*AYP(1)-BYP1(1))/
     2(2.*AL*ETA1*PHBR*REN1)
      D(1)=(.0,1.)*(BJP1(1)-2.*AL*PHCR*B(1)*AJP(1))/
     2(2.*AL*ETA1*PHBR*REN1)
      DO 340 K=2,KK
      SA=SGA(K)
      REN=AL*ETA1*PHBR*BEN(K)
      SUL=B(1)*SGC(K)
      IF(NN.EQ.1)GO TO 330
      DO 320 N=2,NN
      CALL GLJ(K,N,PHBR,PHCR,SGC,FKN)
320   SUL=SUL+B(N)*FKN
```

Fig. 7.

```
330 C(K)=-SA*BYP1(K)+2.*AL*PHCR*AYP(K)*SUL
    C(K)=(.0,1.)*C(K)/REN
    D(K)=SA*BJP1(K)-2.*AL*PHCR*AJP(K)*SUL
340 D(K)=(.0,1.)*D(K)/REN
    Y11=C(1)*AJJ(1)+D(1)*AYY(1)
    DO 360 K=2,KK
360 Y11=Y11+(C(K)*AJJ(K)+D(K)*AYY(K))*SGA(K)
    WRITE(6,2)Y11
    WRITE(6,5)
    GG=REAL(Y11)
    NPH=180./DPH+1.5
    DO 400 L=1,NPH
    PH=(L-1)*DPH
    PHR=.0174533*PH
    HZ=(.0,.0)
    CO=(1.,0.)
    DO 390 I=1,II
    N=I-1
    HZ=HZ+CO*A(I)*COS(N*PHR)
390 CO=CO*(.0,1.)
    HAB=CABS(HZ)/PI
    GAIN=TP*ETA*HAB*HAB/GG
    DB=10.*ALOG10(GAIN)
400 WRITE(6,2)PH,GAIN,DB
    CALL EXIT
    END
```

Fig. 7.

APPENDIX II
SUBROUTINE BESSI

The subroutines are listed in Figs. 8-11. BESSI is a drastically modified and streamlined version of a program developed by Nelson Ma of the Department of Engineering Mechanics at The Ohio State University. This program calculates Bessel and Neumann functions and their derivatives. The argument must be positive and real. The order is positive and real, and it may be integer or noninteger. For the gamma function, BESSI calls subroutine GAMMA from the IBM 360 scientific subroutine package. The input data are defined as follows:

X	argument, greater than zero
ORD	order, greater than zero
IDL	dimension of BJJ, BYY, BJP and BYP
IDM	dimension of BJ and BY

BJ and BY are work arrays for internal use. If ORD is an integer, BJ and BY may have the same names in the calling program as BJJ and BYY to reduce storage requirements. This is illustrated in the third call to BESSI in Fig. 7. The output data are defined as follows:

BJJ(I)	$J_\nu(x)$ with I = 1, 2, 3, $\cdots$ N and ν = (I - 1)*ORD
BYY(I)	$N_\nu(x)$
BJP(I)	$J'_\nu(x)$
BYP(I)	$N'_\nu(x)$
N	maximum value of I

N will not exceed IDL. If IDL and IDM are sufficiently large, N will be determined by the condition that BJJ(N) is less than 10^{-6} or BYY(N) is greater than 10^6. Comparison with other subroutines indicates that the output of BESSI may be accurate even when x is as large as 2000. The upper limit on x is not known.

One call to BESSI generates a series of Bessel and Neumann functions with different orders. For example, if ORD = 0.5 the functions will have orders 0, 1/2, 2/2, 3/2, 4/2, 5/2, etc. If ORD = 2, the functions will have orders 0, 2, 4, 6, etc. These are the orders required in boundary-value problems involving wedges and circular-sector cylinders.

BESSI uses the recursion techniques of Reference [5]. For x greater than 10, the phase amplitude-method is employed[6].

In line 39, the user should replace 1.E-38 with the smallest number his computer can handle without underflow. To obtain a few more Bessel and Neumann functions in the series, one may replace 1.E-6 with a smaller number in line 69, and replace 1.E6 with a larger number in line 154.

To obtain the maximum available number of Bessel and Neumann functions in the series, the required dimensions may be estimated as follows when x is greater than one:

IDM = 1.2 x + 100 - 1500/(x + 20)

IDL = IDM/ORD.

```
      SUBROUTINE BESSI(X,ORD,BJJ,BYY,BJP,BYP,IDL,IDM,N,BJ,BY)
      DIMENSION BJ(1),BY(1),BJJ(1),BYY(1),BJP(1),BYP(1)
      DATA A,PI/.577215665,3.14159265/
      DATA C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16
     B,C17,C18,C19,C20,C21,C22,C23,C24,C25,C26,C27/
     C.25,.15625,-.375,.1171875,-1.15625,1.875,.952148438E-1,
     D-2.38671875,14.2265625,-19.6875,-.809326172E-1,-4.10058593,
     E58.2246094,-277.875,354.375,.416666667E-1,-.25,.0125,-.35,
     F.558035718E-3,-.424107143,3.60267857,-5.625,.30381944E-2,
     G-.486111111,10.2864583,-58.,78.75/
      J=0
      IF(X.LE.0.)GO TO 1
      IF(ORD.LE.0.)GO TO 1
      GO TO 2
1     N=J-1
      RETURN
2     EA=2./X
      INT=ORD+.5
      IN=1000.*(ORD-INT)
      TLOG=ALOG(X/2.)
      PIH=2./PI
      T2=1./(X*X)
      PI4=4./PI
      GAMM1=PIH*(A+TLOG)
      KMAX=X+10.*(2.*X**.333333+1.)
      SQPX=SQRT(.5*PI*X)
      TPX=2./(PI*X)
10    J=J+1
      JM=J-1
      FNUP=JM*ORD
      N=FNUP
      FNU=FNUP-N
      IF(IN.EQ.0)FNU=.0
      IF(IN.EQ.0)N=1
      NP1=N+1
      IF(NP1.GT.IDM)GO TO 1
      NM1=N-1
      K=KMAX
      IF(K.LT.NP1 .AND. IN.NE.0)GO TO 1
      I=K
      BJC=.0
      BJB=1.E-38
      EB=EA*(I+FNU)
35    BJA=EB*BJB-BJC
      IF(I.LE.IDM)BJ(I)=BJA
      EB=EB-EA
      BJC=BJB
      BJB=BJA
      I=I-1
      IF(I.GE.1)GO TO 35
      IF(K.GT.IDM)K=IDM
      M=(K-1)/2
      IF(X.GE.10.)GO TO 59
      PHI=FNU+2.
      MO=3
      ALF=PHI*BJ(3)+BJ(1)
      DO 39 I=2,M
      MO=MO+2
      FM2=2*I
      FM1=I-1
      FI=I
      TEMP=((FNU+FM2)*(FNU+FM1))/(FI*(FNU+FM2-2.0))*PHI
```

Fig. 8. Subroutine BESSI.

```
      PHI=TEMP
 39   ALF=PHI*BJ(MO)+ALF
      GAMM=GAMMA(FNU+1.)
      ALF=EA**FNU*GAMM*ALF
 41   AJ1=1.
      AJ2=1.
      JAN=0
      RALF=1./ALF
      DO 43 I=1,K
      IF(JAN.EQ.1)GO TO 43
      IF(AJ1.LT.1.E-6 .AND. AJ2.LT.1.E-6)JAN=1
      BJ(I)=BJ(I)*RALF
      AJ1=AJ2
      AJ2=ABS(BJ(I))
      IMX=I
 43   CONTINUE
      K=IMX
      M=(K-1)/2
      IF(IN.NE.0 .AND. IMX.LT.NP1)GO TO 1
      GO TO 100
 59   KOUNT=1
       GNU=FNU
 61   AL1=GNU**2-.25
       A2 =C0*AL1
      A4=(C1*AL1+C2)*AL1
      A6=((C3*AL1+C4)*AL1+C5)*AL1
      A8=(((C6*AL1+C7)*AL1+C8)*AL1+C9)*AL1
      A10=((((C10*AL1+C11)*AL1+C12)*AL1+C13)*AL1+C14)*AL1
      B=(((A10*T2+A8)*T2+A6)*T2+A4)*T2+A2
       BNU=B*T2+1.0
      ANU=BNU/SQPX
       A2=.5*AL1
      A4=(C15*AL1+C16)*AL1
      A6=((C17*AL1+C18)*AL1+.75)*AL1
      A8=(((C19*AL1+C20)*AL1+C21)*AL1+C22)*AL1
      A10=((((C23*AL1+C24)*AL1+C25)*AL1+C26)*AL1+C27)*AL1
      B=(((A10*T2+A8)*T2+A6)*T2+A4)*T2+A2
       TPHI=B*T2+1.0
      PHI=TPHI*X-(GNU+.5)/PIH
      F1=ANU*COS(PHI)
      Y1=ANU*SIN(PHI)
      IF(KOUNT.GT.1)GO TO 65
      FSAVE=F1
      BY(1)=Y1
       GNU=FNU+1.0
       KOUNT=2
      GO TO 61
 65   F2=F1
       BY(2)=Y1
       F1=FSAVE
      ALF=BJ(2)/F2
      IF(ABS(F1).GT.ABS(F2))ALF=BJ(1)/F1
      GO TO 41
100   IF(X.GE.10.)GO TO 150
      ARG=FNU*PI
      GARG=GAMM**2
      IF(FNU.EQ.0.)GO TO 116
      TERM=(1./PI)*EA**(2.*FNU)
      GAM1=COS(ARG)/SIN(ARG)-TERM*(GARG/FNU)
      GAM2=2.0*TERM*GARG*(FNU+2.0)/(1.0-FNU)
      GO TO 117
116   GAM1=GAMM1
```

Fig. 8.

```
      GAM2=PI4
  117 BY(2)=-(1./PI)*BJ(1)*EA**(1.+2.*FNU)*GARG+(GAM1-GAM2/2.)*BJ(2)
      YNU=GAM1*BJ(1)
      TXNU=3.0*FNU/X
      AB=ABS(BJ(1))-0.000005
       MP1=M+1
      I2=1
      DO 121 I=2,MP1
      I2=I2+2
      FI=I
      FIM=I-1
      FI2=2*I
      DENOM=FI*(FI-FNU)*(FNU+FI2-2.0)
      GAM3=(FNU+FI2)*(2.0*FNU+FIM)*(FNU+FIM)/DENOM
      GAM3=-GAM3*GAM2
      YNU=GAM2*BJ(I2)+YNU
      IF(AB.GT.0.)GO TO 120
      E1=TXNU*GAM2
      BY(2)=E1*BJ(I2)+BY(2)
      IF(I2.GE.K)GO TO 130
      E1=(GAM2-GAM3)/2.
      BY(2)=E1*BJ(I2+1)+BY(2)
  120 GAM1=GAM2
  121 GAM2=GAM3
  130 BY(1)=YNU
      IF(AB.GT.0.)BY(2)=(YNU*BJ(2)-TPX)/BJ(1)
  150 JAN=0
      ABY=ABS(BY(2))
      MAX=NM1
      IF(IN.EQ.0)MAX=K
      DO 160 I=1,MAX
      IF(JAN.EQ.1)GO TO 160
      IMX=I+2
      IF(ABY.GT.1.E6)JAN=1
      BY(I+2)=EA*(I+FNU)*BY(I+1)-BY(I)
      ABY=ABS(BY(I+2))
  160 CONTINUE
      IF(IN.EQ.0)GO TO 300
      IF(IMX.LT.NP1)GO TO 1
      BJJ(J)=BJ(NP1)
      BYY(J)=BY(NP1)
      IF(J.GT.1)GO TO 210
      BJP(1)=-BJ(2)
      BYP(1)=-BY(2)
      GO TO 220
  210 FAC=FNUP/X
      BJP(J)=-FAC*BJ(NP1)+BJ(N)
      BYP(J)=-FAC*BY(NP1)+BY(N)
  220 IF(J.LT.IDL)GO TO 10
      N=J
      RETURN
  300 BJJ(1)=BJ(1)
      BYY(1)=BY(1)
      BJP(1)=-BJ(2)
      BYP(1)=-BY(2)
      N=K
      IF(IMX.LT.K)N=IMX
      N=1+(N-1)/INT
      IF(N.GT.IDL)N=IDL
      DO 350 I=2,N
      L=1+(I-1)*INT
      LM=L-1
```

Fig. 8.

```
      BJJ(I)=BJ(L)
      BYY(I)=BY(L)
      FAC=LM/X
      BJP(I)=-FAC*BJ(L)+BJ(LM)
  350 BYP(I)=-FAC*BY(L)+BY(LM)
      RETURN
      END
```

Fig. 8.

```
      SUBROUTINE SQROT(C,S,IWR,I12,NEQ)
      COMPLEX C(1),S(1),SS
2     FORMAT(1X,1I5,1F10.3,1F15.7,1F10.0,2F15.6)
3     FORMAT(1H0)
      N=NEQ
      IF(I12.EQ.2)GO TO 20
      C(1)=CSQRT(C(1))
      DO 4 K=2,N
4     C(K)=C(K)/C(1)
      DO 10 I=2,N
      IMO=I-1
      IPO=I+1
      ID=(I-1)*N-(I*I-I)/2
      II=ID+I
      DO 5 L=1,IMO
      LI=(L-1)*N-(L*L-L)/2+I
5     C(II)=C(II)-C(LI)*C(LI)
      C(II)=CSQRT(C(II))
      IF(IPO.GT.N)GO TO 10
      DO 8 J=IPO,N
      IJ=ID+J
      DO 6 M=1,IMO
      MD=(M-1)*N-(M*M-M)/2
      MI=MD+I
      MJ=MD+J
    6 C(IJ)=C(IJ)-C(MJ)*C(MI)
8     C(IJ)=C(IJ)/C(II)
10    CONTINUE
20    S(1)=S(1)/C(1)
      DO 30 I=2,N
      IMO=I-1
      DO 25 L=1,IMO
      LI=(L-1)*N-(L*L-L)/2+I
25    S(I)=S(I)-C(LI)*S(L)
      II=(I-1)*N-(I*I-I)/2+I
30    S(I)=S(I)/C(II)
      NN=((N+1)*N)/2
      S(N)=S(N)/C(NN)
      NMO=N-1
      DO 40 I=1,NMO
      K=N-I
      KPO=K+1
      KD=(K-1)*N-(K*K-K)/2
      DO 35 L=KPO,N
      KL=KD+L
35    S(K)=S(K)-C(KL)*S(L)
      KK=KD+K
40    S(K)=S(K)/C(KK)
      IF(IWR.LE.0) GO TO 100
      WRITE(6,3)
      CNOR=.0
      DO 50 I=1,N
      SA=CABS(S(I))
50    IF(SA.GT.CNOR)CNOR=SA
      IF(CNOR.LE.0.)CNOR=1.
      DO 60 I=1,N
      SS=S(I)
      SA=CABS(SS)
      SNOR=SA/CNOR
      PH=.0
      IF(SA.GT.0.)PH=57.29578*ATAN2(AIMAG(SS),REAL(SS))
60    WRITE(6,2)I,SNOR,SA,PH,SS
```

Fig. 9. Subroutine SQROT.

```
      WRITE(6,3)
  100 RETURN
      END
```

Fig. 9.

```
SUBROUTINE GLJ(LL,JJ,PHBR,PHCR,SGC,FLJ)
DIMENSION SGC(150)
DATA PI/3.14159/
J=JJ-1
SGJ=(-1)**J
L=LL-1
FLJ=.5
GNU=L*PI/PHBR
SC=SGC(LL)
GNUS=GNU*GNU
TEST=ABS(GNU-J*PI/PHCR)
DEN=GNUS-(J*PI/PHCR)**2
IF(TEST.GT..001)FLJ=SGJ*GNUS*SC/DEN
RETURN
END
```

Fig. 10. Subroutine GLJ.

```
SUBROUTINE GNJ(M,JJ,PHCR,SNC,FNJ)
DIMENSION SNC(150)
DATA PI/3.14159/
J=JJ-1
SGJ=(-1)**J
N=M-1
NS=N*N
SC=SNC(M)
FNJ=.5
TEST=ABS(N-J*PI/PHCR)
DEN=NS-(J*PI/PHCR)**2
IF(TEST.GT..001)FNJ=SGJ*NS*SC/DEN
RETURN
END
```

Fig. 11. Subroutine GNJ.

APPENDIX III
SUBROUTINE SQROT

This subroutine considers the matrix equation ZI = V which represents a system of simultaneous linear equations. If the square matrix Z is symmetric, SQROT is useful for obtaining the solution I with V given. NEQ denotes the number of simultaneous equations and the size of the matrix Z.

On entry to SQROT, S is the excitation column V. On exit, the solution I is stored in S. Let Z(I,J) denote the symmetric square matrix. On entry to SQROT, the upper-right triangular portion of Z(I,J) is stored by rows in C(K) with

(37) K = (I - 1)*NEQ - (I*I - I) / 2 + J.

If I12 = 1, SQROT will transform the symmetric matrix into the auxiliary matrix (implicit inverse), store the result in C(K) and use the auxiliary matrix to solve the simultaneous equations. If I12 = 2, this indicates that C(K) already contains the auxiliary matrix.

The transformation from the symmetric matrix to the auxiliary matrix is programmed above statement 10, and the solution of the simultaneous equations is programmed in statements 20 to 40. If IWR is positive, the program below statement 40 will write the solution.

SQROT uses the square root method described in Reference [4]. The original symmetric matrix Z and the upper triangular auxiliary matrix A are related by

(38) Z = A' A

where A' is the transpose of A.

The determinant of the symmetric matrix Z may be obtained by squaring the product of the diagonal elements in the auxiliary matrix.

SQROT was developed by Dr. Robert G. Wickliff Jr., now with Hewlett Packard, Colorado Springs, Colorado 80907.

ACKNOWLEDGMENT

The experimental data presented herein were measured by Melvin C. Gilreath at NASA Langley Research Center. His kind permission to reproduce these antenna patterns is truly appreciated.