2 mic

FLUSH-MOUNTED DIELECTRIC-LOADED AXIAL SLOT ON CIRCULAR CYLINDER

J. H. Richmond

The Ohio State University

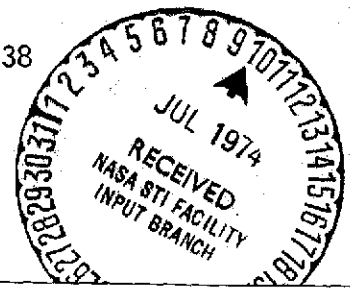
ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

TECHNICAL REPORT 2902-17

Grant Number NGL 36-008-138

June 1974



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

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

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

	Page
I. INTRODUCTION	1
II. THEORY	1
III. NUMERICAL RESULTS	7
IV. SUMMARY AND CONCLUSIONS	13
REFERENCES	14
APPENDIX I - THE MAIN COMPUTER PROGRAM	15
APPENDIX II - SUBROUTINE BESSI	20
APPENDIX III - SUBROUTINE SQROT	29
ACKNOWLEDGMENT	30

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

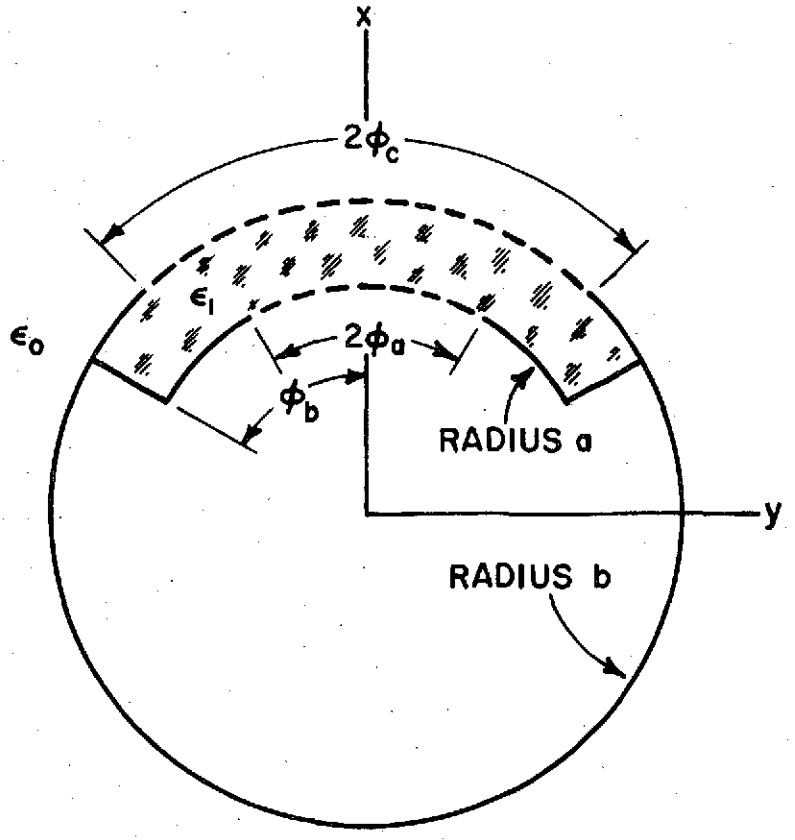


Fig. 1. An axial-slot antenna radiates through a flush-mounted dielectric window in a conducting circular cylinder.

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

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

$$(2) \qquad E_\phi^I = j\eta_1 \sum_k \left[c_k J'_\nu(\rho) + d_k N'_\nu(\rho)\right] \cos\nu\phi$$

$$(3) \qquad H_z^I = \sum_k \left[c_k J_\nu(\rho) + d_k N_\nu(\rho)\right] \cos\nu\phi$$

$$(4) \qquad k_1 = \omega\sqrt{\mu_1 \varepsilon_1}$$

$$(5) \qquad \eta_1 = \sqrt{\mu_1/\varepsilon_1}$$

$$(6) \qquad \nu = k\pi/\phi_b$$

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

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

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

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

The boundary condition at $\rho = a$ is

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

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

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

$$G_k = \int_0^{\phi_a} E \cos\nu\phi \, d\phi \tag{11}$$

where $e_o = 1$ 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

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

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

$$E_\rho^{II} = \frac{j\eta_o}{k_o \rho} \sum_i i \, a_i \, H_i(\rho) \sin i\phi \tag{13}$$

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

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

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

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

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:

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

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 $\rho = b$,

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

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

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

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

From Eqs. (10) and (20),

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

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

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

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

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

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:

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

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

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:

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

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

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

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

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:

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

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.

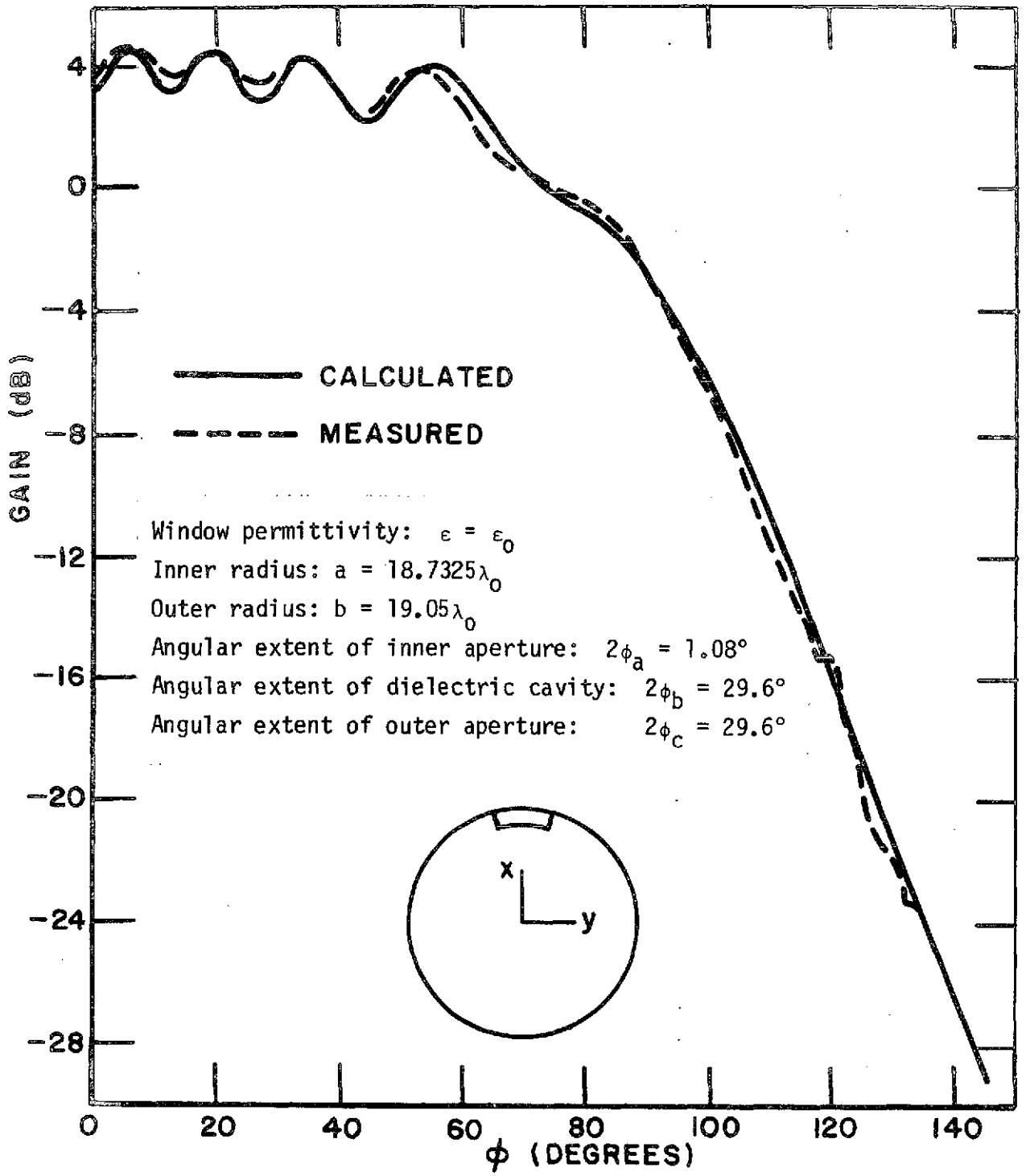


Fig. 2. Far-field pattern with $\varepsilon_r = 1$.

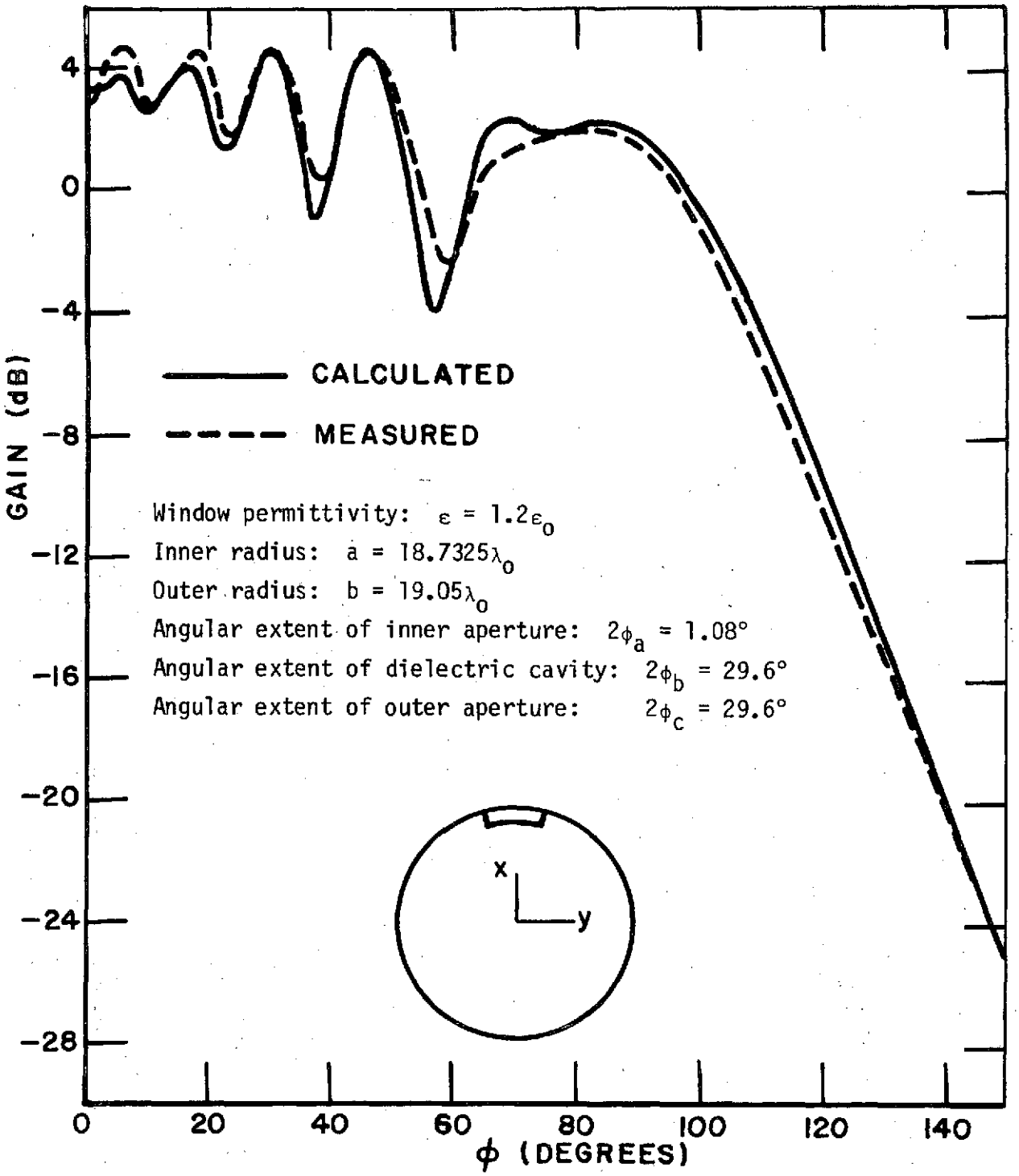


Fig. 3. Far-field pattern with $\varepsilon_r = 1.2$.

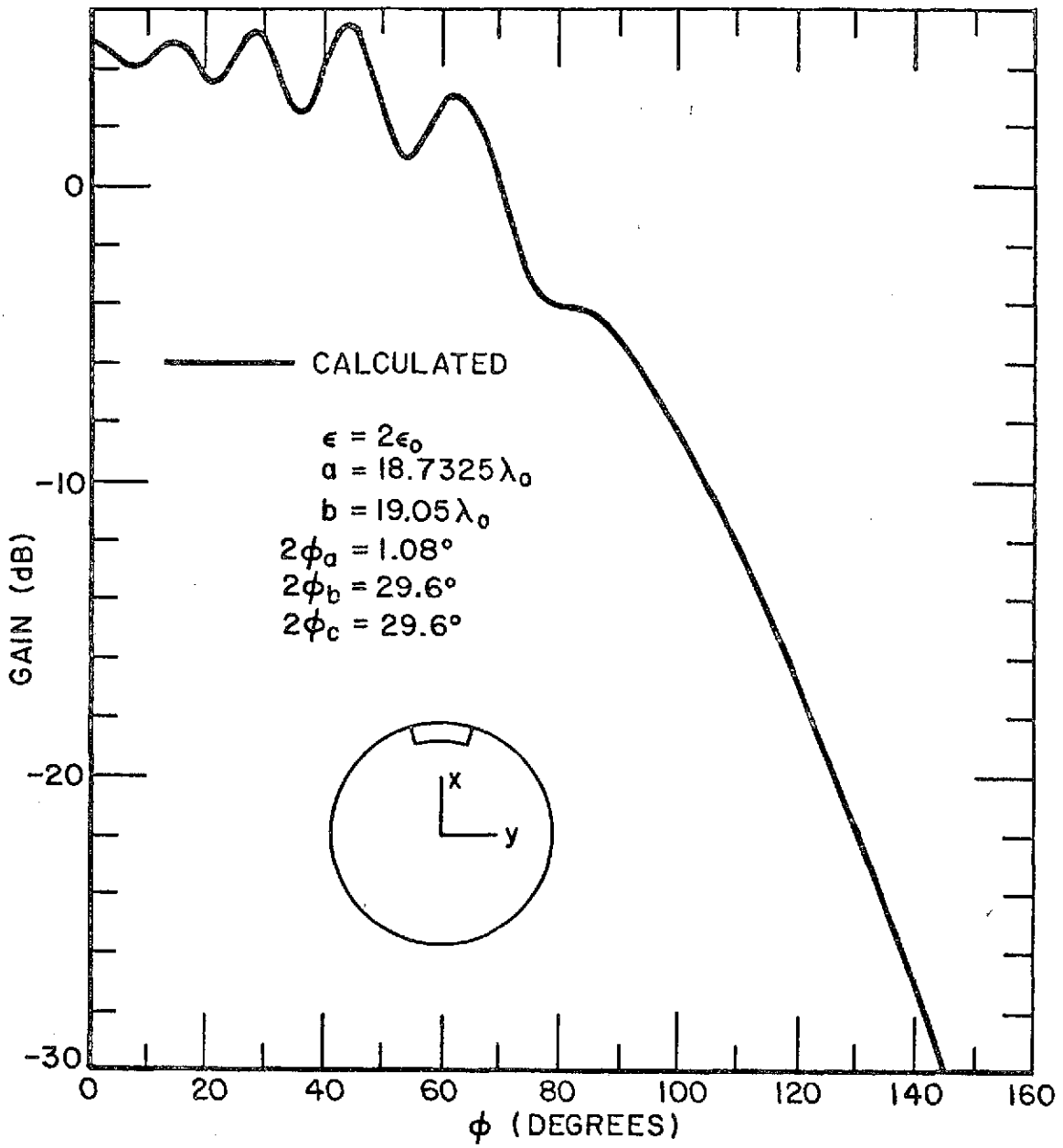


Fig. 4. Far-field pattern with $\epsilon_r = 2$.

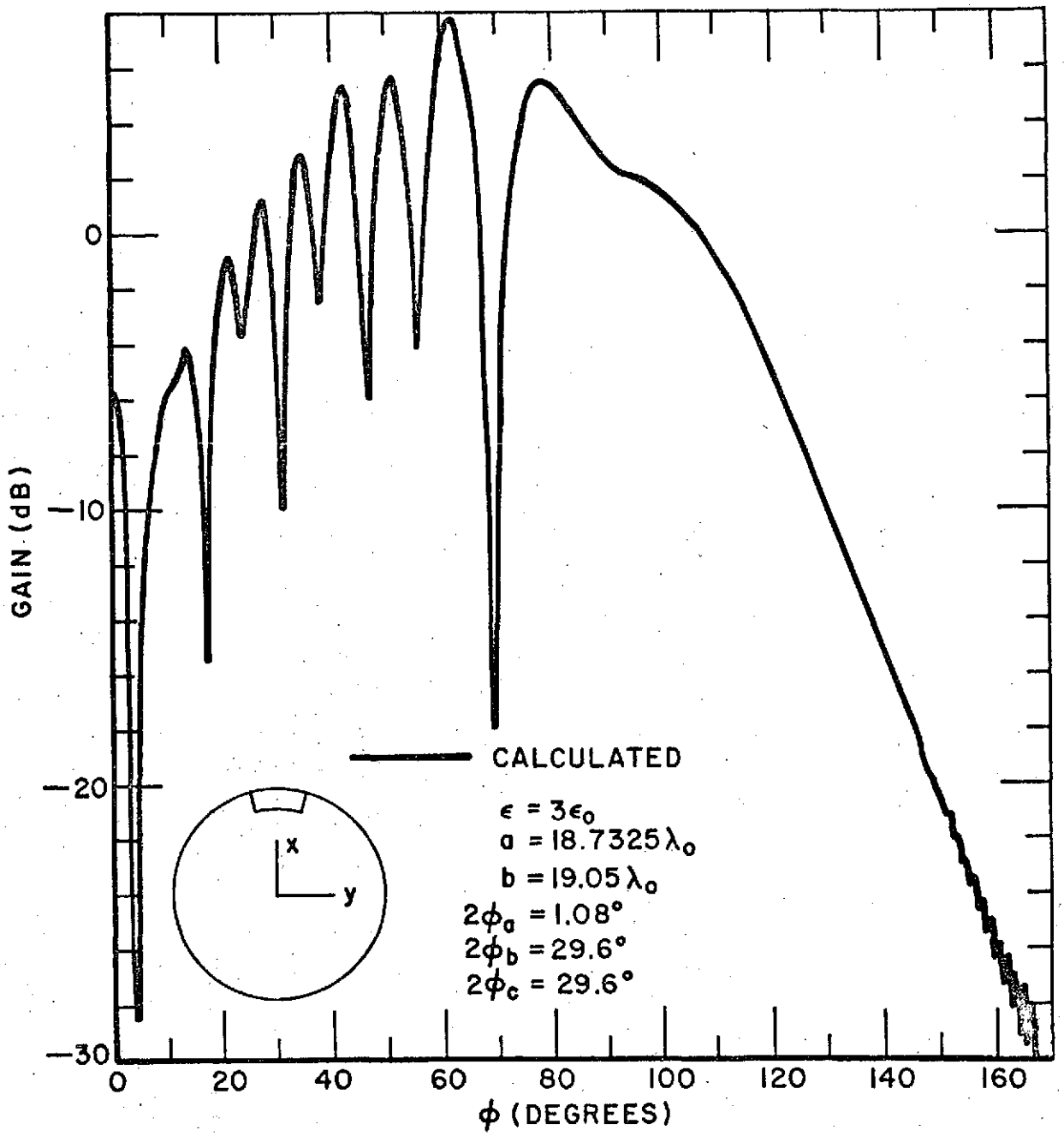


Fig. 5. Far-field pattern with $\varepsilon_r = 3$.

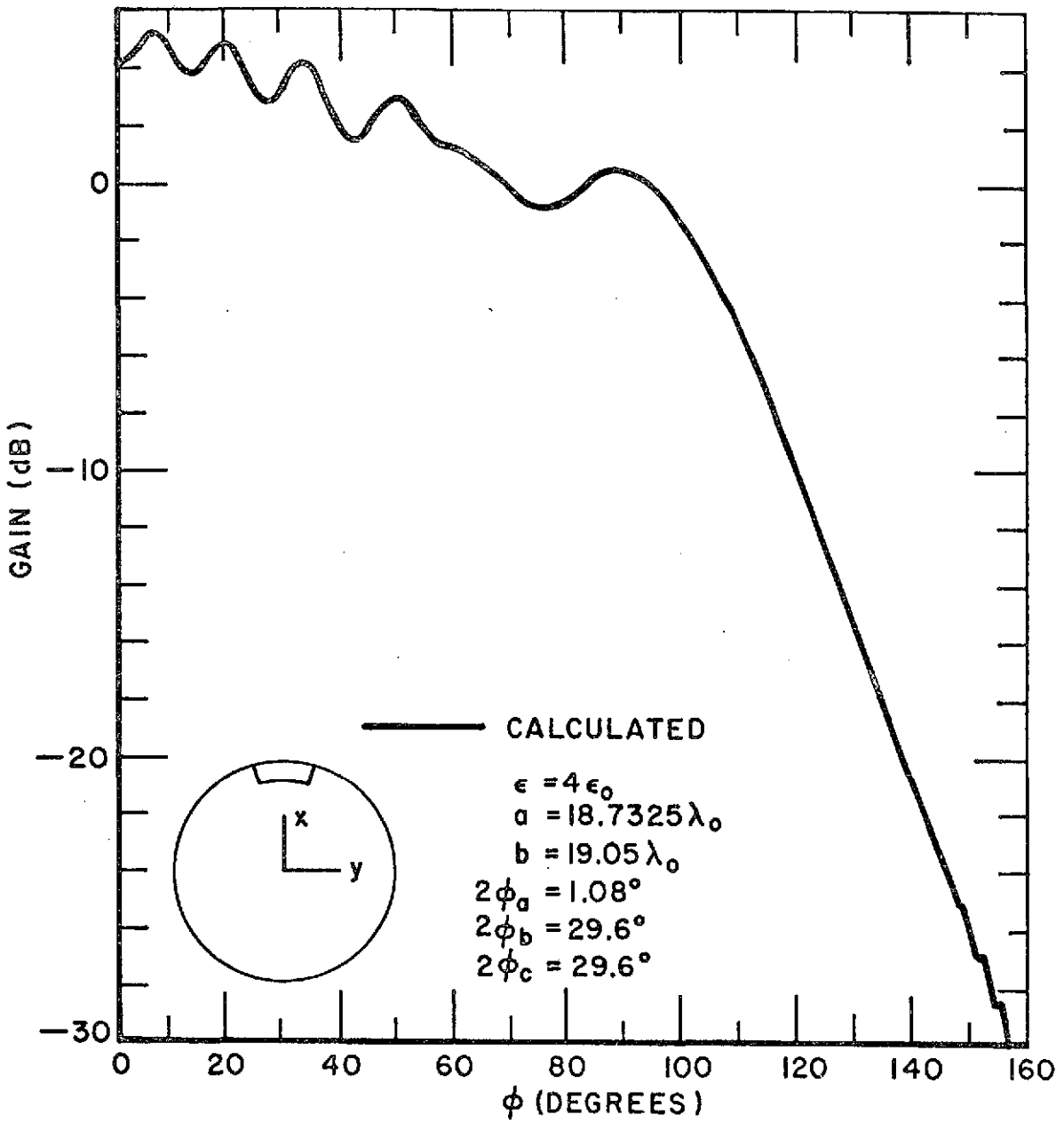


Fig. 6. Far-field pattern with $\varepsilon_r = 4$.

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

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

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	$H_i(b)/H_1'(b)$
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.                0001
C     SLOT RADIATES THROUGH DIELECTRIC WINDOW.                                0002
C     PROGRAM BY JACK H RICHMOND, OHIO STATE UNIVERSITY.                      0003
      COMPLEX V(30),B(30),Z(465),C(150),D(150),A(400),BHR(400)                0004
      COMPLEX CQ,HZ,SUMN,SUL,SJN,Y11                                          0005
      DIMENSION BB(400),YY(400),BP(400),YP(400),SNC(400),SGA(400)             0006
      DIMENSION SGC(150),RBES(150),BEN(150)                                   0007
      DIMENSION AJJ(150),AYY(150),AJP(150),AYP(150)                           0008
      DIMENSION BJ1(150),BY1(150),BJP1(150),BYP1(150)                         0009
      EQUIVALENCE (B,V),(BB,SNC),(YY,SGA),(BJ1,SGC),(BY1,RBES)                0010
      DATA ETA,PI,TP/376.727,3.14159,6.28318/                                 0011
    2 FORMAT(1X,8F15.6)                                                       0012
    4 FORMAT(1X,12I10)                                                        0013
    5 FORMAT(1H0)                                                             0014
      IDC=30                                                                  0015
      IDH=400                                                                 0016
      IDJ=150                                                                 0017
      IDZ=465                                                                 0018
   20 AL=18.7325                                                              0019
      BL=19.05                                                                0020
      ER=3.                                                                   0021
      DPH=2.                                                                  0022
      PHA=0.54                                                                0023
      PHB=14.8                                                                0024
      PHC=PHB                                                                 0025
      IF(PHA.GT.PHB)PHA=PHB                                                   0026
      IF(PHC.GT.PHB)PHC=PHB                                                   0027
      N=.5+(SQRT(1.+8.*IDZ)-1.)/2.                                            0028
      IF(N.LT.IDC)IDC=N                                                       0029
      NN=IDC                                                                  0030
      SQR=SQRT(ER)                                                            0031
      ETA1=ETA/SQR                                                            0032
      TL=BL-AL                                                                0033
      WRITE(6,2)AL,BL,TL,ER,PHA,PHB,PHC                                       0034
      WRITE(6,5)                                                              0035
      AK=TP*AL                                                                0036
      BK=TP*BL                                                                0037
      AK1=AK*SQR                                                              0038
      BK1=BK*SQR                                                              0039
      PHAR=.0174533*PHA                                                       0040
      PHBR=.0174533*PHB                                                       0041
      PHCR=.0174533*PHC                                                       0042
      GNU=PI/PHBR                                                             0043
   30 CALL BESSI(AK1,GNU,AJJ,AYY,AJP,AYP,IDJ,IDH,KK ,BB,YY)                   0044
      CALL BESSI(BK1,GNU,BJ1,BY1,BJP1,BYP1,IDJ,IDH,LL ,BB,YY)                 0045
      CALL BESSI(BK,1.,BB,YY,BP,YP,IDH,IDH,II,BB,YY)                          0046
      IF(LL.LT.KK)KK=LL                                                       0047
      KMIN=BK*PHB/180.                                                        0048
      IMIN=BK                                                                 0049
      NMIN=BK*PHC/180.                                                        0050
      NMAX=.5+KK*PHC/PHB                                                      0051
      IF(NN.GT.IDC)NN=IDC                                                     0052
      IF(NN.GT.NMAX)NN=NMAX                                                   0053
      IF(NN.LT.1)NN=1                                                         0054
      WRITE(6,4)IMIN,II,KMIN,KK,NMIN,NN                                       0055
      WRITE(6,5)                                                              0056
      SUMN=(.0,.0)                                                            0057
      BHR(1)=CMPLX(BB(1),-YY(1))/CMPLX(BP(1),-YP(1))                          0058
      DO 60 I=2,II                                                            0059
      BHR(I)=CMPLX(BB(I),-YY(I))/CMPLX(BP(I),-YP(I))                          0060
      N=I-1                                                                   0061
      SC=SIN(N*PHCR)/(N*PHCR)                                                 0062
```

Fig. 7. The MAIN computer program.

```
      SNC(I)=SC                                                          0063
   60 SUMN=SUMN+BHR(I)*SC*SC                                             0064
      RUM1=AJP(1)*BY1(1)-BJ1(1)*AYP(1)                                   0065
      REN1=AJP(1)*BYP1(1)-BJP1(1)*AYP(1)                                 0066
      SUML=.0                                                            0067
      SUMP=.0                                                            0068
      DO 70 K=2,KK                                                       0069
      BEN(K)=AJP(K)*BYP1(K)-BJP1(K)*AYP(K)                               0070
      RBES(K)=(AJP(K)*BY1(K)-BJ1(K)*AYP(K))/BEN(K)                       0071
      GNU=(K-1.)*PI/PHBR                                                 0072
      SC=SIN(GNU*PHCR)/(GNU*PHCR)                                        0073
      SGC(K)=SC                                                          0074
      SGA(K)=SIN(GNU*PHAR)/(GNU*PHAR)                                    0075
      SUMP=SUMP+SGA(K)*SC/BEN(K)                                         0076
   70 SUML=SUML+RBES(K)*SC*SC                                            0077
      Z(1)=ETA1*PHBR*(.5*BHR(1)+SUMN)/(ETA*PI)-.5*RUM1/REN1-SUML         0078
      Z(1)=PHCR*Z(1)                                                     0079
      V(1)=CMPLX((1./REN1+2.*SUMP)/(AK*BK1),0.)                          0080
      IF(NN.EQ.1)GO TO 130                                               0081
      DO 120 N=1,NN                                                      0082
      MA=2                                                               0083
      IF(N.GT.2)MA=N                                                     0084
      DO 120 M=MA,NN                                                     0085
      SJN=(.0,.0)                                                        0086
      DO 100 I=2,II                                                      0087
      CALL GNJ(I,M,PHCR,SNC,GIM)                                         0088
      GIN=SNC(I)                                                         0089
      IF(N.GT.1)CALL GNJ(I,N,PHCR,SNC,GIN)                               0090
  100 SJN=SJN+BHR(I)*GIM*GIN                                             0091
      SJL=.0                                                             0092
      CJR=.0                                                             0093
      DO 110 K=2,KK                                                      0094
      CALL GLJ(K,M,PHBR,PHCR,SGC,FKM)                                    0095
      FKN=SGC(K)                                                         0096
      IF(N.GT.1)CALL GLJ(K,N,PHBR,PHCR,SGC,FKN)                          0097
      IF(N.EQ.1)CJR=CJR+FKM*SGA(K)/BEN(K)                                0098
  110 SJL=SJL+RBES(K)*FKM*FKN                                            0099
      L=(N-1)*NN-(N*N-N)/2+M                                             0100
      Z(L)=(ETA1*PHBR*SJN/(ETA*PI)-SJL)*PHCR                             0101
  120 IF(N.EQ.1)V(M)=CMPLX(2.*CJR/(AK*BK1),0.)                           0102
  130 CALL SOROT(Z,V,0,1,NN)                                             0103
      A(1)=-(.0,1.)*PHCR*B(1)/(ETA*PI*CMPLX(BP(1),-YP(1)))               0104
      DO 300 I=2,II                                                      0105
      SC=SNC(I)                                                          0106
      SUL=B(1)*SC                                                        0107
      IF(NN.EQ.1)GO TO 300                                               0108
      DO 290 N=2,NN                                                      0109
      CALL GNJ(I,N,PHCR,SNC,GIN)                                         0110
  290 SUL=SUL+B(N)*GIN                                                   0111
  300 A(I)=-2.*PHCR*(.0,1.)*SUL/(ETA*PI*CMPLX(BP(I),-YP(I)))             0112
      C(1)=(.0,1.)*(2.*AL*PHCR*B(1)*AYP(1)-BYP1(1))/                     0113
     2(2.*AL*ETA1*PHBR*REN1)                                             0114
      D(1)=(.0,1.)*(BJP1(1)-2.*AL*PHCR*B(1)*AJP(1))/                     0115
     2(2.*AL*ETA1*PHBR*REN1)                                             0116
      DO 340 K=2,KK                                                      0117
      SA=SGA(K)                                                          0118
      REN=AL*ETA1*PHBR*BEN(K)                                            0119
      SUL=B(1)*SGC(K)                                                    0120
      IF(NN.EQ.1)GO TO 330                                               0121
      DO 320 N=2,NN                                                      0122
      CALL GLJ(K,N,PHBR,PHCR,SGC,FKN)                                    0123
  320 SUL=SUL+B(N)*FKN                                                   0124
```

Fig. 7.

```
330 C(K)=-SA*BYP1(K)+2.*AL*PHCR*AYP(K)*SUL                          0125
    C(K)=(.0,1.)*C(K)/REN                                          0126
    D(K)=SA*BJP1(K)-2.*AL*PHCR*AJP(K)*SUL                          0127
340 D(K)=(.0,1.)*D(K)/REN                                          0128
    Y11=C(1)*AJJ(1)+D(1)*AYY(1)                                    0129
    DO 360 K=2,KK                                                  0130
360 Y11=Y11+(C(K)*AJJ(K)+D(K)*AYY(K))*SGA(K)                       0131
    WRITE(6,2)Y11                                                  0132
    WRITE(6,5)                                                     0133
    GG=REAL(Y11)                                                   0134
    NPH=180./DPH+1.5                                               0135
    DO 400 L=1,NPH                                                 0136
    PH=(L-1)*DPH                                                   0137
    PHR=.0174533*PH                                                0138
    HZ=(.0,.0)                                                     0139
    CQ=(1.,0.)                                                     0140
    DO 390 I=1,II                                                  0141
    N=I-1                                                          0142
    HZ=HZ+CQ*A(I)*COS(N*PHR)                                       0143
390 CQ=CQ*(.0,1.)                                                  0144
    HAB=CABS(HZ)/PI                                                0145
    GAIN=TP*ETA*HAB*HAB/GG                                         0146
    DB=10.*ALOG10(GAIN)                                            0147
400 WRITE(6,2)PH,GAIN,DB                                           0148
    CALL EXIT                                                      0149
    END                                                            0150
```

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, ··· N and $\nu = (I - 1)*ORD$
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                                                          63
   39 ALF=PHI*BJ(MO)+ALF                                                64
      GAMM=GAMMA(FNU+1.)                                                65
      ALF=EA**FNU*GAMM*ALF                                              66
   41 AJ1=1.                                                            67
      AJ2=1.                                                            68
      JAN=0                                                             69
      RALF=1./ALF                                                       70
      DO 43 I=1,K                                                       71
      IF(JAN.EQ.1)GO TO 43                                              72
      IF(AJ1.LT.1.E-6 .AND. AJ2.LT.1.E-6)JAN=1                          73
      BJ(I)=BJ(I)*RALF                                                  74
      AJ1=AJ2                                                           75
      AJ2=ABS(BJ(I))                                                    76
      IMX=I                                                             77
   43 CONTINUE                                                          78
      K=IMX                                                             79
      M=(K-1)/2                                                         80
      IF(IN.NE.0 .AND. IMX.LT.NP1)GO TO 1                               81
      GO TO 100                                                         82
   59 KOUNT=1                                                           83
       GNU=FNU                                                          84
   61 AL1=GNU**2-.25                                                    85
       A2 =C0*AL1                                                       86
      A4=(C1*AL1+C2)*AL1                                                87
      A6=((C3*AL1+C4)*AL1+C5)*AL1                                       88
      A8=(((C6*AL1+C7)*AL1+C8)*AL1+C9)*AL1                              89
      A10=((((C10*AL1+C11)*AL1+C12)*AL1+C13)*AL1+C14)*AL1               90
      B=(((A10*T2+A8)*T2+A6)*T2+A4)*T2+A2                               91
       BNU=B*T2+1.0                                                     92
      ANU=BNU/SQPX                                                      93
       A2=.5*AL1                                                        94
      A4=(C15*AL1+C16)*AL1                                              95
      A6=((C17*AL1+C18)*AL1+.75)*AL1                                    96
      A8=(((C19*AL1+C20)*AL1+C21)*AL1+C22)*AL1                          97
      A10=((((C23*AL1+C24)*AL1+C25)*AL1+C26)*AL1+C27)*AL1               98
      B=(((A10*T2+A8)*T2+A6)*T2+A4)*T2+A2                               99
       TPHI=B*T2+1.0                                                   100
      PHI=TPHI*X-(GNU+.5)/PIH                                          101
      F1=ANU*COS(PHI)                                                  102
      Y1=ANU*SIN(PHI)                                                  103
      IF(KOUNT.GT.1)GO TO 65                                           104
      FSAVE=F1                                                         105
      BY(1)=Y1                                                         106
       GNU=FNU+1.0                                                     107
       KOUNT=2                                                         108
      GO TO 61                                                         109
   65 F2=F1                                                            110
       BY(2)=Y1                                                        111
       F1=FSAVE                                                        112
      ALF=BJ(2)/F2                                                     113
      IF(ABS(F1).GT.ABS(F2))ALF=BJ(1)/F1                               114
      GO TO 41                                                         115
  100 IF(X.GE.10.)GO TO 150                                            116
      ARG=FNU*PI                                                       117
      GARG=GAMM**2                                                     118
      IF(FNU.EQ.0.)GO TO 116                                           119
      TERM=(1./PI)*EA**(2.*FNU)                                        120
      GAM1=COS(ARG)/SIN(ARG)-TERM*(GARG/FNU)                           121
      GAM2=2.0*TERM*GARG*(FNU+2.0)/(1.0-FNU)                           122
      GO TO 117                                                        123
  116 GAM1=GAMM1                                                       124
```

Fig. 8.

```
      GAM2=PI4                                                          125
  117 BY(2)=-(1./PI)*BJ(1)*EA**(1.+2.*FNU)*GARG+(GAM1-GAM2/2.)*BJ(2)    126
      YNU=GAM1*BJ(1)                                                    127
      TXNU=3.0*FNU/X                                                    128
      AB=ABS(BJ(1))-0.000005                                            129
       MP1=M+1                                                          130
      I2=1                                                              131
      DO 121 I=2,MP1                                                    132
      I2=I2+2                                                           133
      FI=I                                                              134
      FIM=I-1                                                           135
      FI2=2*I                                                           136
      DENOM=FI*(FI-FNU)*(FNU+FI2-2.0)                                   137
      GAM3=(FNU+FI2)*(2.0*FNU+FIM)*(FNU+FIM)/DENOM                      138
      GAM3=-GAM3*GAM2                                                   139
      YNU=GAM2*BJ(I2)+YNU                                               140
      IF(AB.GT.0.)GO TO 120                                             141
      E1=TXNU*GAM2                                                      142
      BY(2)=E1*BJ(I2)+BY(2)                                             143
      IF(I2.GE.K)GO TO 130                                              144
      E1=(GAM2-GAM3)/2.                                                 145
      BY(2)=E1*BJ(I2+1)+BY(2)                                           146
  120 GAM1=GAM2                                                         147
  121 GAM2=GAM3                                                         148
  130 BY(1)=YNU                                                         149
      IF(AB.GT.0.)BY(2)=(YNU*BJ(2)-TPX)/BJ(1)                           150
  150 JAN=0                                                             151
      ABY=ABS(BY(2))                                                    152
      MAX=NM1                                                           153
      IF(IN.EQ.0)MAX=K                                                  154
      DO 160 I=1,MAX                                                    155
      IF(JAN.EQ.1)GO TO 160                                             156
      IMX=I+2                                                           157
      IF(ABY.GT.1.E6)JAN=1                                              158
      BY(I+2)=EA*(I+FNU)*BY(I+1)-BY(I)                                  159
      ABY=ABS(BY(I+2))                                                  160
  160 CONTINUE                                                          161
      IF(IN.EQ.0)GO TO 300                                              162
      IF(IMX.LT.NP1)GO TO 1                                             163
      BJJ(J)=BJ(NP1)                                                    164
      BYY(J)=BY(NP1)                                                    165
      IF(J.GT.1)GO TO 210                                               166
      BJP(1)=-BJ(2)                                                     167
      BYP(1)=-BY(2)                                                     168
      GO TO 220                                                         169
  210 FAC=FNUP/X                                                        170
      BJP(J)=-FAC*BJ(NP1)+BJ(N)                                         171
      BYP(J)=-FAC*BY(NP1)+BY(N)                                         172
  220 IF(J.LT.IDL)GO TO 10                                              173
      N=J                                                               174
      RETURN                                                            175
  300 BJJ(1)=BJ(1)                                                      176
      BYY(1)=BY(1)                                                      177
      BJP(1)=-BJ(2)                                                     178
      BYP(1)=-BY(2)                                                     179
      N=K                                                               180
      IF(IMX.LT.K)N=IMX                                                 181
      N=1+(N-1)/INT                                                     182
      IF(N.GT.IDL)N=IDL                                                 183
      DO 350 I=2,N                                                      184
      L=1+(I-1)*INT                                                     185
      LM=L-1                                                            186
```

Fig. 8.

```
      BJJ(I)=BJ(L)                                                    187
      BYY(I)=BY(L)                                                    188
      FAC=LM/X                                                        189
      BJP(I)=-FAC*BJ(L)+BJ(LM)                                        190
  350 BYP(I)=-FAC*BY(L)+BY(LM)                                        191
      RETURN                                                          192
      END                                                             193
```

Fig. 8.

```
      SUBROUTINE SQROT(C,S,IWR,I12,NEQ)                                 0001
      COMPLEX C(1),S(1),SS                                              0002
2     FORMAT(1X,I15,1F10.3,1F15.7,1F10.0,2F15.6)                        0003
3     FORMAT(1H0)                                                       0004
      N=NEQ                                                             0005
      IF(I12.EQ.2)GO TO 20                                              0006
      C(1)=CSQRT(C(1))                                                  0007
      DO 4 K=2,N                                                        0008
4     C(K)=C(K)/C(1)                                                    0009
      DO 10 I=2,N                                                       0010
      IMO=I-1                                                           0011
      IPO=I+1                                                           0012
      ID=(I-1)*N-(I*I-I)/2                                              0013
      II=ID+I                                                           0014
      DO 5 L=1,IMO                                                      0015
      LI=(L-1)*N-(L*L-L)/2+I                                            0016
5     C(II)=C(II)-C(LI)*C(LI)                                           0017
      C(II)=CSQRT(C(II))                                                0018
      IF(IPO.GT.N)GO TO 10                                              0019
      DO 8 J=IPO,N                                                      0020
      IJ=ID+J                                                           0021
      DO 6 M=1,IMO                                                      0022
      MD=(M-1)*N-(M*M-M)/2                                              0023
      MI=MD+I                                                           0024
      MJ=MD+J                                                           0025
6     C(IJ)=C(IJ)-C(MJ)*C(MI)                                           0026
8     C(IJ)=C(IJ)/C(II)                                                 0027
10    CONTINUE                                                          0028
20    S(1)=S(1)/C(1)                                                    0029
      DO 30 I=2,N                                                       0030
      IMO=I-1                                                           0031
      DO 25 L=1,IMO                                                     0032
      LI=(L-1)*N-(L*L-L)/2+I                                            0033
25    S(I)=S(I)-C(LI)*S(L)                                              0034
      II=(I-1)*N-(I*I-I)/2+I                                            0035
30    S(I)=S(I)/C(II)                                                   0036
      NN=((N+1)*N)/2                                                    0037
      S(N)=S(N)/C(NN)                                                   0038
      NMO=N-1                                                           0039
      DO 40 I=1,NMO                                                     0040
      K=N-I                                                             0041
      KPO=K+1                                                           0042
      KD=(K-1)*N-(K*K-K)/2                                              0043
      DO 35 L=KPO,N                                                     0044
      KL=KD+L                                                           0045
35    S(K)=S(K)-C(KL)*S(L)                                              0046
      KK=KD+K                                                           0047
40    S(K)=S(K)/C(KK)                                                   0048
      IF(IWR.LE.0) GO TO 100                                            0049
      WRITE(6,3)                                                        0050
      CNOR=.0                                                           0051
      DO 50 I=1,N                                                       0052
      SA=CABS(S(I))                                                     0053
50    IF(SA.GT.CNOR)CNOR=SA                                             0054
      IF(CNOR.LE.0.)CNOR=1.                                             0055
      DO 60 I=1,N                                                       0056
      SS=S(I)                                                           0057
      SA=CABS(SS)                                                       0058
      SNOR=SA/CNOR                                                      0059
      PH=.0                                                             0060
      IF(SA.GT.0.)PH=57.29578*ATAN2(AIMAG(SS),REAL(SS))                 0061
60    WRITE(6,2)I,SNOR,SA,PH,SS                                         0062
```

Fig. 9. Subroutine SQROT.

```
      WRITE(6,3)                                                    0063
  100 RETURN                                                        0064
      END                                                           0065
```

Fig. 9.

```
      SUBROUTINE GLJ(LL,JJ,PHBR,PHCR,SGC,FLJ)                          0001
      DIMENSION SGC(150)                                               0002
      DATA PI/3.14159/                                                 0003
      J=JJ-1                                                           0004
      SGJ=(-1)**J                                                      0005
      L=LL-1                                                           0006
      FLJ=.5                                                           0007
      GNU=L*PI/PHBR                                                    0008
      SC=SGC(LL)                                                       0009
      GNUS=GNU*GNU                                                     0010
      TEST=ABS(GNU-J*PI/PHCR)                                          0011
      DEN=GNUS-(J*PI/PHCR)**2                                          0012
      IF(TEST.GT..001)FLJ=SGJ*GNUS*SC/DEN                              0013
      RETURN                                                           0014
      END                                                              0015
```

Fig. 10. Subroutine GLJ.

```
      SUBROUTINE GNJ(M,JJ,PHCR,SNC,FNJ)                                0001
      DIMENSION SNC(150)                                               0002
      DATA PI/3.14159/                                                 0003
      J=JJ-1                                                           0004
      SGJ=(-1)**J                                                      0005
      N=M-1                                                            0006
      NS=N*N                                                           0007
      SC=SNC(M)                                                        0008
      FNJ=.5                                                           0009
      TEST=ABS(N-J*PI/PHCR)                                            0010
      DEN=NS-(J*PI/PHCR)**2                                            0011
      IF(TEST.GT..001)FNJ=SGJ*NS*SC/DEN                                0012
      RETURN                                                           0013
      END                                                              0014
```

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.