Polytechnic
UNIVERSITY

Goddard/Grant

IN-32

63675-CR

P.63

FINAL REPORT

TO

GODDARD SPACE FLIGHT CENTER

ON

FREQUENCY SELECTIVE REFLECTION AND TRANSMISSION

AT A

LAYER COMPOSED OF A PERIODIC DIELECTRIC

PREPARED UNDER

GRANT NAG 5-714

Polytechnic University

Department of Electrical Engineering and Computer Science

Brooklyn, NY   11201

Final Report

to

Goddard Space Flight Center

on

Frequency Selective Reflection and Transmission

at a

Layer Composed of a Periodic Dielectric

Prepared Under

by

Henry L. Bertoni

Li-hsiang S. Cheo

Theodor Tamir

Oct. 1, 1985 - Oct. 1, 1986

Abstract

This report examines the feasibility of using a periodic dielectric layer, composed of alternating bars having dielectric constants ε_1 and ε_2, as a frequency selective sub-reflector in order to permit feed separation in large aperture reflecting antenna systems. For oblique incidence, it is found that total transmission and total reflection can be obtained at different frequencies for proper choice of ε_1, ε_2 and the geometric parameters. The frequencies of total reflection and transmission can be estimated form wave phenomena occurring in a layer of uniform dielectric constant equal to the average for the periodic layers. About some of the frequencies of total transmission, the bandwidth for 90% transmission is found to be 40%. However, the bandwidth for 90% reflection is always found to be much narrower; the greatest value found being 2.5%.

I. Introduction

Separation of the feed structures for different frequency bands in large reflecting antennas has been achieved using sub-reflectors whose transmission and reflection coefficients are frequency dependent. For frequencies in one band, the sub-reflector acts as a perfect reflector, while for frequencies in another distinct band the sub-reflector is transparent to the radiation, thus permitting direct illumination of the main reflector by the feed. To date, periodic arrays of conducting plates, or apertures in a conductive screen, have been used as the frequency selective surface [1-4]. Typically, the conductors are placed on a dielectric layer that provides mechanical support.

Use of a dielectric layer with periodically varying dielectric constant has been suggested as an alternative way to obtain a frequency selective surface. As considered here, the layer is composed of alternating strips of two materials having different dielectric constants, as shown in Figure 1. At mm frequencies, such dielectric layers offer the advantage of low absorption loss as compared to metallic screens. Since the layer thickness is on the order of a wavelength, the amount of material required would not be excessive at these high frequencies.

This report describes a theoretical study of frequency selective reflection and transmission at dielectric layers of the type shown in Figure 1. Because this effort was intended as a limited feasibility study, we consider the case when the plane is incident perpendicular to the strips, and assume the electric field to be polarized along the strips, as in Figure 1. The layer is found to exhibit the desired frequency selective properties. It is also found that the approximate
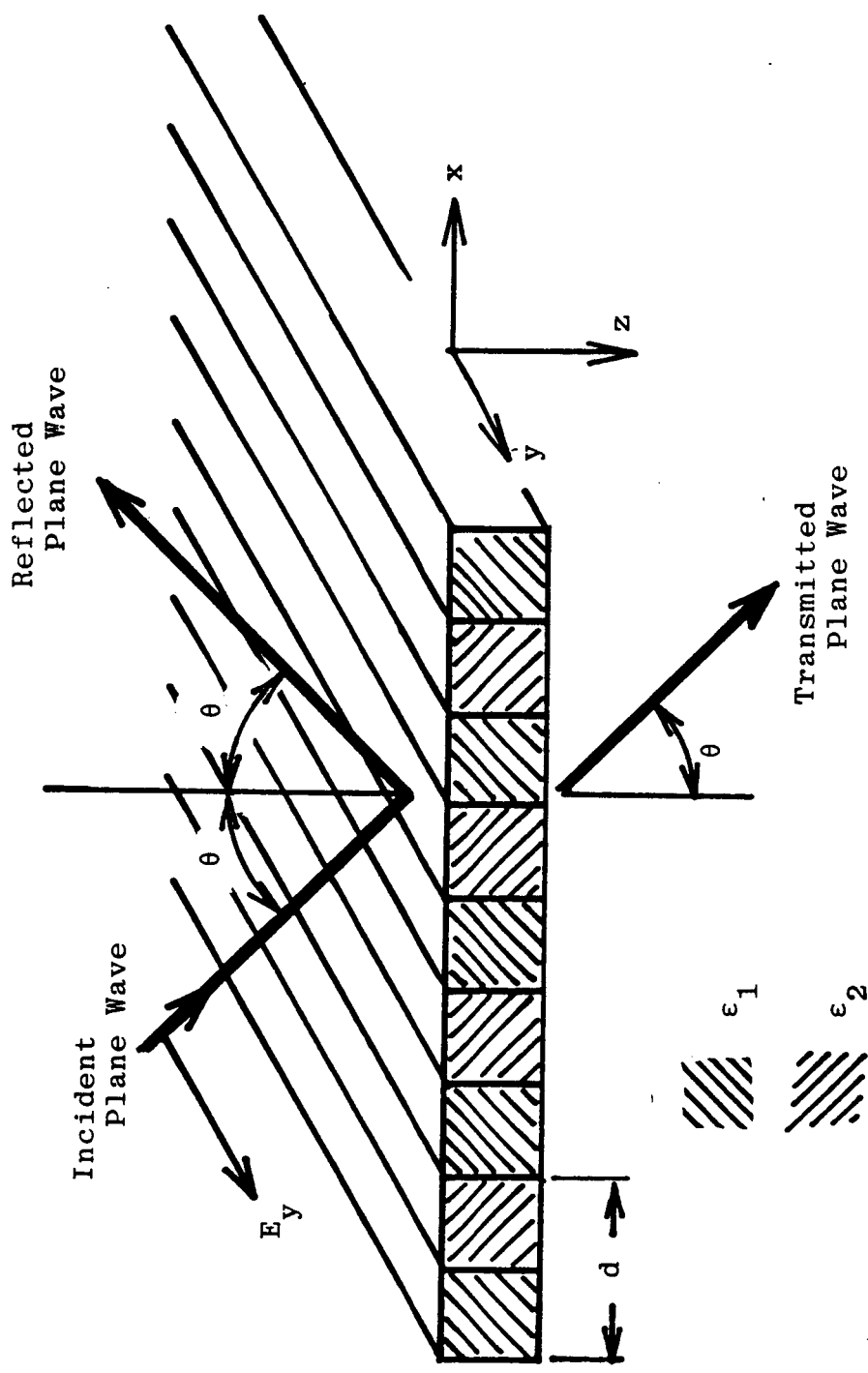
1

Figure 1. Frequency selective surface composed of a periodic array of dielectric strips.

locations of reflection and transmission bands can be predicted from wave properties of a layer having uniform dielectric constant equal to the average of that in the periodic layer.

The significant wave properties are discussed qualitatively in Section II. In Section III, the necessary mathematical analysis is carried out to permit numerical evaluation of the reflection and transmission coefficients. Numerical results are presented in Section IV.

II. Wave Mechanisms for Frequency Selective Behavior

In this section, we describe the wave phenomena that can provide frequency selective reflection and transmission at a periodic dielectric layer. The description given here is intended to clarify the nature of the subsequent analysis, and to provide a context for discussing the numerical results that have been obtained.

Consider first a dielectric that is periodic along x but infinite along z, as shown in Figure 2. For two dimensional propagation in the (x,z) plane, the dielectric will support an infinite set of modes with different wavenumbers, κ_n along z, but each having the same Bloch wavenumber k_x along x [5,6]. At low frequencies, only the lowest n = 0 mode will have a real wavenumber κ_0, while all other modes will be cut off (κ_n imaginary or complex). At somewhat higher frequencies, the n = -1 mode will also propagate (κ_{-1} real), while higher modes remain cut off. Further increase in frequency will result in more propagating modes.

Consider now a semi-infinite, periodic dielectric illuminated by a plan wave incident from vacuum, as shown in Figure 3. The incident wave will excite all of the modes of the periodic structure. At a low enough frequency f_1, only the n = 0 mode will propagate along z, as suggested in Figure 3a. Higher modes will decay exponentially away from the surface z = 0. However, at a higher frequency f_2, the n = 0 and n = -1 modes can propagate. If ε_1 and ε_2 are not close to the dielectric constant of free space, two modes can propagate in the dielectric even for the periodicity d small enough compared to the free space wavelength λ_0 so that no grating lobes are present in the field reflected into the region z < 0. In this case, the reflected field propagates only at the specular angle, as
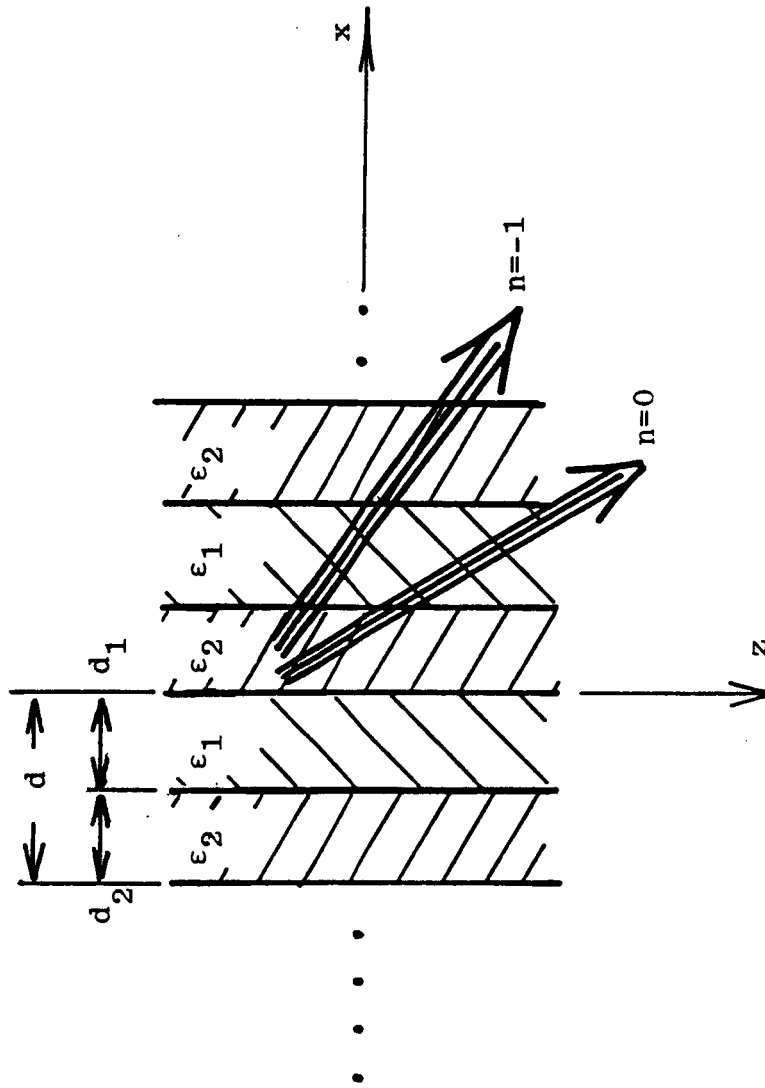
Figure 2. Periodic dielectric of infinite extent along z showing two of the infinite set of Bloch waves. The dielectric and fields are assumed to have no variation along y (out of the plane of the paper).

Figure 3. Excitation of Bloch waves at an air-dielectric interface by an incident plane wave for the cases of: a) low frequency f_1 at which only the $n=0$ Bloch wave propagates along z; and b) higher frequency f_2 at which the $n=0$ and $n=-1$ Bloch waves propagate along z.

6

indicated in Figure 3b. If the periodic dielectric is of finite thickness h, as in Figure 4, the propagating modes will excite a plane wave in the vacuum region z > h below the dielectric.

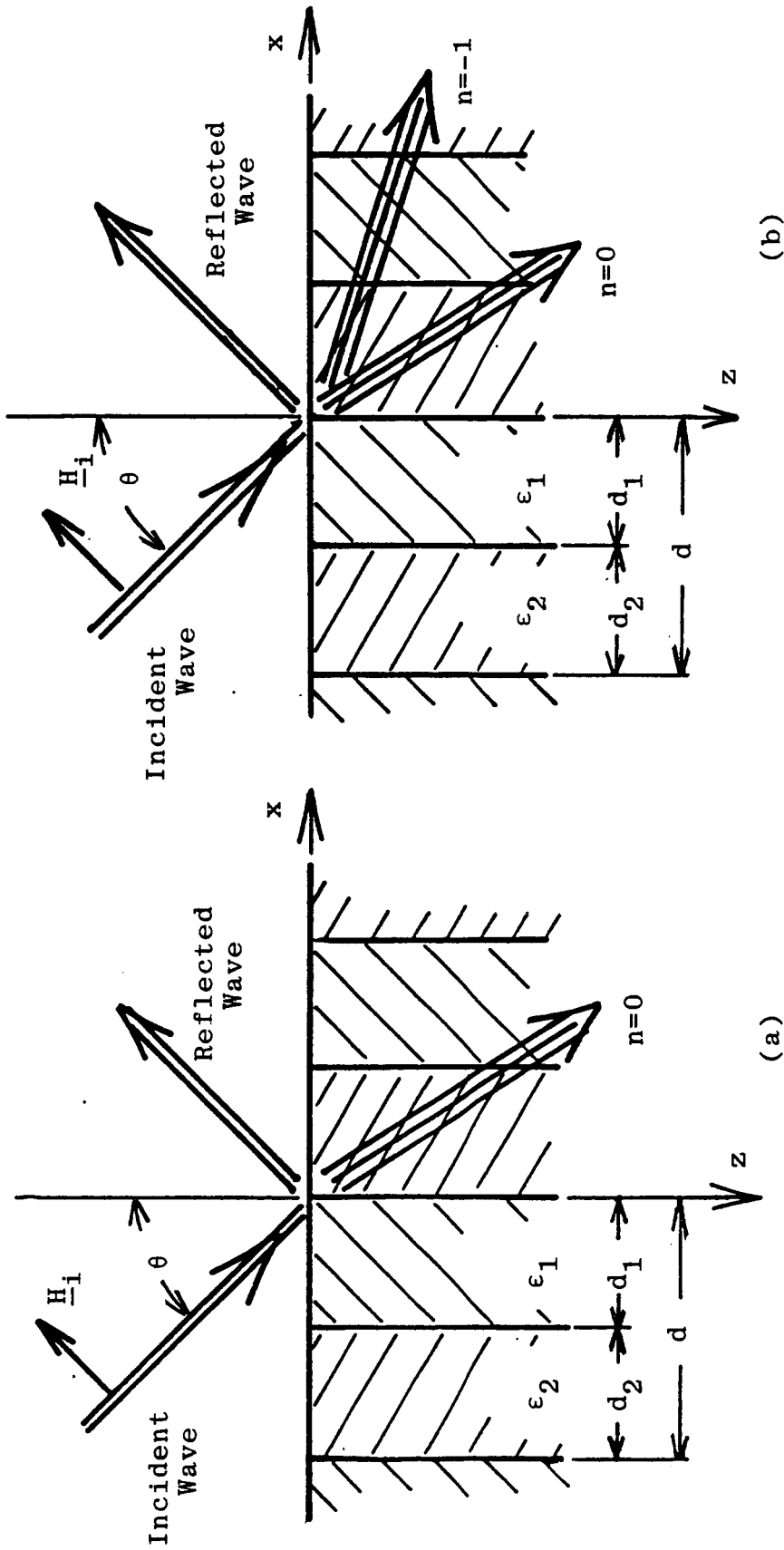At low frequencies f_1, only one mode propagates along z with real wavenumber κ_0, so that the layer acts approximately as if it had a uniform dielectric constant equal to the average of that for the periodic layer. Thus the transmission properties will be similar to those of a uniform layer. In particular, the reflection coefficient will vanish at about the frequency for which $\kappa_0 h = \pi$. In this case, the dielectric acts as a half-wave window, and there will be total transmission of the incident plane wave, as suggested in Figure 4a.

At a higher frequency f_2, both the n = 0 and n = -1 modes will propagate along z. These modes are excited at the top surface of the layer by the incident wave. When each mode reaches the bottom surface, it excites both modes traveling back to the top, as well as a transmitted plane wave in the air. Because of the phase-matching conditions at the top and bottom surfaces, the transmitted plane wave in the air propagates in the same direction as the incident plane wave.

The modes in the layer that are traveling back towards the top surface excite a reflected plane wave in the air above the layer, as well as being scattered back into the layer, as suggested in Figure 4b. Repetition of this scattering process establishes the total field in the layer, and the total reflected and transmitted plane waves in the air. For some frequency f_2, the phases of the two modes in the layer will be such as to add destructively in producing the transmitted plane wave, while constructively adding for the reflected plane wave, thereby produc-
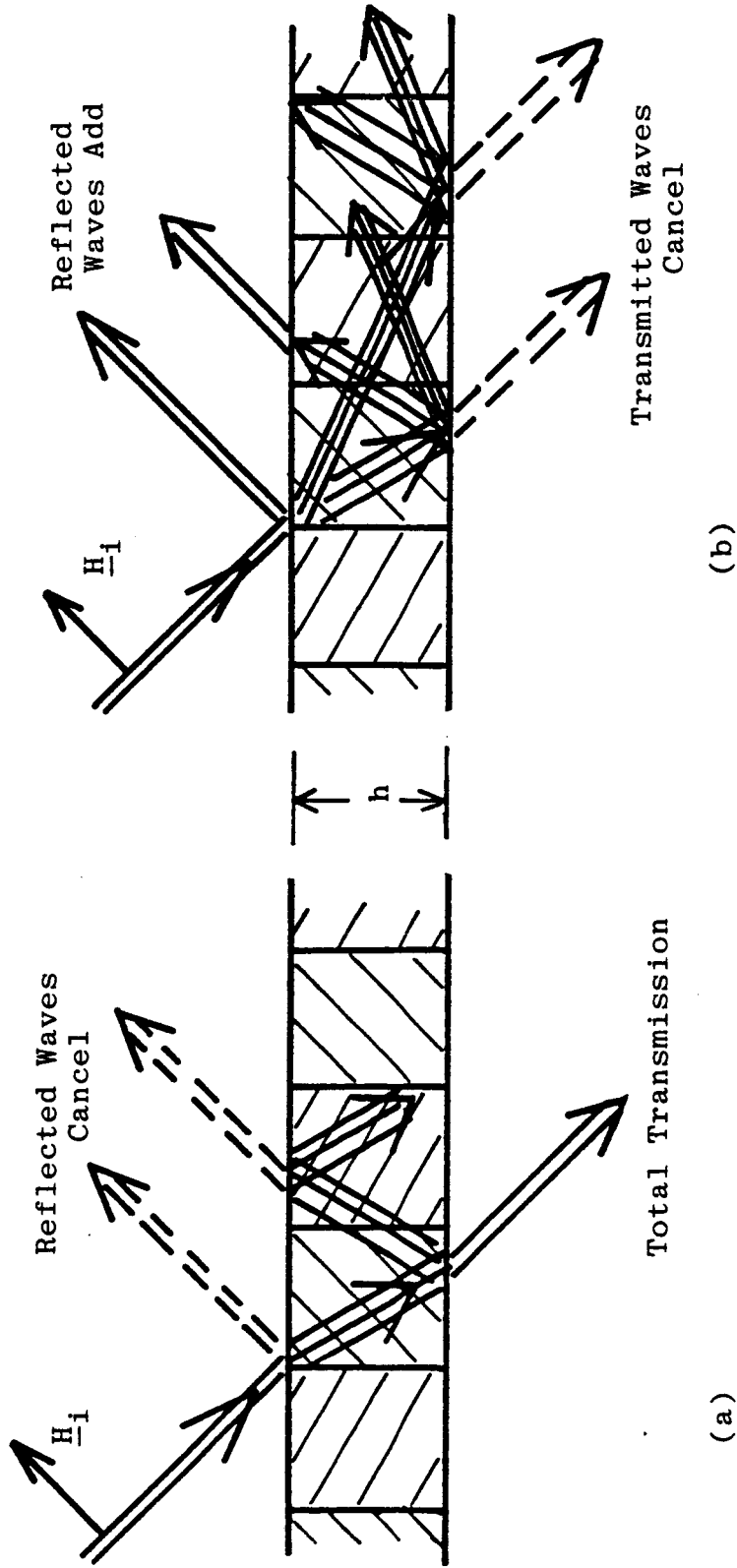
Figure 4. Wave mechanism proposed for achieving frequency selective reflection at a periodically varying dielectric layer by: a) taking $h = \pi/\kappa_o$, so that the layer acts as a half-wave window at the low frequency f_1; κ_o, and b) chosing dielectric and geometric parameters to achieve cancellation of the transmitted wave at a higher frequency f_2.

8

ing the desired frequency selective property. However, at other frequencies in the range where the n = 0 and n = -1 modes propagate along z, the phases of these two modes may be such as to add for the transmitted plane wave and cancel for the reflected wave. Thus it is possible to have multiple frequencies for which total reflection and total transmission take place.

The foregoing behavior is known in other related geometries to be associated with the excitation of waves guided along the layer [7,8]. To understand the connection with the guided waves, consider a layer of uniform dielectric constant equal to an average permittivity of the periodic layer defined by

$$\varepsilon_a = (\varepsilon_1 d_1 + \varepsilon_2 d_2)/d. \tag{1}$$

This layer will support guided waves whose fields vary sinusoidally in the layer, and decay away from the layer in the air [9]. For TE guided wave modes, the normalized wavenumber $\beta_g h$ with g=0,1,2 is plotted along the horizontal axis in Figure 5 versus the normalized free space wavenumber $k_o h = \omega h/c$, which is plotted along the vertical axis for $\varepsilon_a = 2$.

Because β_g is greater than κ_0, there are no angles of incidence θ at which a plane wave can satisfy the phase-match condition $k_o \sin\theta = \beta_g$ for direct excitation of the waveguide modes. However, excitation is possible if the dielectric constant of the layer is a periodic function of x. In this case, the fields of each waveguide mode will consist of a series of space harmonics, one of which has fields that are very similar to those of a mode in a uniform layer having the same average dielectric constant ε_a. This space harmonic is designated m = 0, and has wavenumber
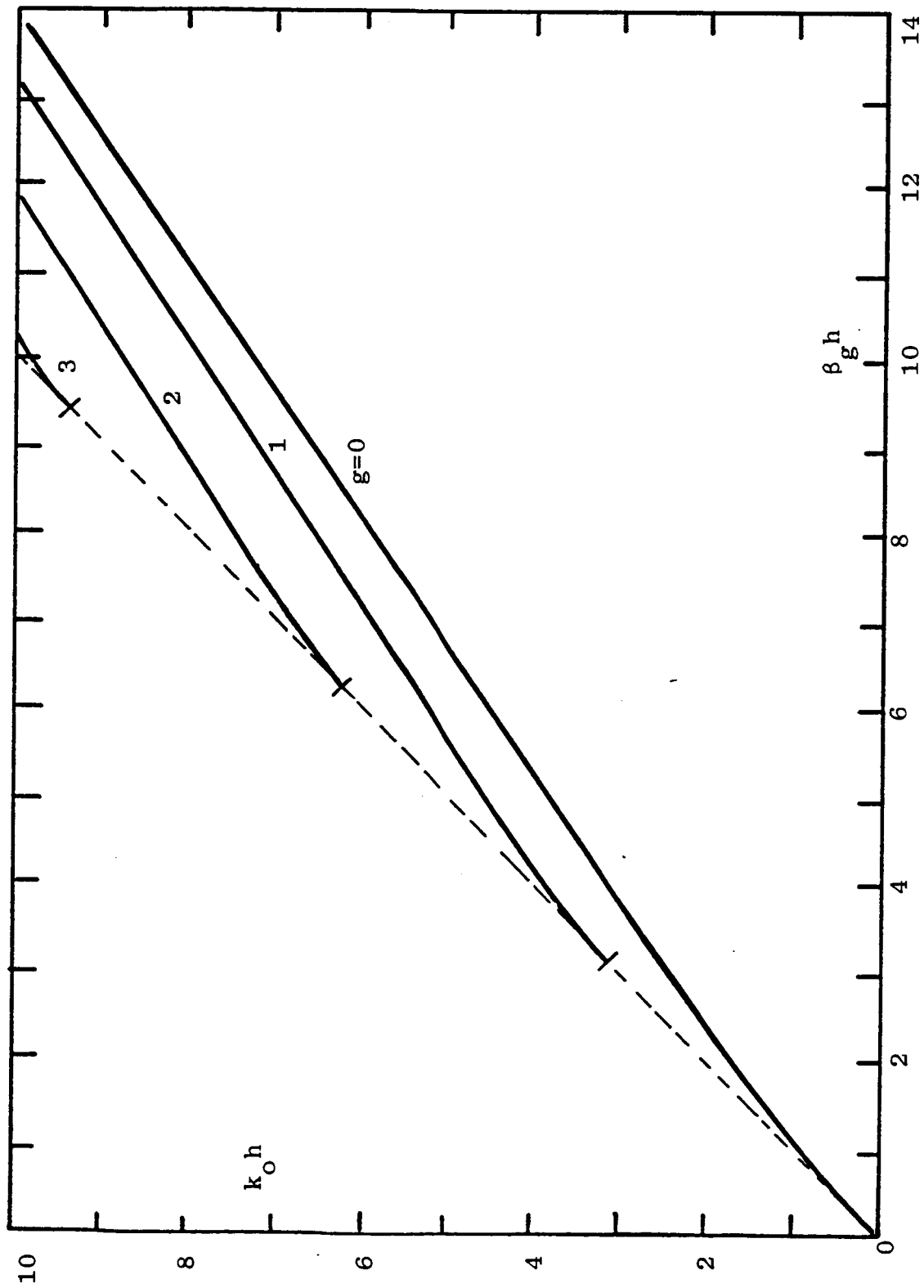
9

Figure 5. Dispersion curves for the g=0,1,2,3 modes guided by a uniform layer of relative dielectric constant 2.

β_{g0} along x that is close to β_g for the uniform layer. Other space harmonics have wavenumbers along x given by

:

$$\beta_{gq} = \beta_{g0} + q2\pi/d, \qquad\qquad (q = \pm1, \pm2, \ldots). \qquad\qquad (2)$$

While $\beta_{g0} \simeq \beta_g > k_o$, it is possible to choose the periodicity d such that $|\beta_{g,-1}| < k_o$ for m = -1. In this case, a plane wave incident at an angle $\theta = \sin^{-1}(|\beta_{g,-1}|/k_o)$ will couple to the space harmonic, and through it excite the waveguide mode. Once excited, this mode will re-radiate plane waves into the air regions above and below the layer through the same space harmonic. The process of excitation and re-radiation is depicted in Figure 6a for the case when $k_o + \beta_g > 2\pi/d > \beta_g$. This condition is sufficient to guarantee that only one space harmonic will give rise to a plane wave propagating away from the layer, and also implies that the plane wave propagates backward with respect to the direction of the waveguide mode. Guided waves that radiate some of their energy as they propagate are known as leaky waves[10].

The same physical processes hold if the incident wave is from the left, as shown in Figure 6b, except that the direction of propagation along x is reversed for the waveguide mode. The re-radiated plane wave above the layer adds to the reflected plane wave generated directly at the top surface of the layer to give the total reflected field. When the two components are in phase, strong reflections take place. However, when they are out of phase the reflected field is small and strong transmission occurs. Because the phases are frequency dependent, the overall reflection can have the desired frequency selective behavior. In

Re-Radiated
Reflected Wave

Directly Reflected
Wave

Incident
Plane Wave

x

z

Guided Wave

Re-Radiated
Transmitted Wave

(a)

Incident
Plane Wave

Total Reflected
Wave

x

z

Guided Wave

Transmitted
Wave

(b)
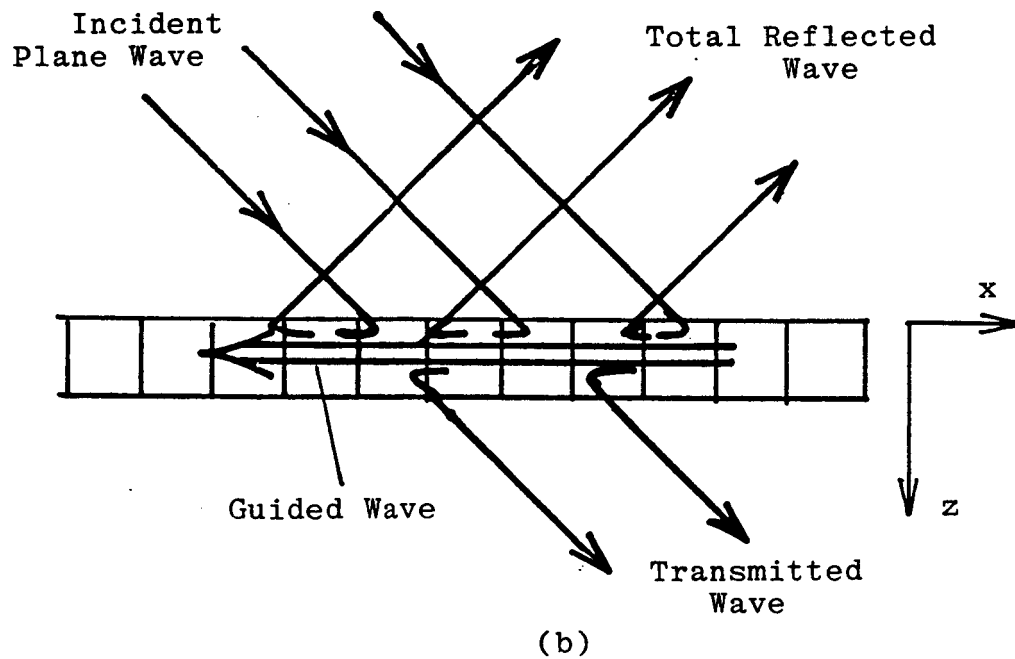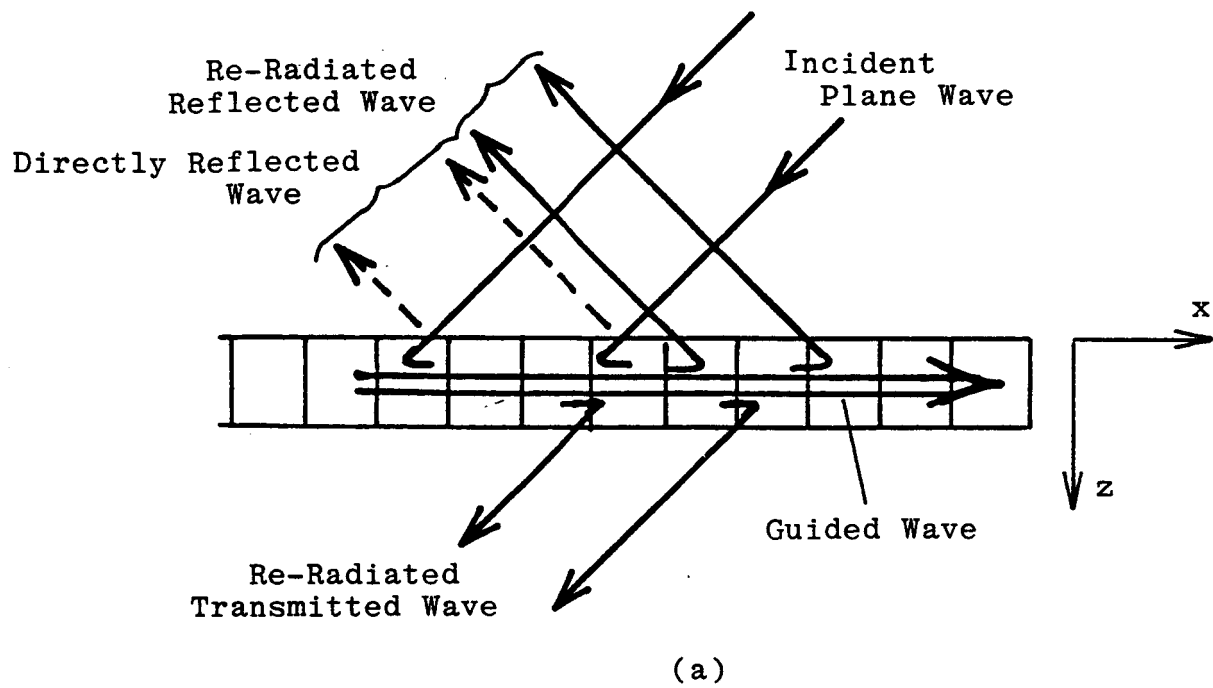
Figure 6.   Plane wave excitation and re-radiation of the wave
            guided by a dielectric layer with periodically
            modulated dielectric constant for the case: a) plane
            wave incident from the right; and b) plane wave
            incident from the left.

Section IV, it is shown that total reflection can be achieved, and that the frequencies of total reflection can be predicted from the properties of the leaky wave modes.

III. Formulation of the Reflection Problem

In this section we develop the mathematical formalism for the re-
flection and transmission coefficients at a layer of a periodic dielec-
tric in a way that can be implemented on a computer for numerical evalua-
tion. The analysis is restricted to the case of TE polarization ($E_x = E_z$
$= 0$) but it can be readily extended to the TM polarization. Expressions
for the fields in the infinite periodic medium of Figure 2 are first
derived. Subsequently the scattering matrix for a single surface normal
to z, as shown in Figure 3, is found. Finally, network concepts are used
to join the scattering matrices for the two surfaces normal to z that are
shown in Figure 4, so as to obtain the reflection and transmission coef-
ficients for the layer.

A. Fields in an Infinite Periodic Medium

In studying the fields in the infinite periodic medium we use the
approach of Collin [11] and Lewis and Hessel [12] which expresses the
fields as a superposition of modes having different wavenumbers along z.
Assuming a time dependence $\exp(-i\omega t)$, the fields of the n-th mode have
dependence along z given by $\exp(\pm i\kappa_n z)$. Thus the only non-zero component
of electric field E_y and the z-component of magnetic field H_z can be
written as the sum of modal fields in the form

$$E_y(x,z) = \sum_{n=-\infty}^{\infty} [A_n \exp(i\kappa_n z) + B_n \exp(-i\kappa_n z)] e_n(x),$$

$$(3)$$

$$H_z(x,z) = \sum_{n=-\infty}^{\infty} [A_n \exp(i\kappa_n z) - B_n \exp(-i\kappa_n z)] h_n(x).$$

Here $e_n(x)$ and $h_n(x)$ are the x-dependent mode functions while A_n and B_n
are the modal amplitudes for waves propagating in the +z and -z direc-

14

tions, respectively.

Because each slab of dielectric has a homogeneous ε, the mode functions can be expressed in trigonometric form. We first define the wavenumbers along z in the two dielectrics as

$$u_n = (k_1^2 - \kappa_n^2)^{1/2},$$
$$v_n = (k_2^2 - \kappa_n^2)^{1/2},$$

(4)

where $k_1^2 = k_o^2 \varepsilon_1$ and $k_2^2 = k_o^2 \varepsilon_2$. We further define the impedances and admittances for the dielectrics by

$$Z_{1n} = 1/Y_{1n} = \omega\mu_o/u_n,$$
$$Z_{2n} = 1/Y_{2n} = \omega\mu_o/v_n.$$

(5)

Referring to the coordinate system in Figure 2, the mode functions for $-d_1, < z < 0$ are given by

$$e_n(x) = V_n \cos(u_n x) + i\, Z_{1n} I_n \sin(u_n x),$$
$$h_n(x) = I_n \cos(u_n x) + i\, Y_{1n} V_n \sin(u_n x).$$

(6)

In the range $0 < z < d_2$, the mode functions are

$$e_n(x) = V_n \cos(v_n x) + i\, Z_{2n} I_n \sin(v_n x),$$
$$h_n(x) = I_n \cos(v_n x) + i\, Y_{2n} V_n \sin(v_n x).$$

(7)

The constants V_n and I_n in (6) and (7) are defined using the Floquet condition discussed below.

The Floquet condition requires that the fields at one end of a period ($z = -d_1$) differ from those at the other end ($z = d_2$) by at most a phase factor. Writing the phase factor as $\exp(iS_0 d)$, the Floquet condi-

tion is

$$e_n(d_2) = e_n(-d_1) \exp(iS_0d),$$

$$h_n(d_2) = h_n(-d_1) \exp(iS_0d). \tag{8}$$

Substituting from (5), (6) and (7) into (8) gives two homogeneous equations in the two unknowns V_n and I_n. In order to have a non-trivial solution of these equations, it is necessary for u_n and v_n to satisfy the secular equation

$$\cos S_0d = \cos(u_nd_1) \cos(v_nd_2)$$

$$+ [(u_n/v_n) + (v_n/u_n)] \sin(u_nd_1) \sin(v_nd_2). \tag{9}$$

Since u_n and v_n are functions of κ_n, (9) can be viewed as a relation between κ_n and S_0. As will be seen later, $S_0 = k_o \sin\theta$ where θ is the angle of incidence in Figure 1. Thus (9) serves as an equation whose roots are the allowed values of κ_n, and represents the dispersion equation of the periodic medium. For large $|n|$, the roots are well approximated by

$$\kappa_n \simeq [k_o^2\varepsilon_a - (S_0+n2\pi/d)^2]^{\frac{1}{2}}, \tag{10}$$

which holds even for relatively small values of $|n|$. From expression (10), it is seen that only for small values of $|n|$ will κ_n be real, whereas higher-order solutions will be below cutoff along z.

When the dispersion equation (9) is satisfied, the ratio I_n/V_n can be determined form (8). After some manipulation, it is found that

$$\frac{I_n}{V_n} = i \frac{\cos(v_nd_2) - \exp(iS_0d) \cos(u_nd_1)}{Z_{2n} \sin(v_nd_2) + Z_{1n} \exp(iS_0d) \sin(u_nd_1)}, \tag{11}$$

whereas I_n can be found from (11) if V_n is known. The value of V_n is

itself arbitrary and is usually obtained by normalizing the mode functions. In this analysis, we use the normalization

$$\int_{-d_1}^{d_2} |e(x)|^2 \, dx = d. \tag{12}$$

This normalization is carried out numerically during the computations, as described subsequently.

Because of the Floquet condition, the mode functions $e_n(x)$ and $h_n(x)$ are periodic functions of x multiplied by the phase factor $\exp(iS_0 x)$. Thus we may write $e_n(x)$ as the Fourier sum

$$e_n(x) = \sum_{q=-\infty}^{\infty} a_{nq} \exp(iS_q x), \tag{13}$$

where

$$S_q = S_0 + q2\pi/d. \tag{14}$$

Alternatively, the expansion coefficients a_{nq} can be found from the integral

$$a_{nq} = \frac{1}{d} \int_{-d_1}^{d_2} e_n(x) \exp(-iS_q x) \, dx. \tag{15}$$

Substituting (6) and (7) into (15), we obtain after much manipulation that

$$a_{nq} = V_n [J_{nq}^{(1)} - J_{nq}^{(2)}], \tag{16}$$

where

$$
\begin{aligned}
J_{nq}^{(1)} = \ & i \, \{ \, (S_q + Y_n)[\exp(iS_q d_1)\cos(u_n d_1) - 1] \\
& + [u_n + (Y_n S_q/u_n)]\exp(iS_q d_1)\sin(u_n d_1)\} \, /(u_n^2 - S_q^2),
\end{aligned}
\tag{17}
$$

with

$$Y_n = \frac{iu_n v_n \, [\cos(v_n d_2) - \exp(iS_0 d)\cos(u_n d_1)]}{u_n \sin(v_n d_2) + v_n \exp(iS_o d)\sin(u_n d_1)}. \tag{18}$$

The quantity $J_{nq}^{(2)}$ in (16) is of the same form as that of $J_{nq}^{(1)}$ with u_n replaced by v_n, and d_1 replaced by $-d_2$ in (17).

The normalization (12) is equivalent to requiring

$$\sum_{q=-\infty}^{\infty} |a_{nq}|^2 = 1. \tag{19}$$

From (16) it is seen that the normalization condition (19) implies

$$V_n = \{ \sum_{q=-\infty}^{\infty} |J_{nq}^{(1)} - J_{nq}^{(2)}|^2 \}^{-\frac{1}{2}}. \tag{20}$$

Substituting expression (13) into (3) for $E_y(x,z)$ and changing the order of summation gives

$$E_y(x,z) = \sum_{q=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [A_n \exp(i\kappa_n z) + B_n \exp(-i\kappa_n z)] \, a_{qn} \exp(iS_q x), \tag{21}$$

When applying boundary conditions at the surface z=0 in Figure 2, it is necessary to consider the x component of magnetic intensity $H_x(x,z)$. For the TE polarization, and using the Maxwell curl equations, it is readily seen that H_x can be found from the derivative with respect to z of E_y. The resulting expression is

$$-i\omega\mu_0 \, H_x(x.z)$$

$$= \sum_{q=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \kappa_n [A_n \exp(i\kappa_n z) - B_n \exp(-i\kappa_n z)] \, a_{qn} \exp(iS_q x). \tag{22}$$

18

Expressions (21) and (22) are used in the next section to find the scattering matrix for a single interface $z = 0$.

B. Scattering at a Single Interface

A single interface at $z=0$ is depicted in Figure 3. The field in the air region $z < 0$ consists of an incident plane wave propagating at an angle θ with respect to the z axis, and reflected plane waves corresponding to the specular and higher space harmonics. We define

$$S_0 = k_0 \sin \theta,$$
$$C_0 = k_0 \cos \theta,$$

(23)

and assume the incident electric field to be polarized along y with amplitude E_0. The electric field of the incident wave is then given by

$$E_0 \exp (iS_0 x) \exp(iC_0 z).$$

(24)

The electric field due to the reflected wave is the sum of the space harmonics and takes the form

$$\sum_{q=-\infty}^{\infty} R_q \exp(iS_q x) \exp(-iC_q z),$$

(25)

where Sq is given by (14) and

$$C_q = (k_0^2 - S_q^2)^{\frac{1}{2}}.$$

(26)

For the conditions of interest here $S_q^2 \geq k_0^2$ for all $q \neq 0$ so that only the fields of the specular ($q = 0$) space harmonic propagate away from the interface, while the fields of all other space harmonics decay. The amplitude coefficients R_q have yet to be determined.

From (24) and (25) the total electric field in the air region is

seen to be

$$E_y(x,z) = \sum_{q=-\infty}^{\infty} [\delta_{q0} E_o \exp(iC_oz) + R_q \exp(-iC_qz)] \exp(iS_qx), \qquad (27)$$

where δ_{q0} is the Kronecker delta. The x component of magnetic intensity in the air can be found from the derivative with respect to z of (27), which yields

$$-i\omega\mu_o H_x(x,y)$$

$$= \sum_{q=-\infty} C_q [\delta_{q0} E_o \exp(iC_oz) - R_q \exp(-iC_qz)] \exp(iS_qx). \qquad (28)$$

The boundary conditions at z=0 require that E_y and H_x be continuous there. Equating the pair (21), (27) and the pair (22), (28), and making use of the orthogonality of the functions exp (iS_qx) over a period, one obtains

$$\delta_{q0}E_o + R_q = \sum_{n=-\infty}^{\infty} [A_n + B_n] a_{qn}, \qquad (29)$$

$$C_q (\delta_{q0}E_o - R_q) = \sum_{n=-\infty}^{\infty} \kappa_n [A_n - B_n] a_{qn}. \qquad (30)$$

In (29) and (30), n ranges over all positive and negative integers, so that these equations represent two infinite sets of equations.

We wish to solve (29), (30) for the amplitudes R_q and A_n of the waves traveling away from the surface in terms of the amplitudes E_o and B_n of the incident waves. In this way, we obtain the scattering matrix of the surface. The waves incident from the periodic medium arise from reflection at the second surface, as suggested in Figure 4. For the

conditions of interest, only one or two waves are propagating in the periodic medium, while all other waves are cutoff along z. It can be shown that the two propagating waves correspond to the indices $n = -1, 0$. Since the two surfaces are separated by at least one half-wavelength, the fields of the cutoff waves excited at one surface are exponentially small at the other surface. Hence, to a good approximation we may assume $B_n = 0$ for $n \neq -1, 0$.

While all the higher modes are excited in the air and in the periodic medium, the interaction at the two surfaces and radiation into the air are described by the amplitudes R_0, A_0, A_{-1} of the propagating waves. Thus we ultimately need only the 3×3 portion of the full scattering matrix relating R_0, A_0, A_{-1} to E_0, B_0, B_{-1}. This scattering relation takes the form

$$
\begin{bmatrix} R_0 \\ A_0 \\ A_{-1} \end{bmatrix} = \begin{bmatrix} S_{1,1} & S_{1,0} & S_{1,-1} \\ S_{0,1} & S_{0,0} & S_{0,-1} \\ S_{-1,1} & S_{-1,0} & S_{-1,-1} \end{bmatrix} \begin{bmatrix} E_0 \\ B_0 \\ B_{-1} \end{bmatrix} . \tag{31}
$$

To solve for the elements $S_{\alpha,\beta}$ in the scattering relation (31), we first multiply (29) by C_q and then add (29) and (30). Since $B_n = 0$ for $n \neq -1, 0$, the resulting equation may be written in the form

$$
2C_0 \, \delta_{q0} \, E_0 + (\kappa_0 - C_q) \, a_{0q} \, B_0 + (\kappa_{-1} - C_q) \, a_{-1q} \, B_{-1}
$$

$$
= \sum_{n=-\infty}^{\infty} (\kappa_n + C_q) \, a_{qn} \, A_n . \tag{32}
$$

Expression (32) represents an infinite set of equations with index q in an infinite number of unknowns A_n.

The terms $(\kappa_n + C_q) \, a_{qn}$ can be viewed as elements of a matrix. With

21

this view, it can be shown that the elements decrease as one moves away from the main diagonal. Thus it is reasonable to solve (32) by using a finite truncation of the summation and a corresponding limitation on the number of values of q considered. For the parameters chosen in this study, it was found to be sufficient to allow q, n to range over the integers -3, -2, -1, 0, + 1, + 2. With this truncation, (32) is solved for A_n with n= -3 ... +2. Returning to (29), we then compute R_0 from

$$R_0 = - E_0 + \sum_{n=-3}^{2} A_n a_{0n} + a_{00} B_0 + a_{0,-1} B_{-1}. \tag{33}$$

Collecting the resulting values of A_0, A_{-1} and R_0 due separately to E_0, B_0 and B_{-1} gives the values of the scattering matrix in (31).

C. Scattering From A Periodic Layer

Having found the scattering matrix for a single surface (z = 0), network concepts can be used to treat the interation with a second surface at z = h. As discussed previously, the interations between the two surfaces are essentially due to the propagating modes in the periodic medium. For our case, these are the n = -1, 0 modes, which are shown in the transmission line model for the interation shown in Figure 7. At the surface z = 0 the incident waves E_0, B_0 and B_{-1} couple to the scattered waves R_0, A_0 and A_{-1}.

At the surface z = h, the incident waves in the layer are $A_0 \exp(i\kappa_0 h)$ and $A_{-1} \exp(i\kappa_{-1}h)$, while no wave is incident from the air side. In this case, the scattered waves are T_0, $B_0 \exp(-i\kappa_0 h)$ and $B_{-1} \exp(-i\kappa_{-1}h)$. The relation between scattered and incident waves is again given by (31), which for the foregoing conditions takes the form
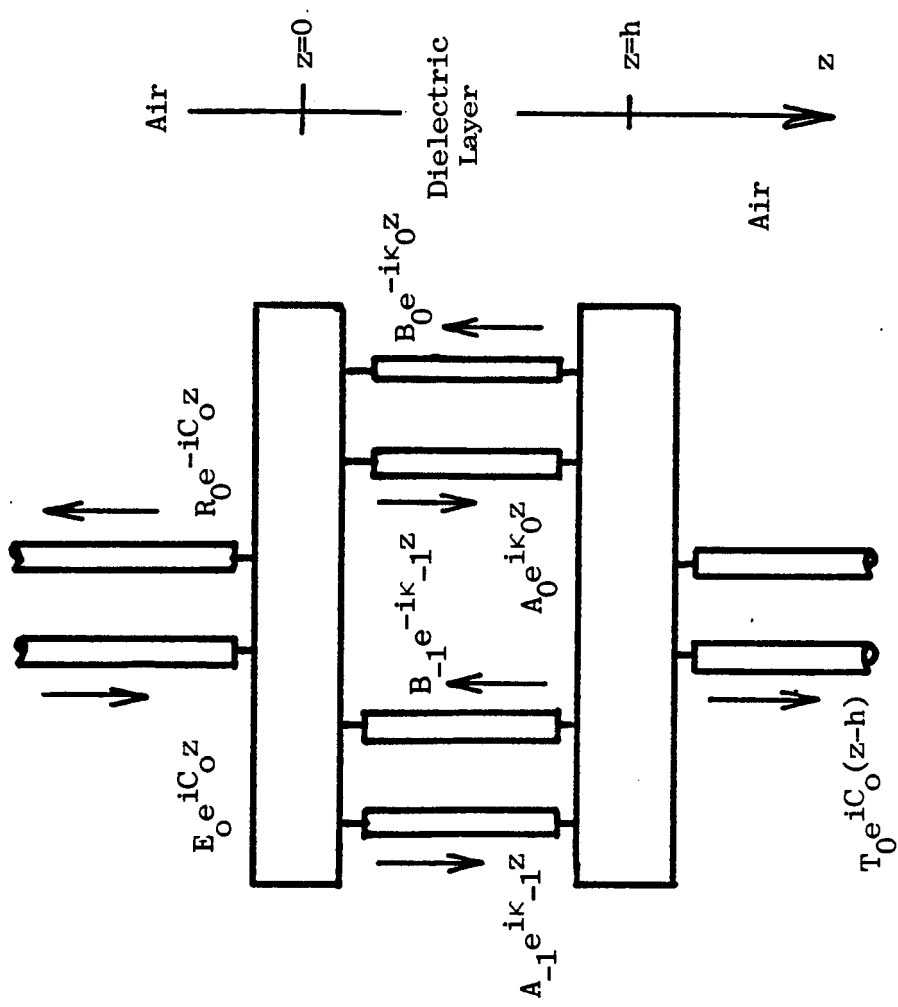
Figure 7. Microwave equivalent network for determining reflection and transmission at a periodically varying dielectric layer.

$$
\begin{bmatrix} T_0 \\ B_0\exp(-i\kappa_0 h) \\ B_{-1}\exp(-i\kappa_{-1}h) \end{bmatrix} = \begin{bmatrix} S_{1,1} & S_{1,0} & S_{1,-1} \\ S_{0,1} & S_{0,0} & S_{0,-1} \\ S_{-1,1} & S_{-1,0} & S_{-1,-1} \end{bmatrix} \begin{bmatrix} 0 \\ A_0\exp(i\kappa_0 h) \\ A_{-1}\exp(i\kappa_{-1}h) \end{bmatrix} \qquad (34)
$$

For the problem of scattering by a layer, the field E_0 is known and one wishes to solve for the reflected and transmitted wave amplitudes R_0 and T_0, respectively. To this end, (31) and (34) can be viewed as six inhomogeneous equations in six unknowns R_0, A_0, A_{-1}, B_0, B_{-1}, T_0. Assuming $E_0=1$, the solution of these equations for R_0 and T_0 give the reflection and transmission coefficients of the layer. This approach has been used as the final stage in our computer program, as discussed below.

D.   Computer Program for R_0 and T_0

A program has been written in the PL-1 language to compute R_0 and T_0 by the methods derived above. The listing of the program is given in Appendix A. An outline of the program is given below.

1.   Given input frequency ω, angle of incidence θ, geometric parameters of the layer d_1, d_2, h and the electrical parameters ε_1 and ε_2.

2.   For integers n between -3 and 2, compute κ_n from (9) using Newton's method with starting value given by (10).

3.   For integers n, q between -3 and 2 compute A_{nq} using (16)-(18) and (20).

4.   Using (32) and (33), solve for A_n $(-3 \le n \le 2)$ and R_0 for $E_0 = 1$, $B_0 = B_{-1} = 0$. This gives the elements $S_{1,1}$, $S_{0,1}$, $S_{-1,1}$ in (31). Repeat for $B_0 = 1$ with $E_0 = B_{-1} = 0$, and then for $B_{-1} = 1$ with $E_0 = B_0 = 0$ to get the remaining scattering coefficients in (31).

5.   Using (31) and (34) with $E_0 = 1$, solve for R_0 and T_0.

Several checks were carried out to ensure that the program was working properly. Choosing ε_1 and ε_2 very close to each other, we computed R_0 and T_0. As expected, they were very close to the values for a homogeneous dielectric layer of value ε_a. For values of ε_1 and ε_2 used subsequently, it was found that the scattering matrix in (31) conserves power. Finally it was observed that $|R_0|^2 + |T_0|^2$ is very close to unity, as required by power conservation.

IV. Numerical Studies of Frequency Selective Reflection

Using the computer program described previously, we have carried out numerical studies for several examples. The purpose of these studies is to gain insight into the frequency selective behavior that can be expected for R_O and T_O, and to relate this behavior to wave processes in the periodic layer. We have therefore arbitrarily chosen the angle of incidence $\theta = 45^O$ and set $d_1 = d_2 = d/2$.

For the initial studies, we have assumed $\varepsilon_1 = 2.56$ and $\varepsilon_2 = 1.44$, which are realistic values for low-loss plastics. With these choices, the average dielectric constant is $\varepsilon_a = 2$. The perodicity d must now be chosen such that, at the high frequency of interest, two modes (with n = -1, 0) propagate in the dielectric layers. Furthermore, only the specular (q = 0) space harmonic must propagate in the air.

A.   Choice of d, h and frequency

The restriction on d needed to insure that only the q = 0 space harmonic propagates in the air can easily be interpreted with the help of Figure 8a. The incident wave has wavenumber $S_O = k_O \sin \theta < k_O$ along x. Wavenumbers S_q of other space harmonics lie at a distance $q(2\pi/d)$ away from S_O, as shown for q = -2, -1 and + 1 in Figure 8a. Provided that d is small enough so that

$$k_O \sin \theta - 2\pi/d < - k_O, \tag{35}$$

the q = -1 space harmonic will lie outside the visible circle defined by $S^2 + C^2 = k_O^2$. It is further seen that, if (35) holds, all other space harmonics will also lie outside the circle, so that C_q defined by (26) is imaginary for $q \neq 0$, and the space harmonics decay away from the layer. Provided that the modulation $(\varepsilon_1 - \varepsilon_2)/\varepsilon_a$ is not too large, the concept

26
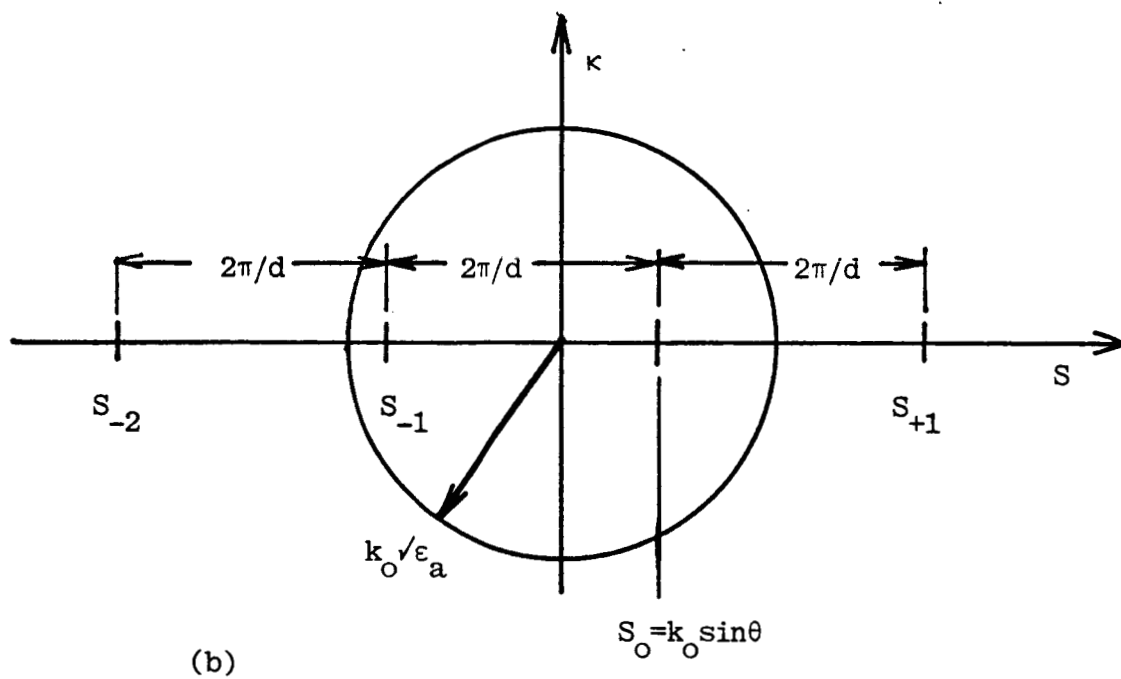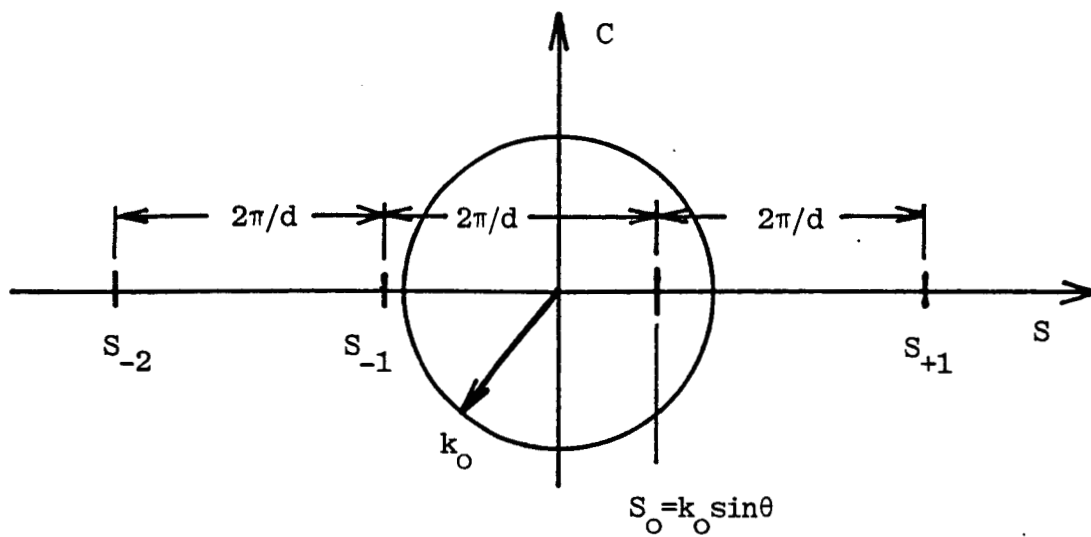
Figure 8. Vissible circle for determining the propagating space harmonics
in: a) air; and b) a medium with dielectric constant ε_a.

27

of the visible circle can also be used to estimate the number of propagating waves in the layer. When the modulation is small, (10) can be used as an estimate for κ_n, in which case the visible circle is given by $s_n^2 + \kappa_n^2 = k_0^2 \varepsilon_a$, as shown in Figure 8b. It is seen from this figure that, for the two modes n= -1, 0 to propagate in the layer, d must satisfy

$$k_0 \sin \theta - 4\pi/d < -k_0\sqrt{\varepsilon_a} < k_0 \sin \theta - 2\pi/d,$$
$$k_0\sqrt{\varepsilon_a} < k_0 \sin \theta + 2\pi/d. \tag{36}$$

Conditions (35) and (36) can be rearranged into the following inequalities:

$$d/\lambda < 1/(1 + \sin \theta),$$

$$1/(\sqrt{\varepsilon_a} + \sin \theta) < d/\lambda < 2/(\sqrt{\varepsilon_a} + \sin \theta), \tag{37}$$

$$d/\lambda < 1/(\sqrt{\varepsilon_a} - \sin \theta).$$

Assuming $\varepsilon_a = 2$ and $\theta = 45^\circ$, these inequalities are $d/\lambda < 0.586$, $0.471 < d/\lambda < 0.943$ and $d/\lambda < 1.414$, respectively. To satisfy these inequalities, we have chosen $d/\lambda = 0.54$.

Initially a value of layer thickness h at which total reflection will occur was obtained by computing R_0 for various values of h/λ. Subsequently, it was found that values of h and d for total reflection could be related via the conditions for guidance of a wave by a layer of uniform dielectric constant ε_a. Whereas our initial approach gave us the value of $h/\lambda = 0.925$ for sample calculations, it is the subsequent interpretation that is discussed below.

B. Variation of R_0 With Frequency

Computations of the frequency dependence of R_0 have been made assuming that $h = 0.925$ and $d = 0.54$ for $\varepsilon_1 = 2.56$ and $\varepsilon_2 = 1.44$. This choice produces total reflection for a frequency f such that $\lambda = c/f$ is about unity. Note that, if h and d are scaled by λ, then total reflection can be obtained at any desired frequency. The results of the calculation for $|R_0|$ are depicted in Figure 9, where we have used the normalized frequency variable $k_0 h = 2\pi f h/c$, and have plotted up to the value $k_0 h = 6.30$ at which the $q = -1$ space harmonic in air switches from cutoff to propagating along z.

For $k_0 h < 5.12$, only the $n = 0$ mode in the periodic dielectric is propagating along z. In the frequency range $0 < k_0 h < 5.12$ for single mode propagation, R_0 vanishes at the two frequencies $k_0 h = 2.56$ and 5.03, at which $\kappa_0 h = 3.145$ and 6.355. Thus, frequencies of total transmission occur when the layer thickness is close to a multiple of one half the effective wavelength along z, as predicted in Section II. The difference between the values of $\kappa_0 h$ for total transmission and π, 2π are due to the non-zero phase of the transmission and reflection coefficients at the individual surfaces $z = 0$ and h, which result from excitation of higher cutoff space harmonics.

For $k_0 h < 5.12$, where the $n = -1$ mode also propagates along z in the layer, total reflection takes place at two frequencies given by $k_0 h = 5.32$ and 5.83. In the vicinity of these frequencies for total reflection, the variation of $|R_0|$ is that associated with resonances wherein a frequency dependent function has a real axis zero and a nearby pole at a complex location.
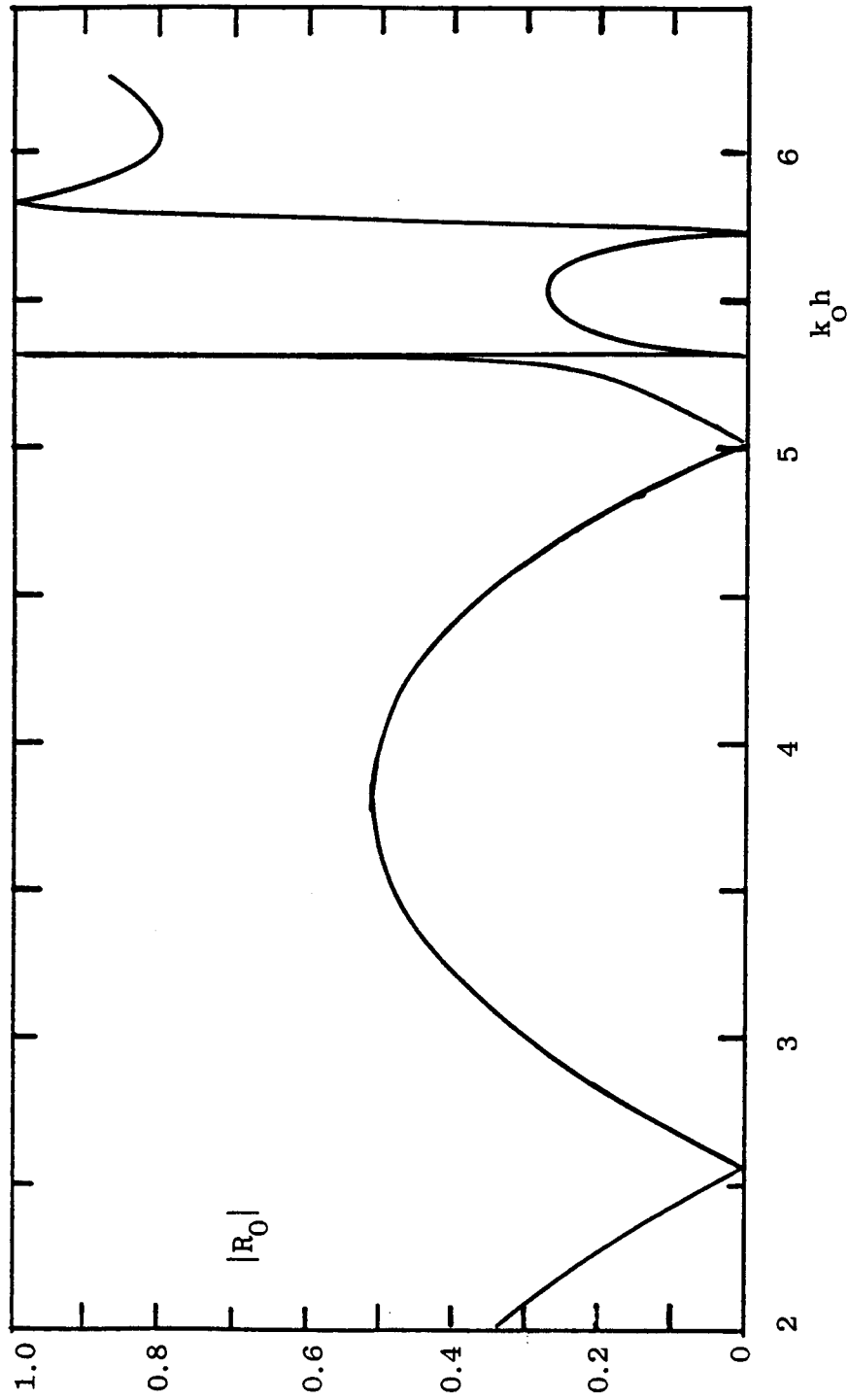
Figure 9. Variation of $|R_0|$ with normalized frequency $k_0 h$ for $\varepsilon_1 = 2.56$, $\varepsilon_2 = 1.44$ and $h/d = 0.925/0.54$.

30

To examine the bandwidth of the total reflection resonance, we have plotted $|R_0|^2$ on an expanded scale in Figure 10. Since $|R_0|^2 + |T_0|^2 = 1$, the curve can also be used to determine $|T_0|^2$ by using the vertical scale to the right of the plot. The wider of the two peaks of $|R_0|^2$ is centered at $k_0 h = 5.83$. For this peak, the fractional bandwidth between the frequencies at which $|R_0|^2 = 0.9$ is 0.86%.

The narrower of the two peaks in Figure 10 is shown further expanded in the insert. For this peak, the fractional bandwidth between the frequencies at which $|R_0|^2 = 0.9$ is less than 0.04%. By comparison, much wider bandwidths are found for total transmission when the n = -1 mode in the layer is cut off. For example, in a region about $k_0 d = 2.56$ in Figure 9, a 40% bandwidth is found between the frequencies for which $|R_0|^2 = 0.1$.

C. Prediction by Means of Guided Waves

The location of the frequencies of total reflection can be predicted from the properties of the waves guided by the periodic layer. Consider first the case of a wave guided along a uniform layer having dielectric constant ε_a. The normalized propagation constant $\beta_g h$ of this guided wave is plotted horizontally in Figure 5, versus the normalized frequency $k_0 h$, which is plotted vertically. The lowest g = 0 guided wave mode starts at the origin and becomes asymptotic to the wavenumber $k_0 \sqrt{\varepsilon_a}$ of the layer. Higher guided-wave modes start at points $k_0 h = g\pi(\varepsilon_a - 1)$ along the 45° line, where g = 1, 2, 3 .....

In Figure 11, we have repeated the plot of Figure 5, and have added the dispersion curves for waves propagating in the negative x direction ($\beta < 0$). We have also plotted as a broken line the transverse wavenumber

31
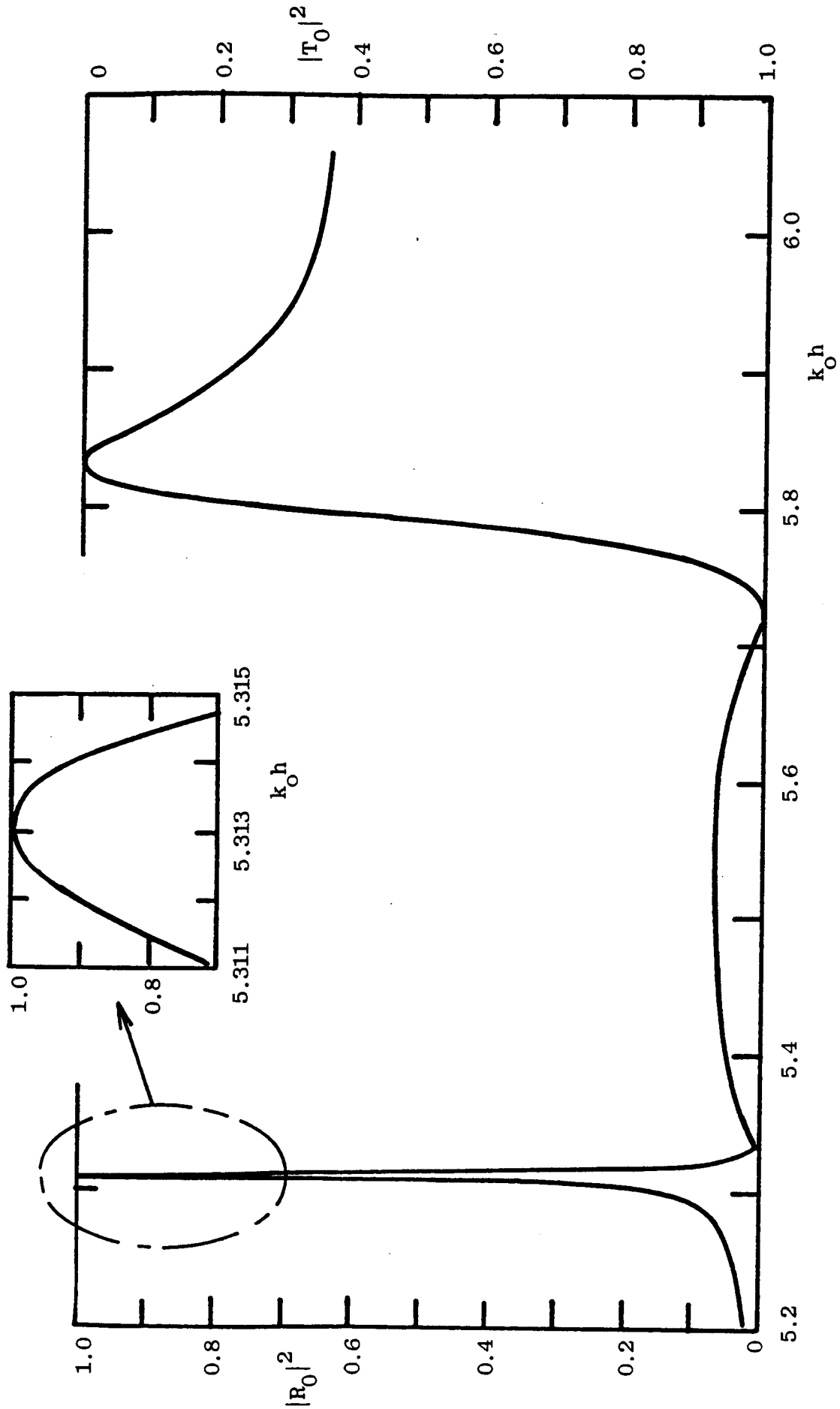
Figure 10. Variation of the power reflection coefficient $|R_0|^2$ and transmission coefficient $|T_0|^2$ with normalized frequency $k_o h$ for the parameters of Figure 9.

32
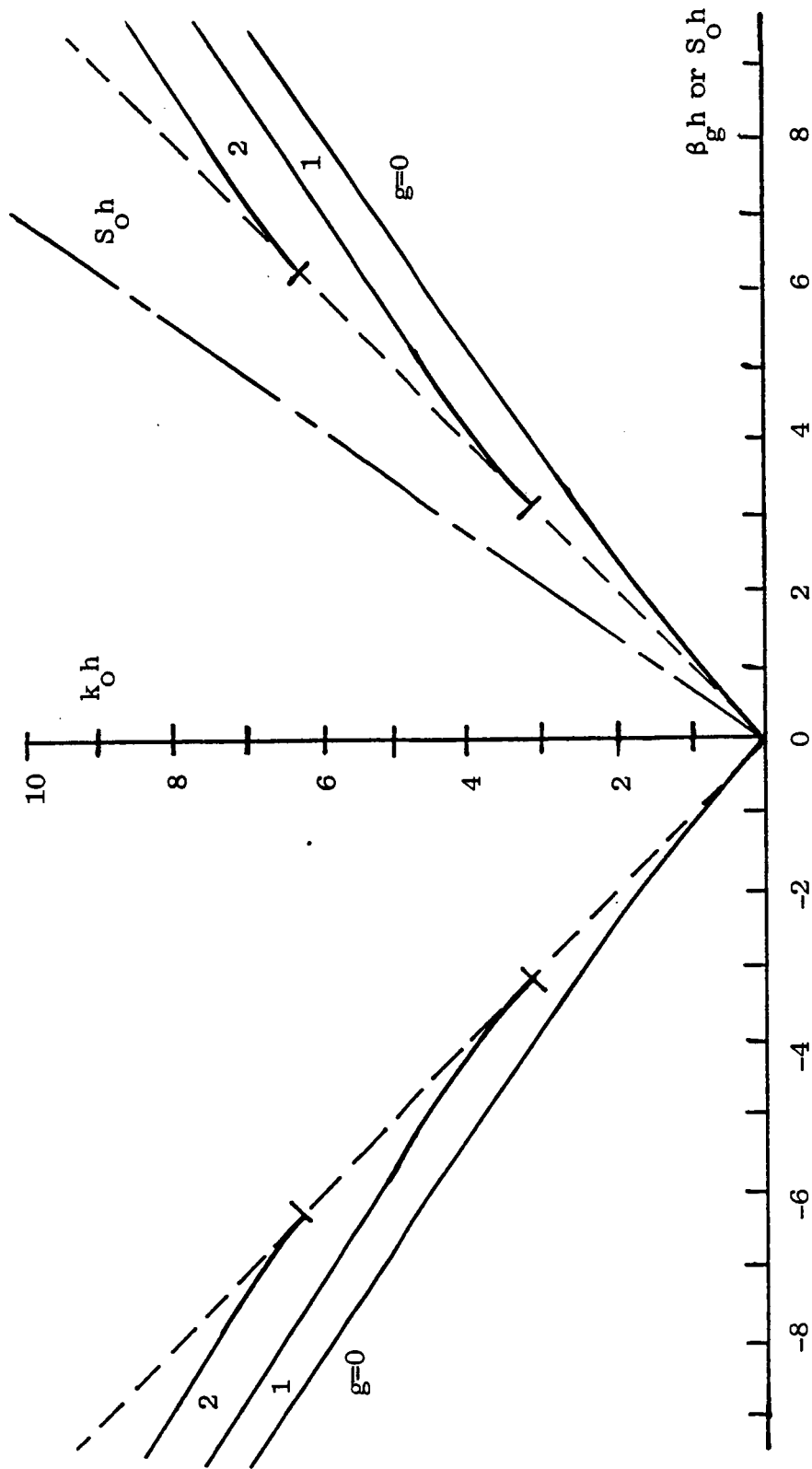
Figure 11. Dispersion curves for surface waves traveling in the ±x direction (solid curves) and the normalized transverse wavenumber $S_o h$ of the incident plane wave (broken line).

$S_oh = k_oh \sin \theta$ of the incident plane wave. Except at $k_oh = 0$, the broken line is never close to the dispersion curves for the guided waves. As a consequence, an incident plane wave cannot couple to the guided waves on a uniform layer.

If the dielectric layer is made periodic along x, the field of each guided wave mode becomes a sum of space harmonics. For the guided waves traveling in the -x direction, the wavenumber of the $q = -1$ space harmonic is $(-\beta_g + 2\pi/d)$. When normalized by h, the dispersion curve of this space harmonic has the same form as $-\beta_g h$ versus k_oh, except for a shift $2\pi h/d$ to the right. While finite modulation affects the value of β_g for the guided wave, for small modulation of the dielectric constant β_g is close to that for a uniform layer.

Dispersion curves for the $q = -1$ space harmonics of the guided waves in the small modulation limit are shown in Figure 12 for $h/d = 0.925/0.54$. We have also drawn a broken line representing S_oh versus k_oh. Intersection of the S_oh line with the dispersion curvers indicates strong coupling between an incident plane wave and guided waves through the $q = -1$ space harmonic. Note that, above the dashed line having an angle of -45°, the $q = -1$ space harmonic in the air propagates along z. Thus for reflection and transmission of a single space harmonic, the operating point along the S_oh line must be kept below the dashed line. For the parameters used in drawing Figure 12, this condition implies that $k_oh \leq 6.30$ for one propagating space harmonic in air.

In Figure 12, the line S_oh intersects the dispersion curve for the g = 0 guided wave at $k_oh = 5.27$, and for the g = 1 guided wave at $k_oh = 5.72$. These values are close to the values $k_oh = 5.32$ and 5.83 for total

34
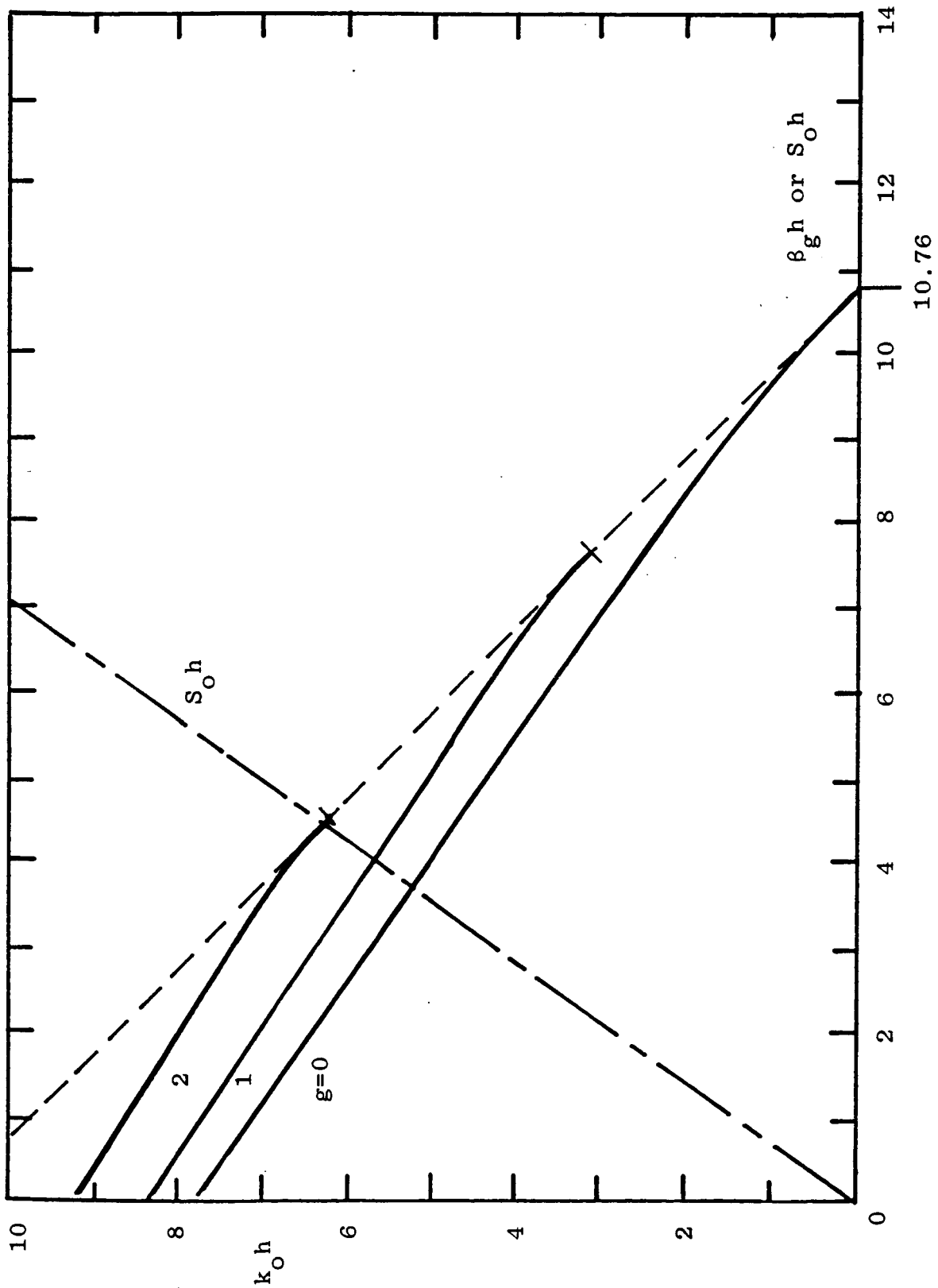
Figure 12. Transverse wavenumber S_oh of the incident plane wave and the dispersion curves of the $q=-1$ space harmonics for the $g=0,1,2$ guided waves in the limit of small modulation for $\epsilon_a=2$ and $h/d=0.925/0.54$.

reflection ($|R_0|=1$) obtained from Figure 11. The deviation between the values of k_0h obtained from Figures 11 and 12 is thought to result from the fact that the finite modulation of the dielectric constant of the layer alters β_g from the value obtained for a uniform slab. Thus, decreasing modulation should bring the values closer together, while increasing modulation should result in greater deviation. This latter condition is shown subsequently.

To further demonstrate the relation between the frequencies of total reflection and the guided waves of the layer, we have considered a layer of increased thickness h = 1.1, but the same periodicity d = 0.54. The dispersion curves of the q = -1 space harmonics of the first three guided-wave modes are shown in Figure 13 for the limiting case of small modulation. The broken line giving $S_0h = k_0h \sin \theta$ is seen to intersect the three dispersion curves at k_0h = 6.22, 6.67 and 7.30. Our model predicts that total reflection should take place at normalized frequencies close to these values.

A plot of $|R_0|$ versus k_0h for h = 1.1 and d = 0.54 is shown in Figure 14. From this plot, total reflection is seen to occur at the three frequencies k_0h = 6.25, 6.78 and 7.42, which are close to those predicted by the small modulation theory. The bandwidth over which $|R_0|^2 \geq 0.9$ about each frequency of total reflection is seen to increase as k_0h approaches the value 7.50 where the line S_0h crosses the dashed line, above which the n = -1 space harmonic propagates in air. The bandwidth about the lowest of these frequencies is only 0.01%, while that of the highest is 0.7%.
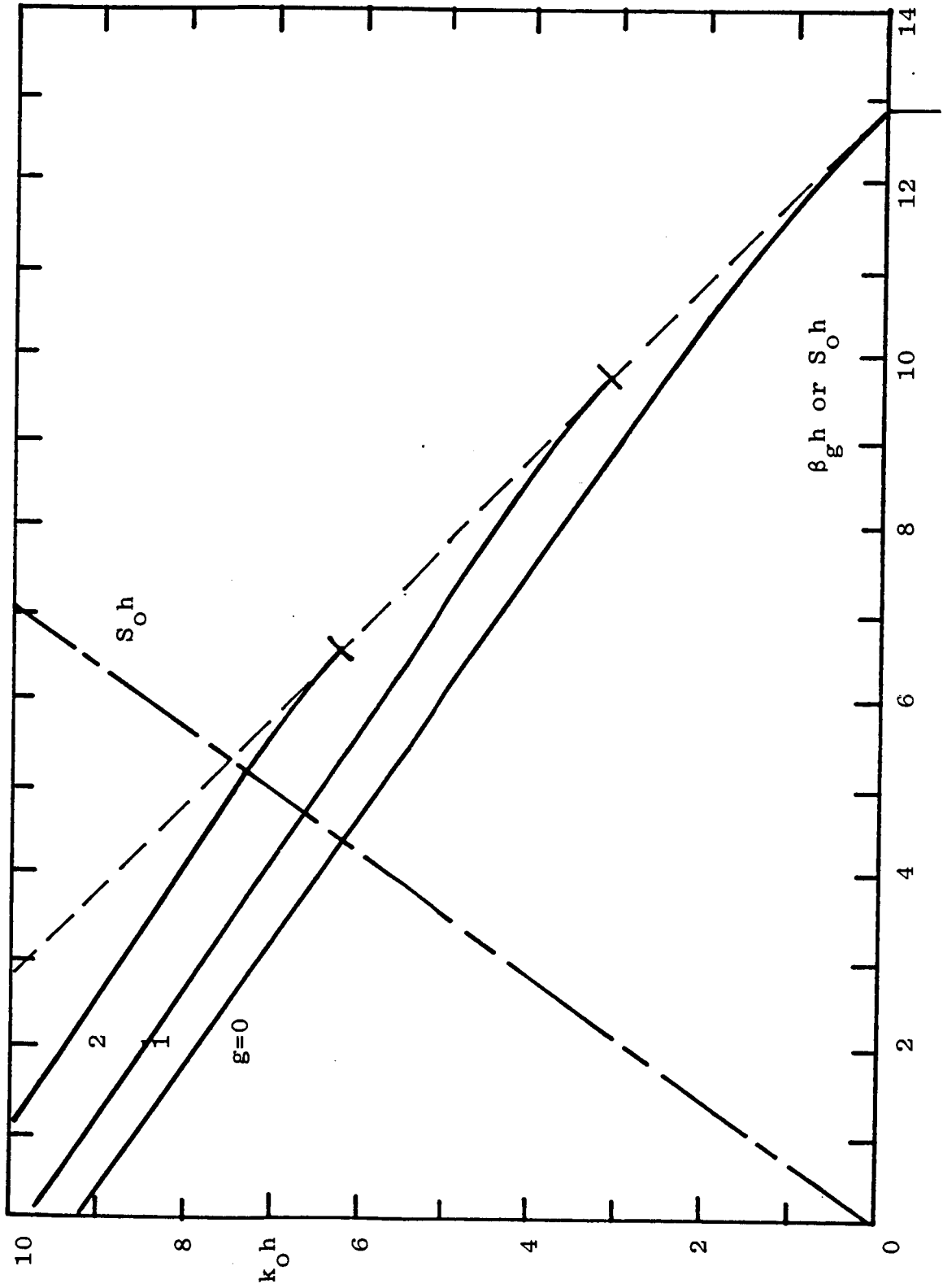
Figure 13. Transverse wavenumber $S_o h$ of the incident plane wave and the dispersion curves of the $q=-1$ space harmonics for the $g=0,1,2$ guided waves in the limit of small modulation for $\epsilon_a=2$ and $h/d=1.1/0.54$.
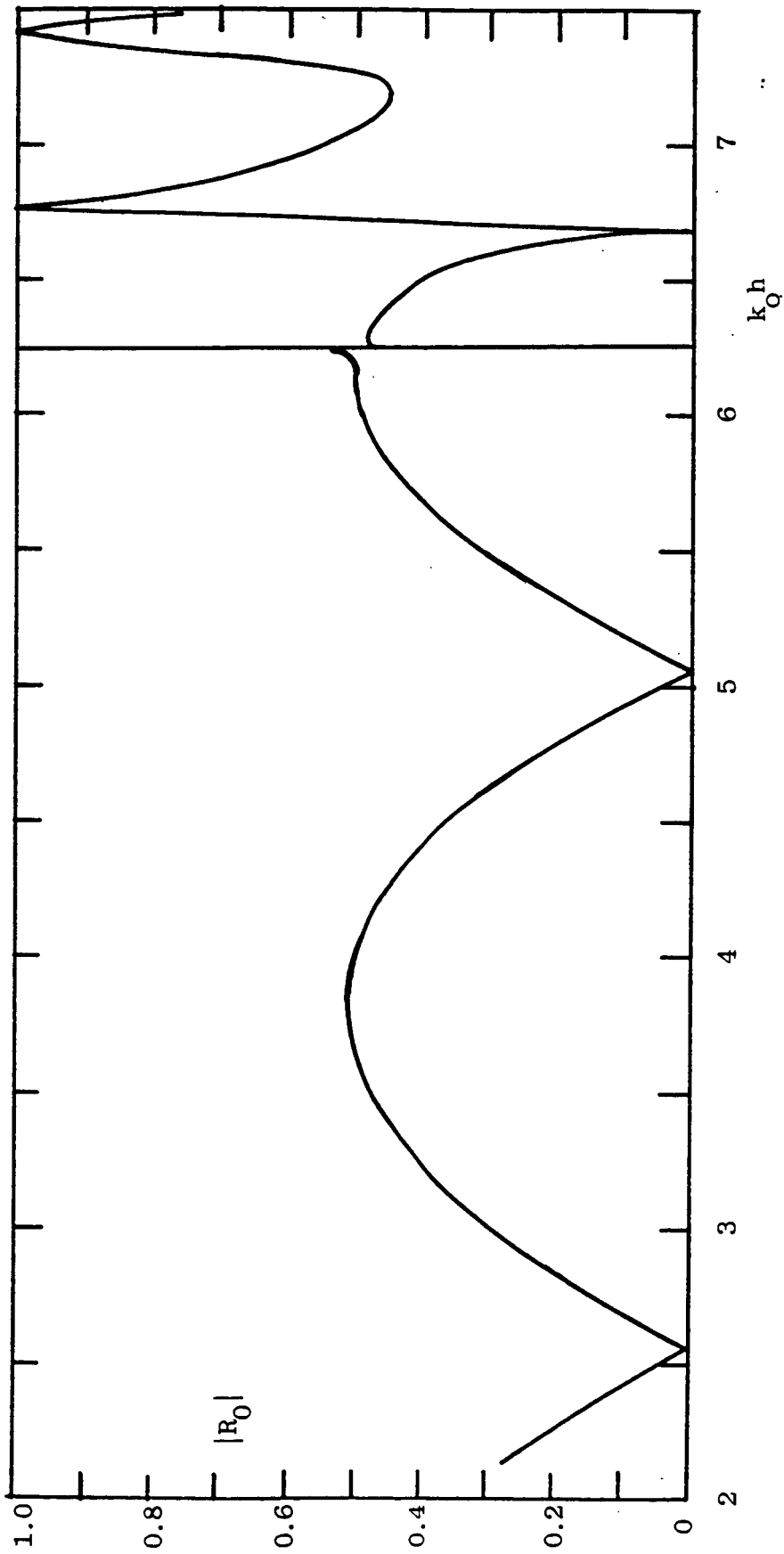
Figure 14. Variation of $|R_0|$ with normalized frequency $k_0 h$ for $\varepsilon_1 = 2.56$, $\varepsilon_2 = 1.44$ and $h/d = 1.1/0.54$.

38

C. Influence of Modulation

To explore the influence of modulation, we have computed the reflection and transmission coefficients for a layer with $\varepsilon_1 = 3$ and $\varepsilon_2 = 1$. This layer has average dielectric constant $\varepsilon_a = 2$, as before. We further assume that $d = 0.54$ and $h = 0.925$, as in the case of the results presented in Figures 10-12. A plot of $|R_0|$ versus normalized frequency $k_0 h$ is shown in Figure 15. The variation of $|R_0|$ is seen to be qualitatively the same as that of Figure 10. The increased modulation is seen to shift the first frequency of total reflection ($|R_0| = 1$) to $k_0 h = 5.45$ and the second to $k_0 h = 6.12$, which are farther from the respective values 5.27 and 5.72 predicted by small modulation theory.

Besides shifting the frequency of total reflection, the modulation influences the bandwidth. At the first total-reflection frequency, the bandwidth for $|R_0|^2 \geq 0.9$ is 0.001%. However, at the higher total reflection frequency, the bandwidth is 2.5%. The modulation is also seen to have a small effect on the frequencies of total transmission ($|R_0| = 0$).
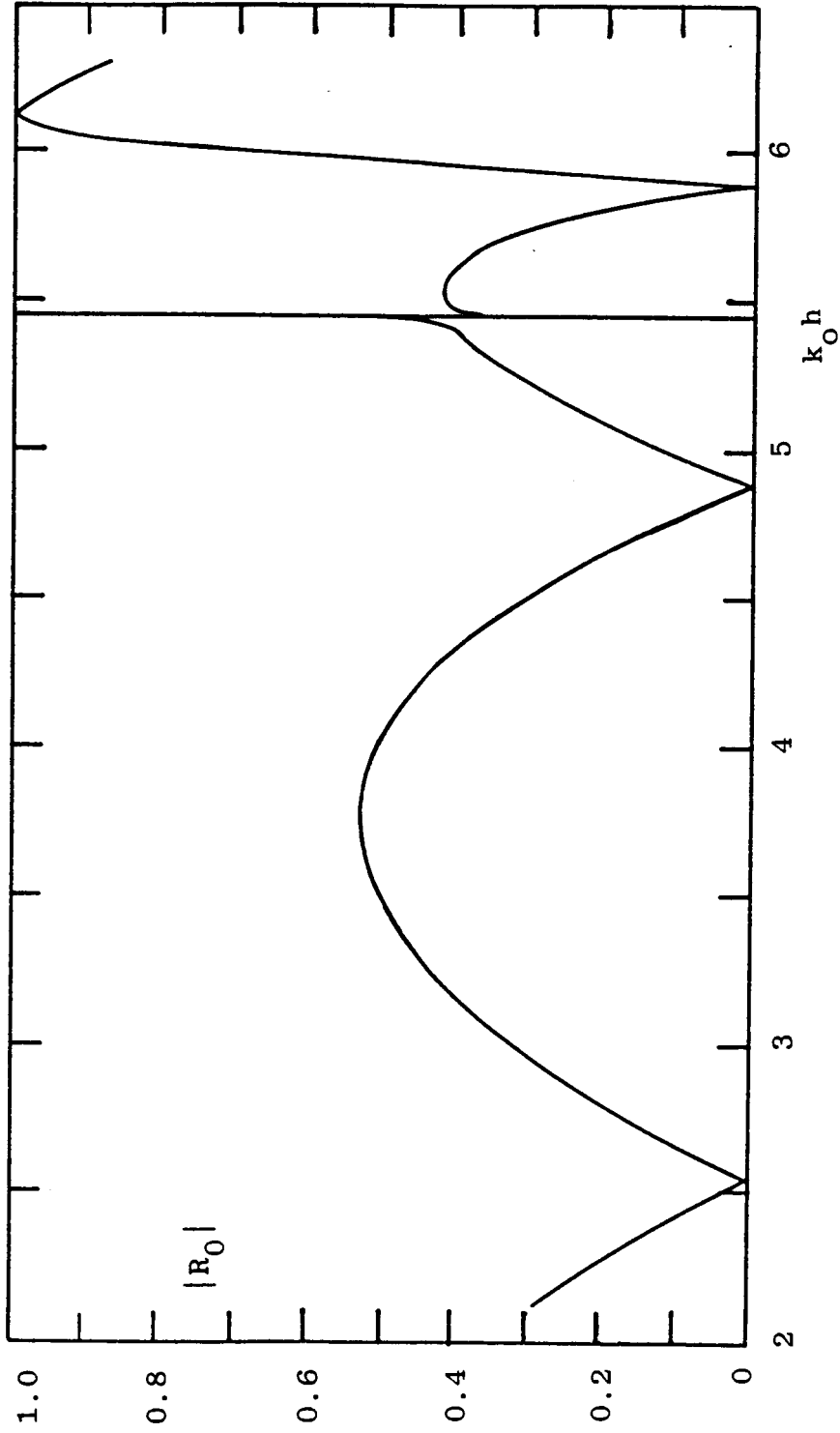
Figure 15. Variation of $|R_0|$ with normalized frequency $k_O h$ for $\epsilon_1 = 3$, $\epsilon_2 = 1$ and $h/d = 0.925/0.54$.

V. Conclusion

It has been shown that frequency selective reflection and transmission takes place at a periodically modulated dielectric layer. Frequencies of total transmission and total reflection were found, and they can be related to various wave phenomena. In the limit of small modulation, these frequencies can be estimated from the appropriate wave phenomenon in a uniform layer having dielectric constant equal to the average of that in the periodic layer.

In the range of low frequencies where a single space harmonic propagates along z in the periodic dielectric, total transmission occurs when the layer thickness h is one half the effective wavelength along z, i.e., when $h = \pi/\kappa_0$. For small modulation, $\kappa_0 \simeq (\varepsilon_a k_0{}^2 - S_0{}^2)^{\frac{1}{2}}$, where ε_a is the average dielectric constant. Total reflection can be achieved at those higher frequencies for which two space harmonics propagate along z in the periodic dielectric. These frequencies of total transmission are associated with the excitation of leaky waves guided by the dielectric layer. In the limit of small modulation, the frequency of total reflection can be approximated from the dispersion characteristics of waves guided by a uniform dielectric layer.

In the examples treated, the bandwidth over which $|R_0|^2 \geq 0.9$ about the frequency of total reflection was found to be small. The largest bandwidth obtained was 2.5%. While angle sensitivity was not computed, the narrow frequency bandwidth suggests that, at the frequency of total reflection, $|R_0|^2$ will be sensitive to the angle of incidence θ.

Whereas the study was carried out only for the TE polarization, the form of the results have implications for the TM polarization. We expect

that the frequencies of total reflection for the TM polarization are also associated with the excitation of the leaky waves guided by the periodic layer. However, the dispersion characteristics of the leaky TM waves will differ from those of the TE polarization. As a result, it is expected that incident plane waves of the TE and TM polarizations will, in general, experience total reflection at different frequencies. Hence, the periodic dielectric layer is expected to be polarization sensitive.

References

1. V.D. Agrawal and W.A. Imbriale, "Design of a Dichroic Cassegrain Subreflector," IEEE Trans., AP-27 (1979), pp. 466-473.

2. B.A. Munk, R.G. Kouyoumjian and L. Petris, "Reflection Properties of Periodic Surfaces of Loaded Dipoles," IEEE Trans., AP-19 (1971), pp. 612-616.

3. E.L. Pelton and B.A. Munk, "Scattering from Periodic Arrays of Crossed Dipoles," IEEE Trans., AP-27 (1979), pp. 323-330.

4. W.V.T. Rusch, "The Current State of the Reflector Antenna Art," IEEE Trans., AP-32 (1984), pp. 313-329.

5. R.S. Chu and T. Tamir, "Guided-Wave Theory of Light Diffraction by Acoustic Microwaves," IEEE Trans., MTT-18 (1970), pp. 486-504.

6. S.T. Peng, T. Tamir and H.L. Bertoni, "Theory of Periodic Dielectric Waveguides," IEEE Trans., MTT-23 (1975), pp. 123-133.

7. A. Hessel and A.A. Oliner, "A new theory of Wood's Anomalies on Optical Gratings," Appl. Optics, 4 (1965), pp. 1275-1297.

8. T. Tamir, "Nonspecular Phenomena in Beam Fields Reflected by Multi-layered Media," J. Opt. Soc. Amer. A, 3 (1986), pp. 558-565.

9. R.E. Collin, *Field Theory of Guided Wave* (McGraw-Hill, 1960), Ch. 11, p. 453.

10. T. Tamir and F.Y. Kou, "Varieties of Leaky Waves and Their Excitation," IEEE J. Quant. Electron, QE-22 (1986), pp. 544-551.

11. R.E. Collin, "Reflection and Transmission at a Slotted Dielectric Interface," Can. J. Phys., 34 (1956), pp. 398-411.

12. L.R. Lewis and A. Hessel, "Propagation Characteristics of Periodic Arrays of Dielectric Slabs," IEEE Trans., MTT-19 (1971), pp. 276-286.

Appendix A: Listing of Computer Program

```
   1.      //TEMNNPOP JOB (F109602,CHEO),'L.S.CHEO'
   2.      // EXEC PLIXCLG
   3.      //PL1L.SYSIN DD *
   4.       TEMODE: PROCEDURE OPTIONS(MAIN);
   4.1           /* FINAL PROGRAM FOR TE MODE WITH ONE AND TWO       */
   4.2           /* PROPAGATION MODES                                */
   5.            /* GAMA(-1) CAN BE EITHER REAL OR IMAGINARY         */
   5.2         /*********************************************************/
   5.3         /* INSTRUCTION FOR DATA ENTRY   :                  */
   5.4         /* THE FIRST DADA CARD ENTRY ORDER IS :            */
   5.5         /* D,D1,D2,E1,E2,LAMTA,DEGREE,H                    */
   5.6         /* THE SECOND DATA CARD IS FOR VALUES OF M AS :    */
   5.7         /* M=-6,-5,-4,-3,-2,-1,0,1,2,3,4,5                 */
   5.8         /*********************************************************/
   6.            /* CONSIDER V(-6:5),U(-6:5), SN(-6:5) AS ARRAYS     */
   7.            /* XI = U/V, U IS REAL , V CAN BE REAL OR IMAGINARY */
   8.
   9.                /*                                              */
  10.                /*   FIND A ROOT NEAR UO=SO OF THE FUNCTION     */
  11.                /* F(U) = 0  BY NEWTON'S METHOD , TO SIX DIGIT  */
  12.                /*  ACCURACY.                                   */
  13.                /*                                              */
  15.            /* THIS PROGRAM COMBINES TEMNNPO, RTLCOMP,AND RTLGAUS */
  16.            DCL (AA(1:8,1:11),X(1:3,1:8)) FLOAT ;
  16.01         /* AA IS THE COEFFICIENT MATRIX FOR GAUSS ELIMINATION  */
  16.02         /* AND X IS THE SOLUTION VECTOR OBTAINED               */
  16.03         /* FROM THE GAUSS ELIMINATION METHOD PROCEDURE         */
  16.1          /*                                              */
  16.2          /* MATRIX MN(M,N), EXCEPT PRINTING PART FOR ALPHA*/
  16.3          /* SUMJREAL,SUMJIMAG, SUMJ1J2 ARE ELIMINATED.   **/
  16.4          /***                                        ***/
  17.           DCL (S,SS,U,(U1,U2)(-6:5),EPSILON) FLOAT  DEC;
  18.           DCL (V,SM)(-6:5) FLOAT ,
  19.               (K,KMAX) FIXED DEC;
  20.           DCL (UD1(-6:5),RTRD2,D2,D1,D,SO,SQRTR) FLOAT ;
  21.           DCL (E1,E2,KO,SOD,UR,RU,SUM1,SUM2,R(-6:5)) FLOAT ;
  22.           DCL (A,B,TERM,PI,SMD(-6:5),GAMAIN,GAMA(-6:5)) FLOAT ;
  23.               DCL M FIXED (4,0)INITIAL(-5);
  24.           DCL (Y,YR,YI,Y1,Y2,Y3,Y4,YMAG)(-6:5,-6:5) FLOAT ;
  25.           DCL (J1R1,J1R2,J1R3,J1R4,J1R)(-6:5,-6:5) FLOAT;
  26.           DCL (J1I1,J1I2,J1I3,J1I4,J1I)(-6:5,-6:5) FLOAT;
  27.           DCL (J1,J2)(-6:5,-6:5) FLOAT;
  28.           DCL (J2R1,J2R2,J2R3,J2R4,J2R)(-6:5,-6:5) FLOAT;
  29.           DCL (J2I1,J2I2,J2I3,J2I4,J2I)(-6:5,-6:5) FLOAT;
  30.           DCL (SUMJ1J2,SUMJR,SUMJI)(-6:5,-6:5) FLOAT;
  31.           DCL (AMN,AMNR,AMNI,MN,MNR,MNI)(-6:5,-6:5) FLOAT;
  32.           DCL SUML(-6:5) FLOAT ;
  33.           DCL (ALPHA,CN,COSVD2,SINVD2)(-6:5) FLOAT;
  34.           DCL (UMD1,SND,SND1,SND2,VMD2)(-6:5) FLOAT;
  35.           DCL (COSHVD2,SINHVD2)(-6:5) FLOAT;
  35.01         DCL (TR,TI)(3,-2:1) FLOAT(6);
  35.02         DCL (SCATR,SCATI)(-2:1,-2:1) FLOAT(6);
  35.03         DCL (ROR,ROI)(1:3) FLOAT (6) ;
  35.04         DCL (LAMTA,DEGREE,THETA,KO2,RATIO)FLOAT;
  35.041        DCL(N,I,J,P) FIXED(5,0) ;
  35.05          DCL (RTLR(4,4),RTLI(4,4)) FLOAT;
  35.06         DCL (AB(12),CD(12)) FLOAT ;
  35.07         DCL GAMA0_1H FLOAT ; /* GAMA(0)*GAMA(-1)*H   */
  35.08         DCL GAMA02H FLOAT; /* 2*GAMA(0)*H */
  35.09         DCL GAMA 12H FLOAT: /* 2*GAMA(-1)*H */
```

44

```
35.1              DCL GAMAOH FLOAT ; /* GAMA(O)*H  */
35.11             DCL GAMA_1H FLOAT;   /* GAMA(-1)*H */
36.               DCL (LL) FIXED(5,0);
36.1                DCL FLAG2 FIXED(2,0);
36.2              DCL(RREAL,RIMAG,RPRIMER,RPRIMEI) FLOAT;
37.                 DCL (CDISC,GAMAD)(-6:5) FLOAT;
37.1              DCL (B2R,B2I,B3R,B3I)(-6:5) FLOAT;
37.2              DCL H FLOAT ;
37.3              DCL(MM,NN,PIVOT,RS) FIXED (5,0) ;
37.31              DCL(TOR,TOI,TMINUS1R,TMINUS1I) FLOAT;
37.32              DCL(RABS,RPHASE,RPRABS,RPRPHASE) FLOAT;
37.4        /* MM=NO. OF COLUMNS & NN=NO. OF ROWS IN MN(N,M)      */
37.5        /* MATRIX IN PROGRAM TEMNNPO                          */
37.6        /* IF PIVOT=1 THEN THE PIVOT IS SUBSCRIBED,           */
37.7        /* RS=NO. OF RHS OF THE AUGMENTED COEFFICIENT MATRIX*/
37.8        /* IN THE PROCEDURE GAUSS(ELIMINATION METHOD          */
37.9        /*                                                    */
38.               /*   DEFINE THE FUNCTION F(X)                          */
39.               F : PROCEDURE(U,M) ;
39.1                 DCL U FLOAT ;
40.                    DCL M FIXED(4,0) ;
41.                    CALL CHECKR(U,M) ;
42.              S=COS(UD1(M))*A-0.5*SUM1*SIN(UD1(M))*B-COS(SOD) ;
43.                 RETURN(S);
44.              END F;
45.              /*                                                    */
46.              /*   DEFINE THE 1ST DERIVATIVE OF F(X)                */
47.              FPRIME: PROCEDURE(U,M);
47.1                    DCL U FLOAT ;
48.                    DCL M FIXED (4,0);
49.              DCL (TERM1,TERM2,TERM3 ) FLOAT;
50.                    CALL CHECKR(U,M) ;
51.              TERM1 =-D1*SIN(UD1(M))*A-D2*COS(UD1(M))*B*UR ;
52.              TERM2=-0.5*SUM2*SIN(UD1(M))*B ;
53.              TERM3=-0.5*SUM1*(D1*COS(UD1(M))*B+TERM);
54.              SS=TERM1 + TERM2 + TERM3 ;
55.                    RETURN(SS);
56.              END FPRIME;
57.              /* PROCEDURE TO CHECK THE SIGN OF R=U**2-KO**2*(E1-E2) */
58.              /*                                                    */
59.              CHECKR : PROC(U,M);
59.1                    DCL U FLOAT ;
60.                 DCL M FIXED(4,0);
61.                    UD1(M) = U * D1 ;
62.                    R(M) =U**2-KO**2*(E1-E2) ;
63.                    SQRTR = SQRT(ABS(R(M))) ;
64.                    RTRD2 = SQRTR*D2 ;
65.                    RU = SQRTR/U ;
66.                    UR = U/SQRTR ;
67.                    SOD = SO * D ;
68.                    IF R(M) > 0 THEN
69.                      DO ;
70.                      A = COS(RTRD2) ;        /* WHEN R > 0  */
71.                      B = SIN(RTRD2) ;
72.                      SUM1 = RU + UR ;
73.                      SUM2 = 2/SQRTR-RU/U-UR**2/SQRTR ;
74.                      TERM = D2*SIN(UD1(M))*A ;
75.                      END ;                /* END OF R>0 CASE */
76.                    ELSE
77.                      DO ;                /* WHEN R<0     */
78.                      A = COSH(RTRD2);
79.                      B = SINH(RTRD2) ;
80.                      SUM1 = UR - RU ;
81.                      SUM2 = 2/SQRTR+RU/U + UR**2/SQRTR ;
82.                      TERM = -D2*SIN(UD1(M))*A ;
83.                      END ;                /*   END OF R<0 CASE   */
```

```
84.                              END CHECKR ;   /** END OF CHECKR PROCEDURE **/
85.              /*                                                          */
86.            FLAG=1;
87.            ON ENDFILE(SYSIN) FLAG=0;
88.            /*  MAIN PROCEDURE                                            */
89.                  PI = 3.14156 ; EPSILON=1.0E-06; KMAX=50 ;
90.              GET LIST(D,D1,D2,E1,E2,LAMTA,DEGREE,H) ;
90.1            KO = 2*PI/LAMTA ;
90.2            PUT SKIP;
90.21            IF LAMTA=1.0 THEN
90.3            PUT SKIP EDIT('TE MODE WITH TWO PROPAGATION MODES')
90.301                (X(5),A);
90.31             ELSE
90.311            DO ;
90.32            PUT SKIP ;
90.33            PUT SKIP EDIT('TE MODE WITH ONE PROPAGATION MODE')
90.34                 (X(5),A);
90.35             END ;
90.5            PUT SKIP EDIT (REPEAT('*',40))(X(5),A);
90.6            PUT SKIP EDIT('D=',D,'D1=',D1,'D2=',D2,'THETA=',DEGREE)
90.7                (X(5),4 (A,F(9,3),X(4)));
90.71           PUT SKIP EDIT('E1=',E1,'E2=',E2,'LAMTA=',LAMTA,'KO=',KO)
90.9                (X(5),4 (A,F(8,5),X(3)));
90.91           PUT SKIP;
90.911          PUT SKIP EDIT('H=',H)(X(5),A,F(8,4));
90.92           PUT SKIP EDIT(REPEAT('*',50))(X(5),A);
90.93           PUT SKIP ;
91.            CALL INPUT(M);
92.            LOOP: DO WHILE(FLAG=1);
93.                CALL CALCULATE(M);
94.                CALL PRINT(M);
95.                FLAG2=1;
96.                CHECK1: DO WHILE(ABS((U2(M)-U1(M))/U1(M))>=EPSILON &
96.1                         FLAG2=1) ;
97.                       IF K<=KMAX THEN DO;
98.                           U1(M)=U2(M);
99.                           CALL CALCULATE(M);
100.                          CALL PRINT(M);
101.                      END;
102.                      ELSE DO;
103.                          PUT SKIP(2) EDIT('FAILS TO CONVERGE')(A);
104.                          FLAG2=0;
105.                      END;
106.                  END CHECK1;
107.                  PUT SKIP(2);
108.                PUT SKIP(2) EDIT('R=',R(M))(X(5),A,E(14,6));
109.                IF R(M) < 0 THEN
110.                PUT SKIP(2) EDIT('V IS IMAGINARY')(X(5),A);
111.                V(M) = SQRTR ;
112.                PUT SKIP(2) EDIT('V(M)=',V(M))(X(5),A,E(14,6));
113.                GAMAIN = KO**2*E1-U2(M)**2 ;
114.                PUT SKIP(2) EDIT('KO**2*E1-U2**2=',GAMAIN)(X(5),A,E(14,6
));
115.                IF GAMAIN < 0 THEN
116.                PUT SKIP(2) EDIT('GAMA IS IMAGINARY')(X(5),A);
117.                GAMA(M) = SQRT(ABS(GAMAIN));
118.                PUT SKIP(2) EDIT('GAMA=',GAMA(M))(X(5),A,E(14,6));
119.                 IF FLAG2=1 THEN CALL OUTPUT(M);
120.                 CALL INPUT(M);
121.            END LOOP;
122.             /*                                                          */
123.        /* TO COMPUTE UM*D1,VM*D2,SN*D1,SN*D2,SN*D **/
124.        /* TO COMPUTE COS(VMD2),COSH(VMD2),SIN(VMD2),SINH(VMD2) **/
125.        /*********************************************************/
126.        LOOPM : DO M=-6 TO 5 ;
127.                UMD1(M) = U2(M) * D1;
```

```
128.                     VMD2(M) = V(M) * D2 ;
129.                     SND1(M) = SM(M) * D1 ;
130.                     SND2(M) = SM(M) * D2 ;
131.                     SND(M)  = SM(M) * D ;
132.                     SINVD2(M) = SIN(VMD2(M));
133.                     SINHVD2(M) = SINH(VMD2(M)) ;
134.                     COSVD2(M) = COS(VMD2(M)) ;
135.                     COSHVD2(M) = COSH(VMD2(M));
136.            END LOOPM ;
137.       /************************************************************/
138.       /* TO COMPUTE Y(M,N) ****************************************/
139.       /*                                                        */
140.       LOOPYM : DO M = -6 TO 5 ;
141.       LOOPYN : DO N = -6 TO 5 ;
142.               Y1(M,N) = COS(UMD1(M))*SIN(SND(N));
143.               Y4(M,N) = SIN(UMD1(M)) * SIN(SND(N))/U2(M);
144.               IF R(M) > 0 THEN
145.                 DO ;
146.                   Y3(M,N)=SINVD2(M)/V(M)+SIN(UMD1(M))*COS(SND(N))/U2(M);
147.                   Y2(M,N)=COSVD2(M)-COS(UMD1(M))*COS(SND(N)) ;
148.                   END;
149.                 ELSE
150.                 DO ;
151.                   Y3(M,N)=SINHVD2(M)/V(M)+SIN(UMD1(M))*COS(SND(N))/U2(M);
152.                   Y2(M,N)=COSHVD2(M)-COS(UMD1(M))*COS(SND(N)) ;
153.                   END ;
154.       /************************************************************/
155.       /* TO COMPUTE ABS(Y(M,N)**2)                            **/
156.       /* TO COMPUTE REAL AND IMAGINARY PART OF Y: YR, YI      **/
157.               YMAG(M,N)=Y3(M,N)**2+Y4(M,N)**2 ;
158.               YR(M,N)=(Y1(M,N)*Y3(M,N)+Y4(M,N)*Y2(M,N))/YMAG(M,N);
159.               YI(M,N)=(Y2(M,N)*Y3(M,N)-Y1(M,N)*Y4(M,N))/YMAG(M,N);
160.            END LOOPYN ;
161.            END LOOPYM ;
162.       /************************************************************/
163.       /* TO COMPUTE J1(M,N)                                    */
164.       /* REAL AND IMAGINARY PART OF JI : J1R, J1I              */
164.1           PUT SKIP ;
165.       LOOPJ1M : DO M = -6 TO 5 ;
166.               DO N = -6 TO 5 ;
167.               J1R1(M,N)=YI(M,N)*(1-COS(UMD1(M))*COS(SND1(N))) ;
168.               J1R2(M,N)=(SM(N)+YR(M,N))*COS(UMD1(M))*SIN(SND1(N));
169.               J1R3(M,N)=(U2(M)+SM(N)*YR(M,N)/U2(M))*COS(SND1(N));
170.               J1R4(M,N)=SIN(SND1(N))*SM(N)*YI(M,N)/U2(M);
171.               J1R(M,N)=(J1R1(M,N)-J1R2(M,N)+SIN(UMD1(M))*
172.                        (J1R3(M,N)-J1R4(M,N)))/(U2(M)**2-SM(N)**2);
173.               J1I1(M,N)=(SM(N)+YR(M,N))*(COS(UMD1(M))*COS(SND1(N))-1);
174.               J1I2(M,N)=YI(M,N)*COS(UMD1(M))*SIN(SND1(N));
175.               J1I3(M,N)=SM(N)*YI(M,N)*COS(SND1(N))/U2(M);
176.               J1I4(M,N)=SIN(SND1(N))*(U2(M)+SM(N)*YR(M,N)/U2(M));
177.               J1I(M,N)=(J1I1(M,N)-J1I2(M,N)+SIN(UMD1(M))*
178.                        (J1I3(M,N)+J1I4(M,N)))/(U2(M)**2-SM(N)**2);
178.1              END ;
179.            END LOOPJ1M;
180.       /************************************************************/
181.       /* TO COMPUTE J2(M,N)                                    */
182.       /* REAL AND IMAGINARY PART OF J2 : J2R, J2I             */
182.1           PUT SKIP;
183.       LOOPJ2M : DO M = -5 TO 5 ;
184.               DO N = -5 TO 5 ;
185.               J2R4(M,N)=SM(N)*YI(M,N)*SIN(SND2(N))/V(M);
186.               J2I3(M,N)=SM(N)*YI(M,N)*COS(SND2(N))/V(M);
187.               IF R(M) >0 THEN
188.                 DO;
189.                   J2R1(M,N)=YI(M,N)*(COSVD2(M)*COS(SND2(N))-1);
190.                   J2R2(M.N)=(SM(N)+YR(M.N))*COSVD2(M)*SIN(SND2(N));
```

47

```
191.                          J2R3(M,N)=(V(M)+SM(N)*YR(M,N)/V(M))*COS(SND2(N));
192.                          J2R(M,N)=(J2R1(M,N)-J2R2(M,N)+SINVD2(M)*
193.                                  (J2R3(M,N)+J2R4(M,N)))/(V(M)**2-SM(N)**2);
194.                          J2I1(M,N)=(SM(N)+YR(M,N))*(COSVD2(M)*COS(SND2(N))-1);
195.                          J2I2(M,N)=COSVD2(M)*SIN(SND2(N))*YI(M,N);
196.                          J2I4(M,N)=SIN(SND2(N))*(V(M)+SM(N)*YR(M,N)/V(M));
197.                          J2I(M,N)=(-J2I1(M,N)-J2I2(M,N)+SINVD2(M)*
198.                                  (J2I3(M,N)-J2I4(M,N)))/(V(M)**2-SM(N)**2);
199.                        END ; /*END OF R(M) > 0 FOR J2 COMPUTATIONS  ***/
200.                        ELSE
201.                          DO ;
202.                          J2R1(M,N)=YI(M,N)*(COSHVD2(M)*COS(SND2(N))-1);
203.                          J2R2(M,N)=(SM(N)+YR(M,N))*COSHVD2(M)*SIN(SND2(N));
204.                          J2R3(M,N)=(-V(M)+SM(N)*YR(M,N)/V(M))*COS(SND2(N));
205.                          J2R(M,N)=-(J2R1(M,N)-J2R2(M,N)+SINHVD2(M)*
206.                                  (J2R3(M,N)+J2R4(M,N)))/(V(M)**2+SM(N)**2);
207.                          J2I1(M,N)=(SM(N)+YR(M,N))*(COSHVD2(M)*COS(SND2(N))-1);
208.                          J2I2(M,N)=COSHVD2(M)*SIN(SND2(N))*YI(M,N);
209.                          J2I4(M,N)=SIN(SND2(N))*(-V(M)+SM(N)*YR(M,N)/V(M));
210.                          J2I(M,N)=-(-J2I1(M,N)-J2I2(M,N)+SINHVD2(M)*(J2I3(M,N)
211.                                  -J2I4(M,N)))/(V(M)**2+SM(N)**2);
212.                        END ; /* END OF R(M) <= 0 FOR J2 COMPUTATION   ***/
212.1                END ;
213.                  END LOOPJ2M ;
214.         /*****************************************************************/
215.         /*                                                             */
216.         /* TO COMPUTE SUMJ1J2                                          */
217.         /* REAL AND IMAGINARY PART OF SUMJ1J2 : SUMJR , SUMJI         */
218.         SUMJM : DO M = -5 TO 5 ;
219.         SUMJN : DO N = -5 TO 5 ;
220.                  SUMJR(M,N)=J1R(M,N)+J2R(M,N) ;
221.                  SUMJI(M,N)=J1I(M,N)+J2I(M,N) ;
222.                  SUMJ1J2(M,N)=SUMJR(M,N)**2+SUMJI(M,N)**2 ;
223.                  END SUMJN;
224.                END SUMJM ;
225.         /****** TO COMPUTE ALPHA(M), M=-6 TO 5       **************/
226.         /* TRUNCATE INFINITE SUM TO SUM SUMJ1J2(M,N)             */
227.         /*                                                       */
228.         ALFA :     DO M = -5 TO 5 ;
229.                  SUML(M)=0 ; PUT SKIP ;
230.         SUMLL :    DO LL = -5 TO 5 ;
231.                  SUML(M) = SUML(M) + SUMJ1J2(M,LL) ;
232.                  END SUMLL;
233.         /***** TO COMPUTE ALPHA(M)                        ********/
234.         /*                                                       */
235.                  ALPHA(M) = SQRT(SUML(M));
235.1          PUT SKIP EDIT('ALPHA(',M,')=',ALPHA(M))(X(5),A,F(2,0),A,E(12,5)
);
236.                  END ALFA;
261.         /** TO COMPUTE CN(N),GAMA(M),AND MN(N,M) M,N = -6 TO 5  **/
262.         /**                                                     **/
262.1                  PUT SKIP;
262.2                  PUT SKIP EDIT(REPEAT('*',40))(X(3),A);
262.3                  PUT SKIP;
263.         LOOPCN:    DO N = -5 TO 5;
264.                  CDISC(N) = K0**2-SM(N)**2 ;
265.                  CN(N) = SQRT(ABS(CDISC(N))) ;
265.2                  PUT SKIP EDIT('CN(',N,')=',CN(N),'CDISC(',N,')=',CDISC(N))
265.3                    (X(3),2 (A,F(3,0),A,E(12,5),X(2)));
266.                  END LOOPCN ;
267.         GAMADM:    DO M = -5 TO 5 ;
268.                  GAMAD(M) = K0**2*E1-U2(M)**2 ;
268.1                  GAMA(M)=SQRT(ABS(GAMAD(M)));
268.2                  PUT SKIP EDIT('GAMA(',M,')=',GAMA(M),'GAMAD(',M,')=',GAMAD(
M))
268.3                    (X(3),2 (A,F(3,0),A,E(12,5),X(2)));
```

48

```
269.              END GAMADM ;
270.        LOOPAMN:  DO M = -5 TO 5;
271.        LOOPMNA : DO N = -5 TO 5 ;
272.              AMNR(M,N)=SUMJR(M,N)/ALPHA(M);
273.              AMNI(M,N)=SUMJI(M,N)/ALPHA(M);
274.              IF CDISC(N) > 0 & GAMAD(M) > 0    THEN
275.        CASE1 :   DO ;
276.              MNR(N,M) = (CN(N)+GAMA(M))*AMNR(M,N) ;
277.              MNI(N,M) = (CN(N)+GAMA(M))*AMNI(M,N) ;
278.              END CASE1 ;
279.              ELSE
280.        CASE2 :   IF  CDISC(N) < 0 & GAMAD(M) > 0 THEN
281.                DO;
282.              MNR(N,M) = GAMA(M)*AMNR(M,N)-CN(N)*AMNI(M,N) ;
283.              MNI(N,M) = CN(N)*AMNR(M,N)+GAMA(M)*AMNI(M,N) ;
284.              END ;   /*** CASE2 ****/
285.              ELSE
286.        CASE3 :    IF CDISC(N) > 0 & GAMAD(M) < 0 THEN
287.                DO ;
288.              MNR(N,M)=CN(N)*AMNR(M,N)-GAMA(M)*AMNI(M,N) ;
289.              MNI(N,M)=GAMA(M)*AMNR(M,N)+CN(N)*AMNI(M,N);
290.              END ;   /*** CASE 3 ***/
291.              ELSE
292.        CASE4 :   DO ;
293.               MNR(N,M)=-(CN(N)+GAMA(M))*AMNI(M,N) ;
294.               MNI(N,M)=(CN(N)+GAMA(M))*AMNR(M,N) ;
295.               END CASE4 ;
295.1             END LOOPMNA ;
295.2             END LOOPAMN ;
296.       /** TO PRINT REAL AND IMAGINARY PART OF MATRIX MN(M,N) **/
297.       /** : MNR(M,N) AND MNI(M,N) : M , N = -6 TO 5          **/
298.       /***                                               ***/
299.              PUT SKIP;
300.              PUT SKIP ;
301.              PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
302.              PUT SKIP EDIT('PRINT VALUES OF MNREAL AND MNIMAG ')
303.                        (X(6),A) ;
304.              PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
305.        MNPRINT: DO N = -5 TO 5 ;
305.1             PUT SKIP;
306.              PUT SKIP EDIT('N=',N)(X(7),A,F(3,0)) ;
307.              PUT SKIP EDIT (REPEAT('*',10))(X(5),A);
308.              PUT SKIP ;
309.              PUT SKIP EDIT('M','MNREAL','MNIMAG')(X(3),A,2 (X(5),A(10)))
310.              PUT SKIP ;
311.        NMPRINT: DO M = -5 TO 5;
312.              PUT SKIP EDIT(M,MNR(N,M),MNI(N,M))(X(2),F(2,0),2 E(15,5));
313.              END NMPRINT;
314.              END MNPRINT ;
314.01      /** TO COMPUTE RIGHT HAND SIDES VECTORS FOR T(II)  AND *******/
314.02      /** T(III) ,B2 AND B3 ,REAL AND IMAGINARY PARTS        *****/
314.03      /* B2(N)=(GAMA(0)-CN(N))*AMN(0,N), N=1,0,-1,-2        *****/
314.04      /* B3(N)=(GAMA(-1)-CN(N))*AMN(-1,N), N=1,0,-1,-2      *****/
314.05      /******                                              *****/
314.06      LOOPB2 : DO N = 1 TO -2 BY -1 ;
314.07             IF CDISC(N) > 0 THEN
314.08             DO ;
314.09             B2R(N)=(GAMA(0)-CN(N))*AMNR(0,N) ;
314.1              B2I(N)=(GAMA(0)-CN(N))*AMNI(0,N) ;
314.101            END ;
314.102            ELSE
314.103            DO ;
314.104            B2R(N)=GAMA(0)*AMNR(0,N)+CN(N)*AMNI(0,N) ;
314.105            B2I(N)=GAMA(0)*AMNI(0,N)-CN(N)*AMNR(0,N) ;
314.106            END : END LOOPB2 ;
```

49

```
314.107    LOOPB3: DO N=1 TO -2 BY -1; /* B3(N) FOR REAL & IMAGINARY GAMA(-1)*/
314.108            IF CDISC(N) > O & GAMAD(-1) > O
314.109            THEN DO ;
314.11               B3R(N)=(GAMA(-1)-CN(N))*AMNR(-1,N) ;
314.12               B3I(N)=(GAMA(-1)-CN(N))*AMNI(-1,N) ;
314.13              END ; /* C(N) & GAMA(-1) ARE BOTH REAL */
314.14              ELSE
314.15               IF CDISC(N) < O & GAMAD(-1) > O
314.16               THEN DO ;
314.18               B3R(N)=GAMA(-1)*AMNR(-1,N)+CN(N)*AMNI(-1,N);
314.181              B3I(N)=GAMA(-1)*AMNI(-1,N)-CN(N)*AMNR(-1,N) ;
314.19               END ; /* END OF C(N) IMAGINARY & GAMA(-1) REAL */
314.191                ELSE
314.192                 IF CDISC(N) > O & GAMAD(-1) < O
314.193                  THEN DO ;
314.194                   B3R(N)=-(GAMA(-1)*AMNI(-1,N)+CN(N)*AMNR(-1,N));
314.195                   B3I(N)=GAMA(-1)*AMNR(-1,N)-CN(N)*AMNI(-1,N);
314.196                  END ;   /* C(N) REAL & GAMA(-1) IMAGINARY */
314.197                  ELSE
314.198                 DO ;
314.199                   B3R(N)=(CN(N)-GAMA(-1))*AMNI(-1,N);
314.2                    B3I(N)=(GAMA(-1)-CN(N))*AMNR(-1,N);
314.201                  END ; /* C(N) & GAMA(-1) ARE BOTH IMAGINARY */
314.21              END LOOPB3 ; /* END COMPUTING B2 AND B3 */
314.22      /** TO PRINT B2R(N),B2I(N),B3R(N),AND B3I(N), N=1,O,-1,-2 **/
314.23      /**                                                      **/
314.24          PUT SKIP ;
314.25          PUT SKIP ;
314.26          PUT SKIP EDIT(REPEAT('*',55))(X(3),A) ;
314.27          PUT SKIP EDIT('PRINT REAL AND IMAGINARY PART OF B2 &B3')
314.28                        (X(6),A);
314.29          PUT SKIP EDIT(REPEAT ('*',55))(X(3),A) ;
314.3           PUT SKIP ;
314.31          PUT SKIP EDIT('N','B2REAL','B2IMAG','B3REAL','B3IMAG')
314.32                        (X(3),A,4 (X(3),A(10))) ;
314.33          PUT SKIP ;
314.34      PRINTB: DO N = -2 TO 1 ;
314.35          PUT SKIP EDIT(N,B2R(N),B2I(N),B3R(N),B3I(N))
314.36                          (X(2),F(2,0),4 E(15,5)) ;
314.361             END PRINTB; /** END PRINTING B2 AND B3 **/
314.37      /** PRINT AMNR(M,N),AMNI(M,N): M,N = -6 TO 5 *********/
314.38      /**                                              **/
314.39           PUT SKIP;
314.4            PUT SKIP ;
314.41           PUT SKIP EDIT(REPEAT('*',50))(X(3),A);
314.42           PUT SKIP EDIT('PRINT VALUES OF AMNREAL AND AMNIMAG')
314.43                          (X(6),A);
314.44           PUT SKIP EDIT (REPEAT('*',55))(X(3),A);
314.45      PRINTAM: DO M = -5 TO 5 ;
314.46           PUT SKIP ;
314.47           PUT SKIP EDIT('M=',M)(X(7),A,F(3,0)) ;
314.48           PUT SKIP EDIT (REPEAT('*',10))(X(5),A) ;
314.49           PUT SKIP;
314.5            PUT SKIP EDIT('N','AMNREAL','AMNIMAG')
314.51                          (X(3),A,2 (X(5),A(10)));
314.52            PUT SKIP;
314.53      PRINTAN: DO N = -5 TO 5 ;
314.54           PUT SKIP EDIT(N,AMNR(M,N),AMNI(M,N))
314.55                          (X(2),F(2,0),2 E(15,5)) ;
314.56            END PRINTAN ;
314.57           END PRINTAM ;
315.       /**************************************************************/
316.            /*  SUBROUTINE TO INPUT DATA                              */
317.            INPUT: PROC(M);
318.                  DCL M FIXED (4,0);
319.                  GET LIST (M);
```

```
320.              K=0;
321.              PI = 3.14156 ;
322.              THETA = PI * DEGREE/180. ;
323.              SO = KO *SIN(THETA) ;
324.              SM(M) = SO + M*2*PI/D ;
325.              KO2 = (D2/D)*KO**2*(E1-E2) ;
326.              U1(M) = SQRT(SM(M)**2+KO2) ;
327.              IF FLAG=1 THEN DO;
328.                  PUT SKIP (2) ;
329.                  PUT SKIP(2) ;
330.                  PUT PAGE EDIT('NEWTON''S METHOD')(X(10),A);
331.                  PUT SKIP(2);
332.                  PUT SKIP(2) EDIT('***TE MODE ***')(X(10),A);
333.                  PUT SKIP(2) ;
334.                  PUT SKIP(3) EDIT('INITIAL VALUES')(X(10),A);
335.                  PUT SKIP (2);
336.                  PUT SKIP(2) EDIT('M=',M)(X(5),A,F(4,0));
337.                  PUT SKIP EDIT('U1(M)= ',U1(M),'EPSILON= ',EPSILON,
338.                       'KMAX= ',KMAX)(X(5),2(A,E(12,5),X(5)),A,F(3,0))

339.                  PUT SKIP(5) EDIT('U1','U2','F(U1)','F''(U1)',
340.                       '!U2-U1!','COUNT')(X(3),A,X(12),A,X(10),A,
341.                       X(7),A,X(8),A,X(2),A);
342.              END;
343.              RETURN;
344.          END INPUT;
345.          /*                                                      */
346.          /*   SUBROUTINE TO PERFORM CALCULATION                  */
347.          CALCULATE: PROC(M);
348.                  DCL M FIXED (4,0) ;
349.                  U2(M)=U1(M)-F(U1(M),M)/FPRIME(U1(M),M);
350.                  K=K+1;
351.                  RETURN;
352.          END CALCULATE;
353.          /*                                                      */
354.          /*   SUBROUTINE TO PRINT TABLE                          */
355.          PRINT:PROC(M);
356.                  DCL M FIXED(4,0);
357.              PUT SKIP EDIT(U1(M),U2(M),F(U1(M),M),FPRIME(U1(M),M),
358.                  ABS(U2(M)-U1(M)),K)
359.                  (5(E(12,5),X(1)),F(2,0));
360.              RETURN;
361.          END PRINT;
362.          /*                                                      */
363.          /*   SUBROUTINE TO PRINT FINAL RESULTS                  */
364.          OUTPUT: PROC(M);
365.                  DCL M FIXED(4,0) ;
366.                  PUT SKIP(5) EDIT('APPROXIMATE ROOT U2= ',U2(M),
367.                      'F(U2)= ',F(U2(M),M))(A,E(14,7),X(5),A,E(14,7));
368.                  RETURN;
369.          END OUTPUT;
369.01      /* TO COMPUTE SCATTRING MATRIX SCAT                       */
369.02      /* CONSTRUCT COEFFICIENT MATRIX AA FROM MATRIX MN(N,M)    */
369.03          MM= 8;
369.04          NN= 8 ;
369.041         AAI1: DO I = 1 TO NN/2 ;
369.042         AAJ1: DO J = 1 TO MM/2;
369.043             AA(I,J)=MNR(2-I,2-J);
369.044             AA(I,J+MM/2)=-MNI(2-I,2-J);
369.045             END AAJ1 ;
369.046             END AAI1 ;
369.05          PIVOT= 1.0 ;
369.06          RS= 3 ;
369.11          AAI2: DO I = NN/2+1 TO NN ;
369.12           AAJ2: DO J = 1 TO MM/2 ;
369.13               AA(I.J)=MNI(NN/2+2-I.2-J);
```

51

```
369.14                        AA(I,J+MM/2)=MNR(NN/2+2-I,2-J);
369.15                        END AAJ2 ;
369.16                        END AAI2 ;
369.21            RHS1: DO I = 1,3 TO NN ;
369.22                        AA(I,MM+1)=0 ;
369.23                        END RHS1 ;
369.24                        AA(2,MM+1)=2*CN(0) ;
369.25       /* END OF RHS1 COLUMN                      */
369.29            RHS23R: DO I = 1 TO NN/2 ;
369.31                        AA(I,MM+2)=B2R(2-I) ;
369.32                        AA(I,MM+3)=B3R(2-I) ;
369.33                        END RHS23R ;
369.34        /* END OF RHS2 COLUMN OF AUGMENTED MATRIX IN GAUSS PROC.*/
369.35            RHS23I: DO I = NN/2+1 TO NN ;
369.36                        AA(I,MM+2)=B2I(MM/2+2-I) ;
369.37                        AA(I,MM+3)=B3I(MM/2+2-I) ;
369.372                       END RHS23I ;
369.38        /* END OF RHS3 COLUMN OF AUGMENTED MATRIX IN GAUSS PROC. */
369.39            CALL GAUSS(AA,MM,NN,PIVOT,RS,X);
369.4         /* INVOKE GAUSS ELIMINATION TO COMPUTE SCATTERING MATRIX */
400.      GAUSS: PROC(AA,M,N,PIVOT,RS,X) ;
404.          /*                                              */
407.          /*   GAUSSIAN ELIMINATION WITH OR WITHOUT PIVOTING.    */
408.          /*   ANSWERS ARE THEN SUBSTITUTED BACK INTO THE        */
409.          /* ORIGINAL EQS. WITH MULTIPLE RHS VECTORS.       */
410.          /*                                              */
456.          DCL (AA(*,*),X(*,*)) FLOAT ;
457.          DCL (M,N,PIVOT,RS) FIXED(5,0) ;
459.             START: BEGIN;
460.                DCL (AC(M,N+RS),BB(M,N+RS),HOLD(N+RS),SUM) FLOAT(6),
461.                      XX(RS,N) FLOAT(6) INIT((RS*N)0);
462.                /*   INPUT AUGMENTED MATRIX                   */
463.                CALL INPUT1;
464.                /*   CONVERT TO UPPER TRIANGULAR MATRIX         */
465.                CALL UPTRI;
466.                /*   BACK SUBSTITUTE                          */
467.                CALL BACKSUB;
468.                CALL OUTPUT1;
469.                /*   PUT ANSWERS BACK IN ORIGINAL EQUATIONS      */
470.                CALL TEST1;
477.                /*                                           */
478.                /*   SUBROUTINE TO INPUT AUGMENTED MATRIX       */
479.                INPUT1: PROC;
480.                   PUT PAGE EDIT('GAUSSIAN ELIMINATION')(X(28),A);
481.                   IF PIVOT=1 THEN PUT SKIP EDIT('WITH PIVOTING')
482.                      (X(31),A);
483.                   ELSE PUT SKIP EDIT('WITHOUT PIVOTING')(X(30),A);
484.                   PUT SKIP EDIT('FOR ',M,' BY ',N,' MATRIX')
485.                             (X(29),A,F(2,0),A,F(2,0),A);
486.                 PUT SKIP EDIT('WITH',RS,'RIGHT HAND SIDES')
487.                          (X(29),A,F(3,0),X(2),A);
488.                   PUT SKIP(5);
489.                   DO I=1 TO M;
490.                      DO J=1 TO N+RS;
491.                         AC(I,J)=AA(I,J) ;
492.                         PUT EDIT(AA(I,J))(X(1),F(8,3));
493.                         BB(I,J)=AC(I,J);
494.                      END;
495.                      PUT SKIP;
496.                   END;
497.                   RETURN;
498.                END INPUT1;
499.                /*                                           */
500.                /*   SUBROUTINE TO PRINT MATRIX               */
501.                PRINT: PROC;
502.                   PUT SKIP(5);
```

52

```
503.            DO I=1 TO M;
504.                DO J=1 TO N+RS;
505.                    PUT EDIT(AC(I,J))(X(1),F(8,3));
506.                END;
507.                PUT SKIP;
508.            END;
509.            RETURN;
510.        END PRINT;
511.        /*                                              */
512.        /*  SUBROUT. CONVERTS MATRIX TO UPPER TRIANGULAR */
513.        UPTRI: PROC;
514.            DO K=1 TO M-1;
515.                IF PIVOT=1 THEN CALL PIVOT1;
516.                DO I=K+1 TO M;
517.                    RATIO = AC(I,K)/AC(K,K) ;
518.                    DO J=K TO N+RS;
519.                        AC(I,J)=AC(I,J)-RATIO*AC(K,J);
520.                    END;
521.                END;
522.            END;
523.            RETURN;
524.        END UPTRI;
525.        /*  SUBROUTINE TO USE PIVOTING                   */
526.        PIVOT1: PROC;
527.            P=K;
528.            DO I=K+1 TO M;
529.                IF ABS(AC(P,K)) < ABS(AC(I,K)) THEN P = I;
530.            END;
531.            IF P^=K THEN DO;
532.                DO J=1 TO N+RS;
533.                    HOLD(J)=AC(K,J);
534.                    AC(K,J)=AC(P,J);
535.                    AC(P,J)=HOLD(J);
536.                END;
537.            END;
538.            RETURN;
539.        END PIVOT1;
540.        /*                                              */
541.        /*  SUBROUTINE TO BACK SUBSTITUTE               */
542.        BACKSUB: PROC;
543.            DO K = 1 TO RS ;
544.            DO I=N TO 1 BY(-1);
545.                SUM=0;
546.                DO J=I TO M;
547.                    SUM=SUM+XX(K,J)*AC(I,J);
548.                END;
549.                XX(K,I)=(AC(I,N+K)-SUM)/AC(I,I);
550.            END;
551.            END ;
552.            RETURN;
553.        END BACKSUB;
554.        /*                                              */
555.        /*  SUBROUTINE TO PRINT ANSWERS                 */
556.        OUTPUT1: PROC;
557.            PUT SKIP(5) EDIT('ANSWERS')(X(34),A);
558.            DO J = 1 TO RS ;
559.                PUT SKIP EDIT ('SET',J)(X(20),A,F(3,0));
560.                PUT SKIP;
561.                DO I=1 TO N;
561.1                   X(J,I)=XX(J,I);
562.                    PUT SKIP EDIT('X(',J,',',I,')= ',XX(J,I))
563.                        (X(28),A,F(2,0),A,F(2,0),A,F(9,6));
564.                END;
565.            END ;
566.            RETURN;
567.        END OUTPUT1:
```

53

```
568.                    /*                                              */
569.                    /*  SUBROUTINE TO PUT ANSWERS BACK IN ORIGINAL  */
570.                    /*  EQUATIONS                                   */
571.                    TEST1: PROC;
572.                       PUT SKIP(5) EDIT
573.                          ('ANSWERS PUT IN ORIGINAL EQUATIONS')(X(21),A);
574.                        DO K = 1 TO RS ;
575.                        PUT SKIP EDIT ('SET ',K)(X(29),A,F(2,0));
576.                        PUT SKIP;
577.                           DO I=1 TO M;
578.                              PUT SKIP;
579.                              SUM=0;
580.                              PUT SKIP EDIT(' ')(X(1),A);
581.                              DO J=1 TO N;
582.                                 SUM=SUM+BB(I,J)*XX(K,J);
583.                                 PUT EDIT(BB(I,J),'X(',J,') ')
584.                                          (F(9,3),A,F(1,0),A);
585.                                 IF J<N THEN PUT EDIT('+ ')(A);
586.                                 ELSE IF J=N THEN PUT EDIT('= ')(A);
587.                              END;
588.                              PUT EDIT(SUM)(F(9,3));
589.                           END;
589.1              END ;
590.                        RETURN;
591.                     END TEST1;
591.1            END START ;
591.2            END GAUSS ;
592.                      /*                                          */
592.1          CALL LOOPT;        /* PROCEDURE TO COMPUTE T VECTORS     */
592.2     /* TO PRINT OUT T VECTORS PROCEDRE CALL         */
592.3          CALL PRINTT;
592.4          CALL INNER ;   /* PROCEDRUE TO COMPUTER DOT PRODUCT */
592.5         CALL SCATTER; /* PROCEDURE TO COMPUTE SCATTERING MATRIX*/
592.6         CALL RTL ; /* PROCEDURE TO COMPUTE RTL MATRIX        */
593.         /* PROCEDURE TO COMPUTE T VECTORS           */
594.         /* TR AND TI ARE REAL AND IMAGINARY PART OF T VECTORS  */
595.         /*                                          */
596.     LOOPT: PROC;
597.            DO K = 1 TO 3 ;
598.            DO N = 1 TO -2 BY -1 ;
599.             TR(K,N)=X(K,2-N);
600.             TI(K,N)=X(K,6-N);
601.            END;
602.            END;
603.               RETURN;
604.            END LOOPT ;
605.         /* PRINT T VECTORS, TREAL AND TIMAGINARY           */
606.         /*                                          */
607.     PRINTT: PROC;
608.            PUT SKIP;
609.            PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
610.            PUT SKIP EDIT('PRINT T VECTORS FOR N = 1,0,-1,-2  ')
611.                  (X(7),A);
612.            PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
613.            PUT SKIP ;
614.            DO K = 1 TO 3 ;
615.            PUT SKIP;
616.            PUT SKIP EDIT('K=',K)(X(10),A,F(3,0));
617.            PUT SKIP EDIT (REPEAT('*',10))(X(5),A);
618.            PUT SKIP;
619.            PUT SKIP EDIT('N','TREAL','TIMAG')(X(7),A,2 (X(4),A(10)));
620.            PUT SKIP;
621.            DO N = -2 TO 1 ;
622.            PUT SKIP EDIT(N,TR(K,N),TI(K,N))(X(5),F(3,0),2 E(15,5));
623.            PUT SKIP;
624.            END;
```

```
625.                    END;
626.                    END PRINTT;
627.             /* PROCEDURE TO COMPUTE INNER PRODUCT OF AMN AND T MATRICES*/
628.             /*                                                         */
629.        INNER : PROC;
630.               DO K = 1 TO 3 ;
631.               ROR(K) = 0;
632.               ROI(K) = 0 ;
633.                DO N = -2 TO 1 ;
634.                ROR(K)=ROR(K)+AMNR(N,0)*TR(K,N)-AMNI(N,0)*TI(K,N) ;
635.                ROI(K)=ROI(K)+AMNI(N,0)*TR(K,N)+AMNR(N,0)*TI(K,N) ;
636.                END ;
637.               END;
638.                ROR(1)=ROR(1)-1;
639.                ROR(2)=ROR(2)+AMNR(0,0) ;
640.                ROI(2)=ROI(2)+AMNI(0,0);
641.                ROR(3)=ROR(3)+AMNR(-1,0);
642.                ROI(3)=ROI(3)+AMNI(-1,0);
643.           /* PRINT INNER PRODUCT OF AMN(N,0) AND T(K,N)        */
644.           /*                                                   */
645.               PUT SKIP;
646.               PUT SKIP EDIT (REPEAT('*',55))(X(7),A);
647.             PUT SKIP EDIT('RO,INNER PRODUCT OF A & T')(X(5),A);
648.               PUT SKIP EDIT(REPEAT('*',55))(X(7),A);
649.        PRINTRO: DO I = 1 TO 3 ;
650.               PUT SKIP EDIT('ROREAL(',I,')=',ROR(I),'ROIMAG(',I,')=',
651.                 ROI(I))(X(10),2 (X(2),A,F(2,0),A,E(12,5)));
652.               END PRINTRO ;
653.               END INNER ;
654.           /* END ON INNER PRODUCT PROCEDURE */
655.           /* PROCEDURE TO COMPUTE SCATTERING MATRIX   */
656.        SCATTER: PROC;
657.        LOOPSC1 : DO J = -1 TO 1 ;
658.               SCATR(1,J)=ROR(2-J) ;
659.               SCATI(1,J)=ROI(2-J);
660.               END LOOPSC1;
661.        LOOPSC2: DO I = 0,-1 ;
662.               DO J = -1 TO 1 ;
663.               SCATR(I,J)=TR(2-J,I);
664.               SCATI(I,J)=TI(2-J,I);
664.1               END ;
665.               END LOOPSC2;
666.           /* PRINT SCATTERING MATRX SCATREAL, SCATIMAG PARTS   */
667.           /*                                                   */
668.               PUT SKIP;
669.               PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
670.               PUT SKIP EDIT('PRINT SCATTERING MATRIX SCAT')
671.                       (X(10),A);
672.               PUT SKIP EDIT(REPEAT('*',55))(X(3),A);
673.               PUT SKIP;
674.               DO K = -1 TO 1 ;
675.               PUT SKIP;
676.               PUT SKIP EDIT('K=',K)(X(10),A,F(2,0));
677.               PUT SKIP EDIT(REPEAT('*',10))(X(5),A);
678.               PUT SKIP;
679.               DO N = -1 TO 1 ;
680.               PUT SKIP EDIT('SREAL(',N,',',K,')=',SCATR(N,K),
681.                           'SIMAG(',N,',',K,')=',SCATI(N,K))
682.                   (X(5),2 (X(2),A,F(2,0),A,F(2,0),A,E(12,5)));
683.               END ;
684.               END;
685.               RETURN;
686.              END SCATTER;
687.         /* COMPUTE COEFFICEINT MATRIX RTL, AND THE RHS VECTOR**/
688.         /* OF THE SYSTEM OF EQUATIONS TO SOLVE R,R',T(0), AND */
689.         /* T(-1)                                             */
```

55

```
690.        /**                                                    **/
691.     RTL  : PROC;
692.              RTLR(1,1)=1.0 ;
693.              RTLR(2,2)=1.0 ;
694.              DO I = 2 TO 4 ;
695.               RTLR(I,1)=0;
696.              END ;                /* RTLR(2,1)=RTLR(3,1)=RTLR(4,1)=0 */
697.              DO I = 1, 3 TO 4 ;
698.               RTLR(I,2)=0;   /* RTLR(1,2)=RTLR(3,2)=RTLR(4,2)=0 */
699.              END ;
700.              DO I = 1 TO 4 ;
701.               RTLI(I,1)=0 ;
702.               RTLI(I,2)=0 ;
703.              END ;     /* RTLI(I,1)=RTLI(1,2)=0,I=1,2,3,4          */
704.        /* END OF DEFINING RTL MATRIX INITIALIZATION           */
705.              GAMAOH=GAMA(0)*H ;
706.              GAMA_1H=GAMA(-1)*H ;
707.              GAMA_12H=2*GAMA_1H ;
708.              GAMAO2H=2*GAMAOH ;
709.              GAMAO_1H=(GAMA(0)+GAMA(-1))*H ;
710.        /* TO COMPUTE RTL(1,3)                  */
711.        /* REAL &  IMAGINARY PARTS OF S(1,0)*S(0,0) & S(1,-1)*S(-1,0) */
712.              DO I = 0 TO -1 BY -1 ;
713.               AB(1-I)=SCATR(1,I)*SCATR(I,0)-SCATI(1,I)*SCATI(I,0) ;
714.               CD(1-I)=SCATR(1,I)*SCATI(I,0)+SCATI(1,I)*SCATR(I,0) ;
715.              END ;   /* END OF DEFINING AB(1),AB(2),CD(1), & CD(2) */
716.        /* RTL(1,3)        */
717.              IF GAMAD(-1) > 0 THEN
718.              DO ;     /* RTL(1,3), WHEN GAMA(-1) IS REAL */
719.              RTLR(1,3)=-(AB(1)*COS(GAMAO2H)-CD(1)*SIN(GAMAO2H)
720.                        + AB(2)*COS(GAMAO_1H)-CD(2)*SIN(GAMAO_1H));
721.              RTLI(1,3)=-(AB(1)*SIN(GAMAO2H)+CD(1)*COS(GAMAO2H)
722.                        + AB(2)*SIN(GAMAO_1H)+CD(2)*COS(GAMAO_1H));
723.              END ;
724.              ELSE
725.              DO ;
726.              RTLR(1,3)=CD(1)*SIN(GAMAO2H)-AB(1)*COS(GAMAO2H)
727.                       +EXP(-GAMA(-1))*(CD(2)*SIN(GAMAOH)-AB(2)*
728.                        COS(GAMAOH));
729.              RTLI(1,3)=-(AB(1)*SIN(GAMAO2H)+CD(1)*COS(GAMAO2H)
730.                       +EXP(-GAMA(-1))*(CD(2)*SIN(GAMAOH)+AB(2)*COS(GAMAOH)));
731.              END;
732.        /* TO COMPUTE RTL(1,4)    */
733.        /* REAL & IMAGINARY PARTS OF S(1,0)*S(0,-1) & S(1,-1)*S(-1,-1)*/
734.              DO I = 0 TO -1 BY -1 ;
735.               AB(3-I)=SCATR(1,I)*SCATR(I,-1)-SCATI(1,I)*SCATI(I,-1);
736.               CD(3-I)=SCATR(1,I)*SCATI(I,-1)+SCATI(1,I)*SCATR(I,-1);
737.              END ;
738.        /* RTL(1,4)      */
739.              IF GAMAD(-1) > 0 THEN
740.              DO ; /* RTL(1,4) WHEN GAMA(-1) IS REAL */
741.              RTLR(1,4)=-(AB(3)*COS(GAMAO_1H)-CD(3)*SIN(GAMAO_1H)
742.                        +AB(4)*COS(GAMA_12H)-CD(4)*SIN(GAMA_12H));
743.              RTLI(1,4)=-(AB(3)*SIN(GAMAO_1H)+CD(3)*COS(GAMAO_1H)
744.                        +AB(4)*SIN(GAMA_12H)+CD(4)*COS(GAMA_12H)) ;
745.              END ;
746.              ELSE
747.              DO ;   /* RTL(1,4) WHEN GAMA(-1) IS IMAGINARY    */
748.              RTLR(1,4)=EXP(-GAMA(-1))*(CD(3)*SIN(GAMAOH)-AB(3)
749.                        *COS(GAMAOH))-AB(4)*EXP(-GAMA_12H);
750.              RTLI(1,4)=-EXP(-GAMA(-1))*(AB(3)*SIN(GAMAOH)+CD(3)
751.                        *COS(GAMAOH))-CD(4)*EXP(-GAMA_12H);
752.              END ;
753.        /* TO COMPUTE RTL(2,3) AND RTL(2,4)    **/
754.              RTLR(2,3)=-(SCATR(1,0)*COS(GAMAOH)-SCATI(1,0)*SIN(GAMAOH));
755.              RTLI(2,3)=-(SCATR(1,0)*SIN(GAMAOH)+SCATI(1,0)*COS(GAMAOH));
```

56

```
756.            IF GAMAD(-1) > 0 THEN
757.            DO ;     /* RTL(2,4) WHEN GAMA(-1) IS REAL */
758.            RTLR(2,4)=-(SCATR(1,-1)*COS(GAMA_1H)-SCATI(1,-1)*
759.                       SIN(GAMA_1H));
760.            RTLI(2,4)=-(SCATR(1,-1)*SIN(GAMA_1H)+SCATI(1,-1)*COS(GAMA_1H))
   ;
761.            END ;
762.            ELSE
763.            DO ;     /* RTL(2,4) WHEN GAMA(-1) IS IMAGINARY */
764.            RTLR(2,4)=-EXP(-GAMA_1H)*SCATR(1,-1);
765.            RTLI(2,4)=-EXP(-GAMA_1H)*SCATI(1,-1);
766.            END ;
767.      /* TO COMPUTE RTL(3,3) */
768.      /* REAL & IMAGINARY PARTS OF S(0,0)**2 & S(0,-1)*S(-1,0) */
769.            DO I = 0 TO -1 BY -1 ;
770.             AB(5-I)=SCATR(0,I)*SCATR(I,0)-SCATI(0,I)*SCATI(I,0) ;
771.             CD(5-I)=SCATR(0,I)*SCATI(I,0)+SCATI(0,I)*SCATR(I,0);
772.            END ; /* END OF COMPUTING REAL & IMAGINARY PARTS OF SCAT */
773.      /* RTL(3,3) */
774.            IF GAMAD(-1) > 0 THEN
775.            DO ; /* RTL(3,3) WHEN GAMA(-1) IS REAL */
776.            RTLR(3,3)=-AB(5)*COS(GAMA02H)+CD(5)*SIN(GAMA02H)
777.                      -AB(6)*COS(GAMA0_1H)+CD(6)*SIN(GAMA0_1H)+1 ;
778.            RTLI(3,3)=-(AB(5)*SIN(GAMA02H)+CD(5)*COS(GAMA02H)
779.                      +AB(6)*SIN(GAMA0_1H)+CD(6)*COS(GAMA0_1H));
780.            END ;
781.            ELSE
782.            DO ;     /* RTL(3,3) WHEN GAMA(-1) IS IMAGINARY */
783.            RTLR(3,3)=1-AB(5)*COS(GAMA02H)+CD(5)*SIN(GAMA02H)-EXP(-GAMA_1H
   )
784.                      *(AB(6)*COS(GAMAOH)-CD(6)*SIN(GAMAOH));
785.            RTLI(3,3)=-AB(5)*SIN(GAMA02H)-CD(5)*COS(GAMA02H)
786.                      -EXP(-GAMA_1H)*(AB(6)*SIN(GAMAOH)+CD(6)*COS(GAMAOH));
787.            END ;
788.      /* TO COMPUTE RTL(3,4) */
789.      /* REAL & IMAGINARY OARTS OF S(0,0)*S(0,-1) & S(0,-1)*S(-1,-1) */
790.            DO I = 0 TO -1 BY -1 ;
791.             AB(7-I)=SCATR(0,I)*SCATR(I,-1)-SCATI(0,I)*SCATI(I,-1);
792.             CD(7-I)=SCATR(0,I)*SCATI(I,-1)+SCATI(0,I)*SCATR(I,-1);
793.            END ;
794.      /* RTL(3,4) */
795.            IF GAMAD(-1) > 0 THEN
796.            DO  /* RTL(3,4) WHEN GAMA(-1) IS REAL */
797.            RTLR(3,4)=-(AB(7)*COS(GAMA0_1H)-CD(7)*SIN(GAMA0_1H)
798.                      +AB(8)*COS(GAMA_12H)-CD(8)*SIN(GAMA_12H));
799.            RTLI(3,4)=-(AB(7)*SIN(GAMA0_1H)+CD(7)*COS(GAMA0_1H)
800.                      +AB(8)*SIN(GAMA_12H)+CD(8)*COS(GAMA_12H));
801.            END ;
802.            ELSE
803.            DO ;     /* RTL(3,4) WHEN GAMA(-1) IS IMAGINARY */
804.            RTLR(3,4)=EXP(-GAMA_1H)*(CD(7)*SIN(GAMAOH)-AB(7)*COS(GAMAOH))
805.                      -EXP(-GAMA_12H)*AB(8);
806.            RTLI(3,4)=-EXP(-GAMA_1H)*(AB(7)*SIN(GAMAOH)+CD(7)*
807.                      COS(GAMAOH))-CD(8)*EXP(-GAMA_12H) ;
808.            END ;
809.      /* TO COMPUTE RTL(4,3) */
810.      /* REAL & IMAGINARY PARTS OF S(-1,0)*S(0,0) & S(-1,-1)*S(-1,0) */
811.            DO I = 0 TO -1 BY -1 ;
812.             AB(9-I)=SCATR(-1,I)*SCATR(I,0)-SCATI(-1,I)*SCATI(I,0) ;
813.             CD(9-I)=SCATR(-1,I)*SCATI(I,0)+SCATI(-1,I)*SCATR(I,0);
814.            END ;
815.      /* RTL(4,3) */
816.            IF GAMAD(-1) > 0 THEN
817.            DO ;     /* RTL(4,3) WHEN GAMA(-1) IS REAL */
818.            RTLR(4,3)=-(AB(9)*COS(GAMA02H)-CD(9)*SIN(GAMA02H)
819.                      +AB(10)*COS(GAMA0 1H)-CD(10)*SIN(GAMA0 1H));
```

```
820.        RTLI(4,3)=-(AB(9)*SIN(GAMAO2H)+CD(9)*COS(GAMAO2H)
821.               +AB(10)*SIN(GAMAO_1H)+CD(10)*COS(GAMAO_1H));
822.        END ;
823.        ELSE
824.        DO ;    /* RTL(4,3) WHEN GAMA(-1) IS IMAGINARY  */
825.        RTLR(4,3)=-AB(9)*COS(GAMAO2H)+CD(9)*SIN(GAMAO2H)
826.               -EXP(-GAMA_1H)*(AB(10)*COS(GAMAOH)-CD(10)*SIN(GAMAOH));
827.        RTLI(4,3)=-AB(9)*SIN(GAMAO2H)-CD(9)*COS(GAMAO2H)
828.               -EXP(-GAMA_1H)*(AB(10)*SIN(GAMAOH)+CD(10)*COS(GAMAOH));
829.        END ;
830.   /* TO COMPUTE RTL(4,4)  */
831.   /* REAL & IMAGINARY PARTS OF S(-1,0)*S(0,-1) & S(-1,-1)**2 */
832.        DO I = 0 TO -1 BY -1 ;
833.          AB(11-I)=SCATR(-1,I)*SCATR(I,-1)-SCATI(-1,I)*SCATI(I,-1);
834.          CD(11-I)=SCATR(-1,I)*SCATI(I,-1)+SCATI(-1,I)*SCATR(I,-1);
835.        END ;
836.   /* RTL(4,4)  */
837.        IF GAMAD(-1) > 0 THEN
838.        DO ;   /* RTL(4,4) WHEN GAMA(-1) IS REAL  */
839.        RTLR(4,4)=-(AB(11)*COS(GAMAO_1H)-CD(11)*SIN(GAMAO_1H)
840.                +AB(12)*COS(GAMA_12H)-CD(12)*SIN(GAMA_12H))+1;
841.        RTLI(4,4)=-(AB(11)*SIN(GAMAO_1H)+CD(11)*COS(GAMAO_1H)
842.                +AB(12)*SIN(GAMA_12H)+CD(12)*COS(GAMA_12H));
843.        END ;
844.        ELSE
845.        DO ;  /* RTL(4,4) WHEN GAMA(-1) IS IMAGINARY  */
846.        RTLR(4,4)=1-EXP(-GAMA_1H)*(AB(11)*COS(GAMAOH)-CD(11)*
847.            SIN(GAMAOH))-EXP(-GAMA_12H)*AB(12);
848.          RTLI(4,4)=-EXP(-GAMA_1H)*(AB(11)*SIN(GAMAOH)+CD(11)*
849.              COS(GAMAOH))-EXP(-GAMA_12H)*CD(12) ;
850.          END ;
851.   /* END OF COMPUTING COEFFICIENT MATRIX RTL FOR SOLVING */
852.   /* R AND RPRIME, T(0) AND T(-1) IN A SYSTEM OF EQUATIONS */
853.   /* TO PRINT ELEMENTS OF RTL MATRIX */
854.        PUT SKIP ;
855.        PUT SKIP EDIT(REPEAT('*',55))(X(3),A) ;
856.        PUT SKIP EDIT('PRINT COEFFICIENT MATRIX RTL')(X(10),A);
857.        PUT SKIP EDIT(REPEAT('*',55))(X(3),A) ;
858.        PUT SKIP ;
859.        DO I = 1 TO 4 ;
860.        PUT SKIP;
861.        PUT SKIP EDIT('I=',I)(X(10),A,F(2,0));
862.        PUT SKIP EDIT(REPEAT('*',10))(X(5),A) ;
863.        PUT SKIP ;
864.        DO J = 1 TO 4 ;
865.        PUT SKIP EDIT('RTLR(',I,',',J,')=',RTLR(I,J),'RTLI(',I,',',
866.                      J,')=',RTLI(I,J))
867.                      (X(5),2 (X(2),A,F(2,0),A,F(2,0),A,E(12,5)));
868.        END ; /* END OF PRINTING RTL(I,J) FOR I,J = 1 TO 4 */
868.1         END ;    /* END OF PRINTING RTL(I,J) FOR I,J = 1 TO 4 */
869.        END RTL ;
898.    /* TO COMPUTE R,RPRIME,T(0),AND T(-1)              */
904.         MM=8;
905.         NN= 8;
906.         PIVOT= 1;
907.         RS=1 ;
908.        AAIII1: DO I=1 TO NN/2;
909.        AAJJ1: DO J= 1 TO MM/2;
910.               AA(I,J)=RTLR(I,J);
911.               AA(I,J+NN/2)=-RTLI(I,J);
911.1            END AAJJ1 ;
912.             END AAIII1;
914.        AAII2: DO I= NN/2+1 TO NN ;
915.        AAJJ2: DO J= 1 TO MM/2 ;
916.               AA(I,J)=RTLI(I-NN/2,J);
917.               AA(I,J+NN/2)=RTLR(I-NN/2,J);
```

```
918.                    END AAJJ2;
919.                    END AAII2;
920.        /* RIHGT HAND VECTOR FOR THE SYSTEM OF EQUATIONS      */
921.        /* TO SOLVE FOR R, RPRIME, T(0),& T(-1)               */
922.                    AA(1,MM+1)=SCATR(1,1);
923.                      AA(5,MM+1)=SCATI(1,1);
924.                    AA(2,MM+1)=0 ;
925.                    AA(6,MM+1)=0 ;
926.            RHSCAT: DO I = 3 TO NN/2;
927.                    AA(I,MM+1)=SCATR(3-I,1);
928.                    AA(I+NN/2,MM+1)=SCATI(3-I,1) ;
929.                    END RHSCAT ;
930.                    CALL GAUSS(AA,MM,NN,PIVOT,RS,X);
1042.                  /*                                          */
1044.              RREAL=X(1,1); /* REAL PART OF R */
1045.              RIMAG=X(1,5) ; /* IMAGINARY PART OF R */
1046.              RPRIMER=X(1,2); /* REAL PART OF R PRIME */
1047.              RPRIMEI=X(1,6);  /* IMAGINARY PART OF R PRIME */
1048.              TOR = X(1,3);     /* REAL PART OF (1,0)    */
1049.              TOI= X(1,7) ;      /* IMAGINARY PART OF (1,0) */
1050.              TMINUS1R=X(1,4); /* REAL PART OF (1,-1) */
1051.              TMINUS1I=X(1,8); /* IMAGINARY PART OF (1,-1) */
1052.              RABS=SQRT(RREAL**2+RIMAG**2);
1053.              RPHASE=ATAND(RIMAG/RREAL); /* PHASE ANGLE OF R */
1054.              RPRABS=SQRT(RPRIMER**2+RPRIMEI**2);
1055.              RPRPHASE=ATAND(RPRIMER/RPRIMEI); /*PHASE OF RPRIME */
1056.        /* TO PRINT R */
1057.              PUT SKIP; PUT SKIP ;
1058.              PUT SKIP EDIT('REAL OF R =',RREAL,'IMAGINARY OF R=',RIMAG)
1059.                            (X(3),2 (X(2),A,E(12,5)));
1060.               PUT SKIP ;
1061.              PUT SKIP EDIT('ABS(R)=',RABS,'PHASE(R)=',RPHASE)
1062.                            (X(3),2 (X(2),A,E(12,5)));
1063.        /* TO PRINT R PRIME */
1064.              PUT SKIP;
1065.              PUT SKIP EDIT("REAL PART OF R PRIME=',RPRIMER,
1066.                            'IMAGINARY PART OF R PRIME=',RPRIMEI)
1067.                            (X(3),2 (X(2),A,E(12,5)));
1068.              PUT SKIP;
1069.              PUT SKIP EDIT('ABS(RPRIME)=',RPRABS,'PHASE(RPRIME)=',
1070.                            RPRPHASE)(X(3),2 (X(2),A,E(12,5)));
1071.              PUT SKIP;
1072.        /* TO PTINT T(0) */
1073.              PUT SKIP;
1074.              PUT SKIP EDIT('REAL PART OF T(0)=',TOR,
1075.                            'IMAGINARY PART OF T(0)=',TOI)
1076.                            (X(3),2 (X(2),A,E(12,5)));
1077.              PUT SKIP ;
1078.          PUT SKIP EDIT('REAL PART OF T(-1)=',TMINUS1R,
1079.                        'IMAGINARY PART OF T(-1)=',TMINUS1I)
1080.                        (X(3),2 (X(2),A,E(12,5)));
1110.    END TEMODE;
1111.    /*
1112.    //GO.SYSIN DD *
1113.    .54,.27,.27,2.56,1.44,1.000,45,1.1
1114.    -6,-5,-4,-3,-2,-1,0,1,2,3,4,5
1115.    /*
1116.    //
```