GEORGIA INSTITUTE OF TECHNOLOGY | OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[X] ORIGINAL [ ] REVISION NO. ______

Project No. E-21-F03 R6194-0A0 GTRC/~~GIT~~ XXX DATE 9 /4 / 86

Project Director: Dr. Kent R. Davey School/~~Lab~~ XXX EE

Sponsor: Office of Naval Research, Arlington, VA

Type Agreement: Contract N00014-86-K-0532

Award Period: From 7/1/86 To 9/30/87 (Performance) 11/30/87 (Reports)

Sponsor Amount:

	This Change	Total to Date
Estimated:	$	$ 60,000
Funded:	$	$ 60,000

Cost Sharing Amount: $ ______ Cost Sharing No: ______

Title: Target Discrimination by the Prediction and Analysis of Natural Resonances

ADMINISTRATIVE DATA OCA Contact E. Faith Gleason X-4820

1) Sponsor Technical Contact:

Dr. Michael Morgan
Office of Naval Research
800 North Quincey Street, Code 1114SE
Arlington, VA 22217-5000

2) Sponsor Admin/Contractual Matters:

G. Daniel Oldre
ONRRR

Defense Priority Rating: N/A

Military Security Classification: N/A
(or) Company/Industrial Proprietary: ______

RESTRICTIONS

See Attached Government Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval — Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with GIT, provided prior approval received from Contracting Officer and unless determination regarding vesting of title is deferred until after acquisition.

COMMENTS:

SEP 1986 RECEIVED OCA

COPIES TO:

SPONSOR'S I. D. NO. 02.103.000.86.028

Project Director	Procurement/GTRI Supply Services	GTRC
Research Administrative Network	Research Security Services	Library
Research Property Management	Reports Coordinator (OCA)	Project File
Accounting	Research Communications (2)	Other A. Jones

FORM OCA 65:285

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION/CLOSEOUT SHEET

Date 12/30/87

Project No. E-21-F03 School/~~Lab~~ EE

Includes Subproject No.(s) N/A

Project Director(s) K. R. Davey GTRC / ~~GIT~~

Sponsor Office of Naval Research, Arlington, VA

Title Target Discrimination by the Prediction and Analysis of Natural Resonances

Effective Completion Date: 9/30/87 (Performance) 11/30/87 (Reports)

Grant/Contract Closeout Actions Remaining:

- [ ] None
- [x] Final Invoice or Final Fiscal Report
- [x] Closing Documents
- [x] Final Report of Inventions Questionnaire sent to P.I.
- [x] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other

Continues Project No. Continued by Project No.

COPIES TO:

Project Director
Research Administrative Network
Research Property Management
Accounting
Procurement/GTRI Supply Services
Research Security Services
Reports Coordinator (OCA)
Legal Services

Library
GTRC
Research Communications (2)
Project File
Other Duane Hutchison
Angela DuBose
Russ Embry

FORM OCA 69.285

Year to Date Report on

Electromagnetic Scattering from Dielectric Bodies Using the Resonance Technique

Contract E21-F03

June 1987

by

Kent Davey
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

The objective of this research is target identification. Specifically, we wish to predict an object's shape and composition from its scattering signature. When a dielectric object is interrogated by a short time radar pulse, the reflected signal has two components. One component decays rather slowly in time and oscillates at a very high frequency. The second component decays rapidly in time and has a low real frequency component. That part of the scattered signal which decays slowly in time is thought to be related directly to the composition of the object. That part of the signature which decays more rapidly, having the slower real frequency component, is thought to be linked to the object's shape. We have sought to address this thesis by examining the scattering off a dielectric coated circular cylinder. Of particular interest is the degree to which the effects of shape and composition are separated in the location of the complex resonance poles. The results to date do not show clear separation of the two effects, i.e., shape and composition. It is thought that it may be necessary to examine a ray solution of the backscattered field from the dielectric coated cylinder via the Watson transformation in order to more accurately predict the scattering field near resonance. This is necessitated by the slow convergence of the terms composing the eigenfunction solution. Consider an infinitely long, perfectly conducting circular cylinder of radius "a" coated with a homogeneous dielectric layer of thickness T. The dielectric material has a relative dielectric constant ε_r. The cylinders are illuminated with a plane wave (of $e^{j\omega t}$ time suppressed). The geometry of the problem is illustrated in Figure 1. Both the transverse electric and transverse magnetic cases will be listed below. In the transverse electric case, the incident magnetic field is parallel with the axis of the cylinder, whereas in the transverse magnetic case, the incident electric field is parallel to the axis of the cylinder.

The eigenfunction solution for the far zone backscattered field can be written as follows:

$$U_z^s \approx \sqrt{\frac{2}{\pi}}\, \frac{e^{jk_o\rho}}{\sqrt{k_o\rho}}\, e^{j\pi/4} \sum_{n=0}^{\infty} \varepsilon_n (-1)^n A_n \tag{1}$$

where

$$A_n = -\frac{N(n)}{D(n)} = -\frac{J_n'(k_ob) + G_nJ_n(k_ob)}{H_n^{(2)'}(k_ob) + G_nH_n^{(2)}(k_ob)} \tag{2}$$

$$G_n = -\frac{1}{\sqrt{\varepsilon_r}}\, \frac{J_n'(kb)N_n'(ka) - J_n'(ka)N_n'(kb)}{J_n(kb)N_n'(ka) - J_n'(ka)N_n(kb)}\,,\quad \text{TE} \tag{3a}$$

$$G_n = -\sqrt{\varepsilon_r}\, \frac{J_n(ka)N_n'(kb) - N_n(ka)J_n'(kb)}{J_n(ka)N_n(kb) - N_n(ka)J_n(kb)}\,,\quad \text{TM} \tag{3b}$$

In the above equations, $k_o = \omega\sqrt{\mu_o\varepsilon_o}$, $k = k_o\sqrt{\varepsilon_r}$, and $b = a + T$. N_n, $H_n^{(2)}$ are the cylindrical Bessel functions, and J_n', N_n', $H_n^{(2)'}$ are the derivatives taken with respect to the argument.

Using the usual Watson's transformation technique, the eigenfunction solution can be cast into a ray solution. Without presenting the details, the final expressions for the far-zone backscattered field is given by

$$U_z^s - U_z^{GO} + U_z^{SW} \tag{4}$$

where

$$U_z^{GO} \sim \frac{e^{-jk_o\rho}}{\sqrt{\rho}}\, R \sqrt{\frac{b}{2}}\, e^{j2k_ob} \tag{5}$$

$$R = \mp \frac{j\tan(kt) - \sqrt{\varepsilon_r}}{j\tan(kt) + \sqrt{\varepsilon_r}} \quad \begin{matrix}\text{TE}\\ \text{TM}\end{matrix} \tag{6}$$

and

$$U_z^{SW} \sim \frac{e^{-jk_o\rho}}{\sqrt{\rho}}\, 4e^{-j\frac{\pi}{4}} \sum_{\ell=1}^{\infty} \left(-\frac{N(\nu)}{\frac{\partial}{\partial\nu}D(\nu)}\right)_{\nu=\nu_\ell} \frac{e^{-j\nu_\ell\pi}}{1 - e^{-j\nu_\ell 2\pi}} \tag{7}$$

In the above, U_z^{GO} is the geometrical-optic field, R is the reflection coefficient for a planar grounded slab illuminating with an normally incident plane wave, and U_z^{SW} is the contribution from two surface waves as illustrated in Figure 1. In (7), v_ℓ are the roots of the transcendental equation

$$D(v_\ell) = H_{v_\ell}^{(2)}(k_o b) + G_{v_\ell} H_{v_\ell}^{(2)}(k_o b) = 0 \tag{8}$$

The factor $[1 - e^{-jv_\ell 2\pi}]^{-1}$ appearing in (7) results from the fact that all the multiply encircling surface waves are included in the analysis. The factor is important and is needed to predict the resonance phenomena.

We have concentrated to date on equation (3a). We've sought the complex roots for k, the wave number, using the following technique. Let C denote a closed contour in the complex z plane containing n zeros z_j $(j = 1,2,...,n)$ of the analytic function f(z). Then from Cauchy's integral

$$\frac{1}{2\pi i} \int_C z^p \frac{f'(z)}{f(z)} dz = \sum_{j=1}^{n} z_j^p \equiv s_p , \tag{9}$$

where p is any non-negative integer. In particular, if $p = 0$, then the integral (9) furnishes an estimate of the number, n, of zeros within C. The integral (9) is also evaluated with $p = 1,2,...,n$ to give sums s_p. If the number of zeros is small (say $n < 4$), these are used with Newton's formulae to produce a polynomial of degree n (with leading coefficient one) having the same roots within C as f(z). Specifically, the polynomial is written as

$$p_n(z) = \sum_{k=0}^{n} a_k z^k \tag{10}$$

with $a_n = 1$. The coefficients a_k for $k = n - 1, n - 2, \ldots, 1, 0$ are evaluated using the recurrence relations

$$(n-k)a_k + s_1 a_{k+1} + s_2 a_{k+2} + \ldots + s_{n-k} a_n = 0 \, . \tag{11}$$

The low order polynomial equation $p_n(z) = 0$ is solved by a standard technique (in our case, Müeller's method) to give numerical estimates $\{z_i'\}$ of the actual set of zeros $\{z_i\}$. The differences $\{z_i - z_i'\}$ are due predominantly to quadrature errors. The estimates $\{z_i'\}$ are then refined by a further application of the Newton Raphson method, this time to the original function $f(z)$.

Figures 2-4 illustrate the change in the complex roots as the thickness of the dielectric coating is varied. Figures 5-7 illustrate the movement of the complex poles as the relative dielectric constant of a coating is varied, keeping the thickness fixed. In all figures, equation (1) is solved keeping the first four terms of the series.

In each case, the arrows show the progression of the complex wavenumber "k" poles from thinest coating (or smallest dielectric constant) to thickest coating (or largest dielectric constant). The abseissa represents the real part of the wavenumber k, the ordinate represents the imaginary part. As expected, the roots are always symmetric about the imaginary axis. The real part shows the resonance frequency while the imaginary part yields the damping rate. The n = 0,1,2,3 roots are designated respectively by squares, triangles, diamonds, and x's. Except for a few exceptions of thickness changes, the general trend is that the resonance wave becomes less damped as the thickness increases and as the dielectric constant increases. For the same changes the resonance frequency decreases. This can be understood from the fact that in both cases, surface waves are more strongly trapped by the presence of the coating, and as a consequence, the surface waves creep along the cylinder surface with less attenuation. There does not appear to be an easy way to separate out the effects of shape and composition from the graphs.

Thus, it appears that both the composition and the shape affect the position of these poles. However, the series of equation (1) converges quite poorly when the product $k_o b$ is not small. This is of course the motivating force behind the Watson transformation. With the scattered field expressed according to equation (7), this solution convergers very rapidly. It is thought that a better prediction of the complex resonances can be realized through the solution of (7) and (8). We will attempt to seek that solution using the asymptotic formula for the Hankel function below.

$$H^{(2)}_{\nu_\ell}(kp) \simeq \frac{2}{\sqrt{\pi(k^2p^2-\nu_\ell^2)}} \exp[\, i(\sqrt{k^2p^2 - \nu_\ell^2} - \nu_\ell \cos^{-1}\frac{\nu_\ell}{kp} - \frac{\pi}{4})] \tag{12}$$

This will be the center of our attention for the next three months of the contract period.

The method is based on a technique suggested by Delves and Lyness (Math. Computations, 21, 1967, p. 543). Let C denote a closed contour in the complex z plane containing n zeros z_j $(j = 1,2,\ldots,n)$ of the analytic function $f(z)$. Then from Cauchy's integral

$$\frac{1}{2\pi i}\int_C z^p \frac{f'(z)}{f(z)}\, dz = \sum_{j=1}^{n} z_j^p \equiv s_p \,, \tag{13}$$

where p is any non-negative integer. In particular, if $p = 0$, then the integral (13) furnishes an estimate of the number, n, of zeros within C. The integral (13) is also evaluated with $p = 1,2,\ldots,n$ to give sums s_p. If the number of zeros is small (say $n < 4$), these are used with Newton's formulae to produce a polynomial of degree n (with leading coefficient one) having the same roots within C as f(z). Specifically, the polynomial is written as

$$p_n(z) = \sum_{k=0}^{n} a_k z^k \tag{14}$$

with $a_n = 1$. The coefficients a_k for $k = n-1, n-2, \ldots, 1, 0$ are evaluated using the recurrence relations

$$(n-k)a_k + s_1 a_{k+1} + s_2 a_{k+2} + \ldots + s_{n-k} a_n = 0 \tag{15}$$

The low-order polynomial equation $p_n(z) = 0$ is solved by a standard technique (in our case, Müller's method) to give numerical estimates $\{z_i'\}$ of the actual set of zeros $\{z_i\}$. The differences $\{z_i - z_i'\}$ are due predominantly to quadrature errors. The estimates $\{z_i'\}$ are then refined by a further application of the Newton/Raphson method, this time to the original function f(z).

$$H_{v_\ell}^{(2)}(kp) \simeq \sqrt{\frac{2}{\pi\left(k^2p^2-v_\ell^2\right)^{1/2}}}\,\exp\left[i\left(\sqrt{k^2p^2-v_\ell^2} - v_\ell \cos^{-1}\frac{v_\ell}{kp} - \frac{\pi}{4}\right)\right]$$

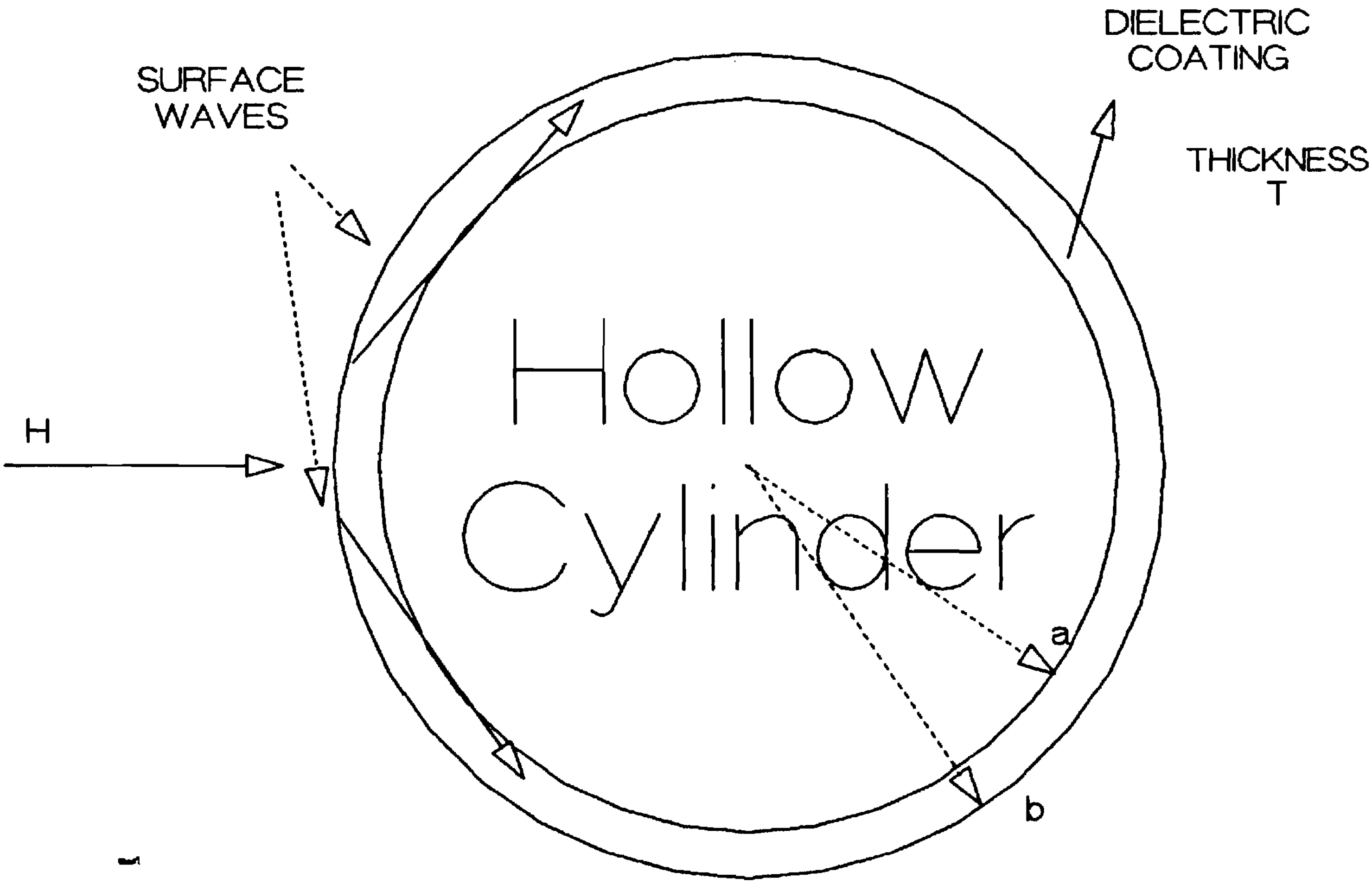

FIG. 1 SCATTERING OFF A DIELECTRIC COATED CYLINDER

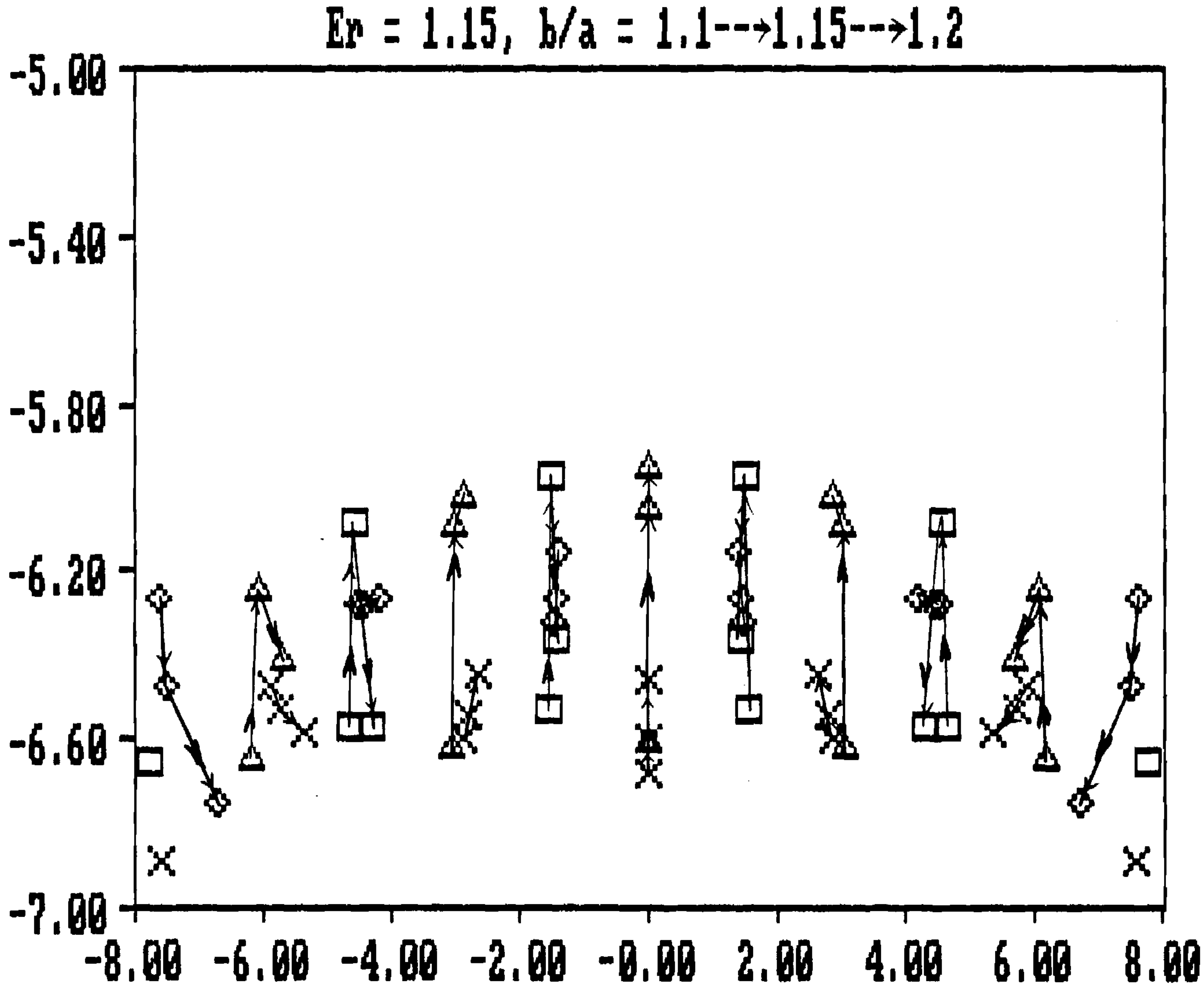

Figure 2. Complex wavenumber (k) poles with $\varepsilon_r = 1.15$ for thickness ratios b/a = 1.1, 1.15, 1.2. Plotted are modes n = 0,1,2,3 as squares, triangles, diamonds, and x's, respectively. Arrows designate pole changes.

□ n=0
△ n=1
◇ n=2
X n=3

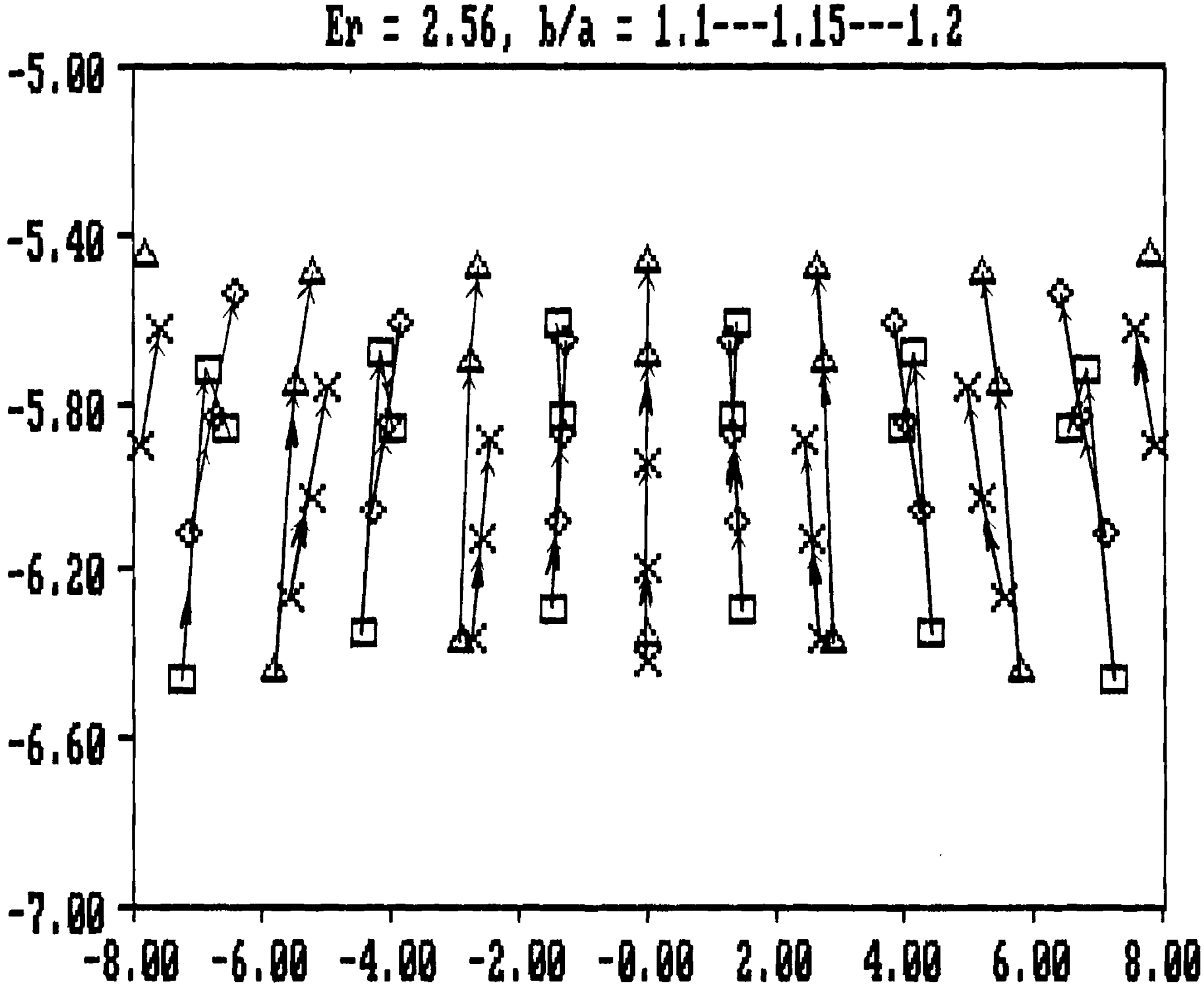

Figure 3. Complex wavenumber (k) poles with $\epsilon_r = 2.56$ for thickness ratios b/a = 1.1, 1.15, 1.2. Plotted are modes n = 0,1,2,3 as squares, triangles, diamonds, and x's, respectively. Arrows designate pole changes.

□ = rebounds

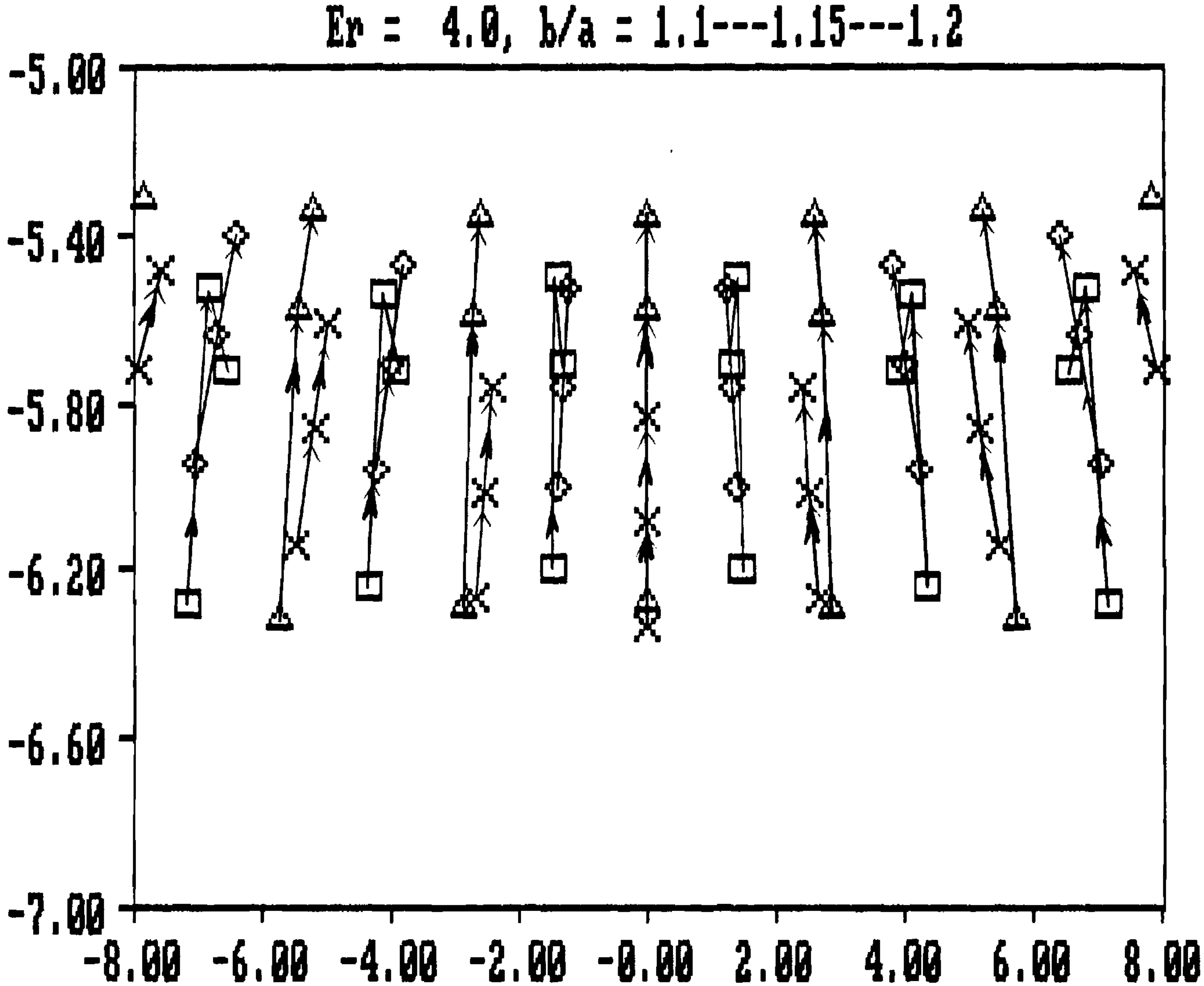

Figure 4. Complex wavenumber (k) poles with $\varepsilon_r = 4.0$ for thickness ratios b/a = 1.1, 1.15, 1.2. Plotted are modes n = 0,1,2,3 as squares, triangles, diamonds, and x's, respectively. Arrows designate pole changes.

□ = rebounds

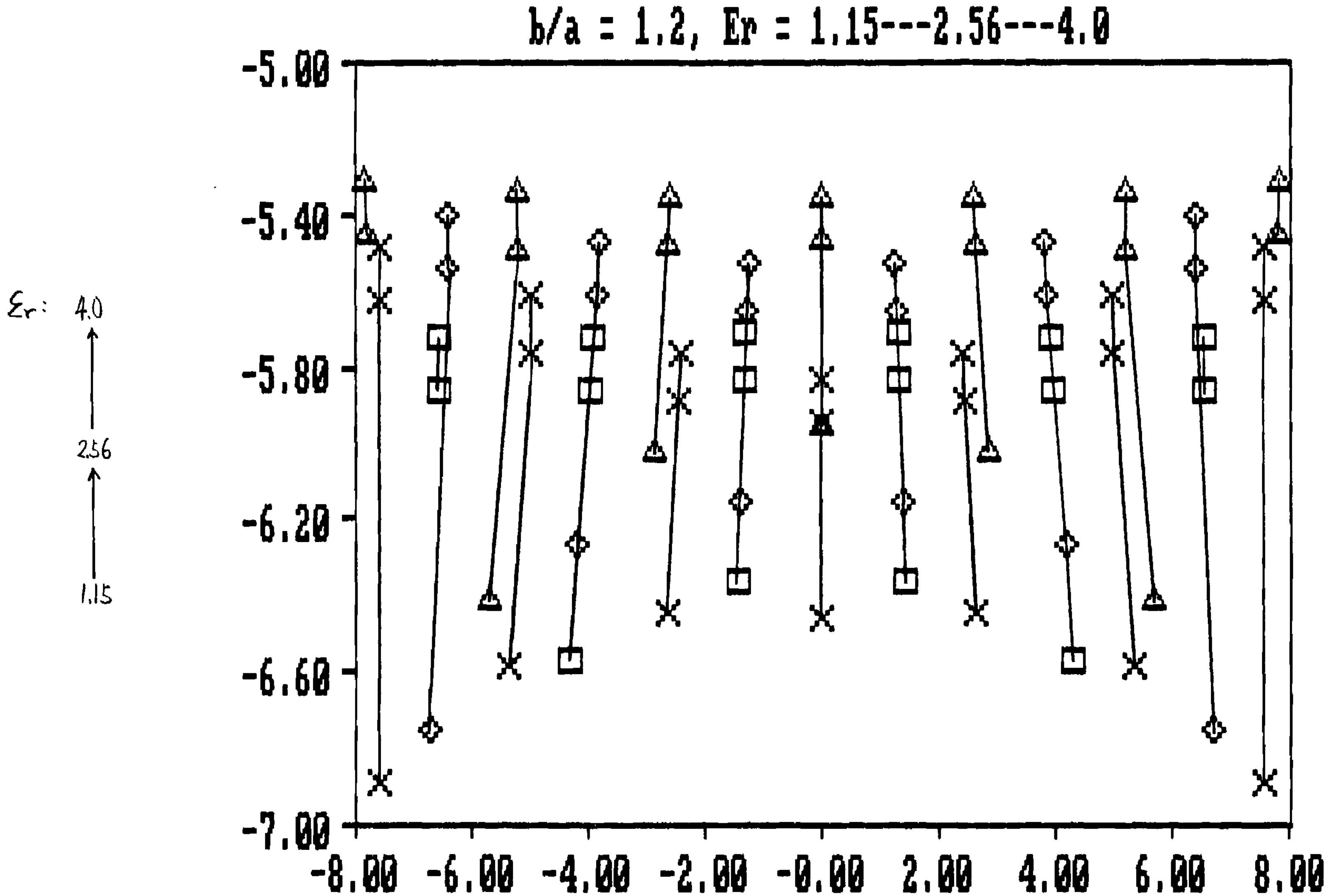

Figure 5. Complex wavenumber (k) poles for thickness ratio b/a = 1.2 with dielectric constants ε_r = 1.15, 2.56, 4.0.

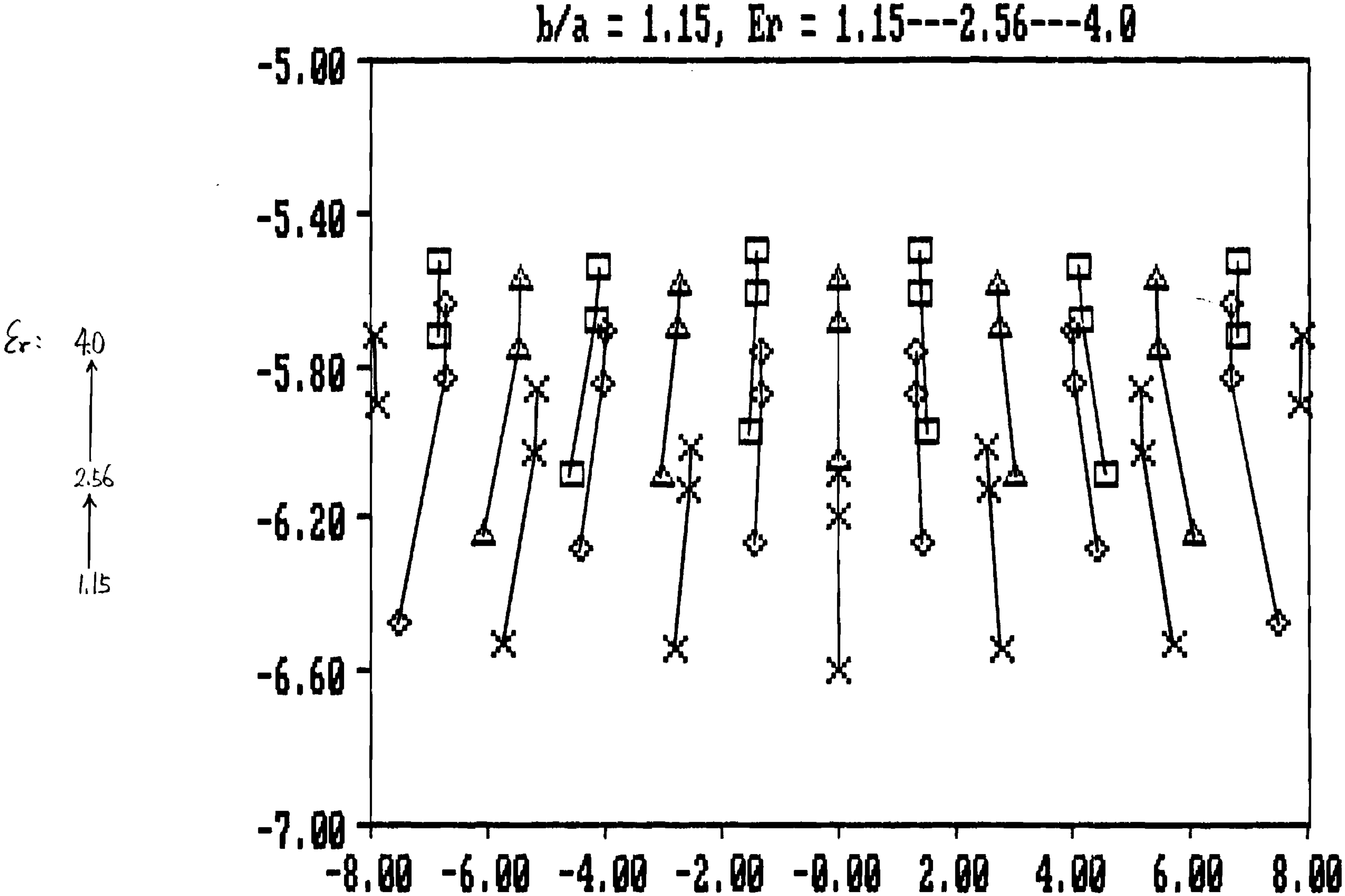

Figure 6. Complex wavenumber (k) poles for thickness ratio b/a = 1.15 with dielectric constants ε_r = 1.15, 2.56, 4.0.

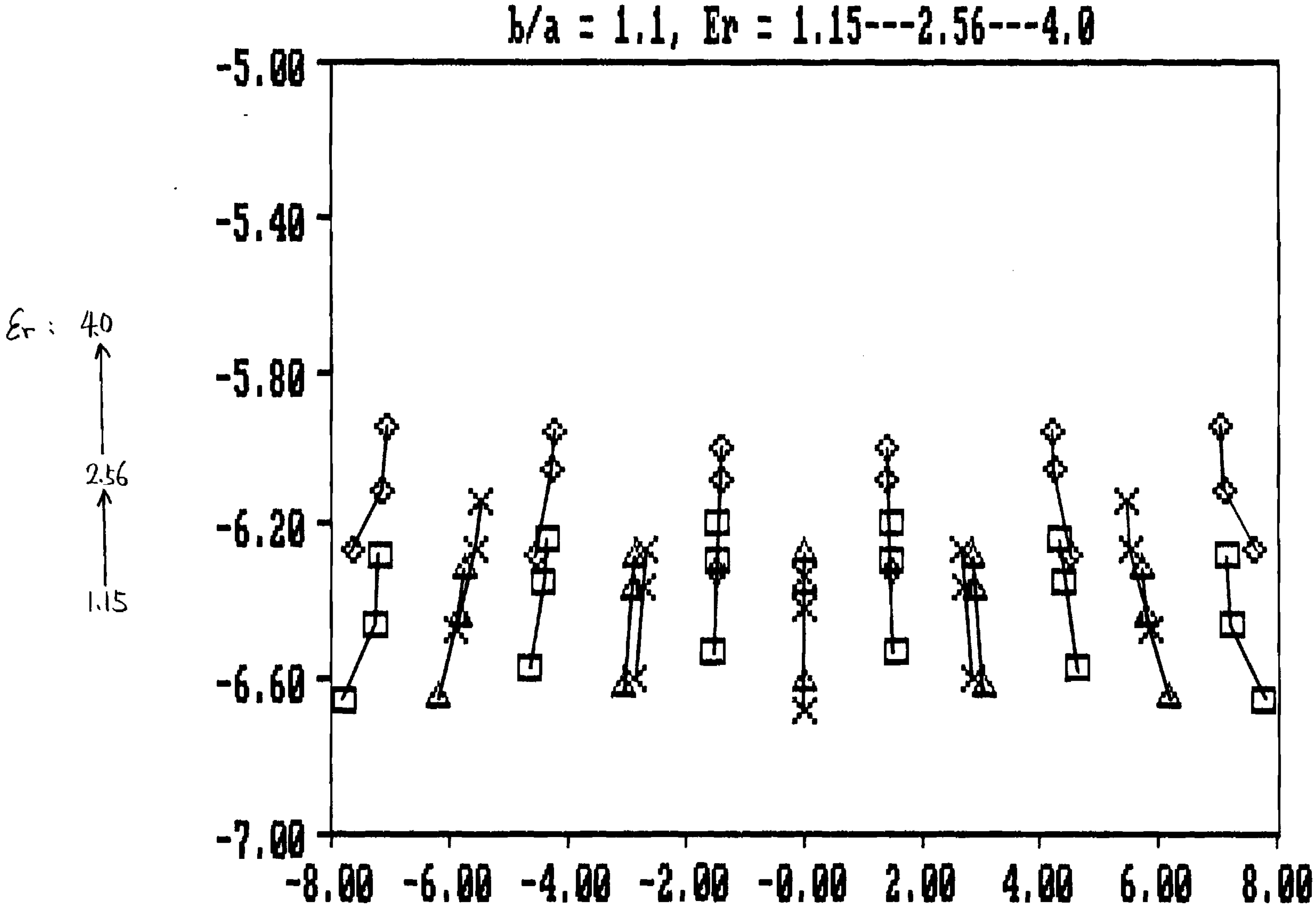

Figure 7. Complex wavenumber (k) poles for thickness ratio b/a = 1.1 with dielectric constants ε_r = 1.15, 2.56, 4.0.

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894-2961

December 10, 1987

Dr. Michael Morgan
Office of Naval Research
800 North Quincey Street, Code 1114SE
Arlington, VA 22217-5000

RE: Final Report, Contract N00014-86-K-0532
Project No. E-21-F03

Dear Dr. Morgan:

The subject report is forwarded in conformance with contract specifications. Should you have questions or comments regarding this report, please call me at 404-894-2961.

Sincerely,

Kathy Knighton
Research Administrator

Enclosures

Final Report on

Radar Target Discrimination
Using the Resonance Technique

Contract E21-F03

October 1987

by

Kent Davey
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

ABSTRACT

The intent of the research was to investigate the possibility of determining an object's shape and composition given its scattering signature. When an object is interrogated by a short time sequence RF pulse, the frequency and characteristic decay of the resonance signal dictate the complex frequency poles (the frequency yielding us the real part of the pole and the characteristic decay the imaginary part). The research for this project is divided into two areas. First, given an object, what are the complex resonance poles and how can they be efficiently numerically computed? Second, how feasible is the inverse problem; that is, can the shape and composition be inferred easily once the poles are known?

The program was tasted on a dielectric coated cylinder. Based on the scattering equations for the wedge, the complex poles were calculated using an integral technique coupled with the Muller algorithm for obtaining complex zeros of a polynomial. The roots were compiled for various thicknesses of the coating and dielectric values. A program was then written to predict the best least squares approximation to a cylinder's thickness and dielectric coating given its complex poles.

The results indicate that unlike the acoustic world, the complex electromagnetic scattering poles do not separate in terms of shape dependent and composition dependent poles. The determination of these parameters in a generalized inverse is thus very difficult, especially when multiple geometric and composition factors are involved.

INTRODUCTION

The objective of this research is target identification. The specific goal is to predict an object's shape and composition from its scattering signature. When a penetrable object is interrogated by a short time sonic pulse, the reflected signal has two components. One component decays rather slowly in time and oscillates at a very high frequency. The second component decays rapidly in time and has a low real frequency component. That part of the scattered signal which decays slowly in time is thought to be related directly to the composition of the object. That part of the signature which decays more rapidly, having the slower real frequency component, is thought to be linked to the object's shape.

It is suggested that this phenomenon also occurs at radar frequencies in electromagnetic scattering. We have sought to address this thesis by examining the scattering off a dielectric coated circular cylinder. Of particular interest is the degree to which the effects of shape and composition are separated in the location of the complex resonance poles. The results to date do not show clear separation of the two effects, i.e., shape and composition. It may be necessary to examine a ray solution of the backscattered field from the dielectric coated cylinder via the Watson transformation in order to more accurately predict the scattering field near resonance. This is necessitated by the slow convergence of the terms composing the eigenfunction solution. Nevertheless, given a set of complex poles, it is possible to give a least squares prediction of the object shape and composition. This is shown for the thickness and dielectric constant of a dielectrically coated cylinder.

THEORY

Consider an infinitely long, perfectly conducting circular cylinder of radius "a" coated with a homogeneous dielectric layer of thickness T (Figure 1). The dielectric material has a relative dielectric constant ε_r. The cylinders are illuminated with a plane wave (of $e^{j\omega t}$ time dependence suppressed). The geometry of the problem is illustrated in Figure 1. Both the transverse electric and transverse magnetic cases will be listed below. In the transverse electric case, the incident magnetic field is parallel with the axis of the cylinder, whereas in the transverse magnetic case, the incident electric field is parallel to the axis of the cylinder.

The eigenfunction solution for the far zone backscattered field can be written as follows:

$$U_z^s \approx \sqrt{\frac{2}{\pi}} \frac{e^{jk_o\rho}}{\sqrt{k_o\rho}} e^{j\pi/4} \sum_{n=0}^{\infty} \varepsilon_n (-1)^n A_n \tag{1}$$

where

$$A_n = -\frac{N(n)}{D(n)} = -\frac{J_n'(k_o b) + G_n J_n(k_o b)}{H_n^{(2)'}(k_o b) + G_n H_n^{(2)}(k_o b)} \tag{2}$$

$$G_n = -\frac{1}{\sqrt{\varepsilon_r}} \frac{J_n'(kb)N_n'(ka) - J_n'(ka)N_n'(kb)}{J_n(kb)N_n'(ka) - J_n'(ka)N_n(kb)}, \quad \text{TE} \tag{3a}$$

$$G_n = -\sqrt{\varepsilon_r} \frac{J_n(ka)N_n'(kb) - N_n(ka)J_n'(kb)}{J_n(ka)N_n(kb) - N_n(ka)J_n(kb)}, \quad \text{TM} \tag{3b}$$

In the above equations, $k_o = \omega\sqrt{\mu_o\varepsilon_o}$, $k = k_o\sqrt{\varepsilon_r}$, and $b = a + T$. N_n, $H_n^{(2)}$ are the cylindrical Bessel functions, and J_n', N_n', $H_n^{(2)'}$ are the derivatives taken with respect to the argument.

Using the usual Watson's transformation technique, the eigenfunction solution can be cast into a ray solution. The final expressions for the far-zone backscattered field is given by

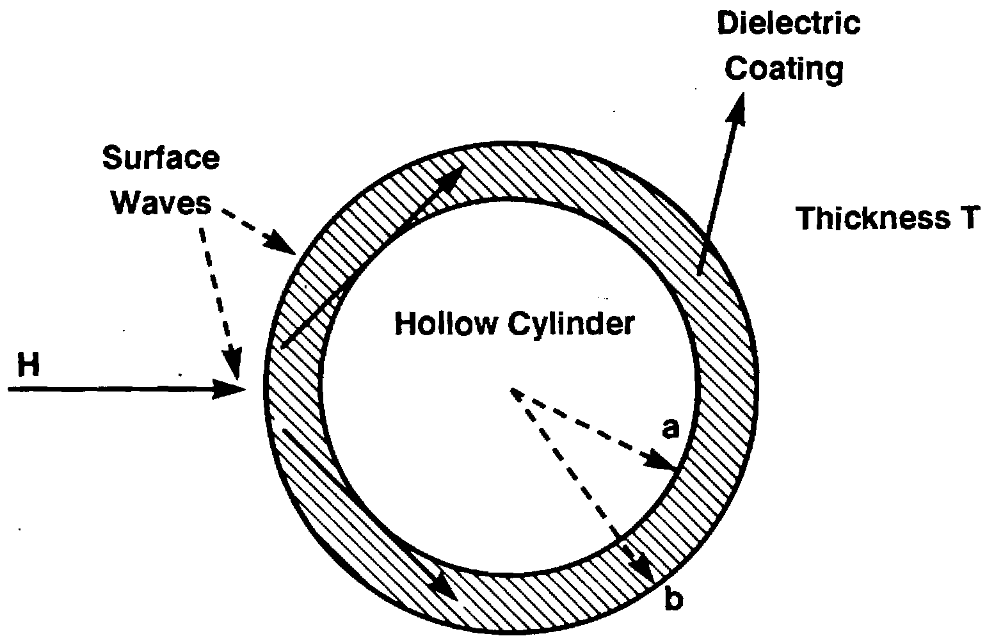

Figure 1. Scattering off a dielectric coated cylinder.

$$U_z^S = U_z^{GO} + U_z^{SW} \tag{4}$$

where

$$U_z^{GO} \sim \frac{e^{-jk_o\rho}}{\sqrt{\rho}} R \sqrt{\frac{b}{2}}\, e^{j2k_ob} \tag{5}$$

$$R = \mp \frac{j\tan(kt) - \sqrt{\varepsilon_r}}{j\tan(kt) + \sqrt{\varepsilon_r}} \quad \begin{matrix} TE \\ TM \end{matrix} \tag{6}$$

and

$$U_z^{SW} \sim \frac{e^{-jk_o\rho}}{\sqrt{\rho}} 4e^{-j\frac{\pi}{4}} \sum_{\ell=1}^{\infty} - \left. \frac{N(v)}{\frac{\partial}{\partial v} D(v)} \right|_{v=v_\ell} \frac{e^{-jv_\ell \pi}}{1 - e^{-jv_\ell 2\pi}} \tag{7}$$

In the above, U_z^{GO} is the geometrical-optic field, R is the reflection coefficient for a planar grounded slab illuminating with an normally incident plane wave, and U_z^{SW} is the contribution from two surface waves as illustrated in Figure 1. In (7), v_ℓ are the roots of the transcendental equation

$$D(v_\ell) = H_{v_\ell}^{(2)}(k_ob) + G_{v_\ell} H_{v_\ell}^{(2)}(k_ob) = 0 \tag{8}$$

The factor $\left[1 - e^{-jv_\ell 2\pi}\right]^{-1}$ appearing in (7) results from the fact that all the multiply encircling surface waves are included in the analysis. The factor is important and is needed to predict the resonance phenomena. Equation (3a) received the bulk of attention in this research.

INTEGRAL TECHNIQUE FOR DETERMINING COMPLEX POLES

The method is based on a technique suggested by Delves and Lyness [13]. We seek the complex roots for k, the wave number, using the following technique. Let C denote a closed contour in the complex z plane containing n zeros z_j $(j = 1,2,...,n)$ of the analytic function $f(z)$. Then from Cauchy's integral theory

$$\frac{1}{2\pi i} \int_C z^p \frac{f'(z)}{f(z)} dz = \sum_{j=1}^{n} z_j^p \equiv s_p , \tag{9}$$

where p is any non-negative integer. In particular, if $p = 0$, then the integral (9) furnishes an estimate of the number, n, of zeros within C. The integral is also evaluated with $p = 1,2,...,n$ to give sums s_p. If the number of zeros is small (say $n < 4$), these are used with Newton's formula to produce a polynomial of degree n (with leading coefficient one) having the same roots within C as $f(z)$. Specifically, the polynomial is written as

$$p_n(z) = \sum_{k=0}^{n} a_k z^k \tag{10}$$

with $a_n = 1$. The coefficients a_k for $k = n - 1, n - 2,...,1,0$ are evaluated using the recurrence relations

$$(n-k)a_k + s_1 a_{k+1} + s_2 a_{k+2} + \dots + s_{n-k} a_n = 0 . \tag{11}$$

The low order polynomial equation $p_n(z) = 0$ is solved by a standard technique (in our case, Müeller's method) to give numerical estimates $\{z_i'\}$ of the actual set of zeros $\{z_i\}$. The differences $\{z_i - z_i'\}$ are due predominantly to quadrature errors. The estimates $\{z_i'\}$ are then refined by a further application of the Newton Raphson method, this time to the original function $f(z)$.

RESULTS

Figures 2-4 illustrate the change in the complex roots as the thickness of the dielectric coating is varied. Figures 5-7 illustrate the movement of the complex poles as the relative dielectric constant of the coating is varied, keeping the thickness fixed. In all figures, equation (1) is solved keeping the first four terms of the series.

In each case, the arrows show the progression of the complex wavenumber "k" poles from thinest coating (or smallest dielectric constant) to thickest coating (or largest dielectric constant). The abscissa represents the real part of the wavenumber k, the ordinate represents the imaginary part. As expected, the roots are symmetric about the imaginary axis. The real part shows the resonance frequency while the imaginary part yields the damping rate. The n = 0,1,2,3 roots are designated respectively by squares, triangles, diamonds, and x's. Except for a few exceptions of thickness changes, the general trend is that the resonance wave becomes less damped as the thickness increases and as the dielectric constant increases. For the same changes the resonance frequency decreases. This can be understood from the fact that in both cases, surface waves are more strongly trapped by the presence of the coating, and as a consequence, the surface waves creep along the cylinder surface with less attenuation. There does not appear to be an easy way to separate out the effects of shape and composition from the graphs.

Additional calculations were performed after modifications to the Bessel function routines regarding the expressions used for comuting derivatives. Figures 8-10 give a closer examination of the effect of dielectric and coating thickness changes on the poles keeping the n = 0 term only. Here the relative dielectric constant is varied from 4-100 for three thicknesses. The effect of

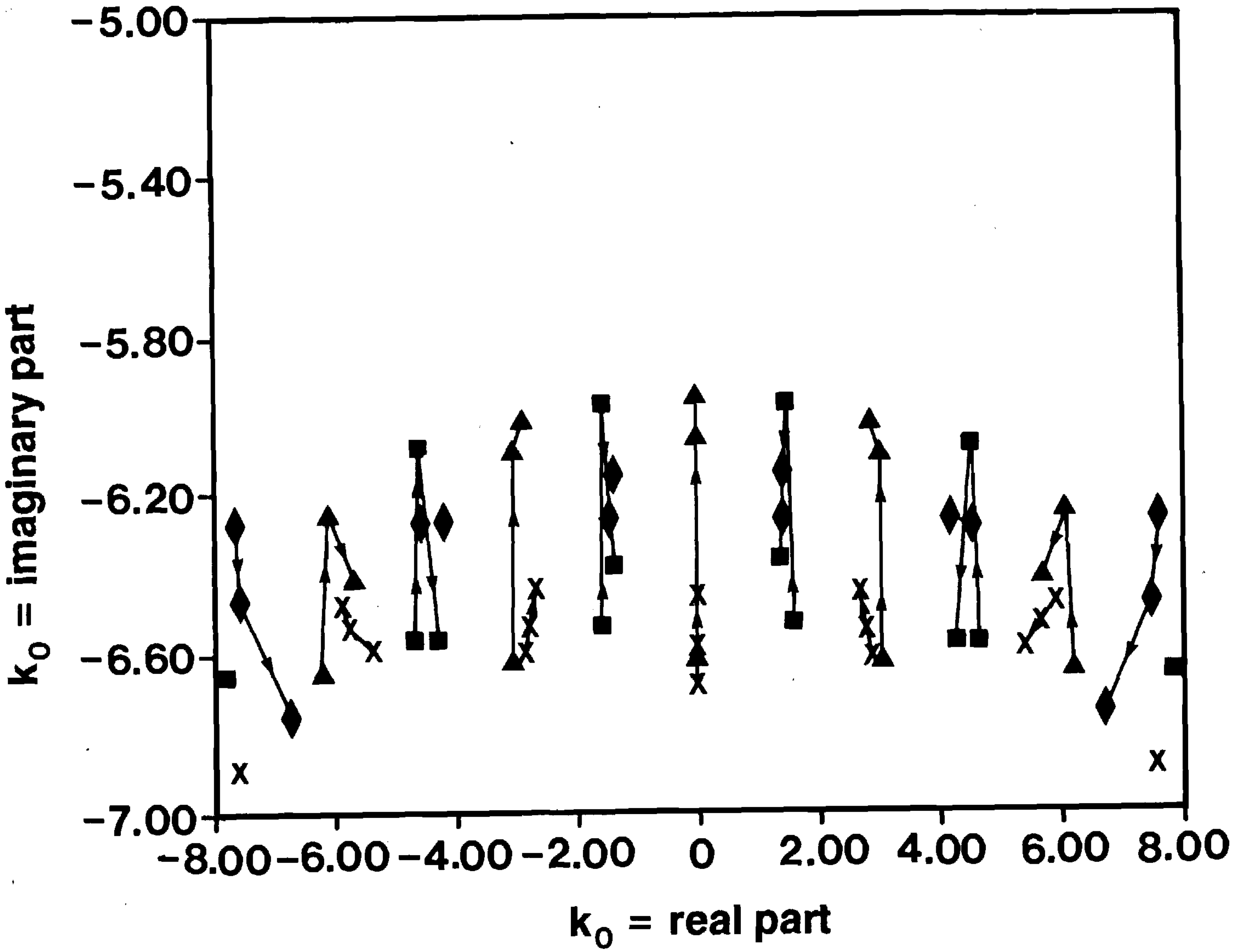

Figure 2. Complex wavenumber (k) poles with ϵ_r = 1.15 for thickness ratios b/a = 1.1, 1.15, 1.2. Plotted are modes in n = 0, 1, 2, 3 as squares, triangles, diamonds, and x's respectively. Arrows designate changes.

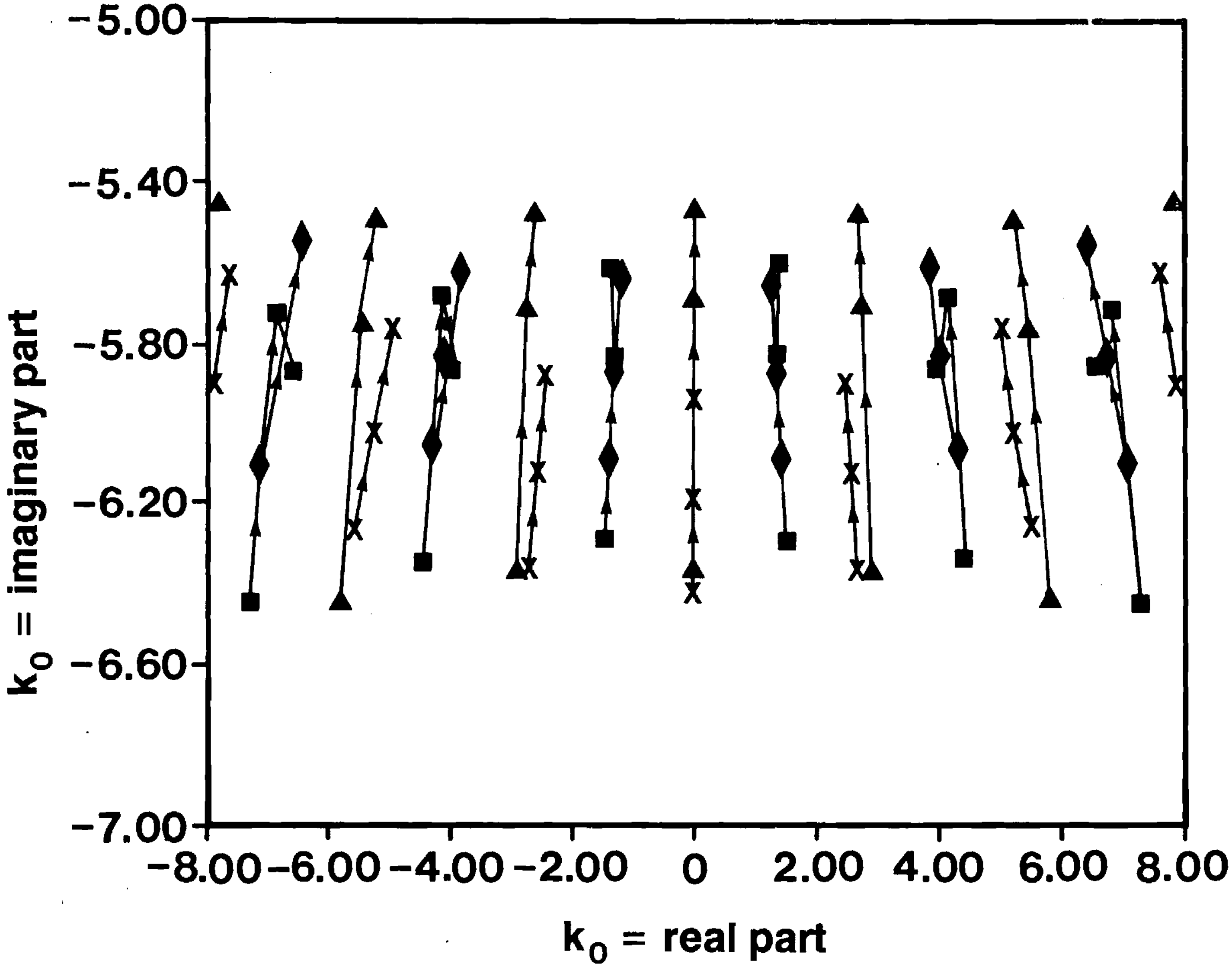

Figure 3. Complex wavenumber (k) poles with $\epsilon_r = 2.56$ for thickness ratios $b/a = 1.1, 1.15, 1.2$. Plotted are modes in $n = 0, 1, 2, 3$, as squares, triangles, diamonds, and x's respectively. Arrows designate changes.

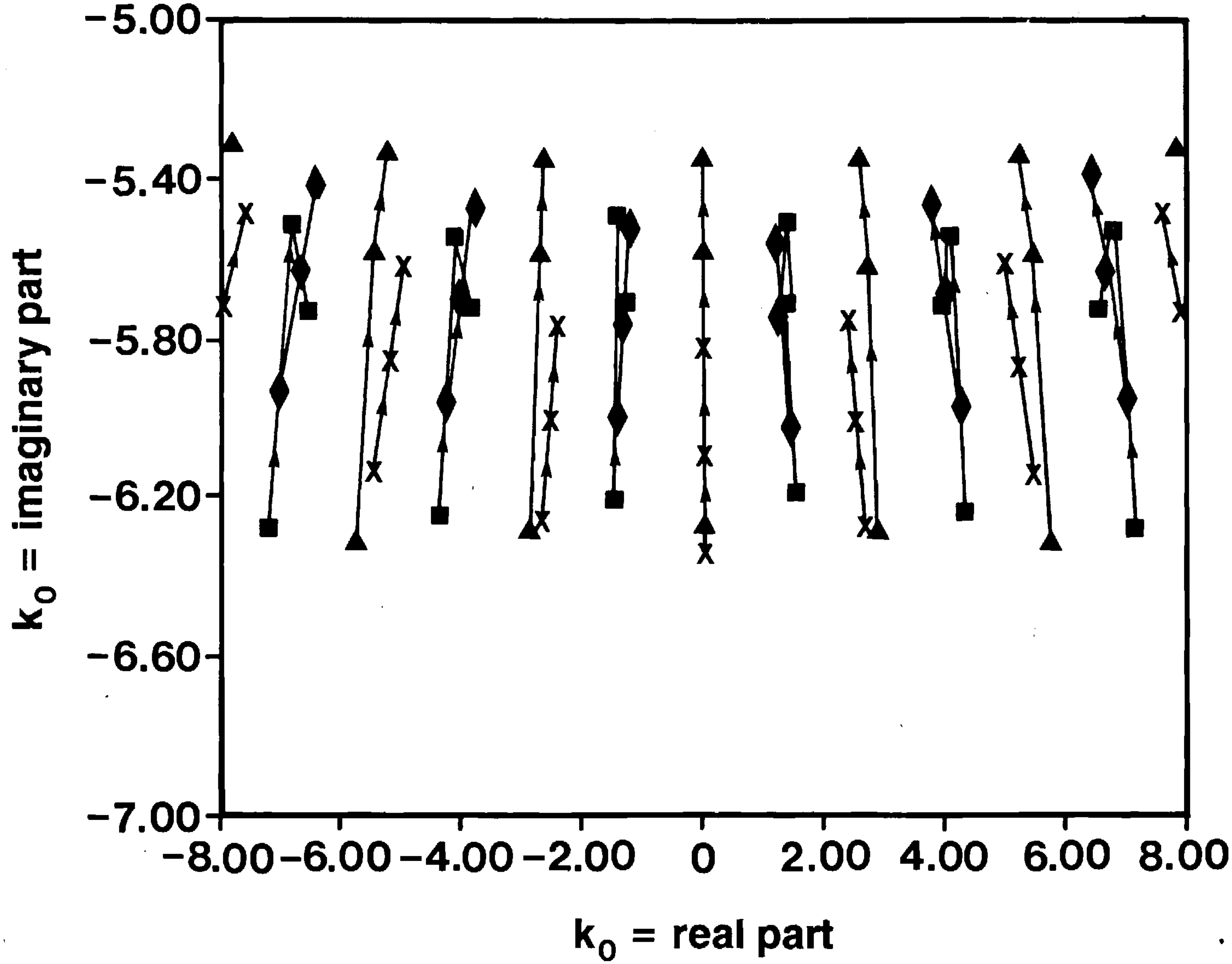

Figure 4. Complex wavenumber (k) poles with $\epsilon_r = 4.0$ for thickness ratios b/a = 1.1, 1.15, 1.2. Plotted are modes in n = 0, 1, 2, 3 as squares, triangles, diamonds, and x's respectively. Arrows designate changes.

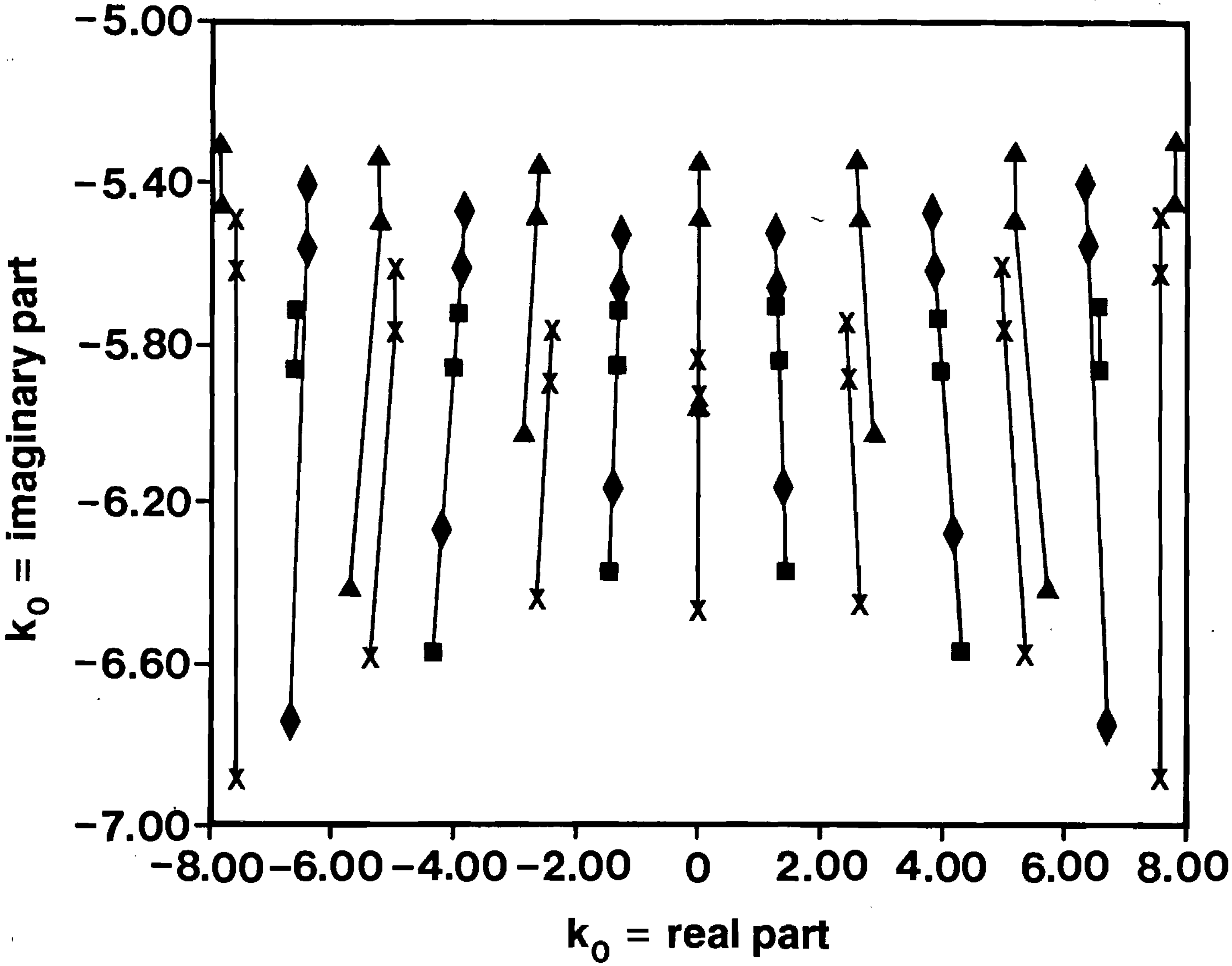

Figure 5. Complex wavenumber (k) poles for thickness ratios b/a = 1.2, with dielectric constants ϵ_r = 1.15, 2.56, 4.0.

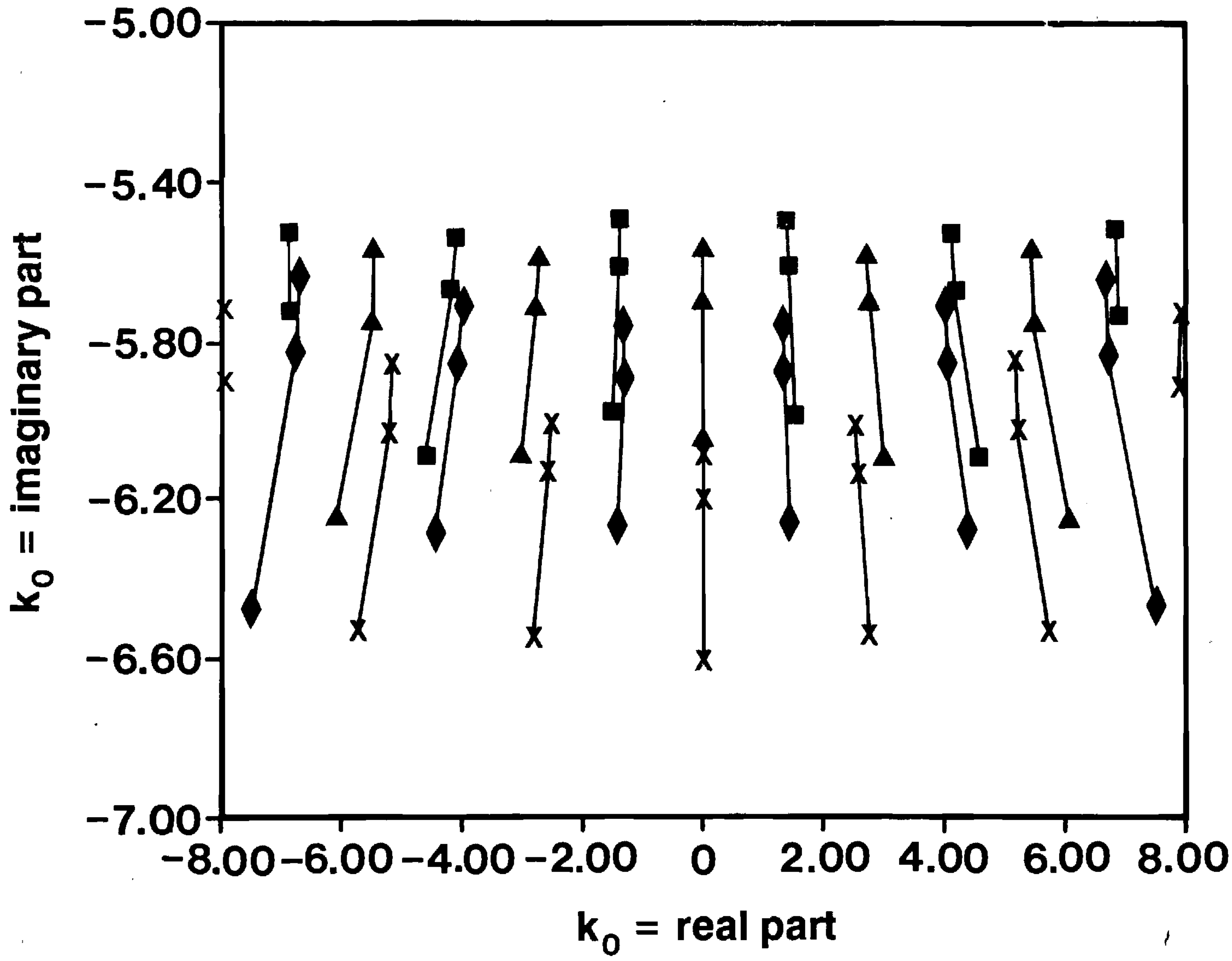

Figure 6. Complex wavenumber (k) poles for thickness ratios b/a = 1.15, with dielectric constants ϵ_r = 1.15, 2.56, 4.0.

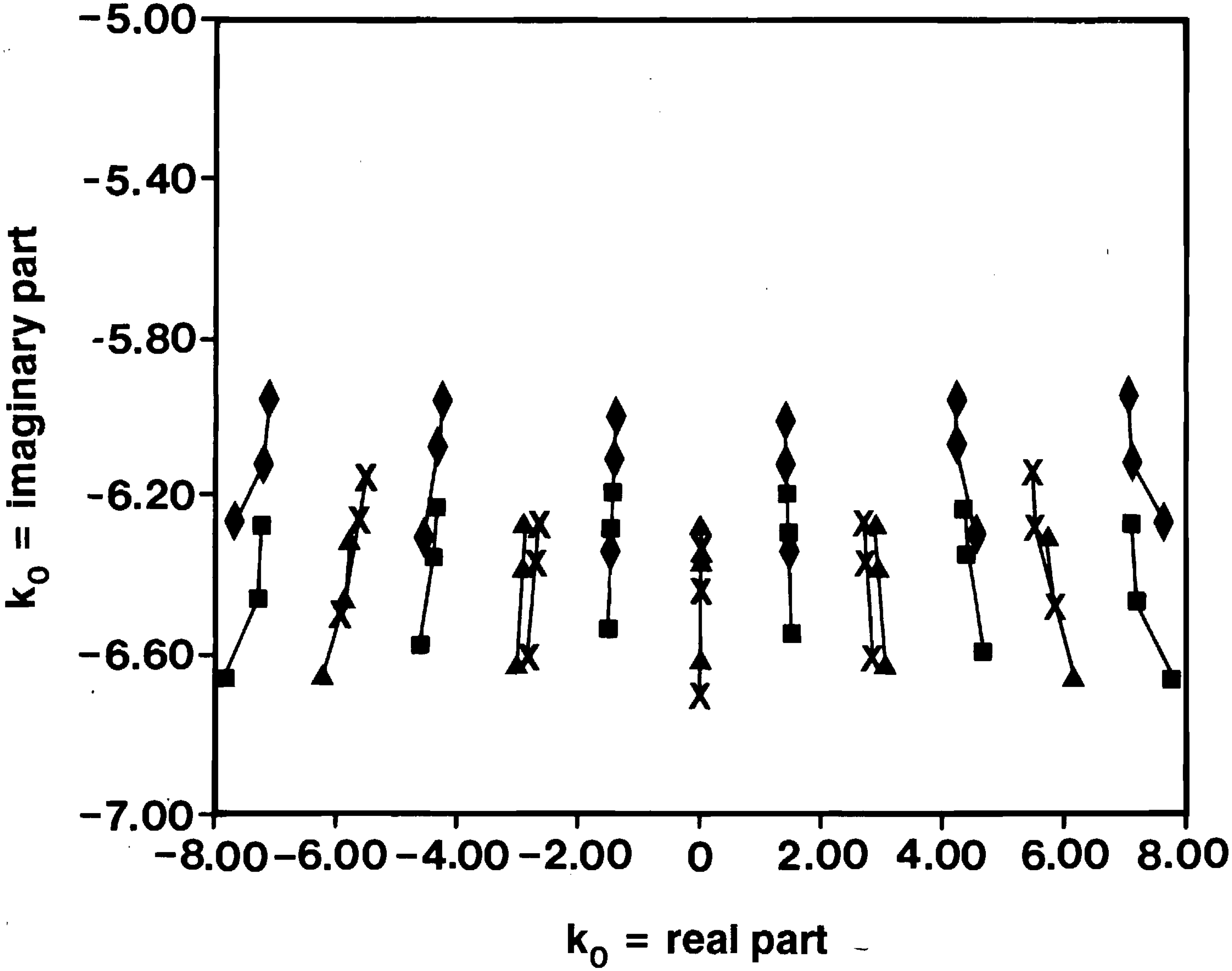

Figure 7. Complex wavenumber (k) poles for thickness ratios b/a = 1.1, with dielectric constants ϵ_r = 1.15, 2.56, 4.0.

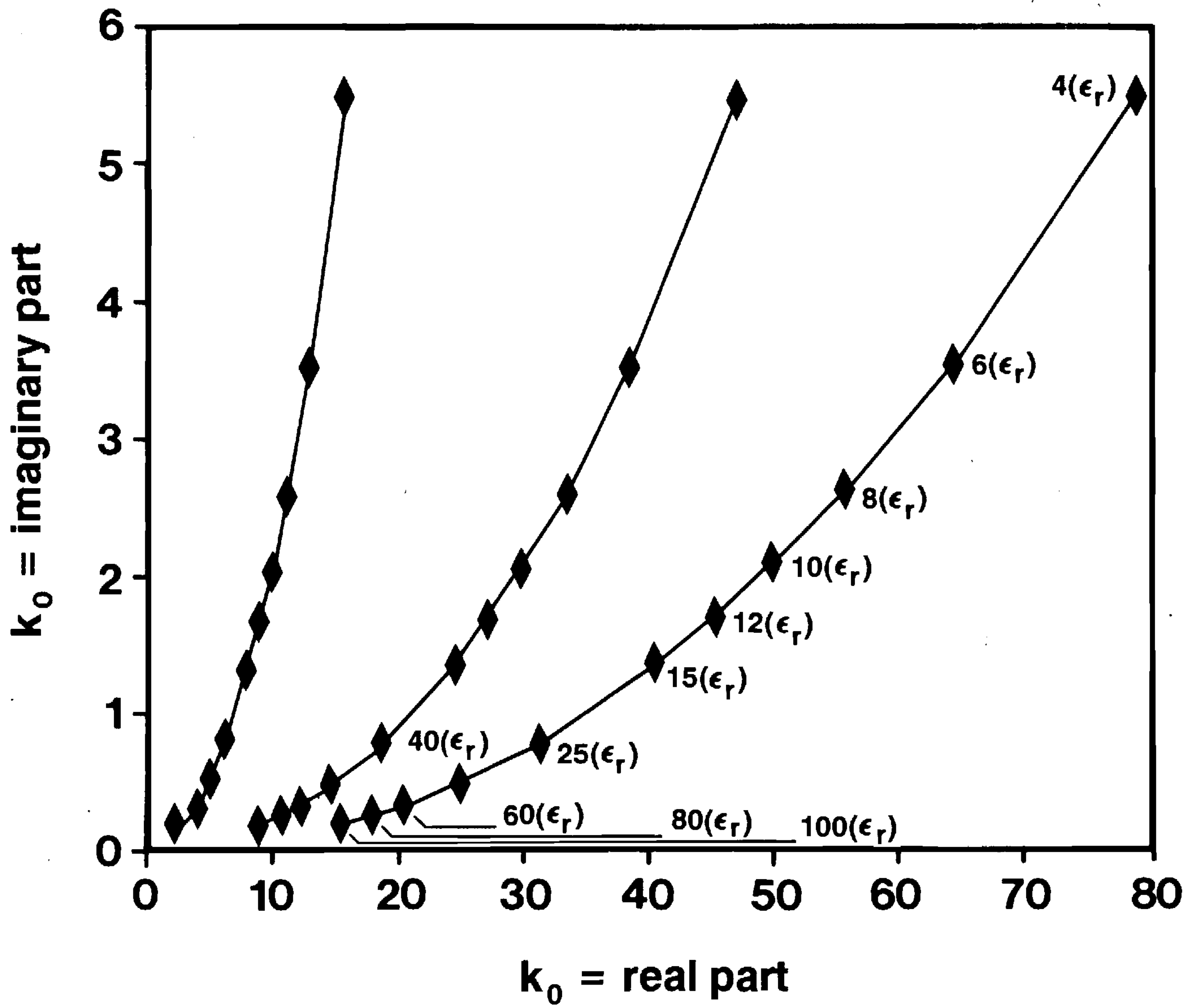

Figure 8. Complex wavenumber ($k_0 = \omega\sqrt{\mu_0 \epsilon_0}$)poles for coating thickness b/a = 1.05, $\epsilon_r = 4 \ldots 100$.
Mode n = 0 is plotted only - first 3 poles.

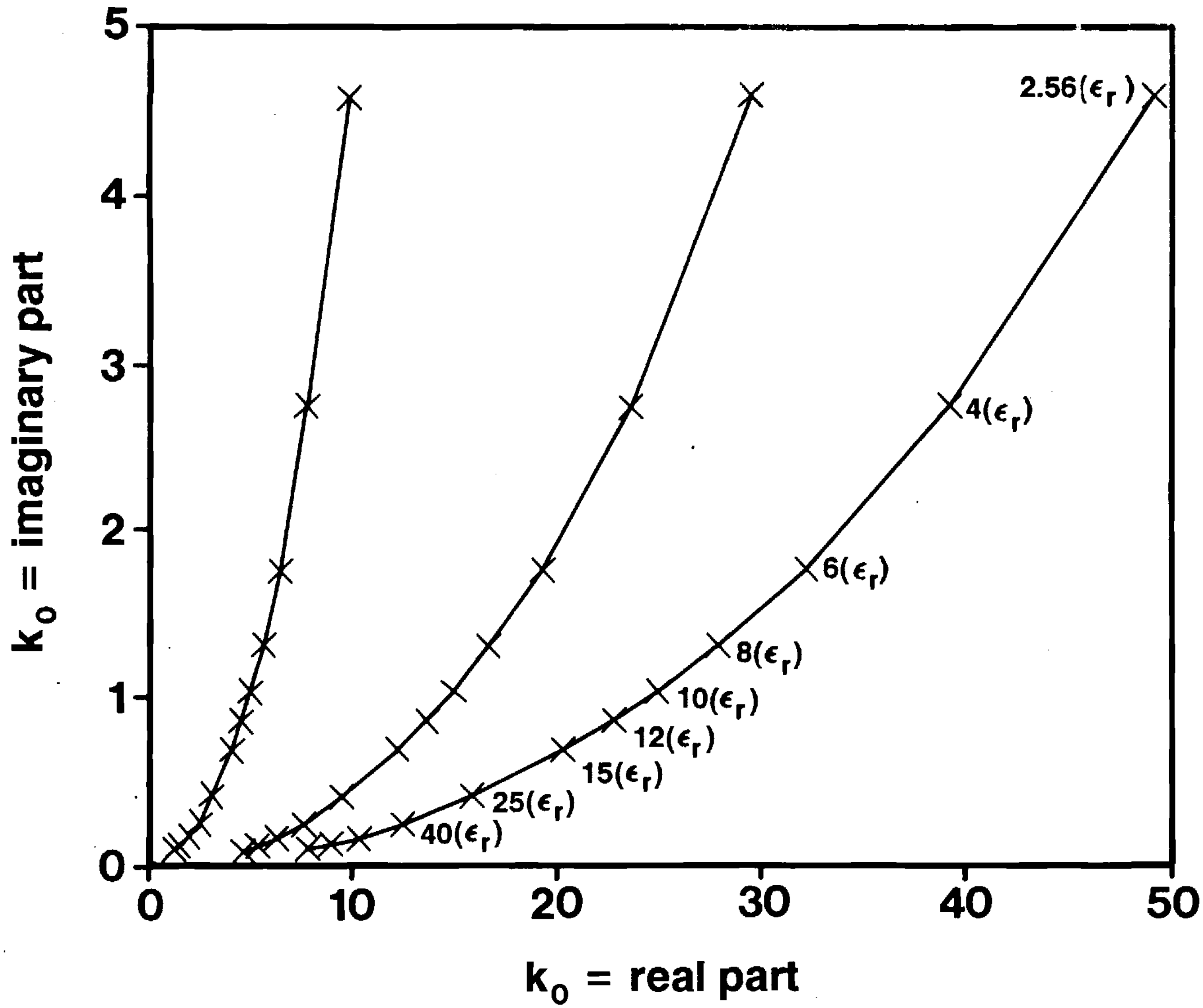

Figure 9. Complex wavenumber ($k_o = \omega\sqrt{\mu_o \epsilon_o}$)poles for coating thickness b/a = 1.10, ϵ_r = 2.56 . . . 100.
Mode n = 0 is plotted only - first 3 poles.

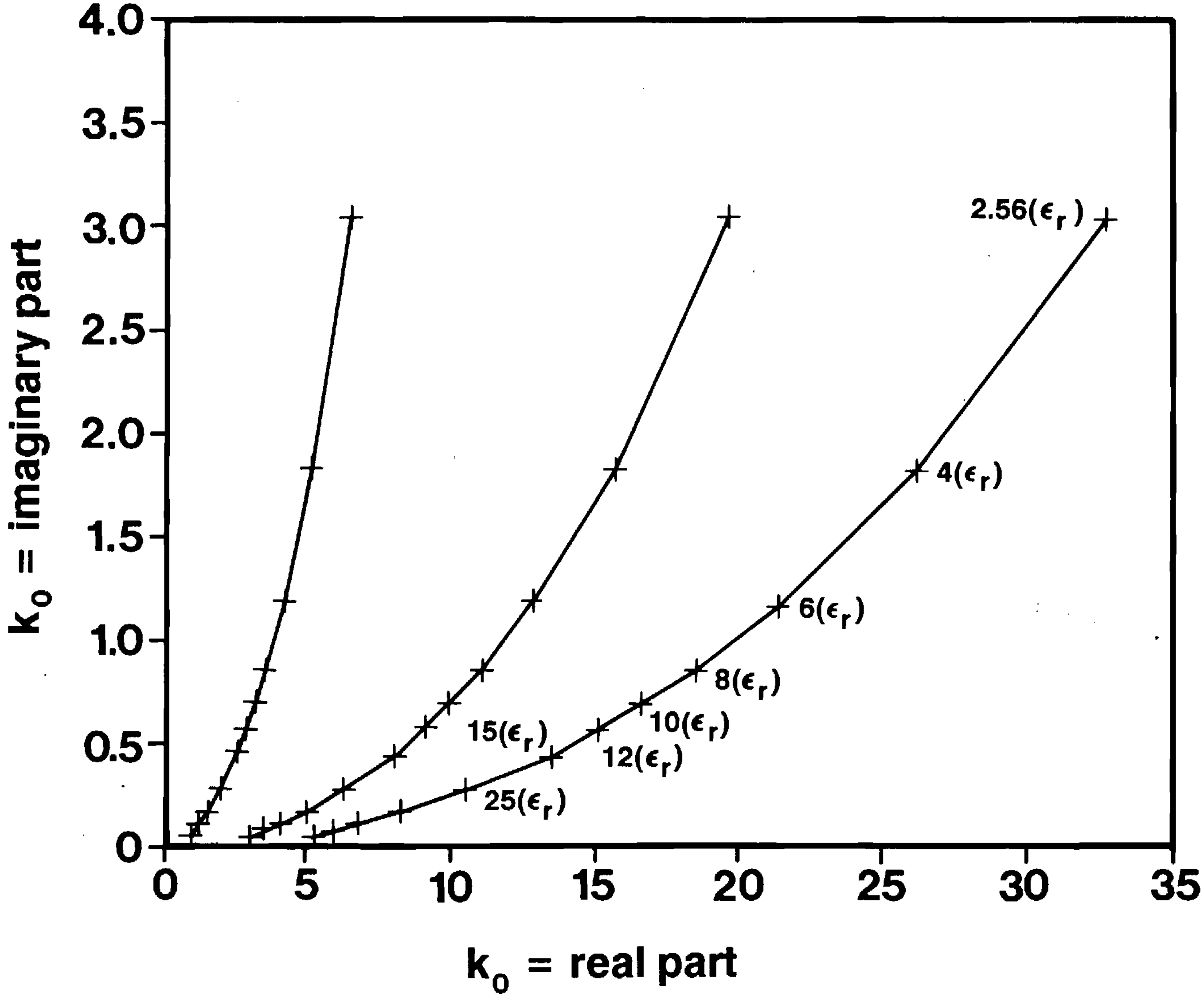

Figure 10. Complex wavenumber ($k_0 = \omega\sqrt{\mu_0\epsilon_0}$)poles for coating thickness b/a = 1.15, ϵ_r = 2.56 . . . 100.
Mode n = 0 is plotted only - first 3 poles.

varying thickness for three dielectric constants is shown in Figures 11-13. In both sets, on the second quadrant redundant roots are omitted. Except for very large dielectric constants, it appears that the effects of thickness and dielectric composition cannot be easily separated.

TARGET DISCRIMINATION

Given the apparent linkage of coating thickness and composition effects, there are two approaches that might be followed in pursuing the inverse problem. The first is to formulate a relationship based on the above curves between the complex roots and the thickness and dielectric constant.

$$
\begin{aligned}
Z_1 &= f_1(\varepsilon_r, th) \\
Z_2 &= f_2(\varepsilon_r, th) \\
Z_3 &= f_3(\varepsilon_r, th)
\end{aligned}
\tag{12}
$$

where the subscripts on the roots designate the order of the roots. The inverse solver would then have the task of minimizing the functional

$$
\begin{aligned}
S = &\left(Z_1\text{measured} - f_1(\varepsilon_r, th)\right)\left(Z_1\text{measured} - f_1(\varepsilon_r, th)\right)^* \\
&+ \left(Z_2\text{measured} - f_2(\varepsilon_r, th)\right)\left(Z_2\text{measured} - f_2(\varepsilon_r, th)\right)^* \\
&+ \left(Z_3\text{measured} - f_3(\varepsilon_r, th)\right)\left(Z_3\text{measured} - f_3(\varepsilon_r, th)\right)^* \\
&+ \ldots.
\end{aligned}
\tag{13}
$$

using a steepest descent or nonlinear optimization technique. Another alternative is to simply compute a priori a table of roots for an array of

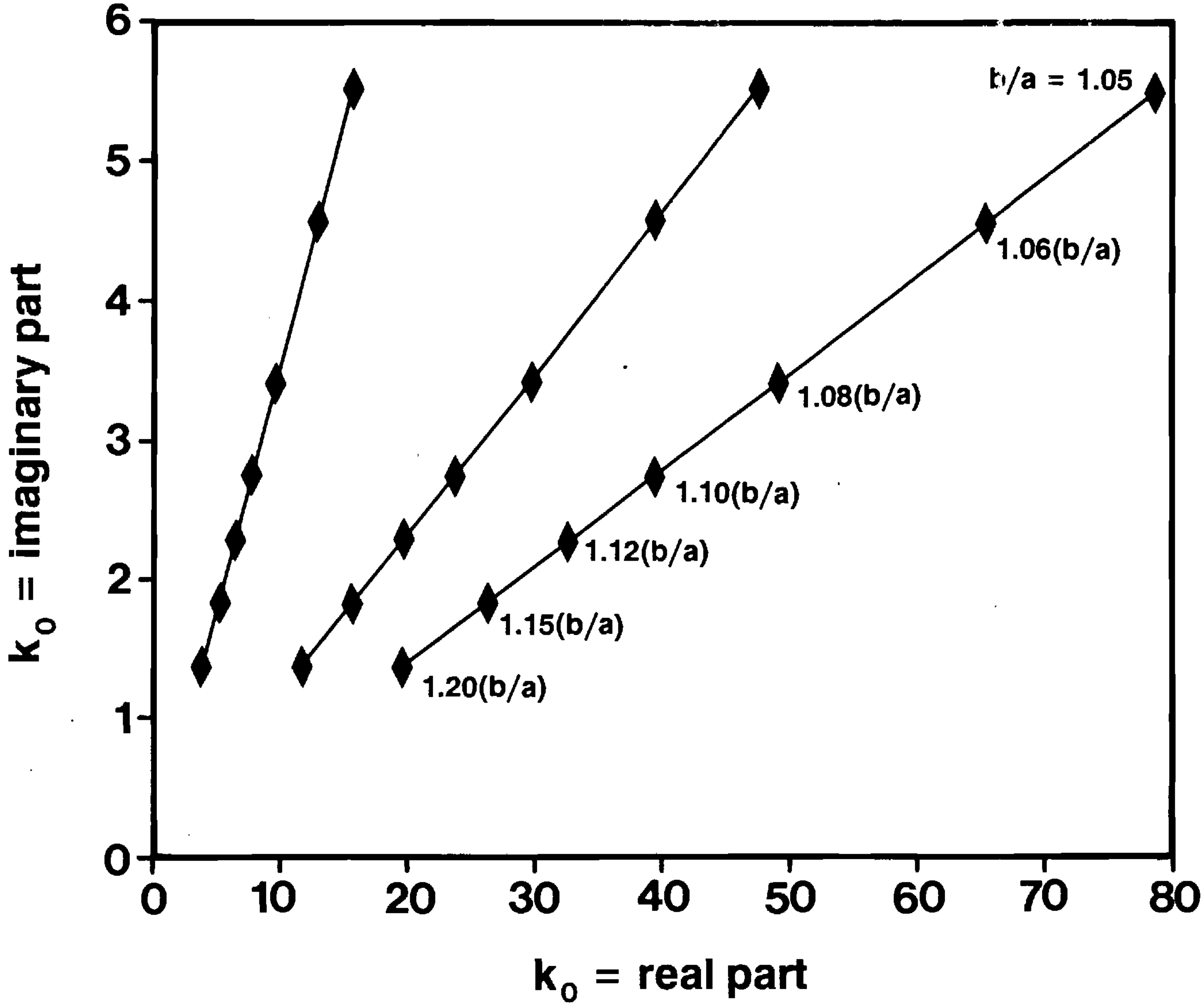

Figure 11. Complex wavenumber($k_0 = \omega\sqrt{\mu_0\epsilon_0}$)poles for coating thickness $\epsilon_r = 4$, b/a = 1.05 . . . 1.2.
Mode n = 0 is plotted only - first 3 poles.

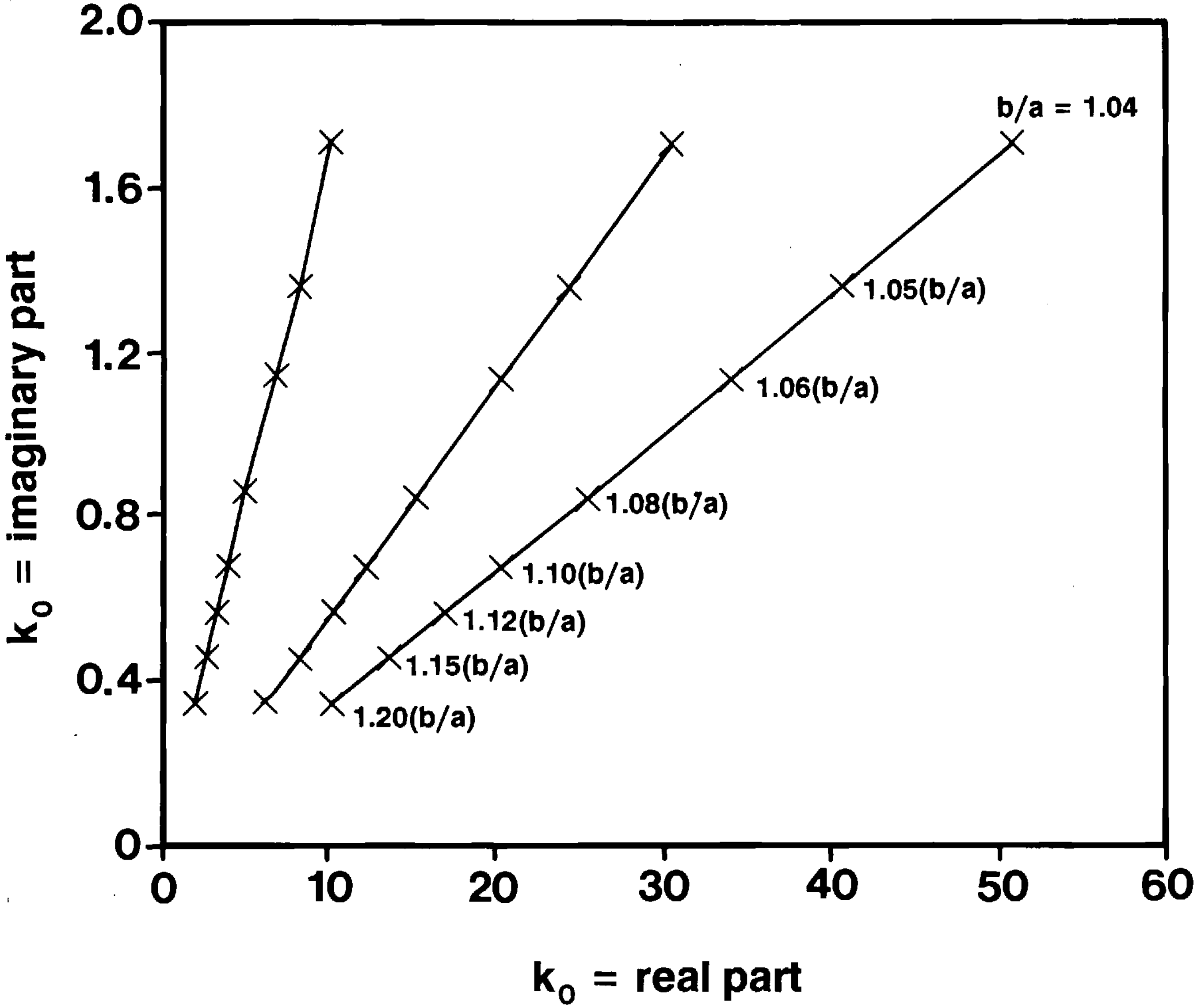

Figure 12. Complex wavenumber ($k_0 = \omega\sqrt{\mu_0\epsilon_0}$)poles for coating thickness $\epsilon_r = 15$, b/a = 1.04 . . .1.2.
Mode n = 0 is plotted only - first 3 poles.

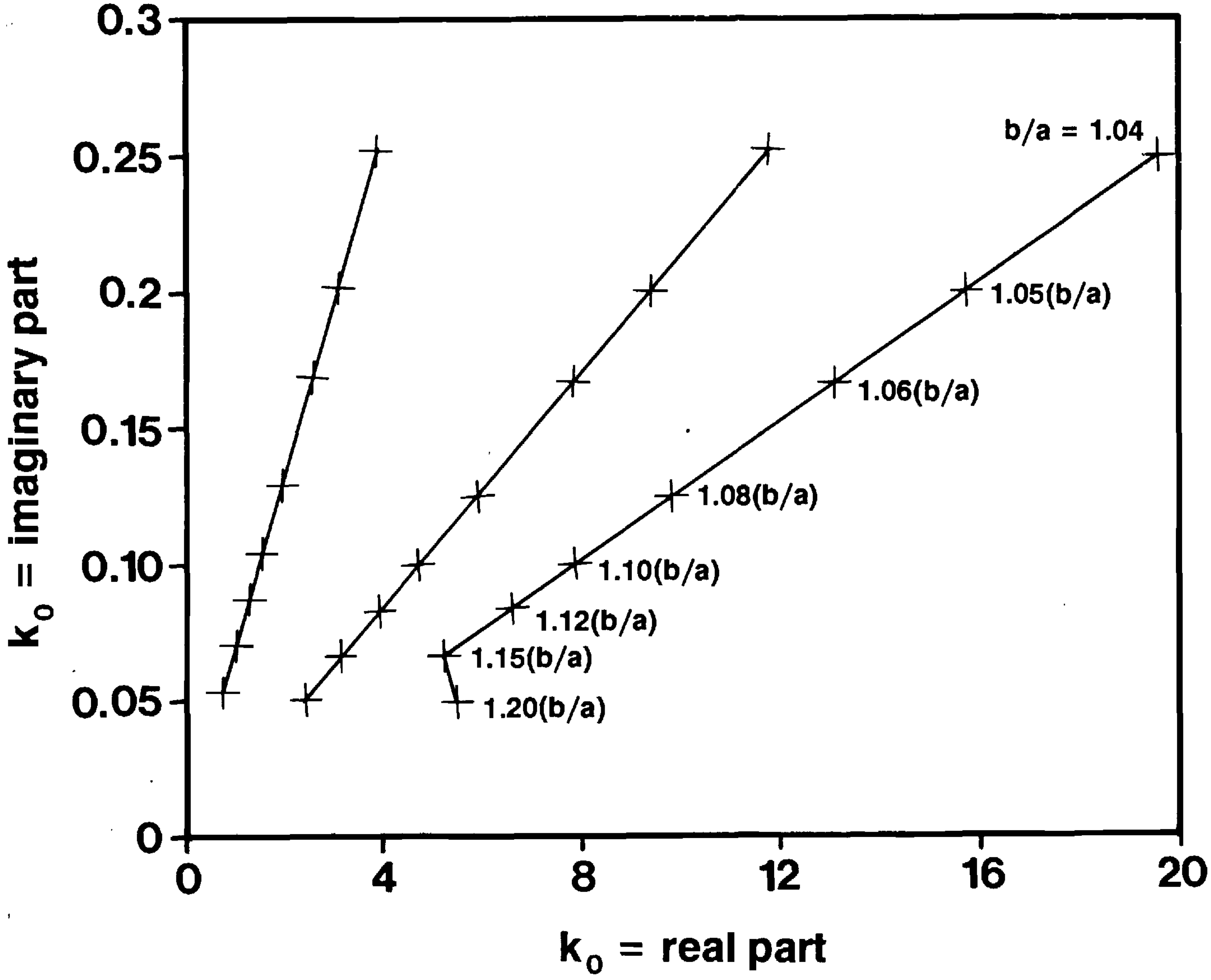

Figure 13. Complex wavenumber ($k_0 = \omega\sqrt{\mu_0\epsilon_0}$)poles for coating thickness ϵ_r = 100, b/a = 1.04 . . . 1.2. Mode n = 0 is plotted only - first 3 poles.

dielectric and thickness values. The task of the inversion algorithm degenerates to finding a least squares optimal fit for the measured pole pairs with those pole pairs in the table. Such a table for the dielectric coated cylinder along with the algorithm for determining the most likely dielectric/thickness pair are listed in the Appendices.

FUTURE WORK/CONCLUSIONS

The separation of composition and shape effects on the resonance scattering poles is not a simple task in the electromagnetic world. Because their effects cannot be isolated, sophisticated optimization schemes must be employed to realize target discrimination capability. This has been accomplished for the dielectric coated cylinder via the series expansion field expression appropriate for this problem [1]. However, the series of equations in (1) converge quite poorly when the product $k_o b$ is not small. This is of course the motivating force behind the Watson transformation. With the scattered field expressed according to equation (7), this solution convergers very rapidly. It is thought that a better prediction of the complex resonances can be realized through the solution of (7) and (8). This solution could be sought using the asymptotic formula for the Hankel function in (14),

$$H_{\nu_\ell}^{(2)}(k\rho) \simeq \frac{2}{\sqrt{\pi\left(k^2\rho^2-\nu_\ell^2\right)}} \exp\left[i\left(\sqrt{k^2\rho^2 - \nu_\ell^2} - \nu_\ell \cos^{-1}\frac{\nu_\ell}{k\rho} - \frac{\pi}{4}\right)\right] \tag{14}$$

and might well shed additional light on the target discrimination problem.

REFERENCES

1. Nan Wang, "Electromagnetic Scattering from a Dielectric Coded Cylinder," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-33, no. 9, p. 960, September 1987.

2. R. Kastner and R. Mittra, "A Spectral-Iteration Technique Analyzing Scattering from Arbitrary Bodies, Part 1: Cylindrical Scatterers with E-Wave Incidents," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-31, no. 3, p. 499, May 1983.

3. H. Uberall and G. Gaunaurd, "The Physical Content of the Singularity Expansion Method," <u>Applied Physics Letters</u>, vol. 39, no. 4, p. 362, August 1981.

4. G. Gaunaurd and A. Kalnins, "Resonances in the Sonar Cross Sections of Coded Spherical Shelves," <u>International Journal of Solid Structures</u>, vol. 18, no. 12, pp. 1083-1102, 1982.

5. G. Gaunaurd, "Sonar Cross Section of a Coded Hollow Cylinder in Water," <u>Journal of Acoustic Society of America</u>, vol. 61, no. 2, p. 360, February 1977.

6. C. Chuang, ‡. Liang, and S. Lee, "High Frequency Scattering from an Open Ended Semi-Infinite Cylinder," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-23, no. 6, p.770, November 1975.

7. C. Eftimiu and P. Huddleston, "Cylindrical Eigencurrents," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-31, no. 2, p. 325, March 1983.

8. J. Roumeliotis, J. Fikioris, and G. Gounaris, "Electromagnetic Scattering from an Eccentrically Coded Infinite Metallic Cylinder," <u>Journal of Applied Physics</u>, vol. 51, no. 8, p. 4488, August 1980.

9. A. Bhattacharyya and S. Tandon, "Radar Cross Section of a Finite Planar Structure Coded with a Lossy Dielectric," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-32, no. 9, p. 1003, September 1984.

10. Reuben Eaves, "Electromagnetic Scattering from a Conducting Circular Cylinder Covered with a Circumferentially Magnetized Ferrite," <u>IEEE Transactions on Antennas and Propagation</u>, vol. AP-24, no. 2, p. 190, March 1976.

11. G. Gaunaurd, H. Uberall, and P. Moser, "Resonances of Dielectrically Coded Conducting Spheres and the Inverse Scattering Problem," <u>Journal of Applied Physics</u>, vol. 52, no. 1, p. 35, January 1981.

12. G. Gaunaurd and H. Uberall, "Electromagnetic Spectral Determination of the Material Composition of Penetrable Radar Targets," <u>Nature</u>, vol. 287, no. 5784, pp. 708-709, October 1980.

13. Delves and Lyness, Math Computations 21, 1967, p. 543.

APPENDICES

These appendices show programs and additional results in the project. Appendix A is the Fortran program used to perform the root integral of the Hankel and Bessel functions in (1) using (9) and then a Muller solver on (11). Appendix B is the algorithm to refine these roots using a Newton Raphson procedure

$$Z^k = Z^{k-1} - \frac{f(Z^k)}{f'(Z^k)} .$$

Appendix C is the algorithm to locate the optimum least squares thickness/ dielectric combination given an arbitrary set of pole locations. Appendix D shows the table of roots associated with various combinations of thickness and dielectric constant, used in Appendix C.

APPENDIX A

ALGORITHM TO PERFORM THE ROOT INTEGRAL (9)

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      COMPLEX PART,SUM,S(30),AR(30),ROOTS(30),K0,J,JDE,JDOUDE
     +        ,Y,YDE,YDOUDE,H2,H2DE,H2DOUDE,G,GDE,DENO,DENODE
      COMPLEX   DENO0,DENO1,DENO2,NEWK0,DELK0
      PI=ACOS(-1.)
C
C     SET PARAMETER FOR THE ORDER OF THE INTEGRATION

    8 NPTS=96
      WRITE(*,*) 'INPUT N: '
      READ(*,*) N
      WRITE(*,*) 'INPUT ER:'
      READ(*,*) ER
      A=1
      WRITE(*,*) 'INPUT A/B RATIO '
      READ(*,*) B

   10 WRITE(6,*)'INPUT LOWER AND UPPER RANGE ALONG REAL AXIS'
      READ(5,*) X1,X2
      WRITE(6,*)'INPUT LOWER AND UPPER RANGE ALONG IMAG. AXIS'
      READ(5,*) Y1,Y2
C
C**************************************************************
C     PERFORM THE LINE INTEGRAL WITH THE POWER OF K0 EQUAL ZERO
C     TO FIND AN APPROXIMATION TO THE NUMBER OF ZERO'S ENCLOSED.
      SUM=(0.,0.)
C
C     EVALUATE FIRST LINE INTEGRAL WITH X (REAL VALUE) FIXED.
      CALL CINTE(PART,Y1,Y2,X2,NPTS,1,0,N,K0,A,B,ER)
      SUM=SUM+PART

C     EVALUATE SECOND LINE INTEGRAL WITH Y (IMAGINARY) FIXED.
      CALL CINTE(PART,X1,X2,Y2,NPTS,2,0,N,K0,A,B,ER)
      SUM=SUM+PART

C     EVALUATE THIRD LINE INTEGRAL WITH X (REAL) FIXED AND
C     EQUAL TO LOWER BOUND.
      CALL CINTE(PART,Y1,Y2,X1,NPTS,3,0,N,K0,A,B,ER)
      SUM=SUM+PART
C
C     EVALUATE FOURTH LINE INTEGRAL WITH Y (IMAG.) FIXED AND
C     EQUAL TO THE LOWER BOUND.
      CALL CINTE(PART,X1,X2,Y1,NPTS,4,0,N,K0,A,B,ER)
      SUM=SUM+PART
C
      SUM=SUM/CMPLX(0.0,2*PI)
      WRITE(6,*)'AN APPROXIMATION TO THE # OF ROOTS=',SUM
      WRITE(6,*)'ENTER GUESS FOR NUMBER OF ROOTS OR'
      WRITE(6,*)'ENTER "0" TO CHANGE REGION OF INTEGRATION'
      READ(5,*)NZER

C     NZER=INT( 0.3 +REAL( SUM ))
C     IF (AIMAG(SUM).GE.0.3)  NZER=0
      IF(NZER.EQ.0) GOTO 10

C**************************************************************
```

```
      DO 20 I=1,NZER
C     PERFORM THE SAME PROCEDURE AS BEFORE TO EVALUATE THE
C     LINE INTEGRAL EXCEPT INCREMENT THE POWER OF THE
C     EXPONENT ON KO IN THE INTEGRAL.
C
      CALL CINTE(PART,Y1,Y2,X2,NPTS,1,I,N,KO,A,B,ER)
      S(I)=PART
C
      CALL CINTE(PART,X1,X2,Y2,NPTS,2,I,N,KO,A,B,ER)
      S(I)=S(I)+PART
C
      CALL CINTE(PART,Y1,Y2,X1,NPTS,3,I,N,KO,A,B,ER)
      S(I)=S(I)+PART
C
      CALL CINTE(PART,X1,X2,Y1,NPTS,4,I,N,KO,A,B,ER)
      S(I)=S(I)+PART
C
      S(I)=S(I)/CMPLX(0.0,2*PI)
      WRITE(6,777)I,S(I)
  777 FORMAT(3X,I2,8X,2(F12.5))
   20 CONTINUE
C
C*******************************************************
C     EVALUATE COEFFICIENTS OF THE NEW POLYNOMIAL TO BE USED
C     IN THE APPROXIMATION.
      AR(NZER+1)=(1.,0.0)
      K=NZER-1
      KNT=K+1
   40 IF(K.LT.0) GOTO 60
      PART=(0.0,0.0)
      DO 50 JS=1,NZER-K
      PART=PART+S(JS)*AR(KNT+JS)

   50 CONTINUE
      AR(KNT)=(-1.,0)*PART/(NZER-K)
      K=K-1
      KNT=K+1
      GOTO 40
   60 CONTINUE
C
      WRITE(6,78)
   78 FORMAT(8X,'COEFFICIENTS OF THE POLYNOMIAL APPROXIMATION')
      WRITE(6,81)
   81 FORMAT(3X,'N',22X,'AR(N)')
      DO 80 I=1,NZER+1
      WRITE(6,79)I-1,AR(I)
   79 FORMAT(3X,I2,8X,2(F12.5))
   80 CONTINUE
C
C     CALL THE SUBROUTINE TO EVALUATE THE COEFFICIENTS OF
C     THE NEW POLYNOMIAL.

      CALL ZROOTS(AR,NZER,ROOTS,.TRUE.)
      WRITE(6,*)'SUBROUTINE ZROOTS COMPLETED'
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,129)
  129 FORMAT(3X,'EVALUATION OF FUNCTION AT APPROXIMATE ROOTS')
```

```
      WRITE(6,*)
      WRITE(6,128)
  128 FORMAT(2X,'#',15X,'ROOT',27X,'DENO(ROOT)')
      DO 130 I=1,NZER
      WRITE(6,131)I,ROOTS(I),DENO(N,ROOTS(I),A,B,ER)
  131 FORMAT(2X,I2,3X,2(F12.5),5X,2(F12.5))
  130 CONTINUE

      WRITE(*,*)'ENTER "1" TO MODIFY  D_FUNCTION   : '
      READ(*,*)MODIFY
      IF(MODIFY.NE.1) GOTO 150

      DO 140 I=1,NZER
      MODOC=0
  135 IF(CABS(DENO(N,ROOTS(I),A,B,ER)).GE.0.0000001) THEN
      MODOC=MODOC+1
      WRITE(*,*)'NUMBERS OF MODIFYING :',MODOC
      DENO0 = DENO(N,ROOTS(I),A,B,ER)
      NEWK0 = ROOTS(I) + (0.0005,0.0005)
      DENO1 = DENO(N,NEWK0,A,B,ER)
      NEWK0 = ROOTS(I) - (0.0005,0.0005)
      DENO2 = DENO(N,NEWK0,A,B,ER)
      DELK0 = DENO0 / ( (DENO1-DENO2)/(0.001,0.001) )
      ROOTS(I)     = ROOTS(I) - DELK0
      WRITE(*,*)'K0          :',ROOTS(I)
      WRITE(*,*)'D_FUNCTION :',DENO(N,ROOTS(I),A,B,ER)
      GOTO 135
      END IF
  140 CONTINUE
      DO 142 I=1,NZER
      WRITE(6,131)I,ROOTS(I),DENO(N,ROOTS(I),A,B,ER)
  142 CONTINUE


 150  WRITE(6,*)'TRY AGAIN?  (YES=1)'
      READ(5,*)ITEST
      IF(ITEST.EQ.1)GOTO 08
      STOP
      END
C
C*******************************************************
      SUBROUTINE CINTE(PART,FIRS,SECO,CNST,NPTS,IDENT,NP,N,K0,A,B,ER)
C     THIS SUBROUTINE EVALUATES THE LINE INTEGRALS.
C     EACH ROOT AND WEIGHTING ARE USED TWICE.  IF IT
C     IS THE SECOND TIME A ROOT IS BEING USED, IT IS
C     MADE NEGATIVE.
      DIMENSION ROOT(48),AC(48)
      COMPLEX K0,DENO,PART,TEMP,TEMP2,J,JDE,JDOUDE
     +         ,Y,YDE,YDOUDE,H2,H2DE,H2DOUDE,G,GDE,DENODE

      PART=(0.0,0.0)

      ITWO=0
      IMAX=48
      ITOP=48
      DO 30 I=1,NPTS
```

```
      ROOTJ=ROOT(ITOP)
      ACJ=AC(ITOP)
      IF(ITOP.EQ.ITWO) ROOTJ=-ROOTJ
      T=((SECO-FIRS)*ROOTJ+ SECO+FIRS)/2
C     IF IT HAS AN IDENTIFIER OF A PATH #1 OR PATH #3 THEN
C     X (REAL) IS FIXED AND IT WILL INTEGRATE ALONG Y (IMAG.).
      IF(IDENT.EQ.1.OR.IDENT.EQ.3) THEN
      KO=CMPLX(CNST,T)
      ELSE
      KO=CMPLX(T,CNST)
      ENDIF
C
      TEMP=KO**NP

      TEMP=TEMP*DENODE(N,KO,A,B,ER)
      TEMP2=DENO(N,KO,A,B,ER)

      IF(TEMP2.EQ.(0.0,0.0)) TEMP2=(1.E-20,1.E-20)
      TEMP=TEMP/TEMP2
      TEMP=TEMP*ACJ
      PART=PART+TEMP

      ITWO=ITOP
      ITOP=IMAX-INT(I/2)
   30 CONTINUE
      IF(IDENT.EQ.2.OR.IDENT.EQ.3) PART=PART*(-1.,0.)
      IF(IDENT.EQ.1.OR.IDENT.EQ.3) PART=PART*(0.,1.)
      PART=PART*(SECO-FIRS)/2
      RETURN
      DATA AC   /.0325506145,.0325161187,.0324471637,
     C   .0323438226,.0322062048,.0320344562,.0318287589,.0315893308,
     C   .0313164256,.0310103326,.0306713761,.0302999154,.0298963441,
     C   .02946109,.0289946142,.0284974111,.0279700076,.0274129627,
     C   .0268268667,.0262123407,.0255700360,.0249006332,.0242048418,
     C   .0234833991,.0227370697,.0219666444,.0211729399,.0203567972,
     C    .0195190811,.0186606796,.0177825023,.0168854799,.0159705629,
     C   .0150387210,.0140909418,.0131282296,.0121516047,.0111621021,
     C   .0101607705,.0091486712,.0081268769,.0070964708,.0060585455,
     C   .0050142027,.0039645543,.0029107318,.0018539608,.0007967921/
      DATA ROOT  /.0162767448,.0488129851,.0812974955,
     C   .1136958501,.1459737147,.1780968824,.2100313105,.2417431562,
     C   .2731988126,.3043649444,.3352085229,.3656968615,.3957976498,
     C   .4254789884,.4547094222,.4834579739,.5116941772,.5393881083,
     C   .5665104186,.5930323648,.6189258401,.6441634038,.66871831,
     C   .6925645366,.7156768123,.7380306437,.7596023412,.7803690439,
     C   .8003087441,.8194003107,.8376235112,.8549590334,.8713885059,
     C   .8868945174,.9014606353,.9150714231,.9277124567,.9393703398,
     C   .9500327178,.9596882914,.9683268285,.9759391746,.9825172636,
     C   .9880541263,.9925439003,.995981843,.9983643759,.9996895039/
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      COMPLEX FUNCTION J(N,Z)
      COMPLEX Z
      CALL BSSLJ(Z,N,J)
      RETURN
      END
```

```
      COMPLEX FUNCTION JDE(N,Z)
      COMPLEX Z,J
      N1=N-1
      JDE= J(N1,Z) - N * J(N,Z) / Z
      RETURN
      END

      COMPLEX FUNCTION JDOUDE(N,Z)
      COMPLEX Z,J,JDE
      JDOUDE= -J(N,Z) - JDE(N,Z)/Z + J(N,Z)/(Z*Z)
C               IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END

      COMPLEX FUNCTION Y(N,Z)
      COMPLEX Z
      CALL BSSLY(Z,N,Y)
      RETURN
      END

      COMPLEX FUNCTION YDE(N,Z)
      COMPLEX Z,Y
      N1=N-1
      YDE = Y(N1,Z) - N*Y(N,Z)/Z
      RETURN
      END

      COMPLEX FUNCTION YDOUDE(N,Z)
      COMPLEX Z,Y,YDE
      YDOUDE= -Y(N,Z) - YDE(N,Z)/Z + Y(N,Z)/(Z*Z)
C               IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END

      COMPLEX FUNCTION H2(N,Z)
      COMPLEX Z,AI,J,Y
      AI=(0,1)
      H2= J(N,Z) - AI * Y(N,Z)
      RETURN
      END

      COMPLEX FUNCTION H2DE(N,Z)
      COMPLEX Z,H2
      N1=N-1
      H2DE= H2(N1,Z) - N * H2(N,Z) / Z
      RETURN
      END

      COMPLEX FUNCTION H2DOUDE(N,Z)
      COMPLEX Z,H2,H2DE
      H2DOUDE= -H2(N,Z) - H2DE(N,Z)/Z + H2(N,Z)/(Z*Z)
C               IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END
```

```
      COMPLEX FUNCTION G(N,K0,A,B,ER)
      COMPLEX K0,J,Y,JDE,YDE,KA,KB,GDIV
      KA = K0 * SQRT(ER) * A
      KB = K0 * SQRT(ER) * B
      G= -SQRT(ER) * ( J(N,KA)*YDE(N,KB)
     +                -Y(N,KA)*JDE(N,KB) )
      GDIV =  J(N,KA)*Y(N,KB) - Y(N,KA)*J(N,KB)
      IF(GDIV.EQ.0) GDIV = (1.E-20,1.E-20)
      G=G/GDIV
      RETURN
      END

      COMPLEX FUNCTION GDE(N,K0,A,B,ER)
      COMPLEX K0,J,Y,JDE,JDOUDE,YDE,YDOUDE,KA,KB,DENOG,NUMOG,FIR,SEC
      KA = K0 * SQRT(ER) * A
      KB = K0 * SQRT(ER) * B
      NUMOG = J(N,KA)*YDE(N,KB)
      NUMOG = NUMOG - Y(N,KA)*JDE(N,KB)
      DENOG = J(N,KA)*Y(N,KB)  - Y(N,KA)*J(N,KB)

      IF(DENOG.EQ.0) DENOG=(1.E-20,1.E-20)
      FIR = JDE(N,KA)*YDE(N,KB)*SQRT(ER)*A+J(N,KA)*YDOUDE(N,KB)
     +        *SQRT(ER)*B
      FIR = FIR - YDE(N,KA)*JDE(N,KB)*SQRT(ER)*A
      FIR = FIR - Y(N,KA)*JDOUDE(N,KB)*SQRT(ER)*B
      FIR = FIR * DENOG
      SEC = JDE(N,KA)*Y(N,KB)*SQRT(ER)*A+J(N,KA)*YDE(N,KB)*SQRT(ER)*B
      SEC = SEC - YDE(N,KA)*J(N,KB)*SQRT(ER)*A
      SEC = SEC - Y(N,KA)*JDE(N,KB)*SQRT(ER)*B
      SEC = SEC * NUMOG
      GDE = -(FIR -SEC) * SQRT(ER) / ( DENOG*DENOG )
      RETURN
      END

      COMPLEX FUNCTION DENO(N,K0,A,B,ER)
      COMPLEX K0,K0B,H2,H2DE,G
      K0B = K0 * B
      DENO = H2DE(N,K0B)   + G(N,K0,A,B,ER)*H2(N,K0B)
      RETURN
      END

      COMPLEX FUNCTION DENODE(N,K0,A,B,ER)
      COMPLEX K0,K0B,H2,H2DE,H2DOUDE,G,GDE
      K0B = K0 * B
      DENODE = H2DOUDE(N,K0B)*B   + GDE(N,K0,A,B,ER)*H2(N,K0B)
     +           + G(N,K0,A,B,ER)*H2DE(N,K0B)*B
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE LAGUER(AR,M,X,EPS,POLISH)
      COMPLEX AR(30),X,DX,X1,B,D,F,G,H
      COMPLEX SQ,GP,GM,G2,ZERO
      LOGICAL POLISH
      PARAMETER (ZERO=(0.,0.),TINY=1.E-15,MAXIT=100)
      IF(POLISH) THEN
         DXOLD=CABS(X)
         NPOL=0
      ENDIF
```

```
      DO 12 ITER=1,MAXIT
      B=AR(M+1)
      D=ZERO
      F=ZERO
        DO 11 J=M,1,-1
        F=X*F+D
        D=X*D+B
        B=X*B+AR(J)
11      CONTINUE
      IF(CABS(B).LE.TINY) THEN
        DX=ZERO
      ELSE IF(CABS(D).LE.TINY.AND.CABS(F).LE.TINY) THEN
        DX=CMPLX(CABS(B/AR(M+1))**(1./M),0.)
      ELSE
        G=D/B
        G2=G*G
        H=G2-2.*F/B
        SQ=CSQRT((M-1)*(M*H-G2))
        GP=G+SQ
        GM=G-SQ
        IF(CABS(GP).LT.CABS(GM)) GP=GM
        DX=M/GP
      ENDIF
      X1=X-DX
      IF(X.EQ.X1) RETURN
      X=X1
      IF(POLISH) THEN
        NPOL=NPOL+1
        CDX=CABS(DX)
        IF(NPOL.GT.9.AND.CDX.GE.DXOLD) RETURN
        DXOLD=CDX
      ELSE
        IF(CABS(DX).LE.EPS*CABS(X)) RETURN
      ENDIF
12    CONTINUE
       WRITE(6,*)'TOO MANY ITERATIONS'
       RETURN
       END


      SUBROUTINE ZROOTS(AR,M,ROOTS,POLISH)
      PARAMETER (EPS=1.E-6,MAXM=30)
      COMPLEX AR(30),ROOTS(M),AD(MAXM),X,B,C
      LOGICAL POLISH
      DO 11 J=1,M+1
        AD(J)=AR(J)
11    CONTINUE
      DO 13 J=M,1,-1
      X=CMPLX(0.,0.)
      CALL LAGUER(AD,J,X,EPS,.FALSE.)
      IF(ABS(AIMAG(X)).LE.2.*EPS**2*ABS(REAL(X)))
     #X=CMPLX(REAL(X),0.)
```

```
      ROOTS(J)=X
      B=AD(J+1)
      DO 12 JJ=J,1,-1
        C=AD(JJ)
        AD(JJ)=B
        B=X*B+C
12      CONTINUE
13    CONTINUE
      IF(POLISH) THEN
        DO 14 J=1,M
        CALL LAGUER(AR,M,ROOTS(J),EPS,.TRUE.)
14      CONTINUE
      ENDIF

      DO 16 J=2,M
      X=ROOTS(J)
      DO 15 I=J-1,1,-1
        IF(REAL(ROOTS(I)).LE.REAL(X)) GOTO 10
        ROOTS(I+1)=ROOTS(I)
15    CONTINUE
      I=0
10    ROOTS(I+1)=X
16    CONTINUE
      RETURN
      END
```

APPENDIX B

ALGORITHM TO REFINE THE ROOTS FOUND FROM

THE ROOT INTEGRAL ALGORITHM IN APPENDIX A

```
      PROGRAM MAIN(INPUT,OUTPUT)
      COMPLEX K0,J,JDE,JDOUDE,Y,YDE,YDOUDE,H2,H2DE,H2DOUDE,G,GDE
     +        ,DENO,DENODE,KA,KB,DENO0,DENO1,DENO2,NEWK0,DELK0
      WRITE(*,*) 'INPUT N'
      READ(*,*) N
      WRITE(*,*) ' INPUT ER '
      READ(*,*) ER
      A=1
      WRITE(*,*) ' INPUT B/A RATIO '
      READ(*,*) B
      DO 10 II=1,100
      WRITE(*,*) 'INPUT (K0.RE,K0.IM) '
      READ(*,*) K0
      WRITE(*,*) K0

9     FORMAT(2F35.7)
      KA= K0*SQRT(ER)
      KB=K0*B*SQRT(ER)

      WRITE(*,*)'ENTER "1" TO MODIFY  D_FUNCTION  '
      WRITE(*,*)'ENTER : '
      READ(*,*)MODIFY
      IF(MODIFY.EQ.0) GOTO 10
      MODOC=0
15    IF(CABS(DENO(N,K0,A,B,ER)).GE.0.0000001) THEN
      MODOC=MODOC+1
      WRITE(*,*)'NUMBERS OF MODIFYING :',MODOC
      DENO0 = DENO(N,K0,A,B,ER)
      NEWK0 = K0 + (0.0005,0.0005)
      DENO1 = DENO(N,NEWK0,A,B,ER)
      NEWK0 = K0 - (0.0005,0.0005)
      DENO2 = DENO(N,NEWK0,A,B,ER)
      DELK0 = DENO0 / ( (DENO1-DENO2)/(0.001,0.001) )
      K0    = K0 - DELK0
      WRITE(*,*)'K0          :',K0
      WRITE(*,*)'D_FUNCTION :',DENO(N,K0,A,B,ER)
      GOTO 15
      END IF

   10 CONTINUE

      STOP
      END
      COMPLEX FUNCTION J(N,Z)
      COMPLEX Z
      CALL BSSLJ(Z,N,J)
      RETURN
      END

      COMPLEX FUNCTION JDE(N,Z)
      COMPLEX Z,J
      N1=N-1
      JDE= J(N1,Z) - N * J(N,Z) / Z
      RETURN
      END

      COMPLEX FUNCTION JDOUDE(N,Z)
      COMPLEX Z,J,JDE
      JDOUDE= -J(N,Z) - JDE(N,Z)/Z + J(N,Z)/(Z*Z)
C             IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END
```

```
      COMPLEX FUNCTION Y(N,Z)
      COMPLEX Z
      CALL BSSLY(Z,N,Y)
      RETURN
      END

      COMPLEX FUNCTION YDE(N,Z)
      COMPLEX Z,Y
      N1=N-1
      YDE = Y(N1,Z) - N*Y(N,Z)/Z
      RETURN
      END

      COMPLEX FUNCTION YDOUDE(N,Z)
      COMPLEX Z,Y,YDE
      YDOUDE= -Y(N,Z) - YDE(N,Z)/Z + Y(N,Z)/(Z*Z)
C                 IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END

      COMPLEX FUNCTION H2(N,Z)
      COMPLEX Z,AI,J,Y
      AI=(0,1)
      H2= J(N,Z) - AI * Y(N,Z)
      RETURN
      END

      COMPLEX FUNCTION H2DE(N,Z)
      COMPLEX Z,H2
      N1=N-1
      H2DE= H2(N1,Z) - N * H2(N,Z) / Z
      RETURN
      END

      COMPLEX FUNCTION H2DOUDE(N,Z)
      COMPLEX Z,H2,H2DE
      H2DOUDE= -H2(N,Z) - H2DE(N,Z)/Z + H2(N,Z)/(Z*Z)
C                 IF N <> 1, THIS TERM MAY NEED MODIFY.
      RETURN
      END

      COMPLEX FUNCTION G(N,KO,A,B,ER)
      COMPLEX KO,J,Y,JDE,YDE,KA,KB,GDIV
      KA = KO * SQRT(ER) * A
      KB = KO * SQRT(ER) * B
      G= -SQRT(ER) * ( J(N,KA)*YDE(N,KB)
     +                     - Y(N,KA)*JDE(N,KB)   )
      GDIV =  J(N,KA)*Y(N,KB)  - Y(N,KA)*J(N,KB)
      IF(GDIV.EQ.0) GDIV = (1.E-20,1.E-20)
      G=G/GDIV
      RETURN
      END

      COMPLEX FUNCTION GDE(N,KO,A,B,ER)
      COMPLEX KO,J,Y,JDE,JDOUDE,YDE,YDOUDE,KA,KB,DENOG,NUMOG,FIR,SEC
      KA = KO * SQRT(ER) * A
      KB = KO * SQRT(ER) * B
      NUMOG = J(N,KA)*YDE(N,KB)
      NUMOG = NUMOG - Y(N,KA)*JDE(N.KB)
      DENOG = J(N,KA)*Y(N,KB)  - Y(N,KA)*J(N,KB)
```

```
      IF(DENOG.EQ.0) DENOG=(1.E-20,1.E-20)
      FIR = JDE(N,KA)*YDE(N,KB)*SQRT(ER)*A+ J(N,KA)*YDOUDE(N,KB)
     +      *SQRT(ER)*B
      FIR = FIR - YDE(N,KA)*JDE(N,KB)*SQRT(ER)*A
      FIR = FIR - Y(N,KA)*JDOUDE(N,KB)*SQRT(ER)*B
      FIR = FIR * DENOG
      SEC = JDE(N,KA)*Y(N,KB)*SQRT(ER)*A+J(N,KA)*YDE(N,KB)*SQRT(ER)*B
      SEC = SEC - YDE(N,KA)*J(N,KB)*SQRT(ER)*A
      SEC = SEC - Y(N,KA)*JDE(N,KB)*SQRT(ER)*B
      SEC = SEC * NUMOG
      GDE = -(FIR -SEC) * SQRT(ER) / ( DENOG*DENOG )
      RETURN
      END

      COMPLEX FUNCTION DENO(N,K0,A,B,ER)
      COMPLEX K0,K0B,H2,H2DE,G
      K0B = K0 * B
      DENO = H2DE(N,K0B)    + G(N,K0,A,B,ER)*H2(N,K0B)
      RETURN
      END

      COMPLEX FUNCTION DENODE(N,K0,A,B,ER)
      COMPLEX K0,K0B,H2,H2DE,H2DOUDE,G,GDE
      K0B = K0 * B
      DENODE = H2DOUDE(N,K0B)*B    + GDE(N,K0,A,B,ER)*H2(N,K0B)
     +         + G(N,K0,A,B,ER)*H2DE(N,K0B)*B
      RETURN
      END
```

APPENDIX C

ALGORITHM TO SEARCH A TABLE OF COMPLEX POLES FOR THE LEAST SQUARES OPTIMAL CHOICE OF THE CYLINDER COATING THICKNESS AND DIELECTRIC CONSTANT

```
PROGRAM SEARCHING(INPUT,OUTPUT);

VAR
   ERABRATIO : ARRAY [1..92,1..2] OF REAL;
   ROOTS : ARRAY [1..92,1..3,1..2]  OF REAL;
   TABLE,RESULT,ORDER : ARRAY [1..5,1..2] OF REAL;
   GIROOT : ARRAY [1..3,1..2] OF REAL;
   LOOP,L1,L2,L3 : INTEGER;
   DIFFVALUE : REAL;
   FIVAR : TEXT;

(*************************************************************)

PROCEDURE SETUP;
VAR I1,I2 : INTEGER;
BEGIN
  FOR I1 := 1 TO 5 DO
   BEGIN
    TABLE [I1,1] := 0;
    TABLE [I1,2] := I1*100;
  END;
END;
(*************************************************************)

PROCEDURE READ_KNOWN_ROOTS;

VAR
     I1,I2 : INTEGER;

BEGIN
  ASSIGN(FIVAR,'REFEREN');
  RESET(FIVAR);
  FOR I1 := 1 TO 92 DO
  BEGIN
    READ(FIVAR,ERABRATIO[I1,1],ERABRATIO[I1,2]);
  END;

  ASSIGN(FIVAR,'GROROOT');
  RESET(FIVAR);
  FOR I1 := 1 TO 92 DO
  BEGIN
    FOR I2 := 1 TO 3 DO
    BEGIN
      READ(FIVAR,ROOTS[I1,I2,1],ROOTS[I1,I2,2]);
    END;
  END;
END;      (* ENDS THE READING, ROOTS[44,3,2] FILLED.  *)
(*************************************************************)
PROCEDURE READ_GIVEN_ROOTS;
VAR I1 : INTEGER;
BEGIN
  FOR I1 := 1 TO 3 DO
  BEGIN
    WRITE('INPUT ',I1,'ST GIVEN ROOT.RE :=');
    READLN(GIROOT[I1,1]);
    WRITE('INPUT ',I1,'ST GIVEN ROOT.IM :=');
    READLN(GIROOT[I1,2]);
  END;
END;
(*************************************************************)
```

```
PROCEDURE SEARCH1(INDEX,VALUE :REAL);

VAR  I1,I2,POINTER,COUNTS : INTEGER;

BEGIN
  COUNTS := 1;
  FOR I1 := 1 TO 5 DO
  BEGIN
    IF VALUE > TABLE[I1,2] THEN COUNTS := COUNTS+1;
  END;

  FOR I2 := 4 DOWNTO COUNTS DO
  BEGIN
    POINTER := I2 + 1;
    TABLE[POINTER,1] := TABLE[I2,1];
    TABLE[POINTER,2] := TABLE[I2,2];
  END;
  TABLE[COUNTS,1] := INDEX;
  TABLE[COUNTS,2] := VALUE;
END;    (*   ENDS THE SEARCHING OF SEQUENCE FOR ONE ROOTS    *)

(************************************************************)
PROCEDURE CALCU_DIFF_SEARCH;
VAR
  I1,I2,POINTER : INTEGER;
  DIFF,COMP_DIFF : REAL;
BEGIN
  FOR I1 := 1 TO 5 DO
  BEGIN
    POINTER := TRUNC(TABLE [I1,1]);
    DIFF := 0;
    FOR I2 := 1 TO 3 DO
    BEGIN
      DIFF := SQR (ROOTS[POINTER,I2,1] - GIROOT[I2,1]) +
              SQR (ROOTS[POINTER,I2,2] - GIROOT[I2,2]) + DIFF;
    END;
    RESULT[I1,1] := POINTER;
    RESULT[I1,2] := DIFF;
    WRITELN(I1,RESULT[I1,1],RESULT[I1,2]);
  END;

  FOR I1 := 1 TO 5 DO
  BEGIN
    COMP_DIFF := 1000;
    FOR I2 := 1 TO 5 DO
    BEGIN
      IF COMP_DIFF > RESULT[I2,2] THEN
      BEGIN
       POINTER := I2;
        COMP_DIFF := RESULT[I2,2];
      END;
    END;

    ORDER[I1,1] := RESULT[POINTER,1];
    ORDER[I1,2] := RESULT[POINTER,2];
    RESULT[POINTER,2] := 2000;
  END;              (* ENDS THE ORDERING OF THE 5 DIFFERENCES *)
```

```
  FOR I1 := 1 TO 5 DO
  BEGIN
    WRITELN(I1,ORDER[I1,1],ORDER[I1,2]);
  END;
END;

(****************************************************************)
PROCEDURE NAMEIT;
VAR
  I1,I2,I3 : INTEGER;
BEGIN
  WRITELN('GIVEN ROOTS :');
  FOR I1 := 1 TO 3 DO
  BEGIN
    WRITELN('(',I1,')',GIROOT[I1,1],GIROOT[I1,2]);
  END;
  WRITELN;

  FOR I1 := 1 TO 3 DO
  BEGIN
    I3 := TRUNC( ORDER[I1,1] );
    WRITELN(I1,'ST CLOSE ROOTS; ER = ',ERABRATIO[I3,1],
                                'A/B RATIO =',ERABRATIO[I3,2]);
    FOR I2 := 1 TO 3 DO
    BEGIN
      WRITELN('(',I2,')',ROOTS[I3,I2,1],ROOTS[I3,I2,2]);
    END;
    WRITELN;
  END;
END;
(*****************************************************************)

BEGIN
  SETUP;
  READ_KNOWN_ROOTS;
  FOR LOOP := 1 TO 10 DO
  BEGIN
   READ_GIVEN_ROOTS;
    FOR L1 := 1 TO 44 DO
    BEGIN
       DIFFVALUE := SQR( ROOTS[L1,1,1] - GIROOT[1,1] )
                  + SQR( ROOTS[L1,1,2] - GIROOT[1,2] );

      SEARCH1(L1,DIFFVALUE);
    END;

    CALCU_DIFF_SEARCH;
    NAMEIT;
    WRITELN;WRITELN('SEARCH AGAIN');
  END;
END.
```

APPENDIX D

TABLE OF ROOTS (n = 0) FOR VARIOUS COMBINATIONS OF CYLIINDER COATING THICKNESS (b/a) AND DIELECTRIC CONSTANT (G_r)

```
ER          :=  2.56000000000000E+000
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  1.22710000000000E+001   ROOT.IM    :=   5.73130000000000E+000
ROOT.RE     :=  3.68140000000000E+001   ROOT.IM    :=   5.72840000000000E+000
ROOT.RE     :=  6.13585000000000E+001   ROOT.IM    :=   5.72810000000000E+000

ER          :=  2.56000000000000E+000
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  9.81630000000000E+000   ROOT.IM    :=   4.58700000000000E+000
ROOT.RE     :=  2.94511000000000E+001   ROOT.IM    :=   4.58300000000000E+000
ROOT.RE     :=  4.90870000000000E+001   ROOT.IM    :=   4.58300000000000E+000

ER          :=  2.56000000000000E+000
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  8.17970000000000E+000   ROOT.IM    :=   3.82350000000000E+000
ROOT.RE     :=  2.45420000000000E+001   ROOT.IM    :=   3.81940000000000E+000
ROOT.RE     :=  4.09050000000000E+001   ROOT.IM    :=   3.82000000000000E+000

ER          :=  2.56000000000000E+000
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  6.54300000000000E+000   ROOT.IM    :=   3.06100000000000E+000
ROOT.RE     :=  1.96330000000000E+001   ROOT.IM    :=   3.05600000000000E+000
ROOT.RE     :=  3.27240000000000E+001   ROOT.IM    :=   3.05521000000000E+000

ER          :=  2.56000000000000E+000
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  4.90590000000000E+000   ROOT.IM    :=   2.29870000000000E+000
ROOT.RE     :=  1.47240000000000E+001   ROOT.IM    :=   2.29230000000000E+000
ROOT.RE     :=  2.45421000000000E+001   ROOT.IM    :=   2.29160000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  1.57073500000000E+001   ROOT.IM    :=   5.49528000000000E+000
ROOT.RE     :=  4.71233900000000E+001   ROOT.IM    :=   5.49346000000000E+000
ROOT.RE     :=  7.85394000000000E+001   ROOT.IM    :=   5.49325000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  1.30893600000000E+001   ROOT.IM    :=   4.58026800000000E+000
ROOT.RE     :=  3.92225000000000E+001   ROOT.IM    :=   4.57793000000000E+000
ROOT.RE     :=  6.54494100000000E+001   ROOT.IM    :=   4.57769000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  9.81660000000000E+000   ROOT.IM    :=   3.43671000000000E+000
ROOT.RE     :=  2.94515400000000E+001   ROOT.IM    :=   3.43365000000000E+000
ROOT.RE     :=  4.90868200000000E+001   ROOT.IM    :=   3.43340000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  7.85280000000000E+000   ROOT.IM    :=   2.75087000000000E+000
ROOT.RE     :=  2.35608400000000E+001   ROOT.IM    :=   2.74712000000000E+000
ROOT.RE     :=  3.92692100000000E+001   ROOT.IM    :=   2.74675000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  6.54348000000000E+000   ROOT.IM    :=   2.29386000000000E+000
ROOT.RE     :=  1.96336500000000E+001   ROOT.IM    :=   2.28946000000000E+000
ROOT.RE     :=  3.27241000000000E+001   ROOT.IM    :=   2.28903000000000E+000
```

```
ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  5.23394000000000E+000   ROOT.IM     :=  1.83714000000000E+000
ROOT.RE     :=  1.57063700000000E+001   ROOT.IM     :=  1.83184000000000E+000
ROOT.RE     :=  2.61789300000000E+001   ROOT.IM     :=  1.83132000000000E+000

ER          :=  4.00000000000000E+000
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  3.92401000000000E+000   ROOT.IM     :=  1.38091000000000E+000
ROOT.RE     :=  1.17789200000000E+001   ROOT.IM     :=  1.37428000000000E+000
ROOT.RE     :=  1.96336600000000E+001   ROOT.IM     :=  1.37364000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  1.60314000000000E+001   ROOT.IM     :=  4.42640000000000E+000
ROOT.RE     :=  4.80953000000000E+001   ROOT.IM     :=  4.42500000000000E+000
ROOT.RE     :=  8.01592000000000E+001   ROOT.IM     :=  4.42450000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  1.28250000000000E+001   ROOT.IM     :=  3.54190000000000E+000
ROOT.RE     :=  3.84760000000000E+001   ROOT.IM     :=  3.54000000000000E+000
ROOT.RE     :=  6.41272000000000E+001   ROOT.IM     :=  3.54000000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  1.06873000000000E+001   ROOT.IM     :=  2.95240000000000E+000
ROOT.RE     :=  3.20630000000000E+001   ROOT.IM     :=  2.95000000000000E+000
ROOT.RE     :=  5.34390000000000E+001   ROOT.IM     :=  2.94980000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  8.01510000000000E+000   ROOT.IM     :=  2.21600000000000E+000
ROOT.RE     :=  2.40471000000000E+001   ROOT.IM     :=  2.21270000000000E+000
ROOT.RE     :=  4.00790000000000E+001   ROOT.IM     :=  2.21240000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  6.41155000000000E+000   ROOT.IM     :=  1.77416000000000E+000
ROOT.RE     :=  1.92373300000000E+001   ROOT.IM     :=  1.77033000000000E+000
ROOT.RE     :=  3.20631900000000E+001   ROOT.IM     :=  1.77020000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  5.34241000000000E+000   ROOT.IM     :=  1.47993000000000E+000
ROOT.RE     :=  1.60307900000000E+001   ROOT.IM     :=  1.47546000000000E+000
ROOT.RE     :=  2.67191100000000E+001   ROOT.IM     :=  1.47505000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  4.27306000000000E+000   ROOT.IM     :=  1.18597000000000E+000
ROOT.RE     :=  1.28241700000000E+001   ROOT.IM     :=  1.18062000000000E+000
ROOT.RE     :=  2.13750000000000E+001   ROOT.IM     :=  1.18013000000000E+000

ER          :=  6.00000000000000E+000
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  3.20330000000000E+000   ROOT.IM     :=  8.92400000000000E-001
ROOT.RE     :=  9.61740000000000E+000   ROOT.IM     :=  8.85800000000000E-001
ROOT.RE     :=  1.60310000000000E+001   ROOT.IM     :=  8.85200000000000E-001
```

```
ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  1.38837000000000E+001   ROOT.IM   :=   3.26780000000000E+000
ROOT.RE     :=  4.16517000000000E+001   ROOT.IM   :=   3.26620000000000E+000
ROOT.RE     :=  6.94198000000000E+001   ROOT.IM   :=   3.26600000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  1.11068000000000E+001   ROOT.IM   :=   2.61510000000000E+000
ROOT.RE     :=  3.33212000000000E+001   ROOT.IM   :=   2.61300000000000E+000
ROOT.RE     :=  5.55358000000000E+001   ROOT.IM   :=   2.61290000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  9.25540000000000E+000   ROOT.IM   :=   2.18010000000000E+000
ROOT.RE     :=  2.77675000000000E+001   ROOT.IM   :=   2.17760000000000E+000
ROOT.RE     :=  4.62797000000000E+001   ROOT.IM   :=   2.17740000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  6.94410000000000E+000   ROOT.IM   :=   1.63660000000000E+000
ROOT.RE     :=  2.08254000000000E+001   ROOT.IM   :=   1.63340000000000E+000
ROOT.RE     :=  3.47096000000000E+001   ROOT.IM   :=   1.63310000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  5.55238000000000E+000   ROOT.IM   :=   1.31075000000000E+000
ROOT.RE     :=  1.66601000000000E+001   ROOT.IM   :=   1.30690000000000E+000
ROOT.RE     :=  2.77675200000000E+001   ROOT.IM   :=   1.30657000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  4.62641000000000E+000   ROOT.IM   :=   1.09373000000000E+000
ROOT.RE     :=  1.38830600000000E+001   ROOT.IM   :=   1.08926000000000E+000
ROOT.RE     :=  2.31394200000000E+001   ROOT.IM   :=   1.08887000000000E+000

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  3.70230000000000E+000   ROOT.IM   :=   8.71640000000000E-001
ROOT.RE     :=  1.11060400000000E+001   ROOT.IM   :=   8.71640000000000E-001
ROOT.RE     :=  1.85112900000000E+001   ROOT.IM   :=   8.71180000000000E-001

ER          :=  8.00000000000000E+000
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  2.77360000000000E+000   ROOT.IM   :=   6.60600000000000E-001
ROOT.RE     :=  8.32900000000000E+000   ROOT.IM   :=   6.54100000000000E-001
ROOT.RE     :=  1.38831000000000E+001   ROOT.IM   :=   6.53500000000000E-001

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  1.24179000000000E+001   ROOT.IM   :=   2.59060000000000E+000
ROOT.RE     :=  3.72544000000000E+001   ROOT.IM   :=   2.58895000000000E+000
ROOT.RE     :=  6.20910000000000E+001   ROOT.IM   :=   2.58899000000000E+000

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  9.93413200000000E+000   ROOT.IM   :=   2.07330000000000E+000
ROOT.RE     :=  2.98034000000000E+001   ROOT.IM   :=   2.07130000000000E+000
ROOT.RE     :=  4.96727000000000E+001   ROOT.IM   :=   2.07110000000000E+000
```

```
ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  8.27820000000000E+000   ROOT.IM    :=   1.72860000000000E+000
ROOT.RE     :=  2.48360000000000E+001   ROOT.IM    :=   1.72610000000000E+000
ROOT.RE     :=  4.13938000000000E+001   ROOT.IM    :=   1.72590000000000E+000

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  6.20820000000000E+000   ROOT.IM    :=   1.29796000000000E+000
ROOT.RE     :=  1.86268000000000E+001   ROOT.IM    :=   1.29480000000000E+000
ROOT.RE     :=  3.10452000000000E+001   ROOT.IM    :=   1.29450000000000E+000

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  4.98270000000000E+000   ROOT.IM    :=   1.04554000000000E+000
ROOT.RE     :=  1.49078300000000E+001   ROOT.IM    :=   1.03160000000000E+000
ROOT.RE     :=  2.48367200000000E+001   ROOT.IM    :=   1.03618000000000E+000

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  4.13778000000000E+000   ROOT.IM    :=   8.67590000000000E-001
ROOT.RE     :=  1.24173800000000E+001   ROOT.IM    :=   8.63500000000000E-001
ROOT.RE     :=  2.06965300000000E+001   ROOT.IM    :=   8.63120000000000E-001

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  3.30930000000000E+000   ROOT.IM    :=   6.96290000000000E-001
ROOT.RE     :=  9.93350000000000E+000   ROOT.IM    :=   6.91030000000000E-001
ROOT.RE     :=  1.65570000000000E+001   ROOT.IM    :=   6.90500000000000E-001

ER          :=  1.00000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  2.48037000000000E+000   ROOT.IM    :=   5.24960000000000E-001
ROOT.RE     :=  7.44957000000000E+000   ROOT.IM    :=   5.18610000000000E-001
ROOT.RE     :=  1.24174000000000E+001   ROOT.IM    :=   5.18060000000000E-001

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  1.13359000000000E+001   ROOT.IM    :=   2.14620000000000E+000
ROOT.RE     :=  3.40085000000000E+001   ROOT.IM    :=   2.14451000000000E+000
ROOT.RE     :=  5.66811000000000E+001   ROOT.IM    :=   2.14436000000000E+000

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  9.06850000000000E+000   ROOT.IM    :=   1.71780000000000E+000
ROOT.RE     :=  2.72067000000000E+001   ROOT.IM    :=   1.71570000000000E+000
ROOT.RE     :=  4.53450000000000E+001   ROOT.IM    :=   1.71550000000000E+000

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  7.55690000000000E+000   ROOT.IM    :=   1.43230000000000E+000
ROOT.RE     :=  2.26721000000000E+001   ROOT.IM    :=   1.42985000000000E+000
ROOT.RE     :=  3.77872000000000E+001   ROOT.IM    :=   1.42964000000000E+000

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  5.66720000000000E+000   ROOT.IM    :=   1.07573000000000E+000
ROOT.RE     :=  1.70038300000000E+001   ROOT.IM    :=   1.07265000000000E+000
ROOT.RE     :=  2.83402800000000E+001   ROOT.IM    :=   1.07229000000000E+000
```

```
ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  4.53321000000000E+000  ROOT.IM    :=  8.62040000000000E-001
ROOT.RE     :=  1.36028300000000E+001  ROOT.IM    :=  8.58220000000000E-001
ROOT.RE     :=  2.26720800000000E+001  ROOT.IM    :=  8.57900000000000E-001

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  3.77707000000000E+000  ROOT.IM    :=  7.19750000000000E-001
ROOT.RE     :=  1.13354600000000E+001  ROOT.IM    :=  7.15350000000000E-001
ROOT.RE     :=  1.88932600000000E+001  ROOT.IM    :=  7.14970000000000E-001

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  3.02071000000000E+000  ROOT.IM    :=  5.77690000000000E-001
ROOT.RE     :=  9.06802000000000E+000  ROOT.IM    :=  5.72500000000000E-001
ROOT.RE     :=  1.51144000000000E+001  ROOT.IM    :=  5.72060000000000E-001

ER          :=  1.20000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  2.26391000000000E+000  ROOT.IM    :=  4.35910000000000E-001
ROOT.RE     :=  6.80048000000000E+000  ROOT.IM    :=  4.29710000000000E-001
ROOT.RE     :=  1.15534800000000E+001  ROOT.IM    :=  4.29710000000000E-001

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  1.01391000000000E+001  ROOT.IM    :=  1.70720000000000E+000
ROOT.RE     :=  3.04181000000000E+001  ROOT.IM    :=  1.70550000000000E+000
ROOT.RE     :=  5.06971000000000E+001  ROOT.IM    :=  1.70534000000000E+000

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  8.11110000000000E+000  ROOT.IM    :=  1.36660000000000E+000
ROOT.RE     :=  2.43344000000000E+001  ROOT.IM    :=  1.36450000000000E+000
ROOT.RE     :=  4.05576000000000E+001  ROOT.IM    :=  1.36430000000000E+000

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  6.75900000000000E+000  ROOT.IM    :=  1.13960000000000E+000
ROOT.RE     :=  2.02785000000000E+001  ROOT.IM    :=  1.13720000000000E+000
ROOT.RE     :=  3.37979000000000E+001  ROOT.IM    :=  1.13700000000000E+000

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  5.06880000000000E+000  ROOT.IM    :=  8.56190000000000E-001
ROOT.RE     :=  1.52087000000000E+001  ROOT.IM    :=  8.53040000000000E-001
ROOT.RE     :=  2.53483000000000E+001  ROOT.IM    :=  8.52780000000000E-001

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  4.05445000000000E+000  ROOT.IM    :=  6.86380000000000E-001
ROOT.RE     :=  1.21667300000000E+001  ROOT.IM    :=  6.82600000000000E-001
ROOT.RE     :=  2.02785200000000E+001  ROOT.IM    :=  6.82280000000000E-001

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  3.37808000000000E+000  ROOT.IM    :=  5.73340000000000E-001
ROOT.RE     :=  1.01387300000000E+001  ROOT.IM    :=  5.68990000000000E-001
ROOT.RE     :=  1.68986400000000E+001  ROOT.IM    :=  5.68630000000000E-001
```

```
ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  2.70149000000000E+000   ROOT.IM    :=  4.60490000000000E-001
ROOT.RE     :=  8.11067000000000E+000   ROOT.IM    :=  4.55420000000000E-001
ROOT.RE     :=  1.35187300000000E+001   ROOT.IM    :=  4.54980000000000E-001

ER          :=  1.50000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  2.02445000000000E+000   ROOT.IM    :=  3.47900000000000E-001
ROOT.RE     :=  6.08250000000000E+000   ROOT.IM    :=  3.41880000000000E-001
ROOT.RE     :=  1.01387600000000E+001   ROOT.IM    :=  3.41350000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  7.86536000000000E+000   ROOT.IM    :=  1.01556000000000E+000
ROOT.RE     :=  2.35617500000000E+001   ROOT.IM    :=  1.01390000000000E+000
ROOT.RE     :=  3.92698000000000E+001   ROOT.IM    :=  1.01370000000000E+000

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  6.28270000000000E+000   ROOT.IM    :=  8.13260000000000E-001
ROOT.RE     :=  1.88493000000000E+001   ROOT.IM    :=  8.11200000000000E-001
ROOT.RE     :=  3.14158000000000E+001   ROOT.IM    :=  8.11000000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  5.23530000000000E+000   ROOT.IM    :=  6.78500000000000E-001
ROOT.RE     :=  1.57077000000000E+001   ROOT.IM    :=  6.76090000000000E-001
ROOT.RE     :=  2.61798000000000E+001   ROOT.IM    :=  6.75900000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  3.92595000000000E+000   ROOT.IM    :=  5.10310000000000E-001
ROOT.RE     :=  1.17806000000000E+001   ROOT.IM    :=  5.07230000000000E-001
ROOT.RE     :=  1.96347200000000E+001   ROOT.IM    :=  5.06980000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  3.14014000000000E+000   ROOT.IM    :=  4.09600000000000E-001
ROOT.RE     :=  9.42429000000000E+000   ROOT.IM    :=  4.05950000000000E-001
ROOT.RE     :=  1.57077000000000E+001   ROOT.IM    :=  4.05640000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  2.61610000000000E+000   ROOT.IM    :=  3.42580000000000E-001
ROOT.RE     :=  7.85340000000000E+000   ROOT.IM    :=  3.38440000000000E-001
ROOT.RE     :=  1.30896000000000E+001   ROOT.IM    :=  3.38090000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  2.09187000000000E+000   ROOT.IM    :=  2.75700000000000E-001
ROOT.RE     :=  6.28246000000000E+000   ROOT.IM    :=  2.70970000000000E-001
ROOT.RE     :=  1.04715500000000E+001   ROOT.IM    :=  2.70560000000000E-001

ER          :=  2.50000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  1.56720000000000E+000   ROOT.IM    :=  2.09000000000000E-001
ROOT.RE     :=  4.71140000000000E+000   ROOT.IM    :=  2.03530000000000E-001
ROOT.RE     :=  7.85344000000000E+000   ROOT.IM    :=  2.03020000000000E-001
```

```
ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  6.20873000000000E+000   ROOT.IM    :=   6.32170000000000E-001
ROOT.RE     :=  1.86272000000000E+001   ROOT.IM    :=   6.30500000000000E-001
ROOT.RE     :=  3.10455000000000E+001   ROOT.IM    :=   6.30370000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  4.96670000000000E+000   ROOT.IM    :=   5.06520000000000E-001
ROOT.RE     :=  1.49017000000000E+001   ROOT.IM    :=   5.04500000000000E-001
ROOT.RE     :=  2.48363500000000E+001   ROOT.IM    :=   5.04320000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  4.13870000000000E+000   ROOT.IM    :=   4.22870000000000E-001
ROOT.RE     :=  1.24180000000000E+001   ROOT.IM    :=   4.20500000000000E-001
ROOT.RE     :=  2.06969000000000E+001   ROOT.IM    :=   4.20300000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  3.10341000000000E+000   ROOT.IM    :=   3.18500000000000E-001
ROOT.RE     :=  9.31335000000000E+000   ROOT.IM    :=   3.15540000000000E-001
ROOT.RE     :=  1.55226000000000E+001   ROOT.IM    :=   3.15290000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  2.48207000000000E+000   ROOT.IM    :=   2.56040000000000E-001
ROOT.RE     :=  7.45054000000000E+000   ROOT.IM    :=   2.52590000000000E-001
ROOT.RE     :=  1.24180000000000E+001   ROOT.IM    :=   2.52290000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  2.06769000000000E+000   ROOT.IM    :=   2.14510000000000E-001
ROOT.RE     :=  6.20863000000000E+000   ROOT.IM    :=   2.10640000000000E-001
ROOT.RE     :=  1.03482500000000E+001   ROOT.IM    :=   2.10290000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  1.65311000000000E+000   ROOT.IM    :=   1.73070000000000E-001
ROOT.RE     :=  4.96669000000000E+000   ROOT.IM    :=   1.68710000000000E-001
ROOT.RE     :=  8.27848000000000E+000   ROOT.IM    :=   1.68310000000000E-001

ER          :=  4.00000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  1.28320000000000E+000   ROOT.IM    :=   1.31730000000000E-001
ROOT.RE     :=  3.72470000000000E+000   ROOT.IM    :=   1.26800000000000E-001
ROOT.RE     :=  6.20867000000000E+000   ROOT.IM    :=   1.26340000000000E-001

ER          :=  6.00000000000000E+001
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  5.06930000000000E+000   ROOT.IM    :=   4.20860000000000E-001
ROOT.RE     :=  1.52090000000000E+001   ROOT.IM    :=   4.19220000000000E-001
ROOT.RE     :=  2.53485000000000E+001   ROOT.IM    :=   4.19100000000000E-001

ER          :=  6.00000000000000E+001
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  4.05517000000000E+000   ROOT.IM    :=   3.37450000000000E-001
ROOT.RE     :=  1.21672000000000E+001   ROOT.IM    :=   3.35460000000000E-001
ROOT.RE     :=  2.02788000000000E+001   ROOT.IM    :=   3.35300000000000E-001
```

```
ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.0600000000000E+000
ROOT.RE      :=  3.3390100000000E+000   ROOT.IM    :=  2.8194000000000E-001
ROOT.RE      :=  1.0139240000000E+001   ROOT.IM    :=  2.7964000000000E-001
ROOT.RE      :=  1.6898960000000E+001   ROOT.IM    :=  2.7945000000000E-001

ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.0800000000000E+000
ROOT.RE      :=  2.5336000000000E+000   ROOT.IM    :=  2.1272000000000E-001
ROOT.RE      :=  7.6043000000000E+000   ROOT.IM    :=  2.0990000000000E-001
ROOT.RE      :=  1.2674200000000E+001   ROOT.IM    :=  2.0960000000000E-001

ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.1000000000000E+000
ROOT.RE      :=  2.0262000000000E+000   ROOT.IM    :=  1.7132000000000E-001
ROOT.RE      :=  6.0833300000000E+000   ROOT.IM    :=  1.6806500000000E-001
ROOT.RE      :=  1.0139250000000E+001   ROOT.IM    :=  1.6777000000000E-001

ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.1200000000000E+000
ROOT.RE      :=  1.6877900000000E+000   ROOT.IM    :=  1.4379000000000E-001
ROOT.RE      :=  5.0693000000000E+000   ROOT.IM    :=  1.4020000000000E-001
ROOT.RE      :=  8.4493000000000E+000   ROOT.IM    :=  1.3986000000000E-001

ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.1500000000000E+000
ROOT.RE      :=  1.3491800000000E+000   ROOT.IM    :=  1.1633000000000E-001
ROOT.RE      :=  4.0552400000000E+000   ROOT.IM    :=  1.1236000000000E-001
ROOT.RE      :=  6.7593400000000E+000   ROOT.IM    :=  1.1196000000000E-001

ER           :=  6.0000000000000E+001
B/A RATIO    :=  1.2000000000000E+000
ROOT.RE      :=  1.0103000000000E+000   ROOT.IM    :=  8.8900000000000E-002
ROOT.RE      :=  3.0411000000000E+000   ROOT.IM    :=  8.4540000000000E-002
ROOT.RE      :=  5.0693500000000E+000   ROOT.IM    :=  8.4070000000000E-002

ER           :=  8.0000000000000E+001
B/A RATIO    :=  1.0400000000000E+000
ROOT.RE      :=  4.3900500000000E+000   ROOT.IM    :=  3.1564000000000E-001
ROOT.RE      :=  1.3171400000000E+001   ROOT.IM    :=  3.1400000000000E-001
ROOT.RE      :=  2.1952000000000E+001   ROOT.IM    :=  3.1390000000000E-001

ER           :=  8.0000000000000E+001
B/A RATIO    :=  1.0500000000000E+000
ROOT.RE      :=  3.5118000000000E+000   ROOT.IM    :=  2.5325000000000E-001
ROOT.RE      :=  1.0537100000000E+001   ROOT.IM    :=  2.5131000000000E-001
ROOT.RE      :=  1.7561950000000E+001   ROOT.IM    :=  2.5114000000000E-001

ER           :=  8.0000000000000E+001
B/A RATIO    :=  1.0600000000000E+000
ROOT.RE      :=  2.9261400000000E+000   ROOT.IM    :=  2.1174000000000E-001
ROOT.RE      :=  8.7808000000000E+000   ROOT.IM    :=  2.0951000000000E-001
ROOT.RE      :=  1.4634930000000E+001   ROOT.IM    :=  2.0932000000000E-001

ER           :=  8.0000000000000E+001
B/A RATIO    :=  1.0800000000000E+000
ROOT.RE      :=  2.1939500000000E+000   ROOT.IM    :=  1.6000000000000E-001
ROOT.RE      :=  6.5855000000000E+000   ROOT.IM    :=  1.5729000000000E-001
ROOT.RE      :=  1.0976130000000E+001   ROOT.IM    :=  1.5705000000000E-001
```

```
ER          :=  8.00000000000000E+001
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  1.75445000000000E+000   ROOT.IM    :=   1.29070000000000E-001
ROOT.RE     :=  5.26830000000000E+000   ROOT.IM    :=   1.25980000000000E-001
ROOT.RE     :=  8.78084000000000E+000   ROOT.IM    :=   1.25690000000000E-001

ER          :=  8.00000000000000E+001
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  1.46132000000000E+000   ROOT.IM    :=   1.08550000000000E-001
ROOT.RE     :=  4.39012000000000E+000   ROOT.IM    :=   1.05130000000000E-001
ROOT.RE     :=  7.31730000000000E+000   ROOT.IM    :=   1.04790000000000E-001

ER          :=  8.00000000000000E+001
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  1.16802000000000E+000   ROOT.IM    :=   8.79800000000000E-002
ROOT.RE     :=  3.51190000000000E+000   ROOT.IM    :=   8.42900000000000E-002
ROOT.RE     :=  5.85375000000000E+000   ROOT.IM    :=   8.39100000000000E-002

ER          :=  8.00000000000000E+001
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  8.74460000000000E-001   ROOT.IM    :=   6.75000000000000E-002
ROOT.RE     :=  2.63360000000000E+000   ROOT.IM    :=   6.34800000000000E-002
ROOT.RE     :=  4.37017000000000E+000   ROOT.IM    :=   6.30400000000000E-002

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.04000000000000E+000
ROOT.RE     :=  3.92650000000000E+000   ROOT.IM    :=   2.52640000000000E-001
ROOT.RE     :=  1.17808600000000E+001   ROOT.IM    :=   2.51050000000000E-001
ROOT.RE     :=  1.96349000000000E+001   ROOT.IM    :=   2.50910000000000E-001

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.05000000000000E+000
ROOT.RE     :=  3.14090000000000E+000   ROOT.IM    :=   2.02830000000000E-001
ROOT.RE     :=  9.42464000000000E+000   ROOT.IM    :=   2.00930000000000E-001
ROOT.RE     :=  1.57078800000000E+001   ROOT.IM    :=   2.00760000000000E-001

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.06000000000000E+000
ROOT.RE     :=  2.61710000000000E+000   ROOT.IM    :=   1.69700000000000E-001
ROOT.RE     :=  7.85381000000000E+000   ROOT.IM    :=   1.67500000000000E-001
ROOT.RE     :=  1.30899000000000E+001   ROOT.IM    :=   1.67330000000000E-001

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.08000000000000E+000
ROOT.RE     :=  1.96210000000000E+000   ROOT.IM    :=   1.28400000000000E-001
ROOT.RE     :=  5.89020000000000E+000   ROOT.IM    :=   1.25800000000000E-001
ROOT.RE     :=  9.81735000000000E+000   ROOT.IM    :=   1.25550000000000E-001

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.10000000000000E+000
ROOT.RE     :=  1.56900000000000E+000   ROOT.IM    :=   1.03730000000000E-001
ROOT.RE     :=  4.71210000000000E+000   ROOT.IM    :=   1.00787000000000E-001
ROOT.RE     :=  7.85383000000000E+000   ROOT.IM    :=   1.00502000000000E-001

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.12000000000000E+000
ROOT.RE     :=  1.30700000000000E+000   ROOT.IM    :=   8.70000000000000E-002
ROOT.RE     :=  3.92700000000000E+000   ROOT.IM    :=   8.41000000000000E-002
ROOT.RE     :=  6.54479000000000E+000   ROOT.IM    :=   8.38000000000000E-002
```

```
ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.15000000000000E+000
ROOT.RE     :=  1.04440000000000E+000  ROOT.IM    :=  7.09400000000000E-002
ROOT.RE     :=  3.14112000000000E+000  ROOT.IM    :=  6.74900000000000E-002
ROOT.RE     :=  5.23575000000000E+000  ROOT.IM    :=  6.71200000000000E-002

ER          :=  1.00000000000000E+002
B/A RATIO   :=  1.20000000000000E+000
ROOT.RE     :=  7.81800000000000E-001  ROOT.IM    :=  5.46000000000000E-002
ROOT.RE     :=  2.35570000000000E+000  ROOT.IM    :=  5.09000000000000E-002
ROOT.RE     :=  5.49760000000000E+000  ROOT.IM    :=  5.03000000000000E-002
```