389741-3-T

Senior, T.B.A.

THE UNIVERSITY OF MICHIGAN

COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

Radiation Laboratory

SCATTERING BY A NARROW GAP

T.B.A. Senior, K. Sarabandi, and J.R. Natzke
Radiation Laboratory
Department of Electrical Engineering
and Computer Science
The University of Michigan
Ann Arbor, MI 48109

April, 1989

McDonnell Aircraft Company
St. Louis, MO 63166

Ann Arbor, Michigan

Abstract

For a plane wave incident on a cavity-backed gap in a perfectly conducting plane, the coupled integral equations for the induced currents have been solved numerically and the far field scattering computed. The results are compared with a quasi-analytic solution previously derived, and for a narrow gap the agreement is excellent for all cavity geometries and for all material fillings that have been tested.

TABLE OF CONTENTS

		Page #
I.	INTRODUCTION	1
II.	FORMULATION	2
	2.1 H-Polarization	4
	2.2 E-Polarization	8
III.	QUASI-ANALYTICAL SOLUTION	11
IV.	NUMERICAL RESULTS	19
V.	CONCLUSIONS	36
	REFERENCES	37
	APPENDIX A: MOMENT METHOD SOLUTION OF THE COUPLED INTEGRAL EQUATIONS	38
	A.1 H-Polarization	38
	A.2 E-Polarization	43
	A.3 Program Listings	45
	APPENDIX B: PROGRAM LISTING FOR QUASI-ANALYTICAL SOLUTION	65

1. Introduction

A topic of some concern in radar cross section studies is the scattering from the gap or crack that may exist where two component parts of a target come together. Even if the crack is wholly or partially filled with a material, it can still provide a significant contribution to the overall scattering pattern of the target, and it is then necessary to develop methods for predicting the scattering.

One method for doing this was described recently [1]. For a plane wave of either principal polarization incident on a narrow ($kw << 1$) resistive strip insert in an otherwise perfectly conducting plane, the low frequency approximations to the integral equations for the currents induced in the strip were solved in a quasi-analytic manner, leading to expressions for the far zone scattered field that are accurate for almost any resistivity R of the insert. If, instead, the insert is characterized by a surface impedance η, the results differ only in having R replaced by $\eta/2$ and the scattered field doubled, and this suggests that for a narrow gap backed by a cavity, the scattered field can be obtained by identifying η with the impedance looking into the cavity.

An alternative approach is to use the equivalence principle [2] to develop coupled integral equations for the electric and magnetic currents which exist on the walls of the cavity and in the aperture, and this is the method employed here. For an incident plane wave either H- or E-polarized, the integral equations are derived for a cavity of arbitrary shape filled with a homogeneous material. The equations are solved by the moment method and data for a variety of simple cavities are presented. For gap widths which are electrically small the results are compared with those obtained using the previous method. The agreement is excellent, and confirms the utility of the original method [1] as an accurate and simple design tool.

2. Formulation

The problem considered is the two-dimensional one shown in Figure 1. The plane y = 0 is perfectly conducting apart from the aperture A: - w/2 < x < w/2, which forms the entrance to a cavity whose walls S are also perfectly conducting. The cavity is filled with a homogeneous dielectric material of permittivity $\varepsilon_1 = \varepsilon_r \varepsilon$ and permeability $\mu_1 = \mu_r \mu$, where the quantities without subscripts refer to free space. A plane wave of either principal polarization is incident on the surface y = 0 from above, and we choose

$$\overline{H}^i = \hat{z}\, e^{-ik(x\cos\phi_o + y\sin\phi_o)} \tag{1}$$

for H-polarization, and

$$\overline{E}^i = \hat{z}\, e^{-ik(x\cos\phi_o + y\sin\phi_o)} \tag{2}$$

for E-polarization, where k is the propagation constant in the free space region above the surface. A time factor $e^{-i\omega t}$ is assumed and suppressed.

In the far zone of the gap the scattered field can be written as

$$\overline{H}^s = \hat{z}\sqrt{\frac{2}{\pi k\rho}}\, e^{i(k\rho - \pi/4)}\, P_H(\phi, \phi_o)$$

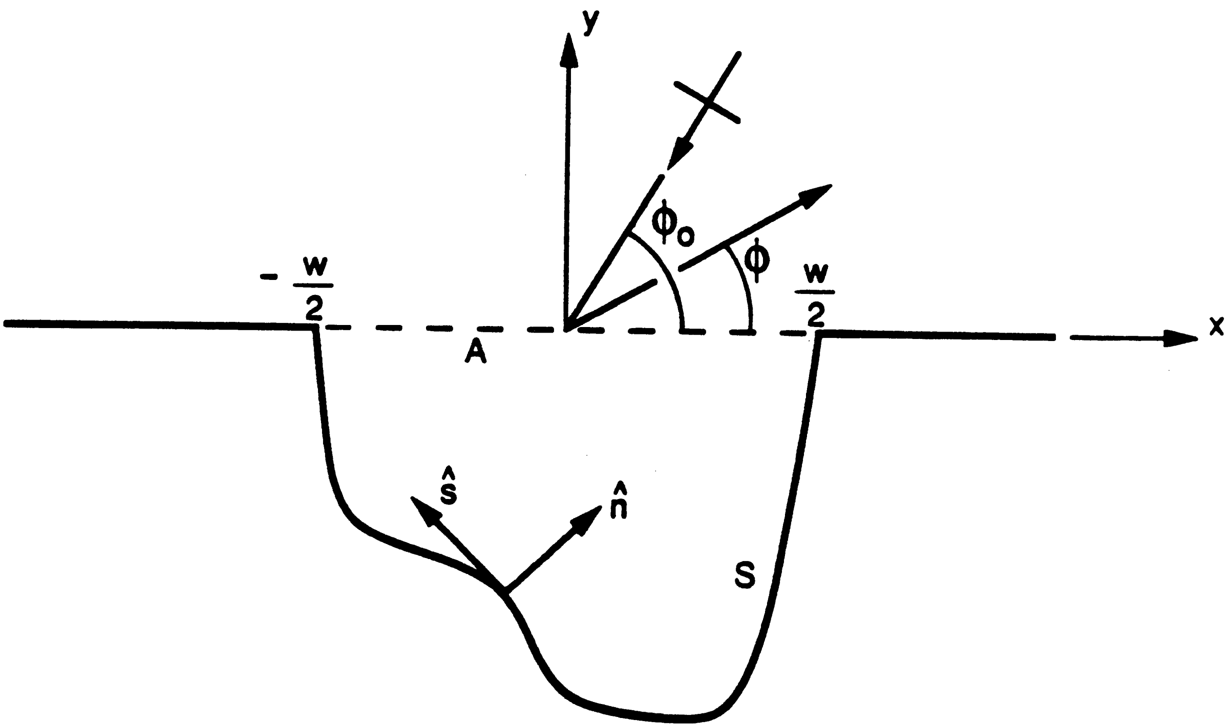

Fig. 1. Gap geometry.

for H-polarization, with a similar result for E-polarization, and the task is to determine the far field amplitudes $P_{H,E}(\phi, \phi_o)$ with particular emphasis on the case of a narrow gap ($kw \leq 1$).

2.1 H-Polarization

We consider first the free space region $y > 0$. Using Green's theorem in conjunction with the half space Green's function

$$G_1 = \frac{i}{4}\left\{H_0^{(1)}\left(k\sqrt{(x-x')^2 + (y-y')^2}\right) + H_0^{(1)}\left(k\sqrt{(x-x')^2 + (y+y')^2}\right)\right\} ,$$

the scattered field can be attributed to a magnetic current $\bar{J}^* = -\hat{y} \times \bar{E}$ in the aperture, and the total magnetic field is then

$$H_z(x,y) = H_z^i(x,y) + H_z^r(x,y) - \frac{kY}{2}\int_{-w/2}^{w/2} J_z^*(x')\, H_0^{(1)}\left(k\sqrt{(x-x')^2 + y^2}\right) dx'$$

where

$$H_z^r = e^{-ik(x\cos\phi_o - y\sin\phi_o)}$$

is the reflected plane wave and Y (= $1/Z$) is the intrinsic admittance of free space. Hence,

$$P_H(\phi, \phi_o) = -\frac{kY}{2}\int_{-w/2}^{w/2} J_z^*(x')\, e^{-ikx'\cos\phi}\, dx' \quad . \tag{3}$$

and in the aperture

$$H_z(x,o) = 2e^{-ikx\cos\phi_o} - \frac{kY}{2}\int_{-w/2}^{w/2} J_z^*(x')\, H_0^{(1)}(k|x-x'|)\, dx' \quad . \tag{4}$$

We now turn to the region y < 0 occupied by the cavity. In accordance with the equivalence principle [2] it is assumed that the gap is closed with a perfect conductor, and that a magnetic current $-\bar{J}^*$ is placed just below, thereby ensuring the continuity of the tangential electric field in the open gap. The magnetic Hertz vector is therefore

$$\bar{\Pi}^*(x,y) = \frac{Y}{4k\mu_r}\int_{-w/2}^{w/2} \bar{J}^*(x')\, H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + y^2}\right) dx' \tag{5}$$

and since $\bar{J}^* = \hat{z}\, J_z^*$, the magnetic field produced is

$$\bar{H}^{(1)}(x,y) = \nabla \times \nabla \times \bar{\Pi}^* = \hat{z}\,\frac{kY}{4}\varepsilon_r \int_{-w/2}^{w/2} J_z^*(x')\, H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + y^2}\right) dx'$$

where $k_1 = k\sqrt{\varepsilon_r\mu_r}$ is the propagation constant. The electric current $\bar{J} = \hat{n} \times \bar{H}$ on the cavity walls S and in the (closed) aperture A also implies an electric Hertz vector

$$\bar{\Pi}(x,y) = -\frac{Z}{4k\varepsilon_r}\int_{S+A} \bar{J}(s')\, H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + (y-y')^2}\right) ds' \quad , \tag{6}$$

and the corresponding magnetic field is

$$\bar{H}^{(2)}(x,y) = -ikY\,\varepsilon_r\,\nabla \times \bar{\Pi}$$

$$= \frac{i}{4}\int_{S+A} \nabla H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + (y-y')^2}\right) \times \bar{J}(s')\,ds'$$

where the tangential unit vector $\hat{s}$ is such that $\hat{n}, \hat{s}, \hat{z}$ form a right-handed system with $\hat{n}$ directed into the cavity. Clearly, $\bar{J}(s) = \hat{s}\,J_s(s)$, and in the aperture, $\hat{s} = \hat{x}$.

The total magnetic field is $\bar{H} = \bar{H}^{(1)} + \bar{H}^{(2)}$, and by allowing the observation point to approach the boundary of the closed cavity, we can construct an integral equation for the currents. We find

$$\bar{J}(s) = (\hat{n} \times \hat{z})\,\frac{kY}{4}\,\varepsilon_r \int_{-w/2}^{w/2} J_z^{*}(x')\,H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + y^2}\right) dx'$$

$$+ \lim_{(x,y)\to S+A}\ \hat{n} \times \frac{i}{4}\int_{S+A} J_s(s')\,\nabla H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + (y-y')^2}\right) \times \hat{s}\,'\,ds'$$

giving

$$J_s(s) = \frac{kY}{2}\,\varepsilon_r \int_{-w/2}^{w/2} J_z^{*}(x')\,H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + y^2}\right) dx'$$

$$+ \frac{ik_1}{2}\int_{S+A} J_s(s')\,\sin\gamma'\,H_1^{(1)}\left(k_1\sqrt{(x-x')^2 + (y-y')^2}\right) ds' \qquad (7)$$

where

$$\sin\gamma' = \hat{z} \cdot \frac{(x-x')\,\hat{x} + (y-y')\,\hat{y}}{\sqrt{(x-x')^2 + (y-y')^2}} \times \hat{s}\,' \quad , \qquad (8)$$

valid at all points of S and A.

The only remaining task is to enforce the continuity of H_z through the aperture. When the observation point is in the aperture

$$H_z(x,o) = \frac{kY}{4} \varepsilon_r \int_{-w/2}^{w/2} \dot{J_z}(x')\, H_0^{(1)}\left(k_1 |x-x'|\right) dx'$$

$$+ \lim_{y \to 0} \hat{z}\, \frac{i}{4} \int_{S+A} J_s(s')\, \nabla H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) \times \hat{s}'\, ds'$$

and therefore

$$H_z(x,o) = \frac{kY}{4} \varepsilon_r \int_{-w/2}^{w/2} \dot{J_z}(x')\, H_0^{(1)}\left(k_1 |x-x'|\right) dx' + \frac{1}{2} J_s(s)$$

$$+ \frac{ik_1}{4} \int_{S+A} J_s(s') \sin\gamma'\, H_1^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) ds' \quad . \tag{9}$$

When this is equated to the expression (4) for $H_z(x,o)$ on the outside of the gap, we obtain

$$2\, e^{-ikx \cos\phi_o} = \frac{kY}{2} \int_{-w/2}^{w/2} \dot{J_z}(x')\, H_0^{(1)}\left(k |x-x'|\right) dx' + \frac{kY}{4} \varepsilon_r \int_{-w/2}^{w/2} \dot{J_z}(x')\, H_0^{(1)}\left(k_1 |x-x'|\right) dx'$$

$$+ \frac{1}{2} J_s(x) + \frac{ik_1}{4} \int_{S+A} J_s(s') \sin\gamma'\, H_1^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) ds' \tag{10}$$

valid for $-w/2 < x < w/2$. Since (7) is also valid in A, it can be used to simplify (10) by eliminating two of the integrals. The result is

$$J_s(x) = 2e^{-ikx\cos\phi_o} - \frac{kY}{2} \int_{-w/2}^{w/2} J_z^*(x')\, H_0^{(1)}\left(k\,|x-x'|\right) dx' \tag{11}$$

valid for x in A, and (7) and (11) constitute a pair of coupled integral equations for $J_z^*(x)$ and $J_s(s)$. These are the equations that will be used, and we note the similarity of (11) and (4).

When the maximum dimension of the cavity is electrically small, the Hankel function $H_0^{(1)}$ can be replaced by its logarithmic approximation, and though this does not significantly simplify the numerical solution of (7) and (11), the fact that $e^{-ikx\cos\phi_o}$ can also be replaced by unity shows that $J_z^*(x)$ and $J_s(s)$ are aspect independent. The same approximation to (3) then leads to a far field amplitude which is independent of ϕ and ϕ_0, and this is a feature of the low frequency situation.

2.2 E-Polarization

The procedure is similar to that given above. For the region y > 0 Green's theorem in conjunction with the Green's function

$$G_2 = \frac{i}{4}\left\{H_0^{(1)}\left(k\sqrt{(x-x')^2 + (y-y')^2}\right) - H_0^{(1)}\left(k\sqrt{(x-x')^2 + (y+y')^2}\right)\right\}$$

gives

$$E_z = E_z^i + E_z^r + \frac{i}{2}\frac{\partial}{\partial y}\int_{-w/2}^{w/2} J_x^*(x')\, H_0^{(1)}\left(k\sqrt{(x-x')^2 + y^2}\right) dx'$$

where $\bar{J}^* = \hat{x}\, J_x^*$ is the assumed magnetic current in the gap and

$$E_z^r = - e^{-ik(x\cos\phi_o - y\sin\phi_o)}$$

is the reflected plane wave. Hence

$$P_E(\phi, \phi_o) = -\frac{k}{2}\sin\phi \int_{-w/2}^{w/2} J_x^*(x')\, e^{-ikx'\cos\phi}\, dx' \quad , \tag{12}$$

and since $H_x = -\frac{iY}{k}\frac{\partial E_z}{\partial y}$, the tangential component of the magnetic field in the aperture is

$$H_x(x,o) = -2Y\sin\phi_o\, e^{-ikx\cos\phi_o} - \frac{kY}{2}\left(1 + \frac{1}{k^2}\frac{\partial^2}{\partial x^2}\right)\int_{-w/2}^{w/2} J_x^*(x')\, H_0^{(1)}\big(k\,|x-x'|\big)\, dx' \ . \tag{13}$$

In the region $y < 0$ occupied by the cavity, the field can be attributed to the magnetic Hertz vector (5) with $\bar{J}^* = \hat{x}\, J_x$ and the electric Hertz vector (6) with $\bar{J} = \hat{z}\, J_z$.

The magnetic field is therefore

$$\bar{H}(x,y) = \nabla \times \nabla \times \frac{Y}{4k\mu_r}\int_{-w/2}^{w/2} \bar{J}^*(x')\, H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + y^2}\right) dx'$$

$$+ \frac{i}{4}\int_{S+A} \nabla H_0^{(1)}\left(k_1\sqrt{(x-x')^2 + (y-y')^2}\right) \times \bar{J}(s')\, ds' \quad ,$$

and by allowing the observation point to approach the boundary of the closed cavity, we obtain the integral equation

$$J_z(s) = \frac{Y}{2k\mu_r} (\hat{n} \bullet \nabla) \frac{\partial}{\partial y} \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) dx'$$

$$+ \frac{ik_1}{2} \int_{S+A} J_z(s') \sin\gamma\, H_1^{(1)}\left(k_1 \sqrt{(x-x')^2 + (y-y')^2}\right) ds' \qquad (14)$$

where

$$\sin\gamma = \hat{z} \bullet \frac{(x-x')\,\hat{x} + (y-y')\,\hat{y}}{\sqrt{(x-x')^2 + (y-y')^2}} \times \hat{s} \qquad (15)$$

valid at all points of S and A.

When the observation point is in the aperture,

$$H_x(x,o) = \frac{kY}{4} \varepsilon_r \left(1 + \frac{1}{k_1^2} \frac{\partial^2}{\partial x^2}\right) \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k_1 |x-x'|\right) dx' + \frac{1}{2} J_z(x)$$

$$+ \frac{ik_1}{4} \int_{S+A} J_z(s') \sin\gamma\, H_1^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) ds' \quad ,$$

and on equating this to the expression (13) for the magnetic field on the outside of the gap, we obtain

$$- 2Y \sin\phi_o\, e^{-ikx \cos\phi_o} = \frac{kY}{2} \left(1 + \frac{1}{k^2} \frac{\partial^2}{\partial x^2}\right) \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k |x-x'|\right) dx'$$

$$+ \frac{kY}{4} \varepsilon_r \left(1 + \frac{1}{k_1^2} \frac{\partial^2}{\partial x^2}\right) \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k_1 |x-x'|\right) dx'$$

$$+ \frac{1}{2} J_z(x) + \frac{ik_1}{4} \int_{S+A} J_z(s') \sin\gamma\, H_1^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) ds' \qquad (16)$$

valid for x in A. This can be simplified using (14) and the result is

$$J_z(x) = -2Y \sin\phi_o \, e^{-ikx\cos\phi_o} - \frac{kY}{2}\left(1 + \frac{1}{k^2}\frac{\partial^2}{\partial x^2}\right) \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k\,|x-x'|\right) dx' \qquad (17)$$

for x in A in accordance with (13), and (14) and (17) constitute a pair of coupled integral equations for $\dot{J}_x(x)$ and $J_z(s)$.

There is a third integral equation that can be developed and this has some advantages for numerical purposes. In the region $y < 0$ the electric field produced by the electric and magnetic Hertz vectors is

$$\bar{E}(x,y) = -\frac{kZ\mu_r}{4} \int_{S+A} J_z(s')\, H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + (y-y')^2}\right) ds' \, \hat{z}$$

$$-\frac{i}{4}\frac{\partial}{\partial y} \int_{-w/2}^{w/2} \dot{J}_x(x')\, H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) dx' \, \hat{z} \; ,$$

and when the boundary condition on the perfectly conducting surface is applied, we find

$$\dot{J}_x(x) = \frac{i}{2} \int_{-w/2}^{w/2} \dot{J}_x(x') \frac{\partial}{\partial y} H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + y^2}\right) dx'$$

$$+\frac{kZ\mu_r}{2} \int_{S+A} J_z(s')\, H_0^{(1)}\left(k_1 \sqrt{(x-x')^2 + (y-y')^2}\right) ds' \qquad (18)$$

valid on S + A. Of course, $\dot{J}_x(x)$ is non-zero only in A, and (17) and (18) are the pair of integral equations used to compute $\dot{J}_x(x)$ and $J_z(s)$.

3. Quasi-Analytical Solution

An alternative approach was proposed by Senior and Volakis [1]. In effect, the problem which they considered is a uniform impedance insert in an otherwise perfectly conducting plane. If η is the surface impedance, the integral equations for H- and E-polarizations are identical to (4) and (13) respectively, with

$$H_z(x,o) = \frac{1}{\eta} J_z^*(x), \quad H_x(x,o) = -\frac{1}{\eta} J_x^*(x) \tag{19}$$

at the insert. At low frequencies for which kw << 1 the integral equations can be simplified, and for H-polarization it is found that

$$\frac{1}{\pi} \int_{-1}^{1} J_2(\zeta') \, \ln |\zeta' - \zeta| \, d\zeta' = 1 + a J_2(\zeta) \tag{20}$$

for $-1 < \zeta < 1$ with

$$a = \frac{2i}{kw} \frac{Z}{\eta} \ . \tag{21}$$

$J_2(\zeta)$ is a modified current in terms of which

$$P_H(\phi, \phi_o) = i\pi \left\{ A + \frac{1}{K_H(a)} \right\}^{-1} \tag{22}$$

with

$$K_H(a) = \frac{1}{\pi} \int_{-1}^{1} J_2(\zeta) \, d\zeta \tag{23}$$

and

$$A = \ln \frac{kw}{4} + \gamma - i\frac{\pi}{2}$$

where $\gamma = 0.5772157....$ is Euler's constant. We observe that $P_H(\phi, \phi_o)$ is independent of ϕ and ϕ_o, and since $K_H(a)$ is real if a is,

$$\left| P_H(\phi, \phi_o) \right| \le 2 \tag{24}$$

for real a.

Similarly, for E-polarization the low frequency approximation to the integral equation is

$$\frac{\partial^2}{\partial \zeta^2} \frac{1}{\pi} \int_{-1}^{1} J_3(\zeta') \ln |\zeta' - \zeta| \, d\zeta' = 1 - b\, J_3(\zeta) \tag{25}$$

for $-1 < \zeta < 1$ with

$$b = -\frac{ikw}{2} \frac{Z}{\eta} \tag{26}$$

where the modified current $J_3(\zeta)$ is such that $J_3(\pm 1) = 0$. In terms of $J_3(\zeta)$

$$P_E(\phi, \phi_o) = -\frac{i\pi}{4} (kw)^2 \sin\phi \sin\phi_o K_E(b) \tag{27}$$

with

$$K_E(b) = \frac{1}{\pi} \int_{-1}^{1} J_3(\zeta)\, d\zeta \quad , \tag{28}$$

and the angle dependence is explicit in the expression for $P_E(\phi, \phi_o)$.

Computer programs were written to solve (20) and (25) by the moment method and, hence, compute $K_H(a)$ and $K_E(b)$ for all complex a and b. From an examination of the results it was found that $K_H(a)$ can be approximated as

$$K_H(a) = -\frac{(a+0.15)(a+0.29)}{\left(\frac{\pi a}{2} + \ln 2\right)(a+0.15)(a+0.29) + 0.10a\,(a+0.20)} \tag{29}$$

for all a apart from those in the immediate vicinity of the portion $-1.1 \le a \le 0.3$ of the real axis is the complex a plane. In this region an empirical expression for $K_H(a)$ is

$$K_H(a) = -\frac{1}{\frac{\pi a}{2} + \ln 2 + 0.1} \tag{30}$$

and since, for other a, (30) differs from (29) by no more than 3 percent, it is sufficient to use (30) for all a. Similarly, for E-polarization the approximation is

$$K_E(b) = \frac{0.62}{b+1.15}\,\frac{(b+4.08)(b+7.26)(b+10.37)(b+13.43)(b+16.46)}{(b+4.27)(b+7.37)(b+10.45)(b+13.49)(b+16.50)} \quad , \tag{31}$$

valid for all b not in the immediate vicinity of the negative real axis. For positive real b, $K_E(b) \le 1/2$ and hence

$$\left|P_E(\phi, \phi_o)\right| \le \frac{\pi}{8}(kw)^2 \sin\phi \sin\phi_o \quad . \tag{32}$$

In their regions of validity, the estimated accuracy of (29) - (31) is about three percent.

To use these results to predict the scattering from a narrow gap, it was proposed that η be identified with the impedance looking into the gap, with η calculated using a simple transmission line (or other) model that takes into account the geometry and material filling of the gap. To show how this is done, consider a crack such as those illustrated in Figure 2. For H-polarization the cavity supports a variety of TE modes, but since the width w is small, the only mode which is not evanescent is the TEM mode, and this is the main contributor to the field in the gap. Under the assumption that this is the only mode that must be considered, the effective surface impedance η can be deduced from the input impedance Z_{1n} of a parallel plate transmission line. The voltage across the gap is

$$V = \int_{-w/2}^{w/2} E_x(x)\, dx \approx wE_x$$

and since the current I is proportional to the tangential magnetic field,

$$\eta = \frac{E_x}{H_z} = \frac{1}{w}\frac{V}{I} = \frac{Z_{1n}}{w} \quad . \tag{33}$$

For a parallel plate transmission line whose plate separation is w, the inductance and capacitance per unit length and width are $L = \mu_0 \mu_r w$ and $C = \varepsilon_0 \varepsilon_r / w$ respectively, and the characteristic impedance is $Z_c = Z_1 w$ with $Z_1 = Z\sqrt{\mu_r/\varepsilon_r}$.

The L-shaped gap in Figure 2(b) can be viewed as two cascaded lines. The first line has length d_1 and characteristic impedance Z_c, whereas the second (of length

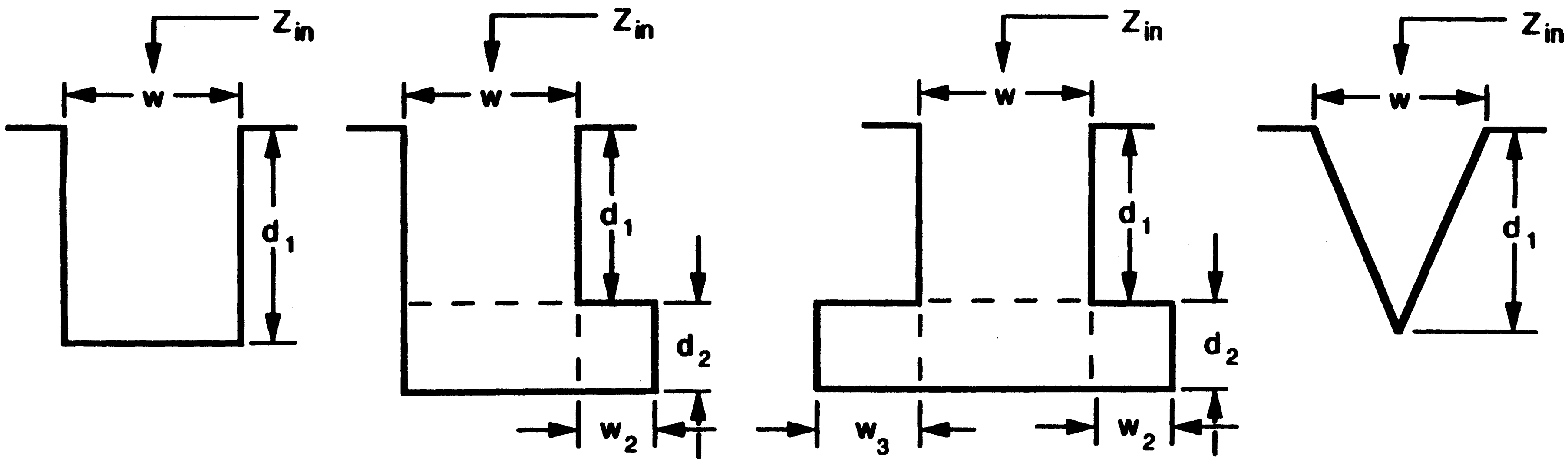

Fig. 2. Gap and cavity configurations. The cavity is filled with a homogeneous dielectric having relative permittivity ε_r and relative permeability μ_r.

w_2) has characteristic impedance $Z_c = Z_1 d_2$ and is shorted. As a load its impedance is

$$Z_L = -i\, Z_c^{'} \tan k_1 w_2 \quad . \tag{34}$$

The junction of these lines can be modelled as a lumped parameter pi-network whose reactance and susceptance elements are [3]

$$X = kZ_1 w\, d_2$$

$$B_1 = \frac{k}{Z_1}\left(\frac{d_2}{d_2 + w}\right)\left(1 - \frac{2}{\pi}\ln 2\right)$$

$$B_2 = \frac{k}{Z_1}\left(\frac{w}{d_2 + w}\right)\left(1 - \frac{2}{\pi}\ln 2\right) \quad .$$

The input impedance of the first line cascaded with the pi-network and the second line is then

$$Z_{1n} = Z_c \frac{Z_L^{'} - iZ_c \tan k_1 d_1}{Z_c - iZ_L^{'} \tan k_1 d_1} \tag{35}$$

where

$$Z_L^{'} = \frac{Z_L - iX\,(1 - iB_2 Z_L)}{(1 - B_1 X)\,(1 - iB_2 Z_L) - iB_1 Z_L} \quad . \tag{36}$$

Similarly, the T-shaped gap in Figure 2(c) can be treated as a transmission line loaded with two shorted lines in series. For the shorted lines of lengths w_2 and w_3, the load impedance is

$$Z_L'' = -i\, Z_c' \left\{ \tan k_1 w_2 + \tan k_1 w_3 \right\} \quad . \tag{37}$$

The junction is modelled with a shunt susceptance and a series reactance in series with Z_L:

$$X_3 = 2kZ_1\, wd_2$$

$$B_3 = \frac{k}{Z_1} \left(\frac{d_2}{d_2 + w} \right) 0.7822$$

where the constant was determined empirically, and the input impedance Z_{1n} is then given by (35) with

$$Z_L' = \frac{Z_L'' - i\, X_3}{1 - i\, B_3\, (Z_L'' - iX_3)} \quad . \tag{38}$$

The rectangular gap is the special case $d_2 = 0$ of either of the above structures, and for this

$$Z_{1n} = -i\, Z_c \tan k_1 d_1 \quad . \tag{39}$$

Finally, for the V-shaped gap in Figure 2 (d), the inductance and capacitance per unit length of the line are functions of position, but when the coupled differential equations for the voltage and current are solved, we obtain

$$Z_{1n} = -i\, Z_c \frac{J_1\,(k_1 d_1)}{J_0\,(k_1 d_1)} \quad . \tag{40}$$

where J_0 and J_1 are Bessel functions.

For E-polarization all of the modes are evanescent, but if we again assume that the first mode dominates in the gap, simple formulas for the surface impedance can be found. In a parallel plate waveguide of width w

$$H_x = \frac{1}{ik Z \mu_r} \frac{\partial E_z}{\partial y} ,$$

and for the lowest order mode the propagation constant is ikp where

$$p = \left\{ \left(\frac{\lambda}{2w} \right)^2 - \epsilon_r \mu_r \right\}^{1/2} . \tag{41}$$

Since E_z/H_x is independent of position, a transmission line analogy can be made. The characteristic impedance of the line is $-iZ\mu_r/p$, which is also the impedance looking into the gap, and the results previously obtained for H-polarization are now applicable if k_1 is replaced by ikp and Z_1 by $-iZ\mu_r/p$. Thus, for a rectangular gap

$$Z_{1n} = -i \frac{Z\mu_r w}{p} \tanh kpd_1 , \tag{42}$$

and for a triangular gap

$$Z_{1n} = -i \frac{Z\mu_r w}{p} \frac{I_1(kpd_1)}{I_0(kpd_1)} , \tag{43}$$

where Z_{1n} and η are related via (33) and I_0 and I_1 are modified Bessel functions [4]. Formulas for L- and T-shaped cracks can be deduced in a similar manner, but since the modes are evanescent, the shape of the lower cavity has little or no effect on the impedance.

4. Numerical Results

The integral equation pairs (7), (11) and (17), (18) for H- and E-polarizations respectively were programmed for solution by the moment method, using pulse basis and point matching functions as described in Appendix A. In the case of (17), the derivative was applied to the kernel, and because of the order of the resulting singularity, the contributions from two cells on either side of the self cell were evaluated analytically, in addition to the contribution of the self cell itself. Comparison with the results of a finite element method [5] for H-polarization showed excellent agreement, and for purposes of comparison with the quasi-analytical solution, the moment method data will be regarded as exact. The computer program used to implement the expressions for the quasi-analytical solution is listed in Appendix B.

Considering first the results for H-polarization, Figure 3 shows the backscattering from a rectangular air-filled gap as a function of aspect for three gap widths. The aspect variation decreases with w. It is less than 4 percent for $w/\lambda = 0.15$, and since aspect independence is a feature of the quasi-analytic solution, we will henceforth confine attention to this case. It is then sufficient to take $\phi = \phi_0 = \pi/2$ corresponding to normal incidence backscatter.

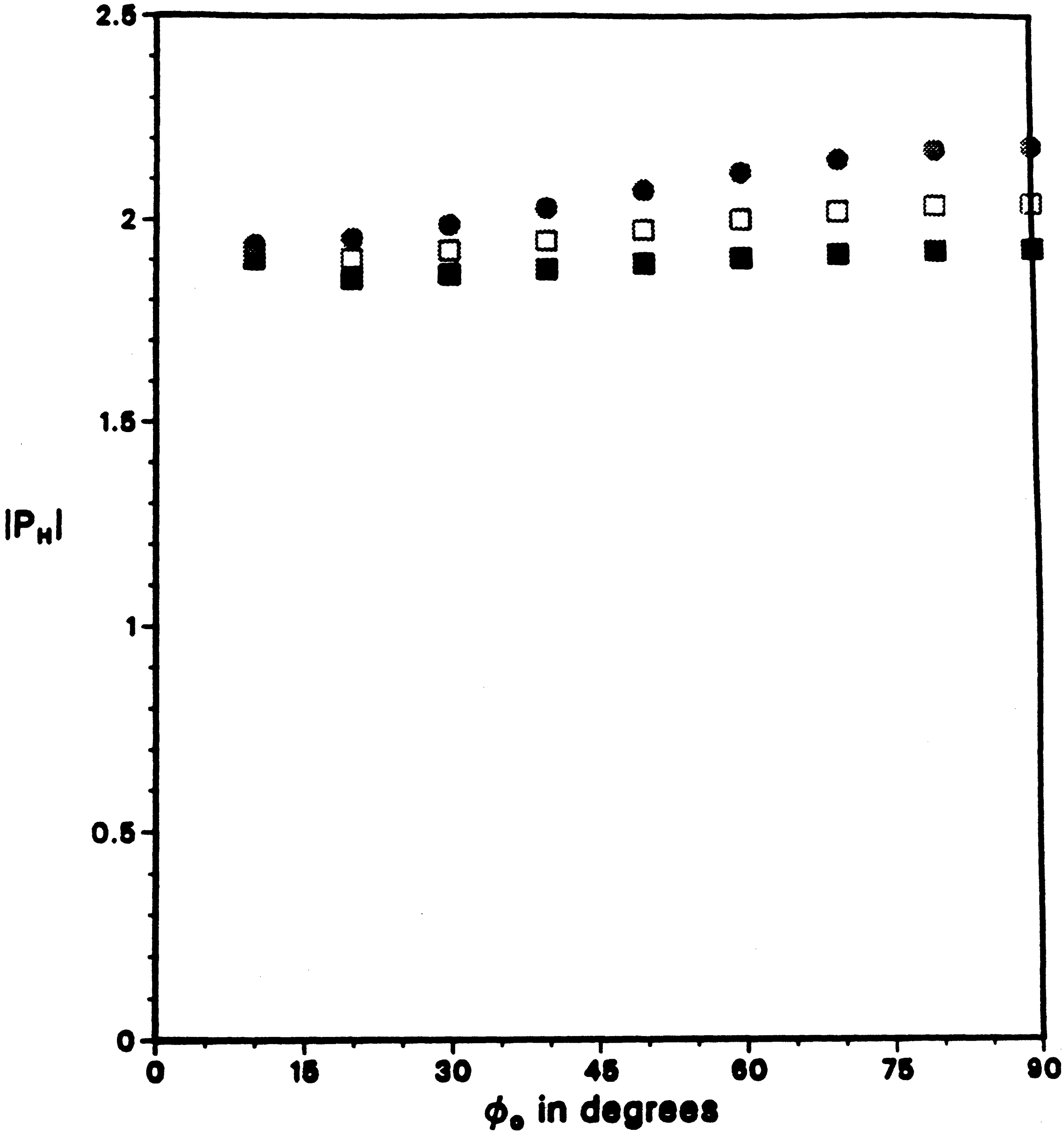

Fig. 3: Modulus of the far field amplitude P_H with respect to aspect ϕ_0 for a rectangular gap with $\phi = \pi/2$ and $d/\lambda = 0.2$: ■ $w/\lambda = 0.15$, □ $w/\lambda = 0.2$, ● $w/\lambda = 0.25$.

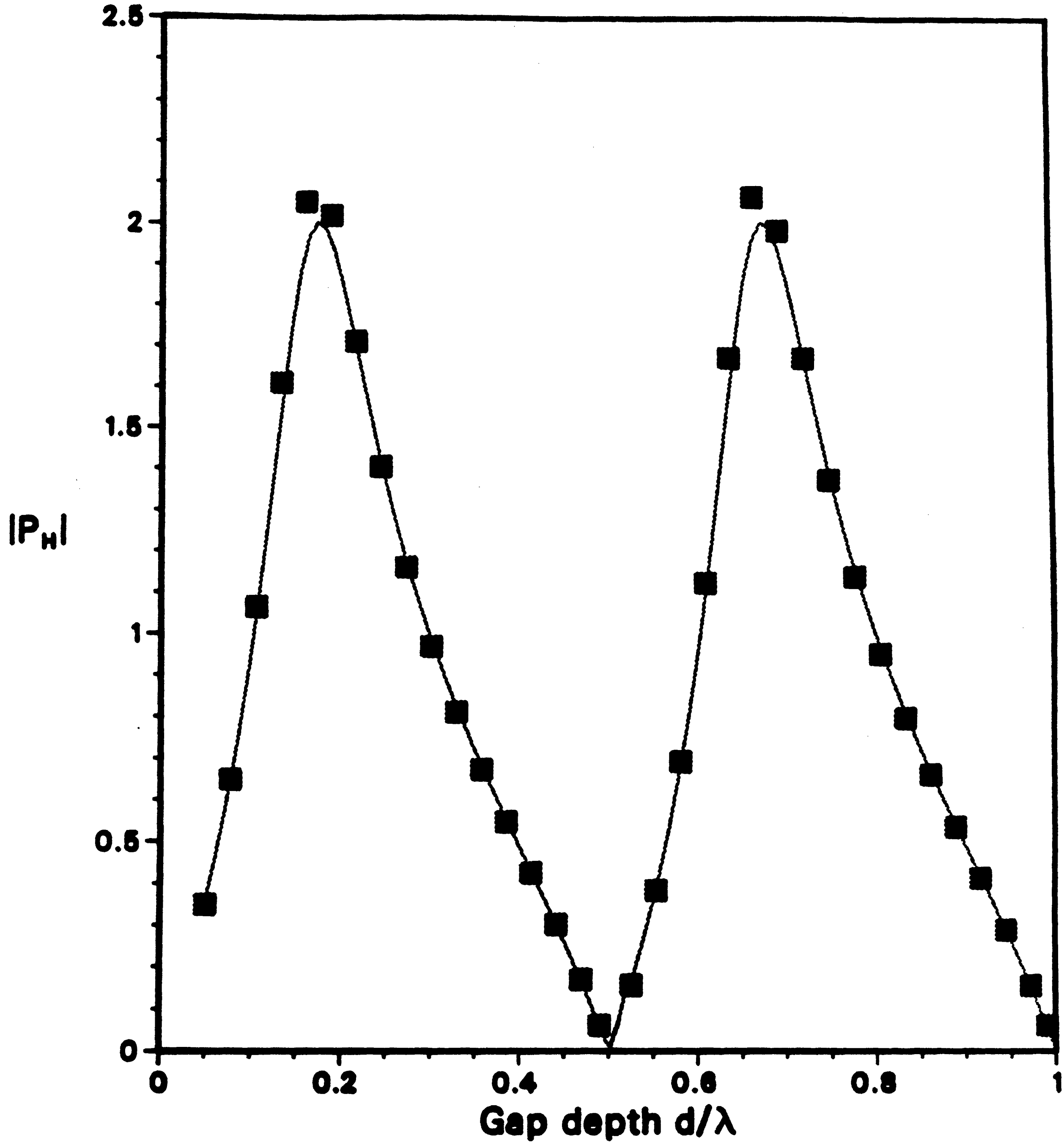

Fig. 4. Modulus of the far field amplitude P_H for a rectangular gap of varying d($d_1 = d$ with $\phi = \phi_0 = \pi/2$ and $w/\lambda = 0.15$: ■ exact, ——— analytical.

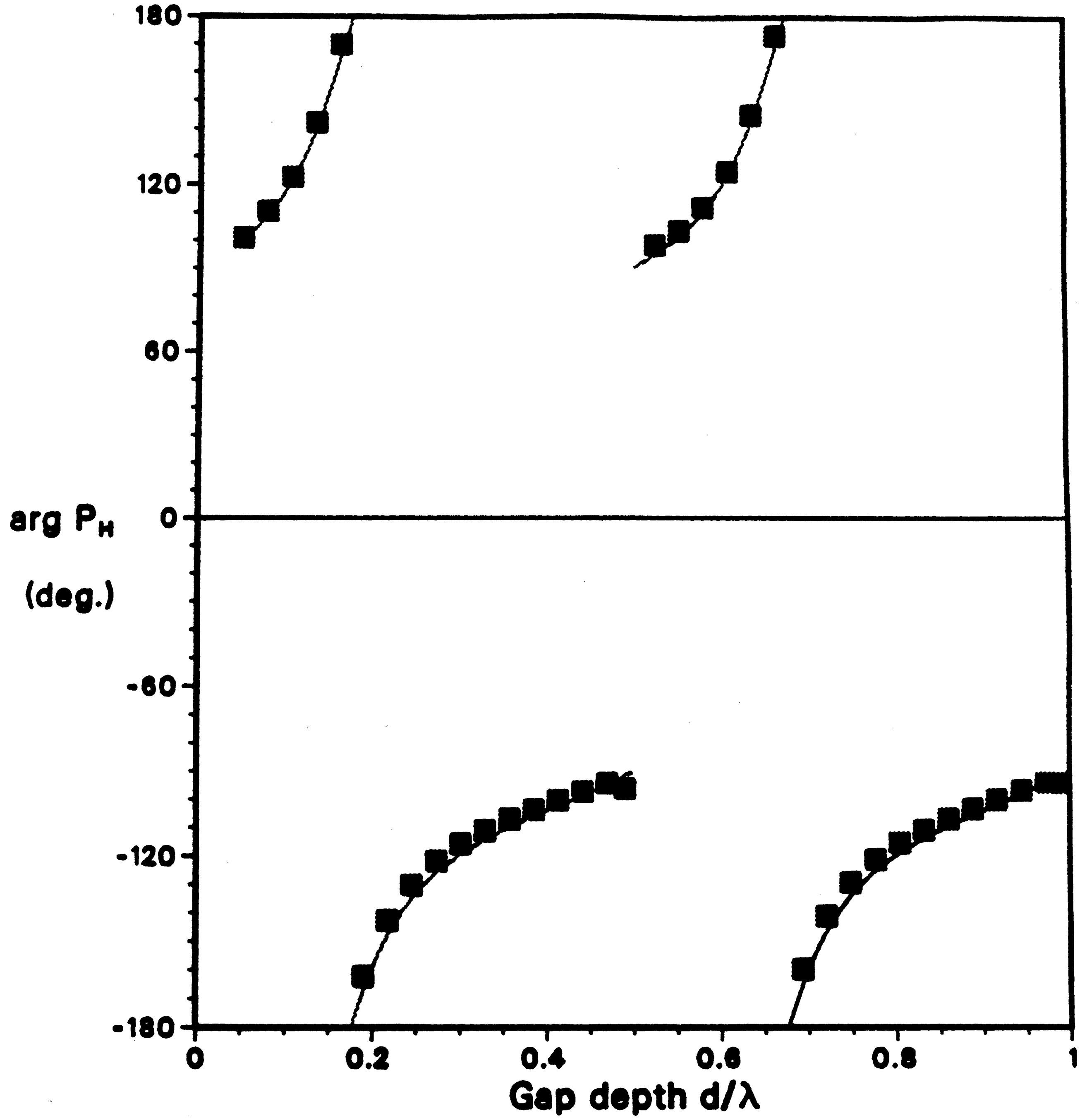

Fig. 5. Argument of the far field amplitude P_H for a rectangular gap of varying depth $d_1 = d$ with $\phi = \phi_0 = \pi/2$ and $w/\lambda = 0.15$: ■ exact, ——— analytical.

In Figures 4 and 5 the amplitude and phase of the far field amplitude $P_H(\pi/2, \pi/2)$ are shown as a function of depth for a rectangular air-filled gap of width $w/\lambda = 0.15$. We observe the cyclical behavior with zeros at $d/\lambda = 0, 0.5, 1.0, \ldots$, resulting from the periodicity of the impedance looking into the gap. From (39) and (21) the corresponding a are real and vary from $-\infty$ to ∞ over each cycle. Over the entire range of d/λ the agreement between the quasi-analytic and moment method results is excellent, but in spite of this the computed aperture impedances do not agree. This is evident from Figure 6 where $|E_x/H_z|$ is plotted as a function of x for $w/\lambda = 0.15$ and $d/\lambda = 0.20$. The U-shaped behavior is in accordance with the edge condition at $x = -w/2$, and the data fit the curve $C\left\{1-\left(\frac{2x}{w}\right)^2\right\}^{1/2}$ with C = 860 ohms. The average value is therefore $\pi C/2 = 1350$ ohms, compared with which (36) gives $|\eta| = 1160$ ohms. A similar discrepancy was found with all gap geometries. Nevertheless, the quasi-analytic solution provides an excellent approximation to the far field, and this is illustrated in Figures 7 through 10 showing $|P_H|$ for a material-filled rectangular gap and for air-filled L-, T- and V-shaped gaps.

Turning now to E-polarization, Figures 11 and 12 show the amplitude and phase of $P_E(\pi/2, \pi/2)$ as functions of w/λ for a rectangular air-filled gap having $d/\lambda = 0.1$. The quasi-analytic and exact data diverge with increasing w/λ, but the difference is less than 4 percent in amplitude and 5 degrees in phase for $w/\lambda \leq 0.20$.

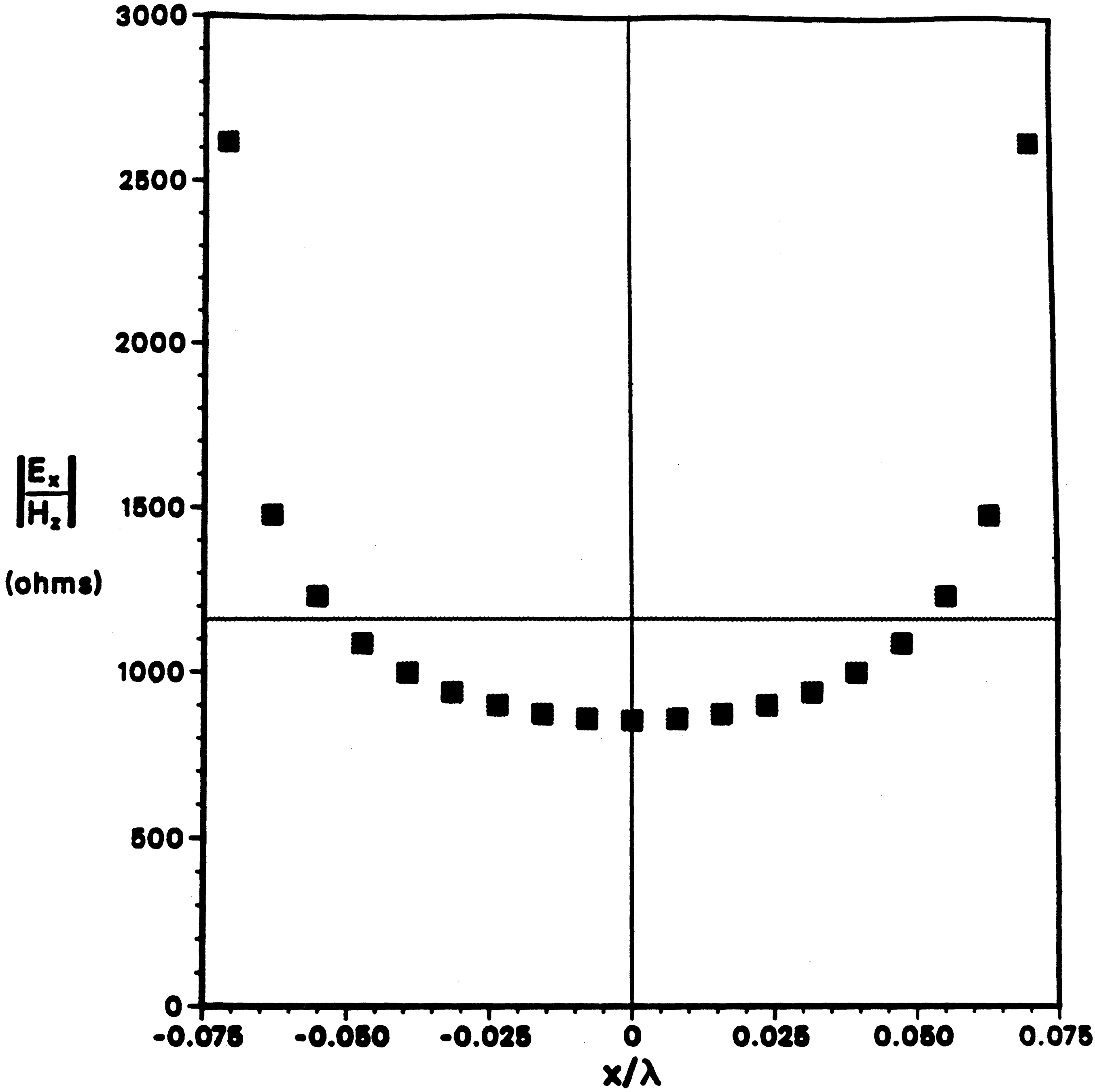

Fig. 6. Aperture impedance $|E_x/H_z|$ evaluated at $-w/2 < x < w/2$ and $y=0$ for a rectangular gap with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, and $d/\lambda = 0.2$: ■ exact, ——— analytical.

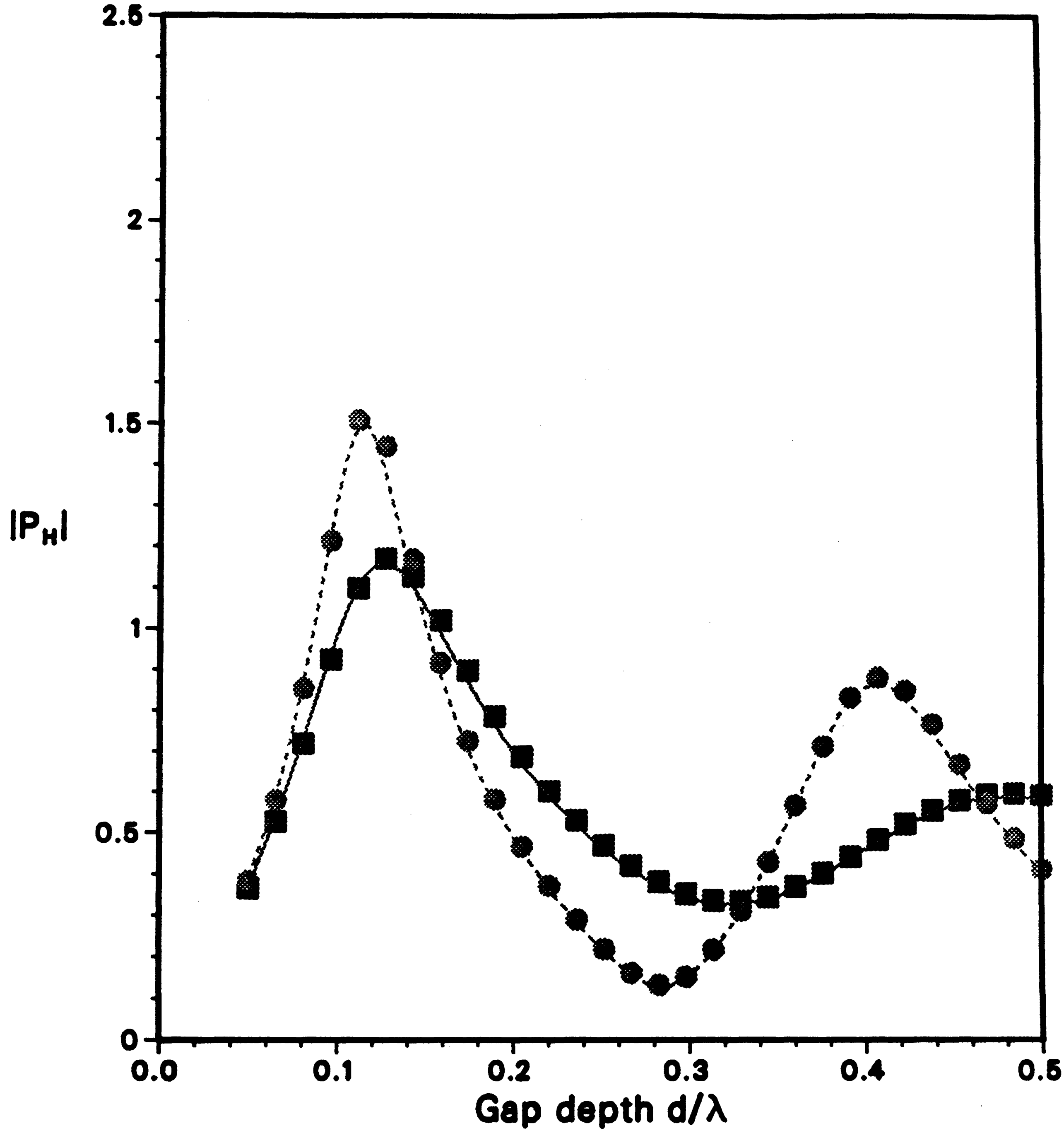

Fig. 7. Modulus of the far field amplitude P_H for a material-filled rectangular gap of varying depth $d_1 = d$ with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, and $\mu_r = 1$:

$\varepsilon_r = 2 + i1$ ■ exact, ——— analytical

$\varepsilon_r = 3 + i0.5$ ● exact, - - - - - analytical.

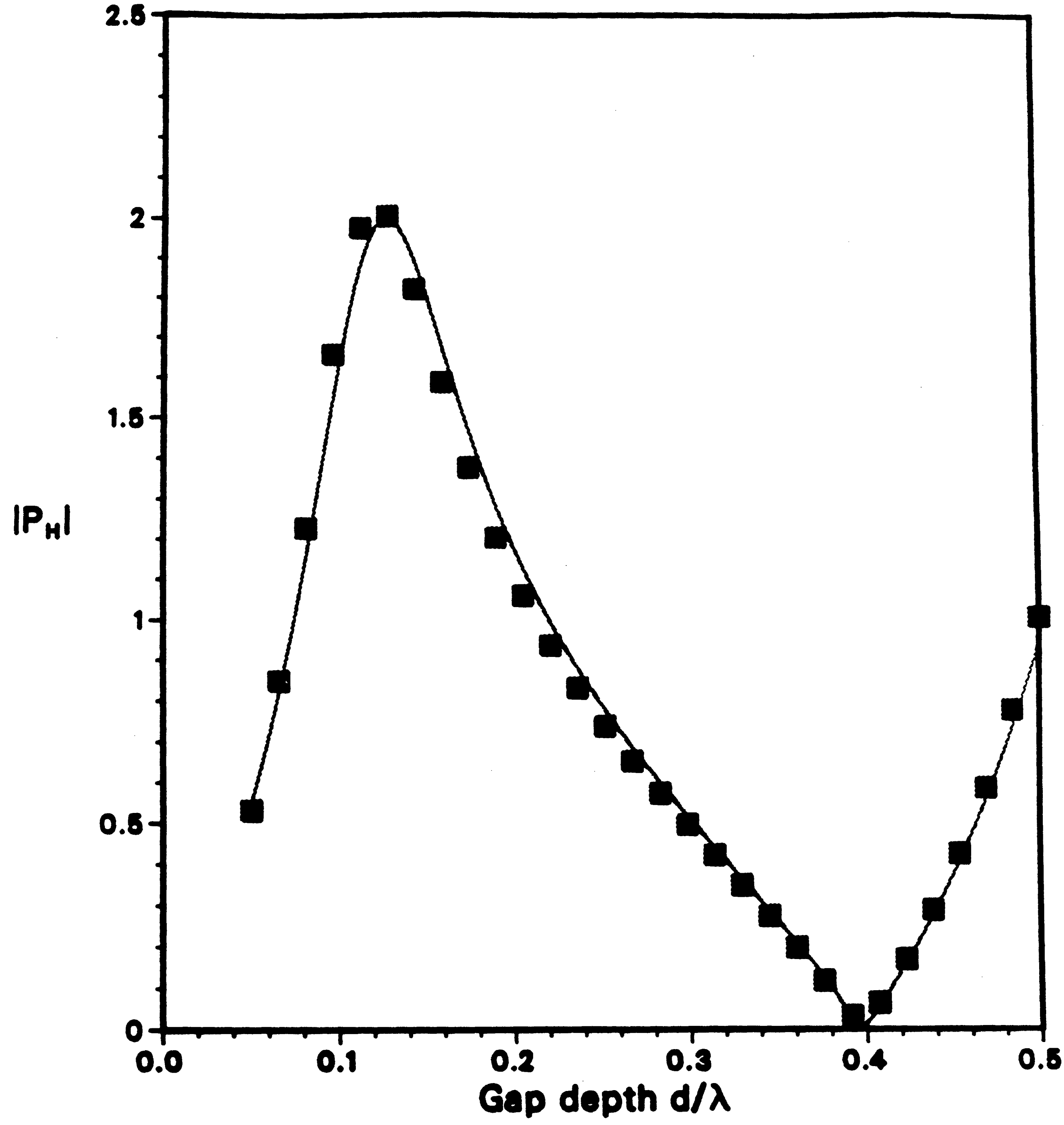

Fig. 8(a). Modulus of the far field amplitude P_H for an air-filled L-shaped gap of varying depth $d_1 + d_2 = d$ with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, $w_2/\lambda = 0.15$, and $d_1/d_2 = 3$: ■ exact, ——— analytical.

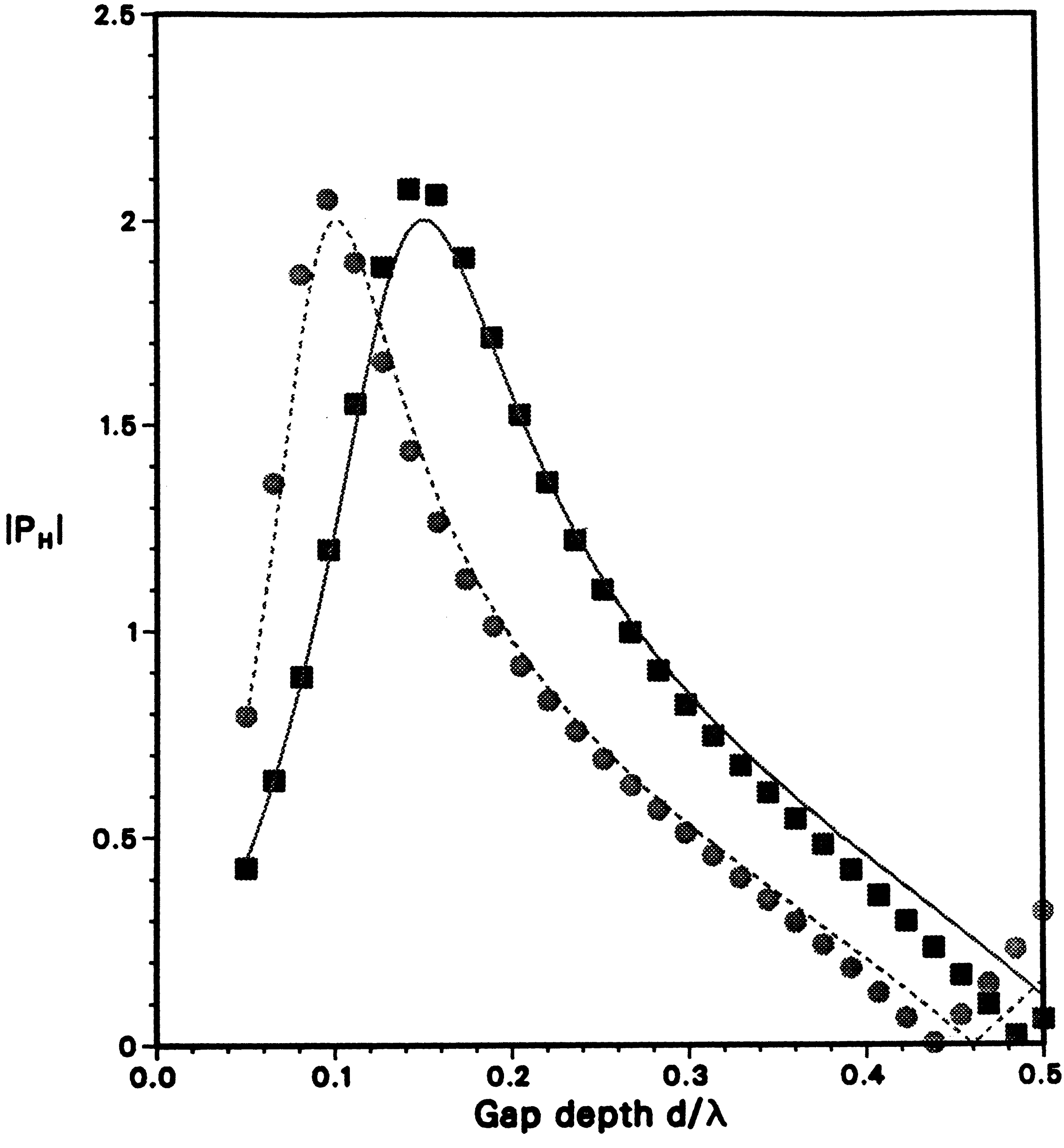

Fig. 8(b). Modulus of the far field amplitude, P_H, for an air-filled L-shaped gap of varying depth $d_1 + d_2 = d$ with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, and $d_1/d_2 = 1$:

$w_2/\lambda = 0.05$ ■ exact, ——— analytical

$w_2/\lambda = 0.15$ ● exact, - - - - analytical.

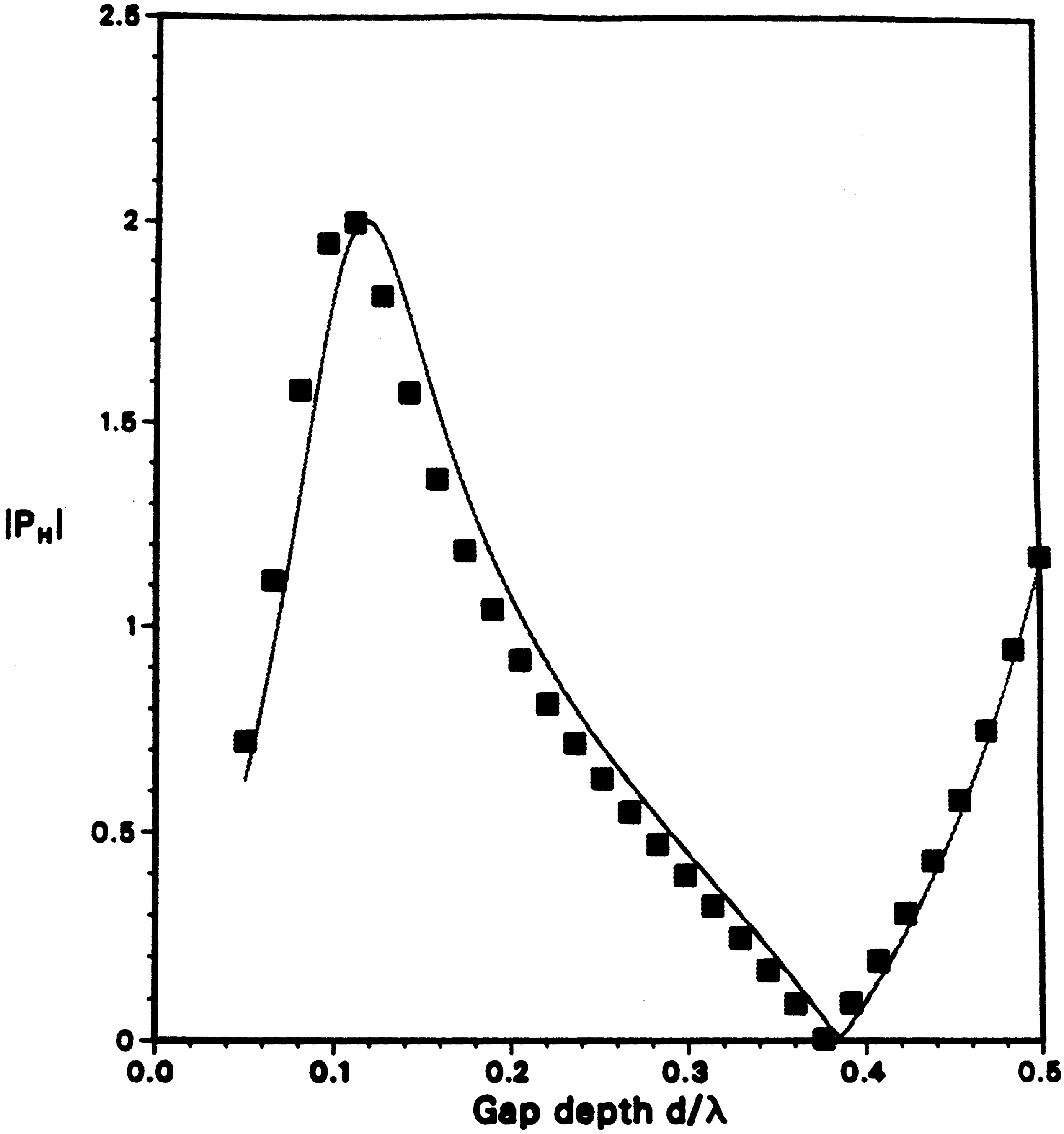

Fig. 9(a). Modulus of the far field amplitude P_H for an air-filled T-shaped gap of varying depth $d_1 + d_2 = d$ with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, $w_2/\lambda = w_3/\lambda = 0.075$, and $d_1/d_2 = 3$: ■ exact, ——— analytical.

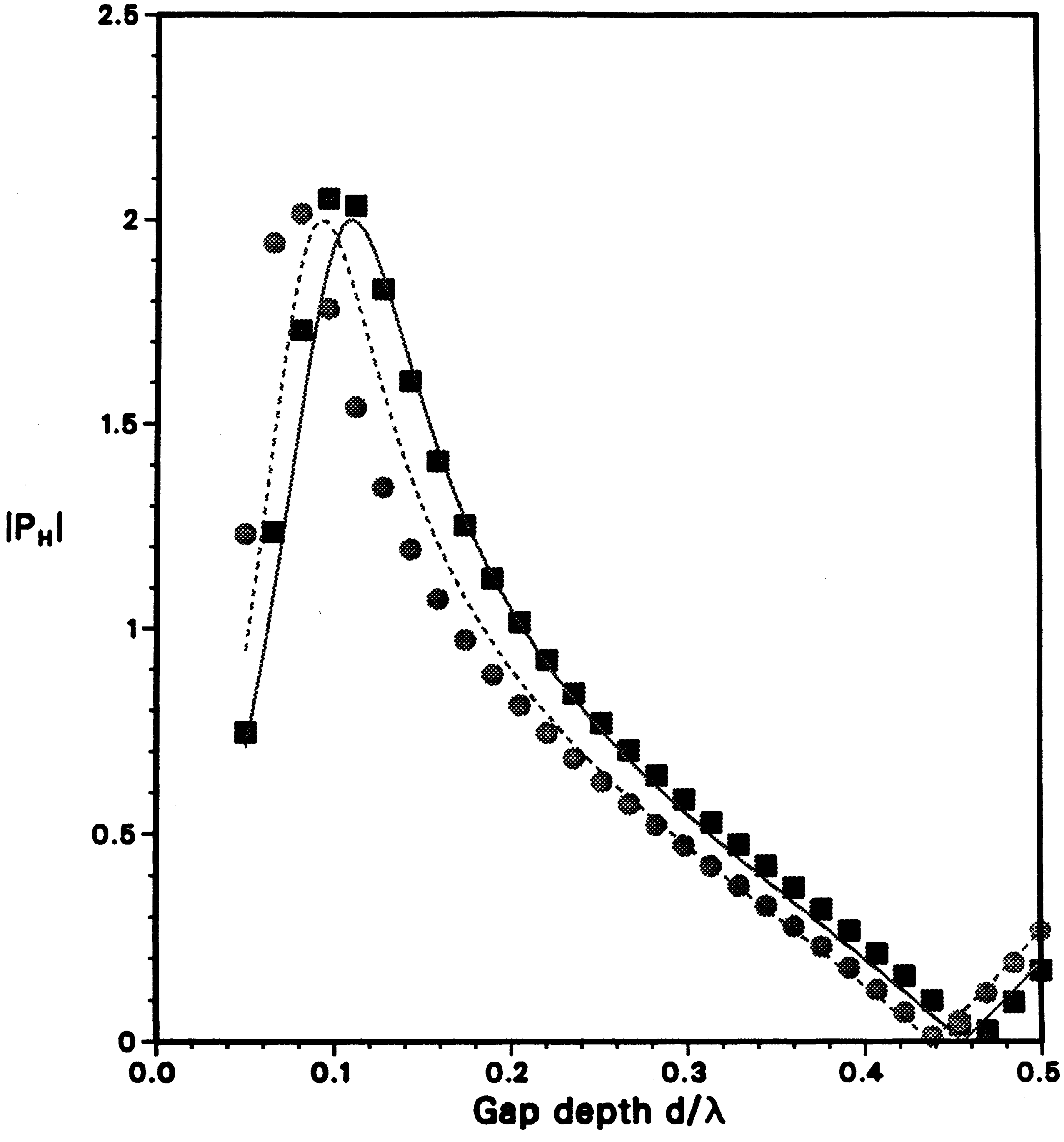

Fig. 9(b). Modulus of the far field amplitude, P_H, for an air-filled T-shaped gap of varying depth $d_1 + d_2 = d$ with $\phi = \phi_0 = \pi/2$, $w/\lambda = 0.15$, and $d_1/d_2 = 1$:

$w_2/\lambda = w_3/\lambda = 0.025$ ■ exact, —— analytical

$w_2/\lambda = w_3/\lambda = 0.075$ ● exact, ---- analytical.

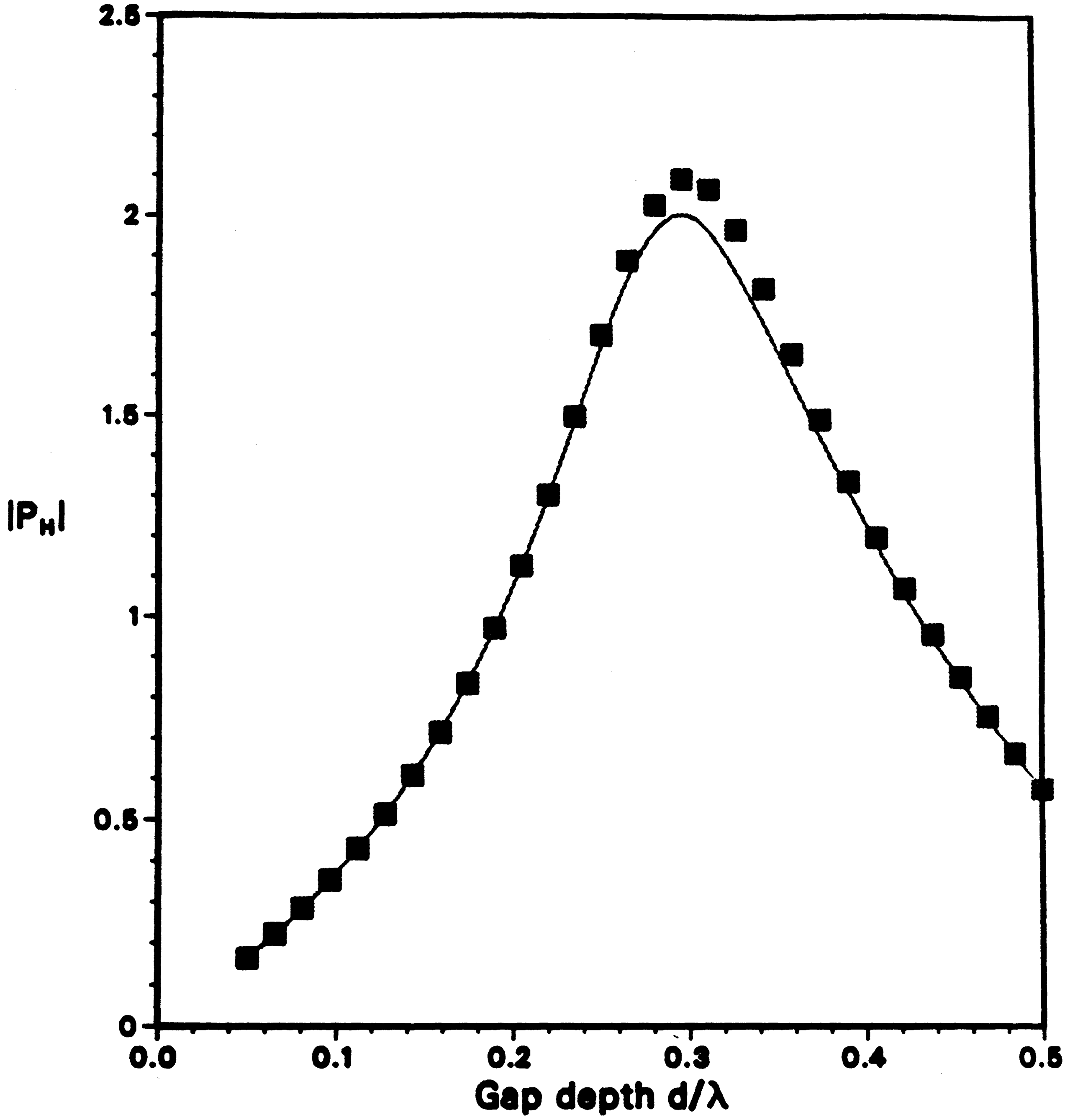

Fig. 10. Modulus of the far field amplitude P_H for an air-filled V-shaped gap of varying depth $d_1 = d$ with $\phi = \phi_0 = \pi/2$ and $w/\lambda = 0.15$:
■ exact, ——— analytical.

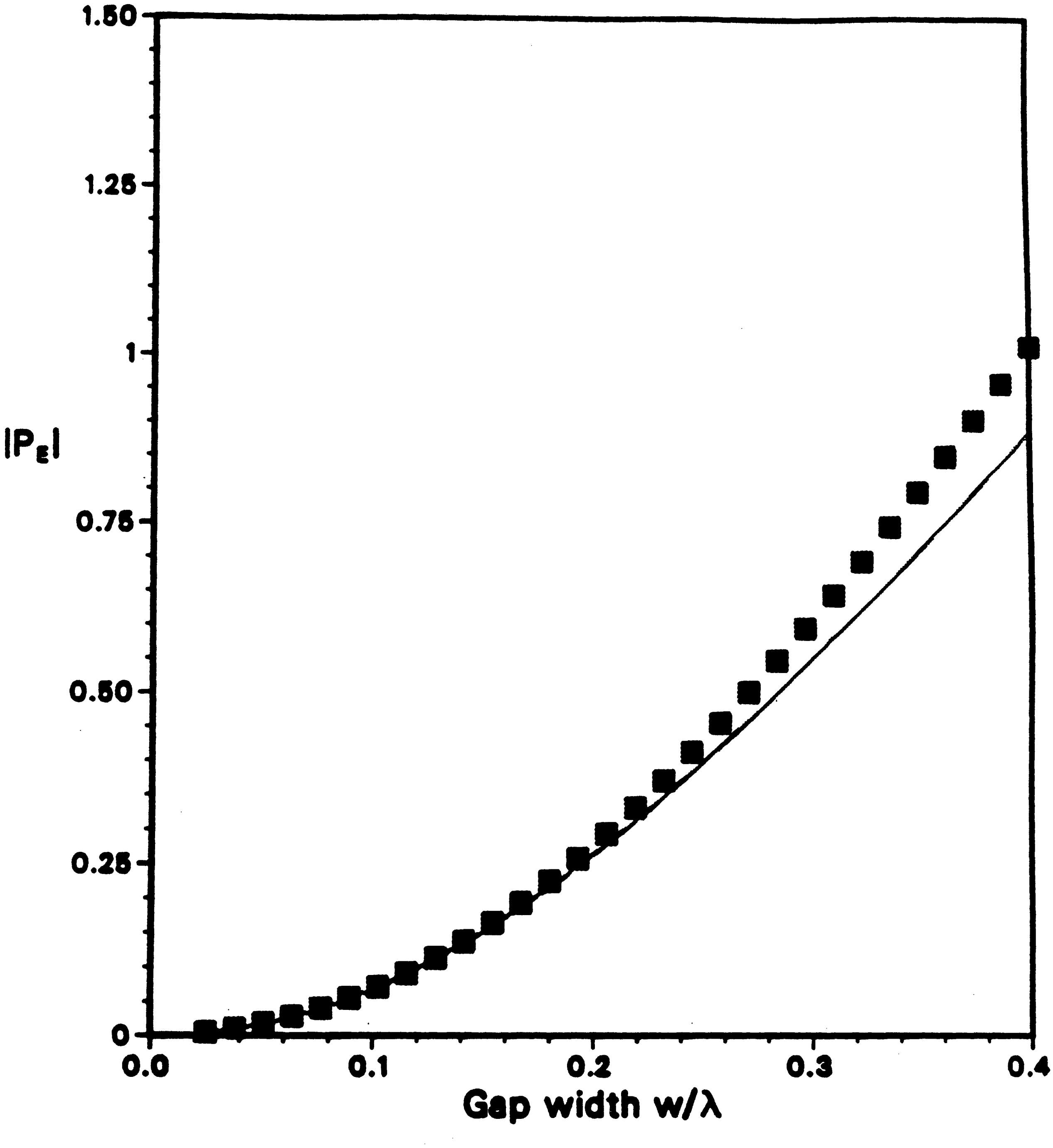

Fig. 11. **Modulus of the far field amplitude P_E for an air-filled rectangular gap o varying width with $\phi = \phi_0 = \pi/2$ and $d/\lambda = 0.1$: ■ exact, ——— analyti**

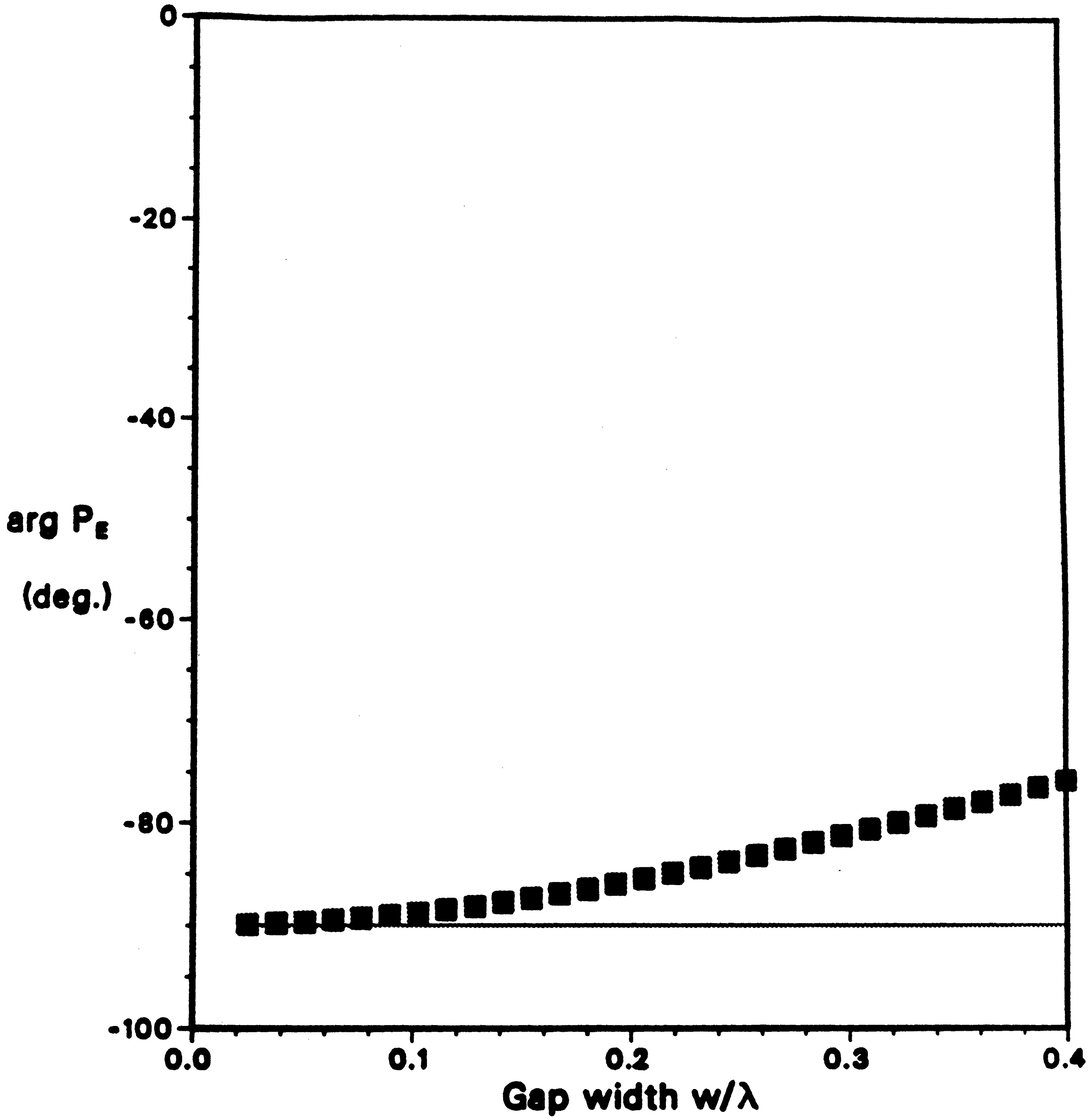

Fig. 12. Argument of the far field amplitude P_E for an air-filled rectangular gap of varying width with $\phi = \phi_0 = \pi/2$ and $d/\lambda = 0.1$: ■ exact, ——— analytical.

For a rectangular gap with $w/\lambda = 0.15$, the quasi-analytic and exact results for $\left| P_E\left(\frac{\pi}{2}, \frac{\pi}{2}\right) \right|$ as a function of d/λ are presented in Figure 13. The agreement is excellent, and as a consequence of the mode attenuation, the scattering is independent of the depth for $d/\lambda \geq 0.15$. A similar comparison for a triangular gap is given in Figure 14.

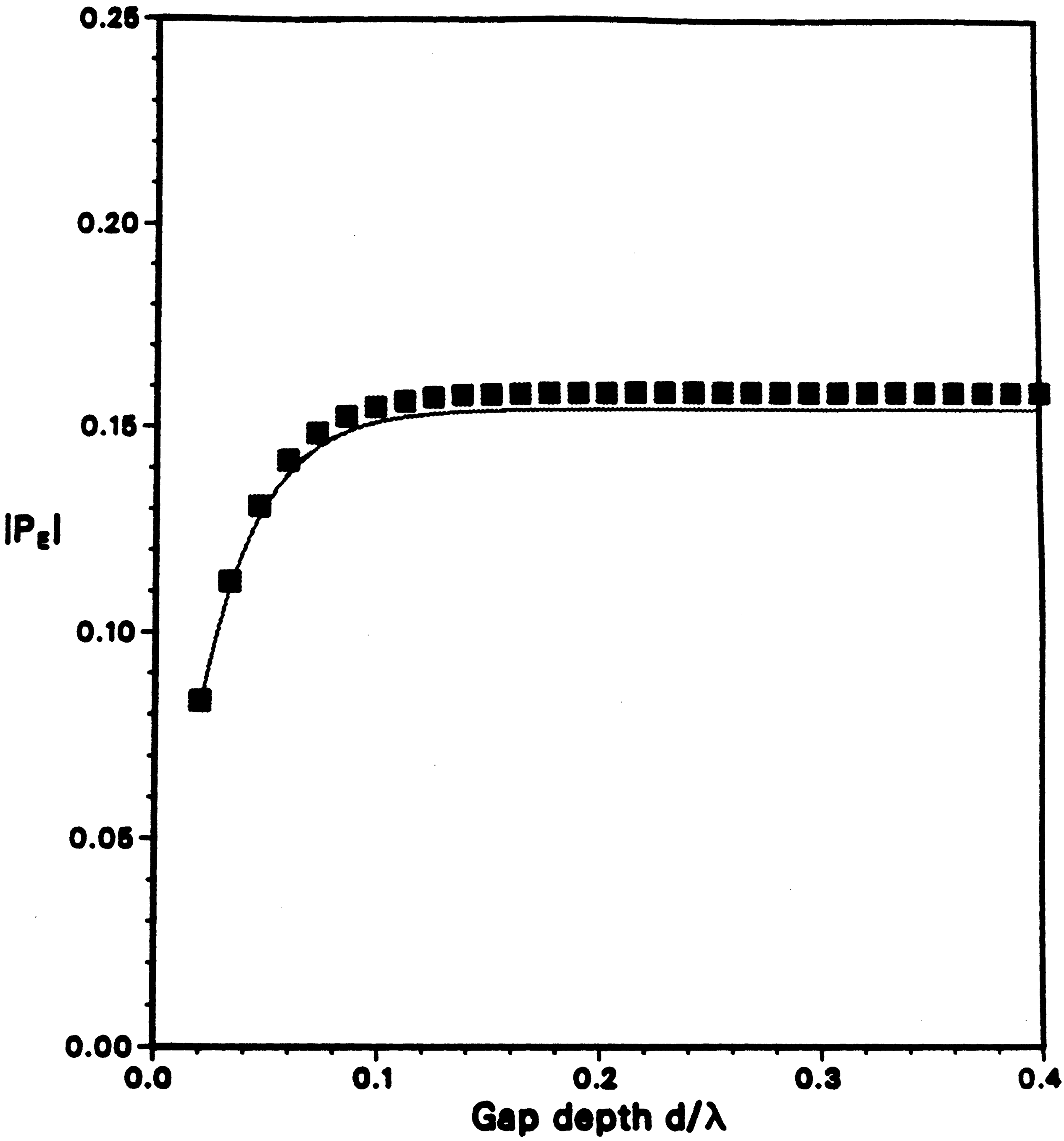

Fig. 13. Modulus of the far field amplitude P_E for an air-filled rectangular gap of varying depth $d_1 = d$ with $\phi = \phi_0 = \pi/2$ and $w/\lambda = 0.15$:
■ exact, ——— analytical.

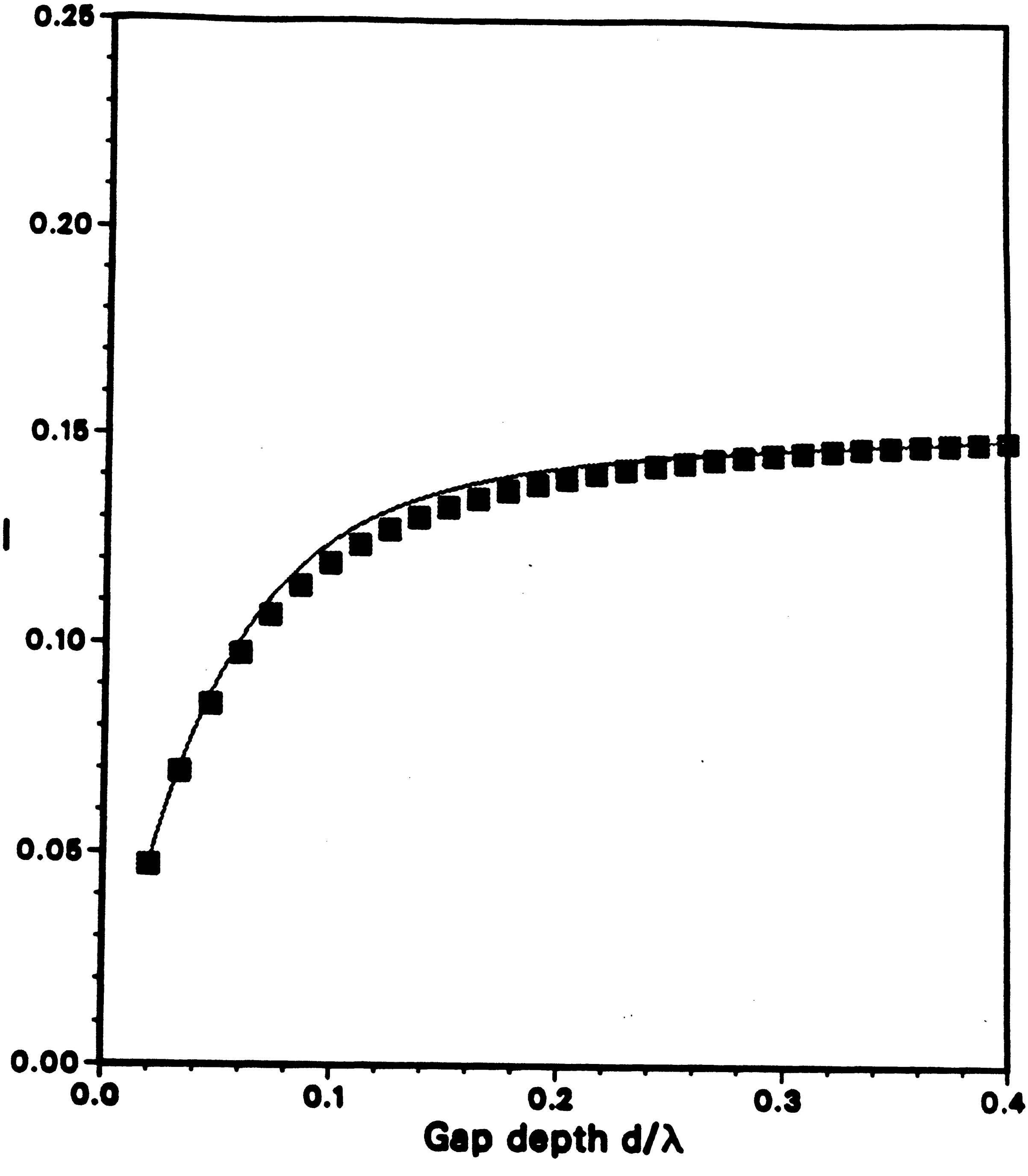

Fig. 14. Modulus of the far field amplitude P_E for an air-filled V-shaped gap of varying depth $d_1 = d$ with $\phi = \phi_0 = \pi/2$ and $w/\lambda = 0.15$: ■ exact, ——— analytical.

5. Conclusions

The quasi-analytic method described in [1] is based on the low frequency solution of the integral equations for a constant impedance insert in a perfectly conducting plane, and when used in conjunction with an estimate of the impedance looking into a gap, it provides a simple approximation to the far field scattering from the gap. To determine its accuracy, we have analyzed the problem of a plane wave incident on a gap backed by a cavity of arbitrary shape. The equivalence principle was used to develop coupled integral equations for the induced electric and magnetic currents, and the equations were then solved by the moment method. When the impedance looking into the cavity was determined using a transmission line model, it was found that for gap widths $w/\lambda \leq 0.15$ the quasi-analytic and moment method results for the scattered field were in excellent agreement for both polarizations and for all gap configurations that were tested. It therefore appears that the quasi-analytic method is an efficient and effective tool for predicting the scattering from the junction where two component parts of a target come together.

References

[1] T.B.A. Senior and J.L. Volakis, "Scattering by Gaps and Cracks," IEEE Trans. Antennas Propagat., vol. AP-37, 1989.

[2] R.F. Harrington, Time Harmonic Electromagnetic Fields, McGraw Hill Book Co., New York, 1961; pp. 106 et seq.

[3] S. Ramo, J.R. Whinnery and T. Van Duzer, "Fields and Waves in Communication Electronics," John Wiley & Sons, New York, 1984, p. 569.

[4] G.N. Watson, "A Treatise on the Theory of Bessel Functions," Cambridge Univ. Press, Cambridge, 1948, p. 77.

[5] S.K. Jeng, "Aperture Admittance Matrix by Finite Element Method for Scattering by a Cavity-Backed Aperture," IEEE AP-S International Symposium, Syracuse, NY, June 1988.

Appendix A Moment Method Solution of the Coupled Integral Equations

The integral equation pairs given by (7), (11) and (17), (18) are solved by the moment method. Using pulse basis functions in the moment method, the aperture A and the cavity walls S of Figure 1 are segmented into N cells of size Δs. The magnetic and electric currents are assumed to be constant over each of these segments. When the integrations of the coupled equations are taken over each segment, the current expressions can be removed as constants from the integrals. With the contour of integration discretized, the (x',y') coordinates become (x_i,y_i), i = 1,...,N, which describe the location of each of the segments. The Hankel functions can then be expressed in terms of rotated coordinates (s,n) for the observation position and (s_i,n_i) for each segment or source position since the integration is with respect to the tangential vector $\hat{s}$ as shown in Figure 1.

The expressions for the numerical solution of the coupled equations are developed in the following sections for the H- and E-polarization cases. Applying point matching, the magnetic and electric currents in the aperture and on the cavity walls are determined, and the far field amplitude is calculated.

A.1 H-Polarization

For the discretized contour of integration, (7) and (11) become

$$J_s(s,n) = \frac{kY}{2}\varepsilon_r \sum_{i=1}^{M} J_z^*(s_i) \int_{\Delta s_i} H_o^{(1)}\left(k_1\sqrt{(s-s_i)^2+n^2}\right) ds_i$$

$$+\frac{ik_1}{2}\sum_{i=1}^{N} J_s(s_i,n_i)\int_{\Delta s_i} \sin\gamma_i \; H_1^{(1)}\left(k_1\sqrt{(s-s_i)^2+(n-n_i)^2}\right) ds_i \qquad (A.1)$$

$$J_s(s) = 2e^{-iks\cos\phi_o} - \frac{kY}{2}\sum_{i=1}^{M} J_z(s_i)\int_{\Delta s_i} H_o^{(1)}\left(k\,|s-s_i|\right) ds_i \qquad (A.2)$$

where M are the number of segments across the aperture and

$$\sin\gamma_i = \frac{(n-n_i)}{\sqrt{(s-s_i)^2 + (n-n_i)^2}} \quad . \qquad (A.3)$$

Applying point matching over the N segments of the aperture and cavity walls,

$$\sum_{i=1}^{N} I_i\left[\,1\Big|_{j=i} + \frac{ik_1}{2}\int_{\Delta s_i}\sin\gamma_{j,i}\, H_1^{(1)}(\,k_1\, R_{j,i}\,)\, ds_i\right]$$

$$+\frac{kY}{2}\varepsilon_r\sum_{i=N+1}^{N+M} I_i\int_{\Delta s_i} H_o^{(1)}(\,k_1\, R_{j,i}\,)\, ds_i = 0 \qquad j = 1,\ldots,N \qquad (A.4)$$

$$\sum_{i=N+1}^{N+M} I_i\left[\,1\Big|_{j=i} + \frac{kY}{2}\int_{\Delta s_i} H_o^{(1)}(k\, R_{j,i})\, ds_i\right] = 2e^{-iks_j\cos\phi_o} \qquad j = N+1,\ldots,N+M \quad (A.5)$$

where

$$R_{j,i} = \sqrt{(s_j-s_i)^2 + (n_j-n_i)^2} \quad . \qquad (A.6)$$

The coordinate (s_j,n_j) is the observation position at the midpoint of the j^{th} segment. Hence, for i,j = 1,...,M,N+1,...,N+M, the segments are located in the aperture, and for i,j = M+1,...,N, the segments are located on the cavity walls. I_i

in (A.4) and (A.5) are the electric currents, for i = 1,...,N, and the magnetic currents, i = N+1,...,N+M, to be determined.

In matrix form, (A.4) and (A.5) become

$$[\, Z_{j,i} \,]\,[\, I_i \,] = [\, V_j \,] \tag{A.7}$$

The impedance matrix is given as

$$[\, Z_{j,i} \,] = \left[\begin{array}{c:c} Z_{e1} & Z_{m1} \\ \hdashline Z_{e2} & Z_{m2} \end{array}\right] \tag{A.8}$$

where the sets of elements are as follows:

$$Z_{e1} = \begin{cases} \dfrac{ik_1}{2} \displaystyle\int_{\Delta s_i} \sin \gamma_{j,i}\, H_1^{(1)}(k_1 R_{j,i})\, ds_i & j \neq i \\ -1 & j = i \end{cases} \tag{A.9}$$

for i = 1,...,N and j = 1,...,N;

$$Z_{m1} = \begin{cases} \dfrac{kY}{2} \varepsilon_r \displaystyle\int_{\Delta s_i} H_o^{(1)}(k_1 R_{j,i})\, ds_i & j \neq i\text{-}N \\ \dfrac{kY}{2} \varepsilon_r \left\{ \dfrac{i2}{\pi} \left[2(s_i - s_j) \ln (R_{j,i}) - \left(2 - A'(k_1)\right) (s_i - s_j) \right] \right\} & j = i\text{-}N \end{cases} \tag{A.10}$$

for i = N+1,...,N+M and j = 1,...,N;

$$Z_{e2} = \begin{cases} 0 & j \neq i+N \\ 1 & j = i+N \end{cases} \qquad \text{(A.11)}$$

for i = 1,...,N and j = N+1,...,N+M;

$$Z_{m2} = \begin{cases} \frac{kY}{2} \int_{\Delta s_i} H_o^{(1)}(k R_{j,i})\, ds_i & j \neq i \\ \frac{kY}{2} \left\{ \frac{i2}{\pi} \left[2(s_i-s_j)\, \ln (R_{j,i}) - \left(2-A'(k)\right)(s_i-s_j) \right] \right\} & j = i \end{cases} \qquad \text{(A.12)}$$

for i = N+1,...,N+M and j = N+1,...,N+M. In (A.10), the expression for A' is

$$A'(k_1) = 2\left(\ln \frac{k_1}{2} + \gamma - i\frac{\pi}{2} \right) \quad .$$

In (A.12), A' is a function of k. For the self-cells in (A.10) and (A.12), s_i is taken to be the endpoint of the i^{th} segment. The self-cell expressions were derived analytically, and a numerical integration is applied to the other segments.

In the case of the V-shaped gap, the adjacent cells needed to be evaluated in the vicinity of y = -d, for $R_{j,i}$ less than one cell size. The analytical expressions for the impedance elements of the adjacent cells are

$$Z_{e1} = \frac{i}{2} \left[-\frac{k_1^2}{2}(n_j-n_i)\, s_i + \frac{i2}{\pi} \frac{(n_j-n_i)}{|n_j-n_i|} \, \text{atan}\left(\frac{s_i-s_j}{|n_j-n_i|} \right) \right] \qquad i = j \pm 1 \qquad \text{(A.13)}$$

$$Z_{m1} = \frac{kY}{2} \varepsilon_r \frac{i}{\pi} \left[2(s_i - s_j) \ln (R_{j,i}) - \left(2 - A'(k_1)\right) s_i + 2|n_j - n_i| \operatorname{atan}\left(\frac{s_i - s_j}{|n_j - n_i|} \right) \right]$$

$$i - N = j \pm 1 \qquad \text{(A.14)}$$

where s_i is evaluated at the endpoints of the i^{th} segment. The adjacent cell expression for Z_{m2} is given by (A.14) with $A'(k_1)$ replaced with $A'(k)$ for $i = j \pm 1$.

The source matrix is given by

$$V_j = \begin{cases} 0 & j = 1,\ldots,N \\ 2e^{-iks_j \cos\phi_o} & j = N+1,\ldots,N+M \end{cases} \quad . \qquad \text{(A.15)}$$

The currents I_i are determined by solving (A.7), given that $[Z_{j,i}]$ is nonsingular.

The aperture impedance is defined in terms of the total fields as

$$\eta_j = \frac{E_x(x_j,0)}{H_z(x_j,0)}$$

where the total electric field is equal to the magnetic current in the aperture, I_j for $j = N+1,\ldots,N+M$, and from (4), the total magnetic field is now expressed as

$$H_z(x_j,0) = 2e^{-ikx_j \cos\phi_o} - \frac{kY}{2} \sum_{i=N+1}^{N+M} I_i \int_{\Delta x_i} H_o^{(1)}\left(k\,|x_j - x_i|\right) dx_i \qquad \text{(A.16)}$$

for $j = N+1,\ldots,N+M$. From (3), the far field amplitude at the angle ϕ is now

$$P_H(\phi,\phi_o) = -\frac{kY}{2} \sum_{i=N+1}^{N+M} I_i \int_{\Delta x_i} e^{-ikx_i \cos\phi} dx_i \quad . \qquad \text{(A.17)}$$

A.2 E-Polarization

The integral equation pair given by (17) and (18) was solved in the same manner as described for the H-polarization case. The elements of the impedance matrix defined in (A.8) for the E-pol case are as follows:

$$Z_{e1} = \begin{cases} \frac{kZ}{2} \mu_r \int_{\Delta s_i} H_o^{(1)}(k_1 R_{j,i})\, ds_i & j \neq i \\ 1 & j = i \end{cases} \qquad \text{(A.18)}$$

for i = 1,...,N and j = 1,...,N;

$$Z_{m1} = \begin{cases} -\frac{ik_1}{2} \int_{\Delta s_i} \frac{y_j}{R_{j,i}} H_1^{(1)}(k_1 R_{j,i})\, ds_i & j \neq i\text{-}N \\ -1 & j = i\text{-}N \end{cases} \qquad \text{(A.19)}$$

for i = N+1,...,N+M and j = 1,...,N;

$$Z_{e2} = \begin{cases} 0 & j \neq i+N \\ -1 & j = i+N \end{cases} \qquad \text{(A.20)}$$

for i = 1,...,N and j = N+1,...,N+M;

$$Z_{m2} = \begin{cases} -\frac{kY}{2}\int_{\Delta s_i} \frac{1}{k R_{j,i}} H_1^{(1)}(k R_{j,i})\, ds_i & j \neq i \\ -\frac{kY}{2}\left\{\frac{i2}{\pi}\left[2(s_i - s_j)\,\ell n(R_{j,i}) - \left(2 - A'(k)\right)(s_i - s_j)\right] - \frac{2}{k} H_1^{(1)}\left(k\,|s_j - s_i|\right)\right\} & j = i \end{cases} \tag{A.21}$$

for $i = N+1,\ldots,N+M$ and $j = N+1,\ldots,N+M$. As for the H-polarization case, in the self-cell expressions, s_i is evaluated at the endpoint of the i^{th} segment.

Because of the sensitivity of the impedance element Z_{m2} to the $1/R_{j,i}$ term for segments near the self-cell, the adjacent cells needed to be evaluated analytically, as follows:

$$Z_{m2} = \frac{kY}{2}\frac{i}{\pi}\left\{\frac{8}{3k^2 \Delta s_i} + \left[2(s_i - s_j)\,\ell n\,(R_{j,i}) - \left(2 - A'(k)\right)s_i + 2|n_j - n_i|\,\mathrm{atan}\left(\frac{s_i - s_j}{|n_j - n_i|}\right)\right]\right\} \tag{A.22}$$

for $i = j \pm 1$, where s_i is evaluated over the i^{th} segment.

The source matrix is given by

$$V_j = \begin{cases} 0 & j = 1,\ldots,N \\ 2Y \sin\phi_o\, e^{-iks_j \cos\phi_o} & j = N+1,\ldots,N+M\ . \end{cases} \tag{A.23}$$

Given that $[Z_{j,i}]$ is nonsingular, the currents I_i can be calculated.

The aperture impedance for the E-polarization case is defined as

$$\eta_j = \frac{E_z(x_j,0)}{H_x(x_j,0)}$$

where the total electric field is equal to the magnetic current in the aperture, I_j for $j = N+1,\ldots,N+M$, and from (13), the total magnetic field in the aperture is now expressed as

$$H_x(x_j,0) = -\,2Y\sin\phi_o\, e^{-ikx_j\cos\phi_o} + \frac{kY}{2}\sum_{i=N+1}^{N+M} I_i \int_{\Delta x_i} \frac{1}{k|x_j-x_i|} H_1^{(1)}(k|x_j-x_i|)\, dx_i \quad \text{(A.24)}$$

for $j = N+1,\ldots,N+M$. From (12), the far field amplitude at the specified angle ϕ is now

$$P_E(\phi,\phi_o) = -\frac{k}{2}\sin\phi \sum_{i=N+1}^{N+M} I_i \int_{\Delta x_i} e^{-ikx_i\cos\phi}\, dx_i \quad . \qquad \text{(A.25)}$$

A.3 Program Listing

The expressions for the impedance and source matrices, the aperture impedance, and the far field amplitude were programmed for solution, as shown in the program listing of GAPSCAT.FTN below. The subroutines used in the program are contained in the file GAPSUB.FTN listed below also.

In running the program, the user is prompted for the polarization of the incident field, angle of incidence, angle of far field observation, and the relative permittivity ε_r of the gap cavity. For the relative permeability, it is assumed that $\mu_r = 1$, although this need not be the case. A menu is provided for the choice of shapes as shown in Figure 2. The dimensions are requested according to those defined in Figure 2. An arbitrary shaped gap may also be evaluated by specifying the coordinates of its corner points. The user is also prompted for the maximum segment size Δs_i to be used for the pulse basis functions. A segment size of $\Delta s_i/\lambda = 0.01$ was used for the results of Figures 3 to 14.

The impedance matrix $[Z_{j,i}]$ is solved for the H- or E-polarization case using the expressions (A.9) to (A.14) or (A.18) to (A.22), respectively. The numerical integration is done for the appropriate segments using Simpson's three-point composite integration over each segment. With the source matrix $[V_j]$ calculated from (A.15) or (A.23), the electric and magnetic currents contained in $[I_i]$ can then be determined. As listed, the program calculates the far field amplitude as a function of the gap depth using (A.17) for H-polarization or (A.23) for E-polarization, where the number of iterations is specified. For one iteration, the program also outputs the aperture impedance calculated from the total fields in the aperture.

```
c
c     GAPSCAT.FTN
c
c
c       This FORTRAN program computes the far field scattering due
c       to a narrow gap of arbitrary shape in an infinite ground
c       plane.  The moment method is applied to solve the currents
c       of two coupled equations.
c
c       INPUT    The user is prompted from the subroutine
c                GAPROM for the polarization and angle of the
c                incident field, angle of far field observation,
c                relative permittivity of gap filling, shape and
c                dimensions of gap, segment size, and number of
c                iterations with respect to gap depth.
c
c       OUTPUT FILES
c                GAPDAT   Contains input data.
c                IMPDAT   For one iteration, field or impedance
c                         in the aperture of gap.
c                AMPDAT   Contains the magnitude of the far field.
c                PHADAT   Contains the phase of the far field.
c
c       SUBROUTINES
c                HANKZ1   Computes the Hankel functions of the first
c                         kind of orders zero and one.
c                CHANK    Computes the Hankel functions of the first
c                         kind of orders zero and one given a
c                         complex argument.
c                CGECO    Factors a complex matrix and estimates the
c                         condition of the matrix.
c                CGESL    Solves the complex set of linear
c                         equations [A][x] = [b].
c
c
      integer pn
      parameter(pn=500)
      integer EorH,N,noS,gN,szN(50)

      real pi,k,phi,phio,w,d,maxC,q(50,2)
      real dStp(50),wStp(50),p(pn,2),m(pn,2),szSd(50),psi(50)
      real posiX,posiY,spaceX,spaceY,stepX
      real ni,si,nj,sj,xj,yj,DelS,R,Rm,tanf

      complex czero,ci,ctemp,er,ur,Ao,A,k1
      complex Z(pn,pn),LH0,LH1,LH2,aLH0,Ii(pn),Vj(pn)
      complex eta(pn),Hi(pn),Hs(pn),E(pn),Lsca,Psca

      integer ipvt(pn),iretrn
      real rc,krho
      complex wk(pn),H0,H1,H0o,H1o,ckrho
      logical Epol,Lossy,side,neg(50)
      common /prompts/ EorH,phio,phi,er,igap,wStp,dStp,w,d,noS,
     &                 q,maxC,noIter

    1 format(i1)
    2 format(i5)
    3 format(g16.8)
    4 format(a4)
    6 format(2g16.8)

      open(1,file='gapdat')
      open(2,file='impdat')
      open(3,file='ampdat')
      open(4,file='phadat')

c...Declaring constant values
      czero=cmplx(0.0,0.0)
      ci=cmplx(0.0,1.)
      pi=4.0*atan(1.0)
      k=2*pi
      Eo=1.0
      Ho=1.0
      Zo=sqrt(4.e-07*pi/8.854e-12)
      Yo=1./Zo
      ur=cmplx(1.,0.0)
      gam=0.5772157
      Ao=2*(log(k/2)+gam-ci*pi/2)
      iprg=1

c...Setting default values
      EorH=1
      phio=90.0
      phi=90.0
      er=cmplx(1.,0.0)
      w=0.15
      d=0.2
      noS=3
      maxC=0.01
      noIter=30
      adj=0.00001
      Epol=.false.
      Lossy=.false.
      side=.true.
```

```
c...Prompting user for input data
      call gaprom(iprg)

      if(EorH .eq. 1) Epol=.true.
      phio=phio*pi/180.0
      phi=phi*pi/180.0
      drat=dStp(1)/d
      k1=k*csqrt(er)
      A=2*(clog(k1/2)+gam-ci*pi/2)
      if(aImag(er) .ne. 0.0) Lossy=.true.

      dmin=0.025
      if(Epol) dmin=0.025
      dmax=d
      if(noIter .ne. 1)then
        dstep=(dmax-dmin)/(noIter-1)
        d=dmin
      endif
      DO 700 iter=1,noIter

c...Determining coordinates of corner points given gap type
        if(igap .eq. 1)then
c   RECTANGULAR
             noS=3
             q(3,1)=w/2
             q(3,2)=-d
             q(4,1)=-w/2
             q(4,2)=-d
        else if(igap .eq. 2)then
c   L-SHAPED
             noS=7
             dStp(1)=d*drat
             dStp(2)=d*(1-drat)
             q(3,1)=w/2
             q(3,2)=-dStp(1)
             q(8,1)=-w/2
             q(8,2)=-dStp(1)
             q(4,1)=w/2+wStp(2)
             q(4,2)=-dStp(1)
             q(7,1)=-w/2-wStp(3)
             q(7,2)=q(4,2)
             q(5,1)=q(4,1)
             q(5,2)=q(4,2)-dStp(2)
             q(6,1)=q(7,1)
             q(6,2)=q(5,2)
        else if(igap .eq. 3)then
c   V-SHAPED
             noS=2
             q(3,1)=0.0
             q(3,2)=-d
             adj=0.75*maxC
        else if(igap .eq. 4)then
c   T-SHAPED
             noS=5
             dStp(1)=d*drat
             dStp(2)=d*(1-drat)
             q(3,1)=w/2
             q(3,2)=-dStp(1)
             q(4,1)=w/2+wStp(2)
             q(4,2)=-dStp(1)
             q(5,1)=q(4,1)
             q(5,2)=q(4,2)-dStp(2)
             q(6,1)=-w/2
             q(6,2)=q(5,2)
        endif
c...Corner points of gap at y=0
        q(1,1)=-w/2
        q(1,2)=0.0
        q(2,1)=w/2
        q(2,2)=0.0
        q(noS+2,1)=-w/2
        q(noS+2,2)=0.0

c*******************************************************************
c*************   Current segment locations (xi,yi)    *************
c*******************************************************************
      N=0
      do 175 l=1,noS+1
c...Size (length) of lth side of gap
         szSd(l)=sqrt((q(l+1,1)-q(l,1))**2
     &                +(q(l+1,2)-q(l,2))**2)
c..Angle of rotation for each side with respect to x axis
         if(q(l+1,1) .lt. q(l,1))then
            psi(l)=asin((q(l,2)-q(l+1,2))/szSd(l))
            neg(l)=.true.
         else
            psi(l)=asin((q(l+1,2)-q(l,2))/szSd(l))
            neg(l)=.false.
         endif

         szN(l)=int(szSd(l)/maxC)+1
         N=N+szN(l)
         spaceX=(q(l+1,1)-q(l,1))/szN(l)
         spaceY=(q(l+1,2)-q(l,2))/szN(l)
         posiX=q(l,1)
         posiY=q(l,2)
```

```
c...ENDPOINTS of each segment are p, MIDPOINTS are m in (x,y)
c   coordinates
         do 170 i=N-szN(1)+1,N
           p(i,1)=posiX
           p(i,2)=posiY
           m(i,1)=posiX+spaceX/2.0
           m(i,2)=posiY+spaceY/2.0
           posiX=posiX+spaceX
           posiY=posiY+spaceY
170      continue
175   continue
      p(N+1,1)=-w/2
      p(N+1,2)=0.0
c...Number of segments in the aperture
      gN=szN(1)
c...Number of current coefficients to be calculated
      NgN=N+gN
      print *, ' d = ',d,' N = ',N,' gN = ',gN
c...Initializing matrices to zero
      do 190 j=1,NgN
      do 180 i=1,NgN
         Z(j,i)=czero
180   continue
         Vj(j)=czero
190   continue

c******************************************************************
c*****************     Impedance, Source,     *********************
c****************     and Current Matrices     ********************
c******************************************************************
c
c       VARIABLES:
c           H0o,H0  Hankel function of zero order in free
c                   space and in material er, respectively.
c           H1o,H1  Hankel function of first order in free
c                   space and in material er, respectively.
c           Green's Function integrals:
c               LH0  Integral of H0.
c               LH1  Integral of H1 for H-pol, of dH1/dy for
c                    E-pol.
c               LH2  Integral of H0o for H-pol, of dH1o/dy
c                    for E-pol.
c               aLH0 Analytical integral of H0o for evaluation
c                    of adjacent cells for LH2, E-pol case.
c
c       The integration is done one side at a time, for j=1,...,N,
c       in the clockwise direction, starting at (x,y)=(-w/2,0).
c
      istart=1
      istop=szN(1)
c...Source point is i of the lth side, observation point is j
      do 230 l=1,noS+1
       do 220 i=istart,istop
       do 210 j=1,N
c      Coordinate rotation for observation point
          sj=m(j,1)*cos(psi(l))+m(j,2)*sin(psi(l))
          nj=m(j,1)*sin(psi(l))-m(j,2)*cos(psi(l))
          if(neg(l))then
             sj=-sj
             nj=-nj
          endif
c...Integration over ith segment
          LH0=czero
          LH1=czero
          LH2=czero
          aLH0=czero
c    Magnitude between midpoints Rm=|r-r'|
         Rm=sqrt((m(j,1)-m(i,1))**2+(m(j,2)-m(i,2))**2)
        if(j .eq. i .or. Rm .le. adj)then
c    SMALL ARGUMENT APPROXIMATION integral for self-cell
c     and adjacent cells
          do 200 ip=i+1,i,-1
c         Coordinate rotation for source segment points
            si=p(ip,1)*cos(psi(l))+p(ip,2)*sin(psi(l))
            ni=p(ip,1)*sin(psi(l))-p(ip,2)*cos(psi(l))
            if(neg(l))then
               si=-si
               ni=-ni
            endif

            R=sqrt((sj-si)**2+(nj-ni)**2)
            if(j .eq. i .or. abs(nj-ni) .eq. 0.0)then
               tanf=pi/2
               absf=1.0
            else
               tanf=atan((si-sj)/abs(nj-ni))
               absf=abs(nj-ni)
            endif

            LH1=-k1**2/2*(nj-ni)*si
     &          +ci*2./pi*(nj-ni)/absf*tanf-LH1
            LH0=ci/pi*(2*(si-sj)*log(R)
     &            -(2-A)*si+2.0*abs(nj-ni)*tanf)-LH0
            LH2=ci/pi*(2*(si-sj)*log(R)
     &            -(2-Ao)*si+2.0*abs(nj-ni)*tanf)-LH2
            if(j .eq. i) GOTO 202
```

```
200       continue
202       if(j .eq. i)then
            LH0=2*(LH0+ci/pi*(2-A)*sj)
            LH2=2*(LH2+ci/pi*(2-Ao)*sj)
            if(Epol)then
              krho=k*abs(sj-si)
              call Hankzl(krho,1,H0,H1)
              LH2=LH2-2./k*H1
            endif
          endif
        else
c   SIMPSON'S THREE POINT COMPOSITE INTEGRATION
          do 204 ip=i+1,i,-1
c         Coordinate rotation for source segment endpoints
             si=p(ip,1)*cos(psi(l))+p(ip,2)*sin(psi(l))
             ni=p(ip,1)*sin(psi(l))-p(ip,2)*cos(psi(l))
             if(neg(l))then
                si=-si
                ni=-ni
             endif
             stepS=si
c         HANKEL FUNCTION evaluation at endpoints of segment
             R=sqrt((sj-si)**2+(nj-ni)**2)
             if(Lossy)then
               ckrho=k1*R
               call cHank(ckrho,2,H0,H1)
             else
               krho=Real(k1)*R
               call Hankzl(krho,2,H0,H1)
             endif
             krho=k*R
             call Hankzl(krho,2,H0o,H1o)

             LH0=H0+LH0
             if(Epol)then
               LH1=k1*m(j,2)/R*H1+LH1
               if(j .eq. (i-1) .or. j .eq. (i+1))then
                 if(abs(nj-ni) .eq. 0.0)then
                    tanf=pi/2
                    absf=1.0
                 else
                    tanf=atan((si-sj)/abs(nj-ni))
                    absf=abs(nj-ni)
                 endif
                 aLH0=ci/pi*(2*(si-sj)*log(R)
     &                -(2.0-Ao)*si+2.0*abs(nj-ni)*tanf)-aLH0
               else
                 LH2=H1o/k/R+LH2
               endif
             else
               LH1=k1*(nj-ni)/R*H1+LH1
               LH2=H0o+LH2
             endif
204       continue

c       Coordinate rotation for source segment midpoints
          si=m(i,1)*cos(psi(l))+m(i,2)*sin(psi(l))
          ni=m(i,1)*sin(psi(l))-m(i,2)*cos(psi(l))
          if(neg(l))then
             si=-si
             ni=-ni
          endif
          stepS=abs(stepS-si)
          DelS=2*stepS
c       HANKEL FUNCTION evaluation at midpoint of segment
          R=sqrt((sj-si)**2+(nj-ni)**2)
          if(Lossy)then
            ckrho=k1*R
            call cHank(ckrho,2,H0,H1)
          else
            krho=Real(k1)*R
            call Hankzl(krho,2,H0,H1)
          endif
          krho=k*R
          call Hankzl(krho,2,H0o,H1o)
c     GREEN'S FUNCTION INTEGRALS
          LH0=stepS/3*(4*H0+LH0)
          if(Epol)then
            LH1=stepS/3*(4*k1*m(j,2)/R*H1+LH1)
            if(j .eq. (i-1) .or. j .eq. (i+1))then
               LH2=-ci*8./3/pi/k**2/(DelS)-aLH0
            else
               LH2=stepS/3*(4*H1o/k/R+LH2)
            endif
          else
            LH1=stepS/3*(4*k1*(nj-ni)/R*H1+LH1)
            LH2=stepS/3*(4*H0o+LH2)
          endif
        endif
        if(i .ne. j .and. Rm .le. adj)then
           LH0=-LH0
        endif

        if(Epol)then
c   E-POL IMPEDANCE MATRIX
          Z(j,i)=k*Zo*ur/2*LH0
```

```
         if(i .le. gN .and. j .ne. i)then
            Z(j,N+i)=-ci/2*LH1
         else if(i .le. gN .and. j .eq. i)then
            Z(j,N+i)=-1.
         endif

         if(j .le. gN)then
           if(j .ne. i)then
              Z(N+j,i)=czero
           else
              Z(N+j,i)=-1.
           endif
           if(i .le. gN)then
              Z(N+j,N+i)=-k*Yo/2*LH2
           endif
         endif
       else
c   H-POL IMPEDANCE MATRIX
         if(j .ne. i)then
            Z(j,i)=ci/2*LH1
         else
            Z(j,i)=-1.
         endif
         if(i .le. gN)then
            Z(j,N+i)=k*Yo*er/2*LH0
         endif

         if(j .le. gN)then
           if(j .ne. i)then
              Z(N+j,i)=czero
           else
              Z(N+j,i)=1.
           endif
           if(i .le. gN)then
              Z(N+j,N+i)=k*Yo/2*LH2
           endif
         endif
       endif
210   continue

c...Incident Field (Source) matrix elements
      xj=m(i,1)
      yj=m(i,2)
      Vj(i)=czero
      if(i .le. gN)then
        if(Epol)then
c   E-POL INCIDENT FIELD Hx
          Vj(N+i)=2*Yo*sin(phio)*cexp(-ci*k*xj*cos(phio))
        else
c   H-POL INCIDENT FIELD Hz
          Vj(N+i)=2*cexp(-ci*k*xj*cos(phio))
        endif
      endif
220   continue
        istart=istop+1
        istop=istop+szN(l+1)
230   continue


c...Calling subroutines to calculate the current matrix
      call CGECO(Z,pn,NgN,ipvt,rc,wk)
      call CGESL(Z,pn,NgN,ipvt,Vj,0)

      print *,'     The condition number is ',rc
      do 310 i=1,NgN
c   CURRENT MATRIX
         Ii(i)=Vj(i)
310   continue

      if(noIter .eq. 1)then
c*****************************************************************
c*******************   Aperture Impedance   ********************
c*****************************************************************
      l=1
      do 500 j=1,gN
         Hs(j)=czero
       do 480 i=1,gN
         sj=m(j,1)*cos(psi(l))+m(j,2)*sin(psi(l))
         nj=m(j,1)*sin(psi(l))-m(j,2)*cos(psi(l))
         if(neg(l))then
            sj=-sj
            nj=-nj
         endif

c...Integration over ith segment
         LH2=czero
         aLH0=czero
       if(j .eq. i)then
c   SMALL ARGUMENT APPROXIMATION integral for self-cell
c    and adjacent cells
         do 470 ip=i+1,i,-1
c        Coordinate rotation for source segment points
           si=p(ip,1)*cos(psi(l))+p(ip,2)*sin(psi(l))
           ni=p(ip,1)*sin(psi(l))-p(ip,2)*cos(psi(l))
           if(neg(l))then
              si=-si
```

```
               ni=-ni
            endif
            R=sqrt((sj-si)**2+(nj-ni)**2)
            if(j .eq. i .or. abs(nj-ni) .eq. 0.0)then
               tanf=0.0
               absf=1.0
            else
               tanf=atan((si-sj)/abs(nj-ni))
               absf=abs(nj-ni)
            endif
            LH2=ci/pi*(2*(si-sj)*log(R)
     &            -(2-Ao)*si+2.0*abs(nj-ni)*tanf)-LH2
            if(j .eq. i) GOTO 472
470      continue
472      if(j .eq. i)then
            LH2=2*(LH2+ci/pi*(2-Ao)*sj)
            if(Epol)then
              krho=k*abs(sj-si)
              call Hankz1(krho,1,H0o,H1o)
              LH2=LH2-2./k*H1o
            endif
         endif
       else
c   SIMPSON'S THREE POINT COMPOSITE INTEGRATION
         do 474 ip=i+1,i,-1
c        Coordinate rotation for source segment endpoints
            si=p(ip,1)*cos(psi(l))+p(ip,2)*sin(psi(l))
            ni=p(ip,1)*sin(psi(l))-p(ip,2)*cos(psi(l))
            if(neg(l))then
               si=-si
               ni=-ni
            endif
            stepS=si
            R=sqrt((sj-si)**2+(nj-ni)**2)
            krho=k*R
            call Hankz1(krho,2,H0o,H1o)
            if(Epol)then
              if(j .eq. (i-1) .or. j .eq. (i+1))then
                if(abs(nj-ni) .eq. 0.0)then
                   tanf=pi/2
                   absf=1.0
                else
                   tanf=atan((si-sj)/abs(nj-ni))
                   absf=abs(nj-ni)
                endif
                aLH0=ci/pi*(2*(si-sj)*log(R)
     &               -(2.0-Ao)*si+2.0*abs(nj-ni)*tanf)-aLH0
              else
                LH2=H1o/k/R+LH2
              endif
            else
              LH2=H0o+LH2
            endif
474      continue
c        Coordinate rotation for source segment midpoints
         si=m(i,1)*cos(psi(l))+m(i,2)*sin(psi(l))
         ni=m(i,1)*sin(psi(l))-m(i,2)*cos(psi(l))
         if(neg(l))then
            si=-si
            ni=-ni
         endif
         stepS=abs(stepS-si)
         R=sqrt((sj-si)**2+(nj-ni)**2)
         krho=k*R
         call Hankz1(krho,2,H0o,H1o)
         if(Epol)then
           if(j .eq. (i-1) .or. j .eq. (i+1))then
              LH2=-ci*8./3/pi/k**2/(2*stepS)-aLH0
           else
              LH2=stepS/3*(4*H1o/k/R+LH2)
           endif
         else
           LH2=stepS/3*(4*H0o+LH2)
         endif
       endif
c   SCATTERED MAGNETIC FIELD IN THE APERTURE
      Hs(j)=k*Yo/2*Ii(N+i)*LH2+Hs(j)

480   continue
c   ELECTRIC FIELD IN THE APERTURE
      E(j)=Ii(N+j)

      xj=m(j,1)
      yj=m(j,2)
c   INCIDENT MAGNETIC FIELDS
      if(Epol)then
         Hi(j)=2*Yo*sin(phio)*cexp(-ci*k*xj*cos(phio))
      else
         Hi(j)=2*cos(k*yj*sin(phio))*cexp(-ci*k*xj*cos(phio))
      endif
c   APERTURE IMPEDANCE
      eta(j)=E(j)/(Hi(j)+Hs(j))
      write(2,*) m(j,1),cabs(eta(j))
500   continue
      endif
```

```
c*********************************************************************
c******************   Far Field Amplitude   **********************
c*********************************************************************
        Psca=czero
        DelX=w/gN/2
        do 600 i=1,gN
c...Simpson's three point composite integration over each
c   segment in the aperture
           Lsca=DelX/3*(cexp(-ci*k*p(i,1)*cos(phi))
     &           +4*cexp(-ci*k*m(i,1)*cos(phi))
     &           +cexp(-ci*k*p(i+1,1)*cos(phi)))
           Psca=Ii(N+i)*Lsca+Psca
600     continue
        if(Epol)then
           Psca=-k*sin(phi)/2*Psca
        else
           Psca=-k*Yo/2*Psca
        endif

        write(3,*) d,cabs(Psca)
        write(4,*) d,180/pi*(atan2(aImag(Psca),Real(Psca)))
        print *,'      Exact: |Psca| = ',cabs(Psca),'  arg Psca = ',
     &  180/pi*(atan2(aImag(Psca),Real(Psca)))

        d=d+dstep
700     continue

800   call exit
      END
```

```fortran
C
C     GAPSUB.FTN
C
C        This file contains the subroutines and functions used by
C        GAPSCAT.FTN and ANAGAP.FTN.
C
C*******************************************************************
      SUBROUTINE GAPROM(IPRG)
C*******************************************************************
C
C        Called to prompt user for the input parameters.
C
      integer EorH,N,noS,gN
      real phi,phio,w,d,maxC,q(50,2)
      real dStp(50),wStp(50)
      complex er,ctemp
      common /prompts/ EorH,phio,phi,er,igap,wStp,dStp,w,d,noS,
     &                 q,maxC,noIter
 1    format(i1)
 2    format(i5)
 3    format(g16.8)
 6    format(2g16.8)
      open(1,file='gapdat')
      print *,' '
 20   write(*,25) EorH
 25   format(/'Polarization of incident field:'/
     &       '  1) E- or 2) H-pol [',i1,']? ')
      read(*,1,err=20) itemp
      if(itemp .eq. 2) EorH=2

 30   write(*,35) phio
 35   format(/,'Angle of incidence = ',g12.6,'degrees ')
      read(*,3,err=20) rtemp
      if(rtemp .ne. 0.0) phio=rtemp
 36   write(*,37) phi
 37   format(/,'Angle of observation = ',g12.6,'degrees ')
      read(*,3,err=30) rtemp
      if(rtemp .ne. 0.0) phi=rtemp

 38   write(*,39) er
 39   format(/,'Relative permittivity = (',g12.6,',',g12.6,')')
      read(*,6,err=36) ctemp
      if(ctemp .ne. 0.0) er=ctemp
        print *,er

 40   write(*,45)
 45   format(/'Type of gap:'/
     &       '  1) rectangular, 2) T-shape, 3) triangular,'/
     &       '  4) L-shape, or 5) arbitrary ? ')
      read(*,1,err=20) igap

      if(igap .eq. 2 .or. igap .eq. 4)then
         if(igap .eq. 2)then
            now=3
         else
            now=2
         endif
         do 47 i=1,now
           write(*,*) 'Enter width w',i
           read(*,*) wStp(i)
           if(i .ne. 3)then
             write(*,*) 'Enter depth d',i
             read(*,*) dStp(i)
           endif
 47      continue
         w=wStp(1)
         d=dStp(1)+dStp(2)
      else if(igap .eq. 1 .or. igap .eq. 3 .or.
     &        (igap .eq. 5 .and. iprg .eq. 1))then
 90      write(*,95) w
 95      format(/,'Gap width = ',g12.6,'wavelengths ')
         read(*,3,err=30) rtemp
         if(rtemp .gt. 0.0) w=rtemp

 100     write(*,105) d
 105     format(/'Gap depth = ',g12.6,'wavelengths ')
         read(*,3,err=90) rtemp
         if(rtemp .gt. 0.0) d=rtemp

         if(igap .eq. 5)then
 110       write(*,115) noS
 115       format(/,'Number of sides (excluding aperture) = ',i2)
           read(*,*,err=30) rtemp
           if(rtemp .gt. 0.0) noS=int(rtemp)
           print *,'Enter the following coordinates, beginning '
           print *,'  with (-w/2,0) and going cw: '
           do 127 i=1,noS+1
 120          write(*,125) i
 125          format(/'Corner ',i1,' (x,y) ')
              read(*,3,err=110) q(i,1)
              read(*,3,err=110) q(i,2)
 127       continue
           noIter=1
         endif
      else
        GOTO 40
```

```
      endif

      if(iprg .eq. 1)then
130      write(*,135) maxC
135      format(/'Max segment size = ',g12.6,' wavelengths ')
         read(*,3,err=110) rtemp
         if(rtemp .gt. 0.0) maxC=rtemp
      endif

      if(igap .ne. 5)then
140      write(*,145) noIter
145      format(/'Number of iterations = ',i3,'  ')
         read(*,*,err=130) rtemp
         if(rtemp .gt. 0.0) noIter=int(rtemp)
      endif

c...Writing input data to file GAPDAT
      write(1,2) EorH
      write(1,3) phio
      write(1,3) phi
      write(1,6) er
      write(1,2) igap
      if(igap .eq. 2 .or. igap .eq. 4)then
         do 150 i=1,2
             write(1,3) wStp(i)
             write(1,3) dStp(i)
150      continue
      else
        write(1,3) w
        write(1,3) d
        if(igap .eq. 5)then
          write(1,2) noS
          do 160 i=1,noS+1
            write(1,6) q(i,1),q(i,2)
160       continue
        endif
      endif
      if(iprg .eq. 1)then
         write(1,3) maxC
      endif
      write(1,2) noIter
      close(1)
      return
      end
C*******************************************************************
        COMPLEX FUNCTION CTAN(CARG)
C*******************************************************************
C       Calculates the tangent given a complex argument
C
        complex ci,carg
        ci=cmplx(0.,1)
        ctan=-ci*(cexp(ci*carg)-cexp(-ci*carg))
     &       /(cexp(ci*carg)+cexp(-ci*carg))
        return
        end
C******************************************************************C
C                                                                  C
      SUBROUTINE HANKZ1(R,N,HZERO,HONE)
C                                                                  C
C******************************************************************C
C
C     Called to compute Hankel functions of the first kind
C     for orders one and zero.  The argument is variable R
C     and must be positive.
C
C.....HANKEL FUNCTIONS ARE OF FIRST KIND--J+IY
C.....   N=0 RETURNS HZERO (H-zero)
C.....   N=1 RETURNS HONE  (H-one)
C.....   N=2 RETURNS HZERO AND HONE
C.....SUBROUTINE REQUIRES R>0
C.....SUBROUTINE ADAM MUST BE SUPPLIED BY USER
C
      DIMENSION A(7),B(7),C(7),D(7),E(7),F(7),G(7),H(7)
      COMPLEX HZERO,HONE
      DATA A,B,C,D,E,F,G,H/1.0,-2.2499997,1.2656208,-0.3163866,
     &0.0444479,-0.0039444,0.00021,0.36746691,0.60559366,-0.74350384,
     &0.25300117,-0.04261214,0.00427916,-0.00024846,0.5,-0.56249985,
     &0.21093573,-0.03954289,0.00443319,-0.00031761,0.00001109,
     &-0.6366198,0.2212091,2.1682709,-1.3164827,0.3123951,-0.0400976,
     &0.0027873,0.79788456,-0.00000077,-0.0055274,-0.00009512,
     &0.00137237,-0.00072805,0.00014476,-0.78539816,-0.04166397,
     &-0.00003954,0.00262573,-0.00054125,-0.00029333,0.00013558,
     &0.79788456,0.00000156,0.01659667,0.00017105,-0.00249511,
     &0.00113653,-0.00020033,-2.35619449,0.12499612,0.0000565,
     &-0.00637879,0.00074348,0.00079824,-0.00029166/
      IF (R.LE.0.0) GO TO 50
      IF (N.LT.0.OR.N.GT.2) GO TO 50
      IF (R.GT.3.0) GO TO 20
      X=R*R/9.0
      IF (N.EQ.1) GO TO 10
      CALL ADAM(A,X,BJ)
      CALL ADAM(B,X,Y)
      BY=0.6366198*ALOG(0.5*R)*BJ+Y
      HZERO=CMPLX(BJ,BY)
      IF (N.EQ.0) RETURN
   10 CALL ADAM(C,X,Y)
```

```
      BJ=R*Y
      CALL ADAM(D,X,Y)
      BY=0.6366198*ALOG(0.5*R)*BJ+Y/R
      HONE=CMPLX(BJ,BY)
      RETURN
   20 X=3.0/R
      IF (N.EQ.1) GO TO 30
      CALL ADAM(E,X,Y)
      FOOL=Y/SQRT(R)
      CALL ADAM(F,X,Y)
      T=R+Y
      BJ=FOOL*COS(T)
      BY=FOOL*SIN(T)
      HZERO=CMPLX(BJ,BY)
      IF (N.EQ.0) RETURN
   30 CALL ADAM(G,X,Y)
      FOOL=Y/SQRT(R)
      CALL ADAM(H,X,Y)
      T=R+Y
      BJ=FOOL*COS(T)
      BY=FOOL*SIN(T)
      HONE=CMPLX(BJ,BY)
      RETURN
   50 WRITE(6,90) N,R
   90 FORMAT(32H0SICK DATA IN HANKZ1  *QUIT*  N=,I2,2X,2HR=,E11.3)
      CALL SYSTEM
      END
C*********************************************************************C
C                                                                     C
      SUBROUTINE ADAM(C,X,Y)
C                                                                     C
C*********************************************************************C
C
C     Called by subroutine HANKZ1 to compute the value of a 7th
C     order polynomial whose argument is X and coefficients are
C     contained in vector C.
C
      DIMENSION C(7)
C
      Y=X*C(7)
C
         DO 10 I=1,5
         Y=X*(C(7-I)+Y)
   10    CONTINUE
C
      Y=Y+C(1)
      RETURN
      END
C*********************************************************************C
C                                                                     C
      SUBROUTINE CHANK(Z,N,H0,H1)
C                                                                     C
C*********************************************************************C
C     CALCULATES HANKEL FUNCTIONS ZEROTH AND FIRST ORDER
C OF THE FIRST KIND OF COMPLEX ARGUMENT H0=J0+I*Y0,  H1=J1+I*Y1.
C THE ACCURACY IS VERY GOOD UP TO |Z|<10.  ABOVE THAT THE
C ACCURACY IS TO THE ORDER OF LARGE ARGUMENT EXPANSION.
C
C      N=0 RETURNS H0
C      N=1 RETURNS H1
C      N=2 RETURNS H0 AND H1
C      N=3 RETURNS J0 AND J1
C
        COMPLEX J0,Y0,J1,Y1,Z,TERM
        COMPLEX I,H0,H1
        I=(0.0,1.0)
        PI=4.*ATAN(1.)
        IF(N.EQ.1)GOTO10
       IF(CABS(Z).GT. 12.)THEN
        J0=CSQRT(2/(PI*Z))*CCOS(Z-PI/4.)
        Y0=CSQRT(2/(PI*Z))*CSIN(Z-PI/4.)
       ELSE
        AONE=1.
        TERM=(1.0,0.0)
        J0=(1.0,0.0)
        Y0=(0.0,0.0)
        SUM=0.0
        M=0
100     M=M+1
        SUM=SUM+1./FLOAT(M)
        AONE=AONE*(-1.)
        TERM=TERM*( (Z/2)/FLOAT(M) )**2
        J0=J0+AONE*TERM
        Y0=Y0+AONE*TERM*SUM
        IF(M.LE.10 .OR. CABS(FLOAT(M)/(Z/2)).LT. 5)GOTO 100
        Y0=2.*(J0*(CLOG(Z/2)+0.57721)-Y0)/PI
       ENDIF
       IF(N.EQ.3)THEN
        H0=J0
       ELSE
        H0=J0+I*Y0
       ENDIF
        IF(N.EQ.0)RETURN
C COMPUTATION OF J1 AND Y1
10     IF(CABS(Z).GT. 15.)THEN
        J1=CSQRT(2/(PI*Z))*CCOS(Z-3.*PI/4.)
```

```
        Y1=CSQRT(2/(PI*Z))*CSIN(Z-3.*PI/4.)
       ELSE
        AONE=1.
        TERM=(1.0,0.0)
        J1=(1.0,0.0)
        Y1=(1.0,0.0)
        SUM=0.0
        M=0
200     M=M+1
        SUM=SUM+1./FLOAT(M)
        AONE=AONE*(-1.)
        TERM=TERM*(Z/2)**2/(FLOAT(M)*FLOAT(M+1))
        J1=J1+AONE*TERM
        Y1=Y1+AONE*TERM*(1./FLOAT(M+1)+2.*SUM)
        IF(M.LE.10 .OR. CABS(FLOAT(M)/(Z/2)).LT. 5)GOTO 200
        J1=J1*Z/2.
        Y1=(2.*J1*(CLOG(Z/2)+0.57721)-2./Z-Y1*Z/2.)/PI
       ENDIF
       IF(N.EQ.3)THEN
        H1=J1
       ELSE
        H1=J1+I*Y1
       ENDIF
       RETURN
       END
C*****************************************************************
C
        SUBROUTINE MODBESS(X,I0,I1)
C
C*****************************************************************
C       CALCULATES THE MODIFIED BESSEL FUNCTION OF THE
C ZEROTH AND FIRST ORDER.  ARGUMENT X IS POSITIVE AND REAL.
C SEE PAGE 378 ABRAMOVITZ
C
        REAL I0,I1
        T=X/3.75
        IF(T .LT. -1)THEN
        PRINT *,'ERROR'
        STOP
        ELSE
        ENDIF
        IF(T .GE. -1 .OR. T.LE.1)THEN
        I0=1+3.5156229*T**2+3.0899424*T**4+1.2067492*T**6+
     &     0.2659732*T**8+0.0360768*T**10+0.0045813*T**12
        I1=0.5+0.8789059*T**2+0.51498869*T**4+0.15084934*T**6+
     &     0.02658733*T**8+0.00301532*T**10+0.00032411*T**12
        I1=X*I1
        ELSE
        I0=0.39894228+0.01328592/T+0.00225319/T**2-0.00157565/T**3
     &     +0.00916281/T**4-0.02057706/T**5+0.02635537/T**6
     &     -0.01647633/T**7+0.00392377/T**8
        I0=I0*EXP(X)*SQRT(1/X)
        I1=0.39894228-0.03988024/T-0.00362018/T**2+0.00163801/T**3
     &      -0.01031555/T**4+0.02282967/T**5-0.02895312/T**6
     &      +0.01787654/T**7-0.00420059/T**8
        ENDIF
        RETURN
        END
C***************************************************************C
C                                                               C
C     The following subroutines are standard LINPACK routines   C
C     to perform L-U decomposition and back substitution on a   C
C     single precision complex matrix.  See CC-Memo 407 sec 2.1 C
C     for documentation on these routines.                      C
C                                                               C
C***************************************************************C
C                                                               C
      SUBROUTINE CGECO(A,LDA,N,IPVT,RCOND,Z)
C                                                               C
C***************************************************************C
C                                                               C
C NAASA  2.1.042 CGECO    FTN-A 05-02-78     THE UNIV OF MICH COMP CTR
C
      INTEGER LDA,N,IPVT(1)
      COMPLEX A(LDA,1),Z(1)
      REAL RCOND
C
C     CGECO FACTORS A COMPLEX MATRIX BY GAUSSIAN ELIMINATION
C     AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C     IF  RCOND  IS NOT NEEDED, CGEFA IS SLIGHTLY FASTER.
C     TO SOLVE  A*X = B , FOLLOW CGECO BY CGESL.
C     TO COMPUTE  INVERSE(A)*C , FOLLOW CGECO BY CGESL.
C     TO COMPUTE  DETERMINANT(A) , FOLLOW CGECO BY CGEDI.
C     TO COMPUTE  INVERSE(A) , FOLLOW CGECO BY CGEDI.
C
C     ON ENTRY
C
C        A       COMPLEX(LDA, N)
C                THE MATRIX TO BE FACTORED.
C
C        LDA     INTEGER
C                THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
```

```
C
C     ON RETURN
C
C        A       AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
C                WHICH WERE USED TO OBTAIN IT.
C                THE FACTORIZATION CAN BE WRITTEN  A = L*U  WHERE
C                L  IS A PRODUCT OF PERMUTATION AND UNIT LOWER
C                TRIANGULAR MATRICES AND  U  IS UPPER TRIANGULAR.
C
C        IPVT    INTEGER(N)
C                AN INTEGER VECTOR OF PIVOT INDICES.
C
C        RCOND   REAL
C                AN ESTIMATE OF THE RECIPROCAL CONDITION OF  A .
C                FOR THE SYSTEM  A*X = B , RELATIVE PERTURBATIONS
C                IN  A  AND  B  OF SIZE  EPSILON  MAY CAUSE
C                RELATIVE PERTURBATIONS IN  X  OF SIZE  EPSILON/RCOND .
C                IF  RCOND  IS SO SMALL THAT THE LOGICAL EXPRESSION
C                           1.0 + RCOND .EQ. 1.0
C                IS TRUE, THEN  A  MAY BE SINGULAR TO WORKING
C                PRECISION.  IN PARTICULAR,  RCOND  IS ZERO  IF
C                EXACT SINGULARITY IS DETECTED OR THE ESTIMATE
C                UNDERFLOWS.
C
C        Z       COMPLEX(N)
C                A WORK VECTOR WHOSE CONTENTS ARE USUALLY UNIMPORTANT.
C                IF  A  IS CLOSE TO A SINGULAR MATRIX, THEN  Z  IS
C                AN APPROXIMATE NULL VECTOR IN THE SENSE THAT
C                NORM(A*Z) = RCOND*NORM(A)*NORM(Z) .
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     LINPACK CGEFA
C     BLAS CAXPY,CDOTC,CSSCAL,SCASUM
C     FORTRAN ABS,AIMAG,AMAX1,CMPLX,CONJG,REAL
C
C     INTERNAL VARIABLES
C
      COMPLEX CDOTC,EK,T,WK,WKM
      REAL ANORM,S,SCASUM,SM,YNORM
      INTEGER INFO,J,K,KB,KP1,L
C
      COMPLEX ZDUM,ZDUM1,ZDUM2,CSIGN1
      REAL CABS1
      CABS1(ZDUM) = ABS(REAL(ZDUM)) + ABS(AIMAG(ZDUM))
      CSIGN1(ZDUM1,ZDUM2) = CABS1(ZDUM1)*(ZDUM2/CABS1(ZDUM2))
C
CCC   Compute 1-NORM of A
C
      ANORM = 0.0E0
      DO 10 J = 1, N
      ANORM = AMAX1(ANORM,SCASUM(N,A(1,J),1))
   10 CONTINUE
C
CCC   Factor
C
      CALL CGEFA(A,LDA,N,IPVT,INFO)
C
C     RCOND = 1/(NORM(A)*(ESTIMATE OF NORM(INVERSE(A)))) .
C     ESTIMATE = NORM(Z)/NORM(Y) WHERE  A*Z = Y  AND  CTRANS(A)*Y = E .
C     CTRANS(A)  IS THE CONJUGATE TRANSPOSE OF A .
C     THE COMPONENTS OF  E  ARE CHOSEN TO CAUSE MAXIMUM LOCAL
C     GROWTH IN THE ELEMENTS OF W  WHERE  CTRANS(U)*W = E .
C     THE VECTORS ARE FREQUENTLY RESCALED TO AVOID OVERFLOW.
C
C     SOLVE CTRANS(U)*W = E
C
      EK = CMPLX(1.0E0,0.0E0)
      DO 20 J = 1, N
         Z(J) = CMPLX(0.0E0,0.0E0)
   20 CONTINUE
      DO 100 K = 1, N
         IF (CABS1(Z(K)) .NE. 0.0E0) EK = CSIGN1(EK,-Z(K))
         IF (CABS1(EK-Z(K)) .LE. CABS1(A(K,K))) GO TO 30
            S = CABS1(A(K,K))/CABS1(EK-Z(K))
            CALL CSSCAL(N,S,Z,1)
            EK = CMPLX(S,0.0E0)*EK
   30    CONTINUE
         WK = EK - Z(K)
         WKM = -EK - Z(K)
         S = CABS1(WK)
         SM = CABS1(WKM)
         IF (CABS1(A(K,K)) .EQ. 0.0E0) GO TO 40
            WK = WK/CONJG(A(K,K))
            WKM = WKM/CONJG(A(K,K))
         GO TO 50
   40    CONTINUE
            WK = CMPLX(1.0E0,0.0E0)
            WKM = CMPLX(1.0E0,0.0E0)
   50    CONTINUE
         KP1 = K + 1
         IF (KP1 .GT. N) GO TO 90
            DO 60 J = KP1, N
```

```
               SM = SM + CABS1(Z(J)+WKM*CONJG(A(K,J)))
               Z(J) = Z(J) + WK*CONJG(A(K,J))
               S = S + CABS1(Z(J))
   60       CONTINUE
            IF (S .GE. SM) GO TO 80
               T = WKM - WK
               WK = WKM
               DO 70 J = KP1, N
                  Z(J) = Z(J) + T*CONJG(A(K,J))
   70          CONTINUE
   80       CONTINUE
   90    CONTINUE
         Z(K) = WK
  100 CONTINUE
      S = 1.0E0/SCASUM(N,Z,1)
      CALL CSSCAL(N,S,Z,1)
C
CCC   Solve CTRANS(L)*Y = V
C
      DO 120 KB = 1, N
         K = N + 1 - KB
         IF (K .LT. N) Z(K) = Z(K) + CDOTC(N-K,A(K+1,K),1,Z(K+1),1)
         IF (CABS1(Z(K)) .LE. 1.0E0) GO TO 110
            S = 1.0E0/CABS1(Z(K))
            CALL CSSCAL(N,S,Z,1)
  110    CONTINUE
         L = IPVT(K)
         T = Z(L)
         Z(L) = Z(K)
         Z(K) = T
  120 CONTINUE
      S = 1.0E0/SCASUM(N,Z,1)
      CALL CSSCAL(N,S,Z,1)
C
      YNORM = 1.0E0
C
CCC   Solve L*V = Y
C
      DO 140 K = 1, N
         L = IPVT(K)
         T = Z(L)
         Z(L) = Z(K)
         Z(K) = T
         IF (K .LT. N) CALL CAXPY(N-K,T,A(K+1,K),1,Z(K+1),1)
         IF (CABS1(Z(K)) .LE. 1.0E0) GO TO 130
            S = 1.0E0/CABS1(Z(K))
            CALL CSSCAL(N,S,Z,1)
            YNORM = S*YNORM
  130    CONTINUE
  140 CONTINUE
      S = 1.0E0/SCASUM(N,Z,1)
      CALL CSSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
CCC   Solve U*Z = V
C
      DO 160 KB = 1, N
         K = N + 1 - KB
         IF (CABS1(Z(K)) .LE. CABS1(A(K,K))) GO TO 150
            S = CABS1(A(K,K))/CABS1(Z(K))
            CALL CSSCAL(N,S,Z,1)
            YNORM = S*YNORM
  150    CONTINUE
         IF (CABS1(A(K,K)) .NE. 0.0E0) Z(K) = Z(K)/A(K,K)
         IF (CABS1(A(K,K)) .EQ. 0.0E0) Z(K) = CMPLX(1.0E0,0.0E0)
         T = -Z(K)
         CALL CAXPY(K-1,T,A(1,K),1,Z(1),1)
  160 CONTINUE
C     MAKE ZNORM = 1.0
      S = 1.0E0/SCASUM(N,Z,1)
      CALL CSSCAL(N,S,Z,1)
      YNORM = S*YNORM
C
      IF (ANORM .NE. 0.0E0) RCOND = YNORM/ANORM
      IF (ANORM .EQ. 0.0E0) RCOND = 0.0E0
      RETURN
      END
C*****************************************************************C
C                                                                 C
      SUBROUTINE CGEFA(A,LDA,N,IPVT,INFO)
C                                                                 C
C*****************************************************************C
C                                                                 C
C NAASA  2.1.043 CGEFA    FTN-A 05-02-78     THE UNIV OF MICH COMP CTR
C
      INTEGER LDA,N,IPVT(1),INFO
      COMPLEX A(LDA,1)
C
C     CGEFA FACTORS A COMPLEX MATRIX BY GAUSSIAN ELIMINATION.
C
C     CGEFA IS USUALLY CALLED BY CGECO, BUT IT CAN BE CALLED
C     DIRECTLY WITH A SAVING IN TIME IF  RCOND  IS NOT NEEDED.
C     (TIME FOR CGECO) = (1 + 9/N)*(TIME FOR CGEFA) .
C
C     ON ENTRY
C
```

```
C        A        COMPLEX(LDA, N)
C                 THE MATRIX TO BE FACTORED.
C
C        LDA      INTEGER
C                 THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N        INTEGER
C                 THE ORDER OF THE MATRIX  A .
C
C     ON RETURN
C
C        A        AN UPPER TRIANGULAR MATRIX AND THE MULTIPLIERS
C                 WHICH WERE USED TO OBTAIN IT.
C                 THE FACTORIZATION CAN BE WRITTEN  A = L*U  WHERE
C                 L  IS A PRODUCT OF PERMUTATION AND UNIT LOWER
C                 TRIANGULAR MATRICES AND  U  IS UPPER TRIANGULAR.
C
C        IPVT     INTEGER(N)
C                 AN INTEGER VECTOR OF PIVOT INDICES.
C
C        INFO     INTEGER
C                 = 0  NORMAL VALUE.
C                 = K  IF  U(K,K) .EQ. 0.0 .  THIS IS NOT AN ERROR
C                      CONDITION FOR THIS SUBROUTINE, BUT IT DOES
C                      INDICATE THAT CGESL OR CGEDI WILL DIVIDE BY ZERO
C                      IF CALLED.  USE  RCOND  IN CGECO FOR A RELIABLE
C                      INDICATION OF SINGULARITY.
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS CAXPY,CSCAL,ICAMAX
C     FORTRAN ABS,AIMAG,CMPLX,REAL
C
C     INTERNAL VARIABLES
C
      COMPLEX T
      INTEGER ICAMAX,J,K,KP1,L,NM1
C
      COMPLEX ZDUM
      REAL CABS1
      CABS1(ZDUM) = ABS(REAL(ZDUM)) + ABS(AIMAG(ZDUM))
C
CCC   Gaussian elimination with partial pivoting
C
      INFO = 0
      NM1 = N - 1
      IF (NM1 .LT. 1) GO TO 70
      DO 60 K = 1, NM1
         KP1 = K + 1
C
C        FIND L = PIVOT INDEX
C
         L = ICAMAX(N-K+1,A(K,K),1) + K - 1
         IPVT(K) = L
C
CCC      Zero pivot implies this column already triangularized
C
         IF (CABS1(A(L,K)) .EQ. 0.0E0) GO TO 40
C
CCC         Interchange if necessary
C
            IF (L .EQ. K) GO TO 10
               T = A(L,K)
               A(L,K) = A(K,K)
               A(K,K) = T
   10       CONTINUE
C
CCC         Compute multipliers
C
            T = -CMPLX(1.0E0,0.0E0)/A(K,K)
            CALL CSCAL(N-K,T,A(K+1,K),1)
C
CCC         Row elimination with column indexing
C
            DO 30 J = KP1, N
               T = A(L,J)
               IF (L .EQ. K) GO TO 20
                  A(L,J) = A(K,J)
                  A(K,J) = T
   20          CONTINUE
               CALL CAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)
   30       CONTINUE
         GO TO 50
   40    CONTINUE
            INFO = K
   50    CONTINUE
   60 CONTINUE
   70 CONTINUE
      IPVT(N) = N
      IF (CABS1(A(N,N)) .EQ. 0.0E0) INFO = N
      RETURN
      END
C                                                                  C
```

```
C********************************************************************C
C                                                                    C
      SUBROUTINE CGESL(A,LDA,N,IPVT,B,JOB)
C                                                                    C
C********************************************************************C
C                                                                    C
C NAASA  2.1.044 CGESL    FTN-A 05-02-78     THE UNIV OF MICH COMP CTR
C
      INTEGER LDA,N,IPVT(1),JOB
      COMPLEX A(LDA,1),B(1)
C
C     CGESL SOLVES THE COMPLEX SYSTEM
C     A * X = B  OR  CTRANS(A)  * X = B
C     USING THE FACTORS COMPUTED BY CGECO OR CGEFA.
C
C     ON ENTRY
C
C        A       COMPLEX(LDA, N)
C                THE OUTPUT FROM CGECO OR CGEFA.
C
C        LDA     INTEGER
C                THE LEADING DIMENSION OF THE ARRAY  A .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C        IPVT    INTEGER(N)
C                THE PIVOT VECTOR FROM CGECO OR CGEFA.
C
C        B       COMPLEX(N)
C                THE RIGHT HAND SIDE VECTOR.
C
C        JOB     INTEGER
C                = 0         TO SOLVE  A*X = B ,
C                = NONZERO   TO SOLVE  CTRANS(A)*X = B  WHERE
C                            CTRANS(A)  IS THE CONJUGATE TRANSPOSE.
C
C     ON RETURN
C
C        B       THE SOLUTION VECTOR  X .
C
C     ERROR CONDITION
C
C        A DIVISION BY ZERO WILL OCCUR IF THE INPUT FACTOR CONTAINS A
C        ZERO ON THE DIAGONAL.  TECHNICALLY THIS INDICATES SINGULARITY
C        BUT IT IS OFTEN CAUSED BY IMPROPER ARGUMENTS OR IMPROPER
C        SETTING OF LDA .  IT WILL NOT OCCUR IF THE SUBROUTINES ARE
C        CALLED CORRECTLY AND IF CGECO HAS SET RCOND .GT. 0.0
C        OR CGEFA HAS SET INFO .EQ. 0 .
C
C     TO COMPUTE  INVERSE(A) * C  WHERE  C  IS A MATRIX
C     WITH  P  COLUMNS
C           CALL CGECO(A,LDA,N,IPVT,RCOND,Z)
C           IF (RCOND IS TOO SMALL) GO TO ...
C           DO 10 J = 1, P
C              CALL CGESL(A,LDA,N,IPVT,C(1,J),0)
C        10 CONTINUE
C
C     LINPACK. THIS VERSION DATED 07/14/77 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LABS.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS CAXPY,CDOTC
C     FORTRAN CONJG
C
C     INTERNAL VARIABLES
C
      COMPLEX CDOTC,T
      INTEGER K,KB,L,NM1
C
      NM1 = N - 1
      IF (JOB .NE. 0) GO TO 50
C
C        JOB = 0 , SOLVE  A * X = B
C        FIRST SOLVE  L*Y = B
C
         IF (NM1 .LT. 1) GO TO 30
         DO 20 K = 1, NM1
            L = IPVT(K)
            T = B(L)
            IF (L .EQ. K) GO TO 10
               B(L) = B(K)
               B(K) = T
   10       CONTINUE
            CALL CAXPY(N-K,T,A(K+1,K),1,B(K+1),1)
   20    CONTINUE
   30    CONTINUE
C
C        NOW SOLVE  U*X = Y
C
         DO 40 KB = 1, N
            K = N + 1 - KB
            B(K) = B(K)/A(K,K)
            T = -B(K)
            CALL CAXPY(K-1,T,A(1,K),1,B(1),1)
```

```
   40    CONTINUE
      GO TO 100
   50 CONTINUE
C
C        JOB = NONZERO, SOLVE  CTRANS(A) * X = B
C        FIRST SOLVE  CTRANS(U)*Y = B
C
         DO 60 K = 1, N
            T = CDOTC(K-1,A(1,K),1,B(1),1)
            B(K) = (B(K) - T)/CONJG(A(K,K))
   60    CONTINUE
C
C        NOW SOLVE CTRANS(L)*X = Y
C
         IF (NM1 .LT. 1) GO TO 90
         DO 80 KB = 1, NM1
            K = N - KB
            B(K) = B(K) + CDOTC(N-K,A(K+1,K),1,B(K+1),1)
            L = IPVT(K)
            IF (L .EQ. K) GO TO 70
               T = B(L)
               B(L) = B(K)
               B(K) = T
   70       CONTINUE
   80    CONTINUE
   90    CONTINUE
  100 CONTINUE
      RETURN
      END
C                                                                      C
C**********************************************************************C
C                                                                      C
      SUBROUTINE CAXPY(N,CA,CX,INCX,CY,INCY)
C                                                                      C
C**********************************************************************C
C                                                                      C
C NAASA  1.1.014 CAXPY     FTN-A 05-02-78     THE UNIV OF MICH COMP CTR
C
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.
C     JACK DONGARRA, LINPACK, 6/17/77.
C
      COMPLEX CX(1),CY(1),CA
      INTEGER I,INCX,INCY,IX,IY,N
C
      IF(N.LE.0)RETURN
      IF (ABS(REAL(CA)) + ABS(AIMAG(CA)) .EQ. 0.0 ) RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GOTO 20
C
CCC      Code for unequal increments or equal increments
CCC      Not equal to 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        CY(IY) = CY(IY) + CA*CX(IX)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
CCC      Code for both increments equal to 1
C
   20 DO 30 I = 1,N
        CY(I) = CY(I) + CA*CX(I)
   30 CONTINUE
      RETURN
      END
C                                                                      C
C**********************************************************************C
      COMPLEX FUNCTION CDOTC(N,CX,INCX,CY,INCY)
C**********************************************************************C
C                                                                      C
C NAASA  1.1.012 CDOTC     FTN-A 05-02-78     THE UNIV OF MICH COMP CTR
C
C     FORMS THE DOT PRODUCT OF TWO VECTORS, CONJUGATING THE FIRST
C     VECTOR.
C     JACK DONGARRA, LINPACK,  6/17/77.
C
      COMPLEX CX(1),CY(1),CTEMP
      INTEGER I,INCX,INCY,IX,IY,N
C
      CTEMP = (0.0,0.0)
      CDOTC = (0.0,0.0)
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GOTO 20
C
CCC      Code for unequal increments or equal increments
CCC      Not equal to 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
```

```
         CTEMP = CTEMP + CONJG(CX(IX))*CY(IY)
         IX = IX + INCX
         IY = IY + INCY
   10 CONTINUE
      CDOTC = CTEMP
      RETURN
C
CCC       Code for both increments equal to 1
C
   20 DO 30 I = 1,N
         CTEMP = CTEMP + CONJG(CX(I))*CY(I)
   30 CONTINUE
      CDOTC = CTEMP
      RETURN
      END
C                                                                    C
C********************************************************************C
C                                                                    C
      SUBROUTINE  CSCAL(N,CA,CX,INCX)
C                                                                    C
C********************************************************************C
C                                                                    C
C NAASA  1.1.019 CSCAL    FTN-A 05-02-78      THE UNIV OF MICH COMP CTR
C
C     SCALES A VECTOR BY A CONSTANT.
C     JACK DONGARRA, LINPACK,  6/17/77.
C
      COMPLEX CA,CX(1)
      INTEGER I,INCX,N,NINCX
C
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GOTO 20
C
CCC       Code for increment not equal to 1
C
      NINCX = N*INCX
      DO 10 I = 1,NINCX,INCX
        CX(I) = CA*CX(I)
   10 CONTINUE
      RETURN
C
CCC       Code for increment equal to 1
C
   20 DO 30 I = 1,N
        CX(I) = CA*CX(I)
   30 CONTINUE
      RETURN
      END
C                                                                    C
C********************************************************************C
C                                                                    C
      SUBROUTINE  CSSCAL(N,SA,CX,INCX)
C                                                                    C
C********************************************************************C
C                                                                    C
C NAASA  1.1.018 CSSCAL   FTN-A 05-02-78      THE UNIV OF MICH COMP CTR
C
C     SCALES A COMPLEX VECTOR BY A REAL CONSTANT.
C     JACK DONGARRA, LINPACK, 6/17/77.
C
      COMPLEX CX(1)
      REAL SA
      INTEGER I,INCX,N,NINCX
C
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GOTO 20
C
CCC       Code for increment not equal to 1
C
      NINCX = N*INCX
      DO 10 I = 1,NINCX,INCX
        CX(I) = CMPLX(SA*REAL(CX(I)),SA*AIMAG(CX(I)))
   10 CONTINUE
      RETURN
C
CCC       Code for increment equal to 1
C
   20 DO 30 I = 1,N
        CX(I) = CMPLX(SA*REAL(CX(I)),SA*AIMAG(CX(I)))
   30 CONTINUE
      RETURN
      END
C                                                                    C
C********************************************************************C
      INTEGER FUNCTION ICAMAX(N,CX,INCX)
C********************************************************************C
C                                                                    C
C NAASA  1.1.021 ICAMAX   FTN-A 05-02-78      THE UNIV OF MICH COMP CTR
C
C     FINDS THE INDEX OF ELEMENT HAVING MAX. ABSOLUTE VALUE.
C     JACK DONGARRA, LINPACK, 6/17/77.
C
      COMPLEX CX(1)
      REAL SMAX
      INTEGER I,INCX,IX,N
      COMPLEX ZDUM
```

```
      REAL CABS1
      CABS1(ZDUM) = ABS(REAL(ZDUM)) + ABS(AIMAG(ZDUM))
C
      ICAMAX = 1
      IF(N.LE.1)RETURN
      IF(INCX.EQ.1)GOTO 20
C
CCC      Code for increment not equal to 1
C
      IX = 1
      SMAX = CABS1(CX(1))
      IX = IX + INCX
      DO 10 I = 2,N
         IF(CABS1(CX(IX)).LE.SMAX) GO TO 5
         ICAMAX = I
         SMAX = CABS1(CX(IX))
    5    IX = IX + INCX
   10 CONTINUE
      RETURN
C
CCC      Code for increment equal to 1
C
   20 SMAX = CABS1(CX(1))
      DO 30 I = 2,N
         IF(CABS1(CX(I)).LE.SMAX) GO TO 30
         ICAMAX = I
         SMAX = CABS1(CX(I))
   30 CONTINUE
      RETURN
      END
C                                                                    C
C********************************************************************C
      REAL FUNCTION SCASUM(N,CX,INCX)
C********************************************************************C
C                                                                    C
C NAASA  1.1.010 SCASUM   FTN-A 05-02-78    THE UNIV OF MICH COMP CTR
C
C     TAKES THE SUM OF THE ABSOLUTE VALUES OF A COMPLEX VECTOR AND
C     RETURNS A SINGLE PRECISION RESULT.
C     JACK DONGARRA, LINPACK, 6/17/77.
C
      COMPLEX CX(1)
      REAL STEMP
      INTEGER I,INCX,N,NINCX
C
      SCASUM = 0.0E0
      STEMP = 0.0E0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1)GOTO 20
C
CCC      Code for increment not equal to 1
C
      NINCX = N*INCX
      DO 10 I = 1,NINCX,INCX
        STEMP = STEMP + ABS(REAL(CX(I))) + ABS(AIMAG(CX(I)))
   10 CONTINUE
      SCASUM = STEMP
      RETURN
C
CCC      Code for increment equal to 1
C
   20 DO 30 I = 1,N
        STEMP = STEMP + ABS(REAL(CX(I))) + ABS(AIMAG(CX(I)))
   30 CONTINUE
      SCASUM = STEMP
      RETURN
      END
```

Appendix B Program Listing for the Quasi-Analytical Solution

The quasi-analytical solution proposed by Senior and Volakis [1] as described in Section 3 was programmed for solution, as listed in the program ANAGAP.FTN below. The subroutines used by this program are listed in GAPSUB.FTN in Appendix A.

As with the program for the exact solution GAPSCAT, the user is prompted for the following: the polarization of the incident field, angle of incidence, angle of far field observation, the relative permittivity ε_r of the gap cavity, the shape of the gap, and the number of iterations for calculating the far field amplitude versus gap depth. The choice of shapes and dimensions requested for the gap are according to Figure 2.

The input impedance of the gap as a parallel plate waveguide is calculated according to the specified shape. For the L- and T-shaped gaps, (35) is used, given the other necessary expressions as contained in Section 3. The input impedance of the rectangular gap is given by (39), and for the V-shaped gap, (40) is used. These expressions are for the H-polarization case. As mentioned previously, for the E-polarization case, the propagation constant k_1 is replaced by ikp and the characteristic impedance Z_1 by $-iZ\mu_r/p$, where p is given by (41). The desired effective surface impedance η is then calculated according to (33).

For H-polarization, the far field amplitude P_H is calculated from (22). P_H is a function of $K_H(a)$ given by (30). The argument a is a function of the effective surface impedance as given in (21). For the E-polarization, the far field amplitude P_E is calculated from (27), where $K_E(b)$ is given by (31). The argument b is a function of the effective surface impedance as given in (26).

```
c
c      ANAGAP.FTN
c
c
c        This FORTRAN program computes the far field scattering due
c        to a narrow gap of specified shape in an infinite ground
c        plane.  The far field amplitude is calculated given the
c        input impedance of the gap, calculated from it's equivalent
c        transmission line model.
c
c        INPUT    The user is prompted from the subroutine
c                 GAPROM for the polarization and angle of the
c                 incident field, angle of far field observation,
c                 relative permittivity of gap filling, shape and
c                 dimensions of gap, segment size, and number of
c                 iterations with respect to gap depth.
c
c        OUTPUT FILES
c                 GAPDAT  Contains input data.
c                 IMPDAT  Effective surface impedance of the gap.
c                 AMPDAT  Contains the magnitude of the far field.
c                 PHADAT  Contains the phase of the far field.
c
c        SUBROUTINES
c                 HANKZ1  Computes the Hankel functions of the first
c                         kind of orders zero and one.
c                 CHANK   Computes the Hankel functions of the first
c                         kind of orders zero and one given a
c                         complex argument.
c                 MODBESS Computes the modified Bessel functions of
c                         the first kind of orders zero and one.
c
c        FUNCTION
c                 CTAN    Calculates the tangent of a complex
c                         argument.
c
c
      integer EorH,N,noS,gN,szN(50)
      real pi,k,phi,phio,w,d,maxC,q(50,2)
      real dStp(50),wStp(50),I0,I1
      complex czero,ci,ctemp,er,ur,k1,kc,carg,ctan,eta
      complex Zi,Z1,Zc,ZL,X,B1,B2,H0,H1,a,b,Ke,Kh,Ao,Psca
      logical Epol,Lossy
      common /prompts/ EorH,phio,phi,er,igap,wStp,dStp,w,d,noS,
     &                 q,maxC,noIter

  1   format(i1)
  2   format(i5)
  3   format(g16.8)
  4   format(a1)
  5   format(i3)
  6   format(2g16.8)

      open(2,file='impdat')
      open(3,file='ampdat')
      open(4,file='phadat')

c...Declaring constant values
      czero=cmplx(0.0,0.0)
      ci=cmplx(0.0,1.)
      pi=4.0*atan(1.0)
      k=2*pi
      Eo=1.0
      Ho=1.0
      Zo=sqrt(4.e-07*pi/8.854e-12)
      Yo=1./Zo
      ur=cmplx(1.,0.0)
      gam=0.5772157
      iprg=2

c...Setting default values
 10   EorH=1
      phio=90.0
      phi=90.0
      er=cmplx(1.,0.0)
      w=0.15
      d=0.5
      noS=3
      maxC=0.01
      noIter=30
      adj=0.00001
      Epol=.false.
      Lossy=.false.
      side=.true.

c...Prompting user for input data
 15   call gaprom(iprg)

      if(igap .eq. 5) GOTO 15
      if(EorH .eq. 1) Epol=.true.
      phio=phio*pi/180.0
      phi=phi*pi/180.0
      drat=dStp(1)/d
      if(aImag(er) .ne. 0.0) Lossy=.true.

      dmin=0.05
```

```
      if(Epol) dmin=0.025
      dmax=d
      if(noIter .ne. 1)then
        dstep=(dmax-dmin)/(noIter-1)
        d=dmin
      endif

      DO 700 iter=1,noIter
c*****************************************************************
c*********************   Gap Impedance    ************************
c*****************************************************************

c...Complex propagation constant k1 and characteristic impedance
c   Z1 of the T-line model
        if(Epol)then
          k1=ci*k*csqrt((1./2/w)**2-er*ur)
          Z1=-ci*Zo*ur/csqrt((1./2/w)**2-er*ur)
        else
          Z1=Zo*csqrt(ur/er)
          k1=k*csqrt(ur*er)
        endif
          if(igap .eq. 1)then
c        RECTANGULAR
            ETA=-ci*Z1*ctan(k1*d)
          else if(igap .eq. 2 .or. igap .eq. 4)then
            w=wStp(1)
            w2=wStp(2)
            w3=wStp(3)
            d1=d*drat
            d2=d*(1-drat)
c          Propagation constant and characteristic impedance of
c          the arms of the T- or L-shaped gaps
            if(Epol)then
              kc=ci*k*csqrt((1./2/d2)**2-er*ur)
              Zc=-ci*Zo*d2*ur/csqrt((1./2/d2)**2-er*ur)
            else
              Zc=Zo*d2*csqrt(ur/er)
              kc=k*csqrt(ur*er)
            endif
            if(igap .eq. 4)then
c        L-SHAPED
              Zi=-ci*Zc*ctan(kc*w2)
              X=k*Zc*w1
              B1=k/Zc*d2*(d2/(d2+w1))*(1.-2./pi*log(2.))
              B2=k/Zc*d2*(w1/(d2+w1))*(1.-2./pi*log(2.))
              ZL=(Zi-ci*X*(1-ci*B2*Zi))
     &            /((1-B1*X)*(1-ci*B2*Zi)-ci*B1*Zi)
            else
c        T-SHAPED
              Zi=-ci*Zc*(ctan(kc*w2)+ctan(kc*w3))
              X=2*k*Zc*w1
              B1=k/Zc*d2*(d2/(d2+w1))*0.7822
              ZL=(Zi-ci*X)/(1-ci*B1*(Zi-ci*X))
            endif
            ETA=Z1*(ZL-ci*Z1*w1*ctan(k1*d1))
     &            /(Z1*w1-ci*ZL*ctan(k1*d1))
          else if(igap .eq. 3)then
c        TRIANGULAR
            if(Lossy)then
               carg=k1*d
               call cHank(carg,3,H0,H1)
               carg=1.
            else
               if(Epol)then
                  rarg=Real(k1/ci)*d
                  call ModBess(rarg,I0,I1)
                  H0=I0
                  H1=I1
                  carg=ci
               else
                  rarg=Real(k1)*d
                  call Hankz1(rarg,2,H0,H1)
                  carg=1.
               endif
            endif
            ETA=-ci*Z1*Real(H1)/Real(H0)*carg
          endif
c           write(2,*) d,cabs(ETA)


c*****************************************************************
c*****************   Far Field Amplitude    *********************
c*****************************************************************
        if(Epol)then
           b=-ci*k*w/2*Zo/ETA
           Ke=0.62/(b+1.15)*(b+4.08)*(b+7.26)*(b+10.37)
     &         *(b+13.43)*(b+16.46)
     &        /((b+4.27)*(b+7.37)*(b+10.45)*(b+13.49)
     &         *(b+16.5))
           Psca=-ci*pi/4*(k*w)**2*sin(phio)*sin(phi)*Ke
        else
           a=ci*2./k/w*Zo/eta
           Kh=-1/(pi/2*a+0.1+log(2.))
           Ao=log(k*w/4)+gam-ci*pi/2
           Psca=ci*pi*Kh/(1+Ao*Kh)
        endif
```

```
c...Outputting the far field magnitude and phase
        print *,' d = ',d
        print *,'      Analytical: |Psca| = ',cabs(Psca),
     & '  arg Psca = ',180/pi*(atan2(aImag(Psca),Real(Psca)))
        write(3,*) d,cabs(Psca)
        write(4,*) d,180/pi*(atan2(aImag(Psca),Real(Psca)))

        d=d+dstep
700   continue

        print *,' Again (1=yes) ? '
        read(*,1)ians
        if(ians .eq. 1) GOTO 10

800   call exit
      END
```