SEMIANNUAL STATUS REPORT
NASA GRANT NAG-1-566

SOUND PROPAGATION OVER UNEVEN GROUND AND IRREGULAR TOPOGRAPHY

By

Allan D. Pierce, Geoffrey L. Main, James A. Kearns,
Daniel R. Benator, and James R. Parish, Jr.

Submitted to

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

NASA Technical Officer:
John A. Preissar
Mail Stop 460A

February 1986

GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM OF GEORGIA
SCHOOL OF MECHA ICAL ENGINEERING
ATLANTA, GEORGIA 30332

1986

SEMIANNUAL STATUS REPORT
NASA GRANT NAG-1-566

SOUND PROPAGATION OVER UNEVEN GROUND AND IRREGULAR TOPOGRAPHY

by

Allan D. Pierce, Geoffrey L. Main, James A. Kearns,
Daniel R. Benator, and James R. Parish, Jr.

School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Submitted to

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

NASA Technical Officer:
John S. Preisser
Mail Stop 460A

February, 1986

TABLE OF CONTENTS

List of Figures

INTRODUCTION

The goal of this research is to develop theoretical, computational, and experimental techniques for predicting the effects of irregular topography on long range sound propagation in the atmosphere. Irregular topography here is understood to imply a ground surface that (1) is not idealizable as being perfectly flat or (2) that is not idealizable as having a constant specific acoustic impedance. The interest of this study focuses on circumstances where the propagation is similar to what might be expected for noise from low-altitude air vehicles flying over suburban or rural terrain, such that rays from the source arrive at angles close to grazing incidence.

PERSONNEL

In addition to the principal investigators, A. D. Pierce and G. L. Main, a graduate student, James Kearns, and two senior undergraduate students in mechanical engineering, Daniel Benator and James Parish, are presently working on the project. The students have up until now been primarily engaged in the construction of the experimental facility, in the construction of equipment, in the procurement of equipment and instrumentation, and in the testing of the components of the facility. Exploratory experiments are now beginning, with all three students participating. The theoretical work has up until now been carried out mostly by Pierce and Main, with tutorial sessions underway to develop Kearn's participation in this phase of the research.

All of the personnel concerned with the project visited NASA Langley Research Center in November 1985 and discussed complementary NASA and Georgia Tech research activities with the NASA technical officer, Dr. John Preisser, and his colleagues.

LABORATORY FACILITY AND INSTRUMENTATION

The principal activity during the first year of the subject grant has been the construction of a laboratory facility (Fig. 1) that will be used in subsequent experiments. Major components of the laboratory are an acoustic source, a model topographical surface with which the acoustic signal interacts, and a data acquisition and analysis system. The room housing the laboratory (Fig. 2) is a standard university small laboratory room of dimensions 8.2 m by 6.1 m by 4.3 m. The walls are cinderblock, the floor is tiled, and some walls are lined with shelves, so in no sense does this space approach the ideal of an anechoic chamber. However, the room, dubbed the "Atmospheric Sound Propagation Facility" within Georgia Tech, is solely dedicated to this project. The investigators have been developing the instrumentation system to be such that the echoes from walls, floor, and ceiling can be gated out in time.

Base tables for topographic experiments

Four tables were made to be used for the scale model experiments. Each table is 1.2 meter wide by 2.4 meter long and 0.9 meter high. The tables were constructed so that they could be bolted together, forming one table 4.9 meter long by 2.4 meter wide. The table top is CDX plywood, 2 cm thick. The table frame is constructed of two-by-six (5 cm by 15 cm) yellow pine grade #1 planks; the table legs are constructed of four-by-four (10 cm by 10 cm) yellow pine grade #2 beams. The fasterners holding the table together are machine bolts and wood screws, so the table assembly is fully portable.

The table frame was made by running two 2.4 meter length two-by-sixes parallel to each other, 1.2 meter apart. Then five equally spaced two-by-sixes were mounted in between these first two. Next, a shelf was cut into the four-by-fours, so that they would fit into the corners of the frame and still leave some of the frame resting on the shelf. Each four-by-four was bolted into the frame with three machine bolts for extra strength. Then the plywood was placed on the frame and secured with wood screws.

In the experiments currently beginning, the four tables are bolted together and are being used in the 4.9 m by 2.4 m configuration; the term 'table' implies this configuration in the remainder of this report.

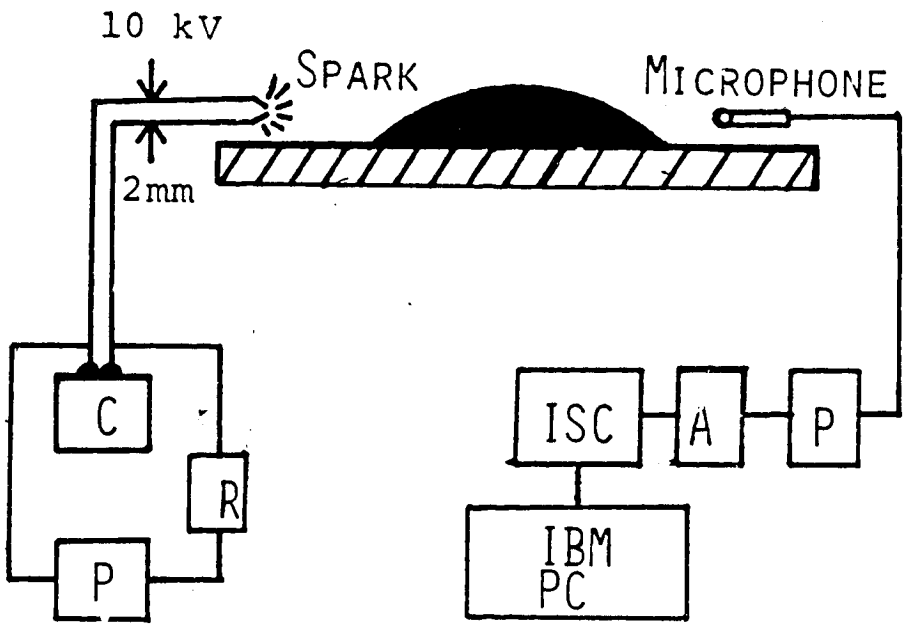

Figure 1. General diagram of experimental configuration for studying sound propagation over model topographical ridge. Here C is capacitor, R is resistor, P is power supply, IBM PC is IBM personal computer, ISC is RC Electronics A/D conversion instrumentation, A is amplifier, and P is preamplifier

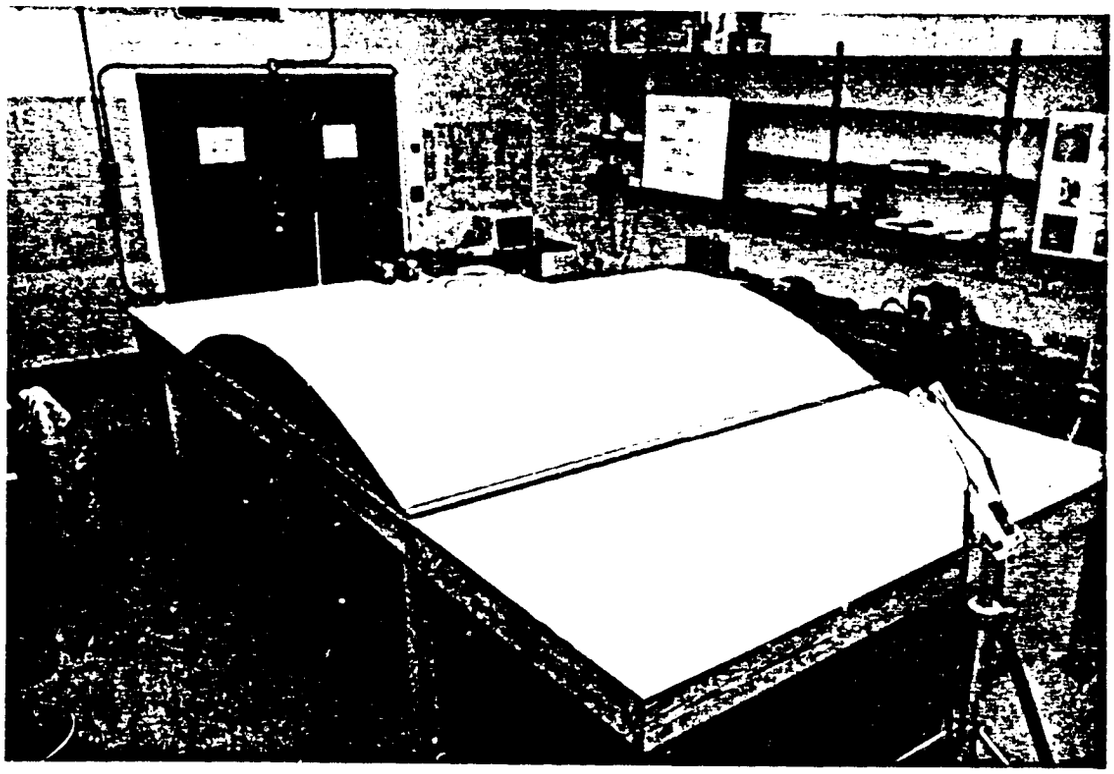

Figure 2. Photogr of interior of laboratory room used in the study, showing most of the relevan quipment and the experimental facility.

Laboratory scale model topographical ridge

A curved surface (Fig. 3) was constructed to be mounted on the table and used as a laboratory scale model of a topographical ridge. The contour of the surface has the shape of an arc of a circle.

Four basic templates (or ribs) for the ridge were cut from a piece of 1.2 meter by 2.4 meter exterior plywood, 2 cm thick. The top edge of each template was an arc of a circle; the bottom edge was the chord of a circle. The chord was 2.4 meter long and the radius of the circle was such that the maximum height of the arc relative to the chord was 29 cm. Thus the radius of curvature of the arc was approximately 2.5 meter.

Identical halves of the curved surface superstructure to the table assemblage were constructed as follows. For each half, a pair of templates were each secured to a 1.24 m by 2.4 m base board of CDX plywood (2 cm thickness) by nailing a strip of 3.8 cm by 3.8 cm yellow pine to each side of the template and then nailing the strips to the CDX plywood. The curved topographical surface was then achieved by bending plywood sheets, 0.5 cm thick and 1.24 cm wide, over the template arcs and then nailing the plywood to the arcs.

Spark Generator

A spark gap (Fig. 4 and Fig. 5) was constructed to serve as an impulsive acoustic source. Sound is generated when a sudden current surge occurs across a 2-3 mm air gap. As indicated in Fig. 6, a 10 kV power supply provides charge at the rated voltage to a 1 uF capacitor. A 1 MΩ resistor is in series with the capacitor and the power supply; the voltage across the spark gap is virtually the same as that across the capacitor plates because of the negligible electrical resistance of the 0.5 cm diameter welding cables that carry current to and from the gap. The gap is between two 0.5 cm diameter tungsten electrodes that form the terminal points of the welding cables. The electrodes are held in position by a two-pronged plexiglass fork which is mounted on a tripod stand (Fig. 7). A rotary grinder is used to shape and polish the electrode points. This spark gap generates an acoustical signal (Fig. 8) whose spectral content is dominated by frequencies of the order of 10 kHz and whose peak amplitude at a distance of the order of 1 m corresponds to roughly 110 dB.

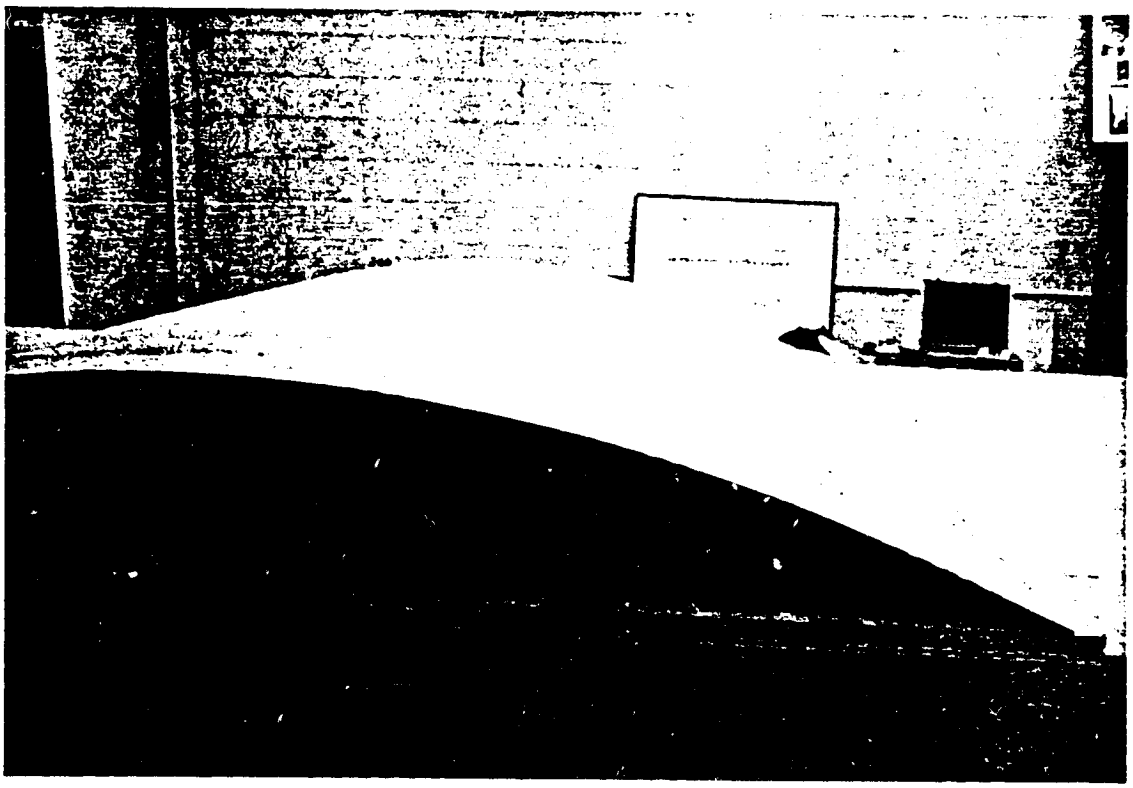

Figure 3. Close-up photograph of oblique side view of curved ridge resting on table that was constructed for studying propagation effects of topographical ridges.

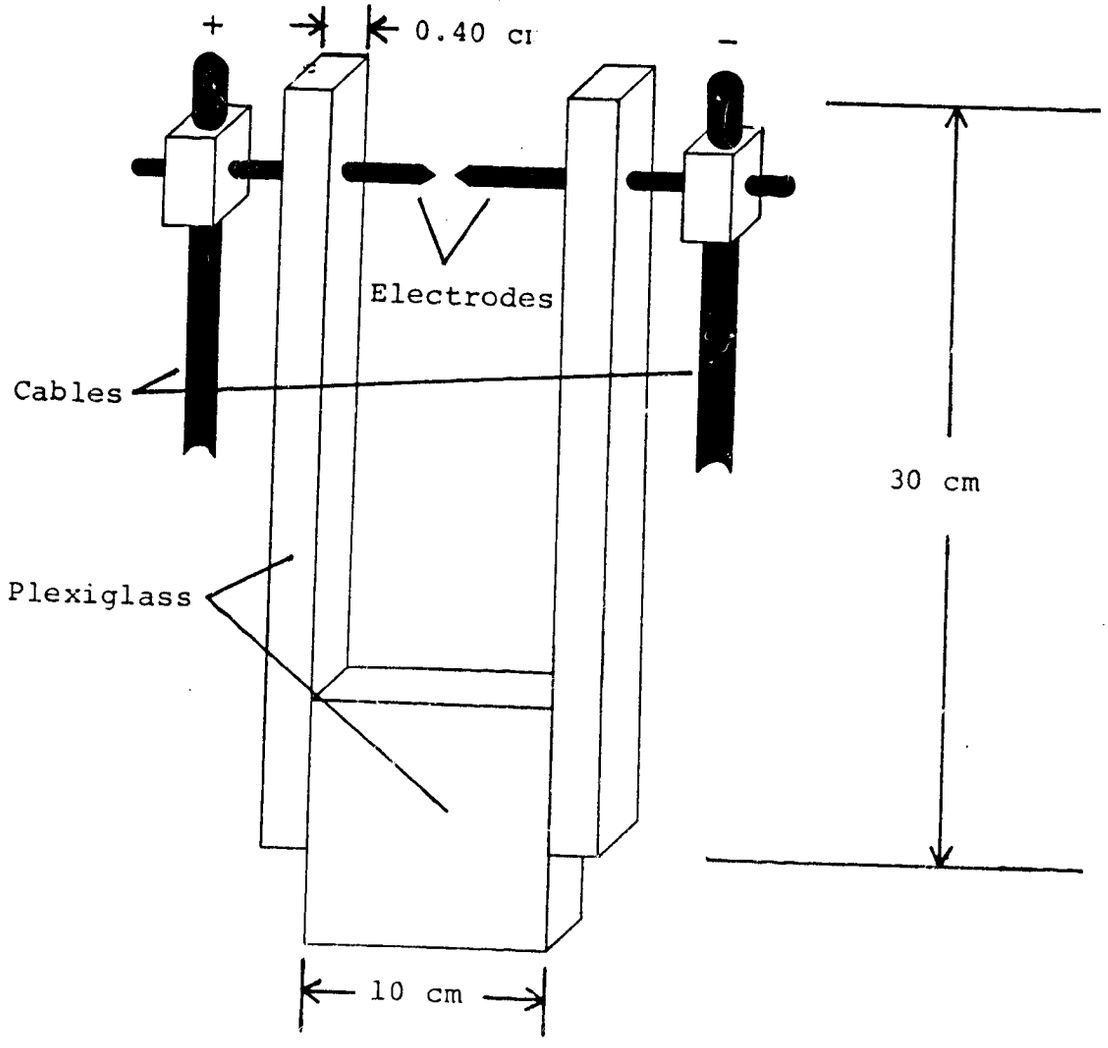

Figure 4. Design drawing of spark gap apparatus used in generation of transient acoustic pulses.

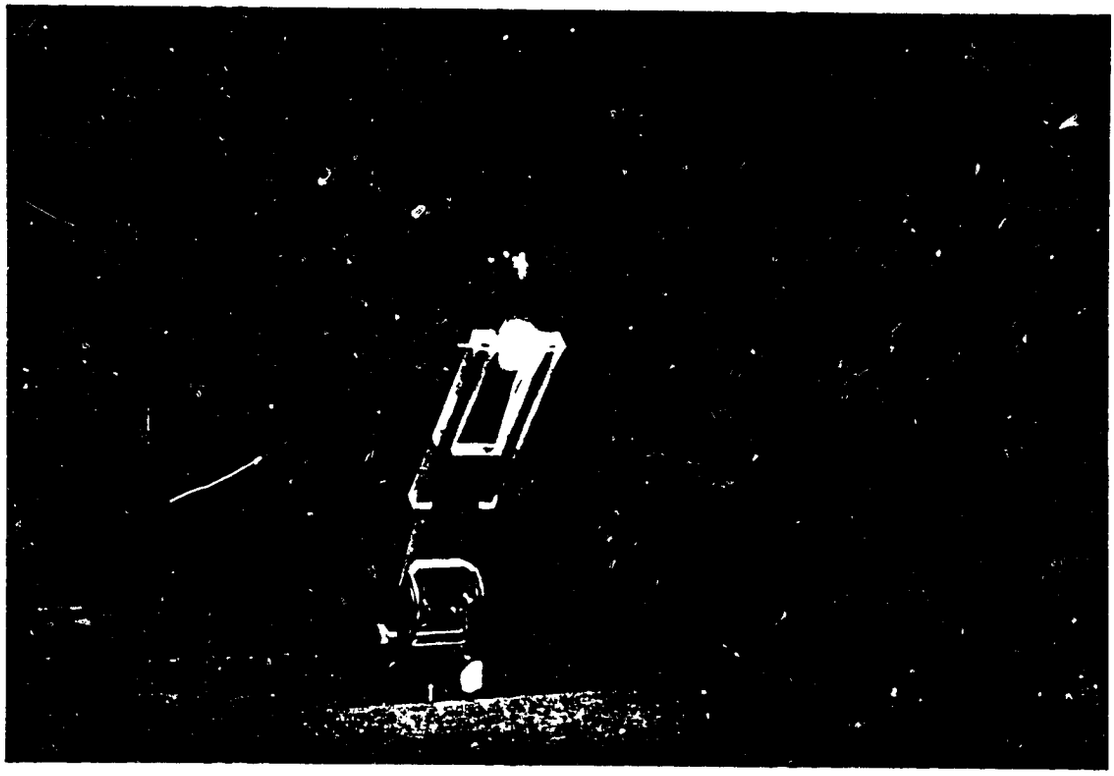

Figure 5. Photograph of spark jumping across gap between electrodes held in plexiglas frame.

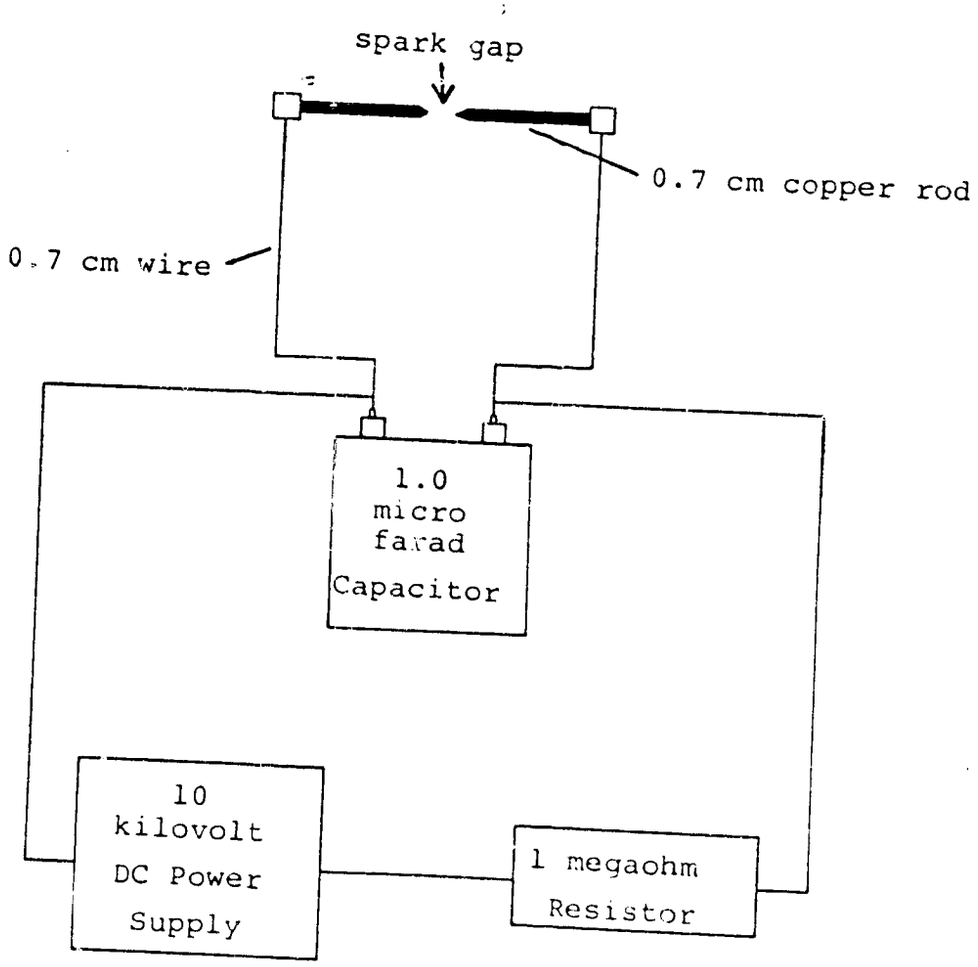

Figure 6. Diagram of spark generation apparatus. Electrodes available for use are either copper or tungsten.

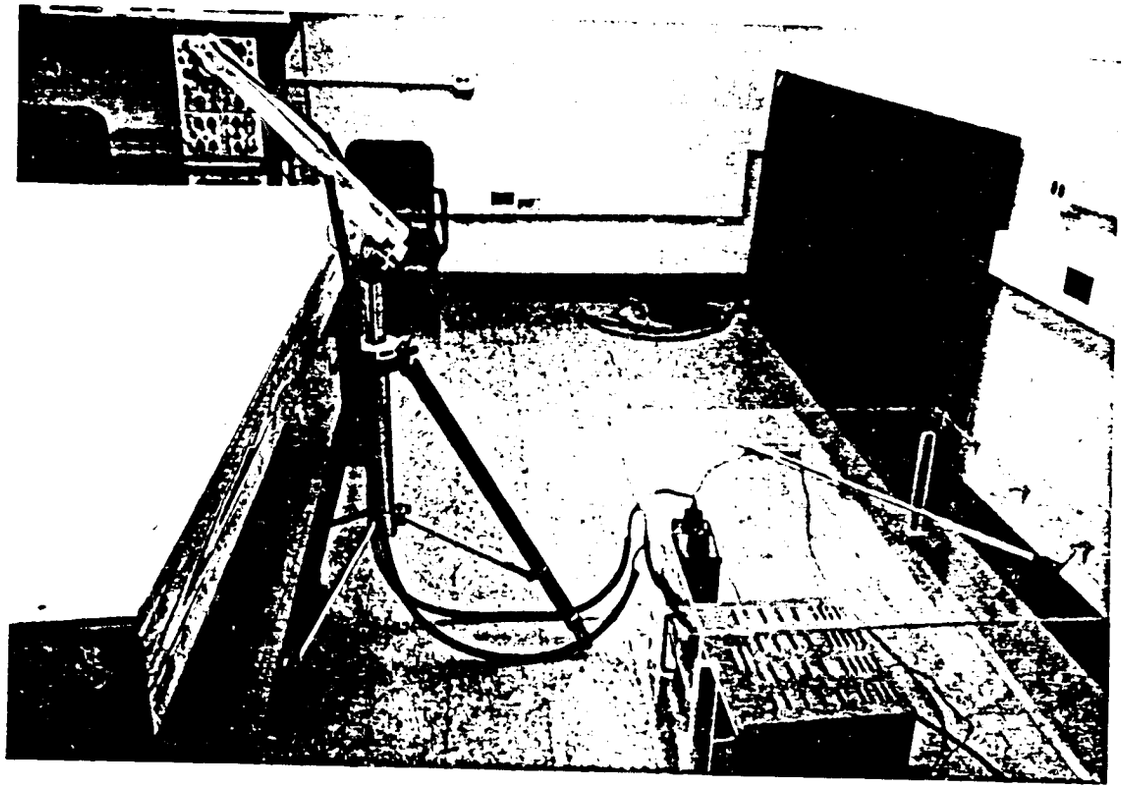

Figure 7. Photograph of spark generation apparatus, showing tripod holding the poexiglas frame with inserted electrodes. The power supply and capacitor are housed in a plexiglas box as a safety precaution.

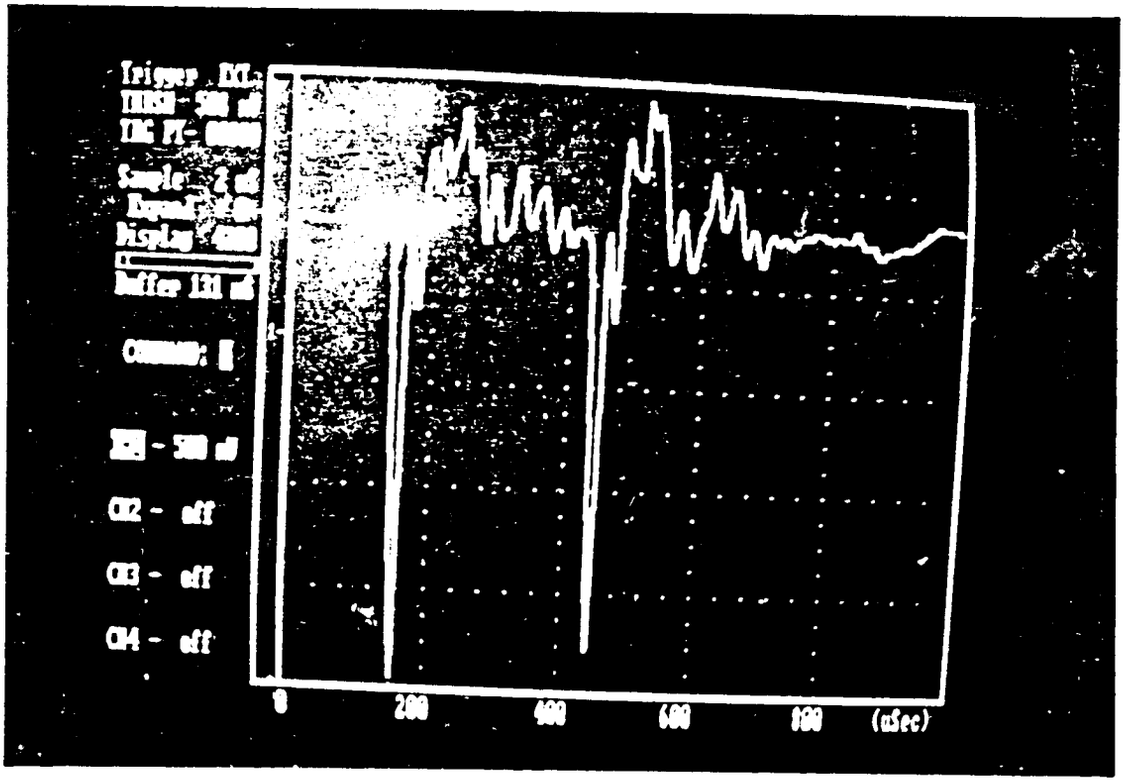

Figure 8. Typical oscilloscope trace on the monitor of an IBM PC. Trace corresponds to acoustic pressure transient of a spark discharge. Positive pressure is downward on the screen; the two large positive peaks correspond to direct wave and wave reflected from the table.

Data Acquisition and Analysis System

The data acquisition system is composed of microphones, amplifiers, an analog-to-digital converter, and an IBM personal computer (Fig. 9 and Fig. 10). The system is capable of gathering data at a rate of 500 kHz which can subsequently be processed by the PC. The acoustics laboratory VAX computer is in an adjoining room and is available for more extensive computations and storage. The amplifiers were designed and built expressly for this project; other system components were purchased. In addition, an apparatus is being designed and constructed to quickly and precisely position the microphones at arbitrary points in the field.

Bruel & Kjaer quarter-inch condenser microphones are used for making the necessary precision sound pressure measurements. The microphone sensitivity (ratio of induced open circuit voltage to external acoustic pressure) for these microphones is certified by the manufacturer be be virtually constant for frequencies up to 70 kHz, so we expect them to yield a relatively undistorted response to pulses predominantly composed of frequencies between 10 and 30 kHz. The microphones are linear in their response over a dynamic range of up to 180 dB with a sensitivity of 0.1 mV/ubar. A Bruel & Kjaer pre-amplifier and power supply are also part of the microphone assembly. The pre-amplifier has a very high inmput impedance and low parallel capacitance which are needed to maintain the flat frequency response. This high impedance is provided by a vacuum tube cathode follower at the input stage. The B & K 2801 power supply is used to provide voltage to the microphone and pre-amplifier.

For the circumstances of the contemplated experiments the microphone assemblage open circuit voltage is typically of the order of 50 mV. Because such voltages are two low to exploit the full dynamic range of the analog-to-digital conversion instrumentation, two low current amplifiers were constructed to magnify the voltage signal. Each amplifier contains a Motorola LF351N FET operational amplifier microchip, which has a high voltage slew rate (13 V/us) and a flat response over a wide range of frequencies. The design of the amplifier contains a non-inverting voltage amplifying circuit (Fig. 11). A variable gain is achieved by an array of feedback resistors controlled by an external multi-position switch. Frequency independent gains from unity up to 100 are possible. The amplifier is powered by two parallel series of 9 volt batteries. The transient voltage level is guaranteed by parallel 0.1 uF capacitors. The amplifier is enclosed by a aluminum box which is grounded to the batteries. The box serves as a shield against electromagnetic noise generated by the spark generator. Shielding considerations also motivated the use of a self-

contained power supply.

The amplified analog signal is converted to digital form by an integrated hardware and software system produced by RC Electronics Inc. and called "Computerscope IBC-16". This system consists of a 16 channel A/D board which is inserted into a slot within an IBM PC, an external instrument interface, and the scope driver software. The system is capable of sampling data at a maximum aggregate rate of 0.5 MHz over as many as 16 channels. This state-of-the-art RC system is relatively new to the market and differs from other commercially available data interfaces for personal computers in its high data accession rate, which is adequate for acoustical experiments at frequencies in the 10's of kilohertzes range; the 0.5 MHz sampling rate provides 50 data points per cycle for a 10 kHz signal. The system effectively transforms a personal computer into a low cost transient recorder or digital storage oscilloscope and should allow a greater flexibility in the digital processing of acoustical data. The system allows an input voltage signal with a peak to peak range of 20 volts centered at zero to be resolved to 12 bit accuracy, or equivalently to 1 part in 4000. The incoming transient signal is stored within a 64 kilobye memory buffer. Various modes of triggering are possible. In particular, an external channel is provided exclusively for triggering without occupying any memory space, although it is possible to trigger off of another channel or off of threshold levels of slope or amplitude. The Scope Driver software allows for flexible manipulation and display of the captured data. The display is similar to that of an oscilloscope. We anticipate that all of the necessary spectrum analysis and transfer function calculations can be carried out, subsequent to data capture, by the host IBM PC, but the acoustic group's VAX in the adjoining room is available for computations too involved or lengthy for the personal computer.

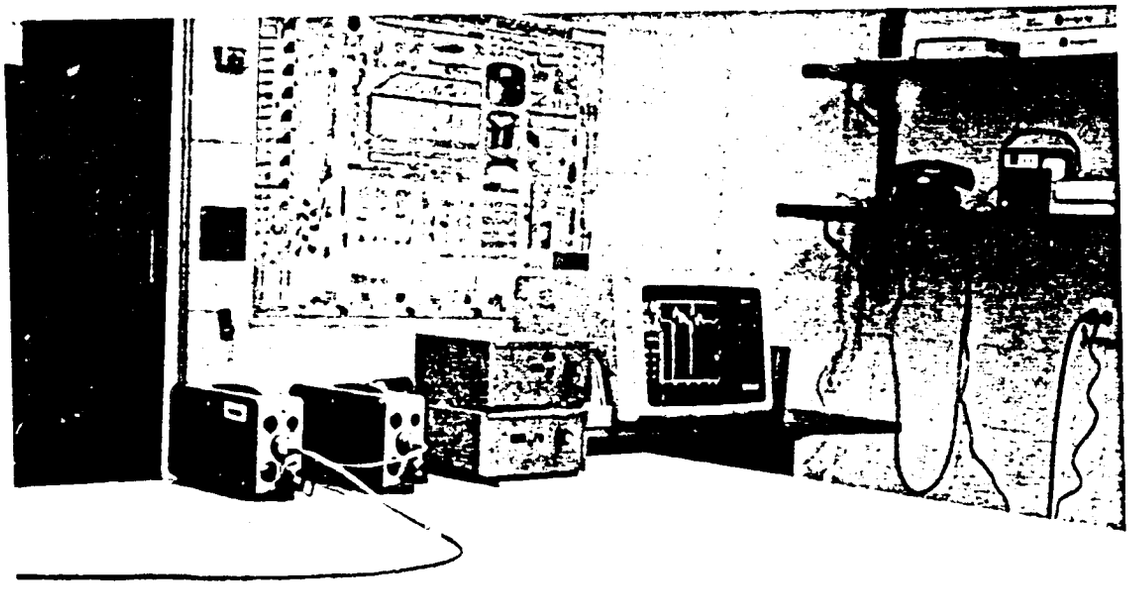

Figure 9. Photograph of part of the experimental equipment showing microphone power supplies, the amplifiers whose design and construction are described in the text, and the monitor of the IBM personal computer. A typical transient acoustic pressure trace from a single microphone can be seen on the monitor screen.

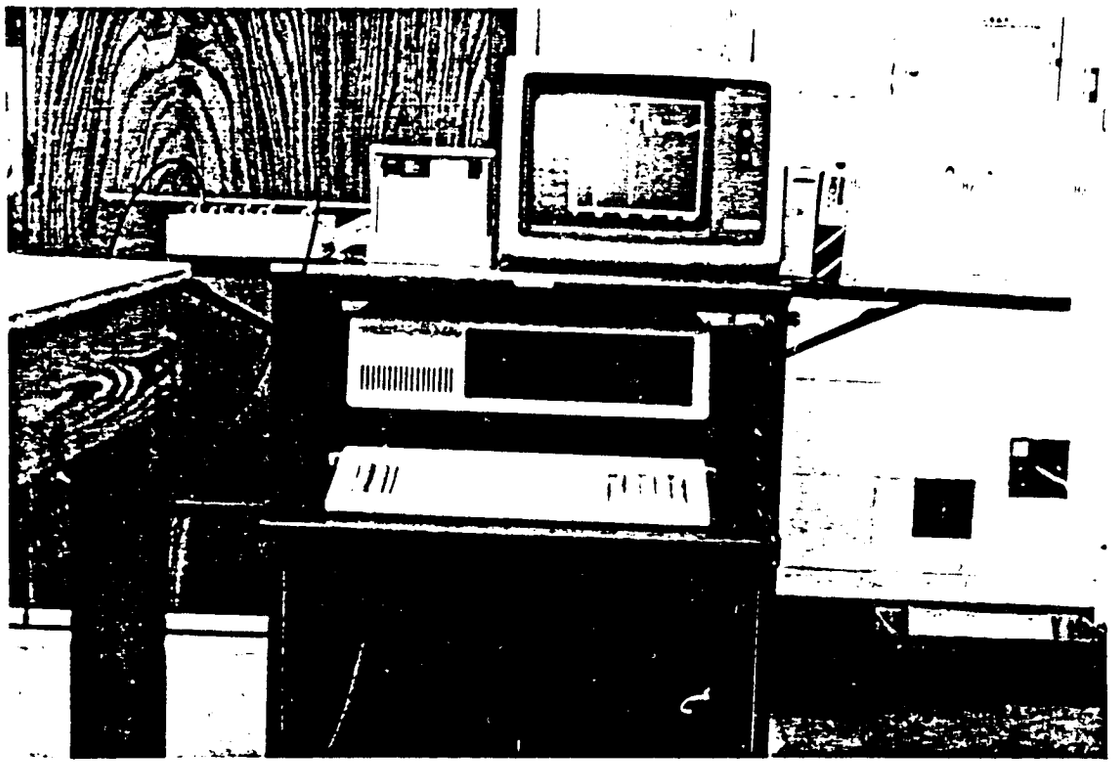

Figure 10. Photograph of the IBM personal computer with peripheral equipment which allows it to function as a digital storage oscilloscope or transient recorder. Corner of table facility of model topographical ridge can be seen at the far left. Cables from microphone assemblages lead to RC Electronics instrumentation interface, which in turn is connected to 16 channel high speed 12 bit A/D plug-in within the personal computer.

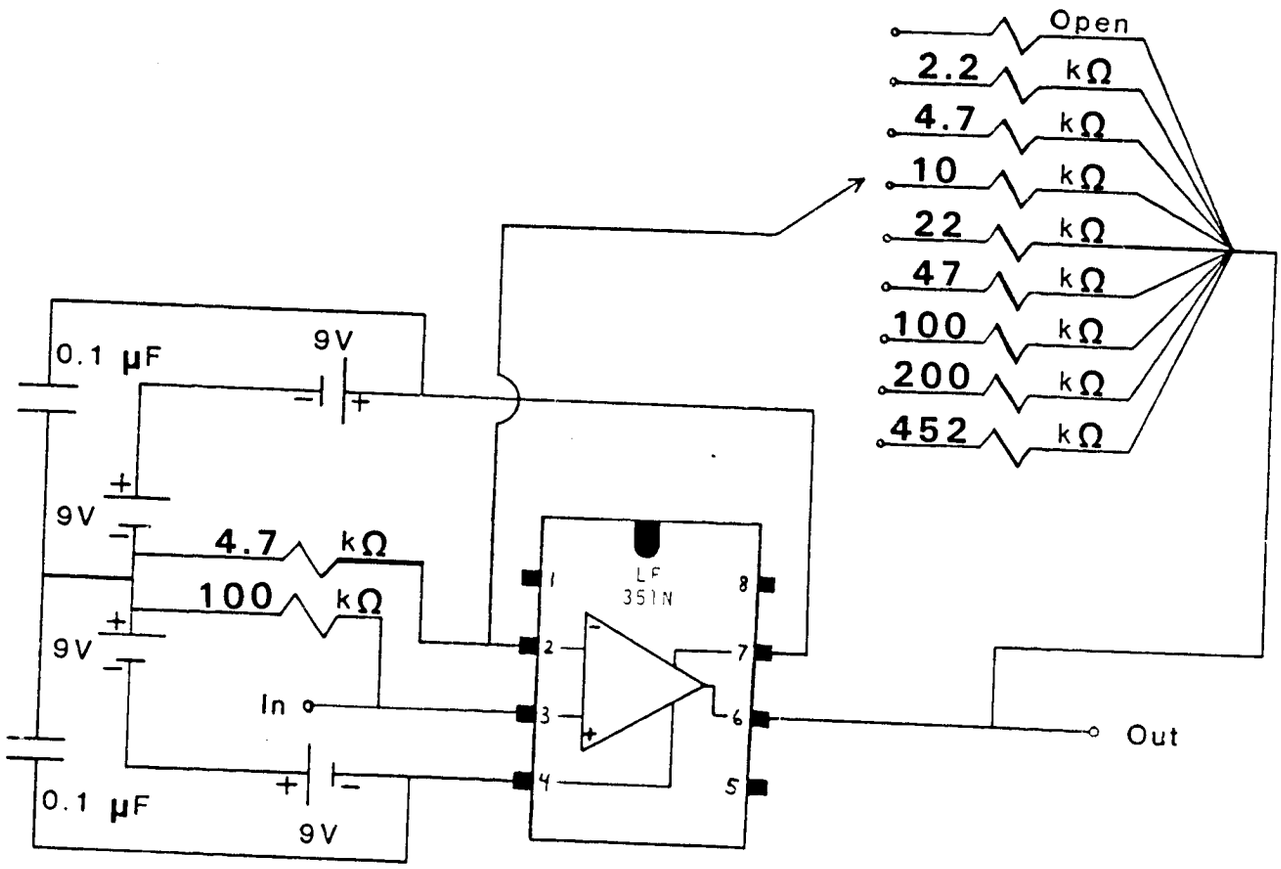

Figure 11. Circuit design for noninverting amplifiers that were constructed as interface between microphones and A/D conversion instrumentation.

ANALYTICAL STUDIES

Successful application of theoretical acoustics to outdoor propagation over undulating terrain is in principle possible, but presents challenges. The authors' considerations are presently limited to when the terrain is slowly varying over distances comparable to a wavelength; many realistic outdoor situations should be well-modelled without violation of such a restriction. The overall hope is that asymptotic and matching techniques can enable one to splice together mathematical models for intricate circumstances (such as multiple undulations) from those for simpler circumstances.

Diffraction by a single ridge of finite impedance

In the research program currently in progress, the understanding of diffraction by a single smooth ridge (Fig. 12) is a key element. Diffraction by a curved surface has a venerable and extensive literature, although much of it is specifically written for electromagnetic wave applications. There is need for a readily assimilable treatment of *acoustic diffraction* by curved surfaces of finite impedance that is *easily adaptable* to servitude as a building block for a broader theory for propagation over irregular terrain. One desires simple analytical models or computational algorithms that are applicable *on* the surface and *throughout* the transition between illumination and shadow, not just deep within the shadow zone. Consequently, curved surface diffraction has been examined afresh, using the modern conceptual framework of matched asymptotic expansions.

The theoretical work on the project to date has been especially influenced by the work of V. A. Fock [1], who wrote a number of important papers on electromagnetic wave diffraction during the 1940's and 1950's that were later translated and republished together in a single volume. However, our method of derivation differs in some major details from that of Fock, and it is believed that the fresh perspective will facilitate the extension to broader classes of circumstances.

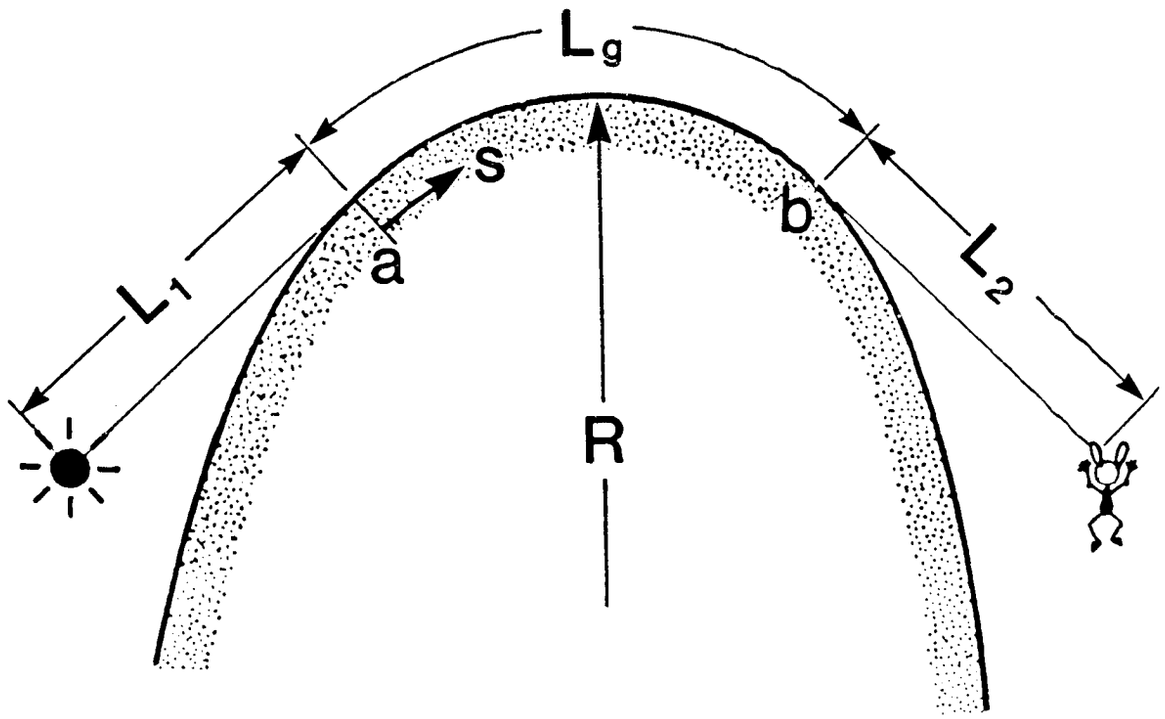

Figure 12. Source and listener on opposite sides of a topographical ridge

Creeping wave solution

Relatively simple results have been derived for the case when the source and listener are on opposite sides of a topographical ridge (Fig. 12) and the listener is deep into the shadow zone. With some minor distinctions, such results have been previously stated within an acoustical context by Hayek, Lawther, Kendig, and Simowitz [2], who adapt a theory developed by Keller [3] to diffraction by a cylinder-topped wedge of finite impedance. Our results, stated here for brevity without a derivation, extend those of Hayek et al. to cases where the radius of curvature and the surface impedance may vary with position. For simplicity, we consider source and listener to be on opposite sides of the ridge; the extension to the oblique incidence case can be worked out without difficulty using relatively simple concepts [4].

The shortest path connecting source and listener has three segments, with lengths L_1, L_g, and L_2. The segment of length L_1 is straight and terminates at the ridge at point a, where the segment is tangent to the curved surface. Similarly, the segment of length L_2 proceeds from a tangent point b to the listener position. The segment of length L_g (g for 'ground') proceeds along the curved top of the ridge from a to b. The surface's local radius $R(s)$ of curvature and specific impedance $Z_S(s)$ are functions of distance s along the surface. Here $s = 0$ corresponds to point a.

The limiting case for which the simplest results most ideally apply is that where kL_1, kL_g, and kL_2 are all substantially larger than unity; here k is the wavenumber $2\pi f/c$ of the sound radiated by the source (strength S). It is also implicitly assumed that the listener is well below the plane tangent to the ridge's surface at point a, which separates the illuminated and shadowed regions. For this limiting case, the sound reaching the listener can be regarded as carried by a succession of 'creeping waves' that travel along the surface from point a and which shed rays into the shadow zone, each such ray proceeding along a straight line that is tangent to the surface; segment L_2 is a path of such a shedded ray. In the extreme limiting case of the sort considered above, the first creeping wave term dominates the sum and the complex amplitude of the acoustic pressure can be written

$$p = \frac{Se^{ikL}}{L^{1/2}L_1^{1/2}L_2^{1/2}}\left(\frac{R_a R_b}{k}\right)^{1/6} A_a^{1/2} A_b^{1/2} e^{i\phi_g} e^{-N_g} \qquad (1)$$

Here $L = L_1 + L_2 + L_g$ is total path length; R_a and R_b are radii of curvature at points a and b. The

ground-induced phase shift ϕ_g and the ground-induced attenuation N_g in nepers are respectively

$$\phi_g = \pi/12 + \int_o^{L_g} (k/2R^2)^{1/3} \tau_R(q)\, ds \tag{2}$$

$$N_g = \int_o^{L_g} (k/2R^2)^{1/3} \tau_I(q)\, ds \tag{3}$$

Here τ_R and τ_I are real and imaginary parts of a quantity τ that is the root having smallest imaginary part of the equation (prime denoting derivative)

$$w_1'(\alpha) - q w_1(\alpha) = 0 \tag{4}$$

where $w_1(\alpha)$ is a Fock function [1,5], given alternatively by

$$w_1(\alpha) = e^{i\pi/6} 2\pi^{1/2} \mathrm{Ai}(\alpha e^{i2\pi/3}) \tag{5}$$

in terms of the Airy function. The root α depends on a normalized surface admittance parameter

$$q = i(kR/2)^{1/3} \rho c / Z_S \tag{6}$$

which varies with distance s if R or Z_S vary with s.

The remaining quantities A_a and A_b that enter into Eq. (1) are values at a and b, respectively, of a quantity $A(q)$, defined such that

$$A(q) = 2^{-5/12} \pi^{-1/4} \left[\tau e^{-i\pi/3} - q^2 e^{-i\pi/3}\right]^{-1/2} \left[\mathrm{Ai}(\alpha e^{i2\pi/3})\right]^{-1} \tag{7}$$

If the surface is rigid, then $q = 0$ and

$$\tau = 1.0188 e^{i\pi/3} = 0.5094 + i0.8823 \tag{8a}$$

$$\mathrm{Ai}(-\tau e^{-i\pi/3}) = 0.5357 \tag{8b}$$

$$A(0) = 0.7817 \tag{8c}$$

For small but nonzero q, an appropriate approximation is

$$\tau = 1.0188 e^{i\pi/3} + e^{i\pi/6}\, q/1.0188 \tag{9}$$

The correction affects the exponent factor N_g and consequently may be of importance; in contrast, little harm is done if $A(q)$ is approximated by $A(0)$. A typical value of q can be estimated by

taking $R = 3\,\mathrm{m}$ and $f = 500\,\mathrm{Hz}$. A survey of ground impedance data is given by Attenborough [6], who fits semi-empirical formulas to such data. Using his expression for the impedances of grassland reported by Embleton, Piercy, and Olson, one obtains $7.19 + i8.19$ for Z_S at 500 Hz. Such values then lead to an estimate of

$$q = 0.22e^{i0.72} \tag{10}$$

Thus, one can regard q as small, but not necessarily negligibly small.

Higher order terms in the creeping wave series have the same form as Eq. (1), the only distinction being that the calculations must use higher roots τ_n of the transcendental equation (5), the roots being ordered according to the magnitude of their imaginary parts. The sum becomes slowly convergent, however, when L_g becomes sufficiently small that many of the N_g are close to zero.

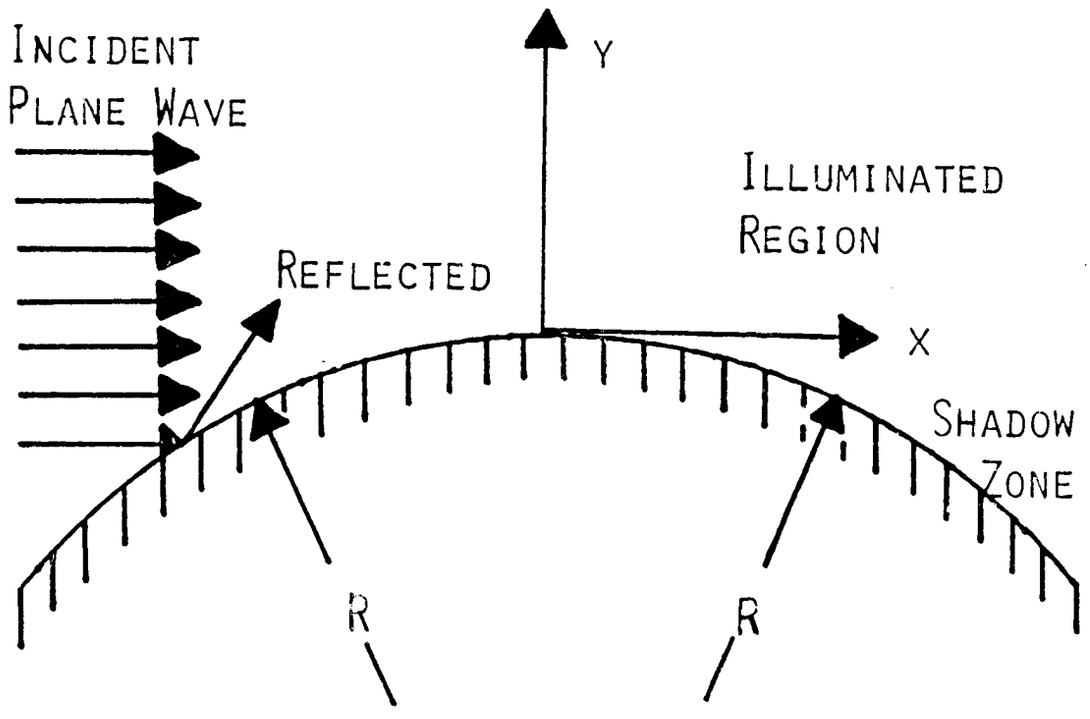

Figure 13. Plane wave incident at curved top of ridge.

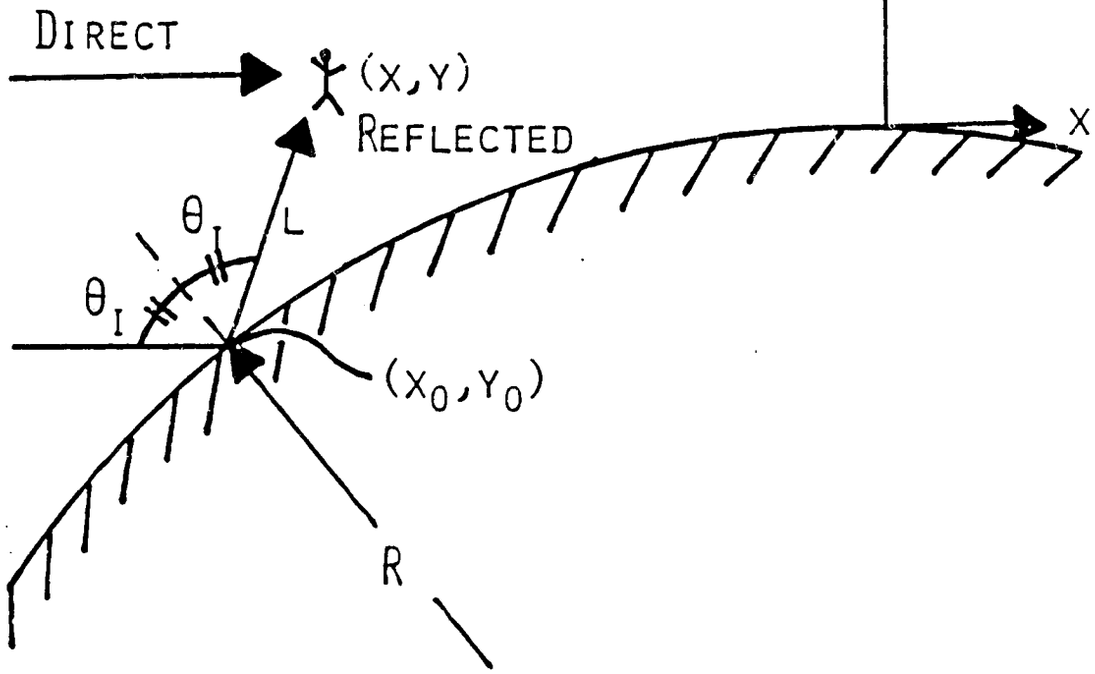

Figure 14. Direct, incident, and reflected rays at a curved surface.

Geometrical acoustics outer solution

At the top of the ridge and in the region of transition from illumination to shadow, a more nearly appropriate solution can be developed using the method of matched asymptotic expansions. A prototype two-dimensional problem (Fig. 13) is when a plane wave of constant frequency with complex pressure amplitude $P_i \exp(ikx)$ reflects and diffracts at a locally reacting (impedance Z_S) curved surface whose radius of curvature R is not necessarily constant, but is nevertheless everywhere large compared with $1/k$. One argues with confidence that the field outside this surface for $x < 0$ can be satisfactorily predicted by geometrical acoustics [4]; this technique should also apply for sufficiently large positive y when $x > 0$. This general region is termed the *outer region*, because in the terminology of matched asymptotic expansions, the geometrical acoustics solution for this region, when extrapolated down to the vicinity of the top of the surface (where $y \approx 0$ and $x \ll R$), furnishes the outer boundary condition for an inner solution that applies near the top of the barrier surface.

The field in this outer region is a superposition of incident and reflected waves, such that

$$p = P_i e^{ikx} + P_i [A(0)/A(\ell)]^{1/2} \Re e^{ikx_o} e^{ik\ell} \tag{11}$$

where $\Re$ is the reflection coefficient and $A(\ell)$ denotes ray tube area after propagation a distance ℓ from the reflection point. The reflection point (x_o, y_o), the local angle of incidence θ_i, the local curvature radius R, and the reflected ray path length ℓ can all be determined for given listener coordinates (x, y) using the law of mirrors and the the mathematical description of the surface (Fig. 14).

Analysis of the so-derived geometrical acoustics solution for the limiting case of points in the vicinity of the curved surface's top yields

$$p \approx P_i e^{ikx} \left\{ 1 + \left[\frac{Q - \frac{2}{3}x}{3Q} \right]^{1/2} \left[\frac{Q - \frac{2}{3}x - \frac{\rho c}{Z_s} R}{Q - \frac{2}{3}x + \frac{\rho c}{Z_s} R} \right] \right\} e^{i\psi} \tag{12}$$

where

$$Q = [(4/9)x^2 + (2/3)Ry]^{1/2} \tag{13a}$$

$$\psi = (2k/R^2)[Q^3 - (8/27)x^3 - (2/3)Rxy] \tag{13b}$$

with R being the radius of curvature of the surface at the top ($x = 0$, $y = 0$).

Solution near top of ridge

Scaling parameters L_x and L_y, equal to $R/(kR)^{1/3}$ and $R/(kR)^{2/3}$, can be introduced such that, when the pe^{-ikx} yielded by Eq. (12) above is expressed in terms of x/L_x and y/L_y, the resulting expression is independent of k and R. Since this furnishes the outer boundary condition on the inner solution, one anticipates that the inner sclution should have comparable features.

To develop the inner solution, the top of the surface is approximated by a parabola $y = -x^2/2R$, and the Helmholtz equation is expressed in parabolic cylinder coordinates u and v, such that

$$x = u\,(1 + [v/R]) \tag{14a}$$

$$y = v\,(1 + [v/2R]) - u^2/(2R) \tag{14b}$$

so $v = 0$ corresponds to the diffracting surface. One then sets p equal to $P_i \exp(iku)$ times a function F of u/L_x and v/L_y. The impedance boundary condition is also expressed in a nondimensional form using these variables. When the derivatives of F with respect to its nondimensionalized arguments are all regarded as being of the order of unity the terms in the partial differential equation satisfied by F become ordered by powers of $(kR)^{-1/3}$.

Substantial agreement with Fock's notation is achieved if one sets

$$\epsilon = (2/kR)^{1/3}, \quad \xi = u/(2^{1/3}L_x), \quad \eta = 2^{1/3}v/L_y \tag{5}$$

$$p = P_i e^{ikx} e^{i\xi^3/3}\, G(\xi, \eta, q) \tag{16}$$

where

$$q = i(kR/2)^{1/3}\rho c/Z_s \tag{17}$$

is an appropriately scaled and nondimensionalized surface admittance. (Expected numerical values of q are discussed further below.) To lowest order in the expansion parameter ϵ, the function G satisfies the parabolic equation

$$i\partial G/\partial\xi + \partial^2 G/\partial\eta^2 + \eta G = 0 \tag{18}$$

with the boundary condition

$$\partial G/\partial\eta + qG = 0 \text{ at } \eta = 0 \tag{19}$$

The outer boundary condition (here imprecisely stated,for brevity) is that Eq. (16) match Eq. (12) at large positive y cr large negative x.

The general solution of the above posed boundary value problem can be developed by Fourier transform and complex variable techniques, with the result

$$G(\xi,\eta,q) = \pi^{-1/2}\int_{-\infty}^{\infty}\left[v(\alpha-\eta) - \frac{v'(\alpha)-qv(\alpha)}{w_1'(\alpha)-qw_1(\alpha)}w_1(\alpha-\eta)\right]e^{i\alpha\xi}\,d\alpha \tag{20}$$

where $v(\varsigma)$ and $w_1(\varsigma)$ (as well as the functions $u(\varsigma)$ and $w_2(\varsigma)$ defined further below) are Fock functions [5] and simply related to Airy functions of complex argument. (The precise definition of these functions is given further below.) The integral solution (20) is trivially related to what is termed [4,5] *Fock's form of the van der Pol–Bremmer diffraction formula.*

Field on top surface of ridge

One simple limiting case of interest is the acoustic pressure on the surface of the ridge, which is

$$p = P_i e^{ikz} e^{i\xi^3/3} G(\xi,0,q) \tag{21}$$

and corresponds to $\eta = 0$. From Eq. (20) one obtains

$$G(\xi,0,q) = \pi^{-1/2}\int_{-\infty}^{\infty}\left[\frac{w_1'(\alpha)v(\alpha) - v'(\alpha)w_1(\alpha)}{w_1'(\alpha)-qw_1(\alpha)}\right]e^{i\alpha\xi}\,d\alpha \tag{21}$$

However, the numerator in the bracketed term in the integrand here is a wronskian of two solutions of the Airy differential equation, so it must be a constant. One finds, after plugging in the leading asymptotic expression for large positive z, that the constant is simply 1, so one has

$$w_1'v - w_1v' = 1 \tag{22}$$

and, consequently,

$$G(\xi,0,q) = \pi^{-1/2}\int_{-\infty}^{\infty}\frac{e^{i\alpha\xi}}{w_1'(\alpha)-qw_1(\alpha)}\,d\alpha \tag{23}$$

Here q is the normalized surface admittance defined in Eq. (17) and ξ is $(u/R)(kR/2)^{1/3}$, with u being interpreted as being approximately the distance s along the surface from the top of the ridge down the shaded side. A more precise identification is that s is $u + (u^3/6R^2)$, such that the sum of the exponents in the factors e^{iku} and $e^{i\xi^3/3}$, which appear in Eq. (21), is iks. Thus one would rewrite that equation as

$$p = P_i e^{iks} G(\xi,0,q) \tag{24}$$

A form of Eq. (23) that is more appropriate for computation at small to moderate values of ξ can be developed by first deforming the integration contour from the real axis to a broken contour that goes from $\infty e^{i2\pi/3}$ to the origin and then to ∞ along the positive real axis. If one uses β to denote distance from the origin along the first leg of this contour and recognizes that

$$w_1(\beta e^{i2\pi/3}) = e^{i\pi/3} w_2(\beta) \tag{25a}$$

$$w_1'(\beta e^{i2\pi/3}) = e^{-i\pi/3} w_2'(\beta) \tag{25b}$$

one can derive

$$G(\xi, 0, q) = \pi^{-1/2} \int_o^\infty \frac{e^{-i\beta\xi/2} e^{-\beta\xi 3^{1/2}/2}}{w_2'(\beta) - e^{i2\pi/3} q w_2(\beta)} d\beta + \pi^{-1/2} \int_o^\infty \frac{e^{i\alpha\xi}}{w_1'(\alpha) - q w_1(\alpha)} d\alpha \tag{26}$$

The two integrals that appear here are highly convergent because at large positive real values of their arguments both w_1 and w_2 approach

$$w_1(z) \to w_2(z) \to z^{-1/4} e^{(2/3) z^{3/2}} \tag{27}$$

Thus one now has a version amenable to numerical computation.

The apparent insertion loss $20 \log(1/|G|)$ calculated using the above formula is plotted versus ξ in Fig. 15. For the rigid barrier case, $q = 0$, the geometrical acoustics solution predicts a pressure doubling at the surface, so the insertion loss must approach -6dB at large negative ξ.

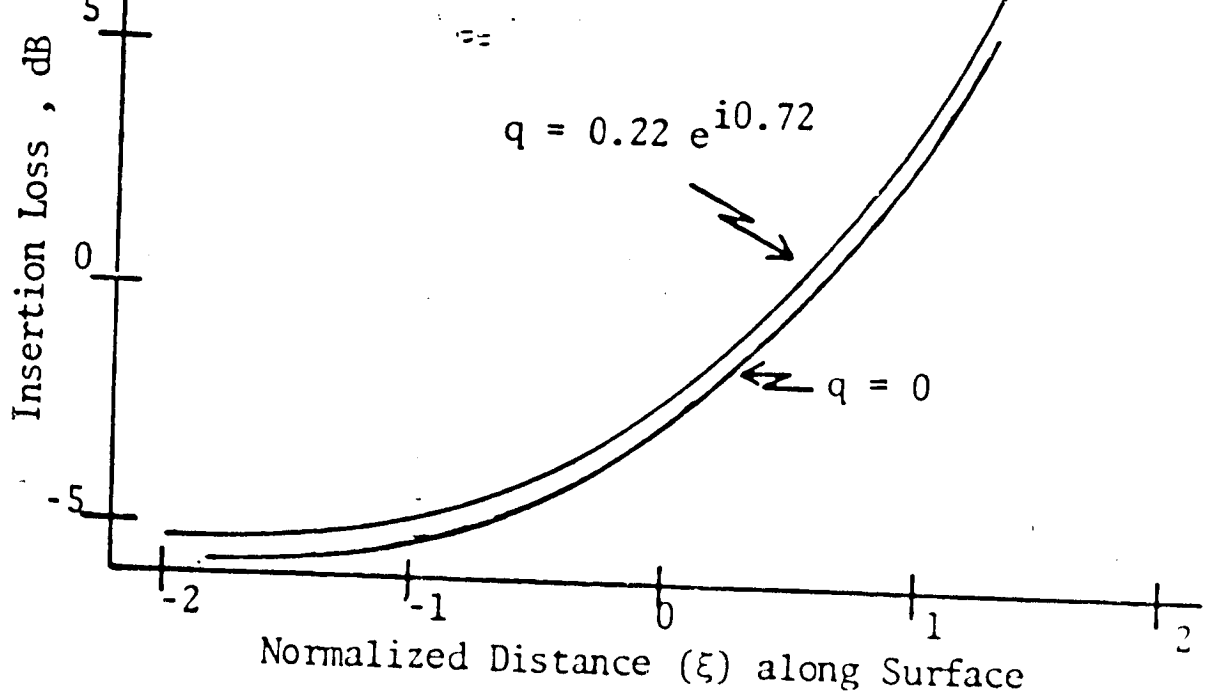

Figure 15. Apparent insertion loss along the ground surface of a topographical ridge.

Transition between illumination and shadow

A discussion of how the inner solution in Eqs. (16) and (20) above matches a further geometrical acoustics solution in the shadow zone is deferred to future reports. Deep in the shadow zone the appropriate version of the integral is a sum over residues from poles in the first quadrant, each such term giving rise to a creeping wave. This creeping wave solution is essentially the same as discussed earlier in the present report.

The creeping wave series is not convergent at the boundary between illumination and shadow, where $\xi = \eta^{1/2}$. An appropriate and suggestive form of the function G near this transition line when η is somewhat larger than unity is

$$G = e^{-i\xi^3/3}e^{i\xi\eta} - e^{i(2/3)\eta^{3/2}}\left[H(X)e^{-i(\pi/2)X^2} + \frac{1+i}{2}A_D(X)\right]$$

$$-\frac{e^{-i\frac{\pi}{12}}e^{i\frac{2}{3}\eta^{\frac{3}{2}}}}{\pi^{1/2}\eta^{1/4}}\int_o^\infty e^{is}e^{i(2\pi/3)(\xi-\eta^{1/2})}\left(\frac{v'(s)-qe^{i\frac{2\pi}{3}}v(s)}{w_2'(s)-qe^{i\frac{2\pi}{3}}w_2(s)}\right)ds$$

$$-\frac{e^{i\frac{\pi}{4}}e^{i\frac{2}{3}\eta^{\frac{3}{2}}}}{\pi^{1/2}\eta^{1/4}}\int_o^\infty e^{is}(\xi-\eta^{1/2})\left(\frac{v'(s)-qv(s)}{w_1'(s)-qw_1(s)}\right)ds \qquad (28)$$

where $X = (2/\pi)^{1/2}\eta^{1/4}(\xi-\eta^{1/2})$ and $A_D(X)$ is the diffraction integral [4], which is simply related to Fresnel integrals and which is invariably present in asymptotic expressions for diffraction by sharp edges. Additional restrictions produce significant analytical simplification.

Airy and Fock functions of complex argument

As described in the preceding sections, the theory of diffraction by curved surfaces of finite acoustic impedance involves integrals (contour integrals in general) with integrands that can be expressed in terms of Airy and Fock functions of complex argument. It would therefore seem imperative that one have algorithms capable of calculating such functions to high precision for arbitrary complex argument. The algorithms we found reported in the literature were developed for functions of real argument only, so some effort was devoted to developing new algorithms. The subroutines described in the present report are in a version (*IBM Professional Fortran* or, briefly, Profort) of FORTRAN 77 that can be used on the IBM PC, but it is intended that they be adapted in the near future to VAX Fortran. To achieve the desired accuracy with Profort (roughly ten significant figures), it was necessary to use double precision. IBM Professional

Fortran (Profort) has complex number capabilities, but not in double precision, so the present version does all complex arithmetic explicitly.

In principle, the Airy function of complex argument can always be calculated from the power series form

$$\mathrm{Ai}(z) = c_1 f(z) - c_2 g(z) \tag{29}$$

where the leading coefficients are

$$c_1 = 3^{-2/3}/\Gamma(2/3) = 0.355028053887817\ldots \tag{30a}$$

$$c_2 = 3^{-1/3}/\Gamma(1/3) = 0.258819403792807\ldots \tag{30b}$$

and the intrinsic power series are

$$f(z) = 1 + \frac{1}{3\cdot 2}z^3 + \frac{1}{6\cdot 5\cdot 3\cdot 2}z^6 + \frac{1}{9\cdot 8\cdot 6\cdot 5\cdot 3\cdot 2}z^9 + \ldots \tag{31a}$$

$$g(z) = z + \frac{1}{4\cdot 3}z^4 + \frac{1}{7\cdot 6\cdot 4\cdot 3}z^7 + \frac{1}{10\cdot 9\cdot 7\cdot 6\cdot 4\cdot 3}z^{10} + \ldots \tag{31b}$$

In practice, however, this representation is useful for calculations only for moderately small arguments z. The program presented here uses it only if $|z| \leq 3$.

For larger arguments, one is initially tempted to use an asymptotic series representation for the Airy function. However, such a series is not convergent absolutely. Although the magnitudes of successive terms may initially decrease, they eventually reach a minimum and then increase without limit. If one keeps only those terms up to and including the term of minimum magnitude, then this is as good as one can do with an asymptotic series. The error is of the order of magnitude of the next neglected term. Some trial calculations suggested this would not be good enough (given a desired precision of at least 1 part in 10^6) when $|z|$ was of the order of 3. Consequently, an alternate procedure was used.

To describe this alternate procedure, one first notes that the Airy function can alternately be described by the contour integral

$$\mathrm{Ai}(z) = \frac{1}{2\pi}\int_{C_{A1}} e^{(zs - s^3/3)}\, ds \tag{32}$$

The contour C_{A1} can be initially thought of as proceding in the complex s-plane along the broken line which goes from $\infty e^{i5\pi/6}$ to the origin and then to $\infty e^{i\pi/6}$. If the argument variable z lies, however, in the right 2/3-rd's of the complex z-plane, then the integration path C_{A1} can

be deformed to one that passes through a saddle point, going to this saddle point *up* a path of steepest descents and then away from this saddle point *down* a path of steepest descents. (By the statement that the argument variable z lies in the right 2/3-rd's of the complex plane, one means that the phase of z lies between $-2\pi/3$ and $2\pi/3$. Because of the identity in Eq. (47a), given further below, this turns out to be no real restriction.)

The applicable saddle point, obtained by setting the derivative of the exponent to zero, is at $s = iz^{1/2}$. To change the contour to the path of steepest descents, one sets $s = iz^{1/2} + u$ such that the exponent in the integrand can be written $-(2/3)z^{3/2} - \ell^2$ where $\ell^2 = z^{1/2}u^2 - (i/3)u^3$. The saddle point now corresponds to $u = 0$ or, equivalently, to $\ell = 0$. The path of steepest descents is a path along which ℓ is real; the definition of ℓ can be refined such that the mapping of C_{Ai} to the ℓ-plane can be deformed to a path that goes from $-\infty$ to ∞ along the real axis. The integral expression for the Airy function can accordingly be rewritten

$$\mathrm{Ai}(z) = \frac{1}{2\pi} e^{-(2/3)z^{3/2}} \int_{-\infty}^{\infty} \frac{2\ell}{2uz^{1/2} - iu^2} e^{-\ell^2} \, d\ell \tag{33}$$

where ℓ and u are related by

$$\ell^2 = z^{1/2}u^2 - (i/3)u^3 \tag{34}$$

The latter is a cubic equation for u as a function of ℓ; the desired root must be zero when ℓ is zero; moreover, $u(\ell)$ must be a continuous function of ℓ. The two possibilities correspond to $u(\ell)$ for small ℓ being either $+\ell/z^{1/4}$ or $-\ell/z^{1/4}$. The requirement that a contour from $-\infty$ to ∞ along the real ℓ axis be an admissible deformation of the mapping of C_{Ai} into the ℓ-plane indicates that the former choice is correct. Thus one can write

$$u = \frac{K\ell}{z^{1/4}} \tag{35}$$

where $K(\ell)$ is such that $K(0) = 1$, and is a solution of the cubic equation

$$1 = K^2 - \frac{i}{3}\frac{\ell K^3}{z^{3/4}} \tag{36}$$

The appropriate solution of the above cubic equation can be worked out with some effort, the result being

$$K = \frac{iz^{3/4}}{\ell}\left[-1 + e^{i\pi/3}A + e^{-i\pi/3}A^{-1}\right] \tag{37}$$

where

$$A = \left[\left(1 + \frac{3\ell^2}{4z^{3/2}}\right)^{1/2} - \frac{3^{1/2}\ell}{2z^{3/4}}\right]^{2/3} \tag{38}$$

For small values of $\ell z^{-3/4}$ it is appropriate to replace the latter by its power series, which to fourth order is

$$A = 1 - \frac{2}{3}s + \frac{2}{9}s^2 + \frac{5}{81}s^3 - \frac{16}{243}s^4 + \dots \tag{39}$$

where here we abbreviate $s = (3^{1/2}/2)\ell z^{-3/4}$.

Since the integral of $e^{-\ell^2}$ over ℓ from $-\infty$ to ∞ is $\pi^{1/2}$, one can derive from the above expressions

$$\mathrm{Ai}(z) = \frac{e^{-(2/3)z^{3/2}}}{2\pi^{1/2}z^{1/4}}\left[1 + \frac{1}{\pi^{1/2}}\int_{-\infty}^{\infty} F_M(\ell, z^{3/4})e^{-\ell^2}\,d\ell\right] \tag{40}$$

where

$$F_M(\ell, z^{3/4}) = \left(K - i\frac{K^2\ell}{2z^{3/4}}\right)^{-1} - 1 \tag{41}$$

or,equivalently, with the substitution of Eq. (37),

$$F_M(\ell, z^{3/4}) = \frac{-2i\ell z^{-3/4} - 1 - e^{i2\pi/3}A^2 - e^{-i2\pi/3}A^{-2}}{e^{i2\pi/3}A^2 + e^{-i2\pi/3}A^{-2} + 1} \tag{42}$$

with

$$A^{\pm 2} = \left[\left(1 + \frac{3\ell^2}{4z^{3/2}}\right)^{1/2} \mp \frac{3^{1/2}\ell}{2z^{3/4}}\right]^{4/3} \tag{43}$$

The leading term in Eq. (40) (i.e., that which results when F_M is formally set to zero) is the first term in the asymptotic expression for $\mathrm{Ai}(z)$.

What is achieved with the introduction of Eq. (40) is that the integrand is not oscillatory, so the integral is highly convergent. The integral is done numerically using a Hermite integration scheme [7], so that

$$\int_{-\infty}^{\infty} F_M(\ell, z^{3/4})\,d\ell = \sum_{i=1}^{10} \tilde{w}_i e^{-\ell_i^2}\left[F_M(\ell_i, z^{3/4}) + F_M(-\ell_i, z3/4)\right] \tag{44}$$

The sampling points ℓ_i and weights $\tilde{w}_i$ are tabulated in the listing of the subroutine *asmairy*.

A similar procedure has been derived for computation of the derivative of the Airy function,

$$\mathrm{Ai}'(z) = -\frac{z^{1/4}}{2\pi^{1/2}}e^{-(2/3)z^{3/2}}\left[1 + \frac{1}{\pi^{1/2}}\int_{-\infty}^{\infty} G_M(\ell, z^{3/4})e^{-\ell^2}\,d\ell\right] \tag{45}$$

where

$$G_M(\ell, z^{3/4}) = -1 - \frac{2i\ell}{z^{3/4}}\frac{e^{i\pi/3}A + e^{-i\pi/3}A^{-1}}{1 + e^{i2\pi/3}A^2 + e^{-i2\pi/3}A^{-2}} \tag{46}$$

As mentioned previously, these above integral expressions, Eqs. (40) and (46), are valid only if the phase of z lies between $-2\pi/3$ and $2\pi/3$. However, one can use these expressions in conjunction with the Airy function identities

$$\mathrm{Ai}(z) = e^{i\pi/3}\mathrm{Ai}(ze^{-i2\pi/3}) + e^{-i\pi/3}\mathrm{Ai}(ze^{-i4\pi/3}) \tag{47a}$$

$$\mathrm{Ai}'(z) = e^{-i\pi/3}\mathrm{Ai}'(ze^{-i2\pi/3}) + e^{i\pi/3}\mathrm{Ai}'(ze^{-i4\pi/3}) \tag{47b}$$

Note that, if the phase of z is between $2\pi/3$ and $4\pi/3$, then the arguments $ze^{-i2\pi/3}$ and $ze-i4\pi/3$, which appear on in the terms on the right sides of the above two equations, have phases between $-2\pi/3$ and $2\pi/3$; thus each such term can be calculated using Eqs. (40) and (46).

For the computation of the Fock functions and their derivatives, we use the relations

$$v(z) = \pi^{1/2}\mathrm{Ai}(z) \tag{48a}$$

$$w_1(z) = e^{i\pi/6}2\pi^{1/2}\mathrm{Ai}(ze^{i2\pi/3}) \tag{48b}$$

$$w_2(z) = e^{-i\pi/6}2\pi^{1/2}\mathrm{Ai}(ze^{-i2\pi/3}) \tag{48c}$$

such that

$$v'(z) = \pi^{1/2}\mathrm{Ai}'(z) \tag{49a}$$

$$w_1'(z) = e^{i5\pi/6}2\pi^{1/2}\mathrm{Ai}'(ze^{i2\pi/3}) \tag{49b}$$

$$w_2'(z) = e^{-i5\pi/6}2\pi^{1/2}\mathrm{Ai}'(ze^{-i2\pi/3}) \tag{49c}$$

The core algorithms are consequently those that evaluate $\mathrm{Ai}(z)$ and $\mathrm{Ai}'(z)$ for arbitrary complex argument.

The algorithms given here have been checked against Fock's tables (which appear on pages 393–412 of his *Electromagnetic Propagation and Diffraction Problems* [1]). Fock tabulates $u(z)$, $u'(z)$, $v(z)$, and $v'(z)$ for real z between -9 and $+9$ to four significant figures. The function $v(z)$ is just $\pi^{1/2}\mathrm{Ai}(z)$, while

$$u(z) = \frac{1}{2}\left(w_1(z) + w_2(z)\right) \tag{50}$$

such that

$$w_1(z) = u(z) + iv(z) \tag{51a}$$

$$w_2(z) = u(z) - iv(z) \tag{51b}$$

If z is real, then both $u(z)$ and $v(z)$ are real, $u(z)$ and $v(z)$ are the real and imaginary parts of $w_1(z)$, or, equivalently, the real and negative imaginary parts of $w_2(z)$. Our program's results (believed to be accurate to 10 significant figures), when rounded off to 4 figures, agree identically with Fock's results. For example, Fock gives $u'(9) = 113.10 \times 10^6$, and we find it to be 113.095831×10^6.

PAPERS AND PUBLICATIONS

The following paper will be presented at the forthcoming meeting of the Acoustical Society of America in Cleveland, Ohio in May 1986.

Curved surface diffraction theory derived and extended using the method of matched asymptotic expansions. Allan D. Pierce, Geoffrey L. Main, and James A. Kearns, School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332. — Consideration is given to the top of a wide barrier with variable radius of curvature R. The surface has finite acoustic surface impedance Z. Because kR is assumed large, the illuminated region can be approximated by geometric acoustics, such that plane wave reflection rules apply locally. The intricate interference pattern between incident and reflected ray fields assumes a tractable analytical form near the barrier top, which is subsequently used in a MAE solution of the overall diffraction problem. Unambiguous length scales result for radial and tangential distances along the barrier top. The inner solution is developed by expressing the wave equation in terms of such scales, subsequently identifying the expansion parameter as $(kR)^{-1/3}$. A parabolic equation emerges, with a boundary condition involving a scaled impedance $(Z/\rho c)(kR)^{-1/3}$; the outer boundary condition results from matching to the geometric acoustics solution. Outer expansion of the solution of the parabolic equation into the shadow zone yields an inner boundary condition on the ray theory solution for the diffracted wave. Results are similar to those previously derived for electromagnetic diffraction problems by V. A. Fock, but the MAE interpretation facilitates an extension to problems of multiple barriers. (Work supported by NASA-Langley Research Center.)

The following paper will be presented at the forthcoming International Congress of Acoustics in Toronto in July 1986, and will appear in the proceedings of that congress.

Sound propagation over large smooth ridges in ground topography. Allan D. Pierce, Geoffrey L. Main, James A. Kearns, Daniel R. Benator, and James R. Parish, Jr. School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332. — A theory similar to those developed by Fock and others during the 1940's and 1950's for electromagnetic wave diffraction by curved surfaces applies to acoustic propagation at low angles with the ground over an intervening ridge of finite impedance. The creeping wave series is not used at the top of the ridge or for the transition between illumination and shadow; the analysis reduces instead to numerical and approximate integration of Fock's form of the van der Pol–Bremmer diffraction formula. Laboratory scale experiments are in progress to test and guide the analytical developments.

The following paper will be presented at the forthcoming 1986 International Conference on Noise Control Engineering (Inter-Noise 86) in Cambridge, Massachusetts in July 1986.

Sound propagation over curved barriers. Allan D. Pierce, Geoffrey L. Main, James A. Kearns, and H.-A. Hsieh, School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332. — A general discussion is given of wide barriers with curved tops; examples of such are naturally occurring topographical ridges and earth berms with rounded tops. The analytical developments reviewed are for circumstances when the local radius of curvature R of the barrier is continuous along the surface and large compared to a wavelength. If the source and listener are at large distances from the barrier top and the listener is deep within the shadow zone, then the creeping wave series previously introduced into noise control applications by Hayek and others gives simple and accurate predictions. The present paper extends this model to instances where the surface impedance varies with position along the surface. The latter part of the paper introduces a matched asymptotic expansion theory that contains concepts and results analogous to those developed by V. A. Fock. Explicit numerical results are given for the acoustic pressure on the surface of the barrier near the point where acoustic shadowing begins. The extended theory also yields simple results for the farfield transition between illumination and shadow.

REFERENCES

[1] V. A. Fock, *Electromagnetic Diffraction and Propagation Problems* (Pergamon, London, 1965).

[2] S. I. Hayek, J. M. Lawther, R. P. Kendig, and K. T. Simowitz, Investigation of Selected Noise Parameters of Acoustical Barriers, report to NAS-NRC Transportation Research Board, 1978.

[3] J. B. Keller, Diffraction by a convex cylinder, *IRE Trans. Antennas and Prop.* **AP-4**, 312–321 (1956).

[4] A. D. Pierce, *Acoustics: An Introduction to its Physical Principles and Applications*, chapters 9 and 10 (McGraw-Hill, New York, 1981).

[5] N. A. Logan, General Research in Diffraction Theory, vol. 1, *Lockheed Missiles Space Div. Rep.* LMSD-288087, December 1959, available from National Technical Information Service, Springfield, VA 22161, accession number AD 241228.

[6] K. Attenborough, Predicted ground effect for highway noise, *J. Sound Vib.* **81**(3), 413–424 (1982).

[7] M. Abramowitz and I. A. Stegun (eds.), *Handbook of Mathematical Functions*, Dover, New York, 1965.

APPENDIX — COMPUTER PROGRAMS

Listing of computer programs for Airy and Fock functions

The input and output subroutines given here are temporary and intended only for checking out the algorithms with a desk-top monitor. The actual program per sé consists of subroutine *Airy* and all those subroutines that it calls. Subroutine *Fock* uses *Airy* to compute the Fock functions. For the numerical evaluation of integrals that describe the diffraction by curved surfaces of finite impedance, programs will be written that call these two subroutines.

```
      program Airychek
*
*                                                    Allan D. Pierce
*                                                        12/25/85

      double precision    x,y,airyr,airyi,dairyr,dairyi,
     +                    vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +                    w2r,w2i,dw2r,dw2i,r,pi,angle

        call input(r,angle)
        pi = 3.1415926535897932D0
        x = r*dcos(angle*pi/180D0)
        y = r*dsin(angle*pi/180D0)
        call Airy(x,y,airyr,airyi,dairyr,dairyi)
        call Fock(x,y,vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +            w2r,w2i,dw2r,dw2i)
        call printAir  (x,y,airyr,airyi,dairyr,dairyi,
     +                  vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +                  w2r,w2i,dw2r,dw2i)

      stop
      end


      subroutine input(r,angle)

      double precision    r,angle
        write  (*,*) 'Program Airychek'
        write  (*,*) 'Version of December 1985'
        write  (*,*)
        write  (*,*)
        write  (*,*) 'What is magnitude of argument?  '
        read   (*,*) r
        write  (*,*) 'What is phase angle in degrees?  '
```

```
      read    (*,*) angle

      return
      end


      subroutine Fock(x,y,vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +              w2r,w2i,dw2r,dw2i)

      double precision    x,y,airyr,airyi,dairyr,dairyi,
     +                    vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +                    w2r,w2i,dw2r,dw2i,pi,ar,ai,cr,ci,
     +                    er,ei,x1,y1,x2,y2,sr,si,tr,ti
        pi = 3.1415926535897932D0
        ar = dsqrt(pi)
        ai = 0D0
        call Airy(x,y,airyr,airyi,dairyr,dairyi)
        call cprod(ar,ai,airyr,airyi,vr,vi)
        call cprod(ar,ai,dairyr,dairyi,dvr,dvi)
        cr = dcos(pi/1.5D0)
        ci = dsin(pi/1.5D0)
        call cprod(x,y,cr,ci,x1,y1)
        er = dcos(pi/6D0)
        ei = dsin(pi/6D0)
        ar = 2D0*ar
        call Airy(x1,y1,airyr,airyi,dairyr,dairyi)
        call cprod(ar,ai,er,ei,sr,si)
        call cprod(sr,si,airyr,airyi,w1r,w1i)
        call cprod(sr,si,cr,ci,tr,ti)
        call cprod(tr,ti,dairyr,dairyi,dw1r,dw1i)
        call cprod(x,y,cr,-ci,x2,y2)
        call Airy(x2,y2,airyr,airyi,dairyr,dairyi)
        call cprod(sr,-si,airyr,airyi,w2r,w2i)
        call cprod(tr,-ti,dairyr,dairyi,dw2r,dw2i)

      return
      end


      subroutine Airy(x,y,airyr,airyi,dairyr,dairyi)

      double precision x,y,airyr,airyi,dairyr,dairyi,
     +                 pi,c1,c2,fork,r
      integer          N

      data     pi /3.1415926535897932D0/,
     +         c1 / 0.355028053887817D0/,
```

```
+        c2 / 0.258819403792807D0/,
+        fork / 3.0D0/,
+        N / 20/
      r = dsqrt(x**2+y**2)
      if (r .le. fork)
+     then
        call Airy1 (x,y,airyr,airyi,dairyr,dairyi,
+                   N,c1,c2)

      else if (r .ge. fork)
+     then
        call Airy2 (x,y,airyr,airyi,dairyr,dairyi,pi)

      end if

      return
      end


      subroutine Airy1 (x,y,airyr,airyi,dairyr,dairyi,
+                   N,c1,c2)

      double precision x,y,airyr,airyi,dairyr,dairyi,
+                br,bi,cr,ci,zetr,zeti,fr,fi,
+                gr,gi,dfr,dfi,dgr,dgi,c1,c2

      integer          N

      br = x
      bi = y
      call cprod(x,y,br,bi,cr,ci)
      call cprod(x,y,cr,ci,zetr,zeti)
      call serairy(zetr,zeti,N,fr,fi,gr,gi,
+                dfr,dfi,dgr,dgi)
      call cprod(x,y,gr,gi,br,bi)
      airyr = c1*fr - c2*br
      airyi = c1*fi - c2*bi
      call cprod(cr,ci,dfr,dfi,br,bi)
      dairyr = 0.5D0*c1*br - c2*dgr
      dairyi = 0.5D0*c1*bi - c2*dgi

      return
      end


      subroutine Airy2 (x,y,airyr,airyi,dairyr,dairyi,pi)
```

```
      double precision x,y,airyr,airyi,dairyr,dairyi,
     +                 pi,r,phi,ar,ai,incr,inci,phase,
     +                 ara,aia,dra,dia,arb,aib,drb,dib,
     +                 phia,phib,phic,arc,aic,drc,dic,
     +                 uar,uai,ubr,ubi,ur,ui,ci

        r = dsqrt(x**2 + y**2)
        phi = phase(x,y,pi)
        if (phi .gt. (4D0/3D0)*pi - 1D-14) phi = phi - 2D0*pi
        if (phi .lt. pi/1.5D0 - 1D-14 .and.
     +      phi .gt. - pi/1.5D0 + 1D-14)
     +     then
           call Airy2a (r,phi,airyr,airyi,dairyr,
     +                  dairyi,pi)
        else if (phi .gt. pi/1.5D0 + 1D-14 .or.
     +           phi .lt. -pi/1.5D0 - 1D-14)
     +     then
           phia = phi - (phi/dabs(phi))*pi/1.5D0
           call Airy2a (r,phia,ara,aia,dra,dia,pi)
           phib = phia - (phi/dabs(phi))*pi/1.5D0
           call Airy2a (r,phib,arb,aib,drb,dib,pi)
           ar = 0.5D0
           ai = (phi/dabs(phi))*dsqrt(3D0)*0.5D0
           call cprod(ar,ai,ara,aia,airyr,airyi)
           call cprod(ar,-ai,dra,dia,dairyr,dairyi)
           call cprod(ar,-ai,arb,aib,incr,inci)
           airyr = airyr + incr
           airyi = airyi + inci
           call cprod(ar,ai,drb,dib,incr,inci)
           dairyr = dairyr + incr
           dairyi = dairyi + inci
        else
           phia = pi/1.5D0 - 1D-15
           call Airy2a (r,phia,ara,aia,dra,dia,pi)
           phib = - pi/1.5D0 + 1D-15
           call Airy2a (r,phib,arb,aib,drb,dib,pi)
           phic = 0D0
           call Airy2a (r,phic,arc,aic,drc,dic,pi)
           ar = dsqrt(3D0)*0.5D0
           ai = 0.5D0
           call cprod(ar,ai,ara,aia,uar,uai)
           call cprod(ar,-ai,arb,aib,ubr,ubi)
           ur = (uar + ubr)/2D0
           ui = (phi/ dabs(phi))*arc/2D0
           ci = (phi/ dabs(phi))*ai
           call cprod(ar,-ci,ur,ui,airyr,airyi)
           call cprod(ar,-ai,dra,dia,uar,uai)
```

```
        call cprod(ar,ai,drb,dib,ubr,ubi)
        ur = (uar + ubr)/2D0
        ui = - (phi/ dabs(phi))*drc/2D0
        call cprod(ar,ci,ur,ui,dairyr,dairyi)

      end if
    return
    end


    subroutine Airy2a (r,phi,airyr,airyi,dairyr,dairyi,pi)

    double precision    r,phi,airyr,airyi,dairyr,dairyi,pi,
   +                    u,e1,e2,a,ar,ai,cr,ci,
   +                    b,br,bi,fr,fi,dfr,dfi

      u = r*dsqrt(r)
      e1 = u*dcos(1.5*phi)/1.5D0
      e2 = u*dsin(1.5*phi)/1.5D0
      a = dexp(-e1) / (2D0*dsqrt(pi*dsqrt(r)))
      ar = a*dcos(e2 +0.25D0*phi)
      ai = - a*dsin(e2 +0.25D0*phi)
      call asmairy(r,phi,fr,fi,dfr,dfi,pi)
      call cprod(ar,ai,fr,fi,cr,ci)
      airyr = cr
      airyi = ci
      b = dsqrt(dsqrt(r)/pi)*dexp(-e1)/2D0
      br = - b*dcos(0.25D0*phi - e2)
      bi = - b*dsin(0.25D0*phi - e2)
      call cprod(br,bi,dfr,dfi,cr,ci)
      dairyr = cr
      dairyi = ci

    return
    end


    function phase(x,y,pi)

    double precision   x,y,px,py,phase,r,pi
      if (x .eq. 0D0 .and. y .eq. 0D0)
   +     then
          phase= 0D0
          return
      end if

      if (dabs(x) .ge. dabs(y))
```

```
+      then
         py = dasin( dabs(y)/ dsqrt(x**2+y**2) )
         if (x .ge. 0D0 .and. y .ge. 0D0)
+           then
             phase = py
         else if (x .le. 0D0 .and. y .ge. 0D0)
+           then
             phase = pi - py
         else if (x .le. 0D0 .and. y .le. 0D0)
+           then
             phase = pi + py
         else if (x .ge. 0D0 .and. y .le. 0D0)
+           then
             phase = - py
         end if

    else if (dabs(x) .le. dabs(y))
+      then
         px = dasin( dabs(x)/ dsqrt(x**2+y**2) )
         if (y .ge. 0D0 .and. x .ge. 0D0)
+           then
             phase = 0.5D0*pi - px
         else if (y .ge. 0D0 .and. x .le. 0D0)
+           then
             phase = 0.5D0*pi + px
         else if (y .le. 0D0 .and. x .le. 0D0)
+           then
             phase = 1.5D0*pi - px
             if (phase .gt. 4D0*pi/3D0)
+               then
                phase = phase - 2D0*pi
             end if
         else if (y .le. 0D0 .and. x .ge. 0D0)
+           then
             phase = -0.5D0*pi + px
         end if
    end if

  return
  end


  subroutine cprod(ar,ai,br,bi,cr,ci)
  double precision   ar,ai,br,bi,cr,ci

    cr = ar*br - ai*bi
    ci = ai*br + ar*bi
```

```
      return
      end


      subroutine serairy(zetr,zeti,N,fr,fi,gr,gi,dfr,dfi,
     +                   dgr,dgi)

      double precision   zetr,zeti,fr,fi,gr,gi,dfr,dfi,
     +                   dgr,dgi,denom

      integer            N,j

        fr = 1D0
        fi = 0D0
        gr = 1D0
        gi = 0D0
        dfr = 1D0
        dfi = 0D0
        dgr = 1D0
        dgi = 0D0

        do 30 k = 1,N
          j = N + 1 - k

          denom = (3D0*j)*(3D0*j-1D0)
          call onestp(denom,zetr,zeti,fr,fi)
          denom = (3D0*j+1D0)*(3D0*j)
          call onestp(denom,zetr,zeti,gr,gi)
          denom = (3D0*j+2D0)*(3D0*j)
          call onestp(denom,zetr,zeti,dfr,dfi)
          denom = (3D0*j)*(3D0*j-2D0)
          call onestp(denom,zetr,zeti,dgr,dgi)
30      continue

      return
      end


      subroutine onestp(denom,zetr,zeti,fr,fi)

      double precision denom,zetr,zeti,fr,fi,
     +                 br,bi,cr,ci

        br = zetr/denom
        bi = zeti/denom
        call cprod(br,bi,fr,fi,cr,ci)
```

```
      fr = 1D0 + cr
      fi = ci

      return
      end


      subroutine asmairy(r,phi,fr,fi,dfr,dfi,pi)

      double precision  z(10), w(10)
      double precision  intr,inti,dintr,dinti,fr,fi,dfr,
     +                  dfi,r,phi,phia,eta,phieta,u,pi,
     +                  kr1,ki1,dkr1,dki1,kr2,ki2,dkr2,dki2

      data    z(1) / 0.2453407083009D0/,
     +        z(2) / 0.7374737285454D0/,
     +        z(3) / 1.2340762153953D0/,
     +        z(4) / 1.7385377121166D0/,
     +        z(5) / 2.2549740020893D0/,
     +        z(6) / 2.7888060584281D0/,
     +        z(7) / 3.3372545673832D0/,
     +        z(8) / 3.9447640401156D0/,
     +        z(9) / 4.6036824495507D0/,
     +        z(10) / 5.3874808900112D0/
      data    w(1) / 0.4909215006667D0/,
     +        w(2) / 0.4938433852721D0/,
     +        w(3) / 0.4999208713363D0/,
     +        w(4) / 0.5096790271175D0/,
     +        w(5) / 0.5240803509486D0/,
     +        w(6) / 0.5448517423644D0/,
     +        w(7) / 0.5752624428525D0/,
     +        w(8) / 0.6222786961914D0/,
     +        w(9) / 0.7043329611769D0/,
     +        w(10) / 0.8985919614532D0/

      intr = 0D0
      inti = 0D0
      dintr = 0D0
      dinti = 0D0
      phia = phi
      if (phia .gt. pi) phia = phia-2.0D0*pi
      if (phia .lt. -pi) phia = phia+2.0D0*pi
      eta = dsqrt(dsqrt(r*r*r))
      phieta = (3D0/4D0)*phia

      do 20 j=1,10
        u = z(j)/eta
```

```
          call steep(u,phieta,pi,kr1,ki1,dkr1,dki1)
          call steep(-u,phieta,pi,kr2,ki2,dkr2,dki2)
          intr = intr + w(j)*(kr1+kr2)*dexp(-z(j)*z(j))
          inti = inti + w(j)*(ki1+ki2)*dexp(-z(j)*z(j))
          dintr = dintr + w(j)*(dkr1+dkr2)*dexp(-z(j)*z(j))
          dinti = dinti + w(j)*(dki1+dki2)*dexp(-z(j)*z(j))
20        continue

        fr = 1D0 + intr/ dsqrt(pi)
        fi = inti/ dsqrt(pi)
        dfr = 1D0 + dintr/ dsqrt(pi)
        dfi = dinti/ dsqrt(pi)

      return
      end


      subroutine steep(u,phieta,pi,kr,ki,dkr,dki)

      double precision  u,phieta,pi,kr,ki,dkr,dki,
     +                  ub,ubr,ubi,dumr,dumi,ubsqr,
     +                  ubsqi,ucubr,ucubi,rr,ri,denom,
     +                  radsqr,radsqi,rad,pt,phase,
     +                  phirad,a32r,a32i,a32,pa32,
     +                  asq,phasq,a,pha,denomr,denomi,
     +                  denomsq,recipr,recipi,yr,yi,
     +                  numr,numi,newr,newi,xr,xi,br,bi

      integer           k,j

        ub = dsqrt(3D0)*0.5D0*u
        ubr = ub*dcos(phieta)
        ubi = ub*dsin(phieta)
        dumr = ubr
        dumi = ubi
        call cprod(dumr,dumi,ubr,ubi,ubsqr,ubsqi)

        if (dabs(ub) .lt. 0.001D0)
     +     then
           call cprod(ubr,ubi,ubsqr,ubsqi,ucubr,ucubi)
           numr = (8D0/9D0)*ubr - (40D0/243D0)*ucubr
           numi = (8D0/9D0)*ubi - (40D0/243D0)*ucubi
           numr = numr - dsqrt(3D0)*(14D0/81D0)*ubsqi
           numi = numi + dsqrt(3D0)*(14D0/81D0)*ubsqr
           denomr = - numr
           denomi = - 4D0/ dsqrt(3D0) - numi
           recipr = denomr / (denomr**2 + denomi**2)
```

```
            recipi = -denomi / (denomr**2 + denomi**2)
            call cprod(recipr,recipi,numr,numi,kr,ki)
            newr = (32D0/9D0)*ubr - (100D0/243D0)*ucubr
            newi = (32D0/9D0)*ubi - (100D0/243D0)*ucubi
            newr = newr + dsqrt(3D0)*(10D0/81D0)*ubsqi
            newi = newi - dsqrt(3D0)*(10D0/81D0)*ubsqr
            call cprod(recipr,recipi,newr,newi,dkr,dki)
          else if (dabs(ub) .ge. 0.001D0)
     +       then

            if (ub .gt. 10D0)
     +         then
               rr = ubsqr/(ubsqr**2+ubsqi**2)
               ri = -ubsqi/(ubsqr**2+ubsqi**2)
               a32r = 0.5D0*ubr/(ubr**2 + ubi**2)
               a32i = -0.5D0*ubi/(ubr**2 + ubi**2)
               do 20 j = 1,10
                 k = 5 - j + 1
                 denom = (0.5D0 - k)/(k + 1D0)
                 call onestp(denom,rr,ri,a32r,a32i)
20               continue
            else if (ub .le. 10D0)
     +         then
               radsqr = 1D0 + ubsqr
               radsqi = ubsqi
               rad = dsqrt(dsqrt(radsqr**2 + radsqi**2))
               pt = phase(radsqr,radsqi,pi)
               if (pt .gt. pi) pt = pt - 2D0*pi
               if (pt .lt. -pi) pt = pt + 2D0*pi
               phirad = pt/2D0
               a32r = rad*dcos(phirad) - ubr
               a32i = rad*dsin(phirad) - ubi
            end if

            a32 = dsqrt(a32r**2 + a32i**2)
            pa32 = phase(a32r,a32i,pi)
            if (pa32 .gt. pi) pa32 = pa32 - 2D0*pi
            if (pa32 .lt. -pi) pa32 = pa32 + 2D0*pi
            asq = a32**(4D0/3D0)
            phasq = pa32*(4D0/3D0)
            a = dsqrt(asq)
            pha = phasq/2D0
            denomr = 1D0 + (asq+1D0/asq)*dcos(phasq+pi/1.5D0)
            denomi = (asq-1D0/asq)*dsin(phasq+pi/1.5D0)
            denomsq = denomr**2 + denomi**2
            recipr = denomr/denomsq
            recipi = -denomi/denomsq
```

```
            yr = -2D0*u*dsin(phieta)
            yi = -2D0*u*dcos(phieta)
            numr = yr - denomr
            numi = yi - denomi
            call cprod(recipr,recipi,numr,numi,kr,ki)
            xr = (a + 1D0/a)*dcos(pi/3D0 + pha)
            xi = (a - 1D0/a)*dsin(pi/3D0 + pha)
            call cprod(yr,yi,xr,xi,br,bi)
            numr = br - denomr
            numi = bi - denomi
            call cprod(recipr,recipi,numr,numi,dkr,dki)
          end if

      return
      end


      subroutine printAir   (x,y,airyr,airyi,dairyr,dairyi,
     +                  vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +                   w2r,w2i,dw2r,dw2i)

      real*8 x,y,airyr,airyi,dairyr,dairyi,vr,vi,dvr,dvi,
     +        w1r,w1i,dw1r,dw1i,w2r,w2i,dw2r,dw2i
         write (*,*)
         write (*,*)
         write (*,1) ' x = ', x, ' y = ', y
1        format (10X, A, F10.4, 10X, A, F10.4)
         write (*,*)
         write (*,3) 'Airyr', 'Airyi', 'Derivr', 'Derivi'
3        format (12X,A,11X,A,10X,A,10X,A)
         write (*,*)
         write (*,101) airyr, airyi, dairyr, dairyi
         write (*,*)
         write (*,3) '  vr', '  vi', '  dvr', '  dvi'
         write (*,*)
         write (*,101) vr,vi,dvr,dvi
         write (*,*)
         write (*,3) '  w1r', '  w1i', '  dw1r', '  dw1i'
         write (*,*)
         write (*,101) w1r,w1i,dw1r,dw1i
         write (*,*)
         write (*,3) '  w2r', '  w2i', '  dw2r', '  dw2i'
         write (*,*)
         write (*,101) w2r,w2i,dw2r,dw2i
101      format (1x, 4D16.8)

      return
```

```
      end
```

Program to compute pressure on ridge surface

Here we list the computer program that was used in the computation of the curves shown in Fig. 15 for the apparent insertion loss of the topographical ridge.

```
      program ridgchek
*
*                                                          Allan D. Pierce
*                                                              2/13/86

      double precision    xi,qr,qi,pi,gr,gi,g,loss

        call ridgput(xi,qr,qi)
        pi = 3.1415926535897932D0
        call ridgint(xi,qr,qi,gr,gi,pi)
        g = dsqrt(gr**2 + gi**2)
        loss = 20D0*dlog10(1D0/g)
        call prtridge(xi,qr,qi,gr,gi,g,loss)

      stop
      end


      subroutine ridgint(xi,qr,qi,gr,gi,pi)

      double precision    xi,qr,qi,gr,gi,pi,
     +                    step,sumr,sumi,x,
     +                    aintr,ainti
      integer             J,n

        call stepfind(xi,step,J)
        sumr = 0.0D0
        sumi = 0.0D0
        x = 0.0D0
        call rgrand(x,xi,qr,qi,aintr,aint pi)
        do 10 n=1,J
          x = x + step
          call rgrand(x,xi,qr,qi,bintr,binti,pi)
          sumr = sumr + aintr + 4D0*bintr
          sumi = sumi + ainti + 4D0*binti
          x = x + step
          call rgrand(x,xi,qr,qi,aintr,ainti,pi)
          sumr = sumr + aintr
          sumi = sumi + ainti
10      continue
```

```
      gr = step*sumr/3D0
      gi = step*sumi/3D0

      return
      end


      subroutine rgrand(x,xi,qr,qi,intr,inti,pi)

      double precision    x,xi,qr,qi,intr,inti,pi,
     +                    y,vr,vi,dvr,dvi,w1r,w1i,
     +                    dw1r,dw1i,w2r,w2i,dw2r,dw2i,
     +                    c,s,ar,ai,br,bi,denomr,denomi,
     +                    denomsq,recipr,recipi,cr,ci.
     +                    mult,int2r,int2i,int1r,int1i

      y = 0D0
      c = 0.5D0
      s = 0.5D0*dsqrt(3D0)
      call Fock(x,y,vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
     +        w2r,w2i,dw2r,dw2i)
      call cprod(-c,s,qr,qi,ar,ai)
      call cprod(ar,ai,w2r,w2i,br,bi)
      denomr = dw2r - br
      denomi = dw2i - bi
      denomsq = denomr**2 + denomi**2
      recipr = denomr/ denomsq
      recipi = - denomi/denomsq
      ar = dcos(x*xi/2D0)
      ai = - dsin(x*xi/2D0)
      call cprod(ar,ai,recipr,recipi,cr,ci)
      mult = dexp(-x*xi*s)/ dsqrt(pi)
      int2r = mult*cr
      int2i = mult*ci

      call cprod(qr,qi,w1r,w1i,br,bi)
      denomr = dw1r - br
      denomi = dw1i - bi
      denomsq = denomr**2 + denomi**2
      recipr = denomr/ denomsq
      recipi = - denomi/denomsq
      ar = dcos(x*xi)
      ai = dsin(x*xi)
      call cprod(ar,ai,recipr,recipi,cr,ci)
      mult = 1D0 / dsqrt(pi)
      int1r = mult*cr
      int1i = mult*ci
```

```
      intr = int1r + int2r
      inti = int1i + int2i

      return
      end


      subroutine stepfind(xi,step,J)

      double precision xi,step,jay
      integer J

      if (dabs(xi) .lt. 2D0)
     +    then
          step = 0.05D0
        else
          step = 0.1D0/ dabs(xi)
      end if
      jay = 3.77D0/ step
      J = jay

      return
      end


      subroutine ridgput(xi,qr,qi)

      double precision xi,qr,qi

      write (*,*) 'Program Ridgchek'
      write (*,*) 'Version of February 1986'
      write (*,*)
      write (*,*)
      write (*,*) 'What is magnitude of argument xi? '
      read (*,*) xi
      write (*,*) 'What is real part of q? '
      read (*,*) qr
      write (*,*) 'What is imaginary part of q? '
      read (*,*) qi

      return
      end


      subroutine prtridge(xi,qr,qi,gr,gi,g,loss)
```

```
      double precision xi,qr,qi,gr,gi,g,loss

        write (*,*)
        write (*,*)
        write (*,1) ' xi = ', xi, ' qr = ', qr, ' qi = ', qi
1       format (10X, A, F10.4, 10X, A, F10.4, A, F10.4)
        write (*,*)
        write (*,3) ' gr', ' gi', ' g', ' loss'
3       format (12X,A,11X,A,10X,A,10X,A)
        write (*,*)
        write (*,101) gr,gi,g,loss
        write (*,*)
101   format (1x, 4D16.8)

      return
      end
```

Complex roots for creeping wave transcendental equation

The program listed here evaluates the roots of the transcendental equation in Eq. (4). The present version is temporary and allows the user to iterate refinements based on Newton's method at the keyboard.

```
      program creepwve
*
*                                                          Allan D. Pierce
*                                                              2/13/86

      double precision xrstart,xistart,qr,qi,
     +                 xrfin,xifin,gr,gi

        call creeput(xrstart,xistart,qr,qi)
        call newtcrp(xrstart,xistart,qr,qi,gr,gi)
        xrfin = xrstart - gr
        xifin = xistart - gi
        call prtcreep(xrstart,xistart,xrfin,xifin)

      stop
      end


      subroutine newtcrp(x,y,qr,qi,gr,gi)

      double precision x,y,qr,qi,gr,gi,ar,ai,
     +                 vr,vi,dvr,dvi,w1r,w1i,
     +                 dw1r,dw1i,w2r,w2i,dw2r,dw2i,
     +                 br,bi,denomr,denomi,
     +                 denomsq,recipr,recipi,
```

```
+                numr,numi,s,c,er,ei,cr,ci

     s =0.5D0*dsqrt(3D0)
     c =0.5D0
     call cprod(x,y,-c,s,er,ei)
     call Fock(er,ei,vr,vi,dvr,dvi,w1r,w1i,dw1r,dw1i,
+              w2r,w2i,dw2r,dw2i)
     call cprod(qr,qi,c,s,cr,ci)
     call cprod(cr,ci,dvr,dvi,ar,ai)
     call cprod(er,ei,vr,vi,br,bi)
     denomr = br + ar
     denomi = bi + ai
     denomsq = denomr**2 + denomi**2
     recipr = denomr/ denomsq
     recipi = - denomi/ denomsq
     call cprod(cr,ci,vr,vi,ar,ai)
     numr = dvr + ar
     numi = dvi + ai
     call cprod(numr,numi,recipr,recipi,er,ei)
     call cprod(er,ei,-c,-s,gr,gi)
   return
   end


   subroutine creeput(xr,xi,qr,qi)

   double precision xr,xi,qr,qi

     write (*,*) 'Program Creepwve'
     write (*,*) 'Version of February 1986'
     write (*,*)
     write (*,*) 'What is real part of q? '
     read (*,*) qr
     write (*,*) 'What is imaginary part of q? '
     read (*,*) qi
     write (*,*) 'What is real part of initial guess for root? '
     read (*,*) xr
     write (*,*) 'What is imaginary part of initial guess? '
     read (*,*) xi

   return
   end


   subroutine prtcreep(xrstart,xistart,xrfin,xifin)

   double precision xrstart,xistart,xrfin,xifin
```

```
      write (*,1) ' xrstart = ', xrstart, ' xrfin = ', xrfin
      write (*,1) ' xistart = ', xistart, ' xifin = ', xifin
1      format (10X, A, F10.4, 10X, A, F10.4)

      return
      end
```

END

DATE

FILMED

NOV 14 1986