MATHEMATICAL MODELS OF ACOUSTIC AND ACOUSTIC-GRAVITY WAVE PROPAGATION IN FLUIDS WITH HEIGHT-DEPENDENT SOUND VELOCITIES

A THESIS

Presented to

The Faculty of the Division

of Graduate Studies

By

Wayne Alan Kinney

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Mechanical Engineering

Georgia Institute of Technology

June, 1976

MATHEMATICAL MODELS OF ACOUSTIC AND ACOUSTIC-GRAVITY WAVE PROPAGATION IN FLUIDS WITH HEIGHT-DEPENDENT SOUND VELOCITIES

Approved:

Allan D. Pierce, Chairman

Melvin R. Corley

W. James Hadden, Jr.

G. Lafayette Maynard

M. B. Sledd

Date approved by Chairman: 6/2/76

ACKNOWLEDGMENTS

Professor Allan D. Pierce has worked very closely with me throughout my graduate education. He has given generously of both his ideas and time, and in his fresh approach to teaching he has been a continual inspiration to me. For his generosity and for the honor of having worked with him for five years, I express my profound gratitude.

I wish to thank Professors W. James Hadden, Melvin R. Corley, and G. Lafayette Maynard for their comments on this dissertation. I am especially grateful to Prof. Marvin B. Sledd for the many hours he spent helping me correct my style and grammar.

Mrs. Sharon Butler performed a very competent job of typing this dissertation. For her help and for her patience in working with me, I am grateful.

I wish to acknowledge the help of my parents which has made my educational experience possible. I wish also to express gratitude to my fiancée, Jane, whose patience and unquestioning support have inspired me and afforded me the freedom of action I have needed.

The effort reported on here was sponsored by the Air Force Geophysics Laboratory.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF ILLUSTRATIONS

Figure | Page

LIST OF SYMBOLS FREQUENTLY USED

Roman Symbols

A_{11}	defined in Eq. (2.7a)
A_{12}	defined in Eq. (2.7b)
a_i	coefficients for cubic splines
c_T	sound speed for upper halfspace
$c(z)$	sound speed as a function of height
$D(\omega,k)$	eigenmode dispersion function defined in Eq. (2.6)
$\hat{f}(\omega)$	Fourier transform of a time-dependent function that characterizes the source
G	function defined in Eq. (2.3)
GR_0, GR_1	gravitational modes
k	horizontal wave number
k_I	imaginary part of horizontal wave number
$k_n(\omega)$	ordered roots of $D(\omega,k)$
k_R	real part of horizontal wave number
k_z	vertical wave number
N_c	number of times a ray has touched a caustic as defined in Eq. (5.63)
p	acoustic pressure
p_o	ambient pressure
q_1, q_2	defined in Eq. (2.14)
Q	function used in Eq. (2.1)
r	horizontal distance of propagation
R	distance from source in the near field

[R]	transmission matrix
R_{11}	[1,1] element of [R]
R_{12}	[1,2] element of [R]
r_{Hor}	distance between source and receiver
s	separation distance between adjacent acoustic rays
$S_0, S_1, S_2, \ldots$	acoustic modes
S(r)	geometric spreading factor used in Eq. (2.1)
t	time
t_{ray}	time of travel along a ray
v and v_p	phase velocity (ω/k)
V	defined in Eq. (2.20)
$v^{(1)}$	complex phase velocity obtained by first iteration with Eq. (2.10a)
$v_a(\omega)$	roots of R_{11} (ω, v)
$v_b(\omega)$	roots of R_{12} (ω, v)
v_L	cutoff value of phase velocity for a non-leaking mode
$v_n(\omega)$	roots of $D(\omega, v)$
w	defined in Eq. (5.5c)
$\bar{w}$	defined in Eq. (5.5d)
x	horizontal dimension in space
X	defined in Eq. (2.10b)
z	height
(z_i, c_i)	lattice points for a sound-speed profile
z_{SC}	source height
z_T	height of bottom of the upper halfspace

Greek Symbols

α	derivative of $R_{11}(\omega,v)$ with respect to v and evaluated at $v = v_a$
β	derivative of $R_{12}(\omega,v)$ with respect to v and evaluated at $v = v_b$
γ	ratio of specific heats for air
Δc_i	defined in Eq. (5.5a)
Δz_i	defined in Eq. (5.5b)
ε	defined in Eq. (2.16)
λ	wavelength
μ,ν	defined in Eq. (2.16)
ρ_o	ambient density
σ	defined in Eq. (2.18)
ω	angular frequency
Ω	defined in Eq. (2.20)
ω_A,ω_B	characteristic frequencies used in Eq. (2.3)
ω_L	cutoff frequency for a non-leaking mode

SUMMARY

Several problems which relate to the propagation of acoustic and acoustic-gravity waves in a medium whose properties vary with height only are considered with the intent of refining existing schemes for the synthesis of waveforms. The contribution from very low frequencies to a modal synthesis of an acoustic-gravity waveform is clarified, and a guide (with numerical examples) is provided for adopting a computer program to include such contributions in the synthesis of waveforms. Also, for the purpose of improving the selection of modes for synthesis, the asymptotic high-frequency behavior of guided modes is explained by use of the W.K.B.J. approximation. Finally, a geometric acoustical scheme is outlined for the prediction of the amplitudes of waves that propagate over long distances. A number of FORTRAN subprograms are provided that exemplify the numerical implementation of this scheme. Recommendations are given for the refinement at low and high frequencies of schemes for the synthesis of waveforms.

CHAPTER I

INTRODUCTION

It was the intent of this dissertation to investigate theoretically the propagation of acoustic and acoustic-gravity waves in fluids whose properties vary with height only. The investigations were carried out for the purpose of refining existing schemes for the synthesis of waveforms. Such schemes have been developed by Harkrider,[1] Pierce and Posey,[2] and others.[3] The propagation of waves which correspond to periods between approximately one and 20 minutes is investigated by use of techniques associated with the synthesis of both modal and geometric acoustical waveforms.

It was the intent of one investigation to clarify the contributions of modes at very low frequencies to a synthesis of waveforms associated with the propagation of acoustic-gravity waves. The computer program INFRASONIC WAVEFORMS[2] had previously been devised to synthesize an infrasonic pressure-time trace as might be generated at long horizontal distances by a large-scale explosion in the atmosphere. In the course of the investigation described here, this program was modified to include contributions at low frequencies from leaking modes of propagation.

In Chapter II, mathematical perturbation techniques are described for the computation of the imaginary part of the horizontal wave number (k_I) for leaking modes. Numerical studies are described in which k_I is calculated for two gravitational modes of interest and for a model atmosphere which is stratified (winds excluded) and terminated by an upper halfspace of constant sound speed. A description of the transition of modes from non-leaking to leaking propagation is also given, and the contribution from branch line integrals in the associated complete Fourier synthesis[2,4] is briefly mentioned.

In Chapter III a detailed description is given of the modification and adaptation of the computer program INFRASONIC WAVEFORMS to include contributions from leaking modes and to improve the accuracy in predicting the early portions of infrasonic arrivals. The numerical implementation of the theory given in Chapter II on the inclusion of leaking modes is also described, and some specific numerical examples which demonstrate that inclusion are given. The complete and current version of INFRASONIC WAVEFORMS is given in the Appendices of reference 5. A hard copy of the program is available from the Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts 01731.

One of the difficulties with the modal approach to the synthesis of waveforms that has arisen in the past has been the presence of what might be called numerical "noise" in derived waveforms due to the fact that the integration over angular frequency in the associated Fourier synthesis[2] was truncated at high frequency. It was felt that, to eliminate this "noise," at least two approaches could be taken. First, the modal synthesis might be extended to higher frequencies by devising a scheme which would carefully select modes for contribution at those frequencies. This selection is difficult with the synthesis in its present state. Secondly, for use at high frequencies where the modal approach is inaccurate, a geometric acoustical scheme might be devised to synthesize waveforms which would serve as the continuations of modal waveforms calculated at lower frequencies.

In Chapter IV the first approach is investigated, wherein the W.K.B.J. method of solution is used to explain the asymptotic high-frequency behavior of guided modes. In Chapter V the second approach is investigated in which a geometric acoustical computational scheme is presented for the description of propagation over long distances. While schemes exist which calculate acoustic ray paths,[6] there appear to be no readily available schemes which are sufficiently accurate to predict the amplitudes of waves that propagate over very long distances.[7]

In the scheme summarized in Chapter V, cubic splines are used to model profiles of sound speed versus height.[8] In addition, techniques are outlined for defining ray paths and for finding distances and times of propagation, turning points for rays, and individual rays that connect source with receiver. Of special significance in the scheme is a parameter that characterizes the spreading of adjacent rays. This parameter is used to determine the number of times that any given ray touches a caustic. It has been shown that a signal propagating along a ray undergoes a phase shift of $-\pi/2$ at a caustic.[9] Thus, the ultimate result of the scheme is a method for computing acoustic amplitudes and waveforms by superposing contributions from individual rays and incorporating phase shifts that occur at caustics. A number of FORTRAN subprograms which exemplify the numerical implementation of this method are given in the Appendix. In addition, some simplified numerical examples are presented which demonstrate the utility of these subprograms.

With the possible exception of the technique for the inclusion of leaking modes, the analytical techniques presented are by no means complete as far as the refinement of existing schemes for synthesizing waveforms is concerned. The main intent here was to investigate and understand avenues of approach which could be useful for such a refinement.

CHAPTER II

PERTURBATION TECHNIQUES FOR THE COMPUTATION OF THE IMAGINARY PART OF THE HORIZONTAL WAVE NUMBER

Introduction

In the formulation of the model on which the computer program INFRASONIC WAVEFORMS is based,[2,10] an intermediate result is derived which expresses the acoustic pressure as a double Fourier integral over angular frequency ω and horizontal wave number k such that

$$p = S(r)\ \mathrm{Re}\left\{\int_0^\infty \hat{f}(\omega)e^{-i\omega t}\int_{-\infty}^{\infty}[Q/D(\omega,k)]e^{ikr}\,dk\,d\omega\right\}. \quad (2.1)$$

Here $S(r)$ is a geometrical spreading factor, which is $1/\sqrt{r}$ for horizontally stratified media and $1/[a_e\sin(r/a_e)]^{1/2}$ if the earth's curvature (a_e = radius of earth) is approximately taken into account. The quantity $\hat{f}(\omega)$ is a Fourier transform of a time-dependent function that characterizes the source. Q is a function of receiver and source heights z_r and z_s, respectively, as well as of ω and k, and possibly of the horizontal direction of propagation if winds are included in the formulation. In any case, given z_r and z_s, Q should have no poles in the complex k-plane when ω is real and positive. The denominator $D(\omega,k)$ (which is termed the <u>eigenmode</u>

dispersion function) may be zero for certain values $k_n(\omega)$ of k.

The k integration contour for Eq. (2.1) is chosen to lie along the real k-axis except where it skirts below or above poles which lie on the real axis (see Fig. 1a, where branch lines are identified by dash marks, poles are indicated by dots, and the k integration contour is marked by arrowheads that show the direction of integration). Let it suffice here to say that the placing of branch cuts and the selection of the k integration contour must be such that the expression for the acoustic pressure dies out at long distance as long as a small amount of damping is included in the formulation. The guided-mode description in the formulation arises when the contour for the k integral is deformed (permissible because of Cauchy's theorem and of Jordan's lemma[11]) to one such as is sketched in Fig. 1b. The poles indicated there above the initial contour are encircled in the counterclockwise sense, and there are contour segments which encircle (also in the counterclockwise sense) each branch cut that lies above the real axis. The integrals around each pole are evaluated by Cauchy's residue theorem so that what remains is a sum of residue terms plus branch line integrals. Each residue term is considered to correspond to a particular guided mode of propagation.

One approximation that was previously made in the guided-mode formulation was to neglect contributions from

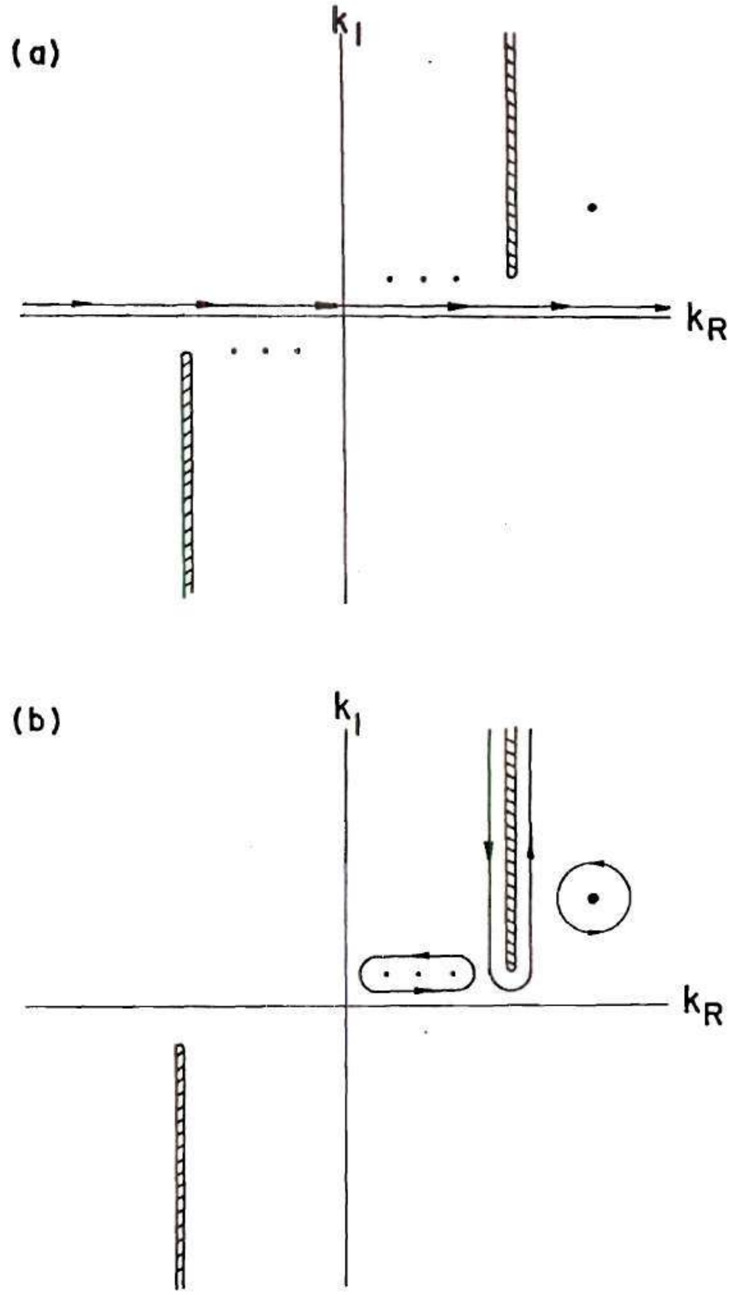

Figure 1. Integration Contours in the Complex k-(Horizontal Wave Number) Plane.
(a) Original Contour.
(b) Deformed Contour.

poles [i.e., the $k_n(\omega)$] which were located above the real k-axis.[2,10] The thought behind this omission was that most of the contributions in the synthesis of waveforms for long propagation distances would come from poles which were on the real k-axis. Another approximation was that, for long distances, the contribution from branch line integrals could be neglected as well. Given these two approximations, the expression for the acoustic pressure in Eq. (2.1) can be approximated as follows:

$$p = \sum_n S(r) \int_{\omega_{Ln}}^{\omega_{Un}} A_n(\omega) \cos[\omega t - k_n(\omega) r + \phi_n(\omega)] \, d\omega, \quad (2.2)$$

where $A_n(\omega)$ and $\phi_n(\omega)$ are defined in terms of the magnitude and phase of the residues of the integrand in Eq. (2.1) and the $k_n(\omega)$ are the real roots for $D(\omega,k)$ (which are numbered in some order with $n = 1,2,3$, etc.).[2] It is understood that in Eq. (2.2), for any given n, $k_n(\omega)$ should be a continuous function of ω between the limits ω_{Ln} (lower) and ω_{Un} (upper). With this understanding, it should be possible to evaluate the resultant integral over ω approximately by the method of stationary phase or by some numerical method.

In spite of the seeming plausibility of the above two approximations, there is a set of circumstances intrinsic to low-frequency infrasonic propagation for which they are not valid, even for distances of propagation of more than 10,000 km. It is these circumstances and their relation to the analytic synthesis of guided-mode atmospheric infrasonic waveforms that are of central interest in the investigation described in this chapter.

Infrasonic Modes

An atmospheric model that is frequently adopted in studies of infrasound[2] is one in which the sound speed $c(z)$ varies continuously with height z in some reasonably realistic manner up to some specified height z_T and is constant (value c_T) for all heights exceeding z_T (see Fig. 2a). Should winds be included in the formulation, the wind velocities are also assumed to be constant in the upper halfspace $z > z_T$. It would seem reasonable to say that one has some choice in specifying the values for both z_T and c_T, even though the computations of such factors as Q and $D(\omega,k)$ in Eq. (1) become more lengthy with increasing z_T. Whatever the choice of z_T, it would seem just as reasonable to choose c_T to be $c(z_T)$ so that the sound-speed profile would then be continuous with height (this is the case for the profile shown in Fig. 2a). Another seemingly plausible choice in modeling the upper halfspace would be to have c_T approach infinity (as illustrated in Fig. 2b). With this choice, the bottom of the upper halfspace would be modeled as a free surface (or pressure release surface) such as is found in models generally adopted in studies of underwater sound for the

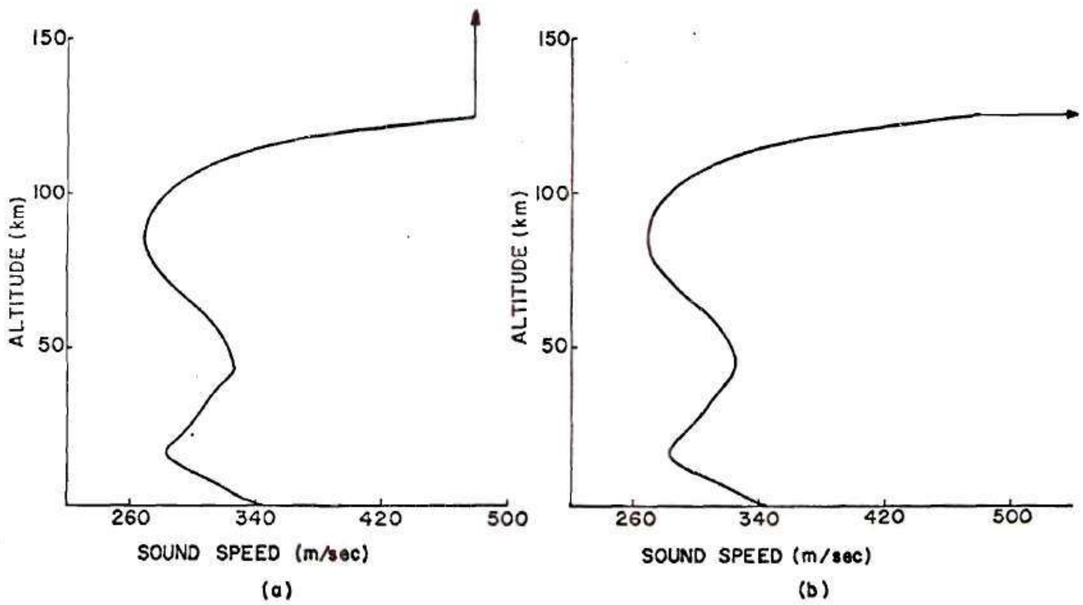

Figure 2. Idealizations of Model Atmospheres. (a) Atmosphere Terminated by an Upper Halfspace with Constant Sound Speed. (b) Atmosphere Sound Speed Formally Approaching Infinity at Some Finite Altitude.

water-air interface. Intuitively, it would seem that if the source and receiver are both near the ground and if the energy actually reaching the receiver travels via modes of propagation channeled primarily in the lower atmosphere, then the actual value of the integral in Eq. (2.1) would be somewhat insensitive to the choices of z_T and c_T. Since this idea, however, remains to be justified in any rigorous sense, it would not seem reasonable to allow c_T to approach infinity at the outset. In typical calculations performed in the past, z_T was taken as 225 km, and c_T was taken as the sound speed ($\approx$ 800 m/sec) at that altitude.[2]

The formulation leading to that version of Eq. (2.1) which is appropriate to infrasound for frequencies at which gravitational effects are important (corresponding to periods greater than one to five minutes) is based on the equations of fluid dynamics with the inclusion of gravitational body forces, the associated nearly exponential decrease of ambient density and pressure with height, and a localized energy source (see in particular pages 17 and 19 of reference 2). When c_T is taken to be finite, the incorporation of gravitational effects in this formulation leads to a dispersion relation for plane waves propagating in the upper halfspace which is (winds neglected)[2,10]

$$k_z^2 = -G^2 = [\omega^2 - \omega_A^2]/c_T^2 - [\omega^2 - \omega_B^2]k^2/\omega^2, \qquad (2.3)$$

where the solution of the linearized equations of fluid dynamics for $z > z_T$ is of the form

$$p/\sqrt{\rho_o} = (\text{Constant})\ e^{-i\omega t}\ e^{ikx}\ e^{ik_z z}\ . \tag{2.4}$$

In these equations p is again the acoustic pressure, ρ_o is ambient density, x is the horizontal space dimension, and k_z is the vertical wave number (alternatively written as iG for inhomogeneous plane waves). ω_A and ω_B are two characteristic frequencies ($\omega_A > \omega_B$) for wave propagation in an isothermal atmosphere where $\omega_A = (\gamma/2)g/c_T$ and $\omega_B = (\gamma - 1)^{1/2}\ g/c_T$ ($g \approx 9.8$ m/sec^2 is the acceleration due to gravity and $\gamma \simeq 1.4$ is the specific heat ratio for air). For given real positive ω and real k, k_z^2 can be positive or negative (G^2 negative or positive, respectively). The values of k at which G^2 is zero turn out, as might be expected, to be the branch points in the k integration in Eq. (2.1). Along the real k-axis, G is either real and positive (so that $e^{ik_z z}$ or e^{-Gz} dies out with increasing z), or else G is of the form ia where a can be positive or negative. From Eq. (2.3), the two branch points are at

$$k_{BR}^{+,-}(\omega) = \pm\ \frac{\omega[\omega_A^2 - \omega^2]^{1/2}}{c_T[\omega_B^2 - \omega^2]^{1/2}}\ . \tag{2.5}$$

Note that for $0 < \omega < \omega_B$, and for k between the branch points on the real axis, G is real and positive. The branch lines extend upwards and downwards from the positive and negative branch points, respectively (recall Fig. 1).

The eigenmode dispersion function $D(\omega,k)$ in the case of atmospheric infrasound can be written in the general form (see page 47 of reference 2)

$$D(\omega,k) = A_{12}R_{11} - A_{11}R_{12} - R_{12}G. \tag{2.6}$$

In this expression, R_{11} and R_{12} are the elements of a transmission matrix [R].[2] They depend on the atmospheric properties only in the altitude range zero to z_T, and are independent of what is assumed for the upper halfspace. In general, their determination requires numerical integration over height of two simultaneous ordinary differential equations (termed the residual equations[2,10,12] in previous literature). They do depend on ω and k (or, alternately, on ω and phase velocity $v = \omega/k$), but are free from branch cuts. Also, they are real when ω and k are real and are finite for all finite values of ω and k. The other parameters A_{12} and A_{11} depend on the properties of the upper halfspace, and on ω and k. A_{11} and A_{12} are given (winds excluded) as

$$A_{11} = gk^2/\omega^2 - \gamma g/[2c_T^2]; \tag{2.7a}$$

$$A_{12} = 1 - c_T^2 k^2/\omega^2. \qquad (2.7b)$$

It may be noted further that, since every quantity in Eq. (2.6) (with the possible exception of G) is real when ω and k are real, the poles that lie on the real k-axis (recall that they are the real roots of D) must be in those regions of the (ω,k)-plane [or, alternatively, the (ω,v)-plane] where $G^2 > 0$. Since at heights above z_T, the integrand of Eq. (2.1) divided by $\sqrt{\rho_o}$ should vary with z as e^{-Gz_T}, there is no leakage of energy into the upper halfspace for those modes that correspond to the above poles. Such modes are termed <u>fully ducted modes</u>. Modes for which there is leakage of energy are termed <u>leaking</u>. If D is considered as a function of ω and phase velocity v, the locus of its real roots $v(\omega)$ (dispersion curves) has (as has been found by numerical computation with the program INFRASONIC WAVEFORMS) the general form sketched in Fig. 3. The nomenclature for labeling the modes (GR for gravity, S for sound) is due to Press and Harkrider.[13] It may be noted from Eq. (2.3) that there are two "forbidden regions" (slashed in the figure) in the (ω,v)-plane. These regions correspond to

$$v < c_T[\omega_B^2 - \omega^2]^{1/2}/[\omega_A^2 - \omega^2]^{1/2} \qquad (2.8a)$$

for $\omega < \omega_B$ and to

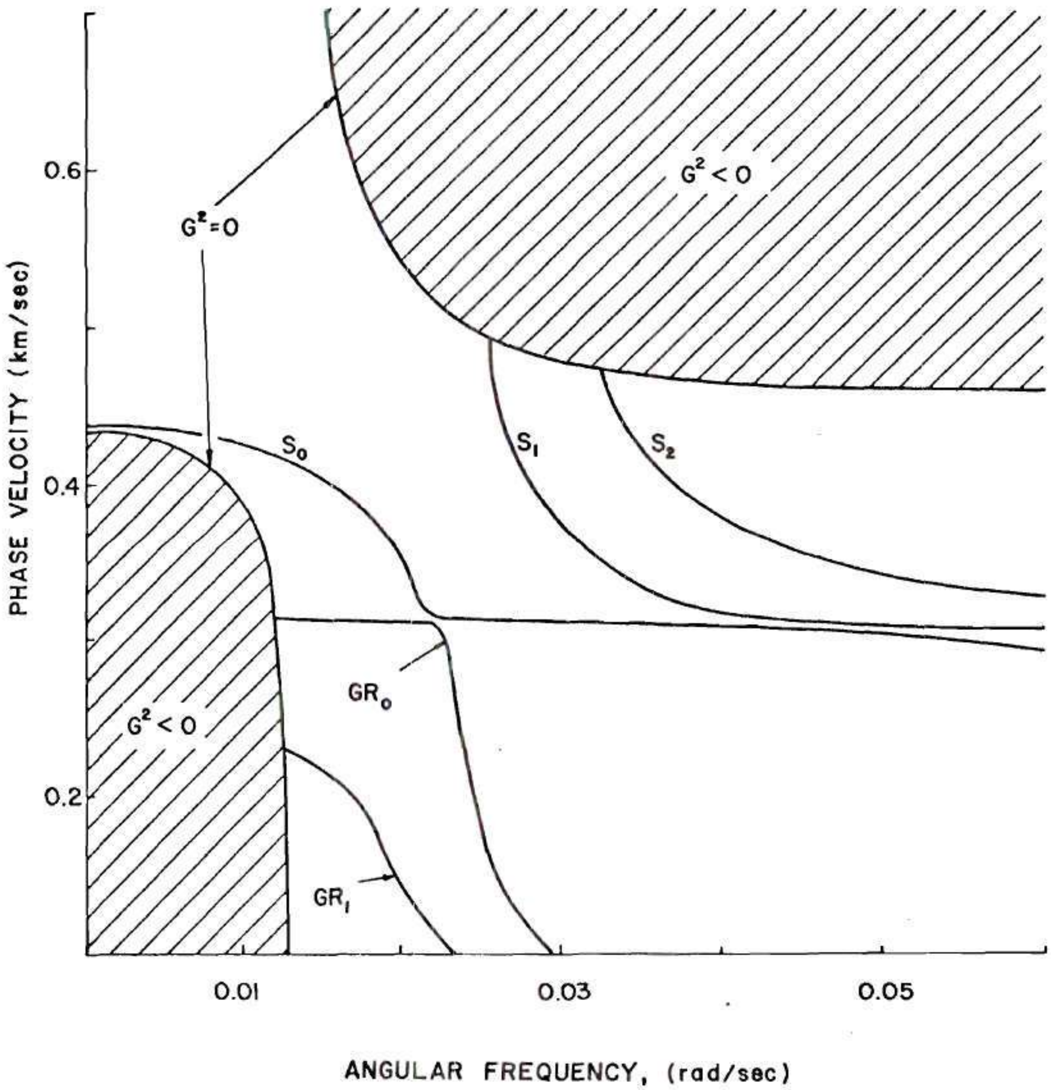

Figure 3. Numerically Derived Plots of Phase Velocity v Versus Angular Frequency ω for Infrasonic Modes.

$$v > c_T[\omega^2 - \omega_B^2]^{1/2}/[\omega^2 - \omega_A^2]^{1/2} \qquad (2.8b)$$

for $\omega > \omega_A$. Within these regions there are no real roots of the function $D(\omega,v)$ because G is imaginary. The existence of the high-frequency upper "forbidden region" implies that the phase velocities for propagating modes are always less than the sound speed chosen for the upper halfspace. It also implies that, in the high-frequency limit, the branch points in the k-plane are at $\pm\ \omega/c_T$. The low-frequency lower-phase-velocity "forbidden region" appears to be due to the incorporation of gravitational effects into the formulation. However, if c_T is allowed to approach infinity, the lower "forbidden region" disappears. Numerical studies were performed with INFRASONIC WAVEFORMS to see just what effect varying c_T had on the dispersion curves shown in Fig. 3. Briefly, the result was that while the forms of the GR_0 and GR_1 modal curves changed little with increasing c_T the lower "forbidden region" shrank in frequency range, and as it did so, the modal curves extended to successively lower frequencies. Thus, it can be seen that the fully ducted GR_0 and GR_1 modes both have a lower frequency cutoff [ω_L in Eq. (2.2)] which depends on c_T. In fact, the larger c_T becomes, the smaller this cutoff frequency becomes.

At this point, there should appear to be the following paradoxes. Given that frequencies below ω_B may be important for the synthesis of the total waveform, an apparently plausible computational scheme based on the reasoning leading to Eq. (2.2) will omit much of the information conveyed by such frequencies. Also, in spite of the plausible premise that energy ducted primarily in the lower atmosphere should be insensitive to the choice for c_T, it can be seen that this choice governs the cutoff frequencies for certain modes and that certain important frequency ranges could conceivably be omitted entirely by a seemingly logical choice for c_T. The resolution of these paradoxes seems to lie in the nature of the approximations made in going from Eq. (2.1) to Eq. (2.2). The latter equation may not be as nearly correct as earlier presumed, and it may be necessary to include contributions from poles off the real axis as well as from the branch line integrals. Even for the case when the propagation distance r is very long, it may be that the imaginary parts of the complex horizontal wave numbers are so small that the magnitude of e^{ikr} in Eq. (2.1) is still not small compared to unity. In addition, a branch line integral may be appreciable in magnitude at large r if there is a pole relatively close to the associated branch cut. These possibilities are investigated in the next section.

Roots of the Dispersion Function

In light of the paradoxes mentioned, it would be desirable to modify the solution represented by Eq. (2.2) so as to remove the apparent artificial low-frequency cutoffs of the GR_0 and GR_1 modes. As a first step, the nature of the eigenmode dispersion function D in the vicinity of the dispersion curve for a particular mode is examined. The curve of values $v_n(\omega)$ of phase velocity v versus ω for a given (n-th) mode is known for frequencies greater than the lower cutoff frequency ω_L. Given this curve, analogous curves $v_a(\omega)$ and $v_b(\omega)$ can be found for values of the phase velocity ω/k at which the functions $R_{11}(\omega,v)$ and $R_{12}(\omega,v)$ in Eq. (2.6), respectively, vanish. One characteristic of the curves $v_n(\omega)$, $v_a(\omega)$, and $v_b(\omega)$ which has been checked numerically for $\omega > \omega_L$ with the use of the program INFRASONIC WAVEFORMS (see Fig. 4) is that, for a given mode of interest, these curves all lie substantially closer to one another than to the corresponding curves for a different mode.

Given the definitions above of $v_a(\omega)$ and $v_b(\omega)$, the dispersion relation $D = 0$ for a single mode may be approximately expressed, through a simple expansion, as

$$D \approx (A_{12})(\alpha)(v - v_a) - [A_{11} + G](\beta)(v - v_b) = 0, \qquad (2.9)$$

where $\alpha = dR_{11}/dv$, and $\beta = dR_{12}/dv$, evaluated at $v = v_a$ and v_b, respectively (for simplicity, D is considered here as a function of ω and $v = \omega/k$ rather than of ω and k). The above equation may also be written in the form

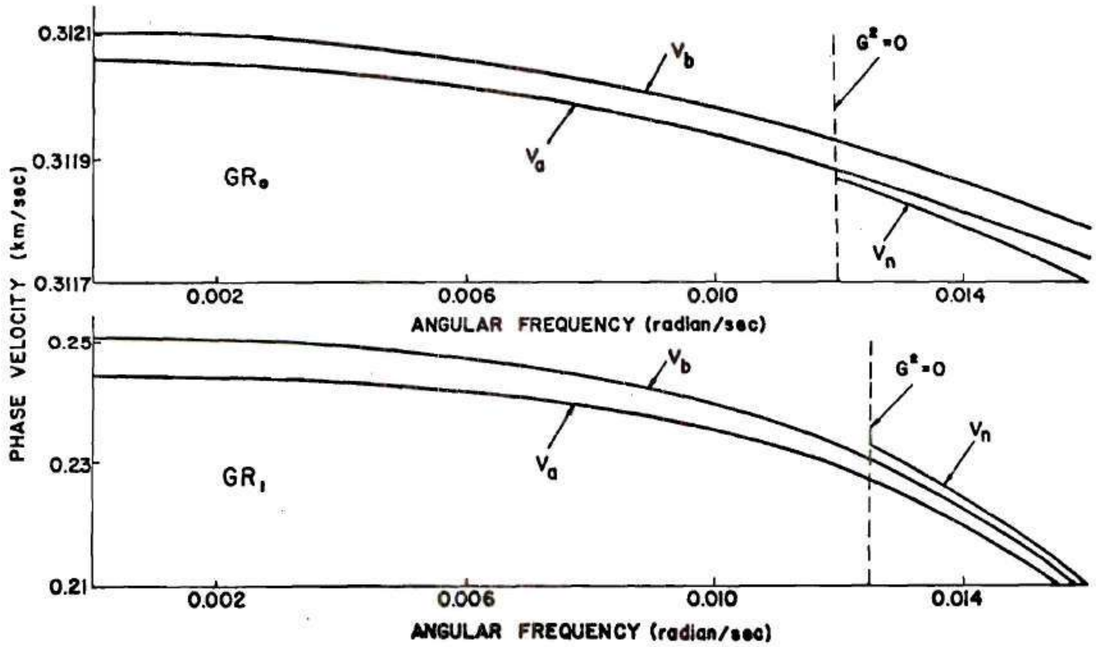

Figure 4. Curves of Phase Velocity (v_n, v_a, v_b) Versus Angular Frequency (ω).

$$v = v_a + (v_a - v_b)X/[1-X], \tag{2.10a}$$

where

$$X = (\beta/\alpha)(A_{11} + G)/A_{12}. \tag{2.10b}$$

Eq. (2.10a) may be considered as a starting point for an iterative solution which develops v in a power series in $v_a - v_b$. With $v = v_a$ as the zeroth iteration, the right hand side of Eq. (2.10a) can be evaluated for the value of v required for the next iteration, etc. This iterative procedure should converge provided that v_a or v_b is not near a point at which G vanishes and provided that G in the vicinity of v_a or v_b is not such that the variable X is close to unity. Among other limitations, the iterative scheme is inappropriate for those values of ω in the immediate vicinity of ω_L. This limitation is discussed further in the next section.

The iterative solutions obtained by the above scheme follow some interesting general trends. In relation to these trends, there are two general theorems of note, the proofs of which follow along lines previously used by Pierce[14] in deriving an integral expression for group velocity. These are that, for ω and v positive and real,

$$R_{12}\ \partial R_{11}/\partial v - R_{11}\ \partial R_{12}/\partial v > 0, \tag{2.11a}$$

$$R_{12}\ \partial R_{11}/\partial\omega - R_{11}\ \partial R_{12}/\partial\omega > 0. \tag{2.11b}$$

Alternately, for $R_{11} = (\alpha)(v - v_a)$ and $R_{12} = (\beta)(v - v_b)$, it follows that

$$\alpha\beta(v_a - v_b) > 0, \tag{2.12a}$$

$$(v - v_b)(v - v_a)(\beta\alpha' - \beta'\alpha) + \beta\alpha[v_b'(v - v_a) - v_a'\ (v - v_b)] > 0, \tag{2.12b}$$

where the primes represent derivatives with respect to ω. Eq. (2.12b) should hold for arbitrary v in the vicinity of v_a and v_b and lead, upon setting $v = v_a$, $v = v_b$, or $v = (v_a v_b' - v_a' v_b)(v_b' - v_a')$, along with the use of Eq. (2.12a), to

$$v_b' < 0, \tag{2.13a}$$

$$v_a' < 0, \tag{2.13b}$$

$$(\alpha/\beta)' > 0. \tag{2.13c}$$

Eq. (2.12a) implies that so long as $\alpha\beta \neq 0$ the two curves $v_a(\omega)$ and $v_b(\omega)$ do not intersect. If α and β have the same sign, then the v_a curve lies above the v_b curve. If α and β differ in sign, then the v_b curve lies above the v_a curve.

To illustrate the general utility of the perturbation approach taken here, values of ω, v_a, v_b, α, β, $v^{(1)}$, and v_n are listed in Table 1 for the GR_0 and GR_1 modes, where $v^{(1)}$ is the result of the first iteration for the phase velocity. The values given there are appropriate to the case of a U. S. Standard Atmosphere[2] without winds which is terminated at a height of 125 km by an upper halfspace possessing a sound speed of 478 m/sec. Note that, for those frequencies at which v_n is computed, the agreement between $v^{(1)}$ and v_n is excellent.

For further illustration of the perturbation technique, detailed plots versus angular frequency are given in Fig. 5 of ω/k_R which is the reciprocal of the real part of $1/v^{(1)}$, and of k_I which is the imaginary part of $\omega/v^{(1)}$ (k_R and k_I are the real and imaginary parts of k, respectively). Note that k_I is zero above the corresponding cutoff frequencies.

Transition of Modes from Non-Leaking to Leaking

The iterative process described by Eqs. (2.10) in the preceding section provides little insight into the behavior of a modal dispersion curve in the immediate vicinity of cutoff (i.e., for values of ω near ω_L). In addition, the process may fail to converge when G is near zero. To explore this transition region, it is sufficient to approximate G in Eq. (2.9) by

Table 1. Frequency-Dependent Parameters Corresponding to the GR_0 and GR_1 Modes .

	ω sec^{-1}	V_a km/sec	V_b km/sec	α sec/km	β sec	$V^{(1)}$ km/sec	V_n km/sec
GR_0	0.0052	0.31203	0.31207	917.4	-2783.7	0.31202121 $-3.184 \times 10^{-6} i$	
	0.0113	0.31190	0.31194	767.9	-3254.2	0.31189059 $-1.721 \times 10^{-6} i$	
	0.0155	0.31176	0.31181	621.9	-3644.3	0.31173763	0.31172882
	0.0165	0.31172	0.31177	581.5	-3738.2	0.31167504	0.31167509
	0.0186	0.31162	0.31168	497.5	-3910.1	0.31153369	0.31153394
GR_1	0.0052	0.24229	0.24816	87.8	-3633.0	0.25267 $-2.715 \times 10^{-3} i$	
	0.0103	0.23433	0.23844	94.7	-3990.0	0.24218 $-1.337 \times 10^{-3} i$	
	0.0144	0.21842	0.22037	150.7	-5307.0	0.21431	0.22178
	0.0165	0.20252	0.20345	265.0	-7767.3	0.20016	0.20463
	0.0175	0.19058	0.19111	418.9	-10,858.0	0.19226	0.19212

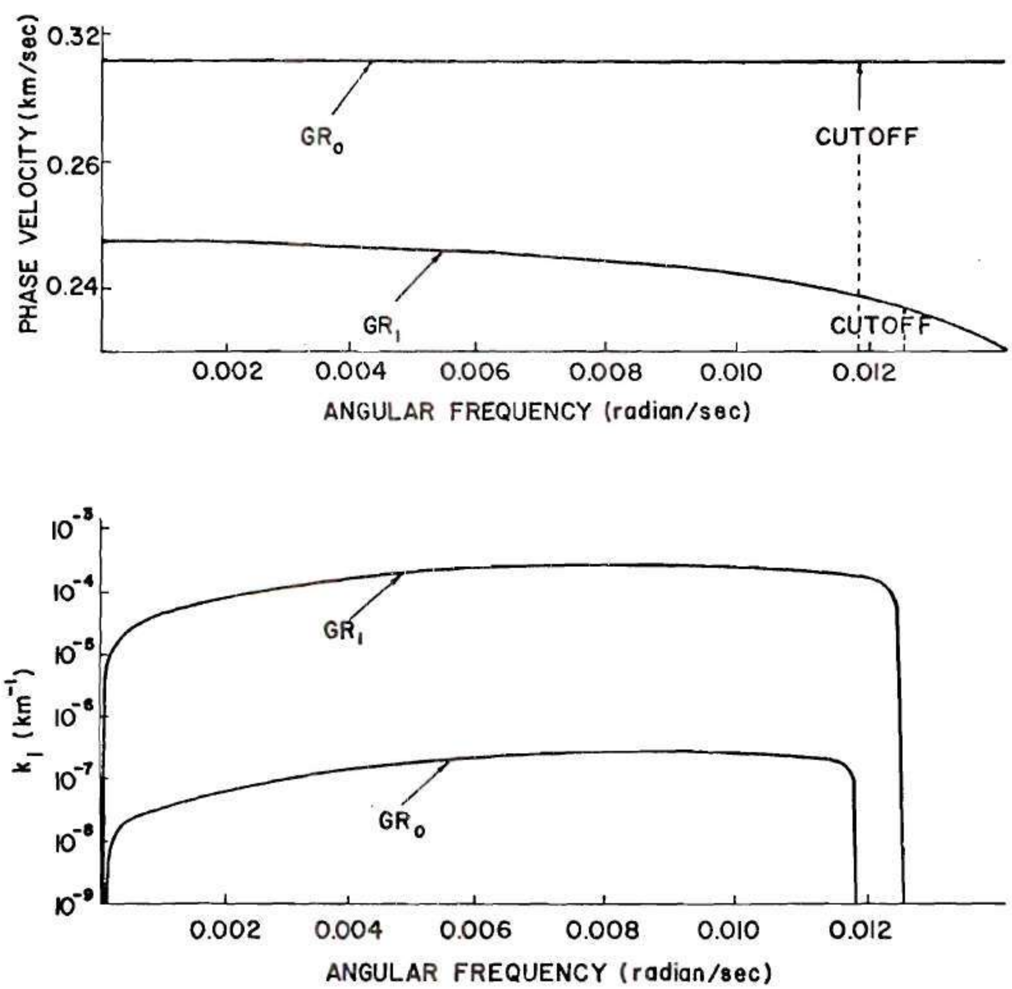

Figure 5. Numerically Derived Plots of Phase Velocity ω/k_R and of the Imaginary Part k_I of the Complex Wave Number k Versus Angular Frequency for the GR_0 and GR_1 Modes.

$$G \simeq [(q_1)(\omega - \omega_L) + (q_2)(v - v_L)]^{1/2}, \qquad (2.14)$$

where q_1 and q_2 are readily identifiable positive numbers which are independent of ω and v [see Eq. (2.9)] and v_L is the limit of the phase velocity on the dispersion curve as ω approaches ω_L from above. The bracketed quantity in Eq. (2.14) may be regarded as a double Taylor series expansion (truncated at first order) of G^2 about the point (ω_L, v_L) at which G^2 vanishes (hence there is no zeroth-order term). That q_1 and q_2 are positive quantities follows from the fact that G^2 is positive outside of the lower "forbidden region" in the (ω, v)-plane (i.e., to the upper right of the line $G^2 = 0$) and also from the fact that the boundary of the lower "forbidden region" slopes obliquely downwards (see Fig. 3).

With the above approximation for G, a further approximation to the eigenmode dispersion function $D(\omega, v)$ [of Eq. (2.9)] in the vicinity of the point (ω_L, v_L) would be

$$D \approx (A_{12}\alpha - A_{11}\beta)\ \{(\Delta v + \mu\Delta\omega) + \varepsilon(\Delta v + \nu\Delta\omega)^{1/2}\}, \qquad (2.15)$$

where $\Delta v = v - v_L$, $\Delta\omega = \omega - \omega_L$, $\nu = q_1/q_2$ and where the quantity μ is either $-dv_a/d\omega$ or $-dv_b/d\omega$ (the two being close in value). The use of the minus sign in the expressions for μ assumes that μ is positive. The quantity ε is

$$\varepsilon = \frac{(q_2^{1/2})(\beta)(v - v_b)}{\beta A_{11} - \alpha A_{12}} . \qquad (2.16)$$

It should be noted that ε depends on v, although, for the purposes of the analytical investigation given here, v may be set equal to v_L. In fact q_1, q_2, β, α, A_{11}, A_{12}, μ, and ν may be considered to be evaluated at $\omega = \omega_L$ and $v = v_L$. Note again that μ and ν are both positive quantities. Furthermore, note that $\nu > \mu$ as is evidenced by the fact that the curve $G^2 = 0$ in the (ω,v)-plane slopes downward more rapidly than the lines $R_{11} = 0$ and $R_{12} = 0$ (see Fig. 4).

From Eq. (2.15) the zeros of D are readily found to be

$$\Delta v = -\mu\Delta\omega + (1/2)\varepsilon^2 \mp \varepsilon(\nu - \mu)^{1/2} [\Delta\omega + \sigma]^{1/2}, \qquad (2.17)$$

where

$$\sigma = \varepsilon^2/[4(\nu - \mu)]. \qquad (2.18)$$

For $|\Delta\omega| << \sigma$, Δv may be further approximated by use of the binomial theorem as

$$\Delta v = -\nu\Delta\omega + [(\nu - \mu)^2/\varepsilon^2](\Delta\omega)^2 \qquad (2.19a)$$

or

$$\Delta v = \varepsilon^2 - (2\mu - \nu)\ \Delta\omega - [(\nu - \mu)^2/\varepsilon^2](\Delta\omega)^2 \qquad (2.19b)$$

for the upper and lower signs of Eq. (2.17), respectively. Eq. (2.19a) (since $\Delta v = 0$ when $\Delta\omega = 0$) is a description of the dispersion curve in the vicinity of the point (ω_L, v_L).

Examination of Eq. (2.19a) shows that as $\Delta\omega$ approaches zero, the dispersion curve becomes tangential to the line $G^2 = 0$. In other words, the two curves do not intersect (refer to Fig. 6). At point A [i.e., at the point (ω_L, v_L)] in the sketch, the two curves are tangent. Between the points A and B, there is a finite gap in the frequency range in which there are no poles in the k- (or v-) plane corresponding to a given n-th mode. The magnitude of the parameter σ (rad/sec) gives an indication of the width of this frequency gap.

In Table 2 the values of ω_L, v_L, q_1, q_2, μ, ν, ε, and σ are given for the GR_0 and GR_1 modes for the model atmosphere corresponding to Fig. 2a. The extremely small values of σ should be noted. Also, a plot of Δv versus $\Delta\omega$ which shows both branches of Eq. (2.17) and which is appropriate for the GR_0 mode is given in Fig. 7. For simplicity, this plot is in normalized form with

$$V = -\{\mu/[2(\nu - \mu)]\}\Omega \mp [1 + \Omega]^{1/2}, \qquad (2.20)$$

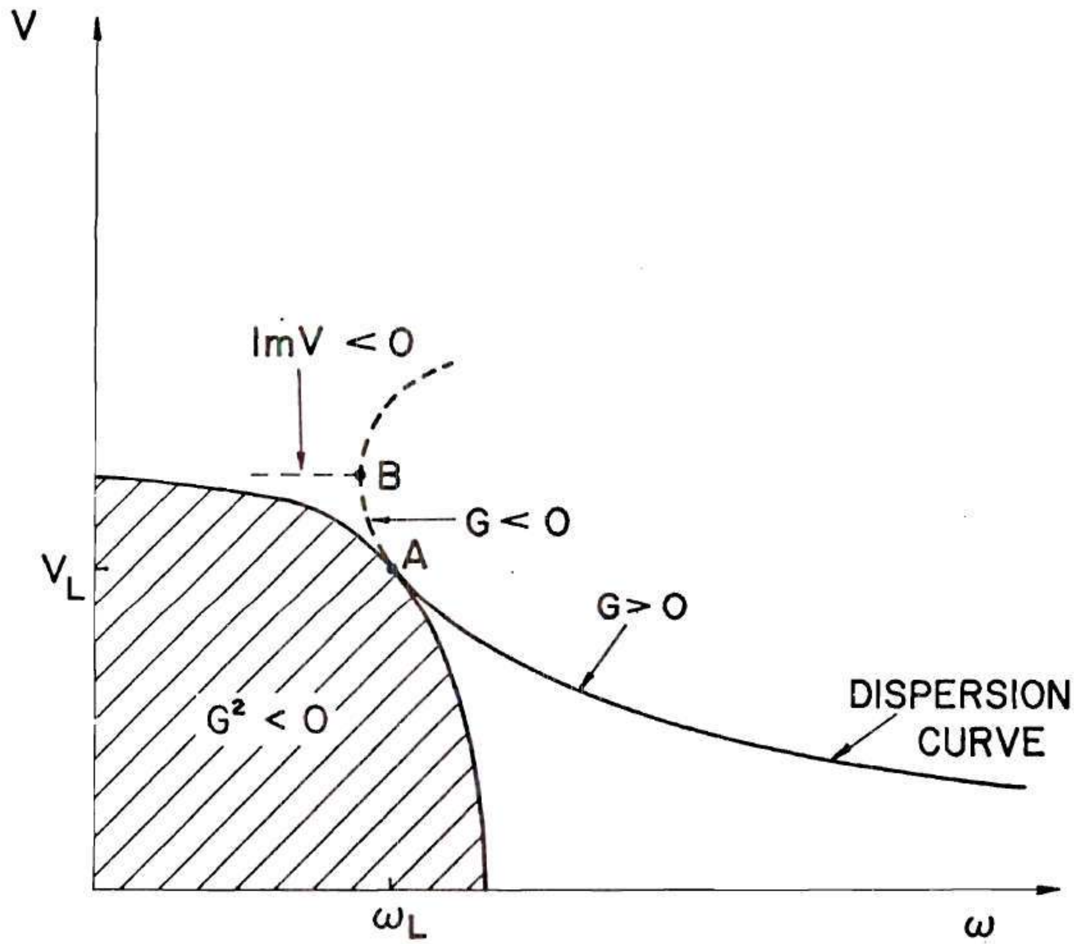

Figure 6. Sketch Illustrating Nature of a Dispersion Curve in the Vicinity of the Line $G^2 = 0$.

Table 2. Parameters Characterizing the Eigenmode Dispersion Function Near the Transition from Leaking to Non-Leaking for the GR_0 and GR_1 Modes.

	GR_0	GR_1
ω_L (rad/s)	0.0118	0.0125
v_L (km/s)	0.31188	0.2323
q_1 (s/km^2)	0.14	0.35
q_2 (s/km^3)	1.84×10^{-3}	1.86×10^{-3}
μ (km)	2.94×10^{-2}	4.15
ν (km)	76	190
ε (km$^{1/2}$/s$^{1/2}$)	9.6×10^{-6}	1.02×10^{-3}
σ (rads/s)	3.04×10^{-13}	1.41×10^{-9}

where $V = \Delta v/[2(\nu - \mu)\sigma]$ and $\Omega = \Delta\omega/\sigma$. Both the real and imaginary parts of V and Ω are shown in the plot. The corresponding plots for the GR_1 mode differ only slightly from those given for the GR_0 mode in Fig. 7. As may be seen from Table 2, $\mu << \nu$ so that, for both modes, the quotient $\mu/[2(\nu - \mu)]$ is small compared to unity.

Concluding Remarks

Since there is a gap in the range of frequencies for which a pole (corresponding to a mode) may exist, it is evident that evaluation of the integral over k in Eq. (2.1) by merely including residues may be insufficient for certain frequencies. Thus it would seem appropriate to include a contribution from branch line integrals. However, there is a line of reasoning which demonstrates that all contributions from branch line integrals are insignificant. Further details on this matter are provided in reference 4.

The investigation described here led to a relatively straightforward perturbation technique for the inclusion of contributions from leaking modes in the synthesis of infrasonic waveforms. It was demonstrated that the imaginary parts of complex horizontal wave numbers can be less than 3×10^{-4} km^{-1}. Consequently, it would be expected that the contributions from leaking modes are significant for realistic propagation distances (i.e., between 1000 and 15,000 km).

In this chapter, a theory of leaking modes has been

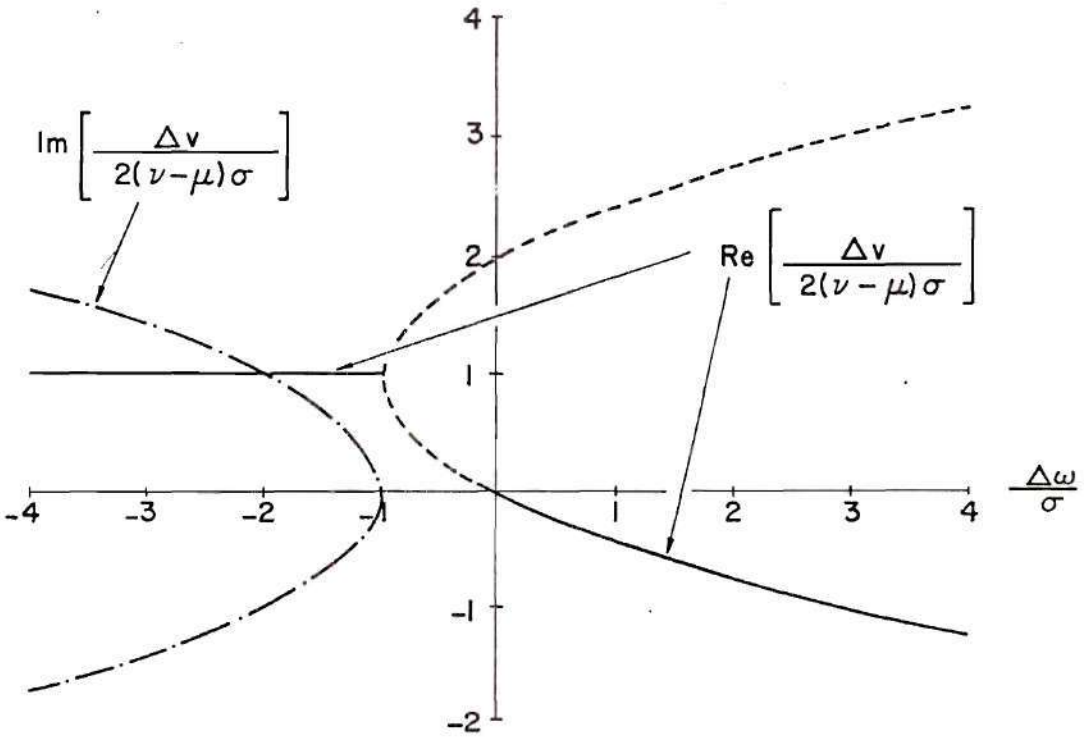

Figure 7. Graph of Normalized Phase Velocity Versus Normalized Frequency in the Vicinity of the Point (v_L, ω_L) for the GR_0 Mode.

presented. The details of the modification of the computer program INFRASONIC WAVEFORMS to incorporate this theory are given in Chapter III.

CHAPTER III

NUMERICAL SYNTHESIS OF WAVEFORMS WHICH INCLUDE LEAKING MODES

Introduction

The computer program INFRASONIC WAVEFORMS[2,5] has been modified to include contributions at low frequencies from leaking modes (specifically the GR_0 and GR_1 modes) to numerically synthesized infrasonic waveforms. The procedure incorporated in this modification involves among other things the calculation (as discussed in Chapter II) of the imaginary and real parts of horizontal wave numbers and phase velocities. The entire procedure for including leaking modes is outlined in detail here. Numbers presented for illustration are appropriate to the case of infrasonic signals observed at 15,000 km distance from a 50-megaton explosion, where the explosion is at three km altitude and the atmosphere [shown in Figs. 8 and 2(a)] is assumed to contain no winds.

Calculation of Complex Wave Numbers and Phase Velocities

The first step in the calculation of complex wave numbers and phase velocities for the GR_0 and GR_1 modes is to obtain values for the phase velocities $v_n(\omega)$, $v_a(\omega)$, and $v_b(\omega)$, and the elements $R_{11}(\omega,v)$ and $R_{12}(\omega,v)$ of the

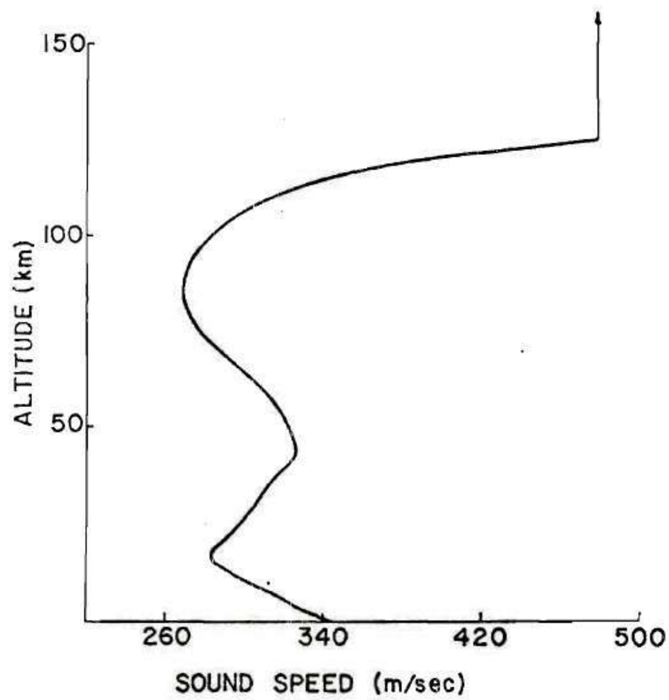

Figure 8. Model Atmosphere Showing Sound Speed Versus Altitude for Numerical Example Treated in the Present Chapter.

transmission matrix [R]. These calculations are done for frequencies below the cutoff frequencies of the two modes. As mentioned in Chapter II, R_{11} and R_{12} depend on atmospheric properties only in the altitude range zero to z_T (the bottom of the upper halfspace) and are independent of what is assumed for the upper halfspace. $v_n(\omega)$ is the phase velocity for a given (n-th) mode for values of ω greater than the lower cutoff frequency ω_L, and $v_a(\omega)$ and $v_b(\omega)$ are values of the phase velocity ω/k at which the functions R_{11} and R_{12}, respectively, vanish. For a given mode, the values of v_a and v_b chosen are those from the curves $v_a(\omega)$ and $v_b(\omega)$ which for $\omega > \omega_L$ lie closest of all such curves to the curve $v_n(\omega)$.

With an alternate version of the subroutine TABLE, INFRASONIC WAVEFORMS may be used to obtain R_{11} and R_{12}. A deck listing of subroutine TABLE with appropriate modifications incorporated is given in Appendix A of reference 5. A deck listing of the input data that is required to calculate R_{11} and R_{12} for the example is given in Fig. 9. Note that only phase velocities between 0.143 and 0.3318 km/sec and frequencies between 0.001 rad/sec and 0.031 rad/sec are used in this calculation. A sample portion of a printout of R_{11} and R_{12} versus phase velocity is given in Fig. 10.

Values of $v_a(\omega)$ and $v_b(\omega)$ for the GR_0 and GR_1 modes are obtained by two successive runs of INFRASONIC WAVEFORMS in which two modified versions of the subroutine NMDFN are

```
$NAM1 NSTART=1, NPRNT=1, NPNCH=-1, NCMPL=-1 $END
$NAM2 IMAX=24,
ZI=1.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.,25.,30.,35.,40.,45.,55.,
    65.,75.,85.,95.,105.,115.,125.,
T=292.,288.,270.,260.,249.,236.,225.,215.,205.,198.,205.,215.,227.,
    237.,249.,265.,260.,240.,205.,185.,184.,200.,250.,400.,570.,
LANGLE=1,
WINDY=25*0.0,
WANGLE=25*0.0
$END
$NAM4
THETKD =35.,
V1 = 0.143, V2 = 0.3318,
ØM1 = 0.001,   ØM2 = 0.031,
NØMI = 30, NVPI = 80,
MAXMØD = 10
$END
$NAM1 NSTART=6, NPRNT=1, NPNCH=-1, NCMPL=-1  $END
```

Figure 9. Listing of Input Data Required to Generate Tabulations of R_{11} and R_{12} Versus Phase Velocity and Angular Frequency.

v_n	R_{11}	R_{12}
OMEGA=	.30928-02	
.14300+00	.21671+01	.65152+02
.14539+00	-.72963-01	.22523+02
.14778+00	-.19992+01	.16898+02
.15017+00	-.34415+01	.49336+02
.15256+00	-.43200+01	.72532+02
.15495+00	-.46324+01	.85619+02
.15734+00	-.44356+01	.88883+02
.15973+00	-.38270+01	.83475+02
.16212+00	-.29260+01	.71114+02
.16451+00	-.18579+01	.53814+02
.16690+00	-.74204+00	.33657+02
.16929+00	.31761+00	.12611+02
.17168+00	.12376+01	-.75995+01
.17407+00	.19579+01	-.25568+02
.17646+00	.24418+01	-.40247+02
.17885+00	.26746+01	-.50952+02
.18124+00	.26605+01	-.57340+02
.18363+00	.24195+01	-.59371+02
.18602+00	.19834+01	-.57261+02
.18841+00	.13917+01	-.51424+02
.19080+00	.68860+00	-.42421+02
.19319+00	-.80574-01	-.30906+02
.19558+00	-.87185+00	-.17582+02
.19797+00	-.16447+01	-.31561+01
.20036+00	-.23637+01	.11690+02
.20275+00	-.29996+01	.26326+02
.20514+00	-.35295+01	.40198+02
.20753+00	-.39379+01	.52832+02
.20992+00	-.42158+01	.63849+02

Figure 10. Sample Printout of R_{11} and R_{12} Versus Phase Velocity for a Fixed Value of Angular Frequency.

used in sequence. These modifications are so minor that they are described here. To obtain $v_a(\omega)$, the third-from-end executable FORTRAN statement of subroutine NMDFN need only be changed from

```
FPP = RPP(1,1)*A(1,2) - RPP(1,2)*(GU + A(1,1))        (3.1)
```

to

```
FPP = RPP(1,1).        (3.2)
```

To obtain $v_b(\omega)$, the same statement need only be changed to

```
FPP = RPP(1,2).        (3.3)
```

The same limits for phase velocity and angular frequency as are used for the calculation of R_{11} and R_{12} are used in the calculations for v_n, v_a, and v_b. In the example, when these limits are used, the GR_1 mode corresponds to mode number three and the GR_0 mode corresponds to mode number four for the case when $v_n(\omega)$ is calculated. For the cases when $v_a(\omega)$ and $v_b(\omega)$ are calculated, the GR_1 mode corresponds to mode number four and the GR_0 mode corresponds to mode number six. A sample listing of $v_n(\omega)$, $v_a(\omega)$, and $v_b(\omega)$ for the two modes is given in Fig. 11. An additional listing of these phase velocities for the two modes is given in Table 3.

GR_0 MODE

ω	v_n
.012375	.31185608
.013407	.31181806
.014438	.31177597
.015469	.31172882
.016501	.31167509
.017532	.31161209
.018563	.31153394
.019070	.31148610
.019079	.31148516
.019595	.31142505
.019853	.31138841
.020111	.31134515
.020626	.31122480
.021658	.31029529
.021659	.31029116
.022005	.30790129
.022139	.30551142
.022173	.30475278
.022240	.30312155
.022329	.30073168
.022412	.29834181
.022490	.29595194
.022566	.29356207
.022639	.29117220
.022689	.28948366

ω	v_a
.001030	.31205939
.002061	.31205552
.003093	.31204906
.004124	.31204001
.005156	.31202834
.006187	.31201405
.007218	.31199710
.008250	.31197748
.009281	.31195515
.010312	.31193006
.011344	.31190215
.012375	.31187139
.013407	.31183768
.014438	.31180093
.015469	.31176104
.016501	.31171786
.017532	.31167120
.018563	.31162087
.019595	.31156653
.020626	.31150781
.021658	.31144415
.022689	.31137478
.023720	.31129855
.024752	.31121368
.025783	.31111721

ω	v_b
.001030	.31209836
.002061	.31209447
.003093	.31208799
.004124	.31207393
.005156	.31206727
.006187	.31205303
.007218	.31203620
.008250	.31201679
.009281	.31199478
.010312	.31197016
.011344	.31194291
.012375	.31191302
.013407	.31188045
.014438	.31184518
.015469	.31180714
.016501	.31176630
.017532	.31172258
.018563	.31167591
.019595	.31162620
.020626	.31157334
.021658	.31151721
.022689	.31145763
.023720	.31139444
.024752	.31132738
.025783	.31125619

GR_1 MODE

ω	v_n
.013407	.22781499
.013624	.22664568
.014040	.22425580
.014424	.22186593
.014438	.22177526
.014778	.21947606
.015107	.21708619
.015413	.21469631
.015469	.21423833
.015699	.21230644
.015966	.20991657
.016217	.20752670
.016453	.20513682
.016501	.20463309
.016675	.20274695
.016886	.20035708
.017085	.19796721
.017274	.19557733
.017454	.19318746
.017532	.19211887
.017626	.19079759
.017790	.18840772
.017946	.18601784
.018096	.18362797
.018240	.18123810

ω	v_a
.001030	.24434330
.002061	.24409612
.003093	.24367787
.003655	.24337478
.004124	.24307887
.005156	.24228453
.006187	.24127431
.006445	.24098491
.007218	.24001984
.008181	.23859504
.008250	.23848240
.009281	.23660913
.009479	.23620517
.010312	.23432748
.010518	.23381529
.011344	.23153728
.011381	.23142542
.012115	.22903555
.012375	.22809942
.012752	.22664568
.013311	.22425580
.013407	.22381942
.013809	.22186593
.014255	.21947606
.014438	.21842295

ω	v_b
.001030	.25073465
.001738	.25054440
.002061	.25042454
.003093	.24990029
.004124	.24915067
.005156	.24815908
.005160	.24815453
.006187	.24690257
.006963	.24576466
.007218	.24535036
.008250	.24346182
.008293	.24337478
.009281	.24118333
.009362	.24098491
.010260	.23859504
.010312	.23844396
.011034	.23620517
.011344	.23514877
.011712	.23381529
.012314	.23142542
.012375	.23116886
.012855	.22903555
.013345	.22664568
.013407	.22632580
.013790	.22425580

Figure 11. A Sample Listing of $v_n(\omega)$, $v_a(\omega)$, and $v_b(\omega)$ for the GR_0 and GR_1 Modes.

Table 3. Tabulation of Frequency-Dependent Parameters for the GR_0 and GR_1 Modes.

GR_0 MODE

ω	v_a	v_b	α	β	A_{11}	A_{12}	G
0.001030	0.31205939	0.31209836	957.1	-2648.5	0.07064925	-1.3492340	0.02861746i
0.005156	0.31202834	0.31206727	917.4	-2783.7	0.07066928	-1.3497015	0.02585957i
0.008250	0.31197748	0.31201679	854.9	-2988.2	0.07070210	-1.3504677	0.02059949i
0.011344	0.31190215	0.31194291	767.9	-3254.2	0.07075075	-1.3515959	$8.16470 \times 10^{-3}i$

ω	X	k_I	k_R	ω/k_R
0.001030	0.14489848 + 0.05869314i	3.29323×10^{-8}	3.3007×10^{-3}	0.31205300
0.005156	0.15887128 + 0.05813477i	1.68605×10^{-7}	0.0165355	0.31202121
0.008250	0.18298964 + 0.05331514i	2.65003×10^{-7}	0.0264444	0.31197553
0.011344	0.22182228 + 0.02559851i	2.00717×10^{-7}	0.0363822	0.31189059

GR_1 MODE

ω	v_a	v_b	α	β	A_{11}	A_{12}	G
0.001030	0.24434330	0.25073465	87.4	-3578	0.13415774	-2.8317742	0.04359249i
0.005156	0.24284530	0.24815908	87.8	-3633	0.13695917	-2.8971705	0.04030849i
0.008250	0.23848240	0.24346182	89.6	-3770	0.14232483	-3.0224265	0.03397304i
0.011344	0.23153728	0.23514877	100.0	-4144	0.15281704	-3.2673565	0.01988061i

ω	X	k_I	k_R	ω/k_R
0.001030	1.9394832 + 0.63020518i	4.96794×10^{-5}	4.0319×10^{-3}	0.25546528
0.005156	1.9560589 + 0.57569611i	2.19268×10^{-4}	0.0204383	0.25269766
0.008250	1.9813366 + 0.47294644i	2.67086×10^{-4}	0.0333205	0.24759561
0.011344	1.9381840 + 0.25214654i	2.05014×10^{-4}	0.0474121	0.23926355

The next step in the calculation of complex phase velocities and wave numbers is to calculate manually values for the parameters α and β which are part of the approximate expression [Eq. (2.9) in Chapter II] for the eigenmode dispersion function. These parameters represent the partial derivatives of R_{11} and R_{12}, respectively, with respect to phase velocity v evaluated at $v = v_a$ and $v = v_b$, respectively. Since R_{11} and R_{12} also depend on ω, α and β may be considered as functions of ω and not of phase velocity.

Recall that $v_a(\omega)$ and $v_b(\omega)$ are values for the phase velocity at which R_{11} and R_{12}, respectively, vanish. From the listing of R_{11} versus v and ω, let the adjacent values R_{111}, R_{211}, R_{311} and R_{411} for R_{11} correspond to the values for phase velocity v_{11}, v_{21}, v_{31} and v_{41}, respectively (for some chosen ω), so that v_{21} and v_{31} bracket a value for v_a. The values R_{211} and R_{311} would then be of opposite sign. In the listing of v, R_{11}, and R_{12} for various ω, the values for v should all turn out to be equally spaced. Given this fact, it is possible to approximate α from the listing of R_{11} by the formula

$$\alpha = (1/\Delta v_1)([5/6]e_{11} + [1/12]f_{11} + [1/4]g_{11}h_{11}), \qquad (3.4)$$

where

$$\Delta v_1 = v_{41} - v_{31} = v_{31} - v_{21} = v_{21} - v_{11}, \qquad (3.5a)$$

$$e_{11} = R_{311} - R_{211}, \tag{3.5b}$$

$$f_{11} = R_{411} - R_{311} + R_{211} - R_{111}, \tag{3.5c}$$

$$g_{11} = (R_{211} - R_{311})/e_{11}, \tag{3.5d}$$

and

$$h_{11} = R_{311} + R_{211} - R_{111} - R_{411}. \tag{3.5e}$$

In like manner, from the listing of R_{12} versus v and ω, let the adjacent values R_{112}, R_{212}, R_{312}, and R_{412} for R_{12} correspond to the values for phase velocity v_{12}, v_{22}, v_{32}, and v_{42}, respectively (for some chosen ω), such that v_{22} and v_{32} bracket a value for v_b. It is then possible to approximate β by the formula

$$\beta = (1/\Delta v_2)([5/6]e_{12} + [1/12]f_{12} + [1/4]g_{12}h_{12}) \tag{3.6}$$

where Δv_2, e_{12}, f_{12}, g_{12}, and h_{12} are defined by equations analogous to Eqs. (3.5) (last subscript changed from '1' to '2').

Because such an approximate method is used to calculate α and β (it would be preferable to have an explicit formula), there is a small amount of false variation in the values obtained. This variation is noticable only for the GR_1 mode and may, for all practical purposes, be eliminated by plotting α and β versus ω and then drawing smooth curves through the respective sets of points (see Figs. 12 and 13). While this graphical procedure is somewhat laborious, it circumvents making additional runs of the computer program to obtain values of R_{11} and R_{12} at more closely spaced values of phase velocity. It also circumvents the elaborate computer programming chore that would be required to calculate α and β automatically. It is suspected that the programming time required for this automation would surpass the time required for manual calculation. In any event, the accuracy of the α and β obtained by Eqs. (3.4) and (3.6) has proven to be more than sufficient.

The complex phase velocity $v^{(1)}(\omega)$ can be calculated by using Eq. (2.10a) in Chapter II. This expression involves the parameters v_a, v_b, and X where X depends on β/α, A_{11}, G, and A_{12} [see Eq. (10b) in Chapter II]. The latter three of these quantities are computed by taking $k^2/\omega^2 = 1/v_a{}^2$ and by using Eqs. (2.3), (2.7a) and (2.7b) of Chapter II, respectively. Listings of G, A_{11}, A_{12}, and X for various values of ω and for the GR_1 and GR_0 modes in the example are given in Table 3.

As explained in Chapter II, below cutoff (e.g., below $\omega_L = 0.0125$ rad/sec for GR_1 and below $\omega_L = 0.0118$ rad/sec for GR_0 in the example) the real part k_R of the horizontal wave number is the real part of $\omega/v^{(1)}$, and the imaginary part k_I is the imaginary part of $\omega/v^{(1)}$. The extension by first iteration of the normal-mode dispersion curves below

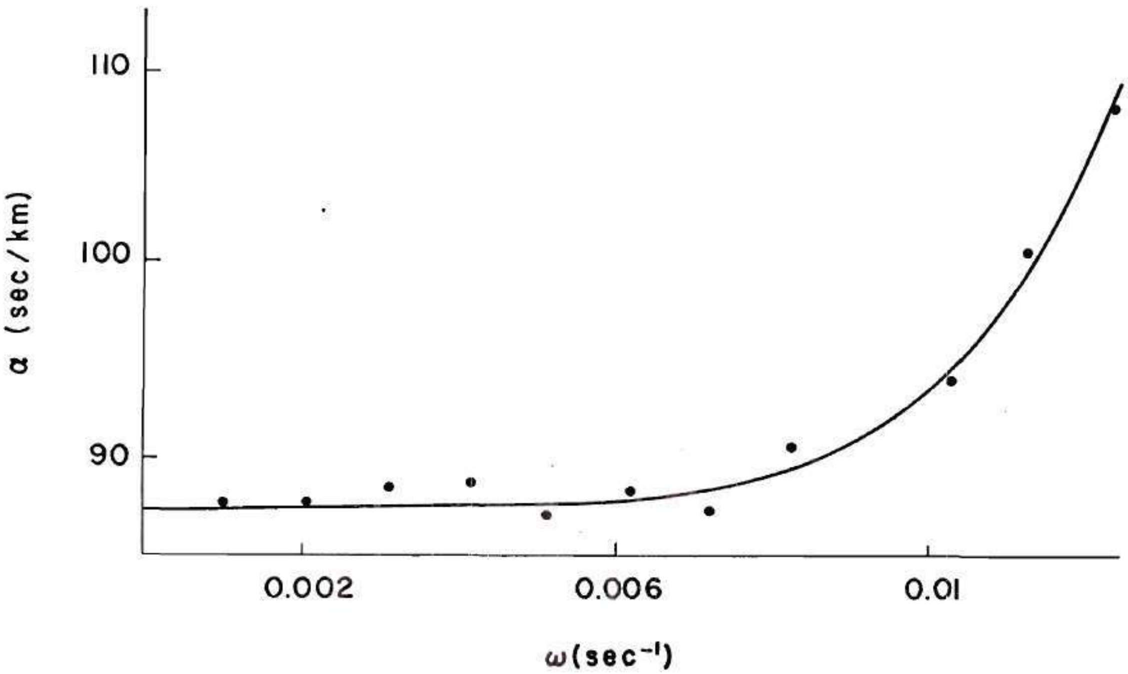

Figure 12. A Plot of the Parameter α Versus ω for the GR_1 Mode.

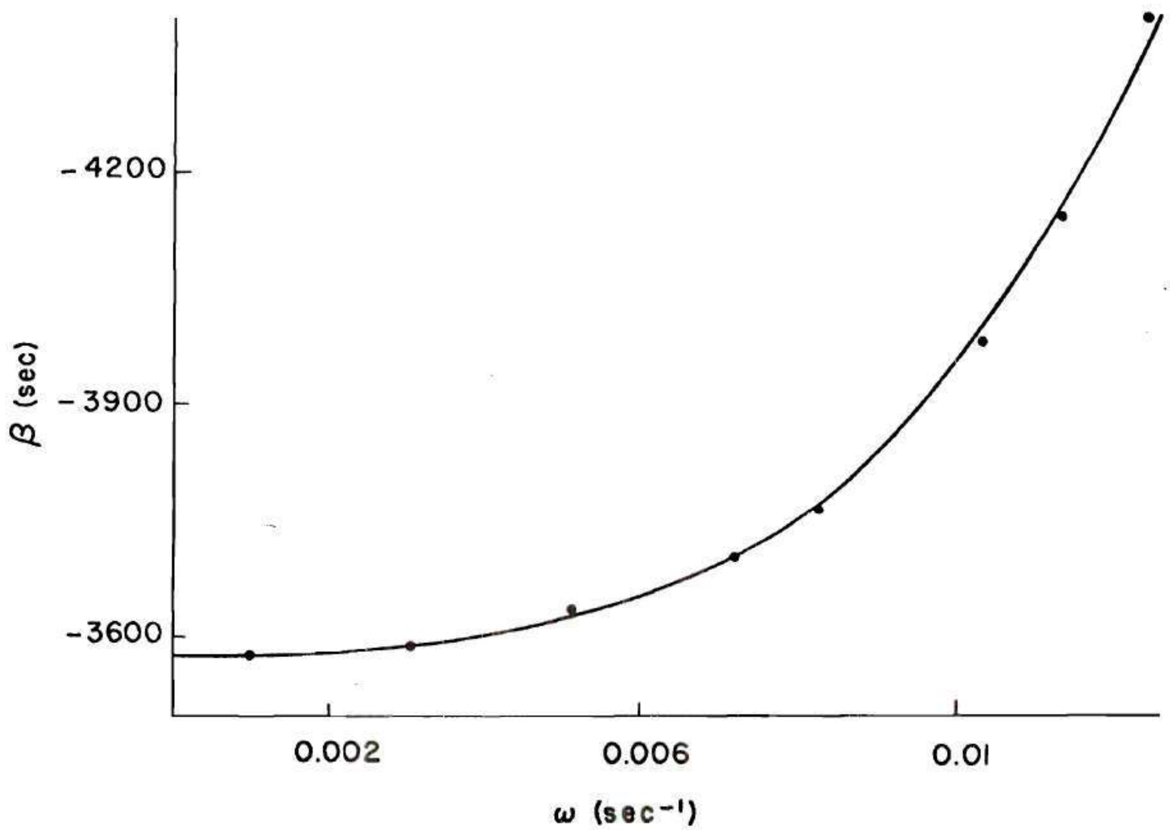

Figure 13. A Plot of the Parameter β Versus ω for the GR_1 Mode.

cutoff is obtained by calculating ω/k_R. Listings of $v^{(1)}$, k_I, k_R, and ω/k_R for various ω for the GR_0 and GR_1 modes in the example are given in Table 3. In addition, plots of k_I and ω/k_R are given in Chapter II in Fig. 5.

Input Data for GR_0 and GR_1

The present version of INFRASONIC WAVEFORMS[5] allows for the phase velocity ω/k_R, imaginary component k_I, and source-free amplitude AMP to be input as functions of angular frequency ω both below and above cutoff for the GR_0 and GR_1 modes. The k_I may be obtained by the procedure described in the previous section. What follows is a description of how the remaining portion of the input data may be obtained.

To obtain values of phase velocity and source-free amplitude at frequencies above cutoff, the current version of INFRASONIC WAVEFORMS is run with the variable NCMPL of NAMELIST NAM1 set less than zero. This run gives an output similar to that which would be obtained with the original version of the program. The input data for this run is the same as if waveforms were being computed without consideration of leaking modes. A listing of such input data which is appropriate to the example is given in Fig. 14. The run with these data will give mode numbers and tabulations of phase velocity VPHSE and amplitude AMP versus angular frequency OMEGA for the GR_0 and GR_1 modes at frequencies above cutoff. The only output which need be retained for future use is the

```
$NAM1 NSTART=1, NPRNT=1, NPNCH=-1, NCMPL=-1  $END
$NAM2 IMAX=24,
ZI=1.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.,25.,30.,35.,40.,45.,55.,
    65.,75.,85.,95.,105.,115.,125.,
T=292.,288.,270.,260.,249.,236.,225.,215.,205.,198.,205.,215.,217.,
    237.,249.,265.,260.,240.,205.,185.,184.,200.,250.,400.,570.,
LANGLE = 1,
WINDY = 25*0.0,
WANGLE = 25*0.0
$END
$NAM4
THETKD = 35.,
V1 = 0.15, V2 = 0.495,
ØM1 = 0.005, ØM2 = 0.1,
NOMI = 30, NVPI = 30,
MAXMOD = 8
$END
$NAM6  ZSCRCE = 3.0, ZØBS = 0.0  $END
$NAM8  YIELD = 50.E3  $END
$NAM10  RØBS = 15000.,
TFIRST = 46.2E3, TEND = 52.2E3,
DELTT = 15.,
IØPT = 11,
$END
$NAM1 NSTART=6 $END
```

Figure 14. Input Data to Obtain Phase Velocity Versus Angular Frequency Above Cutoff Frequency for the GR_0 and GR_1 Modes.

tabulation of VPHSE versus OMEGA for these two modes. Amplitudes at frequencies above cutoff are computed automatically in any run which utilizes this output as input data. A sample tabulation of the pertinent output for the example considered here is given in Fig. 15.

Input data of phase velocity VPHSE and amplitude AMP for frequencies below cutoff may be obtained by a second run of the program with the variable NCMPL set less than zero, but with the original model atmosphere replaced by one which has a thick intermediate layer plus an upper halfspace in place of the original upper halfspace. In other words, in the NAM2 input list, IMAX is increased by one, and the original ZI and T are left unchanged except that a ZI is added which is 100 km greater than the maximum ZI for the original model atmosphere. In addition, the temperature T for the new layer corresponding to IMAX+1 (i.e., for the new upper halfspace) is set to an arbitrarily large value (e.g., 2×10^7 °K). Use of this altered model atmosphere will artificially lower the cutoff frequencies for the GR_0 and GR_1 modes down to values which are very close to zero. In the input data for this second run the angular frequency and phase velocity limits V1, V2, ØM1, and ØM2 of NAM4 must be set to obtain data for the GR_0 and GR_1 modes at frequencies below their original cutoff frequencies. It is imperative that ØM2 not be set too high in value because the program will encounter numerical difficulties at high frequencies when the

GR_0 MODE		GR_1 MODE	
OMEGA	v_n	OMEGA	v_n
.01482759	.31175883	.01482759	.21913010
.01646552	.31167007	.01601253	.20948276
.01728448	.31162838	.01646552	.20500285
.01810345	.31157130	.01711598	.19758621
.01892241	.31150095	.01728448	.19544661
.01933193	.31145750	.01756650	.19163793
.01974138	.31140492	.01796698	.18568966
.02137931	.31079310	.01810345	.18350434
.02151639	.31060345	.01832669	.17974138
.02178879	.30980325	.01865292	.17379310
.02202362	.30762931	.01892241	.16844746
.02210859	.30614224	.01895156	.16784483
.02214435	.30539871	.01909212	.16487069
.02216121	.30502694	.01922762	.16189655
.02217751	.30465517	.01933190	.15953747
.02219828	.30416532	.01948594	.15594828
.02220876	.30391164	.01973352	.15000000
.02223857	.30316810		
.02229504	.30168103		
.02239972	.29870690		
.02259055	.29275862		
.02293273	.28086207		
.02301724	.27771666		
.02324256	.26896552		
.02353065	.25706897		
.02380369	.24517241		
.02406701	.23327586		
.02432538	.22137931		
.02458369	.20948278		
.02465517	.20622217		
.02484741	.19758621		
.02498335	.19163793		
.02512335	.18568966		
.02526862	.17974138		
.02542062	.17379310		
.02558111	.16784483		
.02566520	.16487069		
.02575227	.16189655		
.02593679	.15594828		
.02613807	.15000000		

Figure 15. Sample Output of Phase Velocity Versus Angular Frequency at Frequencies Above Cutoff for the GR_0 and GR_1 Modes.

bottom of the upper halfspace is set as high as considered here. If it were not for this difficulty this second run could be used to generate the same data as is generated in the first run. For comparison, the atmospheric profiles used in the two runs with NCMPL < 0 are shown in Fig. 16.

The second run with NCMPL < 0 gives values for the source-free amplitude AMP and phase velocity VPHSE for the GR_0 and GR_1 modes at frequencies below cutoff. The VPHSE are expected to be close in value to the ω/k_R obtained as described in the previous section. In addition, the source-free amplitudes are expected to match on smoothly above cutoff to those obtained from the first run with NCMPL < 0 even though the model atmospheres used in the two runs are not the same. This expectation is physically reasonable because the energy transported by the GR_0 and GR_1 modes is contained predominantly in the lower atmosphere. Furthermore, these amplitudes should be close in value to those which might be obtained by a perturbation technique similar to that described in Chapter II. Below cutoff, the actual amplitudes should have small imaginary parts. However, in view of the relatively small values obtained for the k_I (less than 10^{-3} neper/km), these imaginary parts may be neglected with confidence. The only characteristic of leaking modes which is of significance in the synthesis of waveforms is the accumulative exponential decay represented by the factor $\exp(-k_I r)$. This factor is retained in subsequent calculations.

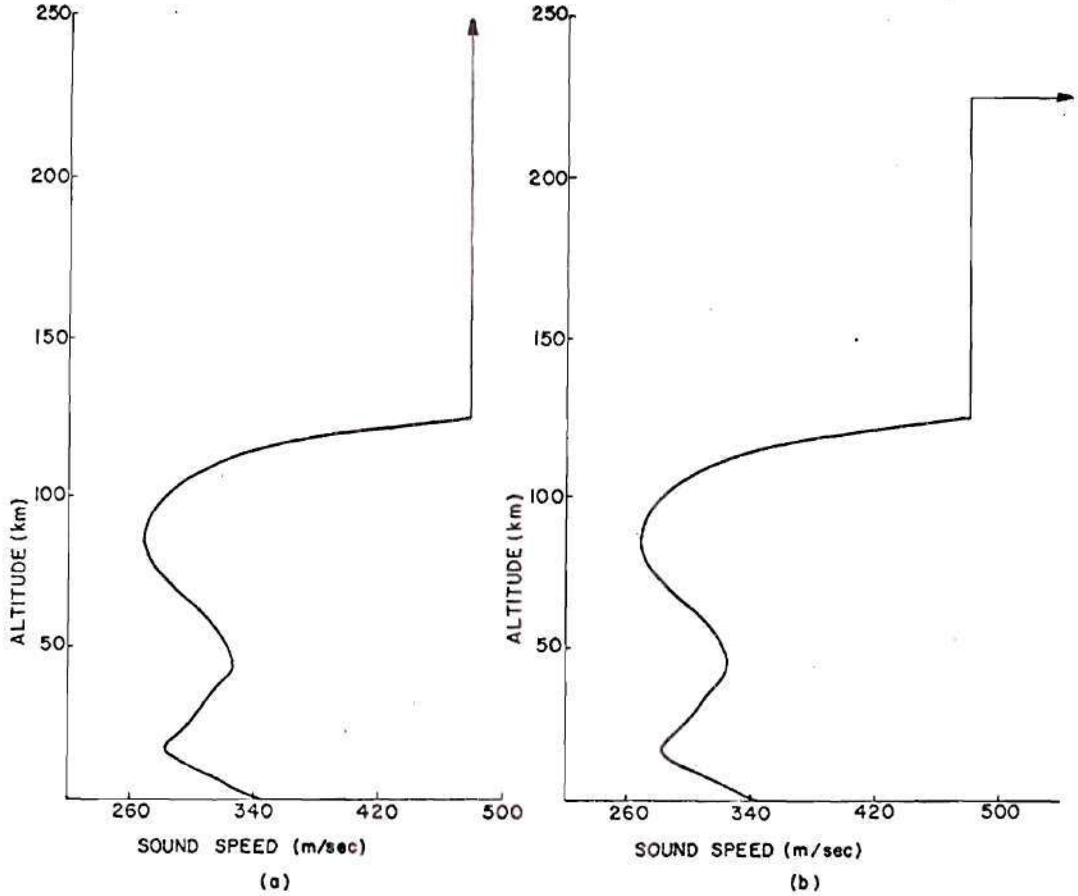

Figure 16. Two Model Atmosphere Profiles. (a) The Same as in Fig. 8. (b) The Same Only with the Original Upper Halfspace Replaced by a Layer of Finite but Large Thickness with a Halfspace Above it of Extremely High Temperature and Sound Speed.

Sample input data for this second run with NCMPL < 0 is given in Fig. 17, and a listing of output values for OMEGA, VPHSE, and AMP below the original cutoff frequencies for the GR_0 and GR_1 modes of the example is given in Fig. 18.

Waveform Synthesis

The final step in the synthesis of waveforms with leaking modes is to run the program INFRASONIC WAVEFORMS with input data that contains data computed for the GR_0 and GR_1 modes as described in the preceding two sections. The only differences between this run and the first run described in the previous section are that here NCMPL > 0 and values are supplied for the variables in the input list NAM51. A listing of the input data for the run with leaking modes which is appropriate to the example is given in Fig. 19. In those data, the AKIGR0 and AKIGR1 are the values of the k_I computed by the perturbation technique of Chapter II as outlined in the first section of this chapter. The source-free amplitudes AMPGR0 and AMPGR1 are taken from the output of the second computer run described in the previous section. The phase velocities VPGR0 and VPGR1 are taken from the outputs of both computer runs described in the previous section. The reason that phase velocities for frequencies below cutoff are used as computed by the first computer run described in the previous section rather than as computed by the perturbation technique of Chapter II is that the values

```
$NAM1 NSTART=1, NPRNT=1, NPNCH=-1, NCMPL=-1  $END
$NAM2 IMAX=25,
ZI=1.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.,25.,30.,35.,40.,45.,55.,
    65.,75.,85.,95.,105.,115.,125.,225.,
T=292.,288.,270.,260.,249.,236.,225.,215.,205.,198.,205.,215.,227.,
    237.,249.,265.,260.,240.,205.,185.,184.,200.,250.,400.,570.,2.E7,
LANGLE=1,
WINDY=26*0.0,
WANGLE=26*0.0
$END
$NAM4
THETKD= 35.,
V1 = 0.18, V2 = 0.34,
ØM1 = 0.001, ØM2 = 0.02,
NØMI = 30, NVPI = 30,
MAXMØD = 8
$END
$NAM1 NSTART=6 $END
```

Figure 17. Input Data to Obtain Phase Velocities and Source Free Amplitudes Below the Cutoff Frequencies for the GR_0 and GR_1 Modes.

GR_0 MODE

OMEGA	VPHSE	AMP
.00100	.31206	-.03102934
.00166	.31205	-.03101968
.00231	.31205	-.03100520
.00297	.31205	-.03098589
.00362	.31204	-.03096170
.00428	.31203	-.03093260
.00493	.31203	-.03089855
.00559	.31202	-.03085951
.00624	.31201	-.03081546
.00690	.31200	-.03076637
.00755	.31198	-.03071222
.00821	.31197	-.03065299
.00853	.31196	-.03062146
.00886	.31196	-.03058865
.00952	.31194	-.03051919
.01017	.31192	-.03044457
.01083	.31190	-.03036475
.01148	.31188	-.03027970
.01214	.31186	-.03018936
.01279	.31184	-.03009365
.01345	.31182	-.02999249
.01410	.31179	-.02988574
.01476	.31176	-.02977324
.01541	.31173	-.02965474
.01607	.31170	-.02952988
.01672	.31166	-.02939809
.01738	.31162	-.02925846
.01803	.31158	-.02910932
.01869	.31152	-.02894743
.01934	.31146	-.02876557
.02000	.31136	-.02854424

GR_1 MODE

OMEGA	VPHSE	AMP
.00100	.28308	-.00003660
.00166	.28237	-.00003722
.00231	.28129	-.00003831
.00297	.27983	-.00004009
.00317	.27931	-.00004082
.00362	.27797	-.00004295
.00428	.27567	-.00004754
.00473	.27379	-.00005235
.00493	.27289	-.00005510
.00559	.26958	-.00006819
.00582	.26828	-.00007507
.00624	.26569	-.00009291
.00668	.26276	-.00012320
.00690	.26116	-.00014672
.00740	.25724	-.00024331
.00755	.25598	-.00029422
.00805	.25172	-.00063749
.00821	.25040	-.00084929
.00853	.24780	-.00156605
.00878	.24621	-.00225436
.00886	.24571	-.00248871
.00937	.24345	-.00335025
.00952	.24292	-.00346229
.01017	.24075	-.00365399
.01019	.24069	-.00365562
.01083	.23860	-.00365194
.01148	.23628	-.00358599
.01178	.23517	-.00354504
.01214	.23372	-.00348656
.01279	.23084	-.00336176
.01304	.22966	-.00330833
.01345	.22758	-.00321275
.01406	.22414	-.00305033
.01410	.22387	-.00303760
.01476	.21961	-.00283239
.01490	.21862	-.00278409
.01541	.21469	-.00259141
.01561	.21310	-.00251310
.01607	.20895	-.00230706
.01621	.20759	-.00223902
.01672	.20220	-.00196998
.01674	.20207	-.00196321
.01720	.19655	-.00168722
.01738	.19420	-.00156992
.01761	.19103	-.00141297
.01798	.18552	-.00114281
.01803	.18462	-.00109941
.01831	.18000	-.00087957

Figure 18. Sample Output of Phase Velocity and Source Free Amplitude at Frequencies Below Cutoff for the GR_0 and GR_1 Modes.

```
$NAM1 NSTART=1,NPRNT=1, NPNCH=-1,NCMPL=1  $END
$NAM2 IMAX=24,
ZI=1.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20.,25.,30.,35.,40.,45.,55.,
    65.,75.,85.,95.,105.,115.,125.,
T=292.,288.,270.,260.,249.,236.,225.,215.,205.,198.,205.,215.,227.,
    237.,249.,265.,260.,240.,205.,185.,184.,200.,250.,400.,570.,
LANGLE=1,
WINDY=25*0.0,
WANGLE=25*0.0,
$END
$NAM4
THETKD=35.,
V1 = 0.15, V2 = 0.495,
ØM1 = 0.005, ØM2 = 0.1,
NØMI = 30,  NVPI = 30,
MAXMØD = 8,
$END
$NAM51  MNGR1=2, NPGR1=25, MNGRO=3, NPGRO=47,
ØMGR1=0.001,0.00231,0.00428,0.00582,0.00805,0.01017,0.01083,0.01178,
    0.01483,0.01592,0.01647,0.01706,0.01729,0.01752,0.01793,0.0181,
    0.0183,0.01864,0.01892,0.01922,0.01933,0.01935,0.01948,0.01961,
    0.01974,
VPGR1-0.28308,0.27983,0.27567,0.26828,0.25122,0.24075,0.23860,0.23517,
    0.21913,0.21034,0.205,0.19828,0.19545,0.19224,0.18621,0.1835,0.18017,
    0.17414,0.16845,0.16207,0.15954,0.15905,0.15603,0.15302,0.15,
ØMGR0=0.001,0.00231,0.00428,0.00624,0.00821,0.01017,0.01083,0.01483,0.01647,
    0.01728,0.0181,0.01892,0.01933,0.01974,0.02138,0.02177,0.02207,0.02214,
    0.02216,0.02218,0.02219,0.0222,0.02221,0.02227,0.02233,0.02253,0.02288,
    0.02302,0.0232,0.02349,0.02377,0.02404,0.02430,0.02456,0.02466,0.02483,
    0.02497,0.02511,0.02526,0.02541,0.02547,0.02575,0.02584,0.02588,0.02593,
    0.02603,0.02614,
VPGR0=0.31206,0.31205,0.31203,0.31201,0.31197,0.31192,0.31190,0.31176,
    0.31168,0.31163,0.31157,0.3115,0.31146,0.31141,0.31079,0.30991,0.30689,
    0.30539,0.30501,0.30463,0.30526,0.30417,0.30388,0.30237,0.30086,0.29483,
    0.28276,0.27772,0.27069,0.25862,0.24655,0.23448,0.22241,0.21034,0.20622,
    0.19828,0.19224,0.18621,0.18017,0.17414,0.17177,0.16207,0.15905,0.15761,
    0.15603,0.15302,0.15,
AMPGR1=-0.00003660,-0.00004009,-0.00004754,-0.00007507,-0.00063749,
    -0.00365399,-0.00365194,-0.00354504,
AMPGR0=-0.03102934,-0.03100520,-0.0309326,-0.03081546,-0.03065299,
    -0.03044457,-0.03036475,
AKIGR1=4.0E-5,9.0E-5,1.75E-4,2.4E-4,27E-4,2.5E-4,2.25E-4,1.4E-4,17*0.0,
AKIGR0=3.0E-8,6.0E-8,1.2E-7,1.9E-7,2.5E-7,2.7E-7,2.3E-7,40*0.0,
$END
$NAM6  ZSCRCE=3.0,  ZØBS=0.0   $END
$NAM8  YIELD=50.E3   $END
$NAM10  RØBS = 15000.,
    TFIRST=46.2E3, TEND=52.2E3,
    DELTT=15.,
    IOPT=11
$END
$NAM1  NSTART=6,  $END
```

Figure 19. Sample Input Data for Synthesis of Infrasonic Waveform Including Leaking Modes.

obtained from the computer run are expected to be more accurate. The values of k_I have to be computed by the technique of Chapter II since the computer program in its present form will not compute them directly.

In Fig. 20 plots are shown for the example of modal and total waveforms obtained with and without leaking modes. Note that the inclusion of leaking modes has eliminated the spurious precursor in the waveform and has raised the amplitude of the first peak. It is also important to note that the waveform with leaking modes begins with a pressure rise, which is realistic.

Further Example (Housatonic)

As a further example, waveforms were computed to model the case of signals observed at Berkeley, California, following the Housatonic detonation at Johnson Island on October 30, 1962. A comparison of theoretical and observed waveforms for this case is given by Pierce and Posey.[10] This case also serves as the main example in the 1970 AFCRL report by Pierce and Posey,[2] and is discussed by Posey[15] within the context of the theory of the Lamb edge mode.

The model atmosphere assumed (winds included) for the computation here is the same as in Fig. 3-12 of reference 2, except that in the present model the upper halfspace begins at 125 km (IMAX = 24) rather than at 225 km (IMAX = 33). To avoid repeating tedious calculations of the k_I for the GR_0

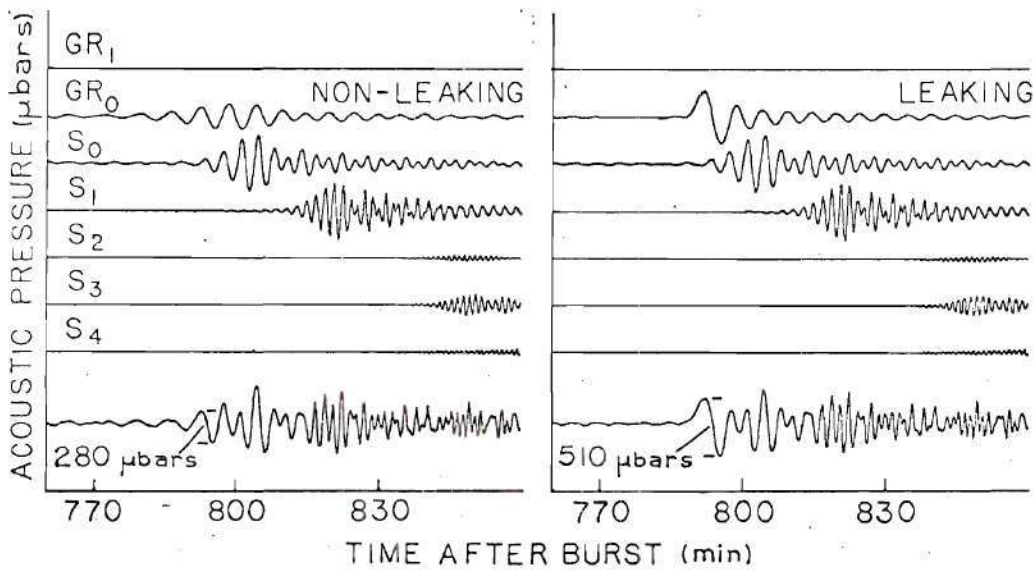

Figure 20. Plots of Modal and Total Waveforms Before and After Inclusion of Leaking Modes (50-Megaton Burst).

and GR_1 modes for this model atmosphere, it was assumed that the k_I would be close in value to those calculated for the example used in the previous sections.

In Fig. 21, sets of plots for the Housatonic case are shown with and without leaking modes. The set with leaking modes excluded does not agree with comparable plots in Fig. 3-10 of reference 2. This relative disagreement exists because the upper halfspace has been taken here to begin at a lower altitude. In spite of this disagreement, the waveform that includes leaking modes is regarded as an improvement in that among other things the spurious initial pressure drop shown in the original waveform is not present here.

In Fig. 7 of reference 10 observed and theoretical waveforms are shown for the Housatonic case. On the basis of the calculations described in this chapter, this figure was redrawn and is given here as Fig. 22. The only difference between the two figures lies in the central waveform. The false precursor is absent in the waveform shown in Fig. 22, and the first peak to trough amplitude has been changed from 157 μbar to 170 μbar (less than a 10% increase). The remainder of the central waveform is virtually unchanged. The discrepancy with the edge-mode synthesis has not been diminished and remains a topic for future study.

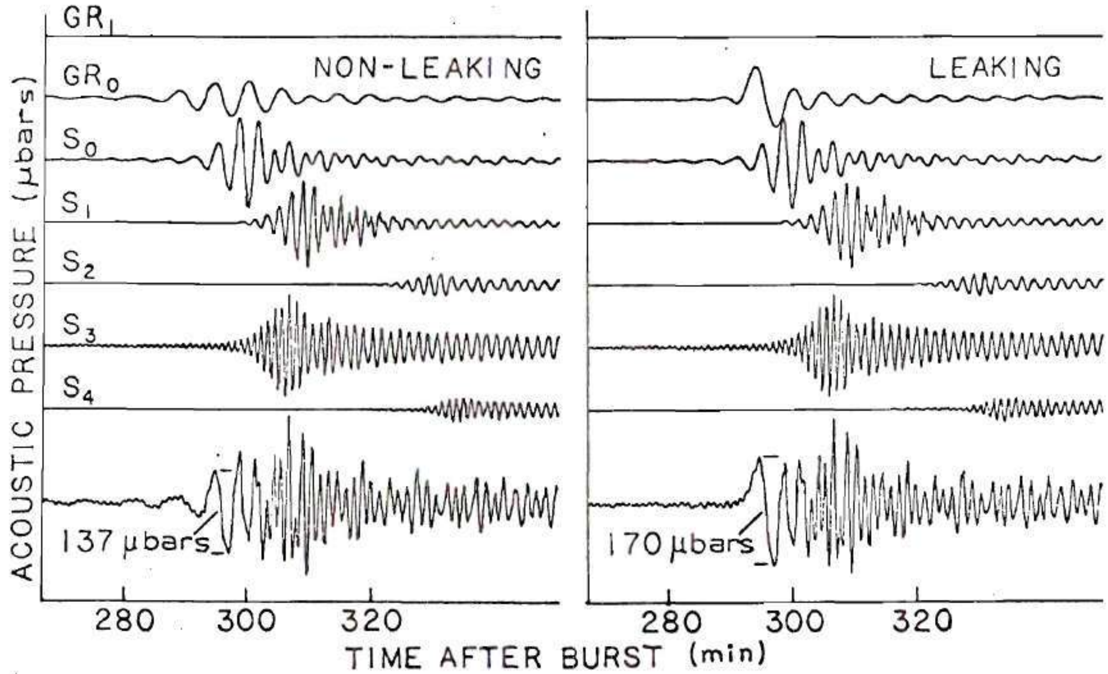

Figure 21. Plots of Modal and Total Waveforms Before and After the Inclusion of Leaking Modes (Housatonic).

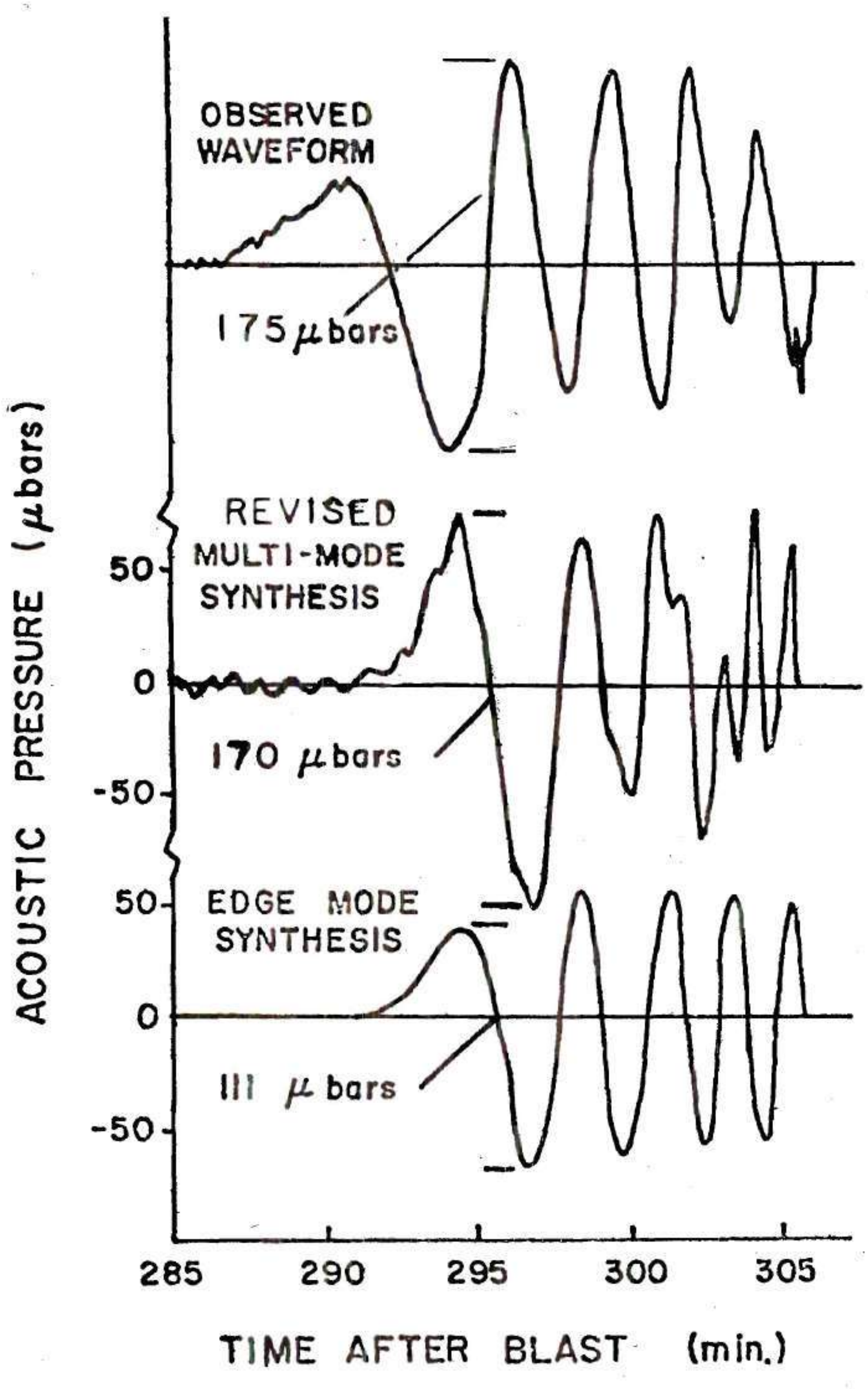

Figure 22. Observed and Theoretical Pressure Waveforms at Berkeley, California, Following the Housatonic Detonation.

CHAPTER IV

ASYMPTOTIC HIGH-FREQUENCY BEHAVIOR OF GUIDED MODES

Introduction

Due to stratification in temperature and wind, the atmosphere possesses sound-speed channels with associated relative sound-speed minima. Fig. 23 shows a standard reference atmosphere wherein two sound-speed channels are indicated, one with a minimum occurring at approximately 16 km altitude and the second with a minimum occurring at approximately 86 km altitude.[15] Given the presence of a channel, an acoustic ducting phenomenon can occur, as is demonstrated in Fig. 24, wherein the energy associated with an acoustic disturbance can become trapped in the region of a relative sound-speed minimum. It is this mechanism of ducting that is of interest here.

In the computer program INFRASONIC WAVEFORMS,[2] the computation of modal waveforms involves the numerical integration over angular frequency of a Fourier transform of acoustic pressure where this integration is truncated at high frequency. It has been speculated that this truncation leads to the generation of what might be called "numerical noise" in the computer output. It was felt useful, therefore,

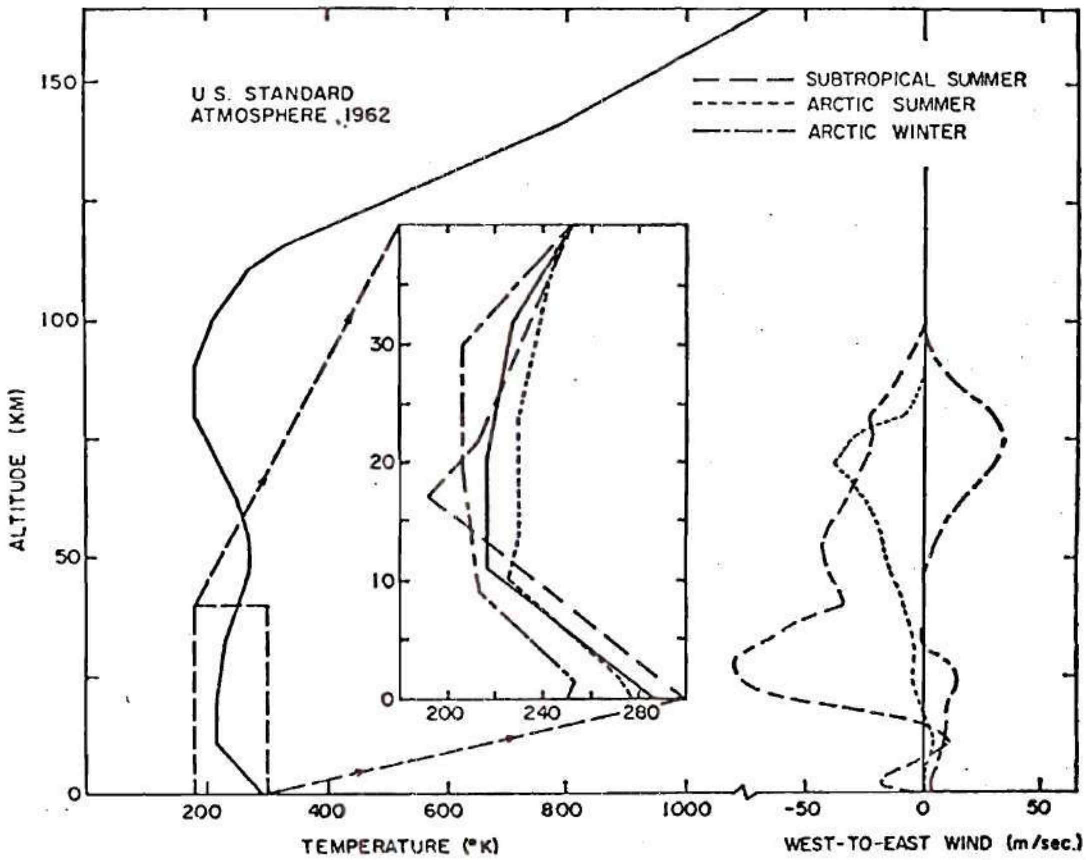

Figure 23. Profiles of Temperature and Wind Speed Versus Height for Standard Reference Atmospheres.

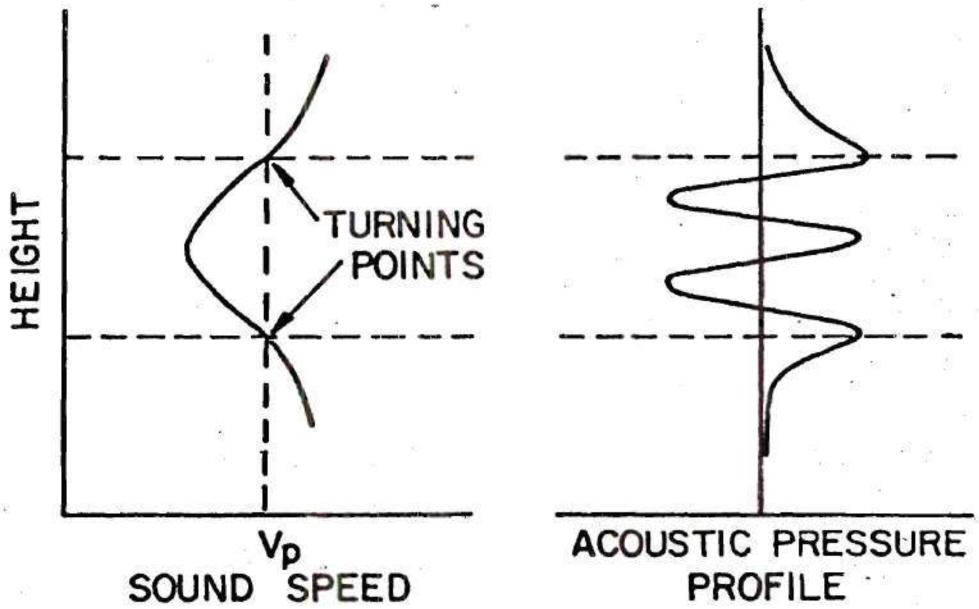

Figure 24. Sketches of Sound Speed and Acoustic Pressure Amplitude Versus Height for a Guided Mode Illustrating the Mechanism of Acoustic Ducting.

to extend this integration beyond the previous upper-angular-frequency limit by means of some high-frequency approximation. In the case of an atmosphere with just one channel, the technique for this extension is well known and dates back to a paper published by N. Haskell[16] in 1951. Haskell's technique involves the W.K.B.J. (Wentzel, Kramers, Brillouin, Jeffreys) method of solution (then in common use in quantum mechanics, although its invention dates back to Carlini[17] and Green[18] in the early 19th century).

The approximations associated with the W.K.B.J. method of solution can be applied to the analytical model on which the computer program INFRASONIC WAVEFORMS is based at frequencies above approximately 0.05 radian/sec (corresponding to periods less than two minutes). Below that frequency, effects due to density stratification in the atmosphere and gravitational forces cannot be neglected. These effects therefore are not germane to the discussion here.

The application of the W.K.B.J. method of solution to the problem of describing propagation of acoustic disturbances in a medium that contains two adjacent sound-speed channels has been discussed in the literature by Eckart.[19] Eckart introduced the technique of devising a W.K.B.J. model for each of the sound-speed channels separately, then combining the results of the two models rather than treating the problem with a single model. In this chapter, Eckart's method is applied to the case of infrasonic waves in the atmosphere.

The W.K.B.J. Model

The W.K.B.J. model for propagation of acoustic disturbances in a single sound-speed channel leads to an approximation for the acoustic pressure p divided by the square root of the ambient density ρ_o as follows:

$$\frac{p}{\sqrt{\rho_o}} = \psi(z)e^{-i\omega t}e^{ikx} \;, \tag{4.1}$$

where ω is angular frequency, k is the wave number associated with the horizontal dimension x, and z is altitude. Here $\psi(z)$ satisfies the reduced wave equation

$$\left[\frac{d^2}{dz^2} + \frac{\omega^2}{c^2(z)} - k^2\right]\psi = 0, \tag{4.2}$$

where c(z) is sound speed as a function of altitude. The W.K.B.J. approximation applies in general to all differential equations of this type if the coefficient of ψ is sufficiently "slowly varying." The approximation would appear to be valid in the present context provided that

$$\frac{c}{|\nabla c|} << \lambda, \tag{4.3}$$

where λ is some representative wavelength of interest.

Eq. (4.3) implies that if the W.K.B.J. model is to apply here, then substantial changes in sound speed should not occur within distances corresponding to a typical wavelength of interest.

Comparison of Dispersion Curves

A particular result of the W.K.B.J. method is that dispersion curves $v(\omega)$ for guided modes can be determined from the equation

$$\int_{z_{bottom}}^{z_{top}} \left[c^{-2} - v^{-2}\right]^{1/2} dz = \frac{(2n + 1)\pi}{2\omega} , \tag{4.4}$$

where v is phase velocity, n = 0, 1, 2, 3, ..., and z_{bottom} and z_{top} identify the lower and upper bounds of the sound-speed channel, respectively.[20]

Particular insight into the high-frequency behavior of guided infrasonic modes in the atmosphere is gained when Eq. (4.4) is solved numerically for both the upper and lower channels (the model atmosphere being that given in Fig. 23 only without winds). The resulting dispersion curves are shown in the lower portion of Fig. 25. One set of curves (the dashed curves) is appropriate to the W.K.B.J. model for the lower channel, and the other set (the solid curves) is appropriate to the W.K.B.J. model for the upper channel. In the upper portion of the same figure dispersion curves are

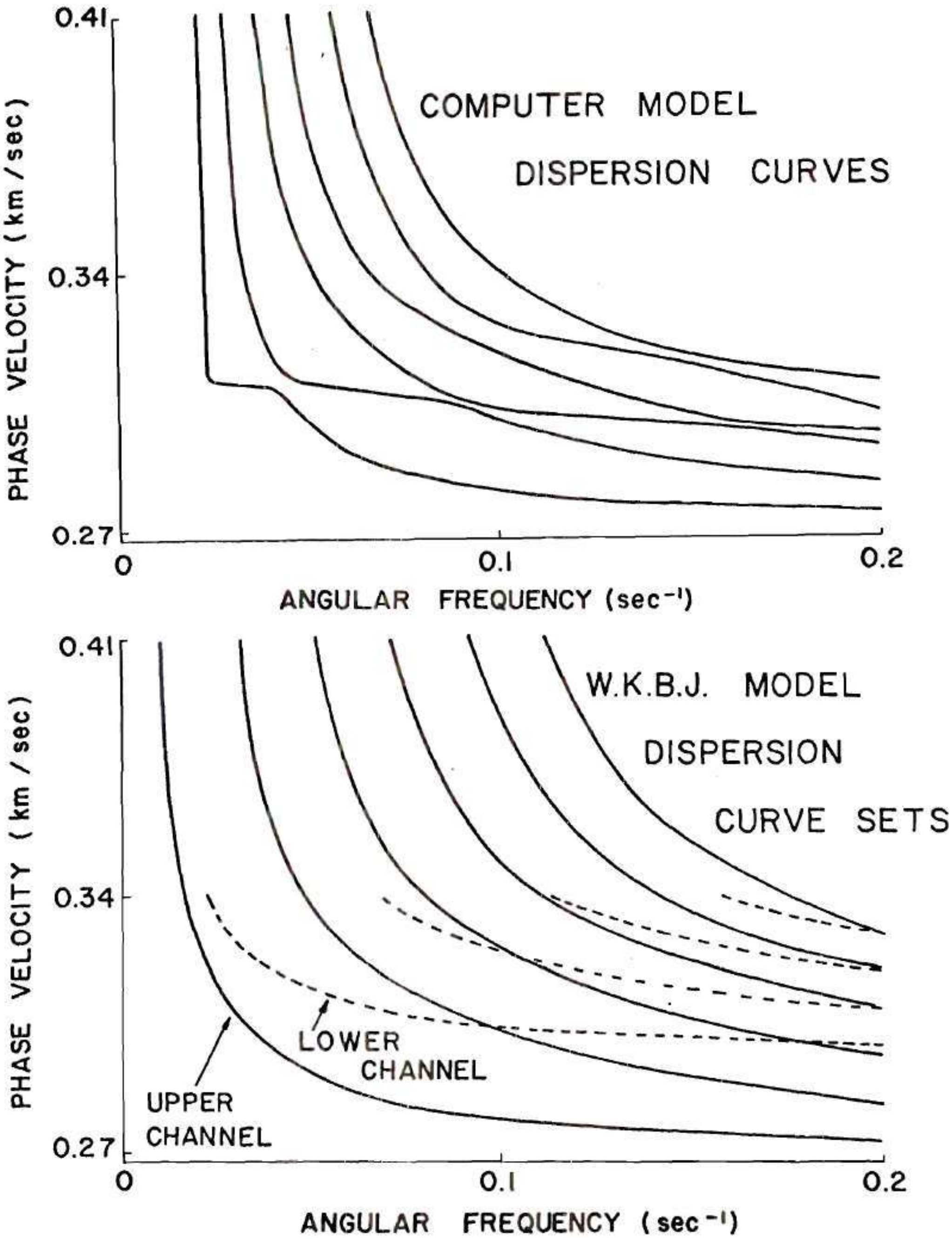

Figure 25. A Comparison of Theoretical Guided-Mode Dispersion Curves for the U. S. Standard Atmosphere, 1962.

shown as generated by the computer model of INFRASONIC WAVEFORMS. The computer model solves a more complex problem in the sense that the approximations inherent in the W.K.B.J. model are not present.[2]

As is illustrated in the lower portion of Fig. 25, the two sets of dispersion curves generated by the W.K.B.J. models intersect at various points. A comparison of the dispersion curves shown in both the upper and lower portions of Fig. 25 reveals that these points of intersection mark regions of near intersection in the (ω, v)-plane between adjacent curves of the computer model. In the right hand portion of Fig. 26, one such region of near intersection is shown (denoted "resonant interaction between adjacent modes") with a corresponding point of intersection between two dispersion curves of the W.K.B.J. models shown to the left. It should be mentioned that the dispersion curves for the computer model never intersect one another. An analytical explanation of this fact has been given by Pierce.[21]

Inferences Concerning the Distribution of Energy with Height

A further comparison of the dispersion curves shown in Fig. 25 reveals that, for relatively high angular frequencies, the dispersion curve corresponding to a given mode of the computer model is comprised of portions of dispersion curves from both sets of the curves generated by the W.K.B.J.

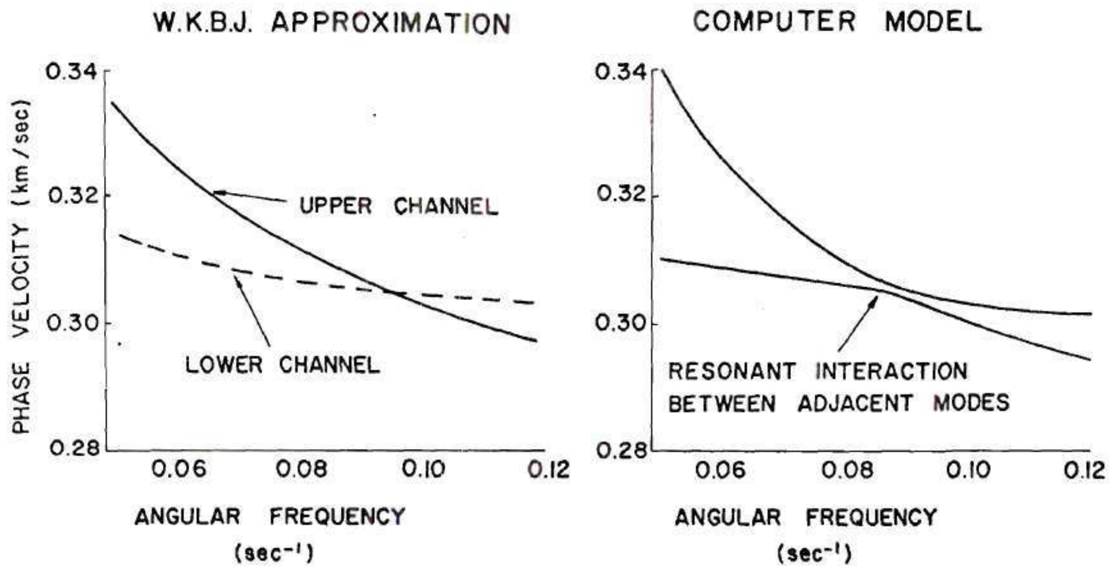

Figure 26. A Detailed Plot of a Section of Fig. 25 Showing a Region of Resonant Interaction Between Two Modes.

models. Two important inferences about the asymptotic high-frequency behavior of guided infrasonic modes can be drawn from this fact. First, for some frequency ranges, and depending on how dispersion curve portions match between curves of the computer model and the W.K.B.J. models, it can be inferred that the acoustic energy associated with a given mode is comprised of energy associated more with propagation of acoustic disturbances in one sound-speed channel than in the other. As frequency increases, this association alternates back and forth between channels. To illustrate, if, for a small range of frequencies, a portion of a dispersion curve of the computer model matches [in the (ω,v)-plane] a portion of one of the curves for the W.K.B.J. model for the upper channel, then this matching implies that, for that mode and for that small frequency range, the acoustic energy density associated with that mode is greater in the upper channel than in the lower channel. Secondly, in the standard reference atmosphere, the sound-speed minimum for the upper channel is less than the sound-speed minimum for the lower channel. It can be inferred, therefore, that those acoustic disturbances for which phase velocities are less than the sound-speed minimum for the lower channel are associated more with acoustic energy trapped in the upper channel than in the lower channel, and thus, for this reason, do not contribute significantly to the acoustic energy at the ground. This second inference implies that care must be taken as to which modes are chosen in the synthesis of a waveform for a ground location, as some may not contribute while others which do may be inadvertently omitted.

Implications for Waveform Synthesis

Currently, in the synthesis of infrasonic waveforms, acoustic guided modes are numbered in order of increasing phase velocity (i.e., S_0, S_1, S_2, ..., etc.) and the sum over modes is truncated at a maximum number of modes.[2] The analysis presented here indicates that this may be a very poor technique for synthesizing high-frequency portions of waveforms for locations near the ground since there is always some frequency above which all of the first N modes correspond to acoustic ducting in the upper sound-speed channel. For the synthesis of ground-level signals from sources below 50 km altitude, a preferable technique would be to ignore the upper sound-speed channel completely for frequencies above, approximately 0.2 rad/sec (possibly 0.1 rad/sec). Dispersion curves could then be taken as given by the W.K.B.J. approximation, and profiles of modal amplitude versus height could be computed by using the method outlined by Haskell.[16] Dispersion curves and amplitudes so computed would fit directly into the general scheme which forms the theoretical basis for the current version of INFRASONIC WAVEFORMS.[2] Altering the technique for synthesis in this manner might eliminate the high-frequency "numerical noise" that is currently present in synthesized waveforms.

CHAPTER V

GEOMETRICAL ACOUSTICAL COMPUTATIONAL MODEL FOR THE PREDICTION OF LONG-RANGE PROPAGATION

Introduction

In this chapter, a description is given of a computational model for the prediction of propagation over long ranges in a medium whose properties vary with height only. This model is based on geometric acoustical concepts and should be applicable for periods less than one minute. To some extent, the model is intended to complement the guided-mode model of propagation which has been discussed in the previous chapters.

The geometric acoustical method of characterizing propagation has a large amount of literature pertaining to it, most of which is concerned with underwater sound. It is not the intent here to discuss the theory associated with the method (that will be assumed to be understood), but rather to present the computational implementation of that theory. Some of the innovations that are introduced here and that are not always included in geometric acoustical models are (1) the use of cubic splines to approximate profiles of sound speed versus height, (2) the inclusion of many acoustic rays which connect source with receiver, (3) a method for computing ray parameters and amplitudes that is based on analytical differentiation of geometric acoustical formulas which are appropriate to a stratified medium, and (4) the inclusion of phase shifts that occur at caustics.

In the most general sense, the propagation medium considered here exists above a flat rigid surface and is stratified with height z with the sound speed c(z) assumed to be a continuous function. For simplicity, it is assumed that no ambient motion of the medium exists with respect to a frame of reference that is attached to the surface (i.e., no winds). In addition, the ambient density (ρ_o) and ambient pressure (p_o) are assumed to be constant throughout (see Fig. 27a). Furthermore, it is assumed that the source is localized at the coordinates $x = 0$, $y = 0$, and $z = z_{SC}$ (see Fig. 27b).

What is of prime interest here is the development of a method for obtaining the acoustic pressure $p(\vec{r},t)$ at moderate distances from a source (greater than 50 km) where $p(\vec{r},t)$ is taken to be the geometric acoustical solution of the wave equation

$$\nabla^2(p/\sqrt{\rho_o}) - (1/c^2)\partial^2(p/\sqrt{\rho_o})/\partial t^2 = -4\pi f(t)\delta(\vec{r} - \vec{r}_{SC}). \quad (5.1)$$

In this equation, $\vec{r}$ is a general vector with x, y, and z components, $\vec{r}_{SC}$ is that vector which locates the source and f(t) is a function which characterizes the time dependence

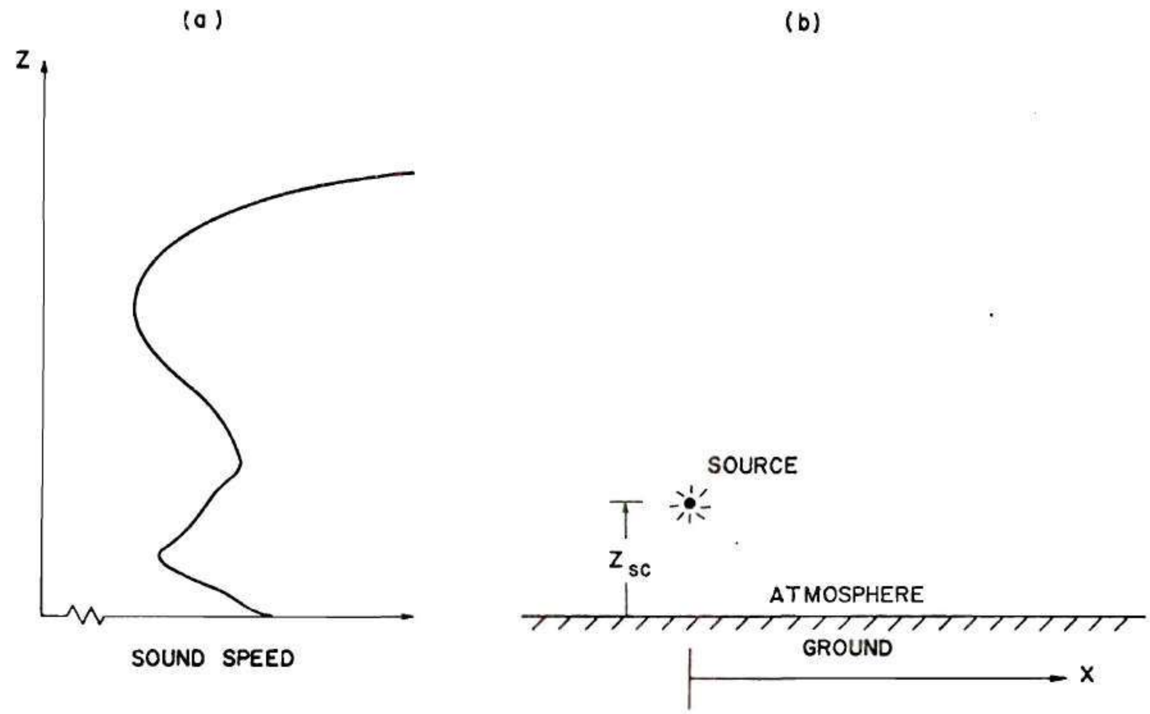

Figure 27. Sketches Illustrating General Propagation Model. (a) Typical Profile of Sound Speed Versus Height. (b) Sketch of Point Source Above a Flat Rigid Ground.

of the source. In addition, $p/\sqrt{\rho_o}$ is taken to satisfy the boundary condition $\partial p/\partial z = 0$ at the ground ($z = 0$).

Acoustic rays are lines that represent paths of propagation which emanate from the source and each of which lies in a vertical plane which contains the source (see Fig. 28). Because of the circular symmetry of the geometry chosen, only those rays that lie in the (x,z)-plane are considered here. A typical ray undergoes refraction. For example, when a ray is proceeding upwards, it will bend downwards if the sound speed in the medium increases with height, or alternatively, the ray will bend upwards if the sound speed decreases with height. Refraction makes it possible for more than one ray to pass through a receiver in the far field. In fact, for long distances of propagation, it would be expected that there be many rays that connect source and receiver. Schemes for computing rays are well known and thoroughly discussed in the literature.[22]

A nonuniform geometric acoustical approximation to the solution of Eq. (5.1) may be taken as

$$p = \sum_{\text{rays}} p_{\text{rays}}, \tag{5.2}$$

where this sum is taken over all rays which connect source and receiver. Individual terms in the sum have signatures and amplitudes which may be computed from the eikonal approximation,[23,24] and from the condition that $p/\sqrt{\rho_o}$ reduces

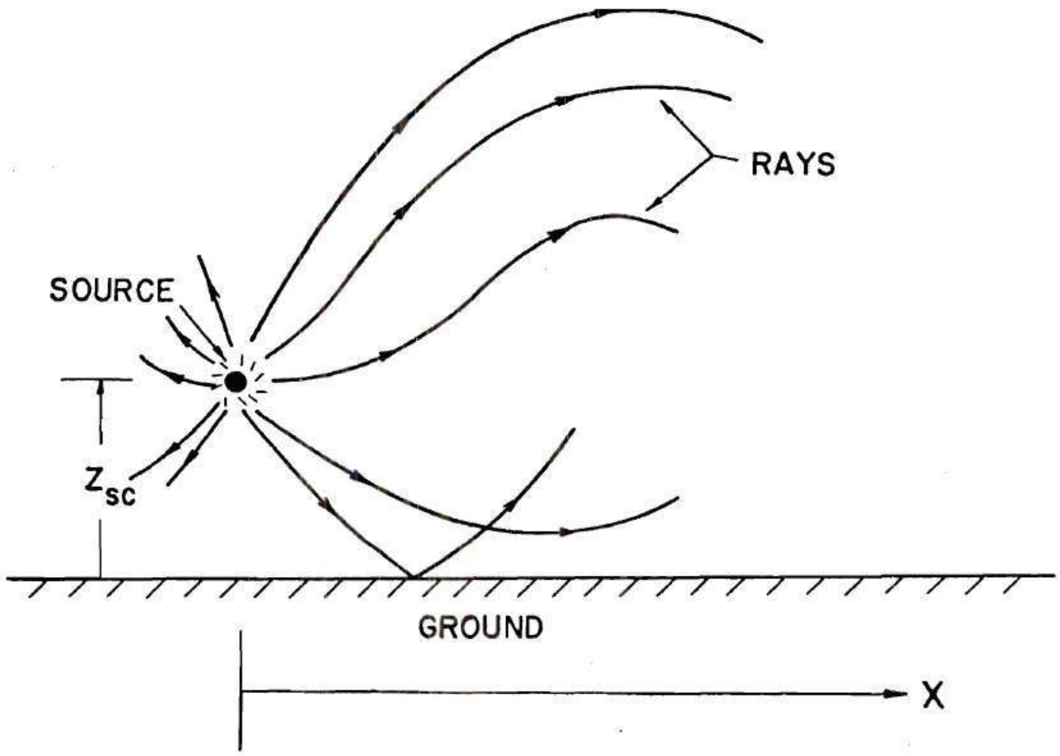

Figure 28. Sketch of Acoustic Rays Emanating from a Source in an Atmosphere in Which the Sound Speed Varies with Height.

to the form $F(t - R/c)/R$ (t being time and R being distance from the source) in the immediate vicinity of the source.

The eikonal approximation is suspect at any point along a ray where ray-tube area vanishes. For the most part, this difficulty with the approximation may be circumvented by including a phase shift of $-\pi/2$ in the signal associated with a ray every time that ray passes through such a point.[25,26] In other words, if a function for a signal is considered to be of the form

$$F(t) = \mathrm{Re}\int_0^\infty \hat{F}(\omega)\ e^{-i\omega t}\ d\omega, \tag{5.3}$$

this function would be replaced by

$$F_{\mathrm{Shift}}(t) = \mathrm{Re}\int_0^\infty e^{+i\pi/2}\hat{F}(\omega)\ e^{-i\omega t}\ d\omega \tag{5.4}$$

upon a single such passage. The shift of $-\pi/2$ is applied each time the ray-tube area goes to zero along a ray, and is in addition to that shift which is due to travel time along a ray. The modeling of successive phase shifts by intervals of $\pi/2$ is relatively straightforward. However, the determination of the number of times that such a shift occurs is more difficult. A method for this determination is provided in this chapter.

For the sake of completeness and versatility in the modeling of propagation over long distances, it is desirable to include explicitly effects that take place at what are termed caustics and lacunae. Lacunae are regions that are characterized by shadow zones (i.e., regions in which there is a relative absence of rays). A caustic is a surface formed by a locus of points at which adjacent rays intersect (or at which ray-tube areas vanish). As mentioned, the eikonal approximation should be suspect in the vicinity of a caustic (indeed it is invalid directly on a caustic).

A lacuna occurs whenever two adjacent rays separate from one another. This separation leaves a region in which there is one less ray than in adjacent regions (refer to Fig. 29). A lacuna can occur when there is a maximum in a sound-speed profile (see LACUNA A in the sketch). A lacuna can also occur near the ground when the sound speed there decreases with height (see LACUNA B in the sketch).

For simplicity, lacunae are not considered here. It seemed more important to investigate first techniques for the inclusion of effects associated with caustics. It is possible to conceive of a hypothetical model atmosphere in which caustics occur, but lacunae do not. Such a model would be one in which the sound speed had a single minimum with height but no maxima, and for which there was no ground (see Fig. 30). While this model may not be wholly realistic, it should suffice for the demonstration of the computational methods presented in this chapter.

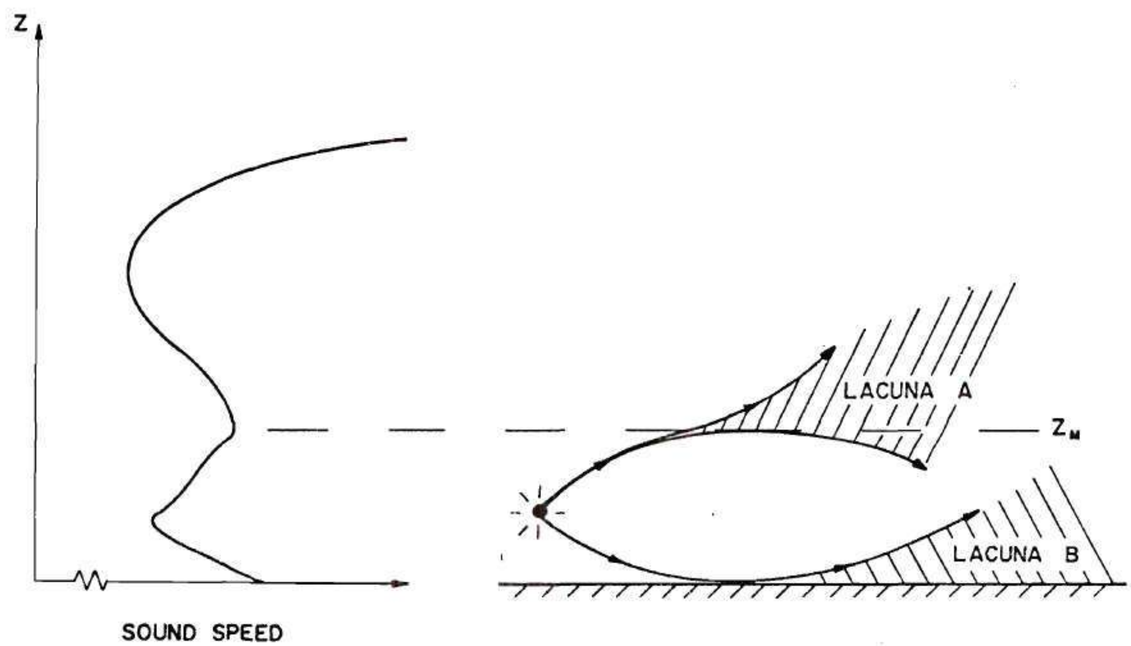

Figure 29. Examples of the Occurrence of Lacunae in the Propagation of Rays from a Source in a Stratified Atmosphere.

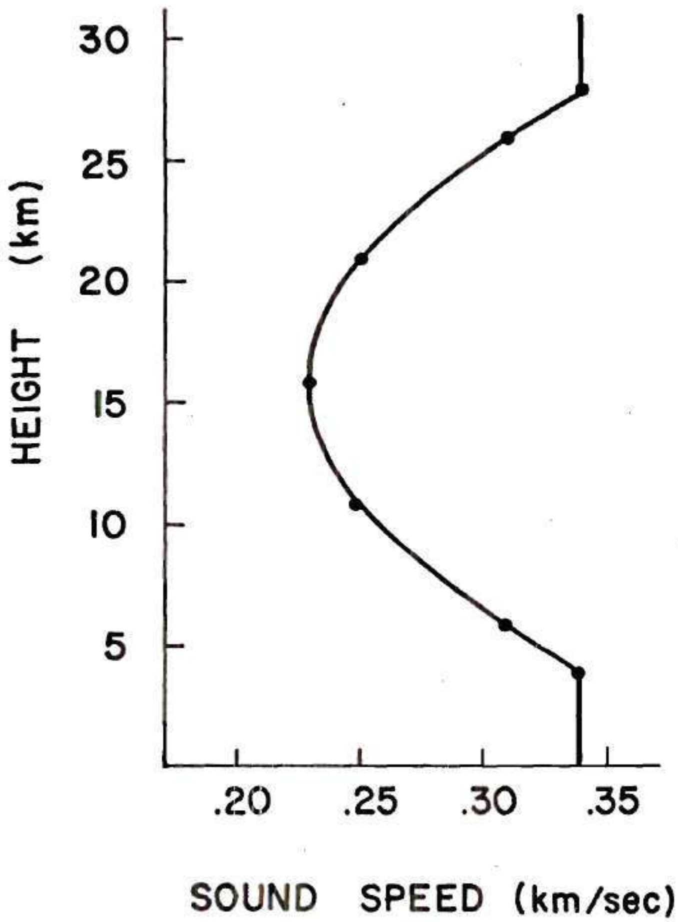

Figure 30. Simplified Hypothetical Sound-Speed Profile with Lattice Points Shown.

In the Appendix, a number of documented FORTRAN subprograms are provided which exemplify the numerical implementation of the computational techniques discussed here. It was not the intent of this investigation of geometric acoustical concepts to devise a completely self-contained computer program for the prediction of acoustic waveforms. Nonetheless, the subprograms were designed to be included in such a program. Emphasis in this chapter is placed on a discussion of computational techniques. A number of simple numerical examples which use the subprograms are presented for illustration.

The Sound-Speed Profile

Typically, in modeling a sound-speed profile, discrete values of sound speed are initially provided [c_i, i = 1,2,3, ..., NCS (NCS meaning number of c's)] which correspond to discrete values of height (z_i, i = 1,2,3,..., NCS). The points (z_i, c_i) are known as lattice points. Lattice points are used to define, by some approximate means, a function $c(z)$ which provides sound-speed values at arbitrary heights. For the calculations often used in geometric acoustical predictions, values of dc/dz and d^2c/dz^2 at arbitrary heights are needed, as well. An interpolation scheme known as the cubic splines method can be used to approximate $c(z)$ and its first two derivatives. This method was recently introduced into the literature on underwater sound by Moler and Soloman.[8]

Using the notation given in that reference, let

$$\Delta z_i = z_i - z_{i-1} \qquad i = 1, \ldots, NCS, \qquad (5.5a)$$

$$\Delta c_i = (c_i - c_{i-1})/\Delta z_i \qquad i = 1, \ldots, NCS, \qquad (5.5b)$$

$$w = (z - z_{i-1})/\Delta z_i \qquad i = 1, \ldots, NCS, \qquad (5.5c)$$

and

$$\bar{w} = 1 - w \qquad i = 1, \ldots, NCS. \qquad (5.5d)$$

Given Eqs. (5.5), the sound speed $c(z)$ for z between z_i and z_{i-1} can be approximated by the cubic polynomial

$$c(z) = \bar{w}c_{i-1} + wc_i + (\Delta z_i)^2[a_{i-1}(\bar{w}^3 - \bar{w}) + a_i(3w^2 - 1)], \qquad (5.6)$$

where the a_i are as defined below. Note from Eq. (5.6) that the sound speed is continuous with height. In particular, when $z = z_i$ and $z = z_{i-1}$, $c(z)$ reduces to c_i and c_{i-1}, respectively.

According to Eq. (5.6), the first, second, and third derivatives of the sound speed are

$$dc/dz = \Delta c_i + \Delta z_i[-a_{i-1}(3\bar{w}^2 - 1) + a_i(3w^2 - 1)], \qquad (5.7)$$

$$d^2c/dz^2 = 6(\bar{w}a_{i-1} + wa_i), \qquad (5.8)$$

and $$d^3c/dz^3 = 6(a_i - a_{i-1})/\Delta z_i, \tag{5.9}$$

respectively, so that

$$dc/dz = \Delta c_i - \Delta z_i(a_i + 2a_{i-1}) \qquad \text{at } z_{i-1}, \tag{5.10}$$

$$= \Delta c_i + \Delta z_i(2a_i + a_{i-1}) \qquad \text{at } z_i, \tag{5.11}$$

$$d^2c/dz^2 = 6a_{i-1} \qquad \text{at } z_{i-1}, \tag{5.12}$$

$$= 6a_i \qquad \text{at } z_i. \tag{5.13}$$

From these equations it can be seen that d^2c/dz^2 is continuous while continuity of dc/dz requires that

$$\Delta c_i + \Delta z_i(2a_i + a_{i-1}) = \Delta c_{i+1} - \Delta z_{i+1}(a_{i+1} + 2a_i) \tag{5.14}$$

for all values of i. Continuity of the third derivative is not imposed on c(z).

As implied by Eq. (5.14), the values for the a_i that are required to insure continuity of dc/dz must be such that

$$a_{i+1} = (\Delta c_{i+1} - \Delta c_i)/\Delta z_{i+1} - 2a_i[1 + \Delta z_i/\Delta z_{i+1}]$$
$$- a_{i-1}\,\Delta z_i/\Delta z_{i+1}. \tag{5.15}$$

Given a_1 and a_2, it is possible to generate all of the succeeding a_i. The linear nature of Eq. (5.15) is such that

$$a_i = K_i + L_i a_2 + M_1 a_1 \tag{5.16}$$

for i > 2, where

$$K_{i+1} = A_i - B_i K_i - C_i K_{i-1}, \tag{5.17a}$$

$$L_{i+1} = -B_i L_i - C_i L_{i-1}, \tag{5.17b}$$

$$M_{i+1} = -B_i M_i - C_i M_{i-1}, \tag{5.17c}$$

$$A_i = (\Delta c_{i+1} - \Delta c_i)/\Delta z_{i+1}, \tag{5.18a}$$

$$B_i = 2[1 + \Delta z_i/\Delta z_{i+1}], \tag{5.18b}$$

$$C_i = z_i/\Delta z_{i+1}, \tag{5.18c}$$

$$K_2 = 0; \qquad K_3 = A_2; \qquad K_4 = A_3 - B_3 A_2, \tag{5.19a}$$

$$L_2 = 1; \qquad L_3 = -B_2; \qquad L_4 = B_3 B_2 - C_3, \tag{5.19b}$$

and

$$M_2 = 0; \qquad M_3 = -C_2; \qquad M_4 = B_3 B_2. \tag{5.19c}$$

Beginning with the values of K_2 and K_3 above, it is possible to generate all of the succeeding K_i.

The boundary conditions on the a_i are taken to be $a_1 = a_{NCS} = 0$. While these boundary conditions may seem arbitrary, they simply require that the sound-speed profile be linear above z_{NCS} and below z_1 (these linear portions being, typically, outside the height range of interest). Given the boundary conditions on the a_i, it follows that

$$a_2 = - K_{NCS}/L_{NCS}. \tag{5.20}$$

The a_i for $i = 3, \ldots, NCS$ can now be computed according to Eq. (5.15).

The numerical implementation of the above computational scheme is realized in the subroutine called DASOL, the deck listing of which is given in the Appendix. When this subroutine is called, the c_i and z_i are presumed to be stored in COMMON. The a_i [denoted by ASOL(I)] are stored in COMMON when DASOL returns.

When the a_i have been computed, the sound speed at a given arbitrary value of z is computed by a function subprogram called CSP(Z). When a value for z is input, this subprogram uses the values for the a_i, the c_i, and the z_i to compute the sound speed at z by Eq. (5.6). In manners analogous to that used in CSP(Z), the function subprograms called DCDZ(Z) and DCDZS(Z) compute dc/dz and d^2c/dz^2, respectively, according to Eqs. (5.7) and (5.8), respectively. The deck listings of CSP(Z), DCDZ(Z), and DCDZS(Z) are also given in the Appendix.

Ray Parameters

For an atmosphere without winds that is vertically stratified in temperature the equations of geometrical acoustics predict that

$$dz/dz = \pm\ c/(v_p^2 - c^2)^{1/2}, \tag{5.21}$$

where x and z are the horizontal and vertical distances, respectively, which define a given ray, and where v_p is the horizontal phase velocity associated with that ray. For any ray, v_p is a constant so that Snell's law (which is a corollary of the ray equations) predicts that, at any point on the ray,

$$v_p = c/(\sin\theta) = \text{constant}, \tag{5.22}$$

where c is the local sound speed and θ is the angle between the momentary ray direction and the vertical (z-axis). The sign convention for Eq. (5.21) is such that dx/dz is positive whenever the ray is moving obliquely upward and negative whenever it is moving obliquely downward. The equations of geometrical acoustics also predict that the rate of change of net travel time t along a given ray with respect to height is

$$dt/dz = \pm\ (v_p/c)/(v_p^2 - c^2)^{1/2}. \tag{5.23}$$

In the collection of FORTRAN subprograms given in the Appendix, the function subprograms RDXDZ(Z) AND RTDTZ(Z) compute the magnitudes $|dx/dz|$ and $|dt/dz|$, respectively. Both of these subprograms use CSP(Z) to find the sound speed value at arbitrary height z. The value for v_p is assumed to be stored in COMMON.

A turning point for a ray is that value of z at which $c(z) = v_p$. In general, when a sound-speed profile contains only one minimum, there are two such turning points, one upper and one lower (denoted ZUP and ZLOW, respectively, in the subprograms). The subroutine TNPNT is used to locate turning points. In TNPNT the horizontal phase velocity VP, and lower and upper height bounds ZBL and ZBU are taken as inputs, and a systematic search is performed between these bounds for the turning points. The search proceeds by dividing the interval (ZBL,ZBU) into NCS + 3 intervals, each of width

$$\text{DELTA} = (\text{ZBU} - \text{ZBL})/(\text{NSCAN} + 1). \tag{5.24}$$

A search for the root of the function CMVP(Z) = CSP(Z) - VP is then conducted by successively examining the sign of

CMVP(Z) at the points ZBL, ZBL + DELTA, ZBL + 2*DELTA, etc., until an interval is found for which the signs of CMVP(Z) at the two interval bounds are opposite. Success at this search implies that a root is bracketed in that interval. The actual value of the root [i.e., the zero of CMVP(Z)] is found by using a library subroutine (see the deck listing of ZREAL2 given in the Appendix). The above search then proceeds to succeeding intervals until a maximum of two roots is found. The output of TNPNT includes NRTS (the number of roots; 0, 1, or 2) and the values ZA and ZB of those roots [ZA is the first root (smallest z), and ZB is the second root (larger z)]. Typically, ZA is expected to correspond to the lower turning point, and ZB to the upper turning point.

In the successive integration between limits (one or both of which are turning points) of expressions such as those given in Eqs. (5.21) and (5.23), care must be taken to insure that these expressions remain real and finite. To insure this, the above search for turning points is supplemented to guarantee that the points are not overshot. For this purpose, another subroutine SHIFT is used to adjust the values of ZA and ZB found by TNPNT to values which are in the immediate neighborhood of these, but which are such that CSP(ZLOW) < VP and CSP(ZUP) < VP where ZLOW is the shifted value for ZA and ZUP is the shifted value for ZB. These adjustments are carried out in units of 10^{-8} until the above criteria are met.

In the subprogram set, the integration in general of any z-dependent quantity between arbitrary limits (not necessarily turning points) is accomplished by the function subprogram called RAINT. For example, in the case of the quantities $|dx/dz|$ and $|dt/dz|$, RAINT performs integration so that

$$\mathrm{RAINT(RDXDZ,ZL,ZU)} = \int_{\mathrm{ZL}}^{\mathrm{ZU}} |dx/dz| \, dz \quad \text{and} \tag{5.25}$$

$$\mathrm{RAINT(RDTDZ,ZL,ZU)} = \int_{\mathrm{ZL}}^{\mathrm{ZU}} |dt/dz| \, dz. \tag{5.26}$$

In the performance of this integration, the range of integration is broken into intervals from ZL to ZAVE and from ZAVE to ZU where ZAVE = (1/2)(ZL + ZU). Thus

$$\mathrm{INTEGRAL} = \int_{\mathrm{ZL}}^{\mathrm{ZAVE}} (\mathrm{INTEGRAND})\, dz - \int_{\mathrm{ZU}}^{\mathrm{ZAVE}} (\mathrm{INTEGRAND})\, dz. \tag{5.27}$$

The reason for separating the integral is that, to perform the actual integration, RAINT uses a library subroutine (see the deck listing of QUAD which is provided in the Appendix) which is most efficient when it integrates away from a singularity. There is the possibility that, as discussed above, the integrand may be singular at the integration limits (e.g., such as is the case with RDXDZ and

RDTDZ at turning points). As will become evident, RAINT is used by a number of subroutines throughout the computational scheme.

In the subroutine RANG, RAINT is used to determine the integrals of $|dx/dz|$ and $|dt/dz|$ between lower and upper turning points. The values of z corresponding to the turning points are supplied as inputs, and the other required information is presumed stored in COMMON. The outputs of RANG are RTIME and RLNTH for the integrals of $|dt/dz|$ and $|dx/dz|$, respectively. These two output parameters are significant because rays for the atmospheric model considered here are periodic in path. For propagation over N ray half-cycles, the travel time is simply (N)*(RTIME), and the horizontal distance traveled is simply (N)*(RLNTH).

Any ray that connects source and receiver may be completely characterized by (1) its associated horizontal phase velocity VP, (2) an index parameter IT (which is one if the ray is proceeding initially obliquely upwards, and minus one if it is proceeding initially obliquely downwards), (3) another index parameter JT (which is one if the ray is proceeding terminally obliquely upwards, and minus one if it is proceeding terminally downwards), (4) the number NUP of upper turning points through which the ray passes, (5) the number NDOWN of lower turning points, (6) the initial height ZSC of the ray (i.e., the source height), and (7) the terminal height ZLIS of the ray (i.e., the receiver height).

The meaning of some of these parameters is graphically illustrated in Fig. 31. It should be noted that, if IT = JT, then NUP = NDOWN, if IT = 1 and JT = -1, then NDOWN = NUP - 1, and if IT = -1 and JT = -1, then NUP = NDOWN - 1.

Given the above parameters, the total horizontal distance which a ray travels can be computed as follows (refer to Fig. 31 again):

$$R = (N)*(RLNTH) + RST + REND, \qquad (5.28)$$

where N is the number of complete half-cycles the ray undergoes given by

$$N = NUP + NDOWN - 1 \qquad (5.29)$$

and where

$$RST = \int_{ZSC}^{ZUP} |dx/dz|\ dz, \qquad IT = 1, \qquad (5.30a)$$

$$= \int_{ZLOW}^{ZSC} |dx/dz|\ dz, \qquad IT = -1, \qquad (5.30b)$$

$$REND = \int_{ZLIS}^{ZUP} |dx/dz|\ dz, \qquad JT = -1, \qquad (5.30c)$$

$$= \int_{ZLOW}^{ZLIS} |dx/dz|\ dz, \qquad JT = 1. \qquad (5.30d)$$

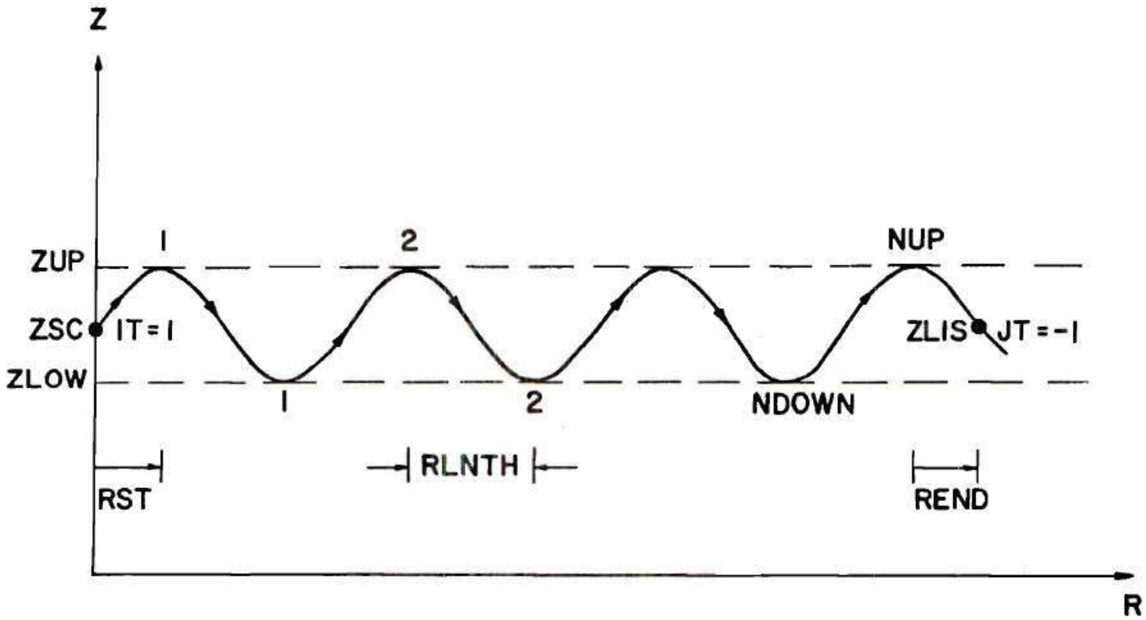

Figure 31. Parameters for Characterizing a Ray.

Eq. (5.28) holds even when NUP and NDOWN are zero. For example, if IT = JT = 1 and NUP and NDOWN are zero, then

$$R = \int_{ZSC}^{ZUP} + \int_{ZLOW}^{ZLIS} - \int_{ZLOW}^{ZUP} |dx/dz|\ dz$$

$$= \int_{ZSC}^{ZLIS} |dx/dz|\ dz. \qquad (5.31)$$

The computation of total range is accomplished by the subroutine TOTRAN. In this subroutine TNPNT is first called to find the turning points, then SHIFT is called to adjust the turning points so that RDXDZ(Z) remains finite throughout the integration range, and then RANG is called to determine the ray half-cycle length RLNTH. The integrals RST and REND are performed with the use of the function subprogram RAINT. The same general scheme used to compute total range can be used in TOTRAN to compute total travel time T, as well. It is only necessary to replace RDXDZ by RDTDZ, RLNTH by RTIME, and R by T in the subroutine.

Rays Connecting Source and Receiver

Given that relevant parameters associated with rays can be computed, a related capability to have in any geometric acoustical computational scheme is that of the identification of all rays which connect source and receiver locations.

Let the source and receiver heights be denoted, respectively, by ZSC and ZLIS as before, and the horizontal distance between the source and receiver be denoted by RANGE. As explained in the previous section, given a realistic set of values for the parameters VP, ZSC, ZLIS, IT, JT, NUP, and NDOWN, it is possible to compute the total range of propagation R associated with these values. Given R, it is possible to define a function RMRAYD(VP) which is the difference between RANGE and R. By holding ZSC and ZLIS fixed, the other parameters VP, IT, JT, NUP and NDOWN can be varied so as to vary R until RMRAYD(VP) vanishes. In doing so, it is possible to define completely a ray that connects the source and receiver. In fact, since there are perhaps several (or in the case of very long ranges, many) groups of values for these parameters such that RMRAYD(VP) vanishes, the above scheme can be used to find all rays that connect source and receiver. A ray type can be thought of as being denoted by IT, JT, NUP, and NDOWN, and a specific ray (given the type) can be thought of as being defined by its associated value for VP.

The function subprogram RMRAYD(VP) computes the above defined difference. In RMRAYD(VP), VP is the independent variable and the remaining necessary parameters are made available through COMMON. To find the values of VP at which

$$RMRAYD(VP) = 0 \tag{5.32}$$

given fixed ZSC, ZLIS, IT, JT, NUP, and NDOWN, the subroutine FNDVP is used. Briefly, FNDVP is used to scan values of VP between the values VPHST and VPHEND at intervals of SDELTA until an interval is found within which RMRAYD(VP) changes sign. Once an interval is found, a library subroutine is called (see ZREAL2 in the Appendix) to find the precise value of the root of RMRAYD(VP). Up to NMAX such roots are found (the number actually found is denoted by NFND), these roots being denoted by VPFND(1), VPFND(2). . .VPFND(NFND). By use of FNDVP, it is possible in principle to find all rays of a given type which connect source and listener. A systematic variation of ray types (IT, JT, NUP, and NDOWN) will, in this manner, identify all the rays that connect source and receiver.

Spreading of Adjacent Rays

Let two coplanar rays, both proceeding initially either obliquely upwards or obliquely downwards, be characterized by phase velocities v_{p1} and v_{p2}. Assuming that v_{p2} is arbitrarily close to (but not equal to) v_{p1}, the separation of the two rays may be characterized by a parameter Δs which (see Fig. 32) is the perpendicular distance from a point on the first ray to the second ray. Δs is positive if the second ray lies above the first, and negative if the

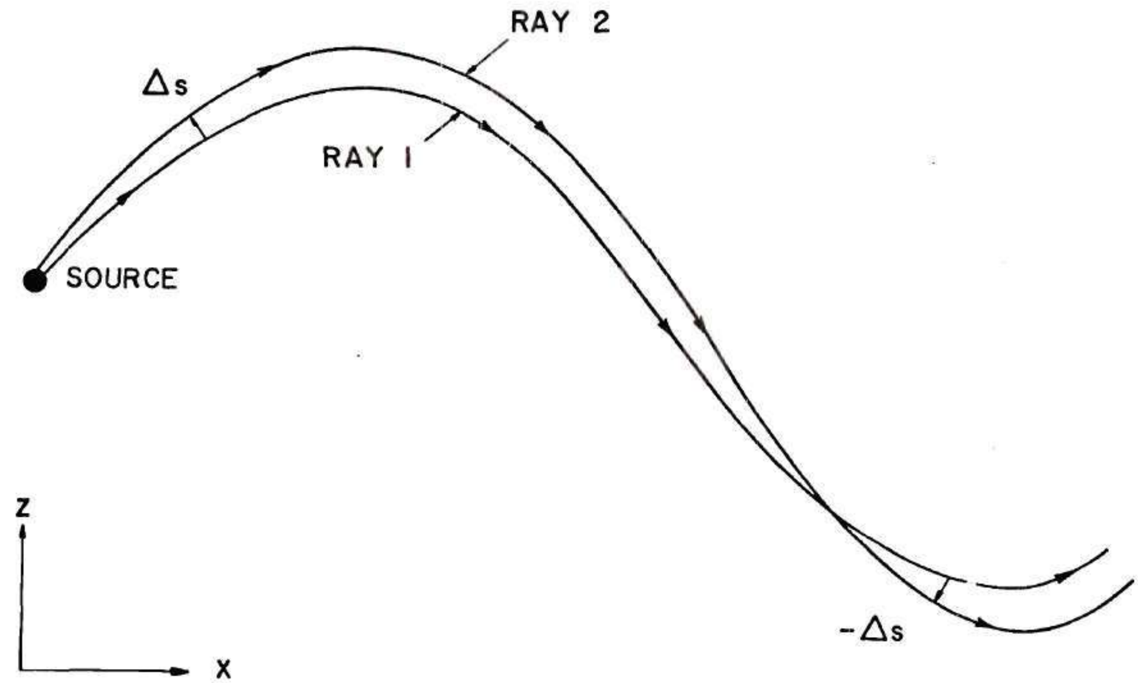

Figure 32. Definition of Parameter Δs Which Characterizes Two Adjacent Rays with Horizontal Phase Velocities v_{p1} and v_{p2}.

reverse is true. The parameter Δs may be considered as a function of horizontal distance x and phase velocity v_p. The limit

$$ds/dv_p = \lim_{v_{p2} \to v_{p1}} \left\{ \Delta s/(v_{p2} - v_{p1}) \right\} \tag{5.33}$$

may be considered to be a uniquely defined function of x, v_p, ray type (IT = 1 or -1), and ray initial height ZSC. The derivative in Eq. (5.33) is termed the ray spreading function. Note that within any ray segment (i.e. between turning points)

$$\begin{aligned} ds/dv_p &= \pm(dx/dv_p)/\{1 + (dx/dz)^2\}^{1/2} \\ &= \pm(dx/dv_p)\{1 - (c/v_p)^2\}^{1/2}, \end{aligned} \tag{5.34}$$

where the plus sign applies if the ray is proceeding obliquely downwards (JT = -1), and the minus sign applies if it is proceeding obliquely upwards (JT = 1). dx/dv_p is the rate of change of the horizontal distance of separation with respect to phase velocity at fixed z and for fixed ZSC. dx/dv_p may be calculated given the general ray type. For a ray that proceeds initially upwards (IT = 1), and which goes through NUP upper turning points and NDOWN = NUP lower turning points, and which ends with the direction of propagation obliquely upwards

$$x = \int_{ZSC}^{ZUP} |dx/dz|\ dz + N \int_{ZLOW}^{ZUP} |dx/dz|\ dz + \int_{ZLOW}^{Z} |dx/dz|\ dz, \tag{5.35}$$

where N = NUP + NDOWN - 1 = 2*(NUP) - 1, and where the integrand |dx/dz| is given by Eq. (5.21). To differentiate this expression with respect to v_p, it is necessary to take into account the fact that ZLOW and ZUP, as well as |dx/dz|, depend on v_p.

In order to evaluate the derivatives with respect to v_p of the integrals in Eq. (5.35), it is necessary to perform integration by parts since singularities arise upon formal differentiation. For this purpose, it is more convenient to rewrite Eq. (5.35) as

$$x = I(ZSC,ZUI) + (N + 1)*I(ZUI,ZUP) +$$

$$+ (N + 1)*I(ZLOW,ZLI) + (N)*I(ZLI,ZUI) + I(ZLI,Z), \tag{5.36}$$

where I(Z1,Z2) represents the integral of |dx/dz| between the indicated limits, ZUI is a fixed (v_p-independent) value of z slightly less than ZUP, and ZLI is a fixed value slightly greater than ZLOW (see Fig. 33). I(ZUI,ZUP) can be rewritten as

$$I(ZUI,ZUP) = \int_{ZUI}^{\infty} U(ZUP - z)|dx/dz|\ dz, \tag{5.37}$$

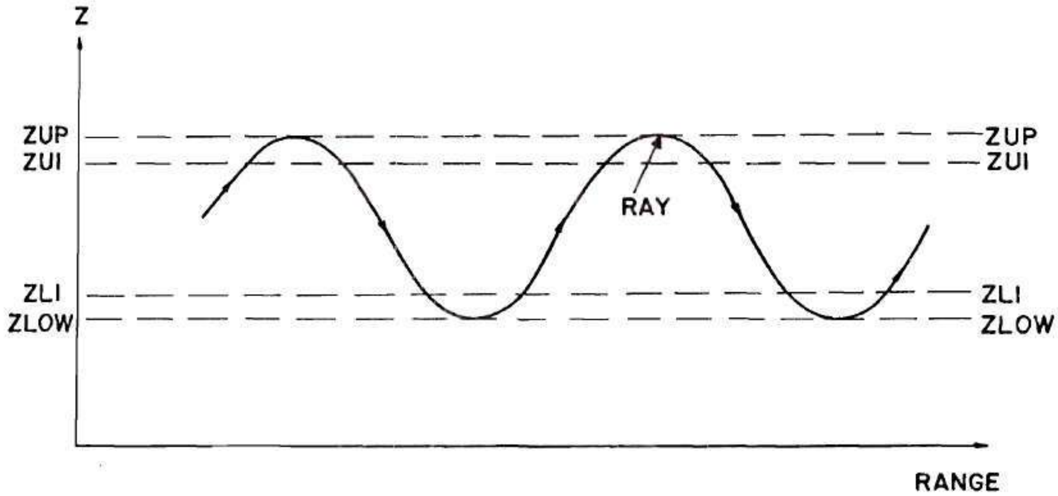

Figure 33. Definition of Parameters ZUI (Slightly Below Upper Turning Point ZUP) and ZLI (Slightly Above Lower Turning Point ZLOW).

where U(ZUP - z) is a step function defined so that

$$U(a - z) = 1, \quad z \le a$$

$$= 0, \quad z > a$$

and where

$$|dx/dz| = -(dc/dz)^{-1}(d/dz)(v_p^2 - c^2)^{1/2}. \qquad (5.38)$$

Integration by parts in Eq. (5.37) gives

$$I(ZUI,ZUP) = [(dc/dz)^{-1}(v_p^2 - c^2)^{1/2}]|_{ZUI}$$

$$+ \int_{ZUI}^{\infty} (v_p^2 - c^2)^{1/2} U(ZUP - z)(d/dz)(dc/dz)^{-1}\, dz. \qquad (5.39)$$

Consequently,

$$(d/dv_p) I(ZUI,ZUP) = [(v_p/c)(dc/dz)^{-1}\, dx/dz]|_{ZUI}$$

$$+ \int_{ZUI}^{ZUP} (v_p/c)|dx/dz|(d/dz)(dc/dz)^{-1}\, dz. \qquad (5.40)$$

Provided that dc/dz does not vanish in the interval between ZUI and ZUP, both of the terms in Eq. (5.40) should be finite.

In a similar manner, it can be shown that

$$(d/dv_p)I(ZLOW,ZLI) = - \{(v_p/c)(dc/dz)^{-1}|dx/dz|\}\Big|_{ZLI}$$

$$+ \int_{ZLOW}^{ZLI} (v_p/c)|dx/dz|(d/dz)(dc/dz)^{-1}\,dz. \qquad (5.41)$$

The derivatives of the remaining terms in the expression for dx/dv_p [Eq. (5.36)] are relatively simple to obtain since the integration limits for these terms are independent of v_p. In particular

$$(d/dv_p)I(ZSC,ZUI) = - \int_{ZSC}^{ZUI} (v_p\ c)(v_p^2 - c^2)^{-3/2}\,dz. \quad (5.42)$$

Thus, the expression for dx/dv_p (IT = 1, JT = 1) can be written

$$dx/dv_p = I1(ZSC,ZUI) + (N + 1)*J1(ZUI) + N + 1)*I2(ZUI,ZUP)$$

$$- (N + 1)*J1(ZLI) + (N + 1)*I2(ZLOW,ZLI) + (N)I1(ZLI,ZUI)$$

$$+ I1(ZLI,Z), \qquad (5.43)$$

where

$$I1(ZA,ZB) = -\int_{ZA}^{ZB} cv_p(v_p^2 - c^2)^{-3/2}\, dz, \tag{5.44a}$$

$$J1(ZA) = \{(v_p/c)(dc/dz)^{-1}|dx/dz|\}\Big|_{z = ZA}, \tag{5.44b}$$

and

$$I2(ZA,ZB) = \int_{ZA}^{ZB} (v_p/c)\,|dx/dz|\,(d/dz)(dc/dz)^{-1}\, dz. \tag{5.44c}$$

In general, for a ray of specified type (IT, JT, NUP, and NDOWN) the corresponding expression for dx/dv_p is

$$dx/dv_p = \begin{Bmatrix} I1(ZSC,ZUI) \\ \\ I1(ZLI,ZSC) \end{Bmatrix} + (2)*(NUP)*J1(ZUI) + (2)*(NUP)*I2(ZUI,ZUP)$$

$$- (2)*(NDOWN)*J1(ZLI) + (2)*(NDOWN)*I2(ZLOW),ZLI)$$

$$+ (NUP + NDOWN - 1)*I1(ZLI,ZUI) + \begin{Bmatrix} I1(ZLI,Z) \\ \\ I1(Z,ZUI) \end{Bmatrix} . \tag{5.45}$$

The upper and lower choices for the first term correspond to IT = 1 and -1, respectively, while the upper and lower choices for the last term correspond to JT = 1 and -1, respectively.

The integrand for the integrals of type I1 is computed by the function subprogram FTRM(Z), and twice the values of those of type I2 are computed by the function subprogram FTRMUL(Z). That is,

$$I1(ZA,ZB) = RAINT(FTRM,ZA,ZB) \tag{5.46a}$$

and

$$I2(ZA,ZB) = RAINT(FTRMUL,ZA,ZB)/2. \tag{5.46b}$$

In addition, the quantity 2[J1(Z)] is computed by the function subprogram TRNPT(Z). In other words,

$$TRNPT(Z) = 2v_p(dc/dz)^{-1}(v_p^2 - c^2)^{-1/2}. \tag{5.47}$$

Thus, the expression for dx/dv_p can be written as

$$\begin{aligned} dx/dv_p &= TRMI + (NUP)*TRNPT(ZUI) \\ &+ (NUP)*RAINT(FTRMUL,ZUI,ZUP) \\ &- (NDOWN)*TRNPT(ZLI) \\ &+ (NDOWN)*RAINT(FTRMUL,ZLOW,ZLI) \\ &+ (NUP + NDOWN - 1)*RAINT(FTRM,ZLI,ZUI) \\ &+ TRMF, \end{aligned} \tag{5.48}$$

where the first and last terms are

$$\mathrm{TRMI} = \mathrm{RAINT(FTRM,ZSC,ZUI)} \quad \text{for} \quad \mathrm{IT} = 1 \tag{5.49a}$$

$$= \mathrm{RAINT(FTRM,ZLI,ZSC)} \quad \text{for} \quad \mathrm{IT} = -1, \tag{5.49b}$$

and

$$\mathrm{TRMF} = \mathrm{RAINT(FTRM,Z,ZUI)} \quad \text{for} \quad \mathrm{JT} = -1 \tag{5.50a}$$

$$= \mathrm{RAINT(FTRM,ZLI,Z)} \quad \text{for} \quad \mathrm{JT} = 1. \tag{5.50b}$$

Finally, ds/dv_p may be calculated from Eq. (5.34) as follows:

$$ds/dv_p = -\mathrm{SIGN(JT)}*(dx/dv_p)[1 - (c/v_p)^2]^{1/2}. \tag{5.51}$$

The sequence of computations described above is carried out by the subroutine CDSDVP. The parameters VP, ZSC, Z, IT, JT, NUP, and NDOWN are inputs, and the output is DSDVP. The parameters ZLI and ZUI are computed internally to CDSDVP and are set to

$$\mathrm{ZLI} = \mathrm{ZLOW} + .01(\mathrm{ZUP} - \mathrm{ZLOW}) \tag{5.52a}$$

$$\mathrm{ZUI} = \mathrm{ZUP} - .01(\mathrm{ZUP} - \mathrm{ZLOW}). \tag{5.52b}$$

The choice of the .01 factor is of course arbitrary. The chief constraint on the use of CDSDVP is that dc/dz should not vanish between ZLOW and ZLI and between ZUI and ZUP.

Along a single ray (with IT = 1) it is apparent that, up to the first upper turning point, ds/dv_p is positive since FTRM(Z) is negative and JT is positive. At the turning point

$$ds/dv_p = \lim_{z \to ZUP} \left\{ [1 - (c/v_p)^2]^{1/2} \int_{ZSC}^{z} dv_p (v_p^2 - c^2)^{-3/2} dz \right\}. \tag{5.53}$$

This limit can be evaluated easily if c is expanded in a power series about its value v_p at z = ZUP so that

$$c \approx v_p + \alpha(z - ZUP), \tag{5.54}$$

where

$$\alpha = (dc/dz)\Big|_{ZUP}, \tag{5.55}$$

and if the integral in Eq. (5.53) is broken into integrals from ZSC to ZUI and from ZUI to z, given that ZUI < z < ZUP. Following these steps, the expression in the braces of Eq. (5.53) becomes

$$\left(\frac{2\alpha}{v_p}\right)^{1/2}(ZUP - z) \left\{ \int_{ZSC}^{ZUI} \frac{[v_p + \alpha(z - ZUP)]v_p}{\{v_p^2 - [v_p + \alpha(z - ZUP)]^2\}^{3/2}} \, dz \right.$$

$$\left. + \int_{ZUI}^{z} \frac{[v_p + \alpha(z - ZUP)]v_p}{\{v_p^2 - [v_p + \alpha(z - ZUP)]^2\}^{3/2}} \, dz \right\} . \qquad (5.56)$$

Thus, in the limit as z approaches ZUP,

$$ds/dv_p = 1/\alpha$$

$$= [1/(dc/dz)]\Big|_{ZUP} \qquad (5.57)$$

which, interestingly, is independent of ZSC.

Between the first upper turning point and the first lower turning point ds/dv_p is given by

$$ds/dv_p = [1 - (c/v_p^2)^2]^{1/2}\{RAINT(FTRM,ZSC,ZUI)$$

$$+ TRNPT(ZUI)$$

$$+ RAINT(FTRMUL,ZUI,ZUP)$$

$$+ RAINT(FTRM,Z,ZUI)\} . \qquad (5.58)$$

It can be shown that Eq. (5.58) may be put in a form which is independent of ZUI so that

$$ds/dv_p = [1 - (c/v_p)^2]^{1/2} \left\{ \frac{(v_p/2)^{1/2}/\alpha^{3/2}}{(ZUP - ZSC)^{1/2}} + \frac{(v_p/2)^{1/2}/\alpha^{3/2}}{(ZUP - z)^{1/2}} \right.$$

$$\left. - \int_{ZSC}^{ZUP} \arg^{(1)}(z_o, ZUP)\, dz_o - \int_{z}^{ZUP} \mathrm{Arg}^{(1)}(z_o, ZUP)\, dz_o \right\}, \quad (5.59)$$

where

$$\mathrm{Arg}^{(1)}(z, ZUP) = \frac{cv_p}{(v_p^2 - c^2)^{3/2}} - \frac{v_p^2}{(ZUP - z)^{3/2}(2\alpha v_p)^{3/2}} . \quad (5.60)$$

The presence of the second term in Eq. (5.60) insures that the integrals in Eq. (5.59) exist. As z approaches ZUP, the second term in the braces of Eq. (5.58) dominates so that in the limit as z approaches ZUP

$$[1 - (c/v_p)^2]^{1/2} \rightarrow (2\alpha/v_p)^{1/2}(ZUP - z)^{1/2}, \quad (5.60)$$

which means that ds/dv_p approaches $1/\alpha$ in accordance with Eq. (5.57). On this basis, it can be concluded that the quantity in braces in Eq. (5.58) starts out large and positive for z close to ZUP, decreases monotonically [since FTRM(Z) is always negative] and eventually goes to minus infinity as z approaches ZLOW. Therefore, there is one and only one point on the ray between the first turning point and the second turning point at which $ds/dv_p = 0$. This point is identified as a point on a caustic (i.e., where adjacent rays intercept).

At the second turning point (first lower turning point) the same sort of limiting process as described above implies that

$$ds/dv_p = [1/(dc/dz)]\Big|_{ZLOW} \tag{5.62}$$

which, as mentioned earlier, is a negative number. Between the first lower (second overall) and the second upper (third overall) turning points, it may be argued that ds/dv_p goes to zero at one and only one point. Before that point, ds/dv_p is negative, and after that point it is positive. ds/dv_p then approaches $[1/(dc/dz)]\Big|_{ZUP}$ at the next upper turning point, and so forth. As an illustration, the subprograms given in the Appendix were used to compute a plot of ds/dv_p versus range for the model atmosphere shown in Fig. 30 and for the case where ZSC and VP were set to 8 km and 0.31 km/sec, respectively. This plot is given in Fig. 34.

The number of times ds/dv_p goes to zero along a ray (i.e., the number of caustics encountered) is simply

$$\text{Number of caustics} = (\text{Number of complete ray half-cycles}) + (\text{zero or one}). \tag{5.63}$$

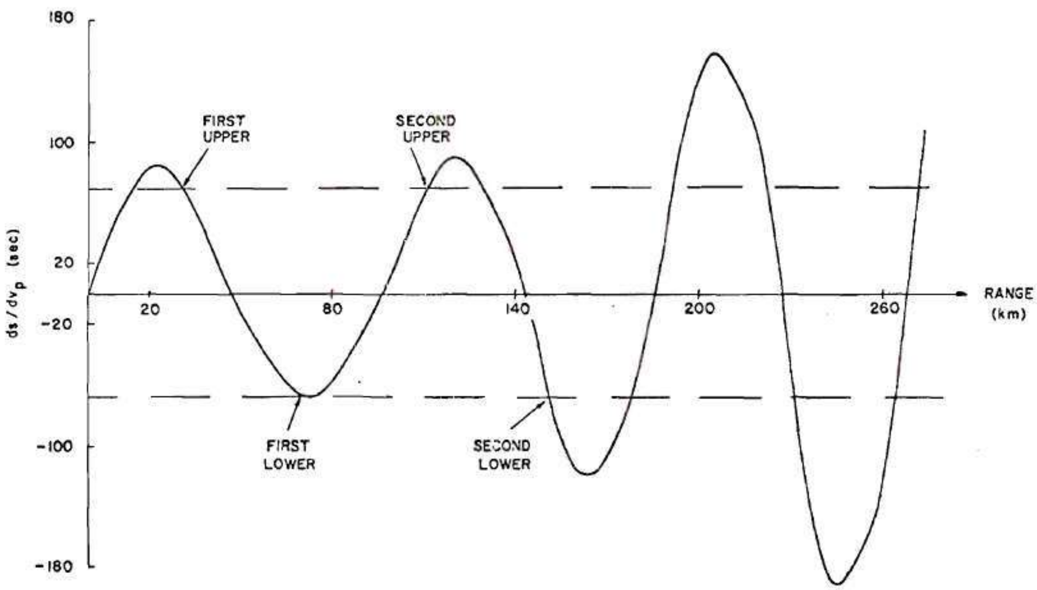

Figure 34. Values of ds/dv_p Along Two Adjacent Rays.

The second term in Eq. (5.63) is zero if JT = 1 (upgoing ray) and the current value of ds/dv_p is negative, or if JT = -1 (downgoing ray) and the current value of ds/dv_p is positive. Otherwise, it is one. The number of complete ray half-cycles is NUP + NDOWN - 1 if either NUP or NDOWN is greater than one. It is a simple matter to determine at a given point on a ray just how many caustics the ray has encountered in passing from the source to that point.

Ray Amplitudes

Given that, in the immediate vicinity of the source, the acoustic pressure $p(\vec{r},t)$ has the functional form $F(t-R/c)/R$ (R is distance from source), then the Fourier transform $\hat{p}(\omega,\vec{r})$ of the acoustic pressure can be inferred (from the geometric acoustical model)[22] to be, in the first approximation, given by a sum over rays. That is, $\hat{p}(\omega,\vec{r})$ can be expressed approximately as

$$\hat{p}(\omega,\vec{r}) = \sum_{\text{rays}} \hat{p}_{ray}, \tag{5.64}$$

where $\hat{p}(\omega,\vec{r})$ is defined so that

$$p(\vec{r},t) = \text{Re}\int_0^{\infty} \hat{p}(\omega,\vec{r})e^{-i\omega t}\, d\omega. \tag{5.65}$$

The contribution $\hat{p}_{ray}$ from any particular ray that connects source and receiver can be expressed simply as

$$\hat{p}_{ray} = \hat{f}(\omega)\rho_o^{1/2}(z_{SC})\{\text{Atmosphere factor}\}\{\text{Spreading factor}\}$$

$$\times\ \{(+i)^{N_c}\}\ e^{i\omega t_{ray}}\ , \tag{5.66}$$

where N_c is the number of times that the ray has touched a caustic, $\hat{f}(\omega)$ is the Fourier transform of the function that characterizes the time dependence of the source, $\rho_o(z_{SC})$ is the ambient density at the source height [in the model considered here, $\rho_o(z)$ is assumed constant throughout], and t_{ray} is the net travel time along a ray. The atmospheric factor is given by

$$\{\text{Atmospheric factor}\} = \{(\rho_o c)_z/(\rho_o c)_{SC}\}^{1/2} \tag{5.67}$$

while the spreading factor is the inverse square root of the ray-tube area normalized so that the factor reduces to 1/R near the source (i.e., at the beginning of the ray). In order to determine these factors, it is necessary that

$$\{|\hat{p}_{ray}|^2/\rho_o c\}\{\text{ray tube area}\} = \text{constant} \tag{5.68}$$

along the ray. It is also necessary that the acoustic pressure have the functional form in the vicinity of the source as specified above and that the net phase change in propagation from source to receiver be $-\omega t_{ray} - N_c\pi/2$.

For a cylindrically symmetric bundle of rays, it can be shown that the associated ray-tube area at the receiver location should be a constant times $|(ds/dv_p) = r_{Hor}|$, where ds/dv_p is the quantity (evaluated at the receiver location) discussed in the previous section and where r_{Hor} is the horizontal distance from source to receiver. It can also be shown that in the vicinity of the source

$$r_{Hor}|ds/dv_p| = \frac{R^2 c^2/v_p^3}{[1 - (c/v_p)^2]^{1/2}} . \tag{5.69}$$

Given Eq. (5.69) the spreading factor can be identified in general as the square root of

$$\{\text{Spreading factor}\}^2 = \frac{c^2/v_p^3}{[1 - (c/v_p)^2]^{1/2}} \frac{1}{r_{Hor}|ds/dv_p|} , \tag{5.70}$$

where c is taken here as the sound speed at the source height.

It should be noted that the spreading factor blows up whenever ds/dv_p goes to zero (i.e., at a caustic). This fact is one indication that the general formula of Eq. (5.60) may not be applicable everywhere. The modification of the method to take explicitly into account proximities to caustics is beyond the scope of the investigation presented here. More information on caustics is available in reference 6.

As an illustration of the above method, the subprograms given in the Appendix were used to compute some of the factors in Eq. (5.66) for the case of a constant-frequency (taken here as $\omega = 1$ rad/sec) source. The example chosen is appropriate to the simplified sound-speed profile of Fig. 30 and for the case where the source and receiver heights are 15 and 17 km, respectively, and the distance of propagation is 340 km. For this example, six rays were found to connect the source and receiver. Parameters and factors for these rays are given in Table 4. There, for each ray, tabulations are given of VP, IT, JT, NUP, NDOWN, N_c, t_{ray}, ds/dv_p, the spreading factor according to Eq. (5.70), and the net phase change which is $-t_{ray} - N_c\pi/2$. From the cubic-spline approximation, the sound speed at the source was found to be 0.23074 km/sec. The atmospheric factor is, of course, one. Below Table 4, a sum over rays is given of the spreading factor times $e^{i(t_{ray} + N_c\pi/2)}$. This sum is then multiplied by e^{-it}. The resulting expression provides information on the amplitude and phase of $p(\vec{r},t)$ at the receiver.

Concluding Remarks

In summary, the computational scheme described in this chapter will provide much of the information needed to describe long-range propagation for the case of a medium that contains a single sound-speed channel. Given lattice points for a sound-speed profile and source and receiver locations, this scheme will model the profile, find rays that connect source and receiver, compute distances and times of propagation, calculate a parameter that characterizes the spreading

Table 4. Ray Parameters and Computed Factors for the Example Described in the Text (ω = 1 rad/sec).

VP (km/sec)	IT	JT	NUP	NDOWN	N_c	t_{ray} (sec)	ds/dv_p (sec)	spreading factor (km^{-1})	net phase change (radians)
.33392	1	1	4	4	7	1443.6	-407.81	3.78×10^{-3}	-1455.6
.271446	1	-1	5	4	8	1478.80	146.69	1.007×10^{-2}	-1491.37
.24461	1	-1	5	4	9	1480.1	-113.6	1.95×10^{-2}	-1494.2
.33835	-1	-1	4	4	7	1431.0	439.5	3.59×10^{-3}	-1442.0
.271453	-1	1	4	5	8	1478.81	-146.76	1.006×10^{-2}	-1491.38
.24448	-1	1	4	5	9	1480.3	114.0	1.69×10^{-2}	-1494.4

$$\left[\sum_{rays} \{\text{spreading factor}\}\, e^{i(t_{ray} + N_c \pi/2)} \right] e^{-it}$$

$$= \{(1.75 \times 10^{-2})\, e^{-1.35i}\}\, e^{-it}$$

of adjacent rays, and allow for the determination of the number of caustics that any given ray has touched. Given that the receiver is not in the vicinity of a caustic, the scheme will provide the information necessary to compute the amplitude and phase of a signal as received in the far field.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Remarks Regarding Leaking Modes

It was shown in Chapter II that, for a model atmosphere in which the sound speed is constant above some arbitrarily large height, the GR_0 and GR_1 modes should have low cutoff frequencies and should be leaking below that height. Given these facts, perturbation techniques were provided for the computation of the imaginary and real parts k_I and k_R, respectively, of the horizontal wave numbers for these modes. Knowledge of the k_I then made it possible to include, in a synthesis of waveforms, contributions from the GR_0 and GR_1 modes at frequencies where these modes were leaking. It was also learned that these contributions were significant enough to warrant such an inclusion. Finally, another perturbation technique was used to explain the transition of these modes from non-leaking to leaking propagation.

In Chapter III, a description was given of the adaptation of the computer program INFRASONIC WAVEFORMS to include leaking modes. It was shown how the program could be used to compute the parameters necessary to calculate the k_I in the manner outlined in Chapter II. It was further shown that, by a judicious choice of model atmospheres, the phase velocity and the source-free amplitude functions of the GR_0 and GR_1 modes could be extended down to frequencies very close to zero. It was then shown how, given these functions and the k_I, waveforms could be synthesized with leaking modes. Numerical examples were provided which demonstrated that the contribution from leaking modes was significant and physically meaningful as far as the prediction of the early portions of infrasonic arrivals was concerned.

The question might be raised as to whether the k_I themselves are physically meaningful. Such would be the case if the earth's atmosphere were terminated by an upper half-space, and if there were no physical dissipative mechanisms present. However, neither of these conditions is fulfilled; and it must be kept in mind that the use of an approximate model atmosphere gives rise to approximate results. It must also be remembered that the actual values of the k_I depend on the choice made for the height of the bottom of the upper halfspace. To that extent, the k_I are arbitrary. Aside from this, the k_I are so small in magnitude that the associated derived waveforms are very much like those derived with the k_I nonexistent.

In light of the above comments. it is recommended that, in the synthesis of waveforms, the calculations of the k_I not be carried out for the GR_0 and GR_1 modes. Instead, the k_I should be taken either as given in the numerical example of Chapter II or set equal to 2×10^{-10} km^{-1} (i.e., for all practical purposes, zero). The k_I cannot be taken to be identically zero because INFRASONIC WAVEFORMS is designed to use the nonzeroness of the k_I as a signal that values for the source-free amplitude (AMP) are input at frequencies below cutoff. With the k_I set to zero, the program will return zero values for the AMP at these frequencies.

It is important to recognize that, while the relatively simple procedures outlined in Chapter III make the perturbation techniques presented in Chapter II computationally unnecessary, those techniques were necessary to establish a rigorous mathematical basis for the inclusion of leaking modes in the synthesis of infrasonic waveforms. In fact, the careful analysis given there made it evident that leaking modes must be included at low frequencies if accurate predictions are to be made of the early portions of arrivals. It was a contribution of this dissertation to clarify the nature and relative importance of leaking modes and to provide a procedure for the inclusion of these modes in the numerical synthesis of infrasonic waveforms. It is recommended that this procedure be made more automatic than as given here.

Remarks Regarding the High-Frequency Behavior of Guided Modes

As discussed in Chapter IV, a modified W.K.B.J. method of approximation may be used to order modes selectively and to compute useful modal parameters at relatively high frequencies. The inclusion of the method into the multi-modal scheme of the program INFRASONIC WAVEFORMS is not only feasible, but may be recommended. There is, however, some question as to whether in general a multi-modal scheme which inforporates a finite number of modes (even though they may be carefully chosen) could ever adequately synthesize the high-frequency portions of infrasonic waveforms. Indeed, this question is in itself difficult to answer because there is limited empirical data available on such portions. Aside from this fact, it is likely that a more fruitful approach to the refinement at high frequency of schemes for synthesizing waveforms lies with an appropriately designed geometric acoustics model. Nevertheless, it was a contribution of this dissertation to clarify the high-frequency behavior of guided infrasonic modes and to suggest a method of incorporating this knowledge in a numerical scheme for synthesizing infrasonic waveforms.

Remarks Regarding the Geometric Acoustical Model

The geometric acoustical computational method presented in Chapter V was designed to overcome many of the limitations customarily associated with such methods. The fact that the method produces amplitudes and phases for rays, rather than merely paths and travel times, is significant. The inclusion into the method of the possibility of having many rays that connect source and receiver coupled with the ability in the method to locate caustics precisely is important for studies of propagation over long range.

It is important to realize, however, that the method presented here is limited in scope. A comprehensive computational scheme should, of necessity, explicitly include effects that take place in the vicinity of caustics and as a result of the existence of lacunae. In addition, if a model is desired of propagation in a medium with two adjacent sound-speed channels (as is typical in the case of the atmosphere), provision would have to be made for the fact that adjacent channels can couple (i.e., some acoustic energy from one channel can penetrate into the other). Finally, and more obviously, a comprehensive computational scheme would incorporate effects due to winds, dispersion due to gravity, spreading due to the earth's curvature, sound absorption due to dissipative processes, and phase shifts as a result of ground reflections. The incorporation of these effects should not be difficult as the theory associated with them is well developed.

A comprehensive geometric acoustical model could be used as a research tool to test simpler models. For example the models developed by P. W. Smith[27,28] to describe underwater propagation, which are based on statistical notions, would lend themselves well to such testing. The intent in testing simpler models would be to refine such models to the point where they could provide precise descriptions of waveforms.

The FORTRAN subprograms provided in the Appendix were designed to be incorporated into a comprehensive computer program (as yet unwritten) for synthesizing waveforms. This program would be devised to interpret, in as straightforward a manner as possible, whatever appropriate high-frequency empirical data is available on waveforms. It was a contribution of this dissertation to outline in detail a numerical scheme for the computation of acoustic parameters required for accurate modeling of propagation over long range.

APPENDIX

DECK LISTING OF FORTRAN SUBROUTINES FOR GEOMETRIC ACOUSTICAL COMPUTATIONS IN A MEDIUM WHERE SOUND SPEED VARIES WITH HEIGHT

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                MAIN
C                                                                         MAIN
C                                                                         MAIN
C       SAMPLE MAIN PROGRAM                                               MAIN
C                                                                         MAIN
C                                                                         MAIN
C                          ----ABSTRACT----                               MAIN
C                                                                         MAIN
C          THIS SAMPLE MAIN PROGRAM IS DESIGNED TO ILLUSTRATE             MAIN
C          HOW THE USER MIGHT USE THE FOLLOWING SUBPROGRAMS TO CALCULATE  MAIN
C          A QUANTITY ASSOCIATED WITH THE DESCRIPTION OF RAY              MAIN
C          ACOUSTIC PROPAGATION.  IN THIS CASE, THE USER IS SHOWN HOW TO  MAIN
C          CALCULATE THE RAY SPREADING PARAMETER DSDVP GIVEN THE          MAIN
C          SOUND-SPEED PROFILE APPROXIMATED BY CUBIC SPLINES, THE         MAIN
C          PHASE VELOCITY ASSOCIATED WITH A GIVEN RAY, THE SOURCE         MAIN
C          AND RECEIVER HEIGHTS, AND THE RAY TYPE PARAMETERS.             MAIN
C                                                                         MAIN
C                                                                         MAIN
C     LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          MAIN
C     AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH,                MAIN
C                 JANUARY, 1976                                           MAIN
C     EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               MAIN
C                                                                         MAIN
C                                                                         MAIN
C                    ----ARGUMENT LIST----                                MAIN
C                                                                         MAIN
C     VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                         MAIN
C                                                                         MAIN
C        NCS      I         ND            I                               MAIN
C        ZI       R         100           I                               MAIN
C        CI       R         100           I                               MAIN
C        VP       R         ND            I                               MAIN
C        IT       I         ND            I                               MAIN
C        JT       I         ND            I                               MAIN
C        NUP      I         ND            I                               MAIN
C        NDOWN    I         ND            I                               MAIN
C        ZSC      R         ND            I                               MAIN
C        ZLIS     R         ND            I                               MAIN
C        ASOL     R         100           O                               MAIN
C        DSDVP    R         ND            O                               MAIN
C                                                                         MAIN
C                                                                         MAIN
C     COMMON STORAGE USED                                                 MAIN
C                                                                         MAIN
C         COMMON VP,I1,NCS,ZI(100),CI(10),ASOL(100)                       MAIN
C                                                                         MAIN
C     (I1 IS A VARIABLE USED BY SOME OF THE FUNCTION SUBPROGRAMS          MAIN
C     IN THEIR OPERATION.  SPACE IS MADE AVAILABLE FOR IT HERE.)          MAIN
C                                                                         MAIN
C                                                                         MAIN
C                    ----INPUTS----                                       MAIN
C                                                                         MAIN
C        NCS     =NUMBER OF LATTICE POINTS PROVIDED FOR THE CUBIC         MAIN
C                 SPLINE APPROXIMATION OF THE SOUND-SPEED PROFILE         MAIN
C        ZI      =HEIGHT VALUES FOR THE LATTICE POINTS IN KM              MAIN
C        CI      =SOUND SPEED VALUES FOR THE LATTICE POINTS IN KM/SEC     MAIN
C        VP      =PHASE VELOCITY IN KM/SEC                                MAIN
C        IT      =1 IF THE RAY PROPAGATES INITIALLY UPWARD                MAIN
C        IT      =-1 IF THE RAY PROPAGATES INITIALLY DOWNWARD             MAIN
C        JT      =1 IF THE RAY PROPAGATES TERMINALLY UPWARD               MAIN
C        JT      =-1 IF THE RAY PROPAGATES TERMINALLY DOWNWARD            MAIN
C        NUP     =NUMBER OF UPPER TURNING POINTS ENCOUNTERED BY RAY       MAIN
C        NDOWN   =NUMBER OF LOWER TURNING POINTS ENCOUNTERED BY RAY       MAIN
C        ZSC     =SOURCE HEIGHT IN KM                                     MAIN
```

```
C     ZLIS     =RECEIVER HEIGHT IN KM                                    MAIN
C                                                                        MAIN
C                 ----OUTPUTS----                                        MAIN
C                                                                        MAIN
C     ASOL     =COEFICIENTS CALCULATED FOR THE CUBIC SPLINES             MAIN
C     DSDVP    =RAY SPREADING PARAMETER: THE DERIVATIVE OF ADJACENT      MAIN
C               RAY SEPARATION DISTANCE WITH RESPECT TO PHASE            MAIN
C               VELOCITY                                                 MAIN
C                                                                        MAIN
C                                                                        MAIN
C                 ----EXTERNAL SUBROUTINES CALLED----                    MAIN
C                                                                        MAIN
C     DASOL,CDSDVP                                                       MAIN
C                                                                        MAIN
C                                                                        MAIN
C                 ----PROGRAM FOLLOWS BELOW----                          MAIN
C                                                                        MAIN
C                                                                        MAIN
C  THE PHASE VELOCITY, NUMBER OF LATTICE POINTS, AND LATTICE POINT       MAIN
C  VALUES ARE STORED IN COMMON IMMEDIATELY, AND THE ASOL ARE             MAIN
C  STORED WHEN DASOL RETURNS.                                            MAIN
      COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                         MAIN
C  ALL INPUT VARIABLES CAN BE READ IN UNFROMATED FORM                    MAIN
      READ(5,*)NCS,(ZI(I),I=1,NCS),(CI(I),I=1,NCS),                      MAIN
     1IT,JT,NUP,NDOWN,ZSC,ZLIS,VP                                        MAIN
C  IT IS CONSIDERED GOOD PRACTICE TO WRITE INPUT VARIABLES               MAIN
C  AFTER THEY HAVE BEEN READ.  THIS TOO IS DONE IN UNFORMATED FORM.      MAIN
      WRITE(6,*)NCS,(ZI(I),I=1,NCS),(CI(I),I=1,NCS),                     MAIN
     1IT,JT,NUP,NDOWN,ZSC,ZLIS,VP                                        MAIN
C  DASOL IS CALLED TO CALCULATE THE COEFFICIENTS FOR THE CUBIC           MAIN
C  SPLINE APROXIMATION TO THE SOUND-SPEED PROFILE.                       MAIN
      CALL DASOL                                                         MAIN
C  SUBROUTINE CDSDVP IS CALLED TO CALCULATE DSDVP, AND THE ANSWER        MAIN
C  IS PRINTED IN UNFORMATED FORM.                                        MAIN
      CALL CDSDVP(VP,ZLIS,ZSC,IT,JT,NUP,NDOWN,DSDVP)                     MAIN
      PRINT*,"DSDVP=",DSDVP                                              MAIN
      CALL EXIT                                                          MAIN
      END                                                                MAIN


      SUBROUTINE TOTRAN(VP,IT,JT,NUP,NDOWN,ZSC,ZLIS,R)                   TOTRAN
C                                                                        TOTRAN
C     TOTRAN(SUBROUTINE)                                                 TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C                       ----ABSTRACT----                                 TOTRAN
C                                                                        TOTRAN
C   TITLE - TOTRAN                                                       TOTRAN
C          THIS SUBROUTINE CALCULATES THE HORIZONTAL PROPAGATION         TOTRAN
C          DISTANCE OF AN ACOUSTIC RAY TRAPPED IN A SOUND-SPEED          TOTRAN
C          CHANNEL GIVEN THE SOURCE AND RECEIVER HEIGHTS, THE            TOTRAN
C          NUMBER OF UPPER AND LOWER TURNING POINTS, AND WHETHER         TOTRAN
C          THE RAY PROPAGATES INITIALLY FROM THE SOURCE IN AN            TOTRAN
C          UPWARD OR DOWNWARD DIRECTION, AND ALSO WHETHER IT             TOTRAN
C          PROPAGATES TERMINALLY TO THE RECEIVER IN AN UPWARD OR         TOTRAN
C          DOWNWARD DIRECTION.  SINCE A RAY IS DEFINED SPECIFICALLY      TOTRAN
C          BY A PHASE VELOCITY, TOTRAN RETURNS THE PROPAGATION           TOTRAN
C          DISTANCE OF A RAY GIVEN PHASE VELOCITY AND THE ABOVE          TOTRAN
C          INFORMATION.                                                  TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C   LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)           TOTRAN
C   AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                TOTRAN
```

```
C                 JANUARY, 1976                                          TOTRAN
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C                 ----ARGUMENT LIST----                                  TOTRAN
C                                                                        TOTRAN
C   VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                          TOTRAN
C                                                                        TOTRAN
C      VP       R         ND             I                               TOTRAN
C      IT       I         ND             I                               TOTRAN
C      JT       I         ND             I                               TOTRAN
C      NUP      I         ND             I                               TOTRAN
C      NDOWN    I         ND             I                               TOTRAN
C      ZSC      R         ND             I                               TOTRAN
C      ZLIS     R         ND             I                               TOTRAN
C      R        R         ND             I                               TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C                  ----INPUTS----                                        TOTRAN
C                                                                        TOTRAN
C      VP      =HORIZONTAL PHASE VELOCITY IN KM/SEC                      TOTRAN
C      IT      =1 IF THE RAY PROPAGATES INITIALLY UPWARD                 TOTRAN
C      IT      =-1 IF THE RAY PROPAGATES INITIALLY DOWNWARD              TOTRAN
C      JT      =1 IF THE RAY PROPAGATES TERMINALLY UPWARD                TOTRAN
C      JT      =-1 IF THE RAY PPOPAGATES TERMINALLY DOWNWARD             TOTRAN
C      NUP     =NUMBER OF UPPER TURNING POINTS                           TOTRAN
C      NDOWN   =NUMBER OF LOWER TURNING POINTS                           TOTRAN
C      ZSC     =SOURCE HEIGHT IN KM                                      TOTRAN
C      ZLIS    =LISTENER HEIGHT IN KM                                    TOTRAN
C                                                                        TOTRAN
C                  ----OUTPUT----                                        TOTRAN
C                                                                        TOTRAN
C      R       =RAY HORIZONTAL PROPAGATION DISTANCE IN KM                TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C                  ----EXTERNAL SUBROUTINES REQUIRED----                 TOTRAN
C                                                                        TOTRAN
C      TNPNT,SHIFT,RANG                                                  TOTRAN
C                                                                        TOTRAN
C                 ----FUNCTION ROUTINES REQUIRED----                     TOTRAN
C                                                                        TOTRAN
C      RAINT,RDXDZ                                                       TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
C                 ----PROGRAM FOLLOWS BELOW----                          TOTRAN
C                                                                        TOTRAN
C                                                                        TOTRAN
      EXTERNAL RDXDZ                                                     TOTRAN
C  OBTAIN THE UPPER AND LOWER TURNING POINTS FOR THE VP SPECIFIED.       TOTRAN
      CALL TNPNT(VP,ZBL,ZBU,NSCAN,NRTS,ZLOW,ZUP)                         TOTRAN
C  SHIFT THE VALUES FOR THESE SO AS TO AVOID INTEGRATION BY              TOTRAN
C  FUNCTION RAINT THROUGH SINGULARITIES.                                 TOTRAN
      CALL SHIFT(ZLOW,ZUP)                                               TOTRAN
C  NOW OBTAIN THE RAY HALF-REPETITION HORIZONTAL PROPAGATION             TOTRAN
C  DISTANCE.                                                             TOTRAN
      CALL RANG(RTIME,RLNTH,ZLOW,ZUP)                                    TOTRAN
C  IF THE RAY PROPAGATES INITIALLY DOWNWARD, GO TO 5 AND                 TOTRAN
C  FIND THE HORIZONTAL PROPAGATION DISTANCE BETWEEN THE                  TOTRAN
C  SOURCE AND THE FIRST LOWER TURNING POINT.  OTHERWISE, HAVE            TOTRAN
C  RAINT INTEGRATE RDXDZ FROM THE UPPER TURNING POINT TO THE             TOTRAN
C  SOURCE HEIGHT AND TAKE THE NEGATIVE OF THIS RESULT AS THE             TOTRAN
C  HORIZONTAL PROPAGATION DISTANCE BETWEEN THE SOURCE AND THE            TOTRAN
C  FIRST UPPER TURNING POINT.  (IT IS BEST TO INTEGRATE AWAY             TOTRAN
C  FROM SINGULARITIES (SEE FUNCTION RAINT).  THE FUNCTION                TOTRAN
```

```
C  RDXDZ(Z) AND RDTDZ(Z) ARE SINGULAR AT ZLOW AND ZUP.)                 TOTRAN
      IF (IT .LT. 0) GO TO 5                                            TOTRAN
      ANS1 = RAINT(RDXDZ,ZUP,ZSC)                                       TOTRAN
      RST = -ANS1                                                       TOTRAN
C  CALCULATE THE HORIZONTAL PROPAGATION DISTANCE BETWEEN                TOTRAN
C  THE LAST TURNING POINT AND THE RECEIVER.                             TOTRAN
      GO TO 10                                                          TOTRAN
    5 CONTINUE                                                          TOTRAN
      ANS2 = RAINT(RDXDZ,ZLOW,ZSC)                                      TOTRAN
      RST = ANS2                                                        TOTRAN
C  IF THE RAY PROPAGATES TERMINALLY DOWNWARD, GO TO 20 AND              TOTRAN
C  CALCULATE THE HORIZONTAL PROPAGATION DISTANCE BETWEEN THE            TOTRAN
C  LAST UPPER TURNING POINT AND THE RECEIVER.  OTHERWISE,               TOTRAN
C  CALCULATE THE HORIZONTAL PROPAGATION DISTANCE BETWEEN THE            TOTRAN
C  LAST LOWER TURNING POINT AND THE RECEIVER.                           TOTRAN
   10 IF (JT .LT. 0) GO TO 20                                           TOTRAN
      ANS3 = RAINT(RDXDZ,ZLOW,ZLIS)                                     TOTRAN
      REND = ANS3                                                       TOTRAN
C  GO TO 30 AND CALCULATE THE TOTAL HORIZONTAL PROPAGATION DISTANCE     TOTRAN
C  BETWEEN SOURCE AND RECEIVER.                                         TOTRAN
      GO TO 30                                                          TOTRAN
   20 CONTINUE                                                          TOTRAN
      ANS4 = RAINT(RDXDZ,ZUP,ZLIS)                                      TOTRAN
      REND = -ANS4                                                      TOTRAN
C  CALCULATE THE TOTAL NUMBER OF RAY HALF-REPETITIONS BETWEEN           TOTRAN
C  SOURCE AND RECEIVER.                                                 TOTRAN
   30 N = NUP + NDOWN - 1                                               TOTRAN
C  CALCULATE THE TOTAL HORIZONTAL PROPAGATION DISTANCE.                 TOTRAN
      R = N*RLNTH + RST + REND                                          TOTRAN
      RETURN                                                            TOTRAN
      END                                                               TOTRAN


      SUBROUTINE FNDVP(NMAX,ZSC,ZLIS,RANGE,IT,JT,NUP,NDOWN,VPHST,       FNDVP
     1VPHEND,SDELTA,NFND,VPFND)                                         FNDVP
C                                                                       FNDVP
C     FNDVP (SUBROUTINE)                                                FNDVP
C                                                                       FNDVP
C                                                                       FNDVP
C                        ----ABSTRACT----                               FNDVP
C                                                                       FNDVP
C  TITLE - FNDVP                                                        FNDVP
C          THIS SUBROUTINE IS DESIGNED TO FIND THE VALUE OF PHASE       FNDVP
C          VELOCITY ASSOCIATED WITH AN ACOUSTIC RAY TRAPPED IN A        FNDVP
C          SOUND-SPEED CHANNEL GIVEN THE SOUND-SPEED PROFILE,           FNDVP
C          SOURCE AND RECEIVER HEIGHTS, THE HORIZONRAL RANGE            FNDVP
C          BETWEEN SUCH, RAY TYPE PARAMETERS, PHASE VELOCITY            FNDVP
C          BOUNDS BETWEEN WHICH A SEARCH FOR THE DESIRED PHASE          FNDVP
C          VELOCITY IS CONDUCTED, AND THE WIDTH OF THE SUBINTERVALS     FNDVP
C          INTO WHICH THIS BOUNDED RANGE IS DIVIDED FOR A SEARCH.       FNDVP
C                                                                       FNDVP
C                                                                       FNDVP
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)           FNDVP
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                FNDVP
C              JANUARY, 1976                                            FNDVP
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                FNDVP
C                                                                       FNDVP
C                                                                       FNDVP
C                    ----ARGUMENT LIST----                              FNDVP
C                                                                       FNDVP
C  VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                          FNDVP
C                                                                       FNDVP
C    NMAX       I        ND            I                                FNDVP
```

```
C     ZSC         R             ND              I                           FNDVP
C     ZLIS        R             ND              I                           FNDVP
C     RANGE       R             ND              I                           FNDVP
C     IT          I             ND              I                           FNDVP
C     JT          I             ND              I                           FNDVP
C     NUP         I             ND              I                           FNDVP
C     NDOWN       I             ND              I                           FNDVP
C     VPHST       R             ND              I                           FNDVP
C     VPHEND      R             ND              I                           FNDVP
C     SDELTA      R             ND              I                           FNDVP
C     NFND        I             ND              O                           FNDVP
C     VPFND       R             1               O                           FNDVP
C                                                                           FNDVP
C                                                                           FNDVP
C   COMMON STORAGE USED.                                                    FNDVP
C                                                                           FNDVP
C      COMMON VPF,I1,NCS,ZI(100),CI(100),ASOL(100),                         FNDVP
C     1ZSCC,ZLISC,RANGEC,ITC,JTC,NUPC,NDOWNC                                FNDVP
C                                                                           FNDVP
C   IN COMMON STORAGE, THE FIRST SIX BLOCKS ARE OCCUPIED                    FNDVP
C   BY VARIABLES USED BY OTHER SUBROUTINES AND FUNCTION                     FNDVP
C   ROUTINES, AND WHICH ARE NOT USED EXPLICITLY BY FNDVP.                   FNDVP
C                                                                           FNDVP
C                                                                           FNDVP
C                  ----INPUTS----                                           FNDVP
C                                                                           FNDVP
C      NMAX   =NUMBER OF VALUES FOR VP TO BE FOUND (IN THIS                 FNDVP
C              VERSION, THE SUBROUTINE IS SET TO FIND ONLY ONE              FNDVP
C              VALUE FOR VP.  HOWEVER, IF THE USER KNOWS THAT MORE THAN     FNDVP
C              ONE VALUE EXISTS, THEN NMAX AND THE DIMENSION OF             FNDVP
C              VPFND COULD BE CHANGED ACCORDINGLY.  FOR EXAMPLE,            FNDVP
C              IF THE USER KNEW THAT TWO SUCH VALUES EXISTED,THEN           FNDVP
C              HE WOULD SET NMAX = 2, AND GIVE VPFND A DIMENSION OF 2.      FNDVP
C      ZSC    =SOURCE HEIGHT IN KM                                          FNDVP
C      ZLIS   =RECEIVER HEIGHT IN KM                                        FNDVP
C      RANGE  =HORIZONTAL DISTANCE BETWEEN SOURCE AND RECEIVER IN           FNDVP
C              KM                                                           FNDVP
C      IT     =1 IF THE RAY PROPAGATES INITIALLY UPWARD                     FNDVP
C      IT     =-1 IF THE RAY PROPAGATES INITIALLY DOWNWARD                  FNDVP
C      JT     =1 IF THE RAY PROPAGATES TERMINALLY UPWARD                    FNDVP
C      JT     =-1 IF THE RAY PROPAGATES TERMINALLY DOWNWARD                 FNDVP
C      NUP    =NUMBER OF UPPER TURNING POINTS THROUGH WHICH THE RAY         FNDVP
C              TRAVELS                                                      FNDVP
C      NDOWN  =NUMBER OF LOWER TURNING POINTS THROUGH WHICH THE             FNDVP
C              RAY TRAVELS                                                  FNDVP
C      VPHST  =LOWER STARTING VALUE OF PHASE VELOCITY WITH WHICH A          FNDVP
C              SEARCH IS INITIATED IN KM/SEC                                FNDVP
C      VPHEND =UPPER FINAL VALUE OF PHASE VELOCITY AT WHICH A               FNDVP
C              SEARCH IS TERMINATED IN KM/SEC                               FNDVP
C      SDELTA =WIDTH IN PHASE VELOCITY OF EACH SUBINTERVAL OF THE           FNDVP
C              RANGE OF SEARCH                                              FNDVP
C                                                                           FNDVP
C                  ----OUTPUTS----                                          FNDVP
C                                                                           FNDVP
C      NFND   =NUMBER OF PHASE VELOCITY VALUES FOUND                        FNDVP
C      VPFND  =VALUE(S) OF PHASE VELOCITY FOUND                             FNDVP
C                                                                           FNDVP
C                                                                           FNDVP
C                  ----EXTERNAL SUBROUTINES REQUIRED----                    FNDVP
C                                                                           FNDVP
C      ZREAL2                                                               FNDVP
C                                                                           FNDVP
C   ZREAL2 IS AN INTERNATIONAL MATH SCIENCE LIBRARY ROUTINE                 FNDVP
C   THAT CALCULATES N REAL ZEROES OF A REAL FUNCTION F(X) WHEN              FNDVP
```

```
C   THE INITIAL GUESSES ARE GOOD.  THIS ROUTINE USES NEWTON'S            FNDVP
C   METHOD WITH A DIVIDED DIFFERENCE APPROXIMATION FOR F'(X).            FNDVP
C   INFORMATION ON ZREAL2 IS AVAILABLE IN THE PUBLICATION               FNDVP
C   IMSL LIB3-3005 (REVISED NOVEMBER, 1975).  MORE INFORMATION          FNDVP
C   MAY BE OBTAINED BY WRITING IMSL, SIXTH FLOOR, GNB BUILDING,         FNDVP
C   7500 BELLAIRE BOULEVARD, HOUSTON, TEXAS  77036.                     FNDVP
C                                                                        FNDVP
C                 ----FUNCTION ROUTINES REQUIRED----                     FNDVP
C                                                                        FNDVP
C      RMRAYD                                                            FNDVP
C                                                                        FNDVP
C                                                                        FNDVP
C                 ----PROGRAM FOLLOWS BELOW----                          FNDVP
C                                                                        FNDVP
C                                                                        FNDVP
C   THE INPUT VARIABLES ZSC, ZLIS, RANGE, IT, JT, NUP, AND NDOWN        FNDVP
C   ARE PLACED IN THE LAST SEVEN COMMON BLOCKS.                         FNDVP
      COMMON VPF,I1,NCS,ZI(100),CI(100),ASOL(100),                      FNDVP
     1ZSCC,ZLISC,RANGEC,ITC,JTC,NUPC,NDOWNC                             FNDVP
C   IN THIS VERSION OF FNDVP, ONLY ONE VALUE OF PHASE VELOCITY          FNDVP
C   IS SOUGHT, SO THAT VPFND IS GIVEN A DIMENSION OF ONLY ONE.          FNDVP
      DIMENSION VPFND(1),X(1)                                           FNDVP
      EXTERNAL RMRAYD                                                   FNDVP
      ZSCC = ZSC                                                        FNDVP
      ZLISC = ZLIS                                                      FNDVP
      RANGEC = RANGE                                                    FNDVP
      ITC = IT                                                          FNDVP
      JTC = JT                                                          FNDVP
      NUPC = NUP                                                        FNDVP
      NDOWNC = NDOWN                                                    FNDVP
C   THE SEARCH FOR VPFND IS INITIATED AT VPHST.                         FNDVP
      NFND = 0                                                          FNDVP
      VP1 = VPHST                                                       FNDVP
C   CALCULATE THE DIFFERENCE BETWEEN THE INPUT RANGE AND THE ACTUAL     FNDVP
C   HORIZONTAL PROPAGATION DISTANCE CALCULATED FOR VP1.                 FNDVP
      F1 = RMRAYD(VP1)                                                  FNDVP
C   CALCULATE THE UPPER BOUND FOR THE FIRST SUBINTERVAL OF SEARCH.      FNDVP
    3 VP2 = VP1 + SDELTA                                                FNDVP
C   THEN CALCULATE THE RANGE DIFFERENCE FOR VP2 AS WITH VP1.            FNDVP
      F2 = RMRAYD(VP2)                                                  FNDVP
C   IF THE PRODUCT OF F1 AND F2 IS NEGATIVE, THEN WE HAVE FOUND A       FNDVP
C   SUBINTERVAL WITH THE DESIRED PHASE VELOCITY VALUE IN IT, IN         FNDVP
C   WHICH CASE WE GO TO 10.  IF IT IS ZERO OR POSITIVE, THEN WE         FNDVP
C   MUST CONTINUE THE SEARCH, AND THUS GO TO 5.  IF VP2 IS              FNDVP
C   GREATER THAN VPHEND, THEN WE HAVE REACHED THE UPPER BOUND           FNDVP
C   FOR THE SEARCH AND WE RETURN.  OTHERWISE, WE SET VP1 EQUAL TO       FNDVP
C   VP2, AND F1 EQUAL TO F2, AND INITIATE A SEARCH IN THE NEXT          FNDVP
C   SUBINTERVAL.                                                        FNDVP
      IF (F1*F2) 10,5,5                                                 FNDVP
    5 IF (VP2 .GT. VPHEND) RETURN                                       FNDVP
      VP1 = VP2                                                         FNDVP
      F1 = F2                                                           FNDVP
      GO TO 3                                                           FNDVP
C   IF WE HAVE ARRIVED HERE, WE CAN MAKE A GOOD GUESS FOR THE           FNDVP
C   VALUE OF VPFND.  FOR THIS GUESS, A SIMPLE AND APPROXIMATE           FNDVP
C   TAYLOR'S SERIES EXPANSION IS USED.                                  FNDVP
   10 GZ = VP1 - F1*SDELTA/(F2 - F1)                                    FNDVP
C   ZREAL2 IS NOW CALLED TO FIND A MORE EXACT VALUE FOR THE PHASE       FNDVP
C   VELOCITY.  TO ACCOMPLISH THIS ZREAL2 LOOKS FOR THE ROOT OF          FNDVP
C   THE FUNCTION RMRAYD(VP).  IN OTHER WORDS, ZREAL2 IS USED TO         FNDVP
C   FIND A VALUE FOR VP FOR WHICH THE CALCULATED PROPAGATION            FNDVP
C   DISTANCE EQUALS THE VALUE FOR THE INPUT VARIABLE RANGE.             FNDVP
      X(1) = GZ                                                         FNDVP
      CALL ZREAL2(RMRAYD,1.E-3,.01,SDELTA,5,1,X,10,IER)                 FNDVP
```

```
C  IF NMAX WAS CHOSEN TO BE GREATER THAN ONE, THEN THE USER COULD          FNDVP
C  CONTINUE THE SEARCH FOR MORE ROOTS OF RMRAYD UNTIL NFND                 FNDVP
C  EQUALLED NMAX.                                                          FNDVP
      NFND = NFND + 1                                                      FNDVP
      VPFND(NFND) = X(1)                                                   FNDVP
      IF (NFND .EQ. NMAX) RETURN                                           FNDVP
      GO TO 5                                                              FNDVP
      END                                                                  FNDVP


      FUNCTION RMRAYD(VPI)                                                 RMRAYD
C                                                                          RMRAYD
C     RMRAYD(FUNCTION)                                                     RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
C                             ----ABSTRACT----                             RMRAYD
C                                                                          RMRAYD
C  TITLE - RMRAYD                                                          RMRAYD
C        THIS FUNCTION ROUTINE CALCULATES THE DIFFERENCE                   RMRAYD
C        BETWEEN AN INPUT VALUE FOR HORIZONTAL PROPAGATION                 RMRAYD
C        RANGE AND THE RANGE THAT IS CALCULATED BY SUBROUTINE              RMRAYD
C        TOTRAN GIVEN PHASE VELOCITY, SOURCE AND RECEIVER                  RMRAYD
C        HEIGHTS, AND THE RAY TYPE PARAMETERS.                             RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)              RMRAYD
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                   RMRAYD
C              JANUARY, 1976                                               RMRAYD
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                   RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
C                   ----USAGE----                                          RMRAYD
C                                                                          RMRAYD
C  THE PHASE VELOCITY VPI IS THE INDEPENDENT VARIABLE INPUT.               RMRAYD
C  ALL OTHER QUANTITIES ARE PASSED THROUGH COMMON WHEN RMRAYD              RMRAYD
C  IS USED.  THE SUBROUTINE TOTRAN IS CALLED TO CALCULATE THE              RMRAYD
C  RANGE GIVEN VPI AND THE OTHER QUANTITIES.                               RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
C                   ----INPUTS----                                         RMRAYD
C                                                                          RMRAYD
C     VPI      =PHASE VELOCITY IN KM/SEC                                   RMRAYD
C     ZSCC     =SOURCE HEIGHT IN KM                                        RMRAYD
C     ZLISC    =RECEIVER HEIGHT IN KM                                      RMRAYD
C     RANGEC   =HORIZONTAL PROPAGATION DISTANCE BETWEEN SOURCE             RMRAYD
C               AND RECEIVER                                               RMRAYD
C     ITC      =1 IF THE RAY IN QUESTION PROPAGATES INITIALLY              RMRAYD
C               UPWARD                                                     RMRAYD
C     ITC      =-1 IF THE RAY PROPAGATES INITIALLY DOWNWARD                RMRAYD
C     JTC      =1 IF THE RAY PROPAGATES TERMINALLY UPWARD                  RMRAYD
C     JTC      =-1 IF THE RAY PROPAGATES TERMINALLY DOWNWARD               RMRAYD
C     NUPC     =NUMBER OF UPPER TURNING POINTS FOR THE RAY                 RMRAYD
C     NDOWNC   =NUMBER OF LOWER TURNING POINTS FOR THE RAY                 RMRAYD
C                                                                          RMRAYD
C                    ----OUTPUTS----                                       RMRAYD
C                                                                          RMRAYD
C     RMRAYD   =THE DIFFERENCE BETWEEN THE INPUT RANGE AND R, WHERE R IS   RMRAYD
C               CALCULATED BY TOTRAN                                       RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
C                   ----PROGRAM FOLLOWS BELOW----                          RMRAYD
C                                                                          RMRAYD
C                                                                          RMRAYD
```

```
      COMMON VPR,I1,NCS,ZI(100),CI(100),ASOL(100),                      RMRAYD
     1ZSCC,ZLISC,RANGEC,ITC,JTC,NUPC,NDOWNC                             RMRAYD
C   OBTAIN THE NECESSARY QUANTITIES FROM COMMON                         RMRAYD
      ZSC = ZSCC                                                        RMRAYD
      ZLIS = ZLISC                                                      RMRAYD
      RCOM = RANGEC                                                     RMRAYD
      IT = ITC                                                          RMRAYD
      JT = JTC                                                          RMRAYD
      NUP = NUPC                                                        RMRAYD
      NDOWN = NDOWNC                                                    RMRAYD
C   CALL TOTRAN TO CALCULATE THE RANGE FOR VPI AND THE ABOVE            RMRAYD
C   QUANTITIES.                                                         RMRAYD
      CALL TOTRAN(VPI,IT,JT,NUP,NDOWN,ZSC,ZLIS,R)                       RMRAYD
C   CALCULATE RMRAYD.                                                   RMRAYD
      RMRAYD = RCOM - R                                                 RMRAYD
      RETURN                                                            RMRAYD
      END                                                               RMRAYD


      SUBROUTINE SHIFT(ZLOW,ZUP)                                        SHIFT
C                                                                       SHIFT
C     SHIFT(SUBROUTINE)                                                 SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C                         ----ABSTRACT----                              SHIFT
C                                                                       SHIFT
C   TITLE - SHIFT                                                       SHIFT
C          SUBROUTINE SHIFT MOVES THE VALUES OF Z (ZLOW,ZUP)            SHIFT
C          FOUND FOR THE TURNING POINTS BY SUBROUTINE TNPNT             SHIFT
C          SO AS TO AVOID INTEGRATION THROUGH SINGULARITIES             SHIFT
C          OF Z-DEPENDENT FUNCTIONS WHERE ONE OR BOTH OF THE            SHIFT
C          INTEGRATION LIMITS IS A TURNING POINT.                       SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C   LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          SHIFT
C   AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,               SHIFT
C               JANUARY, 1976                                           SHIFT
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C                   ----USAGE----                                       SHIFT
C                                                                       SHIFT
C   THIS SUBROUTINE TAKES THE VALUES FOR ZLOW AND ZUP THAT              SHIFT
C   ARE CALCULATED BY TNPNT AND SHIFTS THEM BY UNITS OF 1.E-8           SHIFT
C   UNTIL THE FUNCTION CMVP(Z) IS LESS THAN OR EQUAL TO ZERO            SHIFT
C   FOR THESE VALUES.  IN OTHER WORDS, THE VALUES FOR C(ZLOW) AND       SHIFT
C   C(ZUP) (I.E., THE SOUND SPEED FOR THOSE HEIGHTS) ARE SHIFTED        SHIFT
C   UNTIL THEY ARE SLIGHTLY LESS THAN A GIVEN VALUE OF PHASE            SHIFT
C   VELOCITY.  THIS PHASE VELOCITY VALUE IS AVAILABLE TO CMVP           SHIFT
C   THROUGH COMMON.                                                     SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C                   ----ARGUMENT LIST----                               SHIFT
C                                                                       SHIFT
C   VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                         SHIFT
C                                                                       SHIFT
C     ZLOW      R          ND           BOTH                            SHIFT
C     ZUP       R          ND           BOTH                            SHIFT
C                                                                       SHIFT
C                   ----INPUTS----                                      SHIFT
C                                                                       SHIFT
C     ZLOW      =UNSHIFTED HEIGHT OF LOWER TURNING POINT                SHIFT
C     ZUP       =UNSHIFTED HEIGHT OF UPPER TURNING POINT                SHIFT
```

```
C                                                                       SHIFT
C                 ----OUTPUTS----                                       SHIFT
C                                                                       SHIFT
C     ZLOW      =SHIFTED HEIGHT OF LOWER TURNING POINT                  SHIFT
C     ZUP       =SHIFTED HEIGHT OF UPPER TURNING POINT                  SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C                 ----FUNCTION ROUTINES REQUIRED----                    SHIFT
C                                                                       SHIFT
C     CMVP                                                              SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
C                 ----PROGRAM FOLLOWS BELOW----                         SHIFT
C                                                                       SHIFT
C                                                                       SHIFT
      N = 0                                                             SHIFT
C  CALCULATE THE DIFFERENCE BETWEEN THE SOUND SPEED AT THE LOWER        SHIFT
C  TURNING POINT AND THE PHASE VELOCITY.                                SHIFT
    5 CHKL = CMVP(ZLOW)                                                 SHIFT
C  IF THE SOUND SPEED IS LESS THAN VP, WE'RE SAFE, AND WE GO ON TO      SHIFT
C  CHECK THE UPPER TURNING POINT.  OTHERWISE, WE ADD A TINY AMOUNT      SHIFT
C  TO ZLOW AND CONTINUE DOING SO UNTIL THE SOUND SPEED IS LESS THAN VP. SHIFT
      IF(CHKL .LE. 0.0) GO TO 10                                        SHIFT
      ZLOW = ZLOW + 1.E-8                                               SHIFT
      N = N+1                                                           SHIFT
C  IF SHIFT IS UNSUCCESSFUL IN A 1000 TRIES, WE WANT IT TO STOP.        SHIFT
      IF(N .GE. 1000) RETURN                                            SHIFT
      GO TO 5                                                           SHIFT
C  WE TRY THE SAME FOR THE UPPER TURNING POINT, AND AGAIN, AS LONG AS   SHIFT
C  THE SOUND SPEED IS LESS THAN VP, WE'RE SAFE.                         SHIFT
   10 CHKU = CMVP(ZUP)                                                  SHIFT
      IF(CHKU .LE. 0.0) RETURN                                          SHIFT
      ZUP = ZUP - 1.E-8                                                 SHIFT
      N = N+1                                                           SHIFT
      IF(N .GE. 1000) RETURN                                            SHIFT
      GO TO 10                                                          SHIFT
      END                                                               SHIFT


      FUNCTION CMVP(Z)                                                  CMVP
C                                                                       CMVP
C     CMVP (FUNCTION)                                                   CMVP
C                                                                       CMVP
C                                                                       CMVP
C                       ----ABSTRACT----                                CMVP
C                                                                       CMVP
C  TITLE - CMVP                                                         CMVP
C          THIS FUNCTION ROUTINE CALCULATES THE DIFFERENCE (AS A        CMVP
C          FUNCTION OF HEIGHT Z) BETWEEN THE PHASE VELOCITY             CMVP
C          (WHICH IS INPUT) AND THE SOUND SPEED (WHICH IS A             CMVP
C          FUNCTION OF HEIGHT Z).                                       CMVP
C                                                                       CMVP
C                                                                       CMVP
C  LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          CMVP
C  AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,               CMVP
C               JANUARY, 1976                                           CMVP
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                CMVP
C                                                                       CMVP
C                                                                       CMVP
C                 ----USAGE----                                         CMVP
C                                                                       CMVP
C  THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT, AND CMVP(Z)          CMVP
C  IS THE DEPENDENT VARIABLE OUTPUT.  THE PHASE VELOCITY VP IS          CMVP
```

```
C  AVAILABLE THROUGH COMMON AND THE SOUND SPEED IS CALCULATED          CMVP
C  BY THE FUNCTION ROUTINE CSP(Z).                                    CMVP
C                                                                     CMVP
C                                                                     CMVP
C                ----PROGRAM FOLLOWS BELOW----                        CMVP
C                                                                     CMVP
C                                                                     CMVP
      COMMON VP                                                       CMVP
      CMVP = CSP(Z) - VP                                              CMVP
      RETURN                                                          CMVP
      END                                                             CMVP


      SUBROUTINE TNPNT(VP,ZBL,ZBU,NSCAN,NRTS,ZA,ZB)                   TNPNT
C     TNPNT (SUBROUTINE)                                              TNPNT
C                                                                     TNPNT
C                                                                     TNPNT
C                         ----ABSTRACT----                            TNPNT
C                                                                     TNPNT
C  TITLE - TNPNT                                                      TNPNT
C        THIS SUBROUTINE FINDS TURNING POINTS (VALUES OF Z AT         TNPNT
C        WHICH THE DIFFERENCE BETWEEN THE SOUND SPEED CSP(Z) AND THE  TNPNT
C        THE PHASE VELOCITY VP VANISHES) GIVEN PHASE VELOCITY         TNPNT
C        AND Z.  A TURNING POINT FOR AN ACOUSTIC RAY OCCURS AT        TNPNT
C        THAT HEIGHT AT WHICH THE RAY CHANGES ITS VERTICAL DI-        TNPNT
C        RECTION OF TRAVEL, THAT IS, FROM DOWNWARD TO UPWARD OR       TNPNT
C        FROM UPWARD TO DOWNWARD.  FOR AN ACOUSTIC RAY TRAPPED        TNPNT
C        WITHIN A SOUND-SPEED CHANNEL, THERE ARE NORMALLY TWO         TNPNT
C        SUCH TURNING POINTS.                                         TNPNT
C                                                                     TNPNT
C                                                                     TNPNT
C  LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)        TNPNT
C  AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH, JANUARY,     TNPNT
C               1976                                                  TNPNT
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM              TNPNT
C                                                                     TNPNT
C                                                                     TNPNT
C               ----ARGUMENT LIST----                                 TNPNT
C                                                                     TNPNT
C  VARIABLE    TYPE   DIMENSIONS   INPUT/OUTPUT                       TNPNT
C                                                                     TNPNT
C     VP        R          ND            I                            TNPNT
C     ZBL       R          ND            I                            TNPNT
C     ZBU       R          ND            I                            TNPNT
C     NSCAN     I          ND            I                            TNPNT
C     NRTS      I          ND            O                            TNPNT
C     ZA        R          ND            O                            TNPNT
C     ZB        R          ND            O                            TNPNT
C                                                                     TNPNT
C  COMMON STORAGE USED                                                TNPNT
C     COMMON VPC,I1,NCS,ZI(100)                                       TNPNT
C                                                                     TNPNT
C     VPC       R          ND            I                            TNPNT
C     NCS       I          ND            I                            TNPNT
C     ZI        R          100           I                            TNPNT
C                                                                     TNPNT
C                                                                     TNPNT
C               ----INPUTS----                                        TNPNT
C                                                                     TNPNT
C     VP       =PHASE VELOCITY IN KM/SEC                              TNPNT
C     ZBL      =LOWER BOUND FOR HEIGHT AND ABOVE WHICH THE SEARCH     TNPNT
C               FOR TURNING POINTS IS CONDUCTED.  EXPRESSED IN KM.    TNPNT
C     ZBU      =UPPER BOUND FOR HEIGHT AND BELOW WHICH THE SEARCH     TNPNT
```

```
C             FOR TURNING POINTS IS CONDUCTED.  EXPRESSED IN KM.       TNPNT
C      NSCAN   =NUMBER OF SUBINTERVALS MINUS 1 INTO WHICH THE          TNPNT
C             INTERVAL OF SEARCH BETWEEN ZBL AND ZBU IS SUBDI-         TNPNT
C             VIDED.                                                   TNPNT
C                                                                      TNPNT
C                ----OUTPUTS----                                       TNPNT
C                                                                      TNPNT
C      NRTS    =NUMBER OF TURNING POINTS FOUND (TWO ARE EXPECTED)      TNPNT
C      ZA      =LOWER TURNING POINT (IF FOUND) EXPRESSED IN KM.        TNPNT
C      ZB      =UPPER TURNING POINT (IF FOUND) EXPRESSED IN KM.        TNPNT
C                                                                      TNPNT
C          ----EXTERNAL SUBROUTINES REQUIRED----                       TNPNT
C                                                                      TNPNT
C      ZREAL2(F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER)                       TNPNT
C                                                                      TNPNT
C                ----FUNCTION ROUTINES REQUIRED----                    TNPNT
C                                                                      TNPNT
C      CMVP,CSP                                                        TNPNT
C                                                                      TNPNT
C                                                                      TNPNT
C                ----PROGRAM FOLLOWS BELOW----                         TNPNT
C                                                                      TNPNT
C                                                                      TNPNT
      EXTERNAL CMVP                                                    TNPNT
      DIMENSION X(1)                                                   TNPNT
      COMMON VPC,I1,NCS,ZI(100)                                        TNPNT
      VPC = VP                                                         TNPNT
C  THE USER CAN SPECIFY ZBL, ZBU, AND NSCAN EXTERNALLY AS INPUTS, BUT  TNPNT
C  IN THIS VERSION OF TNPNT, IT WAS MORE CONVENIENT TO SET THEM        TNPNT
C  INTERNALLY.                                                         TNPNT
      ZBL = ZI(1)                                                      TNPNT
      ZBU = ZI(NCS)                                                    TNPNT
      NSCAN = NCS + 3                                                  TNPNT
C  CALCULATE THE WIDTH OF THE SUBINTERVALS.                            TNPNT
      DELTA = (ZBU - ZBL)/(NSCAN + 1)                                  TNPNT
C  CALCULATE CSP(ZBL) - VP.                                            TNPNT
      F1 = CMVP(ZBL)                                                   TNPNT
C  START THE SEARCH AT ZBL.                                            TNPNT
      Z1 = ZBL                                                         TNPNT
      NRTS = 0                                                         TNPNT
C  FIND THE UPPER LIMIT OF THE  SUBINTERVAL.                           TNPNT
   10 Z2 = Z1 + DELTA                                                  TNPNT
C  CALCULATE CSP(Z2) - VP                                              TNPNT
      F2 = CMVP(Z2)                                                    TNPNT
C  TAKE THE PRODUCT OF F1 AND F2, AND IF IT IS POSITIVE, WE HAVEN'T    TNPNT
C  FOUND THE SUBINTERVAL WITH A TURNING POINT IN IT YET, SO WE GO TO 15 TNPNT
C  AND START AT THE BOTTOM OF THE NEXT SUBINTERVAL.                    TNPNT
      TEST = F1*F2                                                     TNPNT
      IF(TEST .GT. 0.0) GO TO 15                                       TNPNT
C  IF F1*F2 IS NEGATIVE, WE'VE GOT A SUBINTERVAL WITH A TURNING        TNPNT
C  POINT IN IT.  AT THIS POINT, WE MAKE A GUESS FOR THE                TNPNT
C  TURNING POINT.                                                      TNPNT
      GZ = Z1 - F1*DELTA/(F2 - F1)                                     TNPNT
      X(1) = GZ                                                        TNPNT
C  ZREAL2 IS AN INTERNATIONAL MATH SCIENCE LIBRARY ROUTINE FOR         TNPNT
C  FINDING THE ZEROES OF A SPECIFIED FUNCTION (SEE SUBROUTINE FNDVP    TNPNT
C  FOR MORE INFORMATION).  ZREAL2 IS CALLED TO FIND THE TURNING        TNPNT
C  POINT IN QUESTION.                                                  TNPNT
      CALL ZREAL2(CMVP,1.E-7,0.01,DELTA,7,1,X,10,IER)                  TNPNT
      NRTS = NRTS + 1                                                  TNPNT
C  IF WE HAVE GONE THROUGH THIS LOOP SUCCESSFULLY ONCE, THEN WE HAVE   TNPNT
C  FOUND THE LOWER TURNING POINT.  IF WE HAVE GONE THROUGH TWICE, WE   TNPNT
C  HAVE FOUND BOTH TURNING POINTS, AND WE'RE DONE.                     TNPNT
      IF(NRTS .EQ. 1) ZA = X(1)                                        TNPNT
```

```
      IF(NRTS .EQ. 2) Z0 = X(1)                                          TNPNT
      IF(NRTS .EQ. 2) GO TO 20                                           TNPNT
   15 Z1 = Z2                                                            TNPNT
      F1 = F2                                                            TNPNT
C  IF WE HAVE SEARCHED ALL THE WAY TO ZBU, WE'RE DONE.  OTHERWISE, WE    TNPNT
C  GO ON TO THE NEXT SUBINTERVAL.                                        TNPNT
      IF(ZBU .GE. Z1) GO TO 10                                           TNPNT
   20 RETURN                                                             TNPNT
      END                                                                TNPNT


      SUBROUTINE RANG (RTIME,RLNTH,ZLOW,ZUP)                             RANG
C                                                                        RANG
C     RANG (SUBROUTINE)                                                  RANG
C                                                                        RANG
C                                                                        RANG
C                         ----ABSTRACT----                               RANG
C                                                                        RANG
C  TITLE - RANG                                                          RANG
C        SUBROUTINE RANG CALCULATES RAY REPETITION TIME AND LENGTH       RANG
C        BY INTEGRATION OF DT/DZ AND DX/DZ, RESPECTFULLY, BETWEEN        RANG
C        TURNING POINTS.                                                 RANG
C                                                                        RANG
C                                                                        RANG
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)            RANG
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                 RANG
C              JANUARY, 1976                                             RANG
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                 PANG
C                                                                        RANG
C                                                                        RANG
C                   ----ARGUMENT LIST----                                RANG
C                                                                        RANG
C  VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                           RANG
C                                                                        RANG
C     RTIME     R          ND            O                               RANG
C     RLNTH     R          ND            O                               RANG
C     ZLOW      R          ND            I                               RANG
C     ZUP       R          ND            I                               RANG
C                                                                        RANG
C                                                                        RANG
C                   ----INPUTS----                                       RANG
C                                                                        RANG
C     ZLOW      =LOWER TURNING POINT HEIGHT IN KM                        RANG
C     ZUP       =UPPER TURNING POINT HEIGHT IN KM                        RANG
C                                                                        RANG
C                   ----OUTPUTS----                                      RANG
C                                                                        RANG
C     RTIME     =RAY REPETITION TIME IN SEC                              RANG
C     RLNTH     =RAY REPETITION LENGTH IN KM                             RANG
C                                                                        RANG
C                                                                        RANG
C                   ----FUNCTION ROUTINES REQUIRED----                   RANG
C                                                                        PANG
C     RDXDZ,RDTDZ                                                        RANG
C                                                                        RANG
C                                                                        RANG
C                   ----PROGRAM FOLLOWS BELOW----                        RANG
C                                                                        RANG
C                                                                        RANG
      EXTERNAL RDTDZ,RDXDZ                                               RANG
      RTIME = RAINT(RDTDZ,ZLOW,ZUP)                                      RANG
      RLNTH = RAINT(RDXDZ,ZLOW,ZUP)                                      RANG
      RETURN                                                             RANG
```

```
      END                                                               RANG

      SUBROUTINE DASOL                                                  DASOL
C                                                                       DASOL
C     DASOL (SUBROUTINE)                                                DASOL
C                                                                       DASOL
C                                                                       DASOL
C                         ----ABSTRACT----                              DASOL
C                                                                       DASOL
C  TITLE - DASOL                                                        DASOL
C            THIS SUBROUTINE CALCULATES THE COEFFICIENTS OF THE         DASOL
C            CUBIC SPLINES USED TO APPROXIMATE THE SOUND-SPEED          DASOL
C            PROFILE.  THESE COEFFICIENTS ARE DEFINED BY THE            DASOL
C            RELATION                                                   DASOL
C                                                                       DASOL
C             DELZ(I)*ASOL(I-1) + 2*(DELZ(I) - DELZ(I+1))*ASOL(I) +     DASOL
C               + DELZ(I+1)*ASOL(I+1) = DELC(I+1) - DELC(I)             DASOL
C                                                                       DASOL
C           WHERE        DELZ(I) = Z(I) - Z(I-1)                        DASOL
C                        DELC(I) = (C(I) - C(I-1))/DELZ(I).             DASOL
C                                                                       DASOL
C                                                                       DASOL
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)           DASOL
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                DASOL
C              JANUARY, 1976                                            DASOL
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                DASOL
C                                                                       DASOL
C                                                                       DASOL
C  COMMON STORAGE USED                                                  DASOL
C                                                                       DASOL
C     COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                        DASOL
C                                                                       DASOL
C  VARIABLE   TYPE   DIMENSIONS   INPUT/OUTPUT                          DASOL
C                                                                       DASOL
C     NCS      I          NO             I                              DASOL
C     ZI       R          100            I                              DASOL
C     CI       R          100            I                              DASOL
C     ASOL     R          100            O                              DASOL
C                                                                       DASOL
C                ----INPUTS----                                         DASOL
C                                                                       DASOL
C     NCS      =NUMBER OF LATTICE POINTS PROVIDED FOR THE CUBIC         DASOL
C               SPLINES                                                 DASOL
C     ZI       =HEIGHT VALUES PROVIDED FOR THE LATTICE POINTS           DASOL
C     CI       =SOUND SPEED VALUES PROVIDED FOR THE LATTICE POINTS      DASOL
C                                                                       DASOL
C                ----OUTPUTS----                                        DASOL
C                                                                       DASOL
C     ASOL     =COEFFICIENTS CALCULATED FOR THE CUBIC SPLINES.  THE     DASOL
C               ASOL(I) ARE STORED IN COMMON WHEN DASOL RETURNS.        DASOL
C                                                                       DASOL
C                                                                       DASOL
C                ----PROGRAM FOLLOWS BELOW----                          DASOL
C                                                                       DASOL
C                                                                       DASOL
      COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                        DASOL
C  INITIAL VALUES ARE PROVIDED FOR THE WORKING VARIABLES.  THE          DASOL
C  BOUNDARY CONDITIONS FOR THE ASOL(I) ARE TAKEN TO BE                  DASOL
C  ASOL(1) = ASOL(NCS) = 0.0.                                           DASOL
      N = 1                                                             DASOL
      DELZ = 1.0                                                        DASOL
      DELC = 0.0                                                        DASOL
```

```
      AKM2 = 0.0                                                      DASOL
      ALM2 = 0.0                                                      DASOL
      AKM1 = 0.0                                                      DASOL
      ALM1 = 1.0                                                      DASOL
      NSTP = NCS - 1                                                  DASOL
C  CALCULATE THE DIFFERENCE IN HEIGHT VALUES AND SOUND SPEED          DASOL
C  VALUES PROVIDED FOR THE LATTICE POINTS.                            DASOL
   10 DELZP = ZI(N+1) - ZI(N)                                         DASOL
      DELCP = CI(N+1) - CI(N)                                         DASOL
C  ASOL(2) CAN BE CALCULATED GIVEN THE BOUNDARY CONDITIONS ON         DASOL
C  THE ASOL(I).                                                       DASOL
      ALPHA = DELZ                                                    DASOL
      GAMMA = DELZP                                                   DASOL
      BETA = 2.0*(ALPHA + GAMMA)                                      DASOL
      DEE = (DELCP/DELZP) - (DELC/DELZ)                               DASOL
      IF(N .EQ. 1) GO TO 30                                           DASOL
      AK = (DEE - ALPHA*AKM2 - BETA*AKM1)/GAMMA                       DASOL
      AL = ( - ALPHA*ALM2 - BETA*ALM1)/GAMMA                          DASOL
      IF(N .EQ. NSTP) GO TO 100                                       DASOL
      AKM2 = AKM1                                                     DASOL
      ALM2 = ALM1                                                     DASOL
      AKM1 = AK                                                       DASOL
      ALM1 = AL                                                       DASOL
   30 N = N + 1                                                       DASOL
      DELZ = DELZP                                                    DASOL
      DELC = DELCP                                                    DASOL
      GO TO 10                                                        DASOL
  100 ASOL(1) = 0.0                                                   DASOL
      ASOL(2) = -AK/AL                                                DASOL
      DELZ = 1.0                                                      DASOL
      DELC = 0.0                                                      DASOL
      N = 1                                                           DASOL
  110 DELZP = ZI(N+1) - ZI(N)                                         DASOL
      DELCP = CI(N+1) - CI(N)                                         DASOL
      ALPHA = DELZ                                                    DASOL
      GAMMA = DELZP                                                   DASOL
      BETA = 2.0*(ALPHA + GAMMA)                                      DASOL
      DEE = (DELCP/DELZP) - (DELC/DELZ)                               DASOL
      IF(N .EQ. 1) GO TO 130                                          DASOL
C  CALCULATE THE ASOL(M) FOR 2 < M < NCS.                             DASOL
      M = N + 1                                                       DASOL
      ASOL(M) = (DEE - ALPHA*ASOL(N-1) - BETA*ASOL(N))/GAMMA          DASOL
      IF(N .EQ. NSTP) GO TO 200                                       DASOL
  130 N = N + 1                                                       DASOL
      DELZ = DELZP                                                    DASOL
      DELC = DELCP                                                    DASOL
      GO TO 110                                                       DASOL
  200 RETURN                                                          DASOL
      END                                                             DASOL


      FUNCTION CSP(Z)                                                 CSP
C                                                                     CSP
C     CSP (FUNCTION)                                                  CSP
C                                                                     CSP
C                                                                     CSP
C                        ----ABSTRACT----                             CSP
C                                                                     CSP
C  TITLE - CSP                                                        CSP
C          THIS FUNCTION ROUTINE CALCULATES INTERMEDIATE VALUES       CSP
C          OF THE SOUND-SPEED PROFILE ACCORDING TO THE EQUATION       CSP
C                                                                     CSP
C           CSP(Z) = WBAR*C(I-1) + W*C(I) +                           CSP
```

```
C                        + ((DELZ(I)**2)*(ASOL(I-1)*(WBAR**3 - WBAR +          CSP
C                                         + ASOL(I)*(W**3 - W)).               CSP
C                                                                              CSP
C                                                                              CSP
C   LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)                CSP
C   AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                     CSP
C                JANUARY, 1976                                                 CSP
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                      CSP
C                                                                              CSP
C                                                                              CSP
C                    ----USAGE----                                             CSP
C                                                                              CSP
C   THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT, AND THE SOUND              CSP
C   SPEED CSP(Z) IS THE DEPENDENT VARIABLE OUTPUT.  OTHER REQUIRED             CSP
C   QUANTITIES ARE MADE AVAILABLE THROUGH COMMON.                              CSP
C                                                                              CSP
C                                                                              CSP
C                    ----INPUTS----                                            CSP
C                                                                              CSP
C      Z      =HEIGHT IN KM                                                    CSP
C      NCS    =NUMBER OF LATTICE POINTS IN THE SOUND-SPEED PROFILE             CSP
C      ZI     =LATICE POINT HEIGHTS IN KM                                      CSP
C      CI     =LATICE POINT SOUND-SPEED VALUES IN KM/SEC                       CSP
C      ASOL   =CUBIC SPLINE COEFFICIENTS AS CALCULATED BY                      CSP
C              SUBROUTINE DASOL                                                CSP
C                                                                              CSP
C                    ----OUTPUT----                                            CSP
C                                                                              CSP
C      CSP    =SOUND SPEED IN KM/SEC                                           CSP
C                                                                              CSP
C                                                                              CSP
C                    ----PROGRAM FOLLOWS BELOW----                             CSP
C                                                                              CSP
C                                                                              CSP
      COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                               CSP
C   DEFINE THE LOWER AND UPPER BOUNDS OF THE SOUND-SPEED PROFILE.              CSP
      ZL = ZI(1)                                                               CSP
      ZP = ZI(NCS)                                                             CSP
C   OUTSIDE OF THESE BOUNDS, LET THE SOUND SPEED BE CONSTANT AND EQUAL         CSP
C   TO THE CORRESPONDING ADJACENT VALUES.                                      CSP
      IF (Z .LT. ZL) GO TO 50                                                  CSP
      IF (Z .GT. ZP) GO TO 60                                                  CSP
      I = NCS                                                                  CSP
   10 J = I-1                                                                  CSP
C   FOR ANY VALUE Z, WE WANT I SUCH THAT Z IS BETWEEN ZI(I-1) AND ZI(I).       CSP
C   WE START WITH THE HIGHEST VALUE FOR I AND WORK DOWNWARD UNTIL WE           CSP
C   FIND THE INTERVAL THAT CONTAINS Z.                                         CSP
      ZTEST = ZI(J)                                                            CSP
C   IF Z IS BETWEEN ZI(I-1) AND ZI(I), WE GO TO 40 AND CALCULATE CSP(Z).       CSP
      IF (Z .GT. ZTEST) GO TO 40                                               CSP
C   IF Z IS NOT BETWEEN ZI(I-1) AND ZI(I), WE CHOSE THE NEXT VALUE LOWER       CSP
C   FOR I AND CONTINUE THE SEARCH.                                             CSP
      I = J                                                                    CSP
      GO TO 10                                                                 CSP
   40 CONTINUE                                                                 CSP
C   IF WE ARRIVE HERE, THEN Z IS BETWEEN ZI(I-1) AND ZI(I) SO THAT             CSP
C   WE CAN NOW CALCULATE CSP(Z).                                               CSP
      DELZ = ZI(I) - ZI(J)                                                     CSP
      W = (Z - ZI(J))/DELZ                                                     CSP
      WBAR = 1.0 - W                                                           CSP
      TERM1 = WBAR*CI(J) + W*CI(I)                                             CSP
      GUT1 = WBAR**3 - WBAR                                                    CSP
      GUT2 = W**3 - W                                                          CSP
      TERM2 = (DELZ**2)*(ASOL(J)*GUT1 + ASOL(I)*GUT2)                          CSP
```

```
      CSP = TERM1 + TERM2                                              CSP
      RETURN                                                           CSP
   50 CSP = CI(1)                                                      CSP
      RETURN                                                           CSP
   60 CSP = CI(NCS)                                                    CSP
      RETURN                                                           CSP
      END                                                              CSP


      FUNCTION DCDZ(Z)                                                 DCDZ
C                                                                      DCDZ
C     DCDZ (FUNCTION)                                                  DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
C                           ----ABSTRACT----                           DCDZ
C                                                                      DCDZ
C  TITLE - DCDZ                                                        DCDZ
C        THE FUNCTION DCDZ(Z) CALCULATES THE FIRST DERIVATIVE          DCDZ
C        OF THE SOUND SPEED WITH RESPECT TO HEIGHT Z ACCORDING         DCDZ
C        TO THE EQUATION                                               DCDZ
C                                                                      DCDZ
C        DCDZ(Z) = DELC(I) + DELZ(I)*(-ASOL(I-1)*(3*WBAR**2 - 1) +     DCDZ
C                    + ASOL(I)*(3*W**2 -1))                            DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          DCDZ
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,               DCDZ
C              JANUARY, 1976                                           DCDZ
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
C                   ----USAGE----                                      DCDZ
C                                                                      DCDZ
C  THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT, AND THE             DCDZ
C  DERIVATIVE DCDZ(Z) IS THE DEPENDENT VARIABLE OUTPUT.  OTHER         DCDZ
C  REQUIRED QUANTITIES ARE MADE AVAILABLE THROUGH COMMON.  THE         DCDZ
C  INPUT VARIABLES FOR THIS FUNCTION ARE THE SAME AS FOR FUNCTION      DCDZ
C  CSP(Z).  FOR INFORMATION ON THESE VARIABLES, THE USER IS DIRECTED   DCDZ
C  TO THAT FUNCTION.                                                   DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
C                   ----OUTPUT----                                     DCDZ
C                                                                      DCDZ
C     DCDZ   =FIRST DERIVATIVE OF SOUND SPEED WITH RESPECT TO          DCDZ
C             HEIGHT                                                   DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
C                   ----PROGRAM FOLLOWS BELOW----                      DCDZ
C                                                                      DCDZ
C                                                                      DCDZ
      COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                       DCDZ
C  DEFINE THE LOWER AND UPPER BOUNDS OF THE SOUND-SPEED PROFILE.       DCDZ
      ZL = ZI(1)                                                       DCDZ
      ZP = ZI(NCS)                                                     DCDZ
C  OUTSIDE OF THESE BOUNDS, LET DC/DZ = 0.                             DCDZ
      IF(Z .LT. ZL) GO TO 50                                           DCDZ
      IF(Z .GT. ZP) GO TO 50                                           DCDZ
      I = NCS                                                          DCDZ
   10 J = I-1                                                          DCDZ
C  FOR ANY VALUE Z, WE WANT I SUCH THAT Z IS BETWEEN ZI(I-1) AND       DCDZ
C  ZI(I).  WE START WITH THE HIGHEST VALUE FOR I AND WORK              DCDZ
C  DOWNWARD UNTIL WE FIND THE INTERVAL THAT CONTAINS Z.                DCDZ
      ZTEST = ZI(J)                                                    DCDZ
```

```
C   IF Z IS BETWEEN ZI(I-1) AND ZI(I), WE GO TO 40 AND CALCULATE        DCDZ
C   DCDZ(Z).                                                            DCDZ
      IF(Z .GT. ZTEST) GO TO 40                                         DCDZ
C   IF Z IS NOT BETWEEN ZI(I-1) AND ZI(I), WE CHOSE THE NEXT            DCDZ
C   VALUE LOWER FOR I AND CONTINUE THE SEARCH.                          DCDZ
      I = J                                                             DCDZ
      GO TO 10                                                          DCDZ
   40 CONTINUE                                                          DCDZ
C   IF WE ARRIVE HERE, THEN Z IS BETWEEN ZI(I-1) AND ZI(I) SO           DCDZ
C   THAT WE CAN NOW CALCULATE DCDZ(Z).                                  DCDZ
      DELZ = ZI(I) - ZI(J)                                              DCDZ
      DELCI = (CI(I) - CI(J))/DELZ                                      DCDZ
      W = (Z - ZI(J))/DELZ                                              DCDZ
      WBAR = 1.0 - W                                                    DCDZ
      TRM3A = ASOL(I)*((3.0*(W**2)) - 1.0)                              DCDZ
      TRM3B = ASOL(J)*((3.0*(WBAR**2)) - 1.0)                           DCDZ
      TRM3 = DELZ*(TRM3A - TRM3B)                                       DCDZ
      DCDZ = DELCI + TRM3                                               DCDZ
      RETURN                                                            DCDZ
   50 DCDZ = 0.0                                                        DCDZ
      RETURN                                                            DCDZ
      END                                                               DCDZ


      SUBROUTINE CDSDVP(VP,ZC,ZSC,IT,JT,NUP,NDOWN,DSDVP)                CDSDVP
C                                                                       CDSDVP
C     CDSDVP(SUBROUTINE)                                                CDSDVP
C                                                                       CDSDVP
C                                                                       CDSDVP
C                          ----ABSTRACT----                             CDSDVP
C                                                                       CDSDVP
C   TITLE - CDSDVP                                                      CDSDVP
C          THIS SUBROUTINE CALCULATES THE DERIVATIVE WITH RESPECT       CDSDVP
C          TO PHASE VELOCITY OF THE SEPARATION DISTANCE BETWEEN         CDSDVP
C          TWO ADJACENT ACOUSTIC RAYS.  THIS DERIVATIVE IS              CDSDVP
C          EVALUATED AT A GIVEN VALUE OF PHASE VELOCITY WHICH           CDSDVP
C          CHARACTERIZES THE TWO RAYS, AND FOR ANY POINT ON THE         CDSDVP
C          RAY THAT CORRESPONDS TO THAT PHASE VELOCITY VALUE AND        CDSDVP
C          SPECIFIED RAY TYPE PARAMETERS.                               CDSDVP
C                                                                       CDSDVP
C                                                                       CDSDVP
C   LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)         CDSDVP
C   AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,              CDSDVP
C                JANUARY, 1976                                          CDSDVP
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               CDSDVP
C                                                                       CDSDVP
C                                                                       CDSDVP
C                     ----ARGUMENT LIST----                             CDSDVP
C                                                                       CDSDVP
C   VARIABLE    TYPE   DIMENSIONS   INPUT/OUTPUT                        CDSDVP
C                                                                       CDSDVP
C      VP        R          ND            I                             CDSDVP
C      ZC        R          ND            I                             CDSDVP
C      ZSC       R          ND            I                             CDSDVP
C      IT        I          ND            I                             CDSDVP
C      JT        I          ND            I                             CDSDVP
C      NUP       I          ND            I                             CDSDVP
C      NDOWN     I          ND            I                             CDSDVP
C      CDSDVP    R          ND            O                             CDSDVP
C                                                                       CDSDVP
C   COMMON STORAGE USED                                                 CDSDVP
C      COMMON VPT                                                       CDSDVP
C                                                                       CDSDVP
```

```
C                                                                      CDSDVP
C                  ----INPUTS----                                      CDSDVP
C                                                                      CDSDVP
C     VP      =PHASE VELOCITY IN KM/SEC                                CDSDVP
C     ZC      =HEIGHT FOR WHICH THE DERIVATIVE DSDVP IS CALCUTATED     CDSDVP
C     ZSC     =SOURCE HEIGHT IN KM                                     CDSDVP
C     IT      =1 IF THE RAY IN QUESTION PROPAGATES INITIALLY           CDSDVP
C              UPWARD                                                  CDSDVP
C     IT      =-1 IF THE RAY PROPAGATES INITIALLY DOWNWARD             CDSDVP
C     JT      =1 IF THE RAY PROPAGATES TERMINALLY UPWARD               CDSDVP
C     JT      =-1 IF THE RAY PROPAGATES TERMINALLY DOWNWARD            CDSDVP
C     NUP     =NUMBER OF UPPER TURNING POINTS FOR THE RAY IN           CDSDVP
C              QUESTION                                                CDSDVP
C     NDOWN   =NUMBER OF LOWER TURNING POINTS FOR THE RAY              CDSDVP
C                                                                      CDSDVP
C                  ----OUTPUTS----                                     CDSDVP
C                                                                      CDSDVP
C     DSDVP   =DERIVATIVE OF RAY SEPARATION DISTANCE WITH RESPECT      CDSDVP
C              PHASE VELOCITY                                          CDSDVP
C                                                                      CDSDVP
C                                                                      CDSDVP
C                  ----EXTERNAL SUBROUTINES REQUIRED----               CDSDVP
C                                                                      CDSDVP
C     TNPNT,SHIFT                                                      CDSDVP
C                                                                      CDSDVP
C                  ----FUNCTION ROUTINES REQUIRED----                  CDSDVP
C                                                                      CDSDVP
C     RAINT,FTRM,FTRMUL,TRNPT,CSP                                      CDSDVP
C                                                                      CDSDVP
C                                                                      CDSDVP
C                  ----PROGRAM FOLLOWS BELOW----                       CDSDVP
C                                                                      CDSDVP
C                                                                      CDSDVP
      COMMON VPT                                                       CDSDVP
      EXTERNAL FTRMUL,FTRM                                             CDSDVP
C   PLACE THE PHASE VELOCITY INPUT IN COMMON.                          CDSDVP
      VPT = VP                                                         CDSDVP
C   CALL SUBROUTINE TNPNT TO CALCULATE THE TURNING POINTS FOR THE      CDSDVP
C   PHASE VELOCITY WHICH IS INPUT.                                     CDSDVP
      CALL TNPNT(VP,ZBL,ZBU,NSCAN,NRTS,ZLOW,ZUP)                       CDSDVP
C   CALL SUBROUTINE SHIFT TO MOVE THESE TURNING POINT SO AS TO         CDSDVP
C   AVOID SINGILARITIES UPON INTEGRATION OF THE FUNCTIONS FTRM AND     CDSDVP
C   FTRMUL.                                                            CDSDVP
      CALL SHIFT(ZLOW,ZUP)                                             CDSDVP
C   CALCULATE A HEIGHT VALUE THAT IS SLIGHTLY BELOW THE UPPER          CDSDVP
C   TURNING POINT ZUP.                                                 CDSDVP
      ZUI = ZUP - 0.01*(ZUP - ZLOW)                                    CDSDVP
C   CALCULATE A HEIGHT VALUE THAT IS SLIGHTLY ABOVE THE LOWER          CDSDVP
C   TURNING POINT ZLOW.                                                CDSDVP
      ZLI = ZLOW + 0.01*(ZUP - ZLOW)                                   CDSDVP
C   INTEGRATE THE FUNCTION FTRM BETWEEN THESE TWO VALUES.              CDSDVP
      TRMM = RAINT(FTRM,ZLI,ZUI)                                       CDSDVP
C   IF THE RAY IN QUESTION PROPAGATES INITIALLY UPWARD, INTEGRATE      CDSDVP
C   FTRM FROM ZSC TO ZUI.  OTHERWISE, GO TO 10 AND INTEGRATE FTRM      CDSDVP
C   FROM ZLI TO ZSC.                                                   CDSDVP
      IF (IT .LT. 0) GO TO 10                                          CDSDVP
      TRMI = RAINT(FTRM,ZSC,ZUI)                                       CDSDVP
      GO TO 15                                                         CDSDVP
   10 TRMI = RAINT(FTRM,ZLI,ZSC)                                       CDSDVP
C   IF THE RAY IN QUESTION PROPAGATES TERMINALLY UPWARD, INTEGRATE     CDSDVP
C   FTRM FROM ZLI TO ZC.  OTHERWISE, GO TO 20 AND INTEGRATE FTRM       CDSDVP
C   FROM ZC TO ZUI.                                                    CDSDVP
   15 IF (JT .LT. 0) GO TO 20                                          CDSDVP
      TRMF = RAINT(FTRM,ZLI,ZC)                                        CDSDVP
```

```
      GO TO 25                                                         CDSDVP
   20 TRMF = RAINT(FTRM,ZC,ZUI)                                        CDSDVP
C  CALCULATE THE REMAINING TERMS NEEDED TO CALCULATE DSDVP.            CDSDVP
   25 CONTINUE                                                         CDSDVP
      TRMU1 = TRNPT(ZUI)                                               CDSDVP
      TRML1 = TRNPT(ZLI)                                               CDSDVP
      TRMU2 = RAINT(FTRMUL,ZUP,ZUI)                                    CDSDVP
      TRML2 = RAINT(FTRMUL,ZLOW,ZLI)                                   CDSDVP
C  BEFORE COMPUTING DSDVP, WE CALCULATE AN INTERMEDIATE QUANTITY       CDSDVP
C  WHICH IS THE DERIVATIVE WITH RESPECT TO PHASE VELOCITY OF THE       CDSDVP
C  HORIZONTAL SEPARATION DISTANCE BETWEEN THE TWO ADJACENT RAYS        CDSDVP
C  THAT ARE CHARACTERIZED BY THE INPUT PHASE VELOCITY.                 CDSDVP
      CDXDVP = TRMI + NUP*(TRMU1 - TRMU2) + (NUP + NDOWN - 1)*TRMM +   CDSDVP
     1 NDOWN*(-TRML1 + TRML2) + TRMF                                   CDSDVP
C  CALCULATE THE SQUARE OF THE PHASE VELOCITY.                         CDSDVP
      VPSQ = VP**2                                                     CDSDVP
C  CALCULATE THE SOUND SPEED AT ZC.                                    CDSDVP
      CSPZC = CSP(ZC)                                                  CDSDVP
C  SQUARE THIS SOUND SPEED.                                            CDSDVP
      CSPZSQ = CSP(ZC)**2                                              CDSDVP
C  IF THE RAY IN QUESTION PROPAGATES TERMINALLY UPWARD, THEN CALCULATE CDSDVP
C  DSDVP ACCORDING TO THE FORMULA HERE.  OTHERWISE, GO TO 30 AND       CDSDVP
C  CALCULATE IT ACCORDING TO THE FORMULA GIVEN THERE.                  CDSDVP
      IF (JT .LT. 0) GO TO 30                                          CDSDVP
      DSDVP = -(CSPZC*(SQRT(VPSQ - CSPZSQ)/VPSQ))*CDXDVP               CDSDVP
      GO TO 35                                                         CDSDVP
   30 DSDVP = (CSPZC*(SQRT(VPSQ - CSPZSQ)/VPSQ))*CDXDVP                CDSDVP
   35 CONTINUE                                                         CDSDVP
      RETURN                                                           CDSDVP
      END                                                              CDSDVP


      FUNCTION FTRM(Z)                                                 FTRM
C                                                                      FTRM
C     FTRM (FUNCTION)                                                  FTRM
C                                                                      FTRM
C                                                                      FTRM
C                           ----ABSTRACT----                           FTRM
C                                                                      FTRM
C  TITLE - FTRM                                                        FTRM
C         THIS ROUTINE CALCULATES THE FUNCTION THAT IS INTEGRATED      FTRM
C         BY SUBROUTINE CDSDVP BETWEEN THE LIMITS ZSC AND ZUI OR       FTRM
C         ZLI, ZC AND ZUI OR ZLI, AND ZLI AND ZUI.  THE EXPRESSION     FTRM
C         FOR FTRM(Z) IS                                               FTRM
C                                                                      FTRM
C                                 -CSP*VP                              FTRM
C                 FTRM(Z) = ---------------------                      FTRM
C                           (VP**2 - CSP**2)**1.5                      FTRM
C                                                                      FTRM
C                                                                      FTRM
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          FTRM
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,               FTRM
C              JANUARY, 1976                                           FTRM
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               FTRM
C                                                                      FTRM
C                                                                      FTRM
C                 ----USAGE----                                        FTRM
C                                                                      FTRM
C  THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT IN KM.  THE          FTRM
C  PHASE VELOCITY VP IS MADE AVAILABLE THROUGH COMMON, AND THE         FTRM
C  SOUND SPEED IS OBTAINED FROM FUNCTION CSP(Z).                       FTRM
C                                                                      FTRM
C                                                                      FTRM
```

```
C                      ----PROGRAM FOLLOWS BELOW----                      FTRM
C                                                                         FTRM
C                                                                         FTRM
      COMMON VP,K                                                         FTRM
C   SQUARE THE PHASE VELOCITY VALUE.                                      FTRM
      VPSQ = VP**2                                                        FTRM
C   SQUARE THE SOUND SPEED.                                               FTRM
      CSPSQ = CSP(Z)**2                                                   FTRM
C   IF THE PHASE VELOCITY SQARED IS GREATER THAN OR EQUAL TO THE          FTRM
C   SOUND SPEED SQUARED, THEN WE CAN GO TO 20 AND CALCULATE               FTRM
C   THE DENOMINATOR OF FTRM(Z).  OTHERWISE, WE SET THE DENOMINATOR        FTRM
C   EQUAL TO 1.E-50, AND THEN GO TO 30 TO CALCULATE FTRM(Z).              FTRM
      IF (VPSQ .GE. CSPSQ) GO TO 20                                       FTRM
      K = 1                                                               FTRM
   10 TRM1 = 1.E-50                                                       FTRM
      GO TO 30                                                            FTRM
   20 K = 0                                                               FTRM
C   IF WE HAVE ARRIVED HERE, WE CALCULATE THE DENOMINATOR FOR             FTRM
C   FTRM(Z).  IF THE DENOMINATOR IS LESS THAN 1.E-50, THEN WE GO          FTRM
C   TO 10 AND SET IT EQUAL TO 1.E-50.                                     FTRM
      TRM1 = (SQRT(VPSQ - CSPSQ))**3                                      FTRM
      IF (TRM1 .LT. 1.E-50) GO TO 10                                      FTRM
C   CALCULATE THE NUMERATOR OF FTRM(Z).                                   FTRM
      TRM2 = CSP(Z)*VP                                                    FTRM
C   CALCULATE FTRM(Z).                                                    FTRM
   30 FTRM = -TRM2/TRM1                                                   FTRM
      RETURN                                                              FTRM
      END                                                                 FTRM


      FUNCTION DCDZS(Z)                                                   DCDZS
C                                                                         DCDZS
C     DCDZS (FUNCTION)                                                    DCDZS
C                                                                         DCDZS
C                                                                         DCDZS
C                           ----ABSTRACT----                              DCDZS
C                                                                         DCDZS
C   TITLE - DCDZS                                                         DCDZS
C          THE FUNCTION DCDZS(Z) CALCULATES THE SECOND DERIVATIVE         DCDZS
C          OF THE SOUND SPEED WITH RESPECT TO HEIGHT Z ACCORDING          DCDZS
C          TO THE EQUATION                                                DCDZS
C                                                                         DCDZS
C             DCDZS(Z) = 6*(WBAR*ASOL(I-1) + W*ASOL(I))                   DCDZS
C                                                                         DCDZS
C                                                                         DCDZS
C   LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)            DCDZS
C   AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                 DCDZS
C               JANUARY, 1976                                             DCDZS
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                 DCDZS
C                                                                         DCDZS
C                                                                         DCDZS
C                      ----USAGE----                                      DCDZS
C                                                                         DCDZS
C   THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT. AND THE               DCDZS
C   SECOND DERIVATIVE DCDZS(Z) IS THE DEPENDENT VARIABLE                  DCDZS
C   OUTPUT.  OTHER REQUIRED QUANTITIES ARE MADE AVAILABLE                 DCDZS
C   THROUGH COMMON.  THE INPUT VARIABLES FOR THIS FUNCTION                DCDZS
C   ARE THE SAME AS FOR FUNCTION CSP(Z).  FOR INFORMATION ON              DCDZS
C   THESE VARIABLES, THE USER IS DIRECTED TO THAT FUNCTION ROUTINE.       DCDZS
C                                                                         DCDZS
C                                                                         DCDZS
C                      ----OUTPUT----                                     DCDZS
C                                                                         DCDZS
```

```
C     DCDZS  =SECOND DERIVATIVE OF SOUND SPEED WITH RESPECT TO          DCDZS
C              HEIGHT Z                                                 DCDZS
C                                                                       DCDZS
C                                                                       DCDZS
C                        ----PROGRAM FOLLOWS BELOW----                  DCDZS
C                                                                       DCDZS
C                                                                       DCDZS
      COMMON VP,I1,NCS,ZI(100),CI(100),ASOL(100)                        DCDZS
C   DEFINE THE UPPER AND LOWER BOUNDS OF THE SOUND-SPEED PROFILE.       DCDZS
      ZL = ZI(1)                                                        DCDZS
      ZP = ZI(NCS)                                                      DCDZS
C   OUTSIDE OF THESE BOUNDS, LET DCDZS(Z) = 0.                          DCDZS
      IF(Z .LT. ZL) GO TO 50                                            DCDZS
      IF(Z .GT. ZP) GO TO 50                                            DCDZS
      I = NCS                                                           DCDZS
   10 J = I-1                                                           DCDZS
C   FOR ANY VALUE Z, WE WANT I SUCH THAT Z IS BETWEEN ZI(I-1)           DCDZS
C   AND ZI(I).  WE START WITH THE HIGHEST VALUE FOR I AND WORK          DCDZS
C   DOWNWARD UNTIL WE FIND THE INTERVAL THAT CONTAINS Z.                DCDZS
      ZTEST = ZI(J)                                                     DCDZS
C   IF Z IS BETWEEN ZI(I-1) AND ZI(I), WE GO TO 40 AND CALCULATE        DCDZS
C   DCDZS(Z).                                                           DCDZS
      IF(Z .GT. ZTEST) GO TO 40                                         DCDZS
C   IF Z IS NOT BETWEEN ZI(I-1) AND ZI(I), WE CHOSE THE NEXT VALUE      DCDZS
C   LOWER FOR I AND CONTINUE THE SEARCH.                                DCDZS
      I = J                                                             DCDZS
      GO TO 10                                                          DCDZS
   40 CONTINUE                                                          DCDZS
C   IF WE HAVE ARRIVED HERE, THEN Z IS BETWEEN ZI(I-1) AND ZI(I)        DCDZS
C   SO THAT WE CAN NOW CALCULATE DCDZS(Z).                              DCDZS
      DELZ = ZI(I) - ZI(J)                                              DCDZS
      W = (Z - ZI(J))/DELZ                                              DCDZS
      WBAR = 1.0 - W                                                    DCDZS
      DCDZS = 6.0*((WBAR*ASOL(J)) + (W*ASOL(I)))                        DCDZS
      RETURN                                                            DCDZS
   50 DCDZS = 0.0                                                       DCDZS
      RETURN                                                            DCDZS
      END                                                               DCDZS


      FUNCTION FTRMUL(Z)                                                FTRMUL
C                                                                       FTRMUL
C     FTRMUL (FUNCTION)                                                 FTRMUL
C                                                                       FTRMUL
C                                                                       FTRMUL
C                            ----ABSTRACT----                           FTRMUL
C                                                                       FTRMUL
C   TITLE - FTRMUL                                                      FTRMUL
C          THIS ROUTINE COMPUTES THE FUNCTION WHICH IS INTEGRATED       FTRMUL
C          BY SUBROUTINE CDSDVP BETWEEN THE LIMITS ZUP AND ZUI,         FTRMUL
C          AND ZLOW AND ZLI.  THE FORMULA FOR FTRMUL(Z) IS              FTRMUL
C                                                                       FTRMUL
C                                   -2.*VP*DCDZS                        FTRMUL
C             FTRMUL(Z) = ------------------------------                FTRMUL
C                          (DCDZ**2)*(VP**2 -CSP**2)**0.5               FTRMUL
C                                                                       FTRMUL
C                                                                       FTRMUL
C   LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)          FTRMUL
C   AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,               FTRMUL
C               JANUARY, 1976                                           FTRMUL
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM               FTRMUL
C                                                                       FTRMUL
C                                                                       FTRMUL
```

```
C                 ----USAGE----                                     FTRMUL
C                                                                   FTRMUL
C   THE HEIGHT Z IS THE INDEPENDENT VARIABLE INPUT.  THE PHASE      FTRMUL
C   VELOCITY VP IS PASSED THROUGH COMMON,  THE SOUND SPEED          FTRMUL
C   IS OBTAINED FROM FUNCTION CSP(Z), AND THE SOUND SPEED           FTRMUL
C   DERIVATIVE IS OBTAINED FROM FUNCTION DCDZ(Z).                   FTRMUL
C                                                                   FTRMUL
C                                                                   FTRMUL
C                 ----PROGRAM FOLLOWS BELOW----                     FTRMUL
C                                                                   FTRMUL
C                                                                   FTRMUL
      COMMON VP,K                                                   FTRMUL
C   SQUARE THE SOUND SPEED, THE PHASE VELOCITY, AND THE DERIVATIVE  FTRMUL
C   OF THE SOUND SPEED.                                             FTRMUL
      CSPSQ = CSP(Z)**2                                             FTRMUL
      VPSQ = VP**2                                                  FTRMUL
      DCDZSQ = DCDZ(Z)**2                                           FTRMUL
C   IF THE SQUARE OF THE PHASE VELOCITY IS GREATER THAN THE SQUARE  FTRMUL
C   OF THE SOUND SPEED, GO TO 50 AND CALCULATE THE DENOMINATOR OF   FTRMUL
C   FTRMUL(Z).  OTHERWISE, SET THE DENOMINATOR EQUAL TO 1.E-50,     FTRMUL
C   AND THEN GO TO 60 AND CALCULATE FTRMUL(Z).                      FTRMUL
      IF(VPSQ .GE. CSPSQ) GO TO 50                                  FTRMUL
      K = 1                                                         FTRMUL
   40 DN = 1.E-50                                                   FTRMUL
      GO TO 60                                                      FTRMUL
   50 K = 0                                                         FTRMUL
C   IF WE HAVE ARRIVED HERE, WE CALCULATE THE DENOMINATOR OF        FTRMUL
C   FTRMUL(Z).  IF THE DENOMINATOR IS LESS THAN 1.E-50, THEN WE GO  FTRMUL
C   TO 40 AND SET IT EQUAL TO 1.E-50.                               FTRMUL
      DN = DCDZSQ*(SQRT(VPSQ - CSPSQ))                              FTRMUL
      IF(DN .LT. 1.E-50) GO TO 40                                   FTRMUL
C   CALCULATE FTRMUL(Z).                                            FTRMUL
   60 FTRMUL = -2.*(VP*DCDZS(Z))/DN                                 FTRMUL
      RETURN                                                        FTRMUL
      END                                                           FTRMUL


      FUNCTION TRNPT(Z)                                             TRNPT
C                                                                   TRNPT
C     TRNPT (FUNCTION)                                              TRNPT
C                                                                   TRNPT
C                                                                   TRNPT
C                        ----ABSTRACT----                           TRNPT
C                                                                   TRNPT
C   TITLE - TRNPT                                                   TRNPT
C           THIS ROUTINE COMPUTES THE FUNCTION WHICH IS EVALUATED   TRNPT
C           AT TURNING POINTS BY SUBROUTINE CDSDVP.  THE FORMULA    TRNPT
C           FOR TRNPT IS                                            TRNPT
C                                                                   TRNPT
C                                 2.*VP                             TRNPT
C            TRNPT(Z) = -----------------------------               TRNPT
C                        DCDZ(Z)*((VPSQ - CSPSQ)**0.5)              TRNPT
C                                                                   TRNPT
C                                                                   TRNPT
C   LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)     TRNPT
C   AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,          TRNPT
C                JANUARY, 1976                                      TRNPT
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM           TRNPT
C                                                                   TRNPT
C                                                                   TRNPT
C                 ----USAGE----                                     TRNPT
C                                                                   TRNPT
C   THE HEIGHT Z IS THE INDEPENDENT REAL VARIABLE INPUT.   THE      TRNPT
```

```
C   PHASE VELOCITY VP IS AVAILABLE FROM COMMON, THE SOUND SPEED IS        TRNPT
C   OBTAINED FROM FUNCTION CSP(Z), AND THE DERIVATIVE OF THE SOUND        TRNPT
C   SPEED IS OBTAINED FROM FUNCTION DCDZ(Z).                              TRNPT
C                                                                         TRNPT
C                                                                         TRNPT
C                 ----PROGRAM FOLLOWS BELOW----                           TRNPT
C                                                                         TRNPT
C                                                                         TRNPT
      COMMON VP,K                                                         TRNPT
C   SQUARE THE PHASE VELOCITY AND SOUND SPEED.                            TRNPT
      CSPSQ = CSP(Z)**2                                                   TRNPT
      VPSQ = VP**2                                                        TRNPT
C   IF THE PHASE VELOCITY SQUARED IS GREATER THAN OR EQUAL TO             TRNPT
C   THE SOUND SPEED SQUARED, THEN GO TO 50 AND CALCULATE                  TRNPT
C   THE DENOMINATOR OF TRNPT(Z).   OTHERWISE, SET THE DENOMINATOR         TRNPT
C   EQUAL TO 1.E-50, AND GO TO 60 TO CALCULATE TRNPT(Z).                  TRNPT
      IF (VPSQ .GE. CSPSQ) GO TO 50                                       TRNPT
      K = 1                                                               TRNPT
   40 DN = 1.E-50                                                         TRNPT
      GO TO 60                                                            TRNPT
   50 K = 0                                                               TRNPT
C   CALCULATE THE DENOMINATOR OF TRNPT(Z).   IF ITS ABSOLUTE VALUE        TRNPT
C   IS LESS THAN 1.E-50, THEN GO TO 40 AND SET IT EQUAL TO                TRNPT
C   1.E-50.                                                               TRNPT
      DN = DCDZ(Z)*(SQRT(VPSQ - CSPSQ))                                   TRNPT
      IF (ABS(DN) .LT. 1.E-50) GO TO 40                                   TRNPT
C   CALCULATE TRNPT(Z).                                                   TRNPT
   60 TRNPT = (2.*VP)/DN                                                  TRNPT
      RETURN                                                              TRNPT
      END                                                                 TRNPT


      FUNCTION RDXDZ(Z)                                                   RDXDZ
C                                                                         RDXDZ
C     RDXDZ (FUNCTION)                                                    RDXDZ
C                                                                         RDXDZ
C                                                                         RDXDZ
C                           ----ABSTRACT----                              RDXDZ
C                                                                         RDXDZ
C   TITLE - RDXDZ                                                         RDXDZ
C         THIS FUNCTION ROUTINE CALCULATES THE DERIVATIVE OF              RDXDZ
C         X WITH RESPECT TO Z WHERE X AND Z ARE HORIZONTAL AND            RDXDZ
C         VERTICAL SPACE COORDINATES, RESPECTIVELY, WHICH                 RDXDZ
C         LOCATE POINTS ON A RAY.   A GIVEN RAY IS DEFINED BY A           RDXDZ
C         PHASE VELOCITY VP.   RDXDZ IS EXPRESSED AS                      RDXDZ
C                                                                         RDXDZ
C                                    1/VP                                 RDXDZ
C            RDXDZ(Z) = ------------------------                          RDXDZ
C                        (1/CSP**2 - 1/VP**2)**0.5                        RDXDZ
C                                                                         RDXDZ
C                                                                         RDXDZ
C   LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)            RDXDZ
C   AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                 RDXDZ
C               JANUARY, 1976                                             RDXDZ
C   EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                 RDXDZ
C                                                                         RDXDZ
C                                                                         RDXDZ
C                 ----USAGE----                                           RDXDZ
C                                                                         RDXDZ
C   THE HEIGHT Z IS THE INDEPENDENT REAL VARIABLE INPUT.   THE            RDXDZ
C   PHASE VELOCITY VP IS AVAILABLE THROUGH COMMON, AND THE                RDXDZ
C   SOUND SPEED IS OBTAINED FROM FUNCTION CSP(Z).                         RDXDZ
C                                                                         RDXDZ
```

```
C                                                              RDXDZ
C               ----PROGRAM FOLLOWS BELOW----                  RDXDZ
C                                                              RDXDZ
C                                                              RDXDZ
      COMMON VP,K                                              RDXDZ
C  SQUARE THE SOUND SPEED AND THE PHASE VELOCITY.              RDXDZ
      CSPSQ = CSP(Z)**2                                        RDXDZ
      VPSQ = VP**2                                             RDXDZ
C  IF THE SQUARE OF THE SOUND SPEED IS LESS THAN OR EQUAL TO   RDXDZ
C  THE SQUARE OF THE PHASE VELOCITY, GO TO 10 AND CALCULATE    RDXDZ
C  THE SQUARE OF THE DENOMINATOR OF RDXDZ(Z).  OTHERWISE       RDXDZ
C  SET THE SQUARE OF THE DENOMINATOR EQUAL TO 1.E-50, AND THEN RDXDZ
C  GO TO 20 AND CALCULATE RDXDZ(Z).                            RDXDZ
      IF (CSPSQ .LE. VPSQ) GO TO 10                            RDXDZ
      K = 1                                                    RDXDZ
    5 DSQ = 1.E-50                                             RDXDZ
      GO TO 20                                                 RDXDZ
   10 K = 0                                                    RDXDZ
C  IF WE HAVE ARRIVED HERE, WE CALCULATE THE SQUARE OF THE     RDXDZ
C  DENOMINATOR OF RDXDZ(Z).  IF THE SQUARE OF THE DENOMINATOR  RDXDZ
C  IS LESS THAN 1.E-50, GO TO 5 AND SET IT EQUAL TO 1.E-50.    RDXDZ
      DSQC = 1./CSPSQ                                          RDXDZ
      DSQV = 1./VPSQ                                           RDXDZ
      DSQ = DSQC - DSQV                                        RDXDZ
      IF (DSQ .LT. 1.E-50) GO TO 5                             RDXDZ
C  CALCULATE RDXDZ(Z).                                         RDXDZ
   20 RDXDZ = (1./VP)/SQRT(DSQ)                                RDXDZ
      RETURN                                                   RDXDZ
      END                                                      RDXDZ


      FUNCTION RDTDZ(Z)                                        RDTDZ
C                                                              RDTDZ
C       RDTDZ (FUNCTION)                                       RDTDZ
C                                                              RDTDZ
C                                                              RDTDZ
C                    ----ABSTRACT----                          RDTDZ
C                                                              RDTDZ
C  TITLE - RDXDZ                                               RDTDZ
C         THIS FUNCTION ROUTINE CALCULATES THE DERIVATIVE OF T RDTDZ
C         WITH RESPECT TO Z WHERE T IS TRAVEL TIME ALONG A RAY RDTDZ
C         AND WHERE Z IS THE HEIGHT OF A POINT ON A RAY.  A    RDTDZ
C         GIVEN RAY IS DEFINED BY A PHASE VELOCITY VP.  RDTDZ(Z) RDTDZ
C         IS EXPRESSED AS                                      RDTDZ
C                                                              RDTDZ
C                              1/CSP**2                        RDTDZ
C           RDTDZ(Z) = ------------------------                RDTDZ
C                       (1/CSP**2 - 1/VP**2)**0.5              RDTDZ
C                                                              RDTDZ
C                                                              RDTDZ
C  LANGUAGE   - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601) RDTDZ
C  AUTHORS    - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,      RDTDZ
C               JANUARY, 1976                                  RDTDZ
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM       RDTDZ
C                                                              RDTDZ
C                                                              RDTDZ
C               ----USAGE----                                  RDTDZ
C                                                              RDTDZ
C  THE HEIGHT Z IS THE INDEPENDENT REAL VARIABLE INPUT.  THE   RDTDZ
C  PHASE VELOCITY VP IS AVAILABLE THROUGH COMMON AND THE SOUND RDTDZ
C  SPEED IS OBTAINED FROM FUNCTION CSP(Z).                     RDTDZ
C                                                              RDTDZ
C                                                              RDTDZ
```

```
C                    ----PROGRAM FOLLOWS BELOW----                      RDTDZ
C                                                                       RDTDZ
C                                                                       RDTDZ
      COMMON VP,K                                                       RDTDZ
C  SQUARE THE SOUND SPEED AND THE PHASE VELOCITY.                       RDTDZ
      CSPSQ = CSP(Z)**2                                                 RDTDZ
      VPSQ = VP**2                                                      RDTDZ
C  IF THE SQUARE OF THE SOUND SPEED IS LESS THAN OR EQUAL TO            RDTDZ
C  THE SQUARE OF THE PHASE VELOCITY, GO TO 30 AND CALCULATE             RDTDZ
C  THE SQUARE OF THE DENOMINATOR OF RDTDZ(Z).  OTHERWISE,               RDTDZ
C  SET THE SQUARE OF THE DENOMINATOR EQUAL TO 1.E-50 AND THEN           RDTDZ
C  GO TO 40 AND CALCULATE RDTDZ(Z).                                     RDTDZ
      IF (CSPSQ .LE. VPSQ) GO TO 30                                     RDTDZ
      K = 1                                                             RDTDZ
   20 DSQ = 1.E-50                                                      RDTDZ
      GO TO 40                                                          RDTDZ
   30 K = 0                                                             RDTDZ
C  IF WE HAVE ARRIVED HERE, WE CALCULATE THE SQUARE OF THE              RDTDZ
C  DENOMINATOR OF RDTDZ(Z).  IF THE SQUARE OF THE DENOMINATOR           RDTDZ
C  IS LESS THAN 1.E-50, GO TO 20 AND SET IT EQUAL TO 1.E-50.            RDTDZ
      DSQC = 1./CSPSQ                                                   RDTDZ
      DSQV = 1./VPSQ                                                    RDTDZ
      DSQ = DSQC - DSQV                                                 RDTDZ
      IF (DSQ .LT. 1.E-50) GO TO 20                                     RDTDZ
C  CALCULATE RDTDZ(Z).                                                  RDTDZ
   40 RDTDZ = (1./CSPSQ)/SQRT(DSQ)                                      RDTDZ
      RETURN                                                            RDTDZ
      END                                                               RDTDZ


      FUNCTION RAINT(DSDZR,ZLOW,ZUP)                                    RAINT
C                                                                       RAINT
C     RAINT (FUNCTION)                                                  RAINT
C                                                                       RAINT
C                                                                       RAINT
C                          ----ABSTRACT----                             RAINT
C                                                                       RAINT
C  TITLE - RAINT                                                        RAINT
C          THIS FUNCTION ROUTINE PERFORMS THE INTEGRATION OF            RAINT
C          ANY Z-DEPENDENT FUNCTION BETWEEN LIMITS ONE OR               RAINT
C          BOTH OF WHICH MAY BE SINGULAR POINTS (E.G., TURNING          RAINT
C          POINTS).                                                     RAINT
C                                                                       RAINT
C                                                                       RAINT
C  LANGUAGE  - FORTRAN EXTENDED VERSION 4 (R.M. CDC 60305601)           RAINT
C  AUTHORS   - W.A.KINNEY AND A.D.PIERCE, GEORGIA TECH.,                RAINT
C              JANUARY, 1976                                            RAINT
C  EQUIPMENT - CDC CYBER 74, N.O.S. 1.1 OPERATING SYSTEM                RAINT
C                                                                       RAINT
C                                                                       RAINT
C                  ----USAGE----                                        RAINT
C                                                                       RAINT
C  THE FUNCTION DSDZR IS INTEGRATED BETWEEN THE LIMITS ZLOW AND         RAINT
C  ZUP (WHICH MAY OR MAY NOT BE TURNING POINTS - SEE SUBROUTINE         RAINT
C  TNPNT).                                                              RAINT
C                                                                       RAINT
C                  ----EXTERNAL SUBROUTINE REQUIRED----                 RAINT
C                                                                       RAINT
C     QUAD(A,B,D,REL,N,ANS,FANS,NERR,IMAP)                              RAINT
C                                                                       RAINT
C  QUAD IS A CDC MATH SCIENCE LIBRARY ROUTINE THAT NUMERICALLY          RAINT
C  INTEGRATES A SPECIFIED FUNCTION BETWEEN SPECIFIED LIMITS BY          RAINT
C  USE OF THE GUASS-LEGENDRE QUADRATURE TECHNIQUE.  A DESCRIPTION       RAINT
```

```
C   OF THE USAGE OF QUAD IS AVAILABLE STARTING ON PAGE 5-272 OF            RAINT
C   THE CDC REFERENCE MANUAL #60327500A.  MORE INFORMATION MAY BE          RAINT
C   OBTAINED BY WRITING CONTROL DATA CORPORATION, DOCUMENTATION            RAINT
C   DEPARTMENT, 215 MOFFETT PARK DRIVE, SUNNYVALE, CALIFORNIA              RAINT
C   94086.  THE SUBROUTINE REFERENCE NUMBER FOR QUAD IS PS-796.            RAINT
C                                                                          RAINT
C                                                                          RAINT
C                  ----PROGRAM FOLLOWS BELOW----                           RAINT
C                                                                          RAINT
C                                                                          RAINT
      EXTERNAL DSDZR                                                       RAINT
C   CALCULATE A POINT HALF WAY BETWEEN ZLOW AND ZUP.                       RAINT
      ZAVE = (ZUP + ZLOW)/2.0                                              RAINT
C   SET THE PARAMETER D FOR QUAD.                                          RAINT
      D = 1.E-6                                                            RAINT
C   INTEGRATE FROM ZLOW TO ZAVE.  SHOULD ZLOW BE A SINGUARITY,             RAINT
C   IT IS BEST TO INTEGRATE AWAY FROM IT.                                  RAINT
      CALL QUAD(ZLOW,ZAVE,D,REL,1,ANS1,DSDZR,NERR,0)                       RAINT
C   INTEGRATE FROM ZUP TO ZAVE.  SHOULD ZUP BE A SINGULARITY,              RAINT
C   IT IS BEST TO INTEGRATE AWAY FROM IT AS WELL.                          RAINT
      CALL QUAD(ZUP,ZAVE,D,REL,1,ANS2,DSDZR,NERR,0)                        RAINT
C   COMBINE THE TWO INTEGRALS.                                             RAINT
      RAINT = (ANS1 - ANS2)                                                RAINT
      RETURN                                                               RAINT
      END                                                                  RAINT
```

```
      SUBROUTINE ZREAL2 (F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER)
C
C-ZREAL2--------S--------LIBRARY 3---------------------------------------
C
C   FUNCTION            - ZREAL2 FINDS THE REAL ZEROS OF A REAL FUNCTION
C                           -- USED WHEN INITIAL GUESSES ARE GOOD
C   USAGE               - CALL ZREAL2(F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER)
C   PARAMETERS   F      - A FUNCTION F(X) SUBPROGRAM WRITTEN BY THE USER
C                EPS    - 2ND STOPPING CRITERION. A ROOT X  IS ACCEPTED
C                           IF THE ABSOLUTE VALUE OF F(X)  .LE. EPS
C                           (INPUT)
C                EPS2   - SPREAD CRITERIA FOR MULTIPLE ROOTS. IF THE
C                ETA        ITH ROOT (X(I)) HAS BEEN COMPUTED AND IT IS
C                           FOUND THAT THE ABSOLUTE VALUE OF
C                           X(I)-X(J) .LT. EPS2 WHERE X(J) IS A
C                           PREVIOUSLY COMPUTED ROOT, THEN THE
C                           COMPUTATION IS RESTARTED WITH A GUESS EQUAL
C                           TO X(I) + ETA. (INPUT)
C                NSIG   - 1ST STOPPING CRITERION. A ROOT IS ACCEPTED IF
C                           TWO SUCCESSIVE APPROXIMATIONS TO A GIVEN
C                           ROOT AGREE IN THE FIRST NSIG DIGITS. (INPUT)
C                N      - THE NUMBER OF ROOTS TO BE FOUND (INPUT)
C                X      - ON INPUT X IS AN N-VECTOR OF INITIAL GUESSES
C                           FOR N ROOTS. ON OUTPUT, X CONTAINS THE
C                           COMPUTED ROOTS.
C                ITMAX  - ON INPUT = THE MAXIMUM ALLOWABLE NUMBER OF
C                           ITERATIONS PER ROOT AND ON OUTPUT = THE
C                           NUMBER OF ITERATIONS USED ON THE LAST ROOT.
C                IER    - ERROR PARAMETER (OUTPUT)
C                         WARNING ERROR = 32 + N
C                           N = 1 INDICATES A SINGLE ROOT WAS BYPASSED
C                             BECAUSE ITMAX WAS EXCEEDED FOR THIS ROOT.
C                             X(I) FOR THIS ROOT IS SET TO 111111.
C                           N = 2 INDICATES A SINGLE ROOT WAS BYPASSED
C                             BECAUSE THE DERIVATIVE OF F FOR THIS
C                             ROOT BECOMES TOO SMALL. X(I) FOR THIS
C                             ROOT IS SET TO 222222. NOTE THAT THIS
C                             ERROR CONDITION MAY CAUSE AN OVERFLOW.
C                           N = 3 INDICATES THAT SEVERAL OF THE ABOVE
C                             ERROR CONDITIONS OCCURRED. EACH X(I) IS
C                             SET TO EITHER 111111. OR 222222. AS ABOVE
C   PRECISION           - SINGLE
C   REQD. IMSL ROUTINES - UERTST
C   LANGUAGE            - FORTRAN
C-----------------------------------------------------------------------
C   LATEST REVISION     - OCTOBER 6, 1973
C
      DIMENSION          X(1)
      DATA               P1,P001,ZERO,ONE,TEN/.1,.001,0.0,1.0,10.0/
      IER = 0
      IR=0
      CRIT1 = TEN**(-NSIG)
      DO 30 I=1,N
         IC = 1
         XI = X(I)
```

```
    5     AXI = ABS(XI)
          IF (I .EQ. 1) GO TO 15
          NM1=I-1
          DO 10 J = 1,NM1
             IF (ABS(XI - X(J)) .LT. EPS2) XI = XI + ETA
   10     CONTINUE
   15     FXI = F(XI)
          AFXI = ABS(FXI)
C                                   TEST FOR CONVERGENCE
          IF (AFXI .LE. EPS) GO TO 25
          DI = .0001
          IF (AXI .GE. P1) DI = P001*AXI
          HI = AMIN1(AFXI,DI)
          FXIPHI = F(XI + HI)
          DER = (FXIPHI - FXI)/HI
          IF (DER .EQ. ZERO) GO TO 20
          XIPI=FXI/DER
          IF (LEGVAR(XIPI) .NE. 0) GO TO 20
          XIPI=XI-XIPI
          ERR = ABS(XIPI - XI)
          XI = XIPI
C                                   TEST FOR CONVERGENCE
          IF(AXI.EQ.ZERO) AXI=ONE
          ERR1=ERR/AXI
          IF (LEGVAR(ERR1) .NE. 0) ERR1 = ERR
          IF(ERR1.LE.CRIT1) GO TO 25
          IC = IC + 1
          IF (IC .LE. ITMAX)  GO TO 5
C                                   ROOT NOT FOUND, NO CONVERGENCE
          X(I) = 111111.
          IR=IR+1
          IER=33
          GO TO 30
C                                   ROOT NOT FOUND, DERIVATIVE = 0.
   20     X(I) = 222222.
          IR=IR+1
          IER=34
          GO TO 30
   25     X(I)=XI
   30 CONTINUE
      ITMAX = IC
      IF(IER.EQ.0) GO TO 9005
      IF(IR.LE.1) GO TO 9000
      IER=35
 9000 CONTINUE
      CALL UERTST(IER,6HZREAL2)
 9005 RETURN
      END
```

```
      SUBROUTINE UERTST (IER,NAME)                                      UERTST
C                                                                       UERTST
C-UERTST----------------LIBRARY 3---------------------------------------UERTST
C                                                                       UERTST
C   FUNCTION            - ERROR MESSAGE GENERATION                      UERTST
C   USAGE               - CALL UERTST(IER,NAME)                         UERTST
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE              UERTST
C                           TYPE= 128 IMPLIES TERMINAL ERROR            UERTST
C                                  64 IMPLIES WARNING WITH FIX          UERTST
C                                  32 IMPLIES WARNING                   UERTST
C                           N   = ERROR CODE RELEVANT TO CALLING ROUTINEUERTST
C                NAME   - INPUT SCALAR CONTAINING THE NAME OF THE       UERTST
C                           CALLING ROUTINE AS A 6-CHARACTER LITERAL    UERTST
C                           STRING.                                     UERTST
C   LANGUAGE            - FORTRAN                                       UERTST
C-----------------------------------------------------------------------UERTST
C   LATEST REVISION     - AUGUST 1, 1973                                UERTST
C                                                                       UERTST
      DIMENSION          ITYP(2,4),IBIT(4)                              UERTST
      INTEGER            WARN,WARF,TERM,PRINTR                          UERTST
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)   UERTST
      DATA      ITYP      /10HWARNING   ,10H          ,                 UERTST
     *                     10HWARNING(WI,10HTH FIX)    ,                UERTST
     *                     10HTERMINAL  ,10H          ,                 UERTST
     *                     10HNON-DEFINE,10HD         /,                UERTST
     *          IBIT      / 32,64,128,0/                                UERTST
      DATA               PRINTR/6LOUTPUT/                               UERTST
      IER2=IER                                                          UERTST
      IF (IER2 .GE. WARN) GO TO 5                                       UERTST
C                                NON-DEFINED                            UERTST
      IER1=4                                                            UERTST
      GO TO 20                                                          UERTST
    5 IF (IER2 .LT. TERM) GO TO 10                                      UERTST
C                                TERMINAL                               UERTST
      IER1=3                                                            UERTST
      GO TO 20                                                          UERTST
   10 IF (IER2 .LT. WARF) GO TO 15                                      UERTST
C                                WARNING(WITH FIX)                      UERTST
      IER1=2                                                            UERTST
      GO TO 20                                                          UERTST
C                                WARNING                                UERTST
   15 IER1=1                                                            UERTST
C                                EXTRACT *N*                            UERTST
   20 IER2=IER2-IBIT(IER1)                                              UERTST
C                                PRINT ERROR MESSAGE                    UERTST
      WRITE (PRINTR,25) (ITYP(I,IER1),I=1,2),NAME,IER2,IER              UERTST
   25 FORMAT(26H *** I M S L(UERTST) ***  ,2A10,4X,A6,4X,I2,            UERTST
     1   8H (IER = ,I3,1H))                                             UERTST
      RETURN                                                            UERTST
      END                                                               UERTST
```

```
      SUBROUTINE QUAD(A,B,D,REL,N,ANS,FUN,NERR,IMAP)                      QUAD
C     A = LOWER LIMIT OF INTEGRATION  (INPUT)                             QUAD
C     B = UPPER LIMIT OF INTEGRATION (INPUT)                              QUAD
C     D = REQUIRED RELATIVE TOLERANCE (INPUT)                             QUAD
C     REL = ESTIMATE OF RESULTING RELATIVE TOLERANCE  (OUTPUT)            QUAD
C     N = SINGULARITY FLAG.  SET  N=0  WHEN NO SINGULARITY ALONG PATH.    QUAD
C             SET  N=1  WHEN ONE OR MORE SINGULARITIES LIE  ON PATH       QUAD
C     ANS = COMPUTED VALUE OF INTEGRAL  (OUTPUT)                          QUAD
C     FUN = NAME OF FUNCTION GENERATING THE INTEGRAND                     QUAD
C     NERR = ERROR FLAG  (OUTPUT)                                         QUAD
C        NERR = -1  STEP SIZE CAN NOT BE MADE SMALL ENOUGH                QUAD
C        NERR = -2  QUAD INCOMPLETE IN   LIM (200)  TRIES                 QUAD
C        NERR = -3  D HAS BEEN SET TOO SMALL                              QUAD
C        NERR .GT. 0  --SUCCESS--GIVES NUMBER OF TRIES REQUIRED           QUAD
C     IMAP = PROGRESS MAP FLAG.  SET IMAP=1 WHEN MAP IS DESIRED.          QUAD
C        SET IMAP=0 WHEN NOT DESIRED                                      QUAD
      DIMENSION W4(2),W8(4),W12(6), Z4(2),Z8(4), Z12(6)                   QUAD
      DOUBLE PRECISION YDBLE                                              QUAD
      DATA W4(1),W4(2),(W8(I),I=1,4),(W12(I),I=1,6)/.652145154862546,     QUAD
     1.347854845137454,.362683783378362,.313706645877887,.222381034453374 QUAD
     15,  .101228536290376,.249147045813403,.233492536538355,             QUAD
     1.203167426723066,.160078328543346,.106939325995318,                 QUAD
     1.047175336386512/                                                   QUAD
C     LIM CAN BE CHANGED IF EITHER MORE OR LESS TRIES ARE DESIRED         QUAD
      LIM=200                                                             QUAD
      C=D                                                                 QUAD
C     IS  C  SET TOO SMALL                                                QUAD
      IF (C.LT. 1.E-13) GO TO  290                                        QUAD
   10 IF (IMAP.EQ. 1) PRINT  1                                            QUAD
    1 FORMAT ( 2X,14HLEFT END POINT,20X,6HLENGTH,26X,12H8-PT. RESULT      QUAD
     1  11X,19HREL.ERROR IN 8-PT. ,11X,4H1000 )                           QUAD
      HOP = 0.0                                                           QUAD
      K = 0                                                               QUAD
      NONSEK = 0                                                          QUAD
      NOUT = 1                                                            QUAD
      ANS = 0.                                                            QUAD
      F2 = 0.                                                             QUAD
      NERR=0                                                              QUAD
      Y = A                                                               QUAD
      YDBLE = DBLE(Y)                                                     QUAD
      F = C/200.                                                          QUAD
      E = 0.                                                              QUAD
C******************************************************************       QUAD
C     FIRST TRY ON FULL SPAN AND ALSO LAST STEP GO THROUGH HERE           QUAD
   20 H = (B-Y)/2.                                                        QUAD
      SGN=SIGN(1.,H)                                                      QUAD
      H=ABS(H)                                                            QUAD
      LAST = 1                                                            QUAD
C     ALL INTERMEDIATE STEPS BEGIN HERE                                   QUAD
   30 X = Y + H*SGN                                                       QUAD
C     IS  H  TOO SMALL TO BE SENSED RELATIVE TO  X                        QUAD
      IF(X+.1*H.EQ.X) GO TO  270                                          QUAD
      IF(K.GT.LIM) GO TO  280                                             QUAD
C******************************************************************       QUAD
C       4 POINT ABSCISSAE                                                 QUAD
      Z4(1)=.339981043584856*H                                            QUAD
      Z4(2)=.861136311594053*H                                            QUAD
```

```
C        8 POINT ABSCISSAE                                              QUAD
      Z8(1)=.183434642495650*H                                          QUAD
      Z8(2)=.525532409916329*H                                          QUAD
      Z8(3)=.796666477413627*H                                          QUAD
      Z8(4)=.960289856497536*H                                          QUAD
C     EVALUATE FUNCTION AND PERFORM WEIGHTED SUM                        QUAD
      G4=H*(W4(1)*(FUN(X+Z4(1))+FUN(X-Z4(1)))+                          QUAD
     1W4(2)*(FUN(X+Z4(2))+FUN(X-Z4(2))))                                QUAD
      G8=0.                                                             QUAD
      DO 40 I=1,4                                                       QUAD
      Z1=FUN(X+Z8(I))                                                   QUAD
      Z2=FUN(X-Z8(I))                                                   QUAD
   40 G8=G8+W8(I)*(Z1+Z2)                                               QUAD
      G8=G8*H                                                           QUAD
C*********************************************************************** QUAD
      ABG=ABS(G8)+1.E-260                                               QUAD
      TE=ABS(G8-G4)+1.E-14*ABG                                          QUAD
C     RE IS THE RELATIVE ERROR IN THE SUBINTERVAL THE 4 PT. RESULT MAKES QUAD
C        IF THE 8 PT. RESULT IS EXACT                                   QUAD
      RE = 1.E-14 + TE/ABG                                              QUAD
      IF(K.EQ.0) P=ABG                                                  QUAD
C     P IS THE MAX ABS VALUE OF ENTIRE INTEGRAL AS WE KNOW IT UP TO HERE QUAD
C     K IS THE COUNTER OF THE NUMBER OF ATTEMPTS                        QUAD
   50 K = K + 1                                                         QUAD
      EW = F*P                                                          QUAD
      ER = TE*RE                                                        QUAD
      Q= EW/ER                                                          QUAD
      IF(IMAF.NE.1) GO TO  70                                           QUAD
   60 XLGNTH=2*H                                                        QUAD
      ERR=RE**2                                                         QUAD
      Q100=Q*100.0                                                      QUAD
      PRINT  2 ,Y,XLGNTH,G8   ,ERR,Q100                                 QUAD
    2 FORMAT (E23.15, 2E30.15, 2E22.5)                                  QUAD
   70 Q16 = Q**.0625                                                    QUAD
      D1 = H/2./RE**.125                                                QUAD
      D2 = H/D1*Q16                                                     QUAD
C     D1 IS THE ESTIMATE OF THE DISTANCE "A" TO THE SINGULARITY         QUAD
C     D2 IS AN IMPORTANCE FACTOR WHICH NORMALLY RANGES FROM ABOUT 10.   QUAD
C        TO 0.1 .  WHEN THE RESULT IS UNIMPORTANT, D2 IS LARGE.         QUAD
C                                                                       QUAD
C     THE MAGIC GO-GO OR NO-GO QUANTITY IS  100Q , FOUND AS FOLLOWS.    QUAD
C     WE REQUIRE THAT THE RELATIVE ERROR IN THE 8 PT. SUBINTERVAL       QUAD
C     VALUE  (RE**2) TIMES THE IMPORTANCE OF THE SUBINTEGRAL (ABG/P)    QUAD
C     BE LESS THAN HALF THE REQUIRED TOLERANCE  C .                     QUAD
C     ALTERNATIVELY, (C/2)*(P/ABG)/(RE**2)  MUST BE GREATER THAN 1.0    QUAD
C     THE ABOVE EXPRESSION, WHEN MULTIPLIED OUT, IS 100Q.               QUAD
      IF(Q.LE. 0.01) GO TO  120                                         QUAD
C     COMPARISON OF 4 PT. AND 8 PT. LOOKS GOOD.                         QUAD
   80 ES = 0.                                                           QUAD
      IF(N.NE.1) GO TO  200                                             QUAD
C*********************************************************************** QUAD
C     CHECK THE 12 POINT RESULT                                         QUAD
C        12 POINT ABSCISSAE                                             QUAD
      Z12(1)=.125233408511469*H                                         QUAD
      Z12(2)=.367831498998180*H                                         QUAD
      Z12(3)=.587317954286617*H                                         QUAD
      Z12(4)=.769902674194305*H                                         QUAD
```

```
      Z12(5)=.904117256370475*H                                          QUAD
      Z12(6)=.981560634246719*H                                          QUAD
C     EVALLATE FLNCTION ANC FERFORM WEIGHTED SUM                         QUAD
      G12=0                                                              QUAD
      DO 100 I=1,6                                                       QUAD
 100  G12=G12+W12(I)*(FUN(X+Z12(I))+FUN(X-Z12(I)))                       QUAD
      G12=G12*H                                                          QUAD
      ES=ABS(G12-G8)                                                     QUAD
      G8=G12                                                             QUAD
      ER=ES                                                              QUAD
      IF(ES - 100.*EW)  200,200,110                                      QUAD
C     NOT GCCD ENCUGH.  TRY AGAIN.                                       QUAD
  110 H=H/4.0                                                            QUAD
      F1 = 0.25                                                          QUAD
      GO TC  190                                                         QUAD
C*********************************************************************** QUAD
C                                                                        QUAD
C     THIS REGICN OF THE PROGRAM MCDIFIES THE STEP LENGTH WHEN           QUAD
C        SUBINTERVAL IS NOT SMALL ENOUGH                                 QUAD
  120 IF(NCLT .NE. 1) GO TO 130                                          QUAD
C     FIRST CUTBACK                                                      QUAD
      F1 = Q16                                                           QUAD
      H=AMIN1(.75*H,C1*Q16)                                              QUAD
      GO TO 190                                                          QUAD
C     SUBSEGLENT CUTBACKS IN THIS SERIES.                                QUAD
  130 F1 = F1*Q16                                                        QUAD
      H = F1*H                                                           QUAD
  190 NCNSEK = 0                                                         QUAD
      NCUT = 0                                                           QUAD
      LAST = 0                                                           QUAD
      GO TO  30                                                          QUAD
C*********************************************************************** QUAD
C                                                                        QUAD
C     SUCCESSFUL SUBINTERVAL INTEGRATION                                 QUAD
C     INCREASE STEP AS INCICATEC                                         QUAD
 200  ANS=ANS+G8                                                         QUAD
      E = E + AMAX1(ER,    ES,1.E-14*ABG)                                QUAD
      IF(LAST.EQ.1) GO TO  300                                           QUAD
C     HCP IS AN CLC SUCCESSFLL STEP                                      QUAD
  210 IF(HCP)  220,220,230                                               QUAD
  220 HCP = H                                                            QUAD
  230 F2 = 0.50*F2 + ALOG(H/HCP)                                         QUAD
      HCP = H                                                            QUAD
      YDBLE = YDBLE + DBLE(2.0*H*SGN)                                    QUAD
      Y = YDBLE                                                          QUAD
      NCNSEK = NCNSEK + 1                                                QUAD
      IF(NCNSEK  .GT. 4 ) GO TO 250                                      QUAD
      IF(F2)  240,250,250                                                QUAD
C     F2 .LT. 0. SAYS IT HAS NOT FCRGOTTEN THE PAST FAILURES YET         QUAD
  240 HC = D1*D2/(1.+2.*D2)                                              QUAD
      GO TC  260                                                         QUAD
C     F2 .GE. 0. SAYS THE HISTORY HAS BEEN SUCCESSFUL                    QUAD
  250 HC = D2*(D1+2.*H)*Q16                                              QUAD
  260 H = HC                                                             QUAD
      NCUT = 1                                                           QUAD
      P = AMAX1(F,ABG)                                                   QUAD
      IF(SGN*Y + 2.0*H - SGN*B) 30,20, 20                                QUAD
```

```
C*********************************************************************** QUAD
C                                                                        QUAD
C     ERROR EXITS                                                        QUAD
  270 NERR=-1                                                            QUAD
      WRITE(6, 3 ) H,Y                                                   QUAD
    3 FORMAT(53H QUAD FAILURE, STEP SIZE CANNOT BE MADE SMALL ENOUGH./   QUAD
     156H IF YOU WISH TO CONTINUE MOVE SINGULARITY TO THE ORIGIN./       QUAD
     211H STEP SIZE=,E24.16, 10X,15HLEFT END POINT=,E24.16)              QUAD
      GO TO  300                                                         QUAD
  280 NERR=-2                                                            QUAD
      WRITE(6, 4 ) LIM,Y,H                                               QUAD
    4 FORMAT(19H1QUAD INCOMPLETE IN  I4, 7H TRIES.,17H  LEFT END POINT=  QUAD
     1E24.16,10X,11H STEP SIZE=,E24.16)                                  QUAD
      GO TO  300                                                         QUAD
  290 NERR=-3                                                            QUAD
      PRINT  5                                                           QUAD
    5 FORMAT (68H REQUESTED TOLERANCE TOO SMALL, ROUTINE WILL PROCEED US QUAD
     1ING 10.0E-14  )                                                    QUAD
      C=10.0E-14                                                         QUAD
      GO TO  10                                                          QUAD
C                                                                        QUAD
C     HERE WE RETURN TO THE MAIN PROGRAM WITH OR WITHOUT AN ANSWER       QUAD
  300 REL= 2.*E/(ABS(ANS)+1.E-290)                                       QUAD
      IF(NERR.GE.0.) NERR=K                                              QUAD
      IF(B-A.LT.0.) ANS=-ANS                                             QUAD
      RETURN                                                             QUAD
      END                                                                QUAD
```

REFERENCES

1. D. G. Harkrider, "Theoretical and Observed Acoustic-Gravity Waves from Explosive Sources in the Atmosphere," J. Geophys. Res. 69, 5295-5321 (1964).

2. A. D. Pierce and J. W. Posey, "Theory of the Excitation and Propagation of Lamb's Atmospheric Edge Mode from Nuclear Explosions," Geophys. J. Roy. Astron. Soc. 26, 341-368 (1971).

3. J. E. Thomas, A. D. Pierce, E. A. Flinn, and L. B. Craine, "Bibliography on Infrasonic Waves," Geophys. J. R. Astr. Soc. 26, 399-426 (1971).

4. A. D. Pierce, W. A. Kinney, and C. Y. Kapper, "Atmosphere Acoustic Gravity Modes at Frequencies near and below Low Frequency Cutoff Imposed by Upper Boundary Conditions," Report No. AFCRL-TR-75-0639, Air Force Cambridge Research Laboratories, Hanscom AFB, Mass. 01731 (1 March 1976).

5. A. D. Pierce and W. A. Kinney, "Computational Techniques for the Study of Infrasound Propagation in the Atmosphere," Report No. AFGL-TR-76-0056, Air Force Geophysics Laboratory, Hanscomb AFB, Mass. 01731 (13 March 1976).

6. A. D. Pierce and W. A. Kinney, "Geometrical Acoustics Techniques in Far Field Infrasonic Waveform Syntheses," Report No. AFGL-TR-76-0055, Air Force Geophysics Laboratory, Hanscom AFB, Mass. 01731 (7 March 1976).

7. L. P. Solomon, et al., "Final Report of the Acoustic Sub-Group of the Technology Group," [Planning Systems, Inc., 7900 Westpark Drive, Suite 507, McLean, Virginia 22101], Section III, Part E.

8. C. B. Moler and L. P. Soloman, "Use of Splines and Numerical Integration in Geometric Acoustics," J. Acoust. Soc. Amer., 48, 739-744, 1970.

9. I. Tolstoy, "Phase Changes and Pulse Deformations in Acoustics," J. Acoust. Soc. Am. 44, 675-683 (1968).

10. A. D. Pierce, J. W. Posey, and E. F. Iliff, "Variation of Nuclear Explosion generated Acoustic-Gravity Waveforms with Burst Height and with Energy Yield," J. Geophys. Res. 76, 5025-5042 (1971).

11. E. T. Copson, An Introduction to the Theory of Functions of a Complex Variable (Clarendon Press, Oxford, 1935) p. 137.

12. A. D. Pierce, "The Multilayer Approximation for Infrasonic Wave Propagation in a Temperature and Wind-Stratified Atmosphere," J. Comp. Phys. 1, 343-366 (1967).

13. F. Press and D. Harkrider, "Propagation of Acoustic-Gravity Waves in the Atmosphere," J. Geophys. Res. 67, 3889-3908 (1962).

14. A. D. Pierce, "Propagation of Acoustic-Gravity Waves in a Temperature and Wind-Stratified Atmosphere," J. Acoust. Soc. Amer. 37, 218-227 (1965).

15. J. W. Posey, "Application of Lamb Edge Mode Theory in the Analysis of Explosively Generated Infrasound," Ph.D. Thesis, Dept. of Mech. Engrg., Mass. Inst. of Tech. (August, 1971).

16. N. A. Haskell, "Asymptotic Approximation for the Normal Modes in Sound Channel Wave Propagation," J. Appl. Phys. 22, 157-168 (1951).

17. F. Carlini, Ricerche sulla convergenza della serie che serva alla soluzione del problema di Keplero, Milan (1917).

18. G. Green, "On the Motion of Waves in a Variable Canal of Small Depth and Width," Trans. Camb. Phil. Soc. 6, 457-462 (1837).

19. C. Eckart, "Internal Waves in the Ocean," Phys. of Fluids 4, 791-799 (1961).

20. P. M. Morse and H. Feshbach, "Perturbation Methods for Scattering and Diffraction," Sec. 9.3 in Methods of Theoretical Physics, Vol. II (McGraw-Hill Book Co., New York, 1953) pp. 1092-1106.

21. A. D. Pierce, "Guided Infrasonic Modes in a Temperature and Wind-Stratified Atmosphere," J. Acoust. Soc. Amer. 41, 597-611 (1967).

22. A. D. Pierce, "Geometric Acoustics' Theory of Waves from a Point Source in a Temperature--and Wind--Stratified Atmosphere," Report AFCRL-66-454, Air Force Cambridge Research Laboratories, Bedford, Mass. (August, 1966).

23. D. Blokhintzer, "The Propagation of Sound in an Inhomogeneous and Moving Medium," I., J. Acoust. Soc. Amer. 18, 322-328 (1946), II., J. Acoust. Soc. Amer. 18, 329-334 (1946).

24. A. Sommerfeld and J. Runge, Ann. der. Physik 35, 277-298 (1911).

25. H. Poincare, Theorie Analytique de la Lumiere (Georges Carre, Paris, 1889).

26. I. Tolstoy, "Phase Changes and Pulse Deformation in Acoustics," J. Acoust. Soc. Amer. 44, 675-683 (1968).

27. P. W. Smith, "The Average Impulse Responses of a Shallow-Water Channel," J. Acoust. Soc. Amer., 50, 332-336, 1971.

28. P. W. Smith, "Averaged Sound Transmission in Range-Dependent Channels," J. Acoust. Soc. Amer., 55, 1197-1204, 1974.

VITA

Wayne Alan Kinney was born on July 27, 1949 in Seattle, Washington to Dr. and Mrs. Glenn C. Kinney, now of Harvard, Massachusetts.

Mr. Kinney attended the public schools in Lexington, Massachusetts and graduated from Lexington High School in June, 1967. In May, 1971, he graduated from Boston University with a double Bachelor of Arts Degree in Mathematics and Physics, and was elected to the Epsilon Chapter of Phi Beta Kappa. In February, 1973, he received a Master of Science in Mechanical Engineering from the Massachusetts Institute of Technology. In September of the same year, Mr. Kinney entered the doctoral program in the School of Mechanical Engineering at the Georgia Institute of Technology.