A new method for broadband surface acoustic wave diffraction analysis

1988

Jacqueline Hanvey Hines
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Systems and Communications Commons

STARS Citation

https://stars.library.ucf.edu/rtd/4289

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
A NEW METHOD FOR BROADBAND
SURFACE ACOUSTIC WAVE DIFFRACTION ANALYSIS

BY

JACQUELINE HANVEY HINES
B.S., Cornell University, 1984

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Graduate Studies Program of the College of Engineering University of Central Florida Orlando, Florida

Fall Term
1988
ABSTRACT

This thesis describes an innovative technique for determining the impulse response of a SAW device including the effects of diffraction on both isotropic and anisotropic substrates. The approach presented is unique in that it determines the point-to-point impulse response for the isotropic case using the two-dimensional wave equation, and utilizes a double integral reduction technique to determine the tap-to-tap response, thereby significantly reducing the complexity of the calculations involved. An extension is then made to describe the impulse response of two cascaded transducers of arbitrary geometry, along with some simplifying cases. Previous attempts to model diffraction effects generally utilized either the Fresnel Integral or the Angular Spectrum of Plane Waves approach. The Fresnel Integral technique is inherently adequate to describe diffraction effects accurately only for the narrowband case, while the Angular Spectrum of Plane Waves requires an integration over frequency if the broadband case is to be considered. The approach presented in this thesis provides an impulse response which is valid over a wide range of frequencies, which allows for an accurate description of diffraction effects for the broadband case. The isotropic impulse response is used as an approximation to the anisotropic response, taking into account the variation of surface wave speed and coupling coefficient with propagation direction.
ACKNOWLEDGEMENTS

I would like to express my deep appreciation for the encouragement and guidance provided me by my thesis advisor, Dr. Donald C. Malocha of the University of Central Florida, Orlando Fl. Without his expertise and involvement, the completion of this work would not be possible.

The author is also most grateful for the suggestions, assistance, and facilities provided her by Dr. Ron Rosenfeld, Dr. Roy Brown, and Brent Horine of SAWTEK, Inc., Apopka Fl.

I am also grateful to Ben Abbott, Melanie West, and the rest of the members of the SSDSL group at the University of Central Florida for their contributions and assistance.

Finally, I would like to thank my husband for his constant support and encouragement.
TABLE OF CONTENTS

CHAPTER

I. INTRODUCTION. ........................................... 1
   Diffraction Problem in SAW Devices. ............... 1
   Goal of Thesis. ......................................... 3

II. REVIEW OF DIFFRACTION THEORY. ....................... 4
   Angular Spectrum of Plane Waves ................. 4
   Gaussian Mode Analysis. .............................. 4
   Geometric Theory of Diffraction .................. 5
   Huygen's Principle. ..................................... 5

III. DERIVATION OF POINT SOURCE IMPULSE RESPONSE .... 7
    Two Dimensional Wave Equation in
       Cylindrical Coordinates .......................... 8
    Impulse Driver Approach .............................. 8
    Hankel Transform Solution ........................... 9
    Point Source Impulse Response .................... 9
    Useful Approximations ............................... 10

IV. TAP TO POINT RESPONSE. ............................... 13
    Huygen's Principle. .................................... 13
    Time Domain Response. ................................ 16
    Frequency Domain Response ........................... 17

V. TAP TO TAP RESPONSE. ................................ 18
    Representation of Double Integral ............... 18
    Double Integral Reduction .......................... 20
    Representation of Single Integral ................. 20

VI. TRANSDUCER TO TRANSDUCER RESPONSE. ............... 24
    Gap Analysis. .......................................... 25
    Two Apodized Transducers. ......................... 26
    Time Domain Impulse Response ...................... 26
    One Uniform, One Apodized ......................... 26

VII. EXTENSION TO THE ANISOTROPIC CASE. ............... 32
     Anisotropic Wave Equation. ......................... 32
     Simplifying Assumptions. ............................ 32
     Generation of c(\(\theta\)) ............................ 33
     Angular Dependence of Electromechanical
        Coupling Coefficient ............................. 35
     Methods for Calculating K(\(\theta\)) ................. 35

VIII. RESULTS AND DISCUSSION ............................ 37
Tap to Point Response ........................... 37
Power Flow ....................................... 37
Tap to tap Response .............................. 41
Transducer to Transducer Response .......... 46

XI. CONCLUSION ................................. 59
   Future Work ................................. 59

APPENDICES ....................................... 61

A. TAP TO POINT AND TAP TO TAP
   INTEGRATION PROGRAMS ..................... 62

B. TRANSUDER TO TRANSUDER
   INTEGRATION PROGRAMS ..................... 88

C. FOURIER TRANSFORM AND UTILITY
   PROGRAMS ..................................... 136

REFERENCES ...................................... 156
CHAPTER I
INTRODUCTION

Surface Acoustic Wave (SAW) devices utilize metallic inter-digital transducers deposited on the surface of a piezoelectric substrate to convert electrical signals into mechanical surface waves. These waves propagate along the surface of the crystal and are detected by a parallel transducer (see Figure 1). As the surface wave launched by the input transducer passes under and is detected by the receiving transducer, the bus bars of the receiving transducer automatically form a weighted sum of present and past inputs. Thus, the SAW device provides a convenient implementation for non-recursive transversal filters [1]. The distance between the transducers can be varied, providing a means to realize a device with desirable delay.

Many SAW device design techniques start with the basic assumption that the wavefront generated by an overlapping pair of electrodes is uniform in the transverse direction. However, diffraction causes the wave front to be distorted from this ideal shape, thereby introducing errors into the analysis. It is desirable to be able to compensate for diffraction effects when designing SAW devices. Several types of diffraction analyses have previously been developed for this purpose [1-8]. Most of these approaches, however, were based on traditional theories of diffraction, utilizing the Fresnel
Figure 1. Example of a Surface Acoustic Wave Device.
Integral or Angular Spectrum of Plane Waves approach. The Fresnel Integral, originally used to describe diffraction in optics [9], results in a narrowband approximation to the impulse response, while the Angular Spectrum of Plane Waves Approach requires a time consuming integration over all frequency if broadband diffraction effects are to be analyzed [3].

The goal of this research is to develop a new model for the impulse response of a SAW device by reconsidering the diffraction theory involved. An impulse driver is applied to the two dimensional wave equation and a point to point impulse response is obtained. This approach does not limit the applicable bandwidth of the response, and eliminates the need to integrate over frequency, thus reducing calculational complexity. An impulse response which is valid over a wide range of frequencies is thereby obtained, allowing for an accurate description of diffraction effects for the broadband case. This analysis is applied first to isotropic substrates, then extended to the anisotropic case.
CHAPTER II
REVIEW OF DIFFRACTION THEORY

The theory used to analyze diffraction of surface acoustic waves is essentially an extension of the principles of optical diffraction to anisotropic media and two dimensional wave propagation [10,11]. Four basic theoretical approaches and several variations have been described in the literature [1-8,10-16].

Angular Spectrum of Plane Waves

This technique is a type of Fourier analysis, in which a known source distribution is broken into its component plane waves, each with a different wave vector \( \vec{k} \). These plane waves are then propagated and recombined with appropriate phase shifts taken into consideration to yield the amplitude of the wave at any field point [4]. This involves an integration over \( k \)-space, which can be time consuming. One approach to reducing the complexity of these calculations is an asymptotic expansion which has been previously presented by Tan and Flory [8].

Gaussian Mode Analysis

This technique again involves resolution of the wave into components, however here the basic components are considered to be Gaussian waveforms rather than plane waves. The reason for utilizing this basis set for decomposing the wave is that in isotropic and
parabolically anisotropic space, Gaussian beams retain their shape during propagation. However, this property does not hold for the case of general anisotropy, and thus this technique has limited usefulness [1].

**Geometric Theory of Diffraction**

In the literature, the exact definition of what the Geometric Theory consists of is somewhat vague. The term has been applied to analyses more closely linked to Huygens Principle (or the Green's Function approach, discussed below). One common description of the Geometric Theory, however, refers to a technique where the wave launched from a transmitting tap is viewed as a planar wave of finite extent combined with two circular waves radiating from point sources located at the edges of the tap [4]. These circular waves can be weighted by amplitude coefficients which are functions of propagation direction [4]. Although this theory is simpler than Huygen's principle, it involves certain assumptions not required by other approaches.

**Huygen's Principle**

The analysis technique commonly referred to as Huygen's Principle is in fact an extension of the Huygens-Fresnel Principle to the case of radiation from a source [9]. According to Huygen's construction, every point on a wavefront can be considered as a point source which gives rise to secondary disturbances, and the resultant wavefront at any given instant can be viewed as the envelope of these secondary wavelets. Fresnel postulated that the secondary
wavelets interfere, thereby accounting for diffraction effects observed in optics [9,17,18]. When the Huygens-Fresnel Principle is extended to describe radiation from a source, the source is viewed as a collection of point sources, and the overall response at any position can be found through superposition of the responses of the individual point sources. When the impulse response of the points is considered, as is generally the case, the Huygen's-Fresnel Principle becomes essentially a Green's function approach [9], where the impulse response of an individual point source (the Green's function) is multiplied by an appropriate weighting factor and integrated over the source distribution to obtain the overall impulse response of the source [18]. Although Huygen's Principle requires integration over the source distribution, it is widely used because of the ease of performing this integration numerically, as well as the simplicity and elegance of the theory.
CHAPTER III
DERIVATION OF POINT SOURCE IMPULSE RESPONSE

In developing and solving the two dimensional wave equation, the surface wave amplitude at any position \( \mathbf{r} \) and time \( t \) can be represented by a scalar function \( \psi(\mathbf{r}, t) \), which can be interpreted as either the electric surface potential \( \Phi(\mathbf{r}, t) \) for a piezoelectric substrate, or a component of the displacement at the surface \( \mathbf{u}(\mathbf{r}, t) \) [3]. The longitudinal component of displacement will be used in this development.

The equation of motion for an isotropic elastic solid, including driving forces, is [14]

\[
\rho \frac{\partial^2 \mathbf{u}(\mathbf{r}, t)}{\partial t^2} = (\lambda + 2G) \nabla^2 \mathbf{u}(\mathbf{r}, t) - G \nabla \times (\nabla \times \mathbf{u}(\mathbf{r}, t)) + \text{driver} \tag{1}
\]

where \( \lambda \) and \( G \) are Lame constants and \( \rho \) is the substrate density. If we let

\[
c = \left[ \frac{\lambda + 2G}{\rho} \right]^{1/2} \tag{2}
\]

then \( c \) will have the dimensions of speed [12]. Considering only the longitudinal component of the wave, for which \( \nabla \times \mathbf{u} = 0 \), equation (1) becomes
\[
\frac{\partial^2 u(\vec{r},t)}{\partial t^2} = \frac{1}{c^2} \nabla^2 u(\vec{r},t) + \text{driver} \tag{3}
\]

Since the surface wave is essentially two-dimensional, cylindrical coordinates will be used to express the wave equation. If we assume angular symmetry and no variation of displacement in the axial direction, equation (3) reduces to the following

\[
\frac{\partial^2 u(r,t)}{\partial r^2} + \frac{1}{r} \frac{\partial u(r,t)}{\partial r} - \frac{1}{c^2} \frac{\partial^2 u(r,t)}{\partial t^2} = -\frac{\delta(r) \delta(t)}{2\pi r} \tag{4}
\]

where a driver consisting of an impulse in both position and time has been chosen.\(^1\) It should be noted that the use of a two dimensional wave equation to model the propagation of a surface acoustic wave, a three dimensional phenomenon, is certainly not rigorous. This approach is common however [3,14,15], and it can be shown that errors introduced into the analysis through the use if this equation can either be compensated for or are negligible.

Taking the Hankel transform of order zero with respect to \(r\) of each term in equation (4) yields [19,20]:

\[
\mathcal{H}_0 \left[ \frac{1}{c^2} \frac{\partial^2 u(r,t)}{\partial t^2} \right] = \frac{1}{c^2} \frac{\partial^2 U(\rho,t)}{\partial t^2} \tag{5a}
\]

\(^1\) The delta function in cylindrical coordinates, \(\delta(r)/\pi r\), can also be represented as the limit of a cylinder as its radius shrinks to zero and its height increases to infinity, i.e.: \(\lim_{a \to 0} \frac{\delta(a-r)}{\pi a^2}\). The solution will be approached slightly differently, but the result is the same.
\[ \mathcal{H}_o \left\{ \frac{\partial^2 u(r,t)}{\partial r^2} + \frac{1}{r} \frac{\partial u(r,t)}{\partial r} \right\} = -\rho^2 U(\rho,t) \] (5b)

\[ \mathcal{H}_o \left\{ -\frac{\delta(r) \delta(t)}{2\pi r} \right\} = -\frac{\delta(t)}{2\pi} \] (5c)

Equation (4) now becomes

\[ \frac{\partial^2 U(\rho,t)}{\partial t^2} + \rho^2 c^2 U(\rho,t) = c^2 \frac{\delta(t)}{2\pi} . \] (6)

Solving this differential equation, subject to initial conditions of zero velocity and displacement at time 0 and the requirement that the response remain finite as \( \rho \to \infty \) yields:

\[ U(\rho,t) = \frac{c \sin(ct)}{2\pi \rho} \] (7)

Taking the inverse Hankel Transform yields the time domain point to point impulse response given by

\[ u(r,t) = \frac{c h(ct-r)}{2\pi \left[ (ct)^2 - r^2 \right]^{1/2}} . \] (8)

The Fourier Transform of equation (8) can be obtained using the following relationship between the Fourier and Laplace Transforms

\[ \mathcal{F}(f(t); \omega) = \mathcal{L}(f(t); s \to j\omega) \] (9)

combined with the relationship
The frequency domain point to point impulse response is thus

$$U(r, \omega) = \frac{1}{2} \chi_0 \left[ \frac{j\omega r}{c} \right]$$  \hspace{1cm} (11)

where $\chi_0$ is the Bessel function of the second kind. Since the Bessel function is difficult to evaluate for use in a practical diffraction analysis, it is useful to consider approximations to this result. Factoring the denominator of equation (8) as follows

$$u(r, t) = \frac{c h(ct-r)}{2\pi [(ct+r)(ct-r)]^{1/2}}$$  \hspace{1cm} (12)

and considering times for which $(ct+r=2r)$, which preserves the behavior of this function near its branch point, yields the approximate time domain impulse response for a point source

$$u'(r, t) = \frac{c h(ct-r)}{2\pi [2r(ct-r)]^{1/2}}$$  \hspace{1cm} (13)

The approximate frequency response given by the Fourier transform of equation (13) is

$$U'(r, \omega) = \frac{1}{4\pi} \left[ \frac{2\pi c}{j\omega r} \right]^{1/2} e^{-j\omega r/c}$$  \hspace{1cm} (14)

This result can also be obtained using an asymptotic approximation for the Bessel function of the second kind, and for most practical transducer geometries and substrates is indistinguishable from
the exact transform expression for frequencies above a few megahertz (see Figure 2). Since this approximation to the ideal point-to-point impulse response is valid in the high frequency region of interest (roughly 10 MHz to several GHz), it can be used along with Huygen's Principle to determine the response at any frequency at a point due to an impulse applied to a tap.
Figure 2. Comparison of Exact and Approximate Point to Point Frequency Response.
CHAPTER IV
TAP TO POINT RESPONSE

Using the principle of superposition, or Huygen's Principle, the response at any given observation point to an impulse applied to a tap can be found by integrating the point to point impulse response along the tap \([4,5,9,21-24]\). This yields the following response in the time domain

\[
 h_{t-p}(y_p,t) = \int_0^L \frac{c}{2\pi} \frac{h(ct - r)}{[(ct)^2 - r^2]^{1/2}} \, dy
\]  

where \( y \) is the position along the transmitting tap, \( y_p \) is the position of the observation point relative to a line parallel to the bottom of the transmitting tap, and \( L \) is the tap length as shown in Figure 3.

This integration yields a response at point \( y_p \) such as the one shown in Figure 5. From Figure 4, it can be seen that no response is expected at the point \( y_p \) until a time \( t_0 = D/c \) has been reached, since this is the minimum amount of time it will take the wave to travel from the transmitter to the observation point. From time \( t_0 \) to \( t_1 \), there are two regions of the transmitting tap making equivalent contributions to the response at point \( y_p \), while from \( t_1 \) to \( t_2 \) only one region contributes directly, and for times beyond \( t_2 \) there are no new or 'direct' contributions from the tap to the response at
Figure 3. Coordinate System for Tap to Point Integration.
Figure 4. Propagation Times for Tap to Point Impulse Response.
Figure 5. Tap to Point Impulse Response.
The behavior of the impulse response shown in Figure 5 reflects these time limits. Prior to time $t_0$, no response is detected at $y_p$. From $t_0$ to $t_1$, the response is fairly constant, reflecting the two regions contributing to the output during this time. The response then drops off from $t_1$ to $t_2$, the time frame when only one direct contribution is being made. For times beyond $t_2$, the response falls off towards zero, as required by the form of the solution to the two dimensional wave equation. The 'tail' on this response for times beyond $t_2$ is purely a mathematical by-product of the formulation of this model, and would not be observed on a real surface wave device. In the frequency domain, this is reflected by a point to point impulse response that varies as $(1/\omega)^{1/2}$, rather than being constant for all frequency. This artifact can be removed after the geometrical diffraction analysis has been performed by multiplying the frequency response obtained by $\omega^{1/2}$.

In the frequency domain, the tap to point integral becomes

$$H_{c\rightarrow p}(y_p, \omega) = \frac{1}{4\pi} \int_0^L \left[ \frac{2\pi c}{j\omega r} \right]^{1/2} e^{-j\omega r/c} dy$$

where the approximate frequency response given by equation (14) has been used. Equation (16) is evaluated numerically and used to generate the transverse intensity or "beam" profiles at various distances from the source tap.
CHAPTER V

TAP TO TAP RESPONSE

In order to determine the impulse response of a pair of parallel taps of arbitrary lengths and positions relative to one another, a double integral over the taps must be performed. This integral can be expressed in the time domain as follows

$$\int_{y=R}^{y=a} \int_{y=-b}^{y=b} \frac{h(ct - r)}{2\pi \sqrt{2 \frac{(ct)}{2} - r}^{1/2}} \ dy \ dy$$

or in the frequency domain as

$$\int_{y=R}^{y=a} \int_{y=-b}^{y=b} \frac{\sqrt{2\pi c}}{\sqrt{j\omega r/c}} e^{-j\omega r/c} \ dy \ dy$$

where $y_R$ is the coordinate along the 'receiving' tap, $y_t$ is the coordinate along the 'transmitting' tap relative to a horizontal line through $y_R$, and $a$, $b$, $L$, $D$, and $r$ are as shown in Figure 6. The receiving tap is shown as shorter than the transmitting one for convenience, but by reciprocity the response of a short tap trans-
Figure 6. Coordinate System for Tap to Tap Double Integral.
mitting to a longer tap would be identical [3,25].

In evaluating this double integral, a technique called double integral reduction, as presented by Datta [1], is used to reduce equation (17) or (18) to a single integral. As shown in Figure 7, for each angle, theta, the energy launched at this angle by the transmitting tap is distributed evenly from \( Y_{\text{min}} \) to \( Y_{\text{max}} \). Since all transmitting and receiving point pairs related by the same angle theta are equivalent [4], the value of the integrand for this value of theta is a constant, and the integral along the receiving tap can be expressed simply as the width of the signal received. This width can be found by determining the overlap between the shadow of the transmitter for this angle, which extends from \( Y_{\text{min}} \) to \( Y_{\text{max}} \), with the limits of the receiving tap, \( a \) and \( b \). Since this received width is dependent solely on theta for a fixed transducer geometry, and theta depends on \( y_{\|} \), the double integral in equation (17) or (18) is reduced to a single integral over \( y_{\|} \) as shown

\[
\int_{Y_{\|}=a}^{Y_{\|}=b} \int_{Y_{\|}=a}^{Y_{\|}=b} F(r) \, dy_{\|} \, dy_\| = \left\{ \begin{array}{ll}
\int_{Y_{\|}=a}^{Y_{\|}=b} F(r)(b+y_{\|}) \, dy_{\|} \\
\int_{Y_{\|}=a}^{Y_{\|}=b} F(r)(L-y_{\|}-a) \, dy_{\|}
\end{array} \right. 
\]

(19)

where \( F(r) \) represents either \( u(r,t) \) or \( U'(r,\omega) \), depending on the domain in which the integral is performed. This set of single integrals can be visualized by drawing out the area of integration of the double integral in the \( Y_{\|} \), \( y_{\|} \) plane. As shown in Figure 8, the
Figure 7. Representation of Integral Limits Used in Double Integral Reduction.
Figure 8. Area of Integration for Double Integral of Equations (17) and (18).
width in the $y_r$ direction can be expressed as a simple function of $y_T$, reducing equation (17) or (18) to the set of integrals in equation (19). Despite the simplicity of this approach and the degree of simplification it yields calculationally, double integral reduction is seldom discussed in the literature [1,4], and does not seem to be in widespread use at the present time.
In the analysis thus far, the response of two 'taps' of arbitrary lengths and positions relative to one another has been developed. However, when considering an entire transducer, the location of these 'taps' must be defined [1,5]. In the analysis to follow, the transducer will be regarded as a discrete set of sources, or taps, each of which is located midway between two transducer fingers (see Figure 9). The assumption that the tap on the acoustic delay line occurs in the middle of the gap between fingers is based on this being the region of maximum torque created by a voltage difference between fingers, and hence it is a region transduction from electrical to mechanical energy. The length of each effective tap is equal to the length of the region of overlap between fingers of opposite polarities [5]. If two adjacent fingers have the same polarity, then there will not be a torque created on the dipoles in the substrate between the two fingers (which are at equal potentials), and therefore the tap located between these fingers will have an effective length of zero. Although this approach does not take into consideration the charge distribution on the taps, a factor similar to the element factor [3,13,25] could be incorporated to include this effect.

The first case considered is the most general case: two
Figure 9. Transducer Geometry for Transducer to Transducer Analysis. Effective taps are indicated by dashed lines located between transducer fingers.
arbitrarily apodized transducers, the first with center frequency \( f_{01} \) and \( M \) fingers and the second with center frequency \( f_{02} \) and \( N \) fingers. The transducers are separated by a center to center distance \( Z_0 \), they can have any apertures desired, and their vertical centers are offset by an arbitrary amount (see Figure 10). The overall impulse response of this device in the time domain will be given by [13,25]

\[
h(t) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} h_{ij}(t)
\]

where \( h_{ij} \) represents the tap to tap impulse response given by equation (19) with an appropriate sign included to take into account the relative polarities of the taps. The convention used in this model is that an effective tap located between a negative finger on the left and a positive finger on the right is considered a positive tap, and vice versa for a negative tap. This method of calculating the impulse response including diffraction is essentially a brute force method, involving \((M-1)(N-1)\) calculations of the diffraction integral. Despite the time savings obtained by using Double Integral Reduction, this can still be quite a lengthy calculation.

Fortunately, it is not usual for two apodized transducers of different center frequencies to be cascaded directly. A much more common arrangement, and one which lends itself to a greatly reduced number of calculations, is the case of an unapodized or uniform transducer cascaded with an apodized transducer of the same center
Figure 10. Effective Tap Geometry for Two Arbitrary Transducers.
frequency. We will also assume that the sampling rates in the two transducers are equal, so that the time intervals $\Delta T$ shown in Figure (11) are the same.

As can be seen from the geometrical arrangement in Figure 12, the distance (or equivalently the time) between the first tap in the first transducer and the first tap in the second transducer is equal to the distance between the second tap in the first transducer and the second tap in the second transducer. This symmetry continues for many of the other tap pairs, and it can be shown that there are only $(M-1)+(N-2)$ possible tap pair separations. Thus, it is only necessary to calculate the diffraction integral for this small number of geometries, as compared to $(M-1)(N-1)$ calculations for the general, brute force method. The diffraction field of a tap of length equal to those in the uniform input transducer can be calculated at each distance of interest, and the amplitude of the response reaching each of a number of points across the transverse distance being considered at the output can be stored in an array. These values will also be calculated for many times within a time range of interest. Once these $(M-1)+(N-2)$ basic diffraction field responses have been calculated, all that is necessary to do in order to find the overall response of the two transducers is to add up the amplitude responses that fall within the spatial extent of each effective tap in the output apodized transducer at each point in time, taking into account the polarities of the effective taps (see Figure 13).
Figure 11. Effective Tap Geometry for One Uniform and One Apodized Transducer.
Figure 12. Equivalent Times Between Transducers. When transducer center frequencies are equal, symmetry allows for a reduction in the number of calculations necessary.
Effective Taps of Transmitting Transducer

Effective Taps of Output Transducer

Figure 13. Utilizing Simplifying Symmetry for Transducers With Equal Center Frequencies. The diffracted field is calculated at many points and several distances (dots). Spatial extent of output taps is shown by the boxes, and can be used to find overall response.
CHAPTER VII
EXTENSION TO THE ANISOTROPIC CASE

The wave equation in cylindrical coordinates used in this model assumes no variation in response with propagation direction on the substrate. However, it is well known that on anisotropic materials, the phase velocity of the surface wave and the electromechanical coupling coefficient both vary with propagation direction [1,3-5,26-29]. In order to analyze the anisotropic case fully, it would be necessary to return to the wave equation in cylindrical coordinates (equation (3)) and solve the equation that arises when angular symmetry is not assumed

\[
\frac{\partial^2 u(r,\phi,t)}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \phi^2} - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = -\frac{\delta(r) \delta(t)}{2\pi r}. \quad (21)
\]

This equation still assumes no variation in displacement with axial direction, since we are considering a two dimensional wave. Since it adds considerable mathematical complexity to include anisotropy in the wave equation itself, a first approximation to the anisotropic displacement is made by substituting surface wave velocity as a function of angle, \( c(\phi) \), directly into the solution to the wave equation for the isotropic case. This approximation assumes that there is a small fractional deviation in velocity with a change in angle.
A look up table for $c(\theta)$ on YZ LiNbO$_3$ was generated using a fifteen term trigonometric expansion, as described by Anhorn, Engan, and Ronneklev [30] (see Figure 14). The expansion for both $c(\theta)$, the phase velocity, and $\Delta c$, the change in phase velocity which occurs when the surface is electrically shorted with an ideal conductor of zero thickness is as follows

$$g_i(\theta) = \sum_{n=0}^{15} g_{i,n} \cos(2n\theta) \tag{22}$$

where $g_i$ can represent either quantity, and the coefficients for the two cases are given in Table 1. These coefficients were obtained by surface wave velocity measurements taken using a laser probe, and by fitting the expansion above to the measured data.

In addition to the velocity variation with propagation direction, the electromechanical coupling coefficient will also vary [3,12,31,32]. Although the entire discussion thus far has been about displacement, it is important to realize that this displacement will be brought about by the application of an electrical signal [3]. It is therefore necessary to define an electrical point to point impulse response, which includes the fact that the energy launched from an electrical point source into an anisotropic medium will not be evenly distributed, with more energy propagating in certain directions than in others. Modifying the time domain impulse response as shown
\[ u(r, \theta, t) \propto \frac{c(\theta) h(c(\theta)t-r)}{[(c(\theta)t)^2 - r^2]^{1/2}} \left\{ \frac{K(\theta)}{K_{\text{max}}} \right\} \]  

(23)

using a normalized version of the coupling coefficient will take this into account. The final response of a transducer can be multiplied by \( K_{\text{max}} \), the maximum value of \( K(\theta) \), in order to take the overall coupling efficiency into account. Several techniques for calculating \( K(\theta) \) were found in the literature \([12, 24, 27-29]\). Two

<table>
<thead>
<tr>
<th>n</th>
<th>( c_0 ) (m/s)</th>
<th>( \Delta c_p ) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3550.98</td>
<td>43.87</td>
</tr>
<tr>
<td>1</td>
<td>-137.83</td>
<td>19.76</td>
</tr>
<tr>
<td>2</td>
<td>42.41</td>
<td>5.62</td>
</tr>
<tr>
<td>3</td>
<td>17.43</td>
<td>3.80</td>
</tr>
<tr>
<td>4</td>
<td>6.21</td>
<td>.98</td>
</tr>
<tr>
<td>5</td>
<td>2.78</td>
<td>.08</td>
</tr>
<tr>
<td>6</td>
<td>1.76</td>
<td>.55</td>
</tr>
<tr>
<td>7</td>
<td>1.20</td>
<td>.48</td>
</tr>
<tr>
<td>8</td>
<td>.57</td>
<td>.25</td>
</tr>
<tr>
<td>9</td>
<td>.49</td>
<td>.32</td>
</tr>
<tr>
<td>10</td>
<td>.27</td>
<td>.00</td>
</tr>
<tr>
<td>11</td>
<td>.13</td>
<td>-.10</td>
</tr>
<tr>
<td>12</td>
<td>.15</td>
<td>.03</td>
</tr>
<tr>
<td>13</td>
<td>.04</td>
<td>.04</td>
</tr>
<tr>
<td>14</td>
<td>.08</td>
<td>.07</td>
</tr>
<tr>
<td>15</td>
<td>.06</td>
<td>.07</td>
</tr>
</tbody>
</table>

methods were used in this investigation, and compared well. First, the coupling coefficient can be calculated from the anisotropic dielectric tensor components

\[ K^2 = \left[ 1 + (\varepsilon_{pr}^\top)^{-1} \right] \left[ \frac{\Delta c}{c} \right] \left[ 1 - \frac{\Delta c}{c} \right]^{-1} \]  

(24)

where \( c \) is the free surface velocity, \( \Delta c \) is the change in velocity between free surface and a shorted surface wave, and \( \varepsilon_{pr}^\top \) is the relative equivalent dielectric constant given by

\[ \varepsilon_{pr}^\top = (\varepsilon_{11} - \varepsilon_{13}^2)^{1/2} = \varepsilon_0 \varepsilon_{pr}^\top \]  

(25)

and \( \varepsilon_0 \) is the permittivity of free space [24,27,29]. The direction of propagation of the surface wave is direction 1, and direction 3 is normal to the surface, in the sagittal plane. The dielectric constants for a given propagation direction can be determined simply by rotating the tensors for these constants to the appropriate orientations. Another approximation for \( K \) which is much simpler was discussed by Kino [28]

\[ \left| \frac{\Delta c}{c} \right| = \left| 1 - \left[ 1 + K^2 \right]^{1/2} \right| \]  

(26)

Despite the apparent simplicity of this approximation, it seems to yield results similar to equation (24).
Figure 14. Velocity Data Generated for YZ LiNbO₃
(a) Phase velocity as a function of direction relative to Z-axis.
(b) Difference between free surface and shorted surface velocity.
CHAPTER VIII
RESULTS AND DISCUSSION

Tap to Point Response

Equation (14) was used to calculate the response at various points across the transverse region of interest and thereby generate beam profiles at a given frequency and at several values of the scaled distance from the tap, $W$. $W$ is a dimensionless parameter given by

$$W = \frac{\lambda D}{a^2} \quad (20)$$

where $D$ is the normal distance from tap to point (as shown in Figure 3), $\lambda$ is the wavelength at the frequency of interest, and the transmitting tap is of length $L=2a$. This method was successful, however it turned out that Double Integral Reduction was able to contribute significantly to saving time in this calculation as well as others. This approach, which was much more efficient, involved calculating equation (19), and keeping track of where the energy was ending up in the transverse direction while performing the integration. This enabled one integration to yield information about the response across a range of spatial bins, thereby generating the entire beam profile in one integration. This greatly reduced calculation times as compared to beam profile generation using equation 16, where an integral had to be performed for each output point of interest.

37
For the isotropic case, the results obtained using the impulse driver approach compare well with those obtained using the more traditional approximation to the point to point impulse response, $e^{ikr}/(r)^{1/2}$ [4,5,18], as shown in Figure 15. For values of $W$ less than 4, near field profiles in which most of the energy launched by the transmitting tap remains within a transverse distance roughly equal to the tap width are observed. As $W$ increases, the beam profiles become those more characteristic of the Fraunhofer region, with a significant amount of energy spreading out transversely beyond the edges of the direct beam path. When $W=4$, the traditional $(\sin(x)/x)^2$ response indicating the onset of the Fraunhofer region is observed [9,11]. Note that the integration technique developed using the impulse response driver model is valid for all ranges.

Since anisotropy does not have as great an effect on the response felt at a given point as it does on the response felt by an entire region (or tap), and since the effects of anisotropy are greatly reduced when a wave is launched on a pure mode axis, the beam profiles generated for YZ LiNbO$_3$ with no misorientation of the propagation direction from the Z axis were very similar to those generated for the isotropic case. The profiles differ from the isotropic profiles only in the near field, where they compare favorably with those generated by Farnell for the case of parabolic anisotropy [11]. Inclusion of transducer misorientation in the analysis should have a major effect on these profiles, causing power flow to result.
Figure 15. Intensity Profiles at Successive Values of the Scaled Distance $W$ for Diffraction on an Isotropic Substrate.
(a) Profiles obtained using equation (16) (times $10^6$).
(b) Normalized profiles using $e^{ikr}/(r)^{1/2}$ impulse response approximation [4].
Since it is possible to derive an approximate point to point impulse response which includes the effects of anisotropy (equation (23)), Huygen's principle can be applied using this approximate Green's function for anisotropic substrates. Now, since a point source radiates energy radially outward, although not necessarily uniformly, the waves generated by many point sources along a line source will add constructively and destructively. Thus if a transducer launches a wave off axis, we expect that, provided the point source response includes the anisotropy correctly, the phenomenon of power flow should appear as a natural consequence of wave interference. The power flow angle was derived to describe the behavior of a plane wave, since for a plane wave launched off axis in an anisotropic substrate the phase velocity and the group (or energy) velocity differ in direction. The phase velocity remains in the direction of propagation of the wave, while the energy propagates off at an angle $\phi$ (the power flow angle) with respect to the propagation direction (see Figure 16). However, this type of analysis cannot be applied to a point source, since a point source launches energy in all directions, and each "bundle" of energy travels radially outward, with no distinction between the phase and group velocity. Indeed, the power flow angle really only has meaning when discussing a wave launched by an extended source. It is gratifying to see, therefore, that beam profiles generated using Huygen's Principle with the impulse driver point source impulse response demonstrate the process of power flow (see Figures 17 and 18). Beam profiles generated using the approximation $e^{ikr}/(r)^{1/2}$
Figure 16. Acoustic Beam Trajectory in an Anisotropic Substrate, Illustrating Power Flow.
also demonstrate power flow [33]. As can be seen, the energy travels off in a direction different than the direction of propagation, causing the beam to walk off to one side. The power flow angles observed are slightly lower than those predicted by plane wave theory. This may be due to the fact that the profiles were generated far enough away from the tap in question that it no longer looked like the generated wave is planar.

Tap to Tap Response

Tap to tap impulse response curves were generated in the time domain from equation (17), using the double integral reduction technique. Frequency domain responses were obtained from equation (18) and inverse Fourier transformed, yielding time domain tap to tap impulse responses in good agreement with those calculated in the time domain.

Figure 19 shows the impulse response of equal length tap with no transverse offset on an isotropic substrate. As expected, no response is observed until the signal launched by the transmitting tap has had time to propagate to the receiving tap. After the impulse initially reaches the receiving tap, the response dies off rapidly. The impulse response for this configuration on Y-cut LiNbO$_3$ with propagation along the Z axis is almost indistinguishable from the isotropic response. If the taps are of equal length, but have a 50% transverse offset, the impulse responses for the isotropic case and for YZ LiNbO$_3$ are as shown in Figure 20. As can be seen, there is now a short time range during which direct contri-
Figure 17. Beam Profiles at Various Normalized Distances and a 1 Degree Angle of Propagation on YZLiNbO$_3$. 
Figure 18. Beam Profiles at Various Normalized Distances and a 3 Degree Angle of Propagation on YZLiNbO₃.
Figure 19. Tap to Tap Impulse Response for Two Taps of Equal Length With No Transverse Offset.
butions from different parts of the transmitting tap reach the receiving tap, followed by a rapid drop similar to the case of no offset. As expected, the impulse response on the anisotropic substrate is broadened slightly compared to the isotropic case. Once again, inclusion of launching misalignments significantly affect these results. Behavior was also investigated and impulse responses obtained for taps of arbitrary lengths and transverse positions relative to one another, yielding expected results.

Transducer to Transducer Responses

Transducer to transducer responses were evaluated for several cases. It was observed that the computation time for two transducers with a given number of taps is considerably lower for the case where one is uniform and the center frequencies are the same. Typical responses in the time domain are shown in Figures xx and aa, and an FFT can be performed [34], yielding the frequency responses shown.

In order to understand the behavior of the complete transducer to transducer impulse response, we first consider the responses of smaller elements that make up these transducers. The results for an isotropic substrate will be considered first.

The impulse response observed between two single taps of equal length with no transverse offset is shown in Figure 19. Since these individual taps do not sample the wave at any specified frequency, they do not have any inherent frequency content. This is demonstrated in Figure 21, where there is no observed fundamental frequency for the response.
Figure 20. Tap to Tap Impulse Response for Two Taps of Equal Length With a Fifty Percent Transverse Offset.
Figure 21. Frequency Domain Response for Two Taps of Equal Length With a Fifty Percent Transverse Offset.
The time domain impulse response of a multi-tap transducer is simply the superposition of single tap impulse responses for all possible pairs of taps consisting of one tap in each transducer. When carrying out this superposition, the relative polarities of the taps must be taken into account. The expected result depends on the geometry of the particular transducers involved.

As an example, consider two uniform transducers, each with 6 effective taps, as shown in Figure 22. Since the envelope of the time domain impulse response of each transducer is a rectangle function, it is expected that the envelope of the output time domain response will be the convolution of two rectangle functions of equal length, or a triangle function. This expected behavior is obtained as a result of diffraction integral calculations, as shown in Figure 23. Considering the geometry, it can be seen that there are $M+N-1$, or 11 times at which individual tap pairs will interact. These times correspond to the sharp jumps observed in Figure 23. The first tap pair interaction occurs between a negative tap in one transducer and a positive tap in the second transducer. This causes a sharp jump downward, followed by the trailing off behavior previously seen in Figure 19. A time interval $\Delta T$ later, two pairs of taps are interacting, each providing positive "unit jump" contributions to the output. This contribution cancels what is left of the original negative jump and yields a positive result. The next time interval produces a negative jump of 3 units, and so on.

In a transducer with more than one tap, there is a built in sampling of the wave due to tap spacing. According to sampling
Figure 22. Two Uniform Transducers With Six Taps Each.
Figure 23. Time Domain Response for the Interaction of Two Uniform Transducers. Transducers have equal apertures, six taps each, and no transverse offset, as shown in Figure 21.
theory [34], this should result in a characteristic frequency response. A signal with a baseband frequency response, when sampled at frequency \( f_s \) will result in the duplication of the baseband response centered at integral multiples of the sampling frequency. For a \( 2f_s \) sampled device, with fundamental response at \( f_0 \), this will result in responses at all the odd harmonic frequencies, including the third harmonic, \( 3f_0 \). Due to the distances involved in the diffraction calculation being larger (in wavelengths) for the harmonics than for the fundamental, it is expected that the harmonics will have somewhat smaller magnitude than the fundamental. Since the transducers under consideration (Figure 22) are \( 2f_s \) sampled, we would expect to observe a frequency domain response consisting of a fundamental and a third harmonic. As shown in Figure 24, this is exactly the response obtained using the impulse driver diffraction analysis. The third harmonic in Figure 24, however, is further reduced in magnitude when compared to the fundamental response due to the \( 1/(\omega)^{1/2} \) characteristic of the solution to the two-dimensional wave equation. In order to remove the effects of this mathematical artifact, we multiply the frequency response by \( (\omega)^{1/2} \), which yields the result shown in Figure 25. Inverse Fourier transforming this response yields the time domain response shown in Figure 26. It is interesting to note that by increasing the level of the third harmonic, the gradual trailing off in time now appears as dips instead. This response does not include an element factor to describe the charge distribution within the transducer, but one could be included in future work, and may have the effect of par-
Figure 24. Frequency Domain Response Corresponding to Response Shown in Figure 23. Note the presence of the third harmonic, characteristic of 2f₀ sampling.
Figure 25. Frequency Domain Response Multiplied by \((\omega)^{1/2}\).
Figure 26. Time Domain Response Obtained by Inverse Fourier Transforming the Response Shown in Figure 25.
tially cancelling these dips.

In order to observe the effects of diffraction on predicted device performance, it is useful to consider a uniform transducer cascaded with an apodized transducer in an arrangement that yields good sidelobe rejection. In the example presented here, a six tap uniform transducer and a twenty-five tap apodized transducer of equal maximum apertures are cascaded with no transverse offset, and the responses are calculated on an isotropic substrate at various separation distances. Since propagation loss will affect the amplitude of the response, we will consider only relative response shapes. A degradation of the upper transition bandwidth with increasing separation distance is expected.

The time domain response shown in Figure 27 is similar to that shown previously for the simpler case of two unapodized transducers. Once again, the time domain response is easily interpreted, with varying size jumps corresponding to varying tap lengths in the apodized transducer. The frequency responses shown in Figure 28 are for two different transducer separation distances, and illustrate the expected spreading of the passband skirt due to diffraction for the larger separation distance. Similar response can be obtained for YZLiNbO₄.
Figure 27. Time Domain Response for a Six Tap Unweighted Transducer Cascaded With a 25 Tap Apodized Transducer.
Figure 28. Theoretical Predictions of Frequency Domain Responses for a Six Tap Unweighted Transducer Cascaded With a 25 Tap Apodized Transducer, Calculated for Two Separation Distances. Note the increased spreading of the passband skirts due to the greater effect of diffraction with increasing separation.
CHAPTER IX
CONCLUSION

This thesis has presented the basic development of a new model for the analysis of SAW diffraction. The impulse driver used in the wave equation yields a response which is valid for all frequencies. Use of double integral reduction simplifies calculations considerably, while performing integrals in the time domain provides clear insight into the geometrical basis for the responses obtained. Observed phenomena such as power flow, beam steering, and passband skirt spreading with increased transducer separation arise naturally as an extension of the theoretical considerations involved. Results compare extremely well with those obtained using a common approximation to the actual point source impulse response, $e^{ikr}/(r)^{1/2}$. Work is continuing to fully develop this model, including extending the analysis to devices utilizing multistrip couplers and arbitrary geometries. The effect of the $1/(\omega)^{1/2}$ dependence of the frequency domain point source response is still under investigation. Experimental work to confirm these theoretical results is beyond the scope of this thesis but will be pursued in the future. Use of this model should yield a diffraction analysis valid for broadband frequency filter responses with reduced computational time required when compared to conventional diffraction analysis. Future work
will involve developing an optimization routine to be utilized when designing practical SAW devices.
APPENDICES
APPENDIX A

TAP TO POINT AND TAP TO TAP INTEGRATION PROGRAMS
PROGRAM MAIN

SUBROUTINE NTPTIME

This routine calculates the response in the time domain at a point due to an impulse applied to a transmitting tap, using the exact point-to-point impulse response.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
CHARACTER*32 ONAME
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D0)

Enter Necessary Values
WRITE(*,*) 'Enter center frequency, in MHz:'
READ(*,*) fo
WRITE(*,*) 'Enter velocity of wave, in m/sec:'
READ(*,*) C
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths:'
READ(*,*) L
WRITE(*,*) 'Enter horizontal distance from tap to point, in wavelengths:'
READ(*,*) D
WRITE(*,*) 'Enter vertical position of point relative to bottom of tap, in wavelengths:'
READ(*,*) Yp
WRITE(*,*) 'Enter number of angular increments for integral:'
READ(*,*) ANGNUM
WRITE(*,*) 'Enter minimum time of interest, in microsec:'
READ(*,*) MININT
WRITE(*,*) 'Enter maximum time of interest, in microsec:'
READ(*,*) MAXINT
WRITE(*,*) 'Enter number of sample points desired:'
READ(*,*) SAMP
WRITE(*,*) 'Enter name of output data file:'
READ(*,10) ONAME

WVLENGTH = C/(fo*1.D0)
L = L*WVLENGTH
D = D*WVLENGTH
Yp = Yp*WVLENGTH
MININT = MININT*1.0D-06
MAXINT = MAXINT*1.0D-06
THETAMIN = DATAN2((Yp-L),D)

C Calculate step sizes for both y and time
YINCR = -L/ANGNUM
DELTAT = (MAXINT - MININT)/(SAMP-1)

C Loop 200 fixes a value of time at which to calculate the integral
C
OPEN(9,FILE=ONAME)
T=MININT
DO 200 J=1,SAMP
C
T=MININT
DO 200 J=1,SAMP
C
DO 100 I=1,ANGNUM
AI=I
Y = L +(AI*YINCR)
THETA = DATAN2((Yp-Y),D)
CALL RESPONSE(C,YINCR,D,T,THETAMIN,resp)
VNEW = RESP

C Provided we're not near 'edge' (branch point),
we can simply use trapezoidal rule for integral.

C
* IF(((VOLD .GT. 0.0) .AND. (VNEW .GT. 0.0)) .OR. 
  ((VOLD .EQ. 0.0) .AND. (VNEW .EQ. 0.0))) THEN 
  TERM = (VNEW + (VOLD-VNEW)/2.0)*(-YINCR) 
  SUM = SUM + TERM 
ELSE

C If we are near the branch point, must perform
integral here more carefully.
DOUBLEY = .TRUE.
IF (VNEW .EQ. 0.0) THEN 
exitng region where response is non-zero 
SIGN = -1.
REFY = Y - 2.*YINCR 
THETA = DATAN2((Yp-REFY),D)
CALL RESPONSE(C,YINCR,D,T,THETA,resp)
VREF = RESP

C Need to allow for when signal first arrives
at point
IF (VREF .EQ. 0.0) THEN
  REFY = Y-YINCR
\[
\text{THETA} = \text{DATAN2}((Y_p - \text{REF}_Y), D) \\
\text{CALL RESPONSE}(C, \text{YINC}_R, D, T, \text{THETA}, \text{resp}) \\
\text{VREF} = \text{RESP} \\
\text{DOUBLE}Y = .\text{FALSE}. \\
\text{ELSE} \\
\text{ENDIF} \\
\text{C} \\
\text{non-zero} \\
\text{ELSE} \\
\text{SIGN} = +1 \\
\text{REF}_Y = Y + \text{YINC}_R \\
\text{THETA} = \text{DATAN2}((Y_p - \text{REF}_Y), D) \\
\text{CALL RESPONSE}(C, \text{YINC}_R, D, T, \text{THETA}, \text{resp}) \\
\text{VREF} = \text{RESP} \\
\text{IF} (\text{VREF} .\text{EQ.} 0.0) \text{THEN} \\
\text{REF}_Y = Y \\
\text{THETA} = \text{DATAN2}((Y_p - \text{REF}_Y), D) \\
\text{CALL RESPONSE}(C, \text{YINC}_R, D, T, \text{THETA}, \text{resp}) \\
\text{VREF} = \text{RESP} \\
\text{DOUBLE}Y = .\text{FALSE}. \\
\text{ELSE} \\
\text{ENDIF} \\
\text{ENDDIF} \\
\text{C} \\
\text{Find exact location of the 'edge'} \\
\text{CTD} = ((C*T)**2 - D**2) \\
\text{YEDGE} = Y_p + \text{SIGN}^*\text{DSQR}T(\text{CTD}) \\
\text{VINIT} = \text{VREF} \\
V = \text{VINIT} \\
\text{DELV} = 50.0^*\text{VINIT}/\text{ANGNUM} \\
\text{EDGESUM} = 0.0 \text{DO} \\
\text{EVAL}_\text{OLD} = \text{ABS}((\text{REF}_Y - \text{YEDGE}) \\
\text{ELIMIT} = .00100^*\text{EVAL}_\text{OLD} \\
50 \\
\text{IF} (\text{EVAL}_\text{OLD} .\text{GT.} \text{ELIMIT}) \text{THEN} \\
V = V + \text{DELV} \\
\text{EVAL} = \text{ABS}((\text{YEDGE} - (Y_p + \text{SIGN}^*\text{DSQR}T(\text{CTD}-(
1.00^*\text{V}^2))))/2.0) + \text{DELV} \\
\text{EVAL}_\text{OLD} = \text{EVAL} \\
\text{GOTO} 50 \\
\text{ELSE} \\
\text{ENDIF} \\
\text{C} \\
\text{Keep track of integral so far} \\
\text{SUM} = \text{SUM} + (\text{VINIT}^*\text{ABS}((\text{YEDGE} - \text{REF}_Y)) + \text{EDGESUM} \\
\text{IF} (\text{DOUBLE}Y) \text{THEN} \\
\text{IF} (\text{SIGN} .\text{EQ.} -1) \text{THEN} \\
\text{SUM} = \text{SUM} - (\text{VNEW} + (\text{VOLD} - \text{VNEW})/2.0)*(-
\text{YINC}_R) \\
\text{ELSE} \\
\text{ENDIF} \\
\text{ENDIF} \\
\text{ENDIF} 
\]
VNEW = VINIT
Y = REFY
I = I + 1
ENDIF
ELSE
ENDIF
ENDIF
VOID = VNEW
CONTINUE
SUM = SUM*C/2.00/PI
C Store results of integral
WRITE(9,150) T,SUM
FORMAT(E16.8,1X,E16.8)
T = T + DELTAT
CONTINUE
CLOSE(9)
END

SUBROUTINE RESPONSE(C,YINCR,D,T,THETA,resp)
C
This routine calculates the response at a given point
and time.
IMPLICIT REAL*8(A-H,L-Z)
PI = 4.0*DATAN(1.0D+00)
R = D/COS(THETA)
IF (R .GE. C*T) THEN
  RESP = 0.0
ELSE
  EDENOM = DSQRT((C*T)**2 - R**2)
  RESP = 1/EDENOM
ENDIF
RETURN
END

SUBROUTINE TPFREQ(L,D,ANGNUM,YP,FINT,C,MAG,PHASE)
C
This routine calculates the approximate
response in the frequency domain at a point yp due to an
impulse applied to a transmitting tap. The tap is of length
L; D is the perpendicular distance from the transmitting
tap to yp; yp is the vertical location of the observation
point relative to the bottom of the transmitting tap.
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*2(I-K)
REAL*8 IPAR
PI=4.0*DATAN(1.0D+0000)
C Calculate values
    THETAMAX = DATAN2(yp,D)
    THETAMIN = DATAN2((yp-L),D)
C Set step sizes for integration along transmitting tap.
    YINCR = -L/ANGNUM
C Initialize variables
    RPART = 0.0
    IPART = 0.0
    MAG = 0.0
    PHASE = 0.0
    OMEGA = 2.0*PI*FINT
C Perform integral over transmitting tap.
C Find initial values
    Y = L
    THETA = DATAN2((YP-Y),D)
    A = (-YINCR)*DSQRT((C*DCOS(THETA))/(PI**2)*D*FINT))
    B = (OMEGA*D)/(C*DCOS(THETA))
    EOLD = A*DCOS(B)
    GOLD = (-1.0)*A*DSIN(B)
    DO 50 J = 0,ANGNUM
        AJ = J
        Y = L + (AJ*YINCR)
        THETA = DATAN2((YP-Y),D)
        A = (-YINCR)*DSQRT((C*DCOS(THETA))/(PI**2)*D*FINT))
        B = (OMEGA*D)/(C*DCOS(THETA))
        ENEW = A*DCOS(B)
        GNEW = (-1.0)*A*DSIN(B)
        RPART = RPART + ENEW + (EOLD-ENEW)/2.0
        IPART = IPART + GNEW + (GOLD-GNEW)/2.0
        EOLD = ENEW
        GOLD = GNEW
      50 CONTINUE
C
    MAG = (DSQRT((RPART**2)+(IPART**2))/(4.0*PI))
C
    PHASE = DATAN2(IPART,RPART) - (PI/4.0)
C
RETURN
END

PROGRAM MAIN
C
SUBROUTINE APRK
This routine calculates the beam profile at any specified frequency, on an isotropic or anisotropic substrate, using a setup similar to that used for double integral reduction, and incorporating the angular dependence of the electromechanical coupling coefficient.

```
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
CHARACTER*32 ONAME,FSUB,FSUB2,SNAME
REAL*8 VEL(1:181,1:2),DVEL(1:181,1:2),PROF(1:1500,1:3)
REAL*8 IM, KTHETA, KMAX
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D+00)
```

Enter Necessary Values

10 WRITE(*,*) 'Enter symbol for substrate name: [I-isotropic, P-parabolic, Q-quartz, YZ, 128]:'
READ(*,20) SNAME
20 FORMAT(A32)

Check for invalid input:

```
IF(SNAME .EQ. 'I') THEN
   FSUB = 'IVEL.DAT'
   FSUB2 = 'IDVEL.DAT'
ELSE IF(SNAME .EQ. 'P') THEN
   GOTO 30
ELSE IF(SNAME .EQ. 'Q') THEN
   GOTO 30
ELSE IF(SNAME .EQ. '128') THEN
   GOTO 30
ELSE IF(SNAME .EQ. 'YZ') THEN
   FSUB = 'YZVEL.DAT'
   FSUB2 = 'YZDVEL.DAT'
ELSE
   WRITE(*,*) 'Invalid entry, try again: '
   GOTO 10
ENDIF
```

```
WRITE(*,*) 'Enter center frequency, in MHz: '
READ(*,*) fo
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths: '
READ(*,*) LTEMP
WRITE(*,*) 'Enter lower limit for observation point relative to bottom of transmitting tap, in wavelengths: '
READ(*,*) MIN
WRITE(*,*) 'Enter upper limit for observation point relative to...
```
to bottom of transmitting tap, in wavelengths:
* READ(*,*) MAX
WRITE(*,*) 'Enter normal distance from tap to point, in
* wavelengths:
READ(*,*) D
WRITE(*,*) 'Enter angle at which wave is launched, with
* respect to pure mode axis of crystal (in degrees):
READ(*,*) ALPHA
WRITE(*,*) 'Enter number of angular increments for
integral:
READ(*,*) ANGNUM
WRITE(*,*) 'Enter number of sample points desired in
profile:
READ(*,*) YPSAMP
C Check for invalid input
35 IF (YPSAMP .GT. 1500) THEN
WRITE(*,*) 'Number of points specified is too large.
* Enter smaller value:
READ(*,*) YPSAMP
GOTO 35
ELSE
ENDIF
C WRITE(*,*) 'Enter frequency of interest, in MHz:
READ(*,*) FINT
WRITE(*,*) 'Enter name of output data file:
READ(*,40) ONAME
40 FORMAT(A32)
C C Read in velocity data appropriate for substrate choice
OPEN(9,FILE=FSUB,ACTION='READ')
OPEN(30,FILE=FSUB2,ACTION='READ')
DO 50 J=1,181
   READ(9,60) VEL(J,1),VEL(J,2)
   READ(30,60) DVEL(J,1),DVEL(J,2)
50 CONTINUE
60 FORMAT(E14.8,1X,E14.8)
CLOSE(9)
C C Find maximum value for k, the square root of coupling
C coefficient
CALL MAXK(SNAME,DVEL,KMAX)
C C Calculate values
Co=VEL(91,1)
WVLNGTH = Co/(fo*1.0D+06)
FINT = FINT*1.0D+06
L = LTEMP*WVLNGTH
D = D*WVLNGTH
YPMIN = MIN*WVLNGTH
YPMAX = MAX*WVLENGTH
THETAMIN = DATAN2((YFMIN-L),D)
THETAMAX = DATAN2(YFMAX,D)
ALPHA = (ALPHA/180.)*PI

Calculate step sizes for both y and time
YINCR = -(L-YFMIN+YPMAX)/ANGNUM
YPSTP = (YPMAX-YFMIN)/(YPsamp-1)

Initialize output array
DO 65 I=1,YPsamp
   AI=I
   PROF(I,1) = MIN-LTEMP/2.+((AI-1.0)*YPSTP/WVLENGTH)
   PROF(I,2) = 0.0
   PROF(I,3) = 0.0
65  CONTINUE

OPEN(9,FILE=ONAME)
WIDTH = 0.0

Loop 100 calculates the integral over transmitting tap

DO 100 I=1,(ANGNUM+1)
   AI=I
   Y = L - YFMIN +((AI-1.)*YINCR)
   THETA = DATAN2((-Y),D)
   BETA = ALPHA + THETA

Determine limits of beam at receiver for this angle
YMIN = D*DTAN(THETA)
YMAX = (D*DTAN(THETA))+L

IF(YMIN .LE. YFMIN) THEN
   LOWLIM = YFMIN
ELSE
   LOWLIM = YMIN
ENDIF

IF(YMAX .GE. YFMAX) THEN
   UPLIM = YFMAX
ELSE
   UPLIM = YMAX
ENDIF

Find out which output spatial 'bin' this is
LOWK = 1
K = 1

IF (((PROF(K,1)+(LTEMP/2.))*WVLENGTH) .LT. LOWLIM) THEN
   K = K + 1
   GOTO 70
ELSE
  LOWK = K
ENDIF

WIDTH = UPLIM - LOWLIM

JTEMP = (WIDTH/YPSTP)-1
K = K + JTEMP

* 80

IF (((PROF(K,1)+(JTEMP/2.))*WVINGIH) .LT. UPLIM) THEN
  K = K + 1
  GOTO 80
ELSE
  IF(K .GE. YPSAMP) THEN
    HIK = YPSAMP
  ELSE
    HIK = K - 1
  ENDIF
ENDIF

CALL subroutine to find velocity at this angle
CALL FINDVEL(SNAME, BETA, VEL, CINT)
C=CINT

Find next value of integrand for use in integral
CALL FRESFONSE(C, YINCR, D, THETA, FINI', RE, IM)

Find inclination factor for this angle
CALL FINDK(SNAME, BETA, DVEL, KIHETA)

Multiply integrand by inclination factor
RE = RE * KIHETA/KMAX*(-YINCR)
IM = IM * KIHETA/KMAX*(-YINCR)

Add response for this angle to appropriate points
DO 90 K=LOWK,HIK
  PROF(K,2) = PROF(K,2) + RE
  PROF(K,3) = PROF(K,3) + IM
90 CONTINUE

CONTINUE

Store results of profile calculations
DO 140 K=1,YPSAMP
  MAG = 0.25*DSQRT(PROF(K,2)**2+PROF(K,3)**2)
  IF (PROF(K,2) .EQ. 0.0) THEN
    PHASE = -PI/4.
  ELSE
    PHASE = DATAN2(PROF(K,3),PROF(K,2)) - PI/4.0
  ENDIF
  WRITE(9,150) PROF(K,1), MAG, PHASE
140 CONTINUE
150 FORMAT(E16.8,1X,E16.8,1X,E16.8,E16.8)
CLOSE(9)
END

SUBROUTINE FRESPONSE(C, YINCR, D, THETA, FINT, RE, IM)
This routine calculates the frequency domain pt-pet response
at a given angle theta between the ray connecting the two
points of interest and a line normal to the transducer.
IMPLICIT REAL*8(A-H, L-Z)
REAL*8 IM
PI = 4.0*DATAN(1.0D+00)
OMEGA = 2.0*PI*FIN
R = D/DCOS(THETA)
A = DSQRT(((C*DCOS(THETA))/((PI**2)*D*FIN))
B = (OMEGA*D)/(C*DCOS(THETA))
RE = A*DCOS(B)
IM = (-1.0)*A*DSIN(B)
RETURN
END

SUBROUTINE FINDVEL(SNAME, THETA, VEL, CINT)
This subroutine calculates the velocity at a given angle
by linear interpolation between data points in the array
VEL.
IMPLICIT REAL*8(A-H, L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 VEL(1:181,1:2)
CHARACTER*32 SNAME
PI = 4.0*DATAN(1.0D+00)

See if substrate is isotropic

IF (SNAME .EQ. 'I') THEN
   CINT = VEL(1,1)
ELSE

Substrate is anisotropic, must find velocity for angle
used

Determine which two angles in the array theta lies between
DO 100 I = 1,180
   AI = I
   TEMPL = -(PI/2.0) + (AI-1.)*PI/180.
   TEMPH = -(PI/2.0) + (AI)*PI/180.
   IF ((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN
   Interpolate between values at K and K+1 in array
   100 CONTINUE
   END
CID = VEL(I, 1)
ANGDIF = THETA - TEMPL
CINT = CID + (CHI-CID)*180.*(ANGDIF)/PI

100 CONTINUE

ENDIF

RETURN

END

SUBROUTINE FINDK(SNAME, THETA, DVEL, KTHETA)

This subroutine calculates k, the square root of
the electromechanical coupling coefficient, using
linear interpolation between points in an array of
delta(v)/v vs. theta (array name: DVEL).

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KTHETA
CHARACTER*32 SNAME
PI = 4.0*DATAN(1.0DO0)

See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
  KTHETA = DSQRT(2.0*DVEL(1,2))
ELSE

Substrate is anisotropic, must find k for angle used

DO 100 I = 1,181
  TEMPL = DVEL(I,1)*PI/180.
  TEMPH = DVEL(I+1,1)*PI/180.

  IF ((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN
    Interpolate between values at I and I+1 in array
    DVLO = DVEL(I,2)
    DVHI = DVEL(I+1,2)
    ANGDIF = THETA - TEMPL
    DVIHETA = DVLO + (DVHI-DVLO)*180.*ANGDIF/PI
  ENDIF

100 CONTINUE

Calculate the value for k using this delta(v)/v
KTHETA = DSQRT(2.0*DVIHETA)
SUBROUTINE MAXK(SNAME, DVEL, KMAX)

This subroutine searches the delta(v)/v array DVEL and returns the maximum value of k, the square root of the coupling coefficient.

IMPLICIT REAL*8(A-H, L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KMAX
CHARACTER*32 SNAME

See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
  KMAX = DSQRT(2.0*DVEL(1,2))
ELSE
Substrate is anisotropic, search for maximum
TEMPMAX = DVEL(1,2)
DO 100 I = 1,180
   IF(DVEL(I+1,2) .GE. TEMPMAX) THEN
      TEMPMAX = DVEL(I+1,2)
   ENDIF
100  CONTINUE
KMAX = DSQRT(2.0*TEMPMAX)
ENDIF

RETURN
END

PROGRAM MAIN

SUBROUTINE TAPRESP

This routine calculates the tap-to-tap impulse response in the time domain on an isotropic substrate, using a double integral reduction technique.

IMPLICIT REAL*8(A-H, L-Z)
IMPLICIT INTEGER*4(I-K)
CHARACTER*32 ONAME
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D+00)

C Enter Necessary Values
WRITE(*,*) 'Enter center frequency, in MHz:'
READ(*,*) f0
WRITE(*,*) 'Enter velocity of wave, in m/sec:'
READ(*,*) C
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths:'
READ(*,*) L
WRITE(*,*) 'Enter width of receiving tap, in wavelengths:'
READ(*,*) LOUT
WRITE(*,*) 'Enter horizontal distance from tap to point, in wavelengths:'
READ(*,*) D
WRITE(*,*) 'Enter vertical position of bottom of second tap relative to bottom of first tap, in wavelengths:'
READ(*,*) BDT
WRITE(*,*) 'Enter number of angular increments for integral:'
READ(*,*) ANGNUM
WRITE(*,*) 'Enter minimum time of interest, in microsec:'
READ(*,*) MININT
WRITE(*,*) 'Enter maximum time of interest, in microsec:'
READ(*,*) MAXINT
WRITE(*,*) 'Enter number of sample points desired:'
READ(*,*) SAMP
WRITE(*,*) 'Enter name of output data file:'
READ(*,10) ONAME

10 FORMAT(A32)

C Calculate values
WVLNGTH = C/(fo*1.D+06)
L = L*WVLNGTH
D = D*WVLNGTH
BDT = BDT*WVLNGTH
TOP = BDT + (LOUT*WVLNGTH)
MININT = MININT*1.0D-06
MAXINT = MAXINT*1.0D-06
THETAMIN = DATAN2((BOT-L),D)
THETAMAX = DATAN2(TOP,D)

C Calculate step sizes for both y and time
YINCR = -(L-BOT+TOP)/ANGNUM
DELTAT = (MAXINT - MININT)/(SAMP-1)

C Loop 200 fixes a value of time at which to calculate the integral
OPEN(9,FILE=ONAME)
T = MININT
DO 200 J=1,SAMP
  C Initialize values used in integral loop
  SUM=0.0
  TERM = 0.0
  WIDTH = 0.0
  O_WIDTH = 0.0
  CALL RESPONSE(C,YINC,D,T,THETAMIN,RESP)
  VOID = RESP
C
C Loop 100 calculates the integral over transmitting tap
C
DO 100 I=1,ANGNUM
  AI=I
  Y = L - BOT +(AI*YINC)
  THETA = DATAN2((-Y),D)
C Determine limits of beam at receiver for this angle
  YMIN = D*DTAN(THETA)
  YMAX = (D*DTAN(THETA))+L
C
  IF(YMIN .LE. BOT) THEN
    LOWLIM = BOT
  ELSE
    LOWLIM = YMIN
  ENDIF

  IF(YMAX .GE. TOP) THEN
    UPLIM = TOP
  ELSE
    UPLIM = YMAX
  ENDIF
C
  WIDTH = UPLIM - LOWLIM
C
C Find next value of integrand for use in integral
  CALL RESPONSE(C,YINC,D,T,THETA,resp)
  VNEW = RESP
C
C Provided we're not near 'edge' (branch point),
  we can simply use trapezoidal rule for integral.
C
  IF(((VOID .GT. 0.0) .AND. (VNEW .GT. 0.0)) .OR. 
    ((VOID .EQ. 0.0) .AND. (VNEW .EQ. 0.0))) THEN
    TERM = (VNEW + (VOID-VNEW)/2.0)*(-YINC)
    SUM = SUM + TERM*WIDTH
    O_WIDTH = WIDTH
  ELSE
   
C If we are near the branch point, must perform
C integral here more carefully.
  DOUBLEY = .TRUE.
IF (VNEW .EQ. 0.0) THEN
  exiting region where response is non-zero
  SIGN = -1.
  REFY = Y - 2.*YINCR
  THETA = DATAN2((-REFY),D)
  CALL RESPONSE(C,YINCR,D,T,THETA,resp)
  VREF = RESP
ENDIF

at point

IF (VREF .EQ. 0.0) THEN
  REFY = Y-YINCR
  THETA = DATAN2((-REFY),D)
  CALL RESPONSE(C,YINCR,D,T,THETA,resp)
  VREF = RESP
  DOUBLEY = .FALSE.
ELSE
  ENDIF
ELSE
  otherwise, entering region where response
  SIGN = +1
  REFY = Y + YINCR
  THETA = DATAN2((-REFY),D)
  CALL RESPONSE(C,YINCR,D,T,THETA,resp)
  VREF = EXP
  IF (VREF .EQ. 0.0) THEN
    REFY = Y
    THETA = DATAN2((-REFY),D)
    CALL RESPONSE(C,YINCR,D,T,THETA,resp)
    VREF = RESP
    DOUBLEY = .FALSE.
  ELSE
    ENDIF
ENDIF

Find exact location of the 'edge'

CTD = ((C*T)**2 - D**2)
VEDGE = SIGN*DSQRT(CTD)
VINIT = VREF
V = VINIT
DELV = 50.*VINIT/ANGNUM
EDGESUM = 0.DO
EVALOLD = ABS(REFY-VEDGE)
ELIMIT = .001DO*EVALOLD

50

IF (EVALOLD .GT. ELIMIT) THEN
  V = V+DELV
  EVAL = ABS(VEDGE-(SIGN*DSQRT(CTD-
(SIGN*DSQRT(CTD-
(1.DO/V**2)))))/2.DO)*DELV
  EDGESUM=EDGESUM+(EVAL+(EVALOLD-
EVAL))/2.DO)*DELV
ENDIF

EVALOLD = EVAL
C
C GOTO 50
ELSE
ENDIF
C Keep track of integral so far
SUM=SUM+((VINIT*ABS(YEDGE-REFY))+EDGESUM)*WIDTH
IF (DOUBLEY) THEN
  IF (SIGN .EQ. -1) THEN
    SUM=SUM-((VOID+VREF)/2.0)*(-YINCR)*OWIDTH
  ELSE
    VNEW = VINIT
    Y = REFY
    I = I + 1
  ENDIF
ELSE
ENDIF
ENDIF
VOID = VNEW
OWIDTH = WIDTH
100 CONTINUE
SUM = SUM*C/2.DO/PI
C Store results of integral
WRITE(9,150) T,SUM
150 FORMAT(E16.8,1X,E16.8)
T = T + DELTAT
200 CONTINUE
CLOSE(9)
END
C
C SUBROUTINE RESPONSE(C,YINCR,D,T,THETA,resp)
C This routine calculates the response at a given point
and time.
IMPLICIT REAL*8(A-H,L-Z)
PI = 4.0*Datan(1.0D+00)
R = D/Dcos(THETA)
IF (R .GE. C*T) THEN
  RESP = 0.0
ELSE
  EDENOM = DSQRT((C*T)**2 - R**2)
  RESP = 1/EDENOM
ENDIF
RETURN
END
C
C PROGRAM MAIN
C
SUBROUTINE TAPRESPF
This routine calculates the approximate tap-to-tap impulse response on an isotropic substrate in the frequency domain, using a double integral reduction technique.

```
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*2(I-K)
INTEGER*4 POINTS
PARAMETER(POINTS=16384)
REAL*8 FMAG(POINTS),FPHASE(POINTS),IPART
COMMON // FMAG,FPHASE
CHARACTER*32 FNAME9,FNAME8
PI=4.0*DATAN(1.0D+0000)
```

Enter necessary values
```
WRITE(*,*) 'Enter center frequency, in MHz: '
READ(*,*) fo
WRITE(*,*) 'Enter velocity of wave, in m/sec: '
READ(*,*) c
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths: '
READ(*,*) L
WRITE(*,*) 'Enter width of receiving tap, in wavelengths: '
READ(*,*) LOUT
WRITE(*,*) 'Enter horizontal distance between taps, in wavelengths: '
READ(*,*) D
WRITE(*,*) 'Enter vertical position of bottom of second tap, relative to bottom of first tap, in wavelengths: '
READ(*,*) BOT
WRITE(*,*) 'Enter number of angular increments desired for integral: '
READ(*,*) ANGNUM
WRITE(*,*) 'Enter frequency sampling rate, in samples per MHz: '
READ(*,*) FSAMP
WRITE(*,*) 'Enter minimum frequency of interest, in MHz: '
READ(*,*) FMIN
WRITE(*,*) 'Enter maximum frequency of interest, in MHz: '
READ(*,*) FMAX
WRITE(*,*) 'Enter name of output file for linear data: '
READ(*,15) FNAME9
WRITE(*,*) 'Enter name of output file for data in dB: '
READ(*,16) FNAME8
15 FORMAT(A32)
16 FORMAT(A32)
```

Calculate values
WVLNGTH = C/(fo*1.E+06)
L = L*WVLNGTH
D = D*WVLNGTH
BOT = BOT*WVLNGTH
TOP = BOT + (LOUT*WVLNGTH)
FMIN = FMIN*1.0D+06
FMAX = FMAX*1.0D+06
THETAMAX = DATAN2(TOP,D)
THETAMIN = DATAN2(BOT-L,D)
YINCR = -((L-BOT+TOP)/ANGNUM)

Set step sizes for frequency
DELTAF = 1.0D+06/FSAMP

Find the minimum number of frequency increments to get from FMIN to FMAX
N = 0
F = FMIN
10 IF (F.GT. FMAX) THEN
   F = F - DELTAF
   N = N+1
   GOTO 10
ENDIF

At least N frequency increments are required
NUMINCRF = N

Initialize variables
DO 120 J = 1,NUMINCRF
   FMAG(J) = 0.0
   FPHASE(J) = 0.0
120 CONTINUE
RFART = 0.0
IPART = 0.0
TRPART = 0.0
TIPART = 0.0

This loop sets a value for frequency, then performs the integral
of real and imaginary parts separately by double integral
reduction, then finds the magnitude and phase of the overall response
at this frequency.

OPEN(9,FILE=FNAME9)
OPEN(8,FILE=FNAME8)
DO 180 J = 1,NUMINCRF+1
   AJ = J
   F = FMIN + ((AJ-1.0)*DELTAF)
   OMEGA = 2.0*PI*F
C
C THIS LOOP PERFORMS INTEGRAL
DO 50, I = 1, ANGNUM+1
   AI = I
   Y = L - BOT + ((AI-1.0)*YINCR)
   R = DSQRT(Y**2 + D**2)
   THETA = DATAN2(-Y, D)
   YMIN = D*DTAN(THETA)
   YMAX = (D*DTAN(THETA)) + L
C
C angle

Determine limits of beam at receiver for this

IF(YMIN .LE. BOT) THEN
   LOWLIM = BOT
ELSE
   LOWLIM = YMIN
ENDIF

IF(YMAX .GE. TOP) THEN
   UPLIM = TOP
ELSE
   UPLIM = YMAX
ENDIF

WIDTH = UPLIM - LOWLIM

C

Calculate real and imaginary parts of integral

COEFF = 0.25*DSQRT((2.0*C)/(OMEGA*PI*R))
ARG = -((OMEGA*R)/C + (PI/4.0))
TRPART = COEFF*DCOS(ARG)*WIDTH*(-YINCR)
TIPART = COEFF*DSIN(ARG)*WIDTH*(-YINCR)
RPART = RPART + TRPART
IPART = IPART + TIPART

50 CONTINUE

FMAG(J+1) = DSQRT((RPART**2)+(IPART**2))
FPHASE(J+1) = DATAN2(IPART,RPART)

180 CONTINUE
DO 200 K=1, NUMINCRF+1
   AK = K
   F = FMIN + (AK-1.0)*DELTAF
   WRITE(9,210) (F/1.0D+06), FMAG(K), FPHASE(K)
   WRITE(8,210) (F/1.0D+06), (20.0*DLOG10(FMAG(K))),
                  (FPHASE(K)*180.0/PI)
200 CONTINUE

210 FORMAT(E18.8,1X,E18.8,1X,E18.8)
CLOSE (9)
CLOSE (8)
C
PROGRAM MAIN

SUBROUTINE ATAPRESPF

This routine calculates the approximate
tap-to-tap impulse response on an isotropic
or anisotropic substrate in the frequency domain,
using a double integral reduction technique.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*2(I-K)
INTEGER*4 POINTS
PARAMETER(POINTS=16384)
REAL*8 FMAG(POINTS),FPHASE(POINTS),IPART
REAL*8 VEL(1:181,1:2),DVEL(1:181,1:2)
COMMON // FMAG,FPHASE
CHARACTER*32 FNAMES9,FNAME8,FSUB,FSUB2
PI=4.0*DATAN(1.0D+0000)

Enter Necessary Values
10 WRITE(*,*) 'Enter symbol for substrate name: [I-isotropic,
P-parabolic, Q-quartz, YZ, 128]:'
READ(*,20) SNAME
20 FORMAT(A32)

Check for invalid input:
IF(SNAME .EQ. 'I') THEN
  FSUB = 'IVEL.DAT'
  FSUB2 = 'IDVEL.DAT'
ELSE IF(SNAME .EQ. 'P') THEN
  GOTO 30
ELSE IF(SNAME .EQ. 'Q') THEN
  GOTO 30
ELSE IF(SNAME .EQ. '128') THEN
  GOTO 30
ELSE IF(SNAME .EQ. 'YZ') THEN
  FSUB = 'YZVEL.DAT'
  FSUB2 = 'YZDVEL.DAT'
ELSE
  WRITE(*,*) 'Invalid entry, try again: '
  GOTO 10
ENDIF
WRITE(*,*) 'Enter center frequency, in MHz: '
READ(*,*) fo
WRITE(*,*) 'Enter velocity of wave, in m/sec: '
READ(*,*) c
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths:'
READ(*,*) L
WRITE(*,*) 'Enter width of receiving tap, in wavelengths:'
READ(*,*) LOUT
WRITE(*,*) 'Enter horizontal distance between taps, in wavelengths:
  * ' READ(*,*) D
WRITE(*,*) 'Enter vertical position of bottom of second tap, relative
  * to bottom of first tap, in wavelengths:' READ(*,*) BOT
WRITE(*,*) 'Enter angle at which wave is launched, with respect
  * to pure mode axis of crystal (in degrees):' READ(*,*) ALPHA
WRITE(*,*) 'Enter number of angular increments desired for integral:'
READ(*,*) ANGNUM
WRITE(*,*) 'Enter frequency sampling rate, in samples per MHz:'
READ(*,*) FSAMP
WRITE(*,*) 'Enter minimum frequency of interest, in MHz:'
READ(*,*) FMIN
WRITE(*,*) 'Enter maximum frequency of interest, in MHz:'
READ(*,*) FMAX
WRITE(*,*) 'Enter name of output file for linear data:'
READ(*,15) FNAME9
WRITE(*,*) 'Enter name of output file for data in dB:'
READ(*,16) FNAME8
!
15 FORMAT (A32)
16 FORMAT (A32)
C
C Read in velocity data appropriate for substrate choice
OPEN (9, FILE=FSUB, ACTION='READ')
OPEN (30, FILE=FSUB2, ACTION='READ')
DO 40 J=1,181
   READ(9,60) VEL(J,1),VEL(J,2)
   READ(30,60) DVEL(J,1),DVEL(J,2)
40 CONTINUE
60 FORMAT (E14.8,1X,E14.8)
CLOSE(9)
C
C Calculate values
CO=VEL(91,1)
WVLANGTH = CO/(fo*1.E+06)
L = L*WVLANGTH
D = D*WVLANGTH
BOT = BOT*WVLANGTH
TOP = BOT + (LOUT*WVLANGTH)
\[ F_{\text{MIN}} = F_{\text{MIN}} \times 1.0 \times 10^6 \]
\[ F_{\text{MAX}} = F_{\text{MAX}} \times 1.0 \times 10^6 \]
\[ \text{THETAMAX} = \text{DATAN2(TOP,D)} \]
\[ \text{THETAMIN} = \text{DATAN2(BOT-L,D)} \]
\[ \text{YINC} = -((L\text{-BOT+TOP})/\text{ANGNUM}) \]

C Set step sizes for frequency
DELTAF = 1.0 \times 10^6 / \text{FSAMP}
C
C Find the minimum number of frequency increments to get from
C FMIN to FMAX
N = 0
F = FMIN
11

IF (F .LT. FMAX) THEN
F = F + DELTAF
N = N + 1
GOTO 11
ENDIF

C At least N frequency increments are required
NUMINC = N
C
C Initialize variables
DO 120 J = 1, NUMINC + 1
FMAG(J) = 0.0
FMASE(J) = 0.0
120 CONTINUE
RPART = 0.0
IPART = 0.0
RE = 0.0
IM = 0.0
C
C This loop sets a value for frequency, then performs the
C integral
C of real and imaginary parts separately by double integral
C reduction,
C then finds the magnitude and phase of the overall response
C at this
C frequency.
C
OPEN(9, FILE=FNAME9)
OPEN(8, FILE=FNAME8)
DO 180 J = 1, NUMINC + 1
AJ = J
F = FMIN + ((AJ-1.0) \times DELTAF)
OMEGA = 2.0 \times \text{PI} \times F
180
C
C THIS LOOP PERFORMS INTEGRAL
DO 50, I = 1, ANGNUM + 1
AI = I
Y = L\text{-BOT} + ((AI-1.0) \times \text{YINC})
C

\[ R = \text{DSQRT}(Y**2 + D**2) \]
\[ \text{THETA} = \text{DATAN2}(-Y, D) \]
\[ \text{BETA} = \text{ALPHA} + \text{THETA} \]
\[ \text{YMIN} = D*\text{DTAN} (\text{THETA}) \]
\[ \text{YMAX} = (D*\text{DTAN} (\text{THETA})) + L \]

C

Determine limits of beam at receiver for this angle

\[ \text{IF}(\text{YMIN} \leq \text{BOT}) \text{ THEN} \]
\[ \text{LOWLIM} = \text{BOT} \]
\[ \text{ELSE} \]
\[ \text{LOWLIM} = \text{YMIN} \]
\[ \text{ENDIF} \]

C

\[ \text{IF}(\text{YMAX} \geq \text{TOP}) \text{ THEN} \]
\[ \text{UPLIM} = \text{TOP} \]
\[ \text{ELSE} \]
\[ \text{UPLIM} = \text{YMAX} \]
\[ \text{ENDIF} \]

C

\[ \text{WIDTH} = \text{UPLIM} - \text{LOWLIM} \]

C

Call subroutine to find velocity at this angle

\[ \text{CALL FINDVEL}(\text{SNAME}, \text{BETA}, \text{VEL}, \text{CINT}) \]
\[ \text{C=CINT} \]

C

Find next value of integrand for use in integral

\[ \text{CALL FRESNPE} (\text{C}, \text{YINC}, \text{D}, \text{THETA}, \text{F}, \text{RE}, \text{IM}) \]

C

Multiply integrand by dy

\[ \text{RE} = \text{RE} * \text{WIDTH} * (-\text{YINC}) \]
\[ \text{IM} = \text{IM} * \text{WIDTH} * (-\text{YINC}) \]

C

Add response for this angle to sum

\[ \text{RPART} = \text{RPART} + \text{RE} \]
\[ \text{IPART} = \text{IPART} + \text{IM} \]

C

\text{CONTINUE}

\[ \text{FMAG} (J) = \text{DSQRT}((\text{RPART}**2)+(\text{IPART}**2)) \]
\[ \text{FFHASE} (J) = \text{DATAN2} (\text{IPART}, \text{RPART}) \]

180 \text{CONTINUE}

DO 200 K=1, \text{NUMINCRF}+1

\[ \text{AK} = K \]
\[ \text{F} = \text{FMIN} + (\text{AK}-1.0) * \text{DELTAF} \]
\[ \text{WRITE}(9,210) (\text{F}/1.0D+06), \text{FMAG}(K), \text{FFHASE}(K) \]
\[ \text{WRITE}(8,210) (\text{F}/1.0D+06), (20.0*\text{DLOG10}(\text{FMAG}(K))), \]
\[ (\text{FFHASE}(K)*180.0/\text{PI}) \]

200 \text{CONTINUE}

210 \text{FORMAT} (E18.8,1X,E18.8,1X,E18.8)

CLOSE (9)
CLOSE (8)

STOP
END

SUBROUTINE FRESPONSE(C,YINCR,D,THETA,FINT,RE,IM)

This routine calculates the frequency domain pt-pt response at a given angle theta between the ray connecting the two points of interest and a line normal to the transducer.

IMPLICIT REAL*8(A-H,L-Z)
REAL*8 IM
PI = 4.0*DATAN(1.0D+00)
OMEGA = 2.0*PI*FINT
R = D/DCOS(THETA)
A = DSQRT((C*DCOS(THETA))/(PI**2)*D*FINT))
B = (OMEGA*D)/(C*DCOS(THETA))
RE = A*DCOS(B)
IM = (-1.0)*A*DSIN(B)
RETURN
END

SUBROUTINE FINDVEL(SNAME,THETA,VEL,CINT)

This subroutine calculates the velocity at a given angle by linear interpolation between data points in the array VEL.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 VEL(1:181,1:2)
CHARACTER*32 SNAME
PI = 4.0*DATAN(1.0d00)

See if substrate is isotropic

IF (SNAME .EQ. 'I') THEN
   CINT = VEL(1,1)
ELSE
   Substrate is anisotropic, must find velocity for angle used
   Determine which two angles in the array theta lies between
   DO 100 I = 1,180
      AI = I
      TEMPL = -(PI/2.0) + (AI-1.)*PI/180.
TEMFH = -(PI/2.0) + (AI)*PI/180.

C
IF ((TEMPL .LE. THETA) .AND. (TEMFH .GE. THETA)) THEN
  Interpolate between values at K and K+1 in array
  CLO = VEL(I,1)
  CHI = VEL(I+1,1)
  ANGDIF = THETA - TEMPL
  CINT = CLO + (CHI-CLO)*180.*(ANGDIF)/PI
ENDIF
100 CONTINUE
C
ENDIF
C
RETURN
END
APPENDIX B

TRANSDUCER TO TRANSDUCER INTEGRATION PROGRAMS
PROGRAM MAIN

SUBROUTINE TRANSRESP

This routine calculates the transducer-to-transducer impulse response in the time domain on an isotropic or anisotropic substrate, using a double integral reduction technique.

IMPLICIT REAL*8(A-H,I-Z)
IMPLICIT INTEGER*4 (I-K)
INTEGER*4 M,N,YPNUM,NSAMPT,SAMP,ANGNUM,NTAP1,NTAP2
INTEGER*4 SEPNUM,HIK,LOK,POS1,POS2,TNUM,YPNUM,TAPNUM
INTEGER*4 SEPNT,LOKTEMP,HIKTEMP
PARAMETER (TNUM=1400, YPNUM=10, TAPNUM=25)
CHARACTER*32 QNAME,FSUB,SNAME,DIREC,DIREC2,FN1,FN2
CHARACTER*32 FSUB2,FLAG1,FLAG2,FLAG3
CHARACTER*32 INFLAG,STORE,FSTORE
REAL*8 VEL(1:181,1:2),DVEL(1:181,1:2),T1DAT(1:TAPNUM,1:2)
REAL*8 T2DAT(1:TAPNUM,1:2),OUT(1:TNUM,1:3)
REAL*8 TIMEREC(TNUM),UNIOUT(1:TNUM,1:YPNUM,1:2)
REAL*8 TAPFIELD(1:TAPNUM,1:TNUM,1:YPNUM,1:2)
REAL*8 FIELD(1:TAPNUM,1:TNUM,1:YPNUM,1:2)
REAL*8 MAXANG,MAXPOSX,MINANG,KMAX
COMMON /B1/UNIOUT
COMMON /B2/OUT
COMMON /B3/T1DAT,T2DAT
COMMON /B4/TIMEREC,FIELD,TAPFIELD
LOGICAL*4 UNIF,EQUALT
PI = 4.0*DATAN(1.0D+00)

Enter Necessary Values
10 WRITE(*,*),'Enter symbol for substrate name: [I-isotropic, P-parabolic, Q-quartz, YZ, 128]:'
READ(*,20) SNAME
20 FORMAT(A32)

Check for invalid input:
IF(SNAME .EQ. 'I') THEN
   FSUB = 'IVEL.DAT'
   FSUB2='IDVEL.DAT'
ELSE IF(SNAME .EQ. 'P') THEN
   GOTO 30
ELSE IF(SNAME .EQ. 'Q') THEN
   GOTO 30
ELSE IF(SNAME .EQ. '128') THEN
   GOTO 30
ELSE IF(SNAME .EQ. 'YZ') THEN
   FSUB = 'YZVEL.DAT'
   FSUB2='YZDVEL.DAT'
ELSE
30 WRITE(*,*) 'Invalid entry, try again:'
   GOTO 10
ENDIF
C
WRITE(*,*) 'Is one transducer uniform? (Y or N)'
READ(*,20) INFLAG
IF((INFLAG .EQ. 'Y') .OR. (INFLAG .EQ. 'y')) THEN
   UNIF = .TRUE.
   WRITE(*,*) 'Enter filename of uniform transducer:'
   READ(*,20) FN1
ELSE
   UNIF = .FALSE.
   WRITE(*,*) 'Enter filename for time file of first transducer:'
   READ(*,20) FN1
ENDIF
WRITE(*,*) 'Enter filename for time file of second transducer:'
READ(*,20) FN2
WRITE(*,*) 'Enter center frequency of first transducer, in MHz:'
READ(*,*) F01
WRITE(*,*) 'Is center frequency of second transducer the same?
   (Y or N)'
READ(*,20) FLAG1
IF ((FLAG1 .EQ. 'Y') .OR. (FLAG1 .EQ. 'y')) THEN
   F02=F01
ELSE
   WRITE(*,*) 'Enter second transducer center frequency, in MHz:'
   READ(*,*) F02
ENDIF
WRITE(*,*) 'Enter maximum aperture for first transducer, in wavelengths at transducer center frequency:'
READ(*,*) W1MAX
WRITE(*,*) 'Enter maximum aperture for second transducer, in wavelengths at transducer center frequency:'
READ(*,*) W2MAX
WRITE(*,*) 'Enter horizontal center to center spacing of transducers, in mils:'
READ(*,*) Z0
WRITE(*,*) 'Enter vertical distance from center line of first transducer to center line of second transducer, in mils:'
READ(*,*) OFFSET
WRITE(*,*) 'Enter angle at which wave is launched, with respect to normal to transducer, in degrees:'
READ(*,*) ALPHA
WRITE(*,*) 'Enter time between fingers in first transducer (in microsec.):'
READ(*,*) DT1
WRITE(*,*) 'Is time between fingers of second transducer the same? (Y or N)'
READ(*,20) FLAG2
IF (((FLAG2 .EQ. 'Y') .OR. (FLAG2 .EQ. 'y'))) THEN
   EQUALT = .TRUE.
   DT2=DT1
ELSE
   EQUALT = .FALSE.
   WRITE(*,*) 'Enter time between fingers in second transducer (in microsec.):'
   READ(*,*) DT2
ENDIF
WRITE(*,*) 'Enter number of fingers in first transducer: '
READ(*,*) M
WRITE(*,*) 'Enter number of fingers in second transducer: '
READ(*,*) N
IF (UNIF) THEN
   SIANT=0.0
   DIREC='UP'
ELSE
   WRITE(*,*) 'Enter apodization slant for first transducer, as a fraction of aperture: '
   READ(*,*) SIANT
   WRITE(*,*) 'Is slant up or down?'
   READ(*,20) DIREC
ENDIF
WRITE(*,*) 'Enter apodization slant for second transducer, as a fraction of aperture: '
READ(*,*) SIANT2
WRITE(*,*) 'Is slant up or down?'
READ(*,20) DIREC2
WRITE(*,*) 'Enter number of angular increments for each tap integral: '
READ(*,*) ANGNUM
WRITE(*,*) 'Enter approx. number of sample points desired in time domain: '
READ(*,*) SAMP
IF((UNIF) .AND. (EQUALT)) THEN
   C Transducers have equal time separations between taps and input
   C transducer is uniform - can use simplifications
   WRITE(*,*) 'Enter number of sample points desired vertically: '
   READ(*,*) YPSAMP
   WRITE(*,*) 'Do you want to store diffracted field data? (Y or N)'
   READ(*,20) STORE
ENDIF
IF ((STORE .EQ. 'Y') .OR. (STORE .EQ. 'y')) THEN
  WRITE(*,*) 'Enter filename for field data storage:'
  READ(*,20) FSTORE
ENDIF
ELSE
  YPSAMP=1
ENDIF

C Check for invalid input
40 IF(YPSAMP .GT. 500) THEN
    WRITE(*,*) 'Number of points too large, enter new number:'
    READ(*,*) YPSAMP
    GOTO 40
ENDIF
WRITE(*,*) 'Enter name of output time domain data file:'
READ(*,20) ONAME

C TEMPORARILY IGNORE CASE WHERE DT1<>DT2
IF((FLAG2 .NE. 'Y') .AND. (FLAG2 .NE. 'y')) THEN
  WRITE(*,*) 'THIS CANNOT BE ANALYZED YET'
  GOTO 500
ENDIF

C Read in velocity data appropriate for substrate choice
OPEN(41,FILE=FSUB)
OPEN(31,FILE=FSUB2)
DO 50 J=1,181
   READ(41,60) VEL(J,1),VEL(J,2)
   READ(31,60) DVEL(J,1),DVEL(J,2)
50 CONTINUE
60 FORMAT(E16.8,1X,E16.8)
CLOSE(41)
CLOSE(31)

C Find max. value of k, the square root of Coupling Coeff.
CALL MAXX(SNAME,DVEL,KMAX)

C Calculate values
C Zero degree velocity in mils/microsec.
  CO=VEL(91,1)/25.4
C Launching angle in radians
  ALPHA=ALPHA*PI/180.
C wavelengths in mils
  WL1 = CO/F01
  WL2 = CO/F02
C transducer widths in mils
  W1 = W1MAX*WL1
  W2 = W2MAX*WL2
C number of effective taps
  NTAP1 = M-1
  NTAP2 = N-1
Calculate distance between taps and minimum transducer separation
(calculated with on-axis velocity data assumed during design)
DELX1=DT1*C0
DELX2=DT2*C0
TEMP1=(NTAP1-1)*DT1
TEMP2=(NTAP2-1)*DT2
SEP=Z0-0.5*(TEMP1+TEMP2)*C0

Calculate actual time between taps at angle of propagation
CALL FINDVEL(ALPHA,VEL,C)
DT1ACT=DELX1*25.4/C
DT2ACT=DELX2*25.4/C

Transducer time lengths Considering angle of propagation (microsec)
TL1 = (NTAP1-1)*DT1ACT
TL2 = (NTAP2-1)*DT2ACT

Approx. minimum and maximum possible interaction times (microsec)

Note: we assume minimum time of interaction is for the path perpendicular to launching transducer. This may not be true.
TMINPOS = (Z0*25.4/C) - 0.5*(TL1 + TL2)

Note: max pos. time is assumed to occur at one extreme or the other
but may not.
TOPY = OFFSET + 0.5*(W2+WL)
BOTY = OFFSET - 0.5*(W2+WL)
MAXPOSX=Z0+0.5*(TL1+TL2)*C/25.4
MAXANG=DATAN2(TOPY,MAXPOSX)
MINANG=DATAN2(BOTY,MAXPOSX)
RTOP=MAXPOSX/DCOS(MAXANG)
RBT=MAXPOSX/DCOS(MINANG)
CALL FINDVEL(ALPHA+MAXANG,VEL,CMAX)
TOPT=RTOP*25.4/CMAX
CALL FINDVEL(ALPHA+MINANG,VEL,CMIN)
BOTT=RBT*25.4/CMIN
IF(TOPT .GE. BOTT) THEN
  TMAXPOS=TOPT
ELSE
  TMAXPOS=BOTT
ENDIF

Give option of varying time samples somewhat, while making sure they remain a multiple divisor of DT2 and DT1
WRITE(*,*) 'Minimum time of interaction is approx. (microsec):',
  * TMINPOS
WRITE(*,*) 'Maximum time of direct interaction is approx.:',
  * TMAXPOS
WRITE(*,*) 'Enter minimum time of interest, in microsec:'
READ(*,*) MNINT
WRITE(*,*) 'Enter maximum time of interest, in microsec:'
READ(*,*) MAXINT
J=(MAXINT-MININT)/DT1ACT+1.
AJ=J
MAXINT=(DT1ACT*J)+MININT
DF=1.0/(MAXINT-MININT)
WRITE(*,*) 'Max. time is ', MAXINT
WRITE(*,*) 'Resolution in frequency domain is approx. (MHz):'
WRITE(*,*) DF
WRITE(*,*) 'Is this resolution high enough? (Y or N)'
READ(*,20) FLAG3
IF((FLAG3 .EQ. 'Y') .OR. (FLAG3 .EQ. 'y')) THEN
   GOTO 29
ELSE
   WRITE(*,*) 'Enter larger max. time:'
   READ(*,*) MAXINT
   GOTO 70
29   CONTINUE
   ENDIF
C
   The number of time samples will be an integer number times
   J, yielding an integer number of time samples in the interval
   DT1ACT. The entire time range is J*DT1ACT long.
   AJ=J
   I=(SAMP/AJ)+1
   NSAMPT=I*J+1
C
   Calculate step size for time
   AI=I
   DELTAT = DT1ACT/AI
C
   Initialize array of time samples
   IF (UNIF .AND. EQUIV) THEN
      DO 81 I=1,NSAMPT
         TIMEREC(I)=0.0
      DO 80 K=1,YPsamp
         DO 79 J=1,NTAP2
            TAPFIELD(J,I,K,1)=0.0
            TAPFIELD(J,I,K,2)=0.0
            FIELD(J,I,K,1)=0.0
            FIELD(J,I,K,2)=0.0
      CONTINUE
      UNIOUT(I,K,1)=0.0
      UNIOUT(I,K,2)=0.0
      CONTINUE
      CONTINUE
      ENDIF
50   CONTINUE
ELSE
      DO 82 I=1,NSAMPT
         TIMEREC(I)=0.0
      AI=I
   ELSE
      DO 82 I=1,NSAMPT
         TIMEREC(I)=0.0
      AI=I
   ENDIF
82
CONTINUE

ENDIF

C Calculate position and lengths of effective taps
CALL FILEPREP(FN1, FN2, M, N, SLANT, DIREC, SLANT2, DIREC2, W1, W2)

C Calculate field response for each possible tap separation and
tap length combination.

C IF (UNI. . AND. EQUALT) THEN
C Input transducer is uniform, equal time spacings, so we can
C use some simplifications in calculations.
L=W1
LOUT=W2
YPSTP=LOUT/YPsamp
BOT=0.5*(W1-W2)+OFFSET
DO 207 IT=1, NSAMP
   DO 207 KY=1, YPSAMP
      AK=KY
      UNIOUT(IT, KY, 1)=BOT+(AK-1.0)*YPSTP
   207 CONTINUE
208 CONTINUE

DO 85 SEPNUM=1, (NTAP1+NTAP2-1)
   DN=SEP+(SEPNUM-1)*DELXI
   CALL UNITAPRESP(VEL, DVEL, L, LOUT, BOT, DN, ALPHA,
                   MININT, DELTAT, ANGNUM, NSAMPT, YPSAMP, KMIX,
                   SNAME, YPSTP)
   *DO 84 ICNT=1, NSAMPT
   DO 83 JCNT=1, YPSAMP
      TAPFIELD(SEPNUM, ICNT, JCNT, 1)=UNIOUT(ICNT, JCNT, 1)
      TAPFIELD(SEPNUM, ICNT, JCNT, 2)=UNIOUT(ICNT, JCNT, 2)
   83 CONTINUE
84 CONTINUE
85 CONTINUE

C Next, we need to put this diffracted tap field together
to give diffracted field of entire uniform transducer
at positions which correspond to output transducer tap
locations and including polarities.
C POS1 is 1 at the right end of first transducer.
C POS2 is 1 at the left end of output transducer.
C T1DAT and T2DAT both store data from left to right.
DO 89 POS2=1, NTAP2
   DO 88 POS1=1, NTAP1
      IF(((T1DAT(NTAP1+1-POS1,1) .GE. 0.0) .AND. (T2DAT
(POS2,1) .GE. 0.0)) .OR. ((T1DAT(NTAP1+1-POS1,1)
* .LT. 0.0) .AND. (T2DAT(POS2,1) .LT. 0.0)) THEN
SIGN=+1.0
ELSE
SIGN=-1.0
ENDIF

SEPCNT=POS2+POS1-1
DO 87 Kl=1,NSAMPT
   DO 86 K2=1,YSAMP
      FIELD(POS2,Kl,K2,1)=TAPFIELD(SEPCNT,Kl,K2,1)
      FIELD(POS2,Kl,K2,2)=FIELD(POS2,Kl,K2,2)+
       SIGN*TAPFIELD(SEPCNT,Kl,K2,2)
   CONTINUE
86 CONTINUE
87 CONTINUE
88 CONTINUE
89 CONTINUE

Now that diffracted field has been calculated, we can output it

file for later use.
IF ((STORE .EQ. 'Y') .OR. (STORE .EQ. 'y')) THEN
   OPEN(53,FILE=STORE,ANYUNIT=INT)
   DO 306 POS2=1,NTAP2
      APOS=POS2
      DO 305 KY=1,YSAMP
         AK=KY
         DO 304 IT=1,NSAMPT
            AI=IT
            WRITE(53,307)
               (APOS-1)*DEIX1,FIELD(POS2,IT,KY,1),
               MINIT+(AI-1)*DEIXAT,FIELD(POS2,IT,KY,2)
   CONTINUE
304 CONTINUE
305 CONTINUE
306 CONTINUE
307 FORMAT(E16.8,1X,E16.8,1X,E16.8,1X,E16.8,1X,E16.8)
   CLOSE(53)
ENDIF

Next, we need to determine where the output taps are and account
for this to determine overall response of output transducer.

DO 200 JSECOND=1,NTAP2
   TAPBOT=T2DAT(JSECOND,2)+BOT
   TAPTOP=ABS(T2DAT(JSECOND,1))+TAPBOT
   TWIDTH=TAPTOP-TAPBOT
C
   Find out which spatial bins this corresponds to
   LOKTEMP=2
C
   Find counter for bottom of bin just above bottom of tap
   201 IF (FIELD(JSECOND,1,LOKTEMP,1) .LT. TAPBOT) THEN
      LOKTEMP=LOKTEMP+1
GOTO 201
ENDIF
C
Find counter for bottom of bin just below top of tap
JUMP=((TAPTOP-TAPBOT)/YPSTP)-2
HIKTEMP=LOKTEMP+JUMP
IF(HIKTEMP.EQ.0) THEN
  HIKTEMP=1
ENDIF
C
202
IF(HIKTEMP.GE. YPSAMP) THEN
  HIKTEMP=YPSAMP
ELSE IF(TAPTOP.GT. FIELD(JSECOND,1,HIKTEMP,1)) THEN
  HIKTEMP=HIKTEMP+1
  GOTO 202
ELSE
  HIKTEMP=HIKTEMP-1
ENDIF
C
C
Check to see if tap is all in one bin
IF(HIKTEMP.LT. LOKTEMP) THEN
C
  Tap is all in one bin
  DO 209 J=1,NSAMPT
       TIMEREC(J)=TIMEREC(J)+(FIELD(JSECOND,J,HIKTEMP,2)
  ENDIF
C
ELSE IF(HIKTEMP.EQ. LOKTEMP) THEN
C
  Tap is short & split between two bins
  DO 150 JTIME=1,NSAMPT
       TIMEREC(JTIME)=TIMEREC(JTIME)+((FIELD(JSECOND,1,
       LOKTEMP,1)-TAPBOT)/YPSTP)*FIELD(JSECOND,JTIME,
       LOKTEMP-1,2)+((TAPTOP-
       FIELD(JSECOND,1,LOKTEMP,1))
       /YPSTP)*FIELD(JSECOND,JTIME,LOKTEMP,2)
  ENDIF
C
ELSE
  LOK=LOKTEMP
  HIK=HIKTEMP-1
C
  LOK and HIK set limits on where tap is located, so
  add
C
  responses for these locations at all times.
  DO 204 JTIME=1,NSAMPT
     DO 203 KCNT=LOK,HIK
         TIMEREC(JTIME)=TIMEREC(JTIME)+((FIELD(JSECOND,
         JTIME,KCNT,2))
  ENDIF
C
203
C
  Need to add in responses for 'ends' sticking
  out beyond
C
  LOK and HIK, proportional to length of 'end'.
  DTAPBOT=FIELD(JSECOND,1,LOK,1)-TAPBOT
  DTAPTOP=TAPTOP-FIELD(JSECOND,1,HIK+1,1)
**TIMERC** (JTIME) = TIMERC(JTIME) + ((DTAPIOP/YPSTP) *  
FIELD(JSECOND, JTIME, HIK+1,2)) +  
((DTAPB0T/YPSTP) * FIELD(JSECOND, JTIME,  
LOK-1,2))

**CONTINUE**

**ENDIF**

**CONTINUE**

C Write to output file

OPEN(51, FILE=ONAME, ANYUNIT=I)

DO 205 J=1, NSAMPT  
    AJ=J
    WRITE(51,210) MININT+(AJ-1)*DELTAT, TIMERC(J)

**CONTINUE**

**210 FORMAT(E16.8,1X,E16.8)**

CLOSE(51)

C

ELSE IF (.NOT.UNIF) THEN

Must use brute force method for calculating response.

DO 110 JFIRST=NTAPI1,1,-1  
    DN=SEP+(NTAPI1-JFIRST)*DELX1  
    L=ABS(T1DAT(JFIRST,1))

DO 100 JSECOND=1, NTAPI2  
    BOT=T2DAT(JSECOND,2) -T1DAT(JFIRST,2) +.5*(W1  
    -W2) +OFFSET  
    LOUT=ABS(T2DAT(JSECOND,1))  
    DTAPS=DN + (JSECOND-1)*DELX2  
    CALL TAPRESP(VEL, DVEL, L, LOUT  
    , BOT, DTAPS, OFFSET, ALPHA,  
    , ANGNUM, NSAMPT, RMAX, SNAME)

IF(((T1DAT(JFIRST,1) .GE. 0.0) .AND.  
(T2DAT(JSECOND,1)  
* .GE. 0.0)) .OR. ((T1DAT(JFIRST,1) .LT. 0.0) .AND.  
(T2DAT(JSECOND,1) .LT. 0.0)) ) THEN

DO 98 I=1, NSAMPT  
    TIMERC(I)=TIMERC(I)+OUT(I,3)

**CONTINUE**

ELSE

DO 99 I=1, NSAMPT  
    TIMERC(I)=TIMERC(I)-OUT(I,3)

**CONTINUE**

**ENDIF**

**CONTINUE**

C OUTPUT TO DATA FILE

OPEN(51, FILE=ONAME)

DO 120 J=1, NSAMPT  
    WRITE(51,130) OUT(J,1), TIMERC(J)

**CONTINUE**
SUBROUTINE TRANSRESPF

This routine calculates the transducer-to-transducer
impulse response in the frequency domain on an isotropic
or anisotropic substrate, using a double
integral reduction technique.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
INTEGER*4 M,N,YPSAMP,NSAMPF,ANGNUM,NTAPI,NTAP2
INTEGER*4 SEPNUM,HIK,LOK,POS1,POS2,FNUM,YPNUM,TAPNUM
INTEGER*4 SEPONT,NUMINCRF,LOKTEMP,HIKTEMP,NINCR
PARAMETER (FNUM=2048,YPNUM=10,TAPNUM=10)
CHARACTER*32 ONAME,FSUB,SNAMF,DIREC,DIREC2,FN1,FN2
CHARACTER*32 FSUB2,FLAG1,FLAG2,FLAG3
CHARACTER*32 INFLAG,STORE,FSTORE
REAL*8 VEL(1:181,1:2),DVEL(1:181,1:2),T1DAT(1:TAPNUM,1:2)
REAL*8 T2DAT(1:TAPNUM,1:2),OUT(1:FNUM,1:3)
REAL*8 FREC(1:FNUM,1:2),UNIOUT(1:FNUM,1:YPNUM,1:3)
REAL*8 TAPFIELD(1:TAPNUM,1:FNUM,1:YPNUM,1:3)
REAL*8 FIELD(1:TAPNUM,1:FNUM,1:YPNUM,1:3)
REAL*8 KMAX
COMMON /B1/UNIOUT
COMMON /B2/OUT
COMMON /B3/T1DAT,T2DAT
COMMON /B4/FREC,FIELD,TAPFIELD
LOGICAL*4 UNIF,EQUALT
PI = 4.0*DATAN(1.0D+00)

Enter Necessary Values
WRITE(*,*) 'Enter symbol for substrate name: [I-isotropic,
P-parabolic, Q-quartz, YZ, 128]:'
READ(*,20) SNAME

Check for invalid input:
IF(SNAME .EQ. 'I') THEN
   FSUB = 'IVEL.DAT'
END IF
FSUB2='IDVEL.DAT'
ELSE IF(SNAME .EQ. 'P') THEN
  GOTO 30
ELSE IF(SNAME .EQ. 'Q') THEN
  GOTO 30
ELSE IF(SNAME .EQ. '128') THEN
  GOTO 30
ELSE IF(SNAME .EQ. 'YZ') THEN
  FSUB = 'YZVEL.DAT'
  FSUB2='YZDVEL.DAT'
ELSE
  WRITE(*,*) 'Invalid entry, try again:'
  GOTO 10
ENDIF

WRITE(*,*) 'Is one transducer uniform? (Y or N)'
READ(*,20) INFLAG
IF((INFLAG .EQ. 'Y') .OR. (INFLAG .EQ. 'y')) THEN
  UNIF = .TRUE.
  WRITE(*,*) 'Enter filename of uniform transducer:'
  READ(*,20) FN1
ELSE
  UNIF = .FALSE.
  WRITE(*,*) 'Enter filename for time file of first transducer:'
  READ(*,20) FN1
ENDIF
WRITE(*,*) 'Enter filename for time file of second transducer:'
READ(*,20) FN2
WRITE(*,*) 'Enter center frequency of first transducer, in MHz:'
READ(*,*) F01
WRITE(*,*) 'Is center frequency of second transducer the same? (Y or N)'
READ(*,20) FLAG1
IF ((FLAG1 .EQ. 'Y') .OR. (FLAG1 .EQ. 'y')) THEN
  F02=F01
ELSE
  WRITE(*,*) 'Enter second transducer center frequency, in MHz:'
  READ(*,*) F02
ENDIF
WRITE(*,*) 'Enter maximum aperture for first transducer, in wavelengths at transducer center frequency:'
READ(*,*) W1MAX
WRITE(*,*) 'Enter maximum aperture for second transducer, in wavelengths at transducer center frequency:'
READ(*,*) W2MAX
WRITE(*,*) 'Enter horizontal center to center spacing of
* transducers, in mils:
READ(*,*) Z0
WRITE(*,*) 'Enter vertical distance from center line of
first transducer to center line of second transducer, in
mils:
READ(*,*) OFFSET
WRITE(*,*) 'Enter angle at which wave is launched, with
* respect to normal to transducer, in degrees:
READ(*,*) ALPHA
WRITE(*,*) 'Enter time between fingers in first transducer
* (in microsec.):
READ(*,*) DT1
WRITE(*,*) 'Is time between fingers of second transducer the
* same? (Y or N)
READ(*,20) FLAG2
IF ((FLAG2 .EQ. 'Y') .OR. (FLAG2 .EQ. 'y')) THEN
   EQUALT = .TRUE.
   DT2=DT1
ELSE
   EQUALT = .FALSE.
   WRITE(*,*) 'Enter time between fingers in second
* transducer
* (in microsec.):
READ(*,*) DT2
ENDIF
WRITE(*,*) 'Enter number of fingers in first transducer:
READ(*,*) M
WRITE(*,*) 'Enter number of fingers in second transducer:
READ(*,*) N
IF (UNIF) THEN
   SLANT=0.0
   DIREC='UP'
ELSE
   WRITE(*,*) 'Enter apodization slant for first
* transducer,
as a fraction of aperture:
READ(*,*) SLANT
WRITE(*,*) 'Is slant up or down?
READ(*,20) DIREC
ENDIF
WRITE(*,*) 'Enter apodization slant for second transducer,
as a
* fraction of aperture:
READ(*,*) SLANT2
WRITE(*,*) 'Is slant up or down?
READ(*,20) DIREC2
WRITE(*,*) 'Enter number of angular increments for each tap
* integral:
READ(*,*) ANGNUM
WRITE(*,*) 'ANGNUM=',ANGNUM
C WRITE(*,*) 'ANGNUM=',ANGNUM
IF((UNIF) .AND. (EQUALT)) THEN
Transducers have equal time separations between taps and input. A transducer is uniform - can use simplifications.

WRITE(*,*) 'Enter number of sample points desired vertically:
READ(*,*) YPSAMP
WRITE(*,*) 'Do you want to store diffracted field data?
(Y or N)
READ(*,20) STORE
IF ((STORE .EQ. 'Y') .OR. (STORE .EQ. 'y')) THEN
  WRITE(*,*) 'Enter filename for field data storage:'
  READ(*,20) FSTORE
ENDIF
ELSE
  YPSAMP=1
ENDIF

Check for invalid input.

IF(YPSAMP .GT. 500) THEN
  WRITE(*,*) 'Number of points too large, enter new number:
  READ(*,*) YPSAMP
  GOTO 40
ENDIF
WRITE(*,*) 'Enter frequency sampling rate, in samples per MHz:
READ(*,*) FSAMP
WRITE(*,*) 'Enter minimum frequency of interest, in MHz:
READ(*,*) FMIN
WRITE(*,*) 'Enter maximum frequency of interest, in MHz:
READ(*,*) FMAX
WRITE(*,*) 'Enter name of output frequency domain data file:
READ(*,20) ONAME

TEMPORARILY IGNORE CASE WHERE DT1<>DT2
IF((FLAG2 .NE. 'Y') .AND. (FLAG2 .NE. 'y')) THEN
  WRITE(*,*) 'THIS CANNOT BE ANALYZED YET'
  GOTO 500
ENDIF

Read in velocity data appropriate for substrate choice.
OPEN(41,FILE=FSUB)
OPEN(31,FILE=FSUB2)
DO 50 J=1,181
  READ(41,60) VEL(J,1),VEL(J,2)
  WRITE(*,*) VEL(J,1),VEL(J,2)
  READ(31,60) DVEL(J,1),DVEL(J,2)
50 CONTINUE
FORMAT(E16.8,1X,E16.8)
CLOSE(41)
CLOSE(31)
Find max. value of k, the square root of Coupling Coeff.
CALL MAXK(SNAME,DVEL,KMAX)

Calculate values
Zero degree velocity in mils/microsec.
C0=VEL(91,1)/25.4
WRITE(*,*) 'C0=',C0

Launching angle in radians
ALPHA=ALPHA*PI/180.

wavelengths in mils
WL1 = C0/F01
WL2 = C0/F02
WRITE(*,*) 'WL1=',WL1,WL2

transducer widths in mils
W1 = W1MAX*WL1
W2 = W2MAX*WL2
WRITE(*,*) 'W1=',W1,W2

number of effective taps
NTAP1 = M-1
NTAP2 = N-1

Calculate distance between taps and minimum transducer separation
(calculate with on-axis velocity data assumed during design)
DELX1=DT1*C0
DELX2=DT2*C0
TEMP1=(NTAP1-1)*DT1
TEMP2=(NTAP2-1)*DT2
SEP=Z0-0.5*(TEMP1+TEMP2)*C0

Find the minimum number of frequency increments to get from FMIN to FMAX
NINCR = 0
F = FMIN
DELTAF=1/FSAMP
103 IF (F .LT. FMAX) THEN
   F = F + DELTAF
   NINCR = NINCR+1
   GOTO 103
ENDIF

At least NINCR frequency increments are required
NSAMPF = NINCR +1

Initialize array of time samples
IF (UNIF .AND. EQUALT) THEN
   DO 81 I=1,NSAMPF
      FREC(I,1)=0.0
      FREC(I,2)=0.0
   DO 80 K=1,YPsamp
   81 FREC(I,1)=
   80 FREC(I,2)=
DO 79 J=1,NTAP2
   TAPFIELD(J,I,K,1)=0.0
   TAPFIELD(J,I,K,2)=0.0
   TAPFIELD(J,I,K,3)=0.0
   FIELD(J,I,K,1)=0.0
   FIELD(J,I,K,2)=0.0
   FIELD(J,I,K,3)=0.0
    CONTINUE
   UNICOUT(I,K,1)=0.0
   UNICOUT(I,K,2)=0.0
   UNICOUT(I,K,3)=0.0
79    CONTINUE
80   CONTINUE
81   CONTINUE
ELSE
   DO 82 I=1,NSAMPF
      FREC(I,1)=0.0
      FREC(I,2)=0.0
      AI=I
      OUT(I,1)=FMIN + (AI-1.0)*DELTAF
      OUT(I,2)=0.0
      OUT(I,3)=0.0
82    CONTINUE
ENDIF

C Calculate position and lengths of effective taps
CALL FILEPREP(FN1,FN2,M,N,SLANT,DIREC,SLANT2,DIREC2,W1,W2)

C Calculate field response for each possible tap separation and
tap length combination.

C IF (UNIF .AND. EQUALT) THEN
   Input transducer is uniform, equal time spacings, so we can
   use some simplifications in calculations.
   L=W1
C WRITE(*,*) 'L=',L
   LOUT=W2
C WRITE(*,*) 'LO=',LOUT
   YPSTP=LOUT/YPSAMP
   BOT=0.5*(W1-W2)+OFFSET
   DO 208 IT=1,NSAMPF
      DO 207 KY=1,YPSAMP
         AK=KY
         UNICOUT(IT,KY,1)=BOT+(AK-1.0)*YPSTP
      207   CONTINUE
208    CONTINUE
   DO 85 SEPNUM=1,(NTAP1+NTAP2-1)
      DN=SEP+(SEPNUM-1)*DELX1
CALL UNIFTAPRESP(VEL,DVEL,L,LOUT,DN,BOT,ALPHA, 
ANGNUM,FMIN,DELTAF,NSAMPF,YPSAMP,KMAX,SNAME,YPSTP) 
DO 84 ICNT=1,NSAMPF 
DO 83 JCNT=1,YPSAMP 
  TAPFIELD(SEPNUM,ICNT,JCNT,1)=UNITOUT(ICNT,JCNT,1) 
  TAPFIELD(SEPNUM,ICNT,JCNT,2)=UNITOUT(ICNT,JCNT,2) 
  TAPFIELD(SEPNUM,ICNT,JCNT,3)=UNITOUT(ICNT,JCNT,3) 
83 CONTINUE 
84 CONTINUE 
85 CONTINUE 
C Next, we need to put this diffracted tap field together 
C to give diffracted field of entire uniform transducer 
C at positions which correspond to output transducer tap 
C locations and including polarities. 
C POS1 is 1 at the right end of first transducer. 
C POS2 is 1 at the left end of output transducer. 
C T1DAT and T2DAT both store data from left to right. 
DO 89 POS2=1,NTAP2 
  DO 88 POS1=1,NTAP1 
    IF(((T1DAT(NTAP1+1-POS1,1) .GE. 0.0) .AND. (T2DAT 
       (POS2,1) .GE. 0.0)) .OR. ((T1DAT(NTAP1+1-POS1,1) 
       .LT. 0.0) .AND. (T2DAT(POS2,1) .LT. 0.0))) THEN 
      SIGN=+1.0 
    ELSE 
      SIGN=-1.0 
    ENDIF 
    SEPNT=POS2+POS1-1 
  DO 87 K1=1,NSAMPF 
    DO 86 K2=1,YPSAMP 
      FIELD(POS2,K1,K2,1)=TAPFIELD(POS2,K1,K2,1) 
      FIELD(POS2,K1,K2,2)=FIELD(POS2,K1,K2,2) 
     * SIGN*TAPFIELD(SEPNT,K1,K2,1) 
      FIELD(POS2,K1,K2,3)=FIELD(POS2,K1,K2,3) 
     * SIGN*TAPFIELD(SEPNT,K1,K2,3) 
86 CONTINUE 
87 CONTINUE 
88 CONTINUE 
89 CONTINUE 
C Now that diffracted field has been calculated, we can 
C output it for later use. 
IF ((STORE .EQ. 'Y') .OR. (STORE .EQ. 'y')) THEN 
  OPEN(52,FILE=FSTORE,ANYUNIT=INT) 
  DO 306 POS2=1,NTAP2 
    APOS=POS2 
    DO 305 KY=1,YPSAMP 
      AK=KY 
      DO 304 IT=1,NSAMPF 
        AI=IT 
        WRITE(52,307) (APOS-1)*DELX1,FIELD(POS2,
CALL UNIFTAPRESP(VEL, DVEL, L, LOUT, DN, BOT, ALPHA, 
   ANGNUM, FMIN, DELTAF, NSAMPF, YPSAMP, KMAX, SNAME, YPSTP)
DO 84 ICNT=1, NSAMPF
   DO 83 JCNT=1, YPSAMP
      TAPFIELD(SEPNUM, ICNT, JCNT, 1)=UNIOUT(ICNT, JCNT, 1)
      TAPFIELD(SEPNUM, ICNT, JCNT, 2)=UNIOUT(ICNT, JCNT, 2)
      TAPFIELD(SEPNUM, ICNT, JCNT, 3)=UNIOUT(ICNT, JCNT, 3)
   CONTINUE
83 CONTINUE
84 CONTINUE
85 CONTINUE
C Next, we need to put this diffracted tap field together
to give diffracted field of entire uniform transducer
at positions which correspond to output transducer tap
locations and including polarities.
C POS1 is 1 at the right end of first transducer.
C POS2 is 1 at the left end of output transducer.
C T1DAT and T2DAT both store data from left to right.
DO 89 POS2=1, NTAP2
   DO 88 POS1=1, NTAP1
      IF(((T1DAT(NTPAPl+1-POS1,1) .GE. 0.0) .AND. (T2DAT 
         (POS2,1) .GE. 0.0)) .OR. ((T1DAT(NTPAPl+1-POS1,1)
         .LT. 0.0) .AND. (T2DAT(POS2,1) .LT. 0.0))) THEN
         SIGN=+1.0
      ELSE
         SIGN=-1.0
      ENDIF
   SEPOlNT=POS2+POS1-1
   DO 87 K1=1, NSAMPF
      DO 86 K2=1, YPSAMP
         FIELD(POS2, K1, K2, 1)=TAPFIELD(POS2, K1, K2, 1)
         FIELD(POS2, K1, K2, 2)=FIELD(POS2, K1, K2, 2)+
            SIGN*TAPFIELD(SEPCT, K1, K2, 2)
         FIELD(POS2, K1, K2, 3)=FIELD(POS2, K1, K2, 3)+
            SIGN*TAPFIELD(SEPCT, K1, K2, 3)
      CONTINUE
   CONTINUE
86 CONTINUE
87 CONTINUE
88 CONTINUE
89 CONTINUE
C Now that diffracted field has been calculated, we can
output it for later use.
IF ((STORE .EQ. 'Y') .OR. (STORE .EQ. 'y')) THEN
   OPEN(52, FILE=FSSTORE, ANYUNIT=INT)
   DO 306 POS2=1, NTAP2
      APOS=POS2
      DO 305 KV=1, YPSAMP
         AK=KV
         DO 304 IT=1, NSAMPF
            AI=IT
            WRITE(52, 307) (APOS-1)*DX1, FIELD(POS2,
IT,KY,1),FMIN+(AI-1)*DELTAF,FIELD(POS2,
IT,KY,2),FIELD(POS2,IT,KY,3)
CONTINUE
CONTINUE
CONTINUE
FORMAT(E16.8,1X,E16.8,1X,E16.8,1X,E16.8,1X,E16.8,1X,E16.8)
CLOSE(52)
ENDIF

Next, we need to determine where the output taps are and
account for this to determine overall response of output
transducer.

DO 200 JSECOND=1,NTAP2
TAPBOT=T2DAT(JSECOND,2)+BOT
TAPTOP=ABS(T2DAT(JSECOND,1))+TAPBOT
TWIDTH=TAPTOP-TAPBOT

Find out which spatial bins this corresponds to
LOKTEMP=2

Find counter for bottom of bin just above bottom of tap
IF(FIELD(JSECOND,1,LOKTEMP,1) .LT. TAPBOT) THEN
  LOKTEMP=LOKTEMP+1
  GOTO 201
ENDIF

Find counter for bottom of bin just below top of tap
JUMP=(TAPTOP-TAPBOT)/YPSTP-2
HIKTEMP=LOKTEMP+JUMP
IF(HIKTEMP .EQ. 0) THEN
  HIKTEMP=1
ENDIF

IF(HIKTEMP .GE. YPSAMP) THEN
  HIKTEMP=YPSAMP
ELSE IF(TAPTOP .GT. FIELD(JSECOND,1,HIKTEMP,1)) THEN
  HIKTEMP=HIKTEMP+1
  GOTO 202
ELSE
  HIKTEMP=HIKTEMP-1
ENDIF

Check to see if tap is all in one bin
IF(HIKTEMP .LT. LOKTEMP) THEN
  Tap is all in one bin
  DO 209 J=1,NSAMPF
    FREC(J,1)=FREC(J,1)+(TWIDTH/YPSTP)*
    FIELD(JSECOND,J,HIKTEMP,2)
    FREC(J,2)=FREC(J,2)+(TWIDTH/YPSTP)*
    FIELD(JSECOND,J,HIKTEMP,3)
  CONTINUE
  ELSE IF(HIKTEMP .EQ. LOKTEMP) THEN
  Tap is short & split between two bins
  DO 150 JTIME=1,NSAMPF
FREC(JTIME, 1) = FREC(JTIME, 1) + ((FIELD(JSECOND, 1, LOKTEMP, 1) - TAPBOT) / YPSTP) * FIELD(JSECOND, JTIME, LOKTEMP - 1, 2) + ((TAPTOP - FIELD(JSECOND, 1, LOKTEMP, 1)) / YPSTP) * FIELD(JSECOND, JTIME, LOKTEMP, 2)

C

FREC(JTIME, 2) = FREC(JTIME, 2) + ((FIELD(JSECOND, 1, LOKTEMP, 1) - TAPBOT) / YPSTP) * FIELD(JSECOND, JTIME, LOKTEMP - 1, 3) + ((TAPTOP - FIELD(JSECOND, 1, LOKTEMP, 1)) / YPSTP) * FIELD(JSECOND, JTIME, LOKTEMP, 3)

150 CONTINUE

ELSE

LOK = LOKTEMP
HIK = HIKTEMP - 1

C

LOK and HIK set limits on where tap is located, so add responses for these locations at all times.

DO 204 JTIME = 1, NSAMPF
    DO 203 KCNT = LOK, HIK
        * 
        
        FREC(JTIME, 1) = FREC(JTIME, 1) + (FIELD(JSECOND, JTIME, KCNT, 2))
        FREC(JTIME, 2) = FREC(JTIME, 2) + (FIELD(JSECOND, JTIME, KCNT, 3))
    203 CONTINUE

C

Need to add in responses for 'ends' sticking out beyond.

C

LOK and HIK, proportional to length of 'end'.

DTAPBOT = FIELD(JSECOND, 1, LOK, 1) - TAPBOT
DTAPTOP = TAPTOP - FIELD(JSECOND, 1, HIK + 1, 1)
FREC(JTIME, 1) = FREC(JTIME, 1) + ((DTAPTOP / YPSTP) * FIELD(JSECOND, JTIME, HIK + 1, 2)) + ((DTAPBOT / YPSTP) * FIELD(JSECOND, JTIME, LOK - 1, 2))

C

FREC(JTIME, 2) = FREC(JTIME, 2) + ((DTAPTOP / YPSTP) * FIELD(JSECOND, JTIME, HIK + 1, 3)) + ((DTAPBOT / YPSTP) * FIELD(JSECOND, JTIME, LOK - 1, 3))

204 CONTINUE

ENDIF

C

Write to output file

OPEN(51, FILE = ONAME)
write(*, *) ' file', oname, ' is open'
DO 205 J = 1, NSAMPF
    AJ = J
    WRITE(51, 210) FMIN + (AJ - 1) * DELTAF, DSQT( FREC(J, 1) ** 2 + FREC(J, 2) ** 2), DATAN2(FREC(J, 2), FREC(J, 1))
write(*, *) ' writing j = ', j
ELSE IF (.NOT.UNIF) THEN
    Must use brute force method for calculating response.
    DO 110 JFIRST=NTAP1,1,-1
    WRITE(*,*),'SEP=',SEP
    DN=SEP+(NTAP1-JFIRST)*DELX1
    WRITE(*,*),DN
    DO 100 JSECOND=1,NTAP2
        BOT=T2DAT(JSECOND,2)-T1DAT(JFIRST,2)+.5*(W1-W2)+OFFSET
        L=ABS(T1DAT(JFIRST,1))
        LOUT=ABS(T2DAT(JSECOND,1))
        DTAPS=DN+(JSECOND-1)*DELX2
        WRITE(*,*),DTAPS
    CALL FTAPresp(VEL,DVEL,L,OUT,BOT,DTAPS,ALPHA,
*        PMIN,DELTAF,ANGNUM,NSAMPF,KMAX,SNAME)
        IF(((T1DAT(JFIRST,1) .GE. 0.0) .AND.
*           (T2DAT(JSECOND,1) .GE. 0.0)) .OR.
*           ((T1DAT(JFIRST,1) .LT. 0.0) .AND.
*           (T2DAT(JSECOND,1) .LT. 0.0))) THEN
            DO 98 I=1,NSAMPF
                FREC(I,1)=FREC(I,1)+OUT(I,2)
                FREC(I,2)=FREC(I,2)+OUT(I,3)
            WRITE(*,*),'F=',FREC(I,1),FREC(I,2)
        CONTINUE
    ELSE
            DO 99 I=1,NSAMPF
                FREC(I,1)=FREC(I,1)-OUT(I,2)
                FREC(I,2)=FREC(I,2)-OUT(I,3)
            WRITE(*,*),'F=',FREC(I,1),FREC(I,2)
        CONTINUE
    ENDIF
    CONTINUE
100 CONTINUE
110 CONTINUE
C OUTPUT TO DATA FILE
OPEN(51,FILE=ONAME)
DO 120 J=1,NSAMPF
    WRITE(51,130) OUT(J,1),DSQRT(FREC(J,1)**2+FREC(J,2)**2),
*           DATAN2(FREC(J,2),FREC(J,1))
120 CONTINUE
130 FORMAT(E16.8,1X,E16.8,1X,E16.8)
CLOSE(51)
ENDIF
500 CONTINUE
C END
SUBROUTINE FINDVEL(THETA, VEL, CINR)

This subroutine calculates the velocity at a given angle by linear interpolation between data points in the array VEL (in m/sec).

IMPLICIT REAL*8(A-H, L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 VEL(1:181,1:2)
PI = 4.0*DATAN(1.0d00)

K = 1

Determine which two angles in the array theta lies between
DO 100 I = 1, 180
   AI = I
   TEMPL = -(PI/2.0) + (AI-1.)*PI/180.
   TEMPH = -(PI/2.0) + (AI)*PI/180.
   IF ((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN
      Interpolate between values at K and K+1 in array
      CLO = VEL(K,1)
      CHI = VEL(K+1,1)
      ANGDIF = THETA - ((-PI/2.0) + (AI-1.)*PI/180.)
      CINR = CLO + (CHI-CLO)*180.*(ANGDIF)/PI
   ELSE
      K = K+1
   ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE MAXK(SNAME, DVEL, KMAX)

This subroutine searches the delta(v)/v array DVEL and returns the maximum value of k, the square root of the Coupling COefficient.

IMPLICIT REAL*8(A-H, L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KMAX

See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
   KMAX = 1.0
ELSE
Substrate is anisotropic, search for maximum
TEMPMAX=DVEL(1,2)
DO 100 I=1,180
IF(DVEL(I+1,2) .GE. TEMPMAX) THEN
  TEMPMAX=DVEL(I+1,2)
ENDIF
100
CONTINUE
KMAX=DSQRT(2.0*TEMPMAX)
ENDIF
RETURN
END

SUBROUTINE TAPRESP(VEL,DVEL,L,LOUT,BOT,D,OFFSET,ALPHA,
  ANGNUM,SAMP,KMAX,SNAME)

This routine calculates the tap-to-tap impulse response in the time domain on an isotropic or anisotropic substrate, using a double integral reduction technique and incorporating the angular dependence of the electromechanical coupling coefficient.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
INTEGER*4 ANGNUM,SAMP,TNUM
PARAMETER(TNUM=1400)
CHARACTER*32 ONAME,SNAME
REAL*8 VEL(1:181,1:2),DVEL(1:181,1:2),OUT(1:TNUM,1:3)
REAL*8 KTHETA,KL,K2,K3,K4,VOID(1:TNUM),KMAX
COMMON /B2/OUT
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D+00)

Calculate position of top of tap (bottom given)
TOP = BOT + LOUT
THETAMIN = DATAN2((BOT-L),D)
THETAMAX = DATAN2(TOP,D)
BETAMIN = THETAMIN + ALPHA
BETAMAX = THETAMAX + ALPHA

Calculate step sizes for y and yp
YINCR = -(L-BOT+TOP)/ANGNUM

Initialize output array
DO 90 J=1,TNUM
  OUT(J,2)=0.0
  OUT(J,3)=0.0
90
CONTINUE
Initialize values used in integral loop
TERM = 0.0
WIDTH = 0.0
\text{O_WIDTH} = 0.0
\text{CALL FINDVEL(BETAMIN, VEL, C)}
C = C/25.4
\text{OLD_G} = C
\text{CALL FINDK(SNAME, BETAMIN, DVEL, KIHETA)}
\text{OLD_K} = KIHETA
\text{DO 100 J = 1, SAMP}
   \text{TJ} = \text{OUT(J, 1)}
   \text{CALL RESPONSE(C, YINC, D, TJ, THETAMIN, RESP)}
   \text{VOLD(J) = RESP*KIHETA/KMAX}
\text{CONTINUE}
\text{DO 150 I = 1, ANGNUM+1}
   \text{AI} = \text{I}
   \text{Y} = \text{L} - \text{BOT} + ((\text{AI}-1.0) \times \text{YINC})
   \text{THETA} = \text{DATAN2}((-\text{Y}), \text{D})
   \text{BETA} = \text{ALPHA+THETA}
\text{Determine limits of beam at receiver for this angle}
   \text{YMIN} = \text{D*DTAN(THETA)}
   \text{YMAX} = (\text{D*DTAN(THETA)}) + \text{L}
\text{IF(YMIN .LE. BOT) THEN}
   \text{LOWLIM = BOT}
\text{ELSE}
   \text{LOWLIM = YMIN}
\text{ENDIF}
\text{IF(YMAX .GE. TOP) THEN}
   \text{UPLIM = TOP}
\text{ELSE}
   \text{UPLIM = YMAX}
\text{ENDIF}
\text{WIDTH} = \text{UPLIM - LOWLIM}
\text{Call subroutine to find velocity at this angle}
\text{CALL FINDVEL(BETA, VEL, C)}
C = C/25.4
\text{Find inclination factor for this angle}
\text{CALL FINDK(SNAME, BETA, DVEL, KIHETA)}
C Find next value of integrand at each time, for use in
integral and multiply integrand by inclination factor

DO 110 J=1,SAMP
  TJ=OUT(J,1)
  CALL RESPONSE(C,YINCR,D,TJ,THETA,resp)
  OUT(J,2) = RESP*KTHETA/KMAX
CNTINUE

C Provided we're not near 'edge' (branch point),
we can simply use trapezoidal rule for integral.

DO 120 J=1,SAMP
  TIME=OUT(J,1)
  VNEW=OUT(J,2)
  IF((VOLD(J) .GT. 0.0) .AND. (VNEW .GT. 0.0)) .OR.
     ((VOLD(J) .EQ. 0.0) .AND. (VNEW .EQ. 0.0)) THEN
    TERM = (VNEW + (VOLD(J)-VNEW)/2.0)*(-YINCR)
    OUT(J,3)=OUT(J,2)+(TERM*C*WIDIH)
  ELSE

C If we are near the branch point, must perform
integral here more carefully.

  DOUBLEY = .TRUE.
  IF (VNEW .EQ. 0.0) THEN
    SIGN = -1.
    REFY = Y - 2.*YINCR
    THETA = DATAN2((-REFY),D)
    BETA=ALPHA+THETA
    CALL FINDVEL(BETA,VEL,C1)
    C1=C1/25.4
    CALL RESPONSE(C1,YINCR,D,TIME,THETA,resp)
    CALL FINDK(SNAME,BETA,DVEL,K1)
    VREF = RESP*K1/KMAX
  ENDIF
  ELSE
    ENDDIF

C otherwise, entering region where response non-zero
ELSE

SIGN = +1
REFY = Y + YINCR
THETA = DATAN2((-REFY),D)
BETA=ALPHA+THETA
CALL FINDVEL(BETA,VEL,C3)
C3=C3/25.4
CALL RESPONSE(C3,YINCR,D,TIME,THETA,resp)
CALL FINDK(SNAME,BETA,DVEL,K3)
VREF = RESP*K3/KMAX
IF (VREF .EQ. 0.0) THEN
  REFY = Y
  THETA = DATAN2((-REFY),D)
  BETA=ALPHA+THETA
  CALL FINDVEL(BETA,VEL,C4)
  C4=C4/25.4
  CALL RESPONSE(C4,YINCR,D,TIME,THETA,resp)
  CALL FINDK(SNAME,BETA,DVEL,K4)
  VREF = RESP*K4/KMAX
DOUBLEY = .FALSE.
ELSE
  ENDIF

ENDIF

Find exact location of the 'edge'
CID = ((C*TIME)**2 - D**2)
YEOOE = SIGN*DSQR(CID)
VINIT = VREF
V = VINIT
DELV = 50.*VINIT/ANGNUM
EDGESUM = 0.D0
EVALOLD = ABS(REFY-YEDGE)
ELIMIT = .001D0*EVALOLD
IF (EVALOLD .GT. ELIMIT) THEN
  V = V+DELV
  EVAL =ABS(YEDGE-(SIGN*DSQR(CID-(1.D0/V**2))))
  EDGESUM = EDGESUM+(EVAL+(EVALOLD-EVAL)/2.D0)*DELV
  EVALOLD = EVAL
GOTO 70
ENDIF

Keep track of integral so far
Add to appropriate positions
TERM=((VINIT*ABS(YEDGE-REFY))+EDGESUM)*WIDTH
OUT(J,3)=OUT(J,3)+(TERM*C)
IF (DOUBLEY) THEN
  IF (SIGN .EQ. -1) THEN
    Subtract previous contribution
    OUT(J,3)=OUT(J,3)-((VOID(J)+VREF)/2.0)*(-YINCR)*WIDTH*OLDC
  ELSE
    VNEW = VINIT
  ENDIF
Y = REFY
ENDIF
ENDIF
ENDIF
VOID(J) = VNEW
CONTINUE
OLDC=C
OLDK=KTHETA
OWIDTH=WIDTH
CONTINUE
RETURN
END

SUBROUTINE RESPONSE(C,YINCR,D,T,THETA,resp)

This routine calculates the response at a given point and time.
IMPLICIT REAL*8(A-H,L-Z)
PI = 4.0*DATA(1,0.0D+00)
R = D/DOGS(THETA)
IF (R .GE. C*T) THEN
   RESP = 0.0
ELSE
   EDENOM = DSQRT((C*T)**2 - R**2)
   RESP = 1.0D00/EDENOM
ENDIF
RETURN
END

SUBROUTINE UNITAPRESP(VEL,DVEL,L,LOUT,BOT,D,ALPHA,
MININT,DELTAT,ANGNUM,SAMP,YPSAMP,KMAX,SNAME,YPSTP)

This routine calculates the tap-to-tap impulse response in the time domain on an isotropic or anisotropic substrate, for the case of a tap of length L transmitting, recording the response at various points over a range LOUT, using a double integral reduction-like technique and incorporating the angular dependence of the electromechanical coupling coefficient.
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
INTEGER*4 OLDHIK,TNUM,YPNUM,TLOK,THIK
INTEGER*4 ANGNUM,YPSAMP,SAMP,LOK,HIK,OLDLOK
PARAMETER (TNUM=1400,YPNUM=10)
![Image](image.png)

```fortran
CHARACTER*32 ONAME, SNAME
REAL*8 VEL(1:181,1:2), DVEL(1:181,1:2), L, LOUT
REAL*8 KMAX, KTHETA, K1, K2, K3, K4, VOLD(1:TNUM)
REAL*8 MININT, UNIOUT(1:TNUM, 1:YPSAMP, 1:2), TEMP(TNUM)
COMMON /EI/UNIOUT
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D+00)

C Calculate position of top of tap (bottom given)
TOP = BOT + LOUT
THETAMIN = DATAN2((BOT-L), D)
THETAMAX = DATAN2(TOP, D)
BETAMIN = THETAMIN + ALPHA
BETAMAX = THETAMAX + ALPHA

C Calculate step sizes for y and yp
YINCR = -(L-BOT+TOP)/ANGNUM

C Initialize output array
DO 90 J=1, SAMP
   TEMP(J)=0.0
   DO 85 K=1, YPSAMP
      UNIOUT(J,K,2)=0.0
85 CONTINUE
90 CONTINUE

C Initialize values used in integral loop
TERM = 0.0
WIDTH = 0.0
OWIDTH = 0.0
CALL FINDVEL(BETAMIN, VEL, C)
C=C/25.4
OLDC=C
OLDLK=0
OLDLK=0
CALL FINDK(SNAME, BETAMIN, DVEL, KTHETA)
OLDK=KTHETA
DO 100 J=1, SAMP
   AJ=J
   T=MININT+(AJ-1.)*DELTAT
   CALL RESPONSE(C, YINCR, D, T, THETAMIN, RESP)
   VOLD(J) = RESP*KTHETA/KMAX
100 CONTINUE

C Loop 150 calculates the integral over transmitting tap
at a set of predetermined times

DO 150 I=1, ANGNUM+1
   AI=I
   Y = L - BOT + (AI-1.)*YINCR
   DO 140 K=1, YPSAMP
      UNIOUT(J,K,2)=UNIOUT(J,K,2)+VOLD(J)*DVEL(AJ,K,2)
140 CONTINUE
150 CONTINUE
```

\[
\theta = \text{DATAN}2((-Y), D) \\
\beta = \text{ALPHA} + \theta \\
\]

Determine limits of beam at receiver for this angle
\[
\text{YMIN} = D \times \text{DIAN} (\theta) \\
\text{YMAX} = (D \times \text{DIAN} (\theta)) + L \\
\]

IF (YMIN \leq \text{BOT}) THEN
\[
\text{LOWLIM} = \text{BOT} \\
\text{ELSE} \\
\quad \text{LOWLIM} = \text{YMIN} \\
\text{ENDIF} \\
\]

IF (YMAX \geq \text{TOP}) THEN
\[
\text{UPLIM} = \text{TOP} \\
\text{ELSE} \\
\quad \text{UPLIM} = \text{YMAX} \\
\text{ENDIF} \\
\]

\text{WIDTH} = \text{UPLIM} - \text{LOWLIM} \\
\]

Find out which spatial bins this corresponds to
\[
\text{TIDK} = 1 \\
\text{IF} (\text{TIDK} \geq \text{YPSAMP}) \text{ THEN} \\
\quad \text{TIDK} = \text{YPSAMP} + 1 \\
\text{ELSE IF} (\text{UNIOUT}(1, \text{TIDK}, 1) \text{ LT} \text{ LOWLIM}) \text{ THEN} \\
\quad \text{TIDK} = \text{TIDK} + 1 \\
\quad \text{GOTO 70} \\
\text{ENDIF} \\
\]

\text{JTEMP} = (\text{WIDTH} / \text{YPSTP}) - 1 \\
\text{THIK} = \text{TIDK} + \text{JTEMP} \\
\text{IF} (\text{THIK} \text{ EQ} 0) \text{ THEN} \\
\quad \text{THIK} = 1 \\
\text{ENDIF} \\
\]

\text{IF} (\text{THIK} \geq \text{YPSAMP}) \text{ THEN} \\
\quad \text{THIK} = \text{YPSAMP} \\
\text{ELSE IF} (\text{UPLIM} \text{ GT} \text{ UNIOUT}(1, \text{THIK}, 1)) \text{ THEN} \\
\quad \text{THIK} = \text{THIK} + 1 \\
\quad \text{GOTO 80} \\
\text{ELSE} \\
\quad \text{THIK} = \text{THIK} - 1 \\
\text{ENDIF} \\
\]

\text{HIK} = \text{THIK} \\
\text{LOK} = \text{TIDK} \\
\]

Call subroutine to find velocity at this angle
\[
\text{CALL FINDVEL(BETA, VEL, C)} \\
\text{C} = \text{C} / 25.4 \]
Find inclination factor for this angle
CALL FINDK(SNAME,BETA,DVEL,KIHETA)

Find next value of integrand at each time, for use in
integral and multiply integrand by inclination factor
DO 110 J=1,SAMP
   AJ=J
   T=MININT+(AJ-1.)*DELTAT
   CALL RESPONSE(C,YINCR,D,T,THETA,resp)
   TEMP(J) = RESP*KIHETA/KMAX
   CONTINUE

Provided we're not near 'edge' (branch point),
we can simply use trapezoidal rule for integral.

DO 120 J=1,SAMP
   AJ=J
   T=MININT+(AJ-1.)*DELTAT
   VNEW=TEMP(J)
   IF(((VOLD(J) .GT. 0.0) .AND. (VNEW .GT. 0.0)) .OR. 
      (VOLD(J) .EQ. 0.0) .AND. (VNEW .EQ. 0.0))) THEN
      TERM = ((VNEW + VOLD(J))/2.0)*(-YINCR)
      DO 112 K=IDK,HIK
         UNIOUT(J,K,2)=UNIOUT(J,K,2)+(TERM*C*YPSTP)
      CONTINUE
   ELSE

If we are near the branch point, must perform
integral here more carefully.

   DOUBLEV = .TRUE.
   IF (VNEW .EQ. 0.0) THEN
      SIGN = -1.
      REFY = Y - 2.*YINCR
      THETA = DATAN2((-REFY),D)
      BETA=ALPHA+THETA
      CALL FINDVEL(BETA,VEL,C1)
      C1=C1/25.4
      CALL RESPONSE(C1,YINCR,D,T,THETA,resp)
      CALL FINDK(SNAME,BETA,DVEL,K1)
      VREF = RESP*K1/KMAX
   ELSE
      IF (VREF .EQ. 0.0) THEN
         REFY = Y-YINCR
         THETA = DATAN2((-REFY),D)
         BETA=ALPHA+THETA
         CALL FINDVEL(BETA,VEL,C2)
         C2=C2/25.4
         CALL RESPONSE(C2,YINCR,D,T,THETA,resp)
   END IF

Need to allow for when signal first arrives at point

IF (VREF .GT. 0.0) THEN
   REFY = Y-YINCR
   THETA = DATAN2((-REFY),D)
   BETA=ALPHA+THETA
   CALL FINDVEL(BETA,VEL,C2)
   CALL RESPONSE(C2,YINCR,D,T,THETA,resp)
ELSE

If we are near the branch point, must perform
integral here more carefully.

   DOUBLEV = .TRUE.
   IF (VNEW .EQ. 0.0) THEN
      SIGN = -1.
      REFY = Y - 2.*YINCR
      THETA = DATAN2((-REFY),D)
      BETA=ALPHA+THETA
      CALL FINDVEL(BETA,VEL,C1)
      C1=C1/25.4
      CALL RESPONSE(C1,YINCR,D,T,THETA,resp)
      CALL FINDK(SNAME,BETA,DVEL,K1)
      VREF = RESP*K1/KMAX
   ELSE
      IF (VREF .EQ. 0.0) THEN
         REFY = Y-YINCR
         THETA = DATAN2((-REFY),D)
         BETA=ALPHA+THETA
         CALL FINDVEL(BETA,VEL,C2)
         C2=C2/25.4
         CALL RESPONSE(C2,YINCR,D,T,THETA,resp)
   END IF
CALL FINDK(SNAME,BETA,DVEL,K2)
VREF = RESP*K2/KMAX
DOUBLEY = .FALSE.
ELSE
ENDIF
otherwise, entering region where response non-zero
ELSE
SIGN = +1
REFY = Y + YINCR
THETA = DATAN2((-REFY),D)
BETA=ALPHA+THETA
CALL FINDVEL(BETA,VEL,C3)
C3=C3/25.4
CALL RESPONSE(C3,YINCR,D,T,THETA,resp)
CALL FINDK(SNAME,BETA,DVEL,K3)
VREF = RESP*K3/KMAX
IF (VREF .EQ. 0.0) THEN
REFY = Y
THETA = DATAN2((-REFY),D)
BETA=ALPHA+THETA
CALL FINDVEL(BETA,VEL,C4)
C4=C4/25.4
CALL RESPONSE(C4,YINCR,D,T,THETA,resp)
CALL FINDK(SNAME,BETA,DVEL,K4)
VREF = RESP*K4/KMAX
DOUBLEY = .FALSE.
ELSE
ENDIF
ENDIF
C
Find exact location of the 'edge'
CID = ((C*T)**2 - D**2)
YEDGE = SIGN*DSQRT(CID)
VINIT = VREF
V = VINIT
DELV = 50.*VINIT/ANGNUM
EDGESUM = 0.D0
EVALOLD = ABS(REFY-YEDGE)
ELIMIT = .001D0*EVALOLD
IF (EVALOLD .GT. ELIMIT) THEN
V = V+DELV
EVAL=ABS(YEDGE-(SIGN*DSQRT(CID-(1.D0/V**2))))
EDGESUM=EDGESUM+(EVAL+(EVALOLD-EVAL)/2.D0)*DELV
EVALOLD = EVAL
GOTO 370
ENDIF
C
Keep track of integral so far
Add to appropriate positions
TERM= ((VINIT*ABS(YEDGE-REFY)) + EDGESUM)
DO 380 K=LOK,HIK
UNIOUT(J,K,2)=UNIOUT(J,K,2)+(TERM*C*YPSTP)

CONTINUE
IF (DOUBLEY) THEN
  IF (SIGN .EQ. -1) THEN
    Subtract previous contribution
    DO 390 K=OLDLOK,OLDHIK
        UNIOUT(J,K,2)=UNIOUT(J,K,2)
        -(VOLD(J)+VREF)/2.0
    END DO
    *(-YINCR) *YPSTP*OLDC
  CONTINUE
  ELSE
    VNEW = VINIT
    Y = REFY
    ENDIF
  ENDIF
ENDIF
VOLD(J) = VNEW

OLDC=C
OLDK=KIHEA
OLDLOK=LOK
OLDHIK=HIK

RETURN
END

SUBROUTINE RESPONSE(C,YINCR,D,T,THETA,resp)

This routine calculates the response at a given point and time.

IMPLICIT REAL*8(A-H,L-Z)
PI = 4.0*DATAN(1.0D+00)
R = D/DOOS(THETA)
IF (R .GE. C*T) THEN
    RESP = 0.0
ELSE
    EDENOM = DSQRT((C*T)**2 - R**2)
    RESP = 1.0D00/EDENOM
ENDIF
RETURN
END

PROGRAM MAIN
SUBROUTINE TAPRESPF

This routine calculates the approximate
tap-to-tap impulse response on an isotropic substrate in the frequency domain, using a double integral reduction technique.

```fortran
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*2(I-K)
INTEGER*4 POINTS
PARAMETER(POINTS=16384)
REAL*8 FMAG(POINTS),FPHASE(POINTS),IPART
COMMON // FMAG,FPHASE
CHARACTER*32 FNAME9,FNAME8
PI=4.0*DATA(1.0D+0000)
```

Enter necessary values

```fortran
WRITE(*,*) 'Enter center frequency, in MHz: ' 
READ(*,*) fo
WRITE(*,*) 'Enter velocity of wave, in m/sec: ' 
READ(*,*) c
WRITE(*,*) 'Enter width of transmitting tap, in wavelengths: 
READ(*,*) L
WRITE(*,*) 'Enter width of receiving tap, in wavelengths: ' 
READ(*,*) LOUT
WRITE(*,*) 'Enter horizontal distance between taps, in wavelengths: 
* ' 
READ(*,*) D
WRITE(*,*) 'Enter vertical position of bottom of second tap, relative to bottom of first tap, in wavelengths: ' 
READ(*,*) BOT
WRITE(*,*) 'Enter number of angular increments desired for integral: ' 
READ(*,*) ANGNUM
WRITE(*,*) 'Enter frequency sampling rate, in samples per MHz: ' 
READ(*,*) FSAMP
WRITE(*,*) 'Enter minimum frequency of interest, in MHz:' 
READ(*,*) FMIN
WRITE(*,*) 'Enter maximum frequency of interest, in MHz: ' 
READ(*,*) FMAX
WRITE(*,*) 'Enter name of output file for linear data: ' 
READ(*,15) FNAME9
WRITE(*,*) 'Enter name of output file for data in dB: ' 
READ(*,16) FNAME8
```

Calculate values

```fortran
WVLENGTH = C/(fo*1.E+06)
L = L*WVLENGTH
```
D = D*WVLNGTH
BOT = BOT*WVLNGTH
TOP = BOT + (LOUT*WVLNGTH)
FMIN = FMIN*1.0D+06
FMAX = FMAX*1.0D+06
THETMAX = DATAN2(TOP,D)
THETMIN = DATAN2(BOT-L,D)
YINCR = -(L-BOT+TOP)/ANGNUM

C Set step sizes for frequency
DELTAF = 1.0D+06/FSAMP

C Find the minimum number of frequency increments to get from FMIN to FMAX
N = 0
F = FMIN
10 IF (F .LT. FMAX) THEN
   F = F + DELTAF
   N = N+1
   GOTO 10
ENDIF

C At least N frequency increments are required
NUMINCRF = N

C Initialize variables
DO 120 J = 1,NUMINCRF
   FMAG(J) = 0.0
   FHASE(J) = 0.0
120 CONTINUE
RPART = 0.0
IPART = 0.0
TRPART = 0.0
TIPART = 0.0

C This loop sets a value for frequency, then performs the integral
C of real and imaginary parts separately by double integral reduction,
C then finds the magnitude and phase of the overall response
C at this frequency.
C
OPEN(9,FILE=FNAME9)
OPEN(8,FILE=FNAME8)
DO 180 J = 1,NUMINCRF+1
   AJ = J
   F = FMIN + ((AJ-1.0)*DELTAF)
   OMEGA = 2.0*PI*F
C
C THIS LOOP PERFORMS INTEGRAL
DO 50, I = 1, ANGNUM+1
   AI = I
   Y = L-BOT + ((AI-1.0)*YINCR)
   R = DSQRT(Y**2 + D**2)
   THETA = DATAN2(-Y, D)
   YMIN = D*TAN(THETA)
   YMAX = (D*TAN(THETA)) + L

C
C Determine limits of beam at receiver for this angle
   IF(YMIN .LE. BOT) THEN
      LOWLIM = BOT
   ELSE
      LOWLIM = YMIN
   ENDIF

C
C IF(YMAX .GE. TOP) THEN
   UPLIM = TOP
   ELSE
      UPLIM = YMAX
   ENDIF

C
C WIDIH = UPLIM - LOWLIM

C
C Calculate real and imaginary parts of integral

C
C COEFF = 0.25*DSQRT((2.0*C)/(OMEGA*PI*R))
   ARG = -((OMEGA*R)/C + (PI/4.0))
   TRPARI' = COEFF*DCOS(ARG) * WIDIH*(-YINCR)
   TIPARI' = COEFF*DSIN(ARG) * WIDIH*(-YINCR)
   RPARI' = RPARI' + TRPARI'
   IPARI' = IPARI' + TIPARI'

50  CONTINUE

FMAG(J+1) = DSQRT((RPARI'**2) + (IPARI'**2))
FPHASE(J+1) = DATAN2(IPARI', RPARI')

180  CONTINUE
DO 200 K=1, NUMINCRF+1
   AK = K
   F = FMIN + (AK-1.0)*DELTAF
   WRITE(9,210) (F/1.0D+06), FMAG(K), FPHASE(K)
   WRITE(8,210) (F/1.0D+06), (20.0*DLOG10(FMAG(K))),
               (FPHASE(K)*180.0/PI)
200  CONTINUE
210  FORMAT(E18.8,1X,E18.8,1X,E18.8)
CLOSE (9)
CLOSE (8)

C
C STOP
END
SUBROUTINE UNIPTAPRESP(VEL,DVEL,L,LOUT,D,BOT,ALPHA,ANGNUM, FMIN,DELITAF,NSAMPF,YPSAMFP,KMAX,SNAME,YPSTP)
* 
This routine calculates the approximate 
tap-to-tap impulse response on an isotropic 
or anisotropic substrate in the frequency domain, 
using a double integral reduction technique.

IMPLICIT REAL*8(A-H,L-Z) 
IMPLICIT INTEGER*2(I-K) 
INTEGER*4 FNUM,YPSAMFP,NSAMPF,LOK,H1K,TLOK,TH1K 
INTEGER*4 YPNUM,ANGNUM 
PARAMETER(FNUM=2048,YPNUM=10) 
REAL*8 UNICXJr(1:FNUM,1:YPNUM,1:3),RPAR1(1:FNUM,1:YPNUM),IM 
REAL*8 IPAR1(1:FNUM,1:YPNUM),VEL(1:181,1:2),DVEL(1:181,1:2) 
REAL*8 KMAX,KTHETA 
CHARACTER*32 SNAME 
COMMON /UN1V/ UNICXJr 
COMMON /UNIBL/ RPAR1,IPAR1 
PI=4.0*DATAN(1.0D+0000)

Calculate values 
TOP = BOT + LOUT 
WRITE(*,*) 'L=',L,'LO=',LOUT,'BOT=',BOT 
WRITE(*,*) 'ANGNUM=',ANGNUM 
THETAMAX = DATAN2(TOP,D) 
THETAMIN = DATAN2(BOT-L,D) 
YINCR = -((L-BOT+TOP)/ANGNUM) 
WRITE(*,*) 'YINC=',YINCR 

Initialize variables 
DO 120 J = 1,NSAMPF 
   DO 110 K=1,YPSAMFP 
      UNICXJr(J,K,2) = 0.0 
      UNICXJr(J,K,3) = 0.0 
      RPAR1(J,K) = 0.0 
      IPAR1(J,K) = 0.0 
   110 CONTINUE 
120 CONTINUE 

This loop sets a value for frequency, then performs the 
integral 
of real and imaginary parts separately by double integral 
reduction, 
then finds the magnitude and phase of the overall response 
at this 
frequency.

THIS LOOP PERFORMS INTEGRAL 
DO 50, I = 1,ANGNUM+1
AI = I
Y = L \cdot \text{BOT} + ((AI - 1.0) \cdot \text{YINC})
R = \text{DSQRT}(Y**2 + D**2)
THETA = \text{DATAN2}(-Y, D)
BETA = \text{ALPHA} + \text{THETA}

\text{C Determine limits of beam at receiver for this angle}
YMIN = D \cdot \text{DTAN}(\text{THETA})
YMAX = (D \cdot \text{DTAN}(\text{THETA})) + L
\text{IF}(\text{YMIN} \leq \text{BOT}) \text{ THEN}
   \text{LOWLIM} = \text{BOT}
\text{ELSE}
   \text{LOWLIM} = \text{YMIN}
\text{ENDIF}

\text{IF}(\text{YMAX} \geq \text{TOP}) \text{ THEN}
   \text{UPLIM} = \text{TOP}
\text{ELSE}
   \text{UPLIM} = \text{YMAX}
\text{ENDIF}

\text{WIDTH} = \text{UPLIM} - \text{LOWLIM}

\text{C Find out which spatial bins this corresponds to}
\text{TLOK} = 1
\text{IF}(\text{TLOK} \geq \text{YPSAMP}) \text{ THEN}
   \text{TLOK} = \text{YPSAMP} + 1
\text{ELSE IF}(\text{UNIOUT}(1, \text{TLOK}, 1) \lt \text{LOWLIM}) \text{ THEN}
   \text{TLOK} = \text{TLOK} + 1
\text{GOTO 70}
\text{ENDIF}

\text{JTEMP} = (\text{WIDTH}/\text{YPSTP}) - 1
\text{THIK} = \text{TLOK} + \text{JTEMP}
\text{IF}(\text{THIK} \leq 0) \text{ THEN}
   \text{THIK} = 1
\text{ENDIF}

\text{IF}(\text{THIK} \geq \text{YPSAMP}) \text{ THEN}
   \text{THIK} = \text{YPSAMP}
\text{ELSE IF}(\text{UPLIM} \gt \text{UNIOUT}(1, \text{THIK}, 1)) \text{ THEN}
   \text{THIK} = \text{THIK} + 1
\text{GOTO 80}
\text{ELSE}
   \text{THIK} = \text{THIK} - 1
\text{ENDIF}

\text{HIK} = \text{THIK}
\text{LOK} = \text{TLOK}

\text{C Call subroutine to find velocity at this angle}
\text{CALL FINDVEL(BETA, VEL, CINT)}
WRITE(*,*) 'CINT=', CINT
C=CINT/25.4

Find inclination factor for this angle
CALL FINDK(SNAME,BETA,DVEL,KTHETA)
WRITE(*,*) 'K=', KTHETA

DO 180 J=1,NSAMPF
AJ=J
F=FMIN+(AJ-1.0)*DELTAF
OMEGA=2.0*PI*F

Find next value of integrand for use in integral
WRITE(*,*) 'ANG=', THETA, 'D=', D, 'F=', F, DCOS(THETA)
CALL FRESPONSE(C,YINCR,D,THETA,F,RE,IM)
WRITE(*,*) 'RE=', RE, 'IM=', IM

Multiply integrand by inclination factor and dy
RE = RE*KTHETA/KMAX*(-YINCR)
IM = IM*KTHETA/KMAX*(-YINCR)

Add response for this angle to sum
DO 170 K1=1,NSAMPF
RPART(J,K1) = RPART(J,K1) + RE*YPSTP
IPART(J,K1) = IPART(J,K1) + IM*YPSTP
CONTINUE
170
180 CONTINUE
50 CONTINUE

SUBROUTINE FRESPONSE(C,YINCR,D,THETA,FNUM,RE,IM)

This routine calculates the frequency domain pt-pt response
at a given angle theta between the ray connecting the two
FNUM of interest and a line normal to the transducer.
IMPLICIT REAL*8(A-H,L-Z)
REAL*8 IM
PI = 4.0*DATAN(1.0D+00)
OMEGA = 2.0*PI*FINIT
R = D/DCOS(THETA)
A = DSQRT((C*DCOS(THETA))/(PI**2)*D*FINIT))
B = (OMEGA*D)/(C*DCOS(THETA))
RE = A*DCOS(B)
IM = (-1.0)*A*DSIN(B)
RETURN
END

SUBROUTINE FINDVEL(THETA, VEL, CINT)

This subroutine calculates the velocity at a given angle by linear interpolation between data PNUM in the array VEL (in m/sec).

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 VEL(1:181,1:2)
CHARACTER*32 SNAME
PI = 4.0*DATAN(1.0d00)

Substrate is anisotropic, must find velocity for angle

DO 100 I = 1, 180
AI = I
TEMPL = -(PI/2.0) + (AI-1.)*PI/180.
TEMPH = -(PI/2.0) + (AI)*PI/180.

IF ((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN
Interpolate between values at K and K+1 in array
CLO = VEL(K,1)
CHI = VEL(K+1,1)
ANGDIF = THETA - TEMPL
CINT = CLO + (CHI-CLO)*180.*(ANGDIF)/PI
ELSE
K=K+1
ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE FINDK(SNAME, THETA, DVEL, KTHETA)
This subroutine calculates \( k \), the square root of the electromechanical coupling coefficient, using linear interpolation between \( F_{\text{NUM}} \) in an array of \( \Delta(v)/v \) vs. \( \theta \) (array name: \( \text{DVEL} \)).

\[
\text{IMPLICIT REAL*8(A-H,L-Z)}
\]
\[
\text{IMPLICIT INTEGER*4(I-K)}
\]
\[
\text{REAL*8 DVEL}(1:181,1:2)
\]
\[
\text{REAL*8 KTHETA}
\]
\[
\text{CHARACTER*32 SNAME}
\]
\[
\text{PI} = 4.0*\text{DATAN}(1.0D00)
\]

See if substrate is isotropic

\[
\text{IF (SNAME .EQ. 'I') THEN}
\]
\[
\text{KTHETA} = 1.0
\]
\[
\text{ELSE}
\]

Substrate is anisotropic, must find \( k \) for angle used

Determine which two angles in the array \( \theta \) lies between

\[
\text{DO 100 I = 1,181}
\]
\[
\text{TEMPL} = \text{DVEL}(I,1)*\text{PI}/180.
\]
\[
\text{TEMPH} = \text{DVEL}(I+1,1)*\text{PI}/180.
\]

\[
\text{IF ((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN}
\]

Interpolate between values at \( I \) and \( I+1 \) in array

\[
\text{DVLO} = \text{DVEL}(I,2)
\]
\[
\text{DVHI} = \text{DVEL}(I+1,2)
\]
\[
\text{ANGDIF} = \theta - \text{TEMPL}
\]
\[
\text{DVNIHETA} = \text{DVLO} + (\text{DVHI}-\text{DVLO})*180.*\text{ANGDIF}/\text{PI}
\]

\[
\text{ENDIF}
\]
\[
\text{100 CONTINUE}
\]

Calculate the value for \( k \) using this \( \Delta(v)/v \)

\[
\text{KTHETA} = \text{DSQRT}(2.0*\text{DVNIHETA})
\]

\[
\text{ENDIF}
\]

\[
\text{RETURN}
\]
\[
\text{END}
\]

\[
\text{SUBROUTINE MAXK(SNAME,DVEL,KMAX)}
\]

This subroutine searches the \( \Delta(v)/v \) array \( \text{DVEL} \) and returns the maximum value of \( k \), the square root of the coupling coefficient.

\[
\text{IMPLICIT REAL*8(A-H,L-Z)}
\]
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KMAX
CHARACTER*32 SNAME

C See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
  KMAX = DSQRT(2.0*DVEL(1,2))
ELSE
  Substrate is anisotropic, search for maximum
  TEMPMAX = DVEL(1,2)
  DO 100 I = 1,180
    IF(DVEL(I+1,2).GE.TEMPMAX) THEN
      TEMPMAX = DVEL(I+1,2)
    ENDIF
  100 CONTINUE
  KMAX = DSQRT(2.0*TEPMAX)
ENDIF

C
RETURN
END

SUBROUTINE FILEPREP(FN1,FN2,M,N,S1ANT,DIREC,S1ANT2,DIREC2,*
  W1,W2)

This routine converts finger length time files produced by IMP to files containing the effective
tap lengths and positions for use in diffraction analysis.

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
INTEGER*4 FIRSTTAP,N1AP1,N1AP2,M,N,TAPNUM
PARAMETER (TAPNUM=25)
CHARACTER*32 CNAME,ONAME2,FSUB,SNAME,DIREC
CHARACTER*32 DIREC2,FN1,FN2
REAL*8 VEL(1:181,1:2),T1DAT(1:TAPNUM,1:2)
REAL*8 T1FING(1:TAPNUM),T2FING(1:TAPNUM)
REAL*8 T2DAT(1:TAPNUM,1:2)
COMMON /B3/T1DAT,T2DAT
LOGICAL*4 DOUBLEY
PI = 4.0*DATAN(1.0D+00)

C
Read in finger length data for first transducer
OPEN(30,FILE=FN1,ANYUNIT=INIV,ACTION='READ')
READ(30,'(ABO)') CHAR STR
READ(30,'(ABO)') CHAR STR
DO 70 J=1,M
    READ(30,80) T1FING(J)
CONTINUE
70 FORMAT(F10.6)
CLOSE(30)
C Read in finger length data for second transducer
OPEN(42,FILE=FN2,ANYUNIT=INT,ACTION='READ')
READ(42,'(ABO)') CHAR STR
READ(42,'(ABO)') CHAR STR
DO 90 J=1,N
    READ(42,80) T2FING(J)
CONTINUE
CLOSE(42)
C C Calculate values
C number of effective taps
NTAP1 = M-1
NTAP2 = N-1
C C Initialize values
DO 91 J=1,NTAP1
    T1DAT(J,1)=0.0
    T1DAT(J,2)=0.0
CONTINUE
DO 92 J=1,NTAP2
    T2DAT(J,1)=0.0
    T2DAT(J,2)=0.0
CONTINUE
C C Calculate effective tap lengths - first transducer
DO 100 J=1,NTAP1
    IF ( ((T1FING(J) .LE. 0) .AND. (T1FING(J+1) .LE. 0))
       * 
       .OR. ((T1FING(J) .GE. 0) .AND. (T1FING(J+1) .GE. 0)))
       * 
       THEN
       T1DAT(J,1)= 0.0
       ELSE IF (T1FING(J) .LE. 0) THEN
       T1DAT(J,1)=+ (ABS(T1FING(J)) + ABS(T1FING(J+1)))
       T1DAT(J,1)=T1DAT(J,1)*W1/2.
       ELSE
       T1DAT(J,1)=- (ABS(T1FING(J)) + ABS(T1FING(J+1)))
       T1DAT(J,1)=T1DAT(J,1)*W1/2.
    ENDIF
CONTINUE
C C Calculate effective tap lengths - second transducer
DO 103 J=1,NTAP2
    IF ( ((T2FING(J) .LE. 0) .AND. (T2FING(J+1) .LE. 0))
       * 
       .OR. ((T2FING(J) .GE. 0) .AND. (T2FING(J+1) .GE. 0)))
* THEN
  T2DAT(J,1)=0.0
ELSE IF (T2FING(J) .LE. 0) THEN
  T2DAT(J,1)=(ABS(T2FING(J))+ABS(T2FING(J+1)))/2.
ELSE
  T2DAT(J,1)=-(ABS(T2FING(J))+ABS(T2FING(J+1)))/2.
ENDIF
ENDIF
CONTINUE

C Perform slanting of apodization for first transducer
IF((DIREC .EQ. 'UP') .OR. (DIREC .EQ. 'up')) THEN
  FIRSTTAP=1
ELSE
  FIRSTTAP=NTAP1
ENDIF

C IF(NTAP1 .GT. 1) THEN
  SIANISP=((SLANT*W1)-.5*(ABS(T1DAT(1,1))+ABS(T1DAT(NTAP1,1))))
  SIANISP=SIANISP/(NTAP1-1)
ENDIF

C IF(FIRSTTAP .EQ. 1) THEN
  Slant is upward
  IF(ABS(T1DAT(1,1)) .GE. .99*W1) THEN
    T1DAT(1,2)=0.0
  ELSE
    T1DAT(FIRSTTAP,2)=(1.-SLANT)*W1/2.
  ENDIF
DO 110 J=2,NTAP1
  IF(ABS(T1DAT(J,1)) .GE. .99*W1) THEN
    T1DAT(J,2)=0.0
  GOTO 110
ENDIF
AJ=J
CNIR=(AJ-1.)*SIANISP+T1DAT(1,2)+(ABS(T1DAT(1,1))/2.)
UPPR=W1-CNIR
IF((CNIR .LT. ABS(T1DAT(J,1)/2.)) .OR. (UPPR .LT.\n  ABS(T1DAT(J,1)/2.))) THEN
  WRITE(*,*) 'SLANT ANGLE 1 TOO LARGE!!'
  WRITE(*,*) 'Enter new slant, as fraction of aperture:'
  READ(*,*) SLANT
  GOTO 105
ELSE
T1DAT(J,2) = CNIR-ABS(T1DAT(J,1)/2.)
ENDIF
CONTINUE
ELSE
C Slant is downward
IF(ABS(T1DAT(FIRSTTAP,1)) .GE. .99*W1) THEN
T1DAT(FIRSTTAP,2)=0.0
ELSE
T1DAT(FIRSTTAP,2)=(1.-SLANT)*W1/2.
ENDIF
DO 120 J=NTAP1-1,1,-1
IF(ABS(T1DAT(J,1)) .GE. .99*W1) THEN
T1DAT(J,2)=0.0
ENDIF
GOTO 120
ENDIF
AJ=J
CNIR=(NTAP1-AJ)*SLANTSTP+T1DAT(NTAP1,2)+
ABS(T1DAT(NTAP1,1)/2.)
UPPR=W1-CNIR
IF((CNIR .LT. ABS(T1DAT(J,1)/2.)) .OR. (UPPR .LT.
ABS(T1DAT(J,1)/2.))) THEN
WRITE(*,*) 'SLANT ANGLE 1 TOO LARGE!!'
WRITE(*,*) 'Enter new slant, as fraction of
aperture: '
READ(*,*) SLANT
GOTO 105
ELSE
T1DAT(J,2) = CNIR-ABS(T1DAT(J,1)/2.)
ENDIF
120 CONTINUE
C
C Perform slanting of apodization for second transducer
C
IF((DIREC2 .EQ. 'UP') .OR. (DIREC2 .EQ. 'up')) THEN
FIRSTTAP=1
ELSE
FIRSTTAP=NTAP2
ENDIF
C
IF(NTAP2 .GT. 1) THEN
SLANTSTP=((SLANT2*W2)-
.5*(ABS(T2DAT(1,1))+ABS(T2DAT(NTAP2,1))))
SLANTSTP=SLANTSTP/(NTAP2-1)
WRITE(*,*) 'SLANTSTP=', SLANTSTP
IF(FIRSTTAP .EQ. 1) THEN
C Slant is upward
IF(ABS(T2DAT(1,1)) .GE. .99*W1) THEN
T2DAT(1,2)=0.0
ELSE
T2DAT(FIRSTTAP,2)=(1.-SLANT2)*W2/2.
ENDIF
DO 130 J=2,NTAP2
IF(ABS(T2DAT(J,1)) .GE. .99*W2) THEN
T2DAT(J,2)=0.0
GOTO 130
ENDIF
**C**

RETURN

END

**C**

SUBROUTINE MAXK(SNAME, DVEL, KMAX)
This subroutine searches the delta(v)/v array DVEL and returns the maximum value of k, the square root of the Coupling Coefficient.

```
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KMAX

See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
  KMAX=1.0
ELSE
  Substrate is anisotropic, search for maximum
  TEMPMAX=DVEL(1,2)
  DO 100 I=1,180
    IF (DVEL(I+1,2) .GE. TEMPMAX) THEN
      TEMPMAX=DVEL(I+1,2)
    ENDIF
  100 CONTINUE
  KMAX=DSQRT(2.0*TEMPMAX)
ENDIF
RETURN
END

SUBROUTINE FINDVEL(THETA,VEL,CINT)

This subroutine calculates the velocity at a given angle by linear interpolation between data points in the array VEL (in m/sec).

```
IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 VEL(1:181,1:2)
PI = 4.0*DATAN(1.0d00)

K = 1

Determine which two angles in the array theta lies between
DO 100 I = 1,180
  AI = I
  TEMPL = -(PI/2.0) + (AI-1.)*PI/180.
  TEMPH = -(PI/2.0) + (AI)*PI/180.
  IF (((TEMPL .LE. THETA) .AND. (TEMPH .GE. THETA)) THEN
    Interpolate between values at K and K+1 in array
    CLO = VEL(K,1)
    CHI = VEL(K+1,1)
    ANGDIF = THETA - ((-PI/2.0) + (AI-1.)*PI/180.)
```

CINT = CLO + (CHI-CLO)*180.*/(ANGDIF)/PI
ELSE
K=K+1
ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE FINDK(SNAME,THETA,DVEL,KTHETA)

This routine calculates k, the square root of
the electromechanical coupling coefficient, using
linear interpolation between points in an array of
delta(v)/v vs. theta (array name:DVEL)

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 DVEL(1:181,1:2)
REAL*8 KTHETA
CHARACTER*32 SNAME
PI=4.0*DATAN(1.0D0)

See if substrate is isotropic
IF (SNAME .EQ. 'I') THEN
KTHETA = 1.0
ELSE
Substrate is anisotropic, must find k for angle used

Determine which two angles in array theta lies between
DO 100 I=1,181
TEMPL=DVEL(I,1)*PI/180.
TEMPH=DVEL(I+1,1)*PI/180.
C
IF((TEMPL.LE.THETA) .AND. (TEMPH.GE.THETA)) THEN
Interpolate between values at I and I+1 in array
DVLO=DVEL(I,2)
DVHI=DVEL(I+1,2)
ANGDIF=THETA-TEMPL
DVTHERA=DVLO+(DVHI-DVLO)*180.*ANGDIF/PI
ENDIF
100 CONTINUE
C
Calculate value of k using this delta(v)/v
KTHETA=DSQRT(2.0*DVTHERA)
C
ENDIF
APPENDIX C

FOURIER TRANSFORM AND UTILITY PROGRAMS
TRANS.F77

PROGRAM MAIN
IMPLICIT INTEGER*2 (I-K)
IMPLICIT REAL*8 (A-H,L-Z)
REAL*8 MAG

TRANSFORM SUBROUTINE

This routine takes an array which lists time, and linear magnitude,
and converts it to two arrays containing the real and imaginary parts of the time response, and then Fourier transforms the time response using the subroutine FFT.

CHARACTER*32 FNAME,FNM2 :
INTEGER NUM
REAL*8 FM(4096),FP(4096)
REAL*4 RARRAY(4096),IARRAY(4096)

Determine size of arrays needed
WRITE(*,*) 'Enter number of samples:'
READ(*,*) NUM

Initialize arrays
DO 50 K=1,4096
   FM(K)=0.0
   FP(K)=0.0
   RARRAY(K)=0.0
   IARRAY(K)=0.0
50 CONTINUE

Read in values from stored data
WRITE(*,*) 'Enter name of data file:'
READ(*,15) FNAME
15 FORMAT(A32)
OPEN(8,FILE=FNAME,STATUS='OLD')

Determine real and imaginary parts
DO 100 J=1,NUM
   READ(8,90) TIME,MAG
   IF(J .EQ. 1) THEN
      TONE=TIME
   ELSE IF (J .EQ. 2) THEN
      TTWO=TIME
   ENDIF
   RARRAY(J)=MAG
   IARRAY(J)=0.0
100 CONTINUE
90 FORMAT(E16.8,1X,E16.8)
CLOSE(8)

DELTAT=TTWO-TONE
Perform Fourier transform
CALL FFT(RARRAY,IARRAY,NUM,NUM,NUM,-1)

Convert to magnitude and phase in frequency domain
DO 200 K=1,NUM
   FM(K) = SQRT((RARRAY(K)**2)+(IARRAY(K)**2))
   FP(K) = ATAN2(IARRAY(K),RARRAY(K))
200 CONTINUE

Write to output files
WRITE(*,*) 'Enter name of output file:'
READ(*,210) FNM2
210 FORMAT(A32)
OPEN(7,FILE= FNM2)
DO 220 I=1,NUM
   AI=I
   WRITE(7,230) (AI-1.)/(NUM*DELTAT),FM(I),FP(I)
220 CONTINUE
230 FORMAT(E16.8,1X,E16.8,1X,E16.8)
CLOSE(7)
STOP
END

HTRANS.F77
PROGRAM MAIN

This routine reads in a file in (frequency,magnitude,phase) form, determines the number of samples to be used in the FFT routine, and then converts the appropriate number of frequency domain samples into Hermitian form. (Note that this assumes that the time domain will be a real function.) The inverse Fourier Transform is then found using the subroutine FFT. The DC offset in the time domain is also calculated.

IMPLICIT REAL*8 (A-H,L-Z)
INTEGER*4 NPTS,N,POINT
PARAMETER(POINTS=16384)
REAL*4 IOUT(POINTS),ROUT(POINTS),TOUT(POINTS),IPART,
* MOUT(POINTS),POUT(POINTS)
COMMON // IOUT,ROUT,TOUT,MOUT,POUT
CHARACTER*32 INAME,ONAME,TEMP

Enter information
WRITE(*,*) 'Enter name of input file:'
READ(*,10) INAME
10 FORMAT(A32)
WRITE(*,*) 'Enter an even number of points to be used for
* inverse transform (may not exceed twice length of input
* file):'
READ(*,*) NPTS
WRITE(*,*) 'Enter name of output file for time domain data:'
READ(*,10) ONAME
WRITE(*,*) 'Enter name of file to hold Hermetian frequency
* info.:'
READ(*,10) TEMP

C Initialize Variables
N = 1
MIN = 0.0
PIN = 0.0
FIN = 0.0
FIN1 = 0.0
FIN2 = 0.0
DO 20 I=1,NPTS
   ROUT(I) = 0.0
   IOUT(I) = 0.0
   TOUT(I) = 0.0
20   CONTINUE

C Read in values from stored data and manipulate them
appropriately.
OPEN(7,FILE=TEMP)
OPEN(8,FILE=INAME,STATUS='OLD')
READ(8,80) FIN,MIN,PIN
   ROUT(1) = MIN*DCOS(PIN)
   IOUT(1) = MIN*DSIN(PIN)
   FIN1 = FIN
DO 100 N=2,NPTS/2
   READ(8,80) FIN,MIN,PIN
80   FORMAT(E16.8,1X,E16.8,1X,E16.8)
C   IF (N .EQ. 2) FIN2 = FIN
C   RPART = MIN*DCOS(PIN)
   IPART = MIN*DSIN(PIN)
C   Make output file Hermetian
   I = NPTS - N + 2
   ROUT(N) = RPART
   ROUT(I) = RPART
   IOUT(N) = IPART
   IOUT(I) = -IPART
100  CONTINUE

C Fill in center value
N = NPTS/2 + 1
C

Find sample spacing in frequency
DELTAF = FIN2 - FIN1
C

Output results for intermediate inspection
DO 150 J=1,NPTS
   WRITE(7,160) (J-1)*DELTAF, ROUT(J), IOUT(J)
150 CONTINUE
160 FORMAT(1E16.8,1X,1E16.8,1X,1E16.8)
C

Perform inverse Fourier Transform:
C
CALL FFT(ROUT,IOUT,NPTS,NPTS,NPTS,+1)
C
Convert to magnitude and phase in time domain
DO 200 J=1,NPTS
   MOUT(J) = SQRT((ROUT(J)**2)+(IOUT(J)**2))
   POUT(J) = ATAN2(IOUT(J),ROUT(J))
200 CONTINUE
C
Write to output files
OPEN(9,FILE=ONAME)
DO 220 K=1,NPTS
   WRITE(9,230) (K-1)/(NPTS*DELTAF*1E+06), * ROUT(K),IOUT(K)
220 CONTINUE
230 FORMAT(1E16.8,1X,1E16.8,1X,1E16.8)
CALL OFFSET(ROUT,OFFSET,NPTS)
WRITE(*,*) 'DC OFFSET = ',OFFSET
CLOSE(9)
STOP
END
C

SUBROUTINE OFFSET(ROUT,OFFSET,NPTS)
IMPLICIT REAL*8 (A-H,L-Z)
IMPLICIT INTEGER*4 (I-K)
INTEGER*4 POINTS,NPTS
PARAMETER(POINTS=4096)
REAL*4 ROUT(POINTS)
OFFSET = 0.0
DO 280 J=1,NPTS
   OFFSET = OFFSET+ROUT(J)
280 CONTINUE
RETURN
END
SUBROUTINE FFT(A,B,NIOT,N,NSPAN,ISN)

MULTIVARIATE COMPLEX FOURIER TRANSFORM, COMPUTED IN PLACE
USING MIXED-RADIX FAST FOURIER TRANSFORM ALGORITHM.
BY R. C. SINGLETON, STANFORD RESEARCH INSTITUTE, OCT. 1968
ARRAYS A AND B ORIGINALLY HOLD THE REAL AND IMAGINARY
COMPONENTS OF THE DATA, AND RETURN THE REAL AND
IMAGINARY COMPONENTS OF THE RESULTING FOURIER COEFFICIENTS.
MULTIVARIATE DATA IS INDEXED ACCORDING TO THE FORTRAN
ARRAY ELEMENT SUCCESSOR FUNCTION, WITHOUT LIMIT
ON THE NUMBER OF IMPLIED MULTIPLE SUBSCRIPTS.
The subroutine is called once for each variate.
The calls for a multivariate transform may be in any order.
NIOT is the total number of complex data values.
N is the dimension of the current variable.
NSPAN/N is the spacing of consecutive data values
while indexing the current variable.
The sign of ISN determines the sign of the complex
exponential, and the magnitude of ISN is normally one.
A tri-variate transform with A(N1,N2,N3), B(N1,N2,N3)
is computed by
CALL FFT(A,B,N1*N2*N3,N1,N1,1)
CALL FFT(A,B,N1*N2*N3,N2,N1*N2,1)
CALL FFT(A,B,N1*N2*N3,N3,N1*N2*N3,1)
FOR A SINGLE-VARIATE TRANSFORM,
NIOT = N = NSPAN = (NUMBER OF COMPLEX DATA VALUES), E.G.
CALL FFT(A,B,N,N,N,1)
THE DATA MAY ALTERNATIVELY BE STORED IN A SINGLE COMPLEX
ARRAY A, THEN THE MAGNITUDE OF ISN CHANGED TO TWO TO
GIVE THE CORRECT INDEXING INCREMENT AND A (2) USED TO
PASS THE INITIAL ADDRESS FOR THE SEQUENCE OF IMAGINARY
VALUES, E.G.
CALL FFT(A,A(2),NIOT,N,NSPAN,2)
ARRAYS AT(MAXF), CK(MAXF), BT(MAXF), SK(MAXF), AND NP(MAXP)
ARE USED FOR TEMPORARY STORAGE. IF THE AVAILABLE STORAGE
IS INSUFFICIENT, THE PROGRAM IS TERMINATED BY A STOP.
MAXP MUST BE .GE. THE MAXIMUM PRIME FACTOR OF N.
MAXP MUST BE .GT. THE NUMBER OF PRIME FACTORS OF N.
IN ADDITION, IF THE SQUARE-FREE PORTION K OF N HAS TWO OR
MORE PRIME FACTORS, THEN MAXP MUST BE .GE. K-1.
DIMENSION A(1),B(1)
ARRAY STORAGE IN NFAC FOR A MAXIMUM OF 11 FACTORS OF N.
IF N HAS MORE THAN ONE SQUARE-FREE FACTOR, THE PRODUCT OF THE
SQUARE-FREE FACTORS MUST BE .LE. 210
DIMENSION NFAC(11),NP(209)
ARRAY STORAGE FOR MAXIMUM PRIME FACTOR OF 23
DIMENSION AT(23),CK(23),BT(23),SK(23)
equivalence (i,ii)
C THE FOLLOWING CONSTANTS SHOULD AGREE WITH THE ARRAY DIMENSIONS.

MAXF=23
MAXFb=209
IF(N .LT. 2) GOTO990
INC=ISN
RAD=8.0*ATAN(1.0)
S72=RAD/5.0
C72=COS(S72)
S72=SIN(S72)
S120=SQR(0.75)
IF(ISN .GE. 0) GO TO 10
S72=-S72
S120=-S120
RAD=-RAD
INC=-INC

10 NI=INC*NTOT
KS=INC*NSPAN
KSPAN=KS
NN=NT-INC
JC=KS/N
RADF=RAD*FLOAT(JC)*0.5
I=0
JF=0

C DETERMINE THE FACTORS OF N
M=0
K=N
GO TO 20

15 M=M+1
NFAC(M)=4
K=K/16

20 IF(K-(K/16)*16 .EQ. 0) GO TO 15
J=3
JJ=9
GO TO 30

25 M=M+1
NFAC(M)=J
K=K/JJ

30 IF(MOD(K,JJ) .EQ. 0) GO TO 25
J=J+2
JJ=J**2
IF(JJ .LE. K) GO TO 30
IF(K .GT. 4) GO TO 40
KT=M
NFAC(M+1)=K
IF(K .EQ. 1) GOTO80
M=M+1
GO TO 80

40 IF(K-(K/4)*4 .NE. 0) GO TO 50
M=M+1
NFAC(M)=2
K=K/4
50 KT=M  
   \( J = 2 \)
60 IF(MOD(K,J) .NE. 0) GO TO 70
   M=M+1
   NFAC(M)=J
   K=K/J
70 J=((J+1)/2)*2+1
   IF(J .LE. K) GO TO 60
80 IF(KT .EQ. 0) GO TO 100
   J=KT
90 M=M+1
   NFAC(M)=NFAC(J)
   J=J-1
   IF(J .NE. 0) GO TO 90
C COMPUTE FOURIER TRANSFORM
100 SD=RADF/FLOAT(KSPAN)
   CD=2.0*SIN(SD)**2
   SD=SIN(SD+SD)
   KK=1
   I=I+1
   IF(NFAC(I) .NE. 2) GO TO 400
C TRANSFORM FOR FACTOR OF 2 (INCLUDING ROTATION FACTOR)
   KSPAN=KSPAN/2
   K1=KSPAN+2
210 K2=KK+KSPAN
   AK=A(K2)
   BK=B(K2)
   A(K2)=A(KK)-AK
   B(K2)=B(KK)-BK
   A(KK)=A(KK)+AK
   B(KK)=B(KK)+BK
   KK=K2+KSPAN
   IF(KK .LE. NN) GO TO 210
   KK=KK-NN
   IF(KK .LE. JC) GO TO 210
   IF(KK .GT. KSPAN) GO TO 800
220 C1=1.0-CD
   S1=SD
230 K2=KK+KSPAN
   AK=A(KK)-A(K2)
   BK=B(KK)-B(K2)
   A(KK)=A(KK)+A(K2)
   B(KK)=B(KK)+B(K2)
   A(K2)=C1*AK-S1*BK
   B(K2)=S1*AK+C1*BK
   KK=K2+KSPAN
   IF(KK .LT. NT) GO TO 230
   K2=KK-NT
   C1=-C1
   KK=K1-K2
   IF(KK .GT. K2) GO TO 230
${\text{AK}} = C1 - (CD * C1 + SD * S1)$

$S1 = (SD * C1 - CD * S1) + S1$

C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE

C

$C1 = AK$

$C1 = 0.5 / (AK ** 2 + S1 ** 2) + 0.5$

$S1 = C1 * S1$

$C1 = C1 * AK$

$KK = KK + JC$

IF (KK .LT. K2) GO TO 230

$KL = KL + INC + INC$

$KK = (KL - KSPAN) / 2 + JC$

IF (KK .LE. JC + JC) GO TO 220

GO TO 100

C TRANSFORM FOR FACTOR OF 3 (OPTIONAL CODE)

320 $KL = KK + KSPAN$

$K2 = KL + KSPAN$

$AK = A(KK)$

$BK = B(KK)$

$AJ = A(K1) + A(K2)$

$BJ = B(K1) + B(K2)$

$A(KK) = AK + AJ$

$B(KK) = BK + BJ$

$AK = -0.5 * AJ + AK$

$BK = -0.5 * BJ + BK$

$AJ = (A(K1) - A(K2)) * S120$

$BJ = (B(K1) - B(K2)) * S120$

$A(K1) = AK - BJ$

$B(K1) = BK + AJ$

$A(K2) = AK + BJ$

$B(K2) = BK - AJ$

$KK = K2 + KSPAN$

IF (KK .LT. NN) GO TO 320

$KK = KK - NN$

IF (KK .LE. KSPAN) GO TO 320

GO TO 700

C TRANSFORM FOR A FACTOR OF 4

400 IF (NFAC(I) .NE. 4) GO TO 600

$KSPN = KSPAN$

$KSPAN = KSPAN / 4$

410 $C1 = 1.0$

$S1 = 0$

420 $KL = KK + KSPAN$

$K2 = KL + KSPAN$

$K3 = K2 + KSPAN$

$AKP = A(KK) + A(K2)$

$AKM = A(KK) - A(K2)$

$AJP = A(KL) + A(K3)$

$AJM = A(K1) - A(K3)$

$A(KK) = AKP + AJP$

$AJP = AKP - AJP$
BKP = B(KK) + B(K2)
BKM = B(KK) - B(K2)
BJP = B(K1) + B(K3)
BJM = B(K1) - B(K3)
B(KK) = BKP + BJP
BJP = BKP - BJP
IF(ISN .LT. 0) GO TO 450
AKP = AKM - BJM
AKM = AKM + BJM
BKP = BKM + AJM
BKM = BKM - AJM
IF(S1 .EQ. 0.0) GO TO 460
430 A(K1) = AKP*C1 - BKP*S1
B(K1) = AKP*S1 + BKP*C1
A(K2) = AJP*C2 - BJP*S2
B(K2) = AJP*S2 + BJP*C2
A(K3) = AKM*C3 - BKM*S3
B(K3) = AKM*S3 + BKM*C3
KK = K3 + KSPAN
IF(KK .LE. NT) GO TO 420
440 C2 = C1 - (CD*C1 + SD*S1)
S1 = (SD*C1 - CD*S1) + S1
C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
C
C1 = C2
C1 = 0.5/(C2**2 + S1**2) + 0.5
S1 = C1*S1
C1 = C1*C2
C2 = C1**2 - S1**2
S2 = 2.0*C1*S1
C3 = C2*C1 - S2*S1
S3 = C2*S1 + S2*C1
KK = KK - NT + JC
IF(KK .LE. KSPAN) GO TO 420
KK = KK - KSPAN + INC
IF(KK .LE. JC) GO TO 410
IF(KSPAN .EQ. JC) GO TO 800
GO TO 100
450 AKP = AKM + BJM
AKM = AKM - BJM
BKP = BKM + AJM
BKM = BKM - AJM
IF(S1 .NE. 0.0) GO TO 430
460 A(K1) = AKP
B(K1) = BKP
A(K2) = AJP
B(K2) = BJP
A(K3) = AKM
B(K3) = BKM
KK = K3 + KSPAN
IF(KK .LE. NT) GO TO 420
GO TO 440
C TRANSFORM FOR FACTOR OF 5 (OPTIONAL CODE)
510 C2=C72**2-S72**2
     S2=2.0*C72*S72
520 KL=KK+KSPAN
     K2=K1+KSPAN
     K3=K2+KSPAN
     K4=K3+KSPAN
     AKP=A(K1)+A(K4)
     AKM=A(K1)-A(K4)
     BKP=B(K1)+B(K4)
     BKM=B(K1)-B(K4)
     AJP=A(K2)+A(K3)
     AJM=A(K2)-A(K3)
     BJP=B(K2)+B(K3)
     BJM=B(K2)-B(K3)
     AA=A(KK)
     BB=B(KK)
     A(KK)=AA+AKP+AJP
     B(KK)=BB+BKP+BJP
     AK=AKP*C72+AJP*C2+AA
     BK=BKP*C72+BKP*C2+BB
     AJ=AKM*S72+AJM*S2
     BJ=BKM*S72+BJM*S2
     A(K1)=AK-BJ
     A(K4)=AK+BJ
     B(K1)=BK+AJ
     B(K4)=BK-AJ
     AK=AKP*C2+AJP*C72+AA
     BK=BKP*C2+BKP*C72+BB
     AJ=AKM*S2-AJM*S72
     BJ=BKM*S2-BJM*S2
     A(K2)=AK-BJ
     A(K3)=AK+BJ
     B(K2)=BK+AJ
     B(K3)=BK-AJ
     KK=K4+KSPAN
     IF(KK .LT. NN) GO TO 520
     KK=KK-NN
     IF(KK .EQ. KSPAN) GO
     TO 520
800 GO TO 700
C TRANSFORM FOR ODD FACTORS
600 K=NFACT(I)
     KSPAN=KSPAN
     KSPAN=KSPAN/K
     IF(K .EQ. 3) GO TO 320
     IF(K .EQ. 5) GO TO 510
     IF(K .EQ. JF) GO TO 640
     JF=K
     S1=RAD/FLOAT(K)
     C1=COS(S1)
S1=SIN(S1)
IF(JF.GT.MAXF) GO TO 998
CK(JF)=1.0
SK(JF)=0.0
J=1
630 CK(J)=CK(K)*C1+SK(K)*S1
SK(J)=CK(K)*S1-SK(K)*C1
K=K-1
CK(K)=CK(J)
SK(K)=-SK(J)
J=J+1
IF(J.LT.K) GO TO 630

640 K1=KK
K2=KK+KSPAN
AA=A(KK)
BB=B(KK)
AK=AA
BK=BB
J=1
K1=K1+KSPAN
650 K2=K2-KSPAN
J=J+1
AT(J)=A(K1)+A(K2)
AK=AT(J)+AK
BT(J)=B(K1)+B(K2)
BK=BT(J)+BK
J=J+1
AT(J)=A(K1)-A(K2)
BT(J)=B(K1)-B(K2)
K1=K1+KSPAN
IF(K1.LT.K2) GO TO 650
A(KK)=AK
B(KK)=BK
K1=KK
K2=KK+KSPAN
J=1
660 K1=K1+KSPAN
K2=K2-KSPAN
JJ=J
AK=AA
BK=BB
AJ=0.0
BJ=0.0
K=1
670 K=K+1
AK=AT(K)*CK(JJ)+AK
BK=BT(K)*CK(JJ)+BK
K=K+1
AJ=AT(K)*SK(JJ)+AJ
BJ=BT(K)*SK(JJ)+BJ
JJ=JJ+J
IF(JJ.LT.JF) GOTO680
JJ=JJ-JF

680 IF(K.LT.JF) GOTO670
K=JF-J
A(K1)=AK-BJ
B(K1)=BK+AJ
A(K2)=AK+BJ
B(K2)=BK-AJ
J=J+1
IF(J.LT.K) GO TO 660
KK=KK+KSR-m
IF(KK.LT.JF) GOTO 660
KK=KK+J

C MULTIPLY BY ROTATIONAL FACTOR (EXCEPT FOR FACTORS OF 2 AND 4)
700 IF(I.EQ. M) GO TO 800
KK=JC+1

710 C2=1.0-CD
S1=SD

720 C1=C2
S2=S1
KK=KK+KSPAN

730 AK=A(KK)
A(KK)=C2*AK-S2*B(KK)
B(KK)=S2*AK+C2*B(KK)
KK=KK+KSPAN
IF(KK.LT.K) GOTO 730
AK=S1*S2
S2=S1*C2+C1*S2
C2=C1*C2-AK
KK=KK-NI-JKSPAN
IF(KK.LE.KSPAN) GOTO 730
C2=C1-(CD*C1+SD*S1)
S1=S1+(SD*C1-CD*S1)

C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC IS USED, THEY MAY
C BE DELETED
C
C PEIM1rE THE RESULTS TO NORMAL ORDER—DONE IN TWO STAGES
C PERMUTATION FOR SQUARE FACTORS OF N
800 NP(1)=KS
IF(KT.LT.0) GO TO 890
K=KT+KT+1
IF(M.GE.K) GOTO 805
K=K-1
805 J=1
    NP(K+1)=JC
810    NP(J+1)=NP(J)/NFAC(J)
    NP(K)=NP(K+1)*NFAC(J)
    J=J+1
    K=K-1
    IF(J.LT.K) GO TO 810
    K3=NP(K+1)
    KSPAN=NP(2)
    KK=JC+1
    K2=KSPAN+1
    J=1
    IF(N.NE.NTOT) GO TO 850
C PERMUTATION FOR SINGLE-VARIATE TRANSFORM (OPTIONAL CODE)
820 AK=A(KK)
    A(KK)=A(K2)
    A(K2)=AK
    BK=B(KK)
    B(KK)=B(K2)
    B(K2)=BK
    KK=KK+INC
    K2=KSPAN+K2
    IF(K2.LT.KS) GO TO 820
830    K2=K2-NP(J)
    J=J+1
    K2=NP(J+1)+K2
    IF(K2.GT.NP(J)) GO TO 830
    J=1
840 IF(KK.LT.K2) GO TO 820
    KK=KK+INC
    K2=KSPAN+K2
    IF(K2.LT.KS) GO TO 840
    IF(KK.LT.KS) GO TO 830
    JC=K3
    GO TO 890
C PERMUTATION FOR MULTIVARIATE TRANSFORM
850 K=KK+JC
860 AK=A(KK)
    A(KK)=A(K2)
    A(K2)=AK
    BK=B(KK)
    B(KK)=B(K2)
    B(K2)=BK
    KK=KK+INC
    K2=K2+INC
    IF(KK.LT.K) GO TO 860
    KK=KK+KS-JC
    K2=K2+KS-JC
    IF(KK.LT.NT) GO TO 850
    K2=K2-NT+KSPAN
KK = KK - NT + JC
IF(K2 .LT. KS) GO TO 850

870 K2 = K2 - NP(J)
J = J + 1
K2 = NP(J + 1) + K2
IF(K2 .GT. NP(J)) GO TO 870
J = 1

880 IF(KK .LT. K2) GO TO 850
KK = KK + JC
K2 = KSPAN + K2
IF(K2 .LT. KS) GO TO 880
IF(KK .LT. KS) GO TO 870
JC = K3

890 IF(2*KT + 1 .GE. M) GOTO 990
KSPAN = NP(KT + 1)

C PERMUTATION FOR SQUARE-FREE FACTORS OF N

J = M - KT
NFAC(J+1) = 1

900 NFAC(J) = NFAC(J) * NFAC(J+1)
J = J - 1
IF(J .NE. KT) GO TO 900

KT = KT + 1
NN = NFAC(KT) - 1
IF(NN .GT. MAXP) GO TO 998
JJ = 0
J = 0
GO TO 906

902 JJ = JJ - K2
K2 = KK
K = K + 1
KK = NFAC(K)

904 JJ = KK + JJ
IF(JJ .GE. K2) GO TO 902
NP(J) = JJ

906 K2 = NFAC(KT)
K = KT + 1
KK = NFAC(K)
J = J + 1
IF(J .LE. NN) GO TO 904

C DETERMINE THE PERMUTATION CYCLES OF LENGTH GREATER THAN 1

J = 0
GO TO 914

910 K = KK
KK = NP(K)
NP(K) = KK
IF(KK .NE. J) GO TO 910
K3 = KK

914 J = J + 1
KK = NP(J)
IF(KK .LT. 0) GO TO 914
IF(KK .NE. J) GO TO 910
151

NP(J)=-J
IF(J .NE. NN) GO TO 914
MAXF=INC*MAXF

C REORDER A AND B, FOLLOWING THE PERMUTATION CYCLES
GO TO 950

924 J=J-1
IF(NP(J) .LT. 0) GO TO 924
JJ=JC

926 KSPAN=JJ
IF(JJ .LE. MAXF) GOTO 927
KSPAN=MAXF

927 JJ=JJ-KSPAN
K=NP(J)
KK=JC*K+II+JJ
K1=KK+KSPAN
K2=0

928 K2=K2+1
AT(K2)=A(K1)
BT(K2)=B(K1)
K1=K1-INC
IF(K1 .NE. KK) GO TO 928

932 K1=KK+KSPAN
K2=K1-JC*K+(NP(K))
K=-NP(K)

936 A(K1)=A(K2)
B(K1)=B(K2)
K1=K1-INC
K2=K2-INC
IF(K1 .NE. KK) GO TO 936
KK=K2
IF(K .NE. J) GO TO 932
K1=KK+KSPAN
K2=0

940 K2=K2+1
A(K1)=AT(K2)
B(K1)=BT(K2)
K1=K1-INC
IF(K1 .NE. KK) GO TO 940
IF(JJ .NE. 0) GO TO 926
IF(J .NE. 1) GO TO 924

950 J=J+1
NT=NT-KSPAN
II=NT-INC+1
IF(NT .GE. 0) GO TO 924

990 RETURN

C ERROR FINISH, INSUFFICIENT ARRAY STORAGE
998 ISN=0

C PRINT 999
STOP

999 FORMAT(44H0ARRAY BOUNDS EXCEEDED WITHIN SUBROUTINE FFT)
END
SUBROUTINE REALTR(A,B,N,ISN)
C IF ISN=1, THIS SUBROUTINE COMPLET ES THE FOURIER TRANSFORM
C OF 2*N REAL DATA VALUES, WHERE THE ORIGINAL DATA VALUES ARE
C STORED ALTERNATELY IN ARRAYS A AND B, AND ARE FIRST
C TRANSFORMED BY A COMPLEX FOURIER TRANSFORM OF DIMENSION N.
C THE COSINE COEFFICIENTS ARE IN A(1), A(2), ..., A(N+1) AND
C THE SINE COEFFICIENTS ARE IN B(1), B(2), ..., B(N+1).
C A TYPICAL CALLING SEQUENCE IS
C   CALL FFT(A,B,N,N,N,1)
C   CALL REALTR(A,B,N,1)
C THE RESULTS SHOULD BE MULTIPLIED BY 0.5/N TO GIVE THE
C USUAL SCALING OF COEFFICIENTS.
C IF ISN=-1, THE INVERSE TRANSFORMATION IS DONE, THE FIRST STEP
C IN EVALUATING A REAL FOURIER SERIES.
C A TYPICAL CALLING SEQUENCE IS
C   CALL REALTR(A,B,N,-1)
C   CALL FFT(A,B,N,N,N,-1)
C THE RESULTS SHOULD BE MULTIPLIED BY 0.5 TO GIVE THE USUAL
C SCALING, AND THE TIME DOMAIN RESULTS ALTERNATE IN ARRAYS A
C AND B, I.E. A(1), B(1), A(2), B(2), ..., A(N), B(N).
C THE DATA MAY ALTERNATIVELY BE STORED IN A SINGLE COMPLEX
C ARRAY A, THEN THE MAGNITUDE OF ISN CHANGED TO TWO TO
C GIVE THE CORRECT INDEXING INCREMENT AND A(2) USED TO
C PASS THE INITIAL ADDRESS FOR THE SEQUENCE OF IMAGINARY
C VALUES, E.G.
C   CALL FFT(A,A(2),N,N,N,2)
C   CALL REALTR(A,A(2),N,2)
C IN THIS CASE, THE COSINE AND SINE COEFFICIENTS ALTERNATE IN A.
C BY R. C. SINGLETON, STANFORD RESEARCH INSTITUTE, OCT. 1968
DIMENSION A(1), B(1)
REAL IM
INC=IABS(ISN)
NK=N*INC+2
NH=NK/2
SD=2.0*ATAN(1.0)/FLOAT(N)
CD=2.0*SIN(SD)**2
SD=SIN(SD+SD)
SN=0.0
IF(ISN .LT. 0) GO TO 30
CN=1.0
A(NK-1)=A(1)
B(NK-1)=B(1)
10  DO 20 J=1,NH,INC
    K=NK-J
    AA=A(J)+A(K)
    AB=A(J)-A(K)
    BA=B(J)+B(K)
    BB=B(J)-B(K)
    RE=CN*BA+SN*AB
    IM=SN*BA-CN*AB
    B(K)=IM-BB
    20  CONTINUE
B(J)=IM+BB
A(K)=AA-RE
A(J)=AA+RE
AA=CN-(CD*CN+SD*SN)
SN=(SD*CN-CD*SN)+SN
THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
C 20 CN=AA
   CN=0.5/(AA**2+SN**2)+0.5
   SN-CN*SN
20 CN=CN*AA
RETURN
30 CN=-1.0
   SD=-SD
   GO TO 10
END

C PROGRAM MAIN
SUBROUTINE YZCOEF
This routine fills a lookup table with velocity as a function of angle, using a trigonometric expansion with fifteen terms beyond the zero angle velocity. It fills a second lookup table with the coupling parameter delta(v)/v as a function of angle using the same type of trigonometric expansion. The coefficients used are from laser measurements on YZ-Lithium Niobate performed by Anhorn, Engan, and Ronnekleiv (1987 IEEE Ultrasonics Symposium Proceedings, p.279-284)

IMPLICIT REAL*8(A-H,L-Z)
IMPLICIT INTEGER*4(I-K)
REAL*8 YZCOEFF(1:181,1:3),CF(16),CF2(16)
CHARACTER*32 FNAME,FNAME2
PI = 4.0*DATAN(1.0D0)

WRITE(*,*),'Enter name of file to contain velocity data:'
READ(*,10) FNAME
WRITE(*,*),'Enter name of file to contain delta(v)/v data:'
READ(*,10) FNAME2
10 FORMAT(A32)
Input coefficients used in expansion
Velocity coefficients
DATA CF(1),CF(2),CF(3),CF(4),CF(5),CF(6),CF(7),CF(8),CF(9),
* CF(10),CF(11),CF(12),CF(13),CF(14),CF(15),CF(16)/3550.98,
* $-137.83, 42.41, 17.43, 6.21, 2.78, 1.76, 1.20, 0.57, 0.49, 0.27, 0.13,$
* $0.15, 0.04, 0.08, 0.06/$

C Delta(v) coefficients
DATA CF2(1), CF2(2), CF2(3), CF2(4), CF2(5), CF2(6), CF2(7), CF2(8),
* CF2(9), CF2(10), CF2(11), CF2(12), CF2(13), CF2(14), CF2(15), CF2(16)/
* $43.87, 19.76, 5.62, 3.80, .98, .08, .55, .48, .25, .32, .00, -.10, .03, .04,$
* $.07, .07/

C Fill table with values
C Loop 100 sets angle
DO 100 J=1,91
  AJ = J
  SUM = 0.0
  SUM2 = 0.0
  THETA = $-(PI/2.0)+(PI/180.0)\times (AJ-1.)$
C Loop 50 calculates the speed at this angle
DO 50 K=0,15
  AK = K
  TERM = CF(K+1) \times \text{DCOS}(2.0\times AK \times \text{THETA})
  TERM2 = CF2(K+1) \times \text{DCOS}(2.0\times AK \times \text{THETA})
  SUM = SUM + TERM
  SUM2 = SUM2 + TERM2
50 CONTINUE
C Since DCOS is symmetric about \theta=0, we can assign this
C value to two angles
YZCOEFF(J,1) = SUM
YZCOEFF(181-J,1) = SUM
C Calculate delta(v)/v and assign to appropriate angles
YZCOEFF(J,3) = SUM2/SUM
YZCOEFF(181-J,3) = SUM2/SUM
100 CONTINUE
C Calculate Power Flow Angle
C Assume power flow angle is zero at +90 and -90 degrees.
YZCOEFF(1,2) = 0.0
YZCOEFF(181,2) = 0.0
C Calculate derivative and PFA (power flow angle)
DO 200 J=2,180
  DVEL = (YZCOEFF(J+1,1)-YZCOEFF(J-1,1))/(PI/90.)
  PFA = D\tan2(DVEL, YZCOEFF(J,1))
  YZCOEFF(J,2) = PFA
200 CONTINUE
C Write to output file
OPEN(9,FILE=FNAME)
OPEN(10, FILE=FNAME2)
DO 300 I = 1, 181
   AI = I
   WRITE(9,310) YZCOEFF(I,1), YZCOEFF(I,2)
   WRITE(10,320) (-91.+AI), YZCOEFF(I,3)
300 CONTINUE
310 FORMAT(E14.8,1X,E14.8)
320 FORMAT(E14.8,1X,E14.8)
C
C RETURN
END
REFERENCES


