APPLICATION OF THE BOUNDARY INTEGRAL EQUATION METHOD
TO A
DISCONTINUITY IN BEDROCK

by

Ernest Heymsfield

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1995

UMI Number: 9605602

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

August 15, 1995
Date

Prof. Carl J. Costantino,
Chair of Examining Committee

August 15, 1995
Date

Dean Gerard G. Lowen,
Executive Officer

Prof. Charles A. Miller

Prof. Mumtaz Kassir

Prof. Ali M. Sadegh

Prof. Aspasia Zerva
Drexel University, Philadelphia

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

APPLICATION OF THE BOUNDARY INTEGRAL EQUATION METHOD

TO A

DISCONTINUITY IN BEDROCK

by

Ernest Heymsfield

Adviser: Prof. Carl J. Costantino

The objective of this thesis is to investigate the significance that an inhomogeneity in a rock half-space has on surface amplification. The particular soil-bedrock configuration considered consists of a homogeneous soil layer on a half-space. Included in the half-space is an embedded semi-infinite rock intrusion with its upper surface abutting the soil layer. Materials are considered viscoelastic except for the portion of the half-space below the embedded rock layer. The importance of the rock inclusion is determined by investigating the variation of the amplification of body waves along the ground surface for a range of frequencies and incidence angles. The results of a parametric study are presented indicating the sensitivity of the amplification function to the rock inclusion's thickness and shear wave velocity. Comparisons are made between the results of the parametric study and the standard one-dimensional analysis considering the two soil-rock profiles taken at a far horizontal distance away from the scattering boundary. These comparisons between the one-dimensional and two-dimensional solutions are summarized in scattering limit plots. These plots indicate the horizontal distance from the bedrock discontinuity in which the two-dimensional and one-dimensional surface amplifications are similar. Therefore, limits can be determined for this particular soil-rock configuration within which a one-dimensional solution is deemed inappropriate and therefore requires a two-dimensional solution.

The boundary element method is used for the two-dimensional solution. Included in the investigation was the development of two boundary element codes, one for surface amplification of anti-plane motion and a second for the surface amplification of in-plane motion. Of special interest in the development of the codes is the correction due to the truncation of the infinite regions involved in soil profiles.

Acknowledgements

Throughout the years that I have been involved in the Doctoral program at the City College of New York I have experienced God's provision of love, strength, guidance, and people to help me on my way. I'm thankful to God for allowing me to accomplish this goal. I can say with the psalmist:

"Let everything that has breath praise the LORD. Praise the LORD."
Psalm 150:6; Holy Bible (New International Version)

Many people have played an important role in my life to allow for this accomplishment. I'm especially appreciative of my grandmother, mother, wife, and two children. My grandmother and mother raised me to believe in my dreams and that anything is possible. For my wife of twelve years who made my dreams reality; for her love and friendship. And for my children: Grace Anna, and Christian Lauritz, who were a motivating factor to get the work done; I hope this accomplishment will better their lives.

For the last eight years, I have worked for the Earthquake Research Center under the supervision of Professors Carl J. Costantino and Charles A. Miller. I'm appreciative of my mentor Prof. Carl J. Costantino for compelling me to reach my potential and his pursuit for excellence. For Prof. Charles A. Miller, for allowing me to enter the Doctoral program and for what I have learned from him. I hope I have gained a small portion of the wealth of knowledge of these two men.

Being at City College I was fortunate to meet Prof. Aspasia Zerva whose enthusiasm and early encouragement led me to pursue earthquake engineering, a field of study that I enjoy and hope to make some contribution to.

As a young engineering graduate, I was fortunate to have as supervisors Mr. Walter Timoshenko and Mr. George Timoshenko who trained me to put theory to practice and become an engineer. Their encouragement in the initial period of the Ph. D. program was very helpful.

It has been good to have a fellow sympathizer in Mr. David E. Manuel who also is a Doctoral student. I'm appreciative of his encouragement and prayers. I hope that he will finish his studies soon.

Many others have also played a significant part in this process and to those I also say thank you.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

During a seismic event, the motion felt at the ground surface varies with both time and distance from the fault. A major cause to this variance is the geologic configuration between the source of the earthquake and the ground surface. It is this dependency on the geophysical characteristics of an area with which this thesis is concerned. An example of such dependency occurred in the Mexico City earthquake where large centralized damaged was attributed to the alluvium deposit on which the city is built (Eshraghi and Dravinski, 1989). The alluvium soil deposit at this site created large amplification of the earthquake motion in the city area, 220 miles away from the epicenter of the earthquake, while minimal amplification occurred in areas closer to the epicenter. Other earthquakes have likewise shown a dependency of a site to an area's geologic description. (Sanchez-Sesma, 1987).

When an earthquake occurs, there is a release of energy at the fault rupture originating at the focus (Figure 1-1). A portion of this energy is translated into seismic motion which propagates through the earth in the form of compression (P) and shear (SH and/or SV) waves. These waves represent body waves in which P and SV waves are in-plane waves while SH waves are out-of-plane (anti-plane) waves. At geologic discontinuities, seismic waves reflect and refract generating new waves. When seismic waves are diffracted at the surface due to irregularities at the ground surface, surface waves, Rayleigh (plane) and Love (anti-plane) waves, are created. The motion at any point in the soil profile is the cumulative effect of these body and surface waves. Because surface motion is the effect of wave motion propagating from the source of an earthquake to the surface, seismic motion at the ground surface is dependent on the soil and bedrock description of the entire area.

As seismic waves propagate through the earth's near surface geology, earthquake motion is modified. The governing equation for this propagation of earthquake motion is

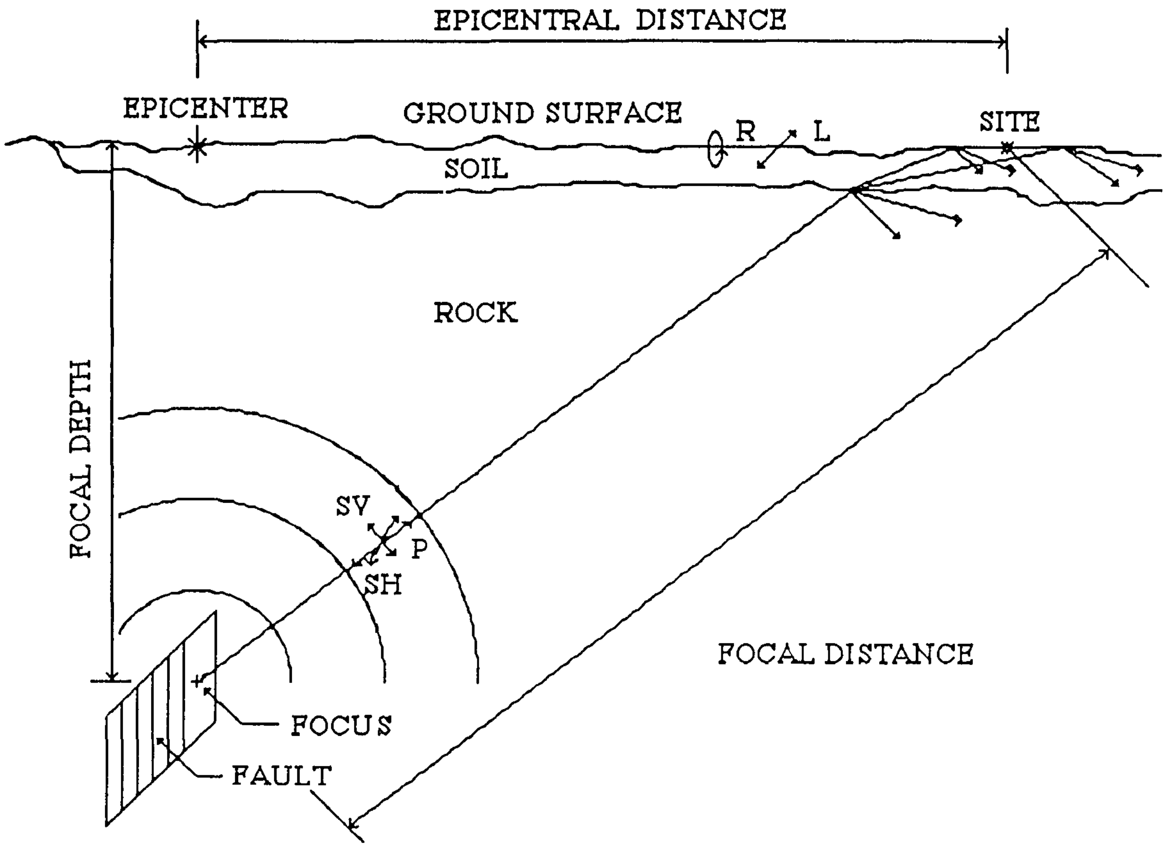


Figure 1-1: Earthquake Motion

the wave equation. Exact solutions for wave propagation problems are generally limited to geometries where separation of variables can be used and where strains are small enough such that linear behavior can reasonably be assumed. One exact solution for soil amplification is the one-dimensional solution which assumes that the seismic motion propagates through horizontal layers of homogeneous and isotropic material (Roesset, 1969) . The one-dimensional analysis has been found to yield acceptable results at sites where these assumptions are satisfied and in other situations for waves of low frequency. For high frequencies, wave scattering due to nonuniformities of the soil profile of a site is significant and a two-dimensional analysis is generally required. For two-dimensional site effects, various geometries have been investigated. Solutions for dipping layers representing soil deposits have been obtained for surface amplification numerically by the source method (Dravinski, 1982a: 1982b), wave functions (Eshraghi and Dravinski, 1989a: 1989b), and boundary element method (Dominguez and Abascal, 1989). Analytic solutions for the amplification of an incident SH wave are available for the cases of a semi-cylindrical shape valley (Trifunac, 1971) and for a semi-elliptical alluvial valley (Wong and Trifunac, 1974). Vogt studied the effects of a canyon embedded in soil layers bounded by rock (Vogt et al., 1988). Cavities and surface irregularities have been investigated (Altay, 1986) and this work extended to study inhomogeneities in soil layers for out-of-plane motion (Hadley et al., 1989). Although much work has been done in studying the effects of soil layer geometry and properties, sufficient work has yet to be done on the effects of bedrock geometry and properties on earthquake motion.

The purpose of this thesis is to investigate the effect that a particular type of bedrock discontinuity has on the amplification of seismic motion due to various incident wave types. The soil-bedrock configuration studied is shown in Figure 1-2 with the soil and bedrock properties given in Table 1-A. Figure 1-2 shows a bedrock half-space with an embedded bedrock layer overlaid with a homogeneous soil layer. The soil and bedrock

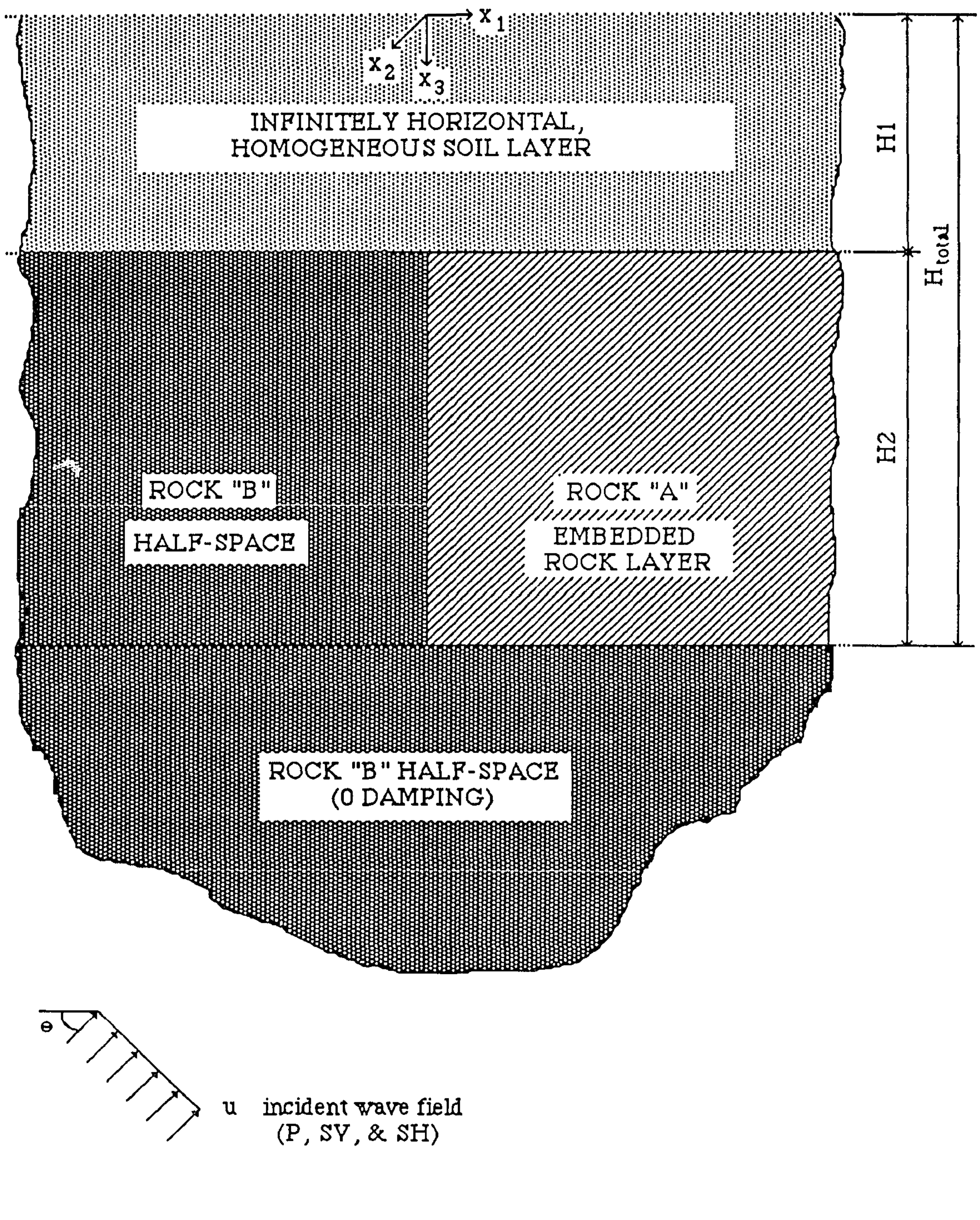


Figure 1-2: Soil-Bedrock Geometry

SOIL PROPERTIES

POISSON RATIO (ν_s)	0.25
DAMPING (β_s)	5.00 %

ROCK PROPERTIES

	ROCK "A"	ROCK "B"
UNIT WT. RATIO (γ_r / γ_s)	1.32	1.32
POISSON RATIO (ν_r)	0.25	0.25
DAMPING (β_r)	2.00 %	1.00%

Table 1-A: Bedrock & Soil Properties

properties in Table 1-A represent typical values for sand and limestone rock.

As a result of strong motion, stress-strain relationships in the soil and rock are non-linear. In order to incorporate this non-linear behavior, an equivalent linear model is used with an adjusted shear modulus and damping (Idriss & Seed, 1968). The adjusted shear modulus is an average shear modulus during cyclic loading. The energy loss incurred by the system during a complete stress-strain cycle is included in the damping term. This is done by setting the energy loss during the cyclic loop equal to the energy loss of the equivalent linear viscous system. Since the damping is related to the hysteresis loop, it is often called hysteretic damping. Although hysteretic damping is a result of friction loss due to harmonic loading, it has been found to be insensitive to the frequency of the loading in the frequency range of interest. Therefore, this type of damping is sometimes called "rate-independent linear damping" (Chopra, 1995). As the induced maximum strain in a material increases, the nonlinear effects of the material increase resulting in the equivalent shear modulus decreasing and the equivalent damping increasing. Each material in Figure 1-2, except for the rock half-space below H_{total}, incorporates hysteretic damping. In this analysis, the shear velocities and damping used represent reduced values and are kept constant. The exception for the rock half-space to be elastic below H_{total} is required for the numerical correction factor used for the half-space. Typically, rock outcrop motions are defined at the top of the rock half-space. Therefore, the assumption of zero damping in the rock half-space is not considered significant for this problem.

Amplification of seismic motion is dependent on the wave type and angle of incidence. Incident wave motions are considered planar and cases for body waves, compression (P) and shear waves (SH & SV), are investigated. In order to determine the significance of the rock properties of the embedded rock layer and embedment depth, H2, a parametric study is made which summarizes the soil amplification along the ground surface as a function of frequency. The analysis includes each of the different body wave types for the various incident angles. The parametric study is accomplished by varying the half-space angle of incidence, the shear velocity ratio of the inclusion, and thickness ratio of the inclusion. This is all done for each of the different body wave types, (SH, SV, and P), and for a suite of dimensionless frequencies (0.5, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.5, and 3.0). These dimensionless frequencies are equal to $(\omega * H1)/Cs_{\text{soil layer}}$. For a soil layer with a shear velocity of 1,000 fps and a thickness of 20', the dimensionless frequencies correspond to actual frequencies in the range of 4.0 and 23.9 cps. Incident angles of 90, 75, and 60 degrees are used for each of the wave types. Shear velocity ratios of rock "A" to soil of 2.5 and 5 are used for the in-plane study and 1, 2.5, and 5 are used for the anti-plane motion study. The shear velocity of rock "B" to soil is kept constant at 10 . The ratio of embedded rock thickness to soil thickness varies from 1 to 5 (1, 2, &5) for anti-plane motion. For in-plane motion, two thickness ratios, 1 and 2, are used. Final results are presented as surface amplifications in the area above the bedrock discontinuity.

Comparison is made between the two-dimensional results and one-dimensional results by examining where the two-dimensional results approach the one-dimensional results along the surface boundary. The end-points of the region where the results are within 10% are defined in this thesis as the scattering limits. Therefore, the scattering limits define the limits of the additional scattering due to the half-space inhomogeneity and is the horizontal distance from the bedrock discontinuity to where the two-dimensional results do not significantly differ from the one-dimensional results. Scattering limits are plotted versus dimensionless frequency, $(\omega H1/Cs_{soil})$, for each combination of shear velocity and thickness ratio .

The boundary element method is used in the two-dimensional study to solve for the amplification of earthquake motion. The method is chosen to be used because of its inherent advantages:

(1) The boundary element method accurately describes the topography and soil configuration of the problem and allows for the inclusion of hysteretic damping.

(2) The method reduces the domain problem to a boundary integral and therefore requires only the discretization of the boundary of the domain.

And

(3) The method automatically satisfies the radiation condition through influence functions used which alleviates the need to model non-reflecting boundaries.

In the following chapters, the problem of a specific type of bedrock discontinuity is investigated. In Chapter 2 the wave equation is derived in the frequency domain. In order to have an initial basis of the importance of bedrock characteristics, a parametric study is performed using the one-dimensional analysis. Chapter 3 includes the formulation for the one-dimensional analysis and the results of the one-dimensional analysis are given in Chapter 4. These amplification curves from the one-dimensional analysis illustrate the significance of the soil-rock shear wave velocity ratio and depth ratio of the inclusion and establish an initial basis for the two-dimensional study . Because the boundary element method is used for the two dimensional analysis, the boundary element method is described and developed in Chapters 5-9 for application to soil amplification problems. The validity of the codes is proved in Chapter 10. Results of the surface amplification due to an unit incident body wave are given in Chapter 11. Comparison between the two-dimensional analysis and the one-dimensional analysis are displayed by scattering limit plots that are also given in Chapter 11. These plots of scattering limits show the significance of the scattering effect of the embedded layer and the variance of the two-dimensional solution from the one-dimensional solution. The computer codes used to generate the results and sample input files for the codes are included in the appendix.

Chapter 2
Wave Equation

The wave equation is developed from Newton's second law ($\Sigma F=ma$). The equations of motion acting on a infinitesimal element of a medium in each direction can be written in indicial form as:

$$\sigma_{ij,j} + f_i = \rho \ddot{u}_i \tag{2.1}$$

f_i = body force
ρ = density

where repeated indices indicate summation.

The stress-strain relation for an elastic material is:

$$\sigma_{ij} = \lambda\, \delta_{ij}\, \varepsilon_{mm} + 2\mu\varepsilon_{ij} \tag{2.2}$$

$\lambda = \mu(\frac{2\nu}{(1-2\nu)})$

λ = elastic constant

μ = shear modulus

ν = Poisson ratio

Substituting displacements for strain, the stress-displacement relation becomes:

$$\sigma_{ij} = \lambda\, \delta_{ij}\, u_{m,m} + \mu(u_{i,j} + u_{j,i}) \tag{2.3}$$

and the derivative of stress with respect to coordinate j:

$$\sigma_{ij,j} = (\lambda + \mu)\ \ u_{j,ji} + \mu\ \ u_{i,jj} \tag{2.4}$$

The governing wave equation is then derived by substituting equation (2.4) into equation (2.1). This leads to Navier's equation which is Newton's law in terms of displacements.

$$(\lambda + \mu)\, u_{j,ij} + \mu u_{i,jj} + f_i = \rho \ddot{u}_i \tag{2.5}$$

This is the governing equation for wave propagation in an elastic material. Since our interest is in the displacement variation from the static position, the body forces, f_i, will be excluded in the following equations. In order to consider damping, linear viscous damping terms are included through compression and shear viscous damping terms. To determine which term in equation 2.5 to apply compression viscous damping and which term to apply shear damping, equation (2.5) is rewritten in a form of a compression wave and shear wave by adding and subtracting $u_{j,ji}$ to the left side of equation (2.5). Equation 2.5 then becomes:

$$(\lambda + 2\mu)\; u_{j,ji} + \mu\, (u_{i,jj} - u_{j,ji}) = \rho \ddot{u}_i \tag{2.6}$$

Now it can be seen that the first term on the left side of equation (2.6) is dependent on the compression wave velocity while the second term on the shear wave velocity. Therefore, the compression wave viscosity should be applied to the first term while the shear wave viscosity applied to the second term. The stress-displacement relation can be written as:

$$(\lambda + 2\mu)u_{j,ji} + \mu(u_{i,jj} - u_{j,ji}) + \eta_p \frac{\partial u_{j,ji}}{\partial t} + \eta_s \frac{\partial (u_{i,jj} - u_{j,ji})}{\partial t} = \rho \ddot{u}_i \tag{2.7}$$

η_p = compression wave viscosity

η_s = shear wave viscosity

The work in this thesis is done in the frequency domain. In order to develop the wave equation in the frequency domain, a Fourier transform is used where the Fourier transform of f(t) is defined as:

$$\bar{f}(\omega) = F\{f(t)\} = \int_0^{\infty} f(t)\, e^{+i\omega t} dt \tag{2.8}$$

t = time

ω = frequency

Therefore, taking the Fourier transform of equation (2.7), the Fourier transform of the wave equation for a viscoelastic material is defined as:

$$F\left[(\lambda + 2\mu)u_{j,ji} + \mu(u_{i,jj} - u_{j,ji}) + \eta_p \frac{\partial u_{j,ji}}{\partial t} + \eta_s \frac{\partial(u_{i,jj} - u_{j,ji})}{\partial t}\right] = F[\,\rho\ddot{u}_i\,] \tag{2.9}$$

Taking the Fourier transform by applying equation (2.8) to both sides of equation (2.9) leads to:

$$(\lambda + 2\mu - i\omega\eta_P)\, F[u_{j,ji}] + (\mu - i\omega\eta_S)\, F\,(u_{i,jj} - u_{j,ji}) = -\,\rho\omega^2 F[u_i] \tag{2.10}$$

or

$$(\lambda + 2\mu - i\omega\eta_P)\, \bar{u}_{j,ji} + (\mu - i\omega\eta_S)\, (\bar{u}_{i,jj} - \bar{u}_{j,ji}) = -\,\rho\omega^2 \bar{u}_i \tag{2.11}$$

Material viscosity effects are represented by means of a hysteretic damping model. The hysteretic damping ratio is related to linear viscous damping by:

$$\eta_P = \frac{2\beta_P(\lambda + 2\mu)}{\omega} \quad ; \quad \eta_S = \frac{2\beta_S \mu}{\omega} \tag{2.12}$$

β_P = compression wave hysteretic damping

β_S = shear wave hysteretic damping

and equation (2.11) can be rewritten as:

$$(\lambda + 2\mu)(1 - i2\beta_P)\,\bar{u}_{j,ji} + \mu\,(1 - i2\beta_S)\,(\bar{u}_{i,jj} - \bar{u}_{j,ji}) = -\rho\omega^2\bar{u}_i \tag{2.13}$$

This equation is further simplified by substituting complex terms for the elastic Lame constants. The equation now becomes:

$$(\lambda + 2\mu)^*\,\bar{u}_{j,ji} + \mu^*(\bar{u}_{i,jj} - \bar{u}_{j,ji}) = -\rho\omega^2\bar{u}_i \tag{2.14}$$

where

$$(\lambda + 2\mu)^* = (\lambda + 2\mu)(1 - i2\beta_P)$$

$$\mu^* = \mu\,(1 - i2\beta_S)$$

As a simplification to the analysis in this work, the hysteretic damping for the compression wave and shear wave are considered equal and both set to β .

Chapter 3
One-Dimensional Solution

A one-dimensional study is carried out to determine the significant characteristics of the soil-rock configuration of Figure 1-2. The one-dimensional solution is based on the assumption that each layer of the studied soil profile is homogeneous, horizontal, and infinite in the x_1 and x_2 directions (Figure 3-1). In the following, one-dimensional solutions are developed for in-plane (P,SV) and anti-plane (SH) motions due to a unit incident wave in the frequency domain. The equations are written for layer "j" and therefore a repeated "j" index does not imply summation. In Chapter 4, surface displacement plots due to a unit incident wave are generated for each of the possible soil columns of Figure 1-2. These plots serve as a preliminary basis to determine the dependency of surface displacements to the discontinuity in the rock half-space and what characteristics may be significant for the two-dimensional study.

3.1: Anti-Plane Motion

An anti-plane shear wave (SH) reflects and refracts at a boundary into anti-plane shear waves resulting in no mode conversion. Within each layer of Figure 3-1, displacement in the x_2 direction, u_2, is the result of an incoming wave in the $-x_3$ direction, Ash_j, and an outgoing wave in the $+x_3$ direction, Bsh_j.

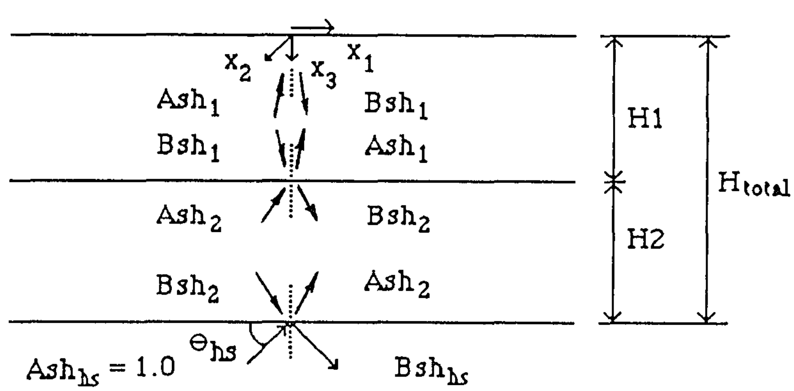


Figure 3-1: One-Dimensional Anti-Plane Wave Motion

For anti-plane motion only displacement , u_2, in the x_2 direction will occur and equation (2.14) simplifies to:

$$\mu^* \bar{u}_{2,nn}(x,\omega), + \rho\omega^2 \bar{u}_2(x,\omega) = 0 \tag{3.1}$$

Further simplifying equation (3.1) to include the shear velocity results in :

$$\bar{u}_{2,nn}(x,\omega) + \left(\frac{\omega^2}{C_s^*}\right) \bar{u}_2(x,\omega) = 0 \tag{3.2}$$

where

$\bar{u}_2(x,\omega)$ = displacement in x_2 direction

Cs^* = complex shear velocity

Within each layer "j", the anti-plane wave equation, equation (3.2), is satisfied by the displacement :

$$(\bar{u}_2)_j\,(x,\omega) = [Ash_j \exp\{-i\omega/Cs_j^* \sin(p_j)\, x_3\} + Bsh_j \exp\{+i\omega/Cs_j^* \sin(p_j)\, x_3\}] \exp\{+i\omega/Cs_j^* \cos(p_j)\, x_1\} \tag{3.3}$$

where

Ash_j = magnitude of incoming wave within layer j

Bsh_j = magnitude of outgoing wave within layer j

Cs_j^* = complex shear wave velocity of layer j (includes hysteretic damping)

μ_j^* = shear modulus of layer j (includes hysteretic damping)

p_j = the incidence angle of the SH wave (θ_j) for the case of zero damping, is complex otherwise

ω = circular frequency

The shear stress within each layer is determined from the constitutive relationship, $(\bar{\sigma}_{32})_j = \mu_j^* \, \partial(u_2)_j / \partial x_3$, and results in:

$$(\bar{\sigma}_{32})_j (x,\omega) =$$

$$i\mu_j^* \, \omega/Cs_j^* (\text{sine}(p_j))[-Ash_j \exp\{-i\omega/Cs_j^* \text{sine}(p_j)\, x_3\} + Bsh_j \exp\{+i\omega/Cs_j^* \text{sine}(p_j)\, x_3\}] \exp\{+i\omega/Cs_j^* \cos(p_j)\, x_1\} \tag{3.4}$$

At each soil layer interface, continuity of displacement and stress is enforced. Examining equations (3.3) and (3.4), it can be concluded that in order to satisfy welded conditions along a layer interface the function $(\omega/Cs_j^*)\cos(p_j)$ must be equal for each of the layers and half-space. This function is called the wave number and is represented by k. Since for a given soil profile k is constant for each frequency, p_j can be determined for each soil layer from the known half-space angle of incidence:

$$\frac{\cos p_1}{Cs_1^*} = \frac{\cos p_2}{Cs_2^*} = \frac{\cos p_{hs}}{Cs_{hs}^*} \tag{3.5}$$

Equation (3.5) indicates that p_j is frequency independent.

Stress free conditions at the surface result in the incoming wave coefficient, Ash_1, being equal to the reflected wave coefficient, Bsh_1. Incident motion is introduced by setting the half-space incident wave coefficient, Ash_{hs}, equal to 1.0 . Displacement, equation (3.3), and shear stress, equation (3.4), are simplified and the shear stress for layer "j" is normalized by multiplying by the thickness of the top soil layer and dividing by the complex shear modulus of the layer that it pertains to:

$$(\bar{u}_2)_j(x,\omega) = [Ash_j \exp\{-ikt_j x_3\} + Bsh_j \exp\{+ikt_j x_3\}] \exp\{+ikx_1\} \tag{3.6}$$

$$\frac{(\bar{\sigma}_{32})_j(x,\omega)\,H1}{\mu_j^*} = i\left(\frac{\omega H1}{Cs_{A1}^*}\right)\left(\frac{m_{xj}t_j}{\left(\frac{Cs_j^*}{Cs_{A1}^*}\right)}\right)[-Ash_j \exp\{-ikt_j x_3\} + Bsh_j \exp\{+ikt_j x_3\}] \exp\{+ikx_1\} \tag{3.7}$$

$$\frac{(\bar{\sigma}_{12})_j(x,\omega)\,H1}{\mu_j^*} = i\left(\frac{\omega H1}{Cs_{A1}^*}\right)\left(\frac{m_{x1}}{\left(\frac{Cs_1^*}{Cs_{A1}^*}\right)}\right)[Ash_j \exp\{-ikt_j x_3\} + Bsh_j \exp\{+ikt_j x_3\}] \exp\{+ikx_1\} \tag{3.8}$$

where

Cs_{A1}^* = shear wave velocity of area 1

$m_{xj} = \cos(p_j)$

$t_j = \sqrt{\frac{1}{m_{xj}^2} - 1}$

From equations (3.6), (3.7), and (3.8), it is seen that the displacement and stresses are functions of x_3 and of $\exp(ikx_1)$ in the x_1 direction. The latter term indicates the propagation in the x_1 direction. For 0 damping, t_j represents the tangent of the angle of incidence within layer j, however for layers which include damping, the angle of incidence is not obvious. In order to determine the angle of incidence within the damped soil layers, the exponential terms of equation (3.3) are rewritten in terms of real and imaginary numbers (Wolf, 1985). Since the half-space is considered elastic, the wave number is real and only t_j is complex:

$$(\bar{u}_2)_j(x,\omega) =$$
$$\left[Ash_j \exp\left\{-ik\left(Re(t_j) + i\, Im(t_j)\right) x_3\right\} + Bsh_j \exp\left\{+ik\left(Re(t_j) + i\, Im(t_j)\right) x_3\right\}\right] \exp\{+ikx_1\}$$

(3.9)

Carrying out the multiplication of the exponential terms results in an incident wave which increases in magnitude with increasing x_3 and a reflected wave which decreases with increasing x_3.

$$(\bar{u}_2)_j(x,\omega) =$$
$$\begin{bmatrix} Ash_j \exp\left\{-ik\left(Re(t_j)\right) x_3\right\} \exp\left\{+k\left(Im(t_j)\right)\ x_3\right\} \\ + Bsh_j \exp\left\{+ik\left(Re(t_j)\right) x_3\right\} \exp\left\{-k\left(Im(t_j)\right)\ x_3\right\} \end{bmatrix} \exp\{+ikx_1\}$$

(3.10)

It can be concluded from equation (3.10) that the angle of incidence is equal to the arc tangent of the real part of t_j .

Considering the surface boundary condition and welded condition at the layer interfaces, equations (3.6) and (3.7) can be put into matrix form with the wave coefficients as unknowns and the incident half-space wave coefficient set to 1.0 . After solving for the wave coefficients of the soil layers, wave displacements and stresses can be determined using equations (3.6), (3.7), and (3.8). Equation (3.11) is the matrix equation used to solve the problem of a soil profile consisting of two-layers on a half-space.

$$\begin{bmatrix} +1 & -1 & 0 & 0 & 0 \\ +e^{-ikt_1z_1} & +e^{+ikt_1z_1} & -e^{-ikt_2z_1} & -e^{+ikt_2z_1} & 0 \\ -e^{-ikt_1z_1} & +e^{+ikt_1z_1} & +\xi_1\frac{t_2}{t_1}e^{-ikt_2z_1} & -\xi_1\frac{t_2}{t_1}e^{+ikt_2z_1} & 0 \\ 0 & 0 & +e^{-ikt_2z_2} & +e^{+ikt_2z_2} & -e^{+ikt_{hs}z_2} \\ 0 & 0 & -e^{-ikt_2z_2} & +e^{+ikt_2z_2} & -\xi_2\frac{t_{hs}}{t_2}e^{+ikt_{hs}z_2} \end{bmatrix} \begin{Bmatrix} Ash_1 \\ Bsh_1 \\ Ash_2 \\ Bsh_2 \\ Bsh_{hs} \end{Bmatrix}$$

$$= \begin{Bmatrix} 0 \\ 0 \\ 0 \\ +Ash_{hs}e^{-ikt_{hs}z_2} \\ -\xi_2\frac{t_{hs}}{t_2}Ash_{hs}e^{-ikt_{hs}z_2} \end{Bmatrix} \tag{3.11}$$

where

z_j = depth to base of layer j

$$\xi_j = \left(\frac{Cs_{j+1}}{Cs_j}\right)^2 \frac{\gamma_{j+1}}{\gamma_j}$$

The rock outcrop motion is calculated at $x_3 = H_{total}$ and assumes that there is no material above this level. Therefore, for outcrop motion the incident and reflected wave magnitudes are equal and the rock outcrop displacement is equal to:

$$(\bar{u}_2)_{outcrop}(x,\omega) = 2.0 * Ash_{hs} \exp(-ikH_{total}) \qquad (3.12)$$

3.2: In-Plane Motion

An incoming in-plane wave reflects and refracts at a boundary into in-plane waves. For the case of vertical incidence, the wave type remains unchanged. However, for non-vertical incident waves mode conversion occurs in which an incoming wave reflects and refracts into P and SV waves. Within each layer displacement is the summation of the displacement due to the incoming, $-x_3$ direction, P and SV waves , and the outgoing, $+x_3$ direction, P and SV waves (Figure 3-2).

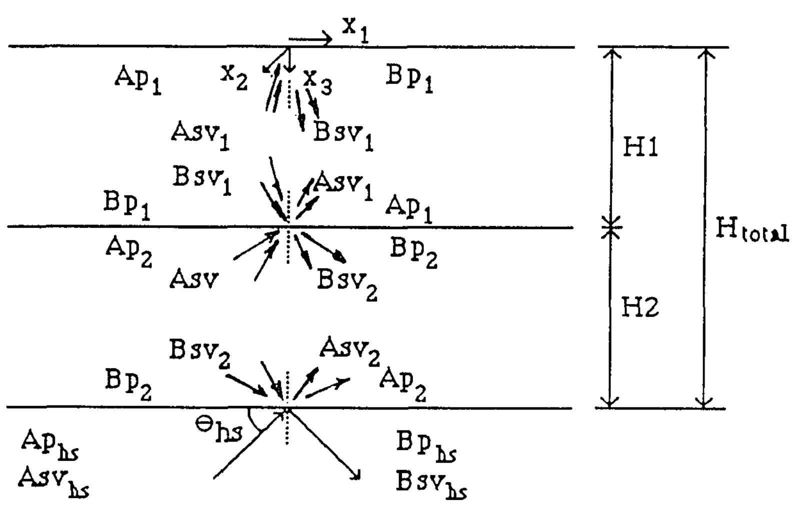


Figure 3-2: One-Dimensional In-Plane Wave Motion

Within each layer "j" the wave equation is satisfied by the displacement equations:

$$(\bar{u}_1)_j(x,\omega) =$$
$$[l_{xj}\{Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\} + \sqrt{1-m_{xj}^2}\ \{-Asv_j\exp(-ikt_jx_3) + Bsv_j\exp(+ikt_jx_3)\}]$$
$$*\exp(+ikx_1)$$

(3.13)

$$(\bar{u}_3)_j\ (x,\omega) =$$
$$[\sqrt{1-l_{xj}^2}\ \{-Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\} + m_{xj}\ \{-Asv_j\exp(-ikt_jx_3) - Bsv_j\exp(+ikt_jx_3)\}]$$
$$*\exp(+ikx_1)$$

(3.14)

The stress equations are generated by applying the constitutive relations to equations (3.13) and (3.14) which result in:

$$(\bar{\sigma}_{11})_j(x,\omega) = +ik\mu_j^*[\{+Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}\left\{\frac{2}{l_{xj}}(l_{xj}-1) + \frac{l_{xj}}{m_{xj}^2}\right\} -$$
$$\{+Asv_j\exp(-ikt_jx_3) - Bsv_j\exp(+ikt_jx_3)\}(2\sqrt{1-m_{xj}^2})]\ \exp(+ikx_1) \qquad (3.15)$$

$$(\bar{\sigma}_{33})_j(x,\omega) = +ik\mu_j^*[\{+Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}l_{xj}(\frac{1}{m_{xj}^2} - 2)+$$
$$\{+Asv_j\exp(-ikt_jx_3) - Bsv_j\exp(+ikt_jx_3)\}(2\sqrt{1-m_{xj}^2})]\ \exp(+ikx_1) \qquad (3.16)$$

$$(\bar{\sigma}_{31})_j(x,\omega) = +ik\mu_j^*[\{-Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}l_{xj}(2\sqrt{1-l_{xj}^2})+$$
$$\{+Asv_j\exp(-ikt_jx_3) + Bsv_j\exp(+ikt_jx_3)\}(\frac{1-2m_{xj}^2}{m_{xj}})]\exp(+ikx_1) \tag{3.17}$$

where

$(\bar{u}_k)_j(x,\omega)$ = displacement in k direction within layer "j"

Ap_j = incoming compression wave coefficient within layer "j"

Bp_j = outgoing compression wave coefficient within layer "j"

Asv_j= incoming shear wave coefficient within layer "j"

Bsv_j= outgoing shear wave coefficient within layer "j"

$$s_j = \sqrt{\frac{1}{l_{xj}^2} - 1}$$

$$t_j = \sqrt{\frac{1}{m_{xj}^2} - 1}$$

μ_j^* = complex shear modulus of layer "j"

ω = frequency

As in the case of anti-plane motion, the wave number is constant for each frequency. Therefore, from the prescribed half-space angle of incidence, l_{xj} and m_{xj} can be calculated from equation (3.18):

$$\frac{Cs_{hs}/Cs_1}{m_{xhs}} = \frac{Cs_j^*/Cs_1}{m_{xj}} = \frac{Cp_j^*/Cs_1}{l_{xj}} = \frac{Cp_{hs}/Cp_1}{l_{xhs}} \tag{3.18}$$

Displacement and stress solutions are again functions of x_3 and of $\exp(ikx_1)$. At each soil layer interface continuity of displacement and stress is enforced, and stress free conditions are set at the ground surface. The development of the solution for in-plane wave motion is more complicated than the out-of-plane case due to the wave mode conversion which occurs at the boundary interfaces. Stresses are normalized through multiplying by the thickness of the top soil layer and dividing by the complex shear modulus of the layer. Normalized stresses are shown in equations (3.19-3.21)

$$\frac{(\bar{\sigma}_{11})_j(x,\omega)\,H1}{\mu_j^*} =$$

$$+i\exp(+ikx_1)[\{+Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}(\frac{\omega H1}{Cs_{A1}^*})[\frac{1}{(\frac{Cs_j^*}{Cs_{A1}^*})}\frac{Cp_j^*}{Cs_j^*} + \frac{2(l_{xj}-1)}{(\frac{Cp_j^*}{Cs_j^*})(\frac{Cs_j^*}{Cs_{A1}^*})}]$$

$$-\ \{+Asv_j\exp(-ikt_jx_3) - Bsv_j\exp(+ikt_jx_3)\}(2(\frac{\omega H1}{Cs_{A1}^*})\frac{m_{x1}}{(\frac{Cs_1^*}{Cs_{A1}^*})}[1-m_{xj}^2]^{1/2})]$$

(3.19)

$$\frac{(\bar{\sigma}_{33})_j(x,\omega)\,H1}{\mu_j^*} =$$

$$+i\exp(+ikx_1)[\{+Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}(\frac{\omega H1}{Cs_{A1}^*})[\frac{1}{(\frac{Cs_j^*}{Cs_{A1}^*})}\frac{Cp_j^*}{Cs_j^*}(1-2m_{xj}^2)$$

$$+\ \{+Asv_j\exp(-ikt_jx_3) - Bsv_j\exp(+ikt_jx_3)\}(2(\frac{\omega H1}{Cs_{A1}^*})\frac{m_{x1}}{(\frac{Cs_1^*}{Cs_{A1}^*})}[1-m_{xj}^2]^{1/2})]$$

(3.20)

$$\frac{(\bar{\sigma}_{31})_j(x,\omega)\,H1}{\mu_j^*} =$$

$$+i\exp(+ikx_1)[\{-Ap_j\exp(-iks_jx_3) + Bp_j\exp(+iks_jx_3)\}2(\frac{\omega H1}{Cs_{A1}^*})m_{x1}\sqrt{1-l_{xj}^2}\,/\frac{Cs_1^*}{Cs_{A1}^*}$$

$$+ \{+Asv_j\exp(-ikt_jx_3) + Bsv_j\exp(+ikt_jx_3)\}(\frac{\omega H1}{Cs_{A1}^*})\frac{1}{(\frac{Cs_j^*}{Cs_{A1}^*})}(1-2m_{xj}^2)]$$

(3.21)

Considering the stress free conditions at the ground surface and the welded conditions at the layer interfaces, equations (3.13), (3.14), (3.20), and (3.21) can be put into matrix form. Incident motion is introduced into the equation by setting either or both the half-space wave coefficients to 1.0 . Equation (3.22) and (3.23) are the matrix equations used to solve for the incident and reflected wave coefficients for each layer "j". Equation (3.22) is the matrix equation for a vertically incident wave while equation (3.23) is for an oblique half-space wave. After wave coefficients are determined, displacements and stresses can be found using equations (3.13) through (3.17).

$$\begin{bmatrix} +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\Psi_{-11} & +\Psi_{+11} & 0 & 0 & +\Psi_{-21} & -\Psi_{+21} & 0 & 0 \\ -\Phi_{-11} & +\Phi_{+11} & 0 & 0 & +\Phi_{-21} & -\Phi_{+21} & 0 & 0 & 0 & 0 \\ +\chi_1\Phi_{-11} & +\chi_1\Phi_{+11} & 0 & 0 & -\zeta_1\chi_2\Phi_{-21} & -\zeta_1\chi_2\Phi_{+21} & 0 & 0 & 0 & 0 \\ 0 & 0 & +\Psi_{-11} & +\Psi_{+11} & 0 & 0 & -\zeta_1\Psi_{-21} & -\zeta_1\Psi_{+21} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\Psi_{-22} & +\Psi_{+22} & 0 & -\Psi_{+hs\,2} \\ 0 & 0 & 0 & 0 & -\Phi_{-22} & +\Phi_{+22} & 0 & 0 & -\Phi_{+hs\,2} & 0 \\ 0 & 0 & 0 & 0 & +\chi_2\Phi_{-22} & +\chi_2\Phi_{+22} & 0 & 0 & -\zeta_2\chi_{hs}\Phi_{+hs\,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & +\Psi_{-22} & +\Psi_{+22} & 0 & -\zeta_2\Psi_{\ hs\,2} \end{bmatrix}$$

$$* \begin{Bmatrix} Ap_1 \\ Bp_1 \\ Asv_1 \\ Bsv_1 \\ Ap_2 \\ Bp_2 \\ Asv_2 \\ Bsv_2 \\ Bp_{hs} \\ Bsv_{hs} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix} \qquad \begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix} = \begin{Bmatrix} -\Psi_{-hs\,2}\, Asv_{hs} \\ -\Phi_{-hs\,2}\, Ap_{hs} \\ +\zeta_2\chi_{hs}\Phi_{-hs\,2} Ap_{hs} \\ +\zeta_2\Psi_{-hs\,2} Asv_{hs} \end{Bmatrix}$$

$$\zeta_j = (Cs^*_{j+1}/Cs^*_j)\,(\gamma_{j+1}/\gamma_j) \qquad \chi_j = \sqrt{\frac{2(1-\nu_j)}{(1-2\nu_j)}}$$

$$\phi_{+jk} = e^{+iks_jz_k}\ ;\ \phi_{-jk} = e^{-iks_jz_j}\ ;\ \psi_{+jk} = e^{+ikt_jz_k}\ ;\ \psi_{-jk} = e^{-ikt_jz_k} \tag{3.22}$$

$$
\begin{bmatrix}
+l_{x1}\alpha_1 & +l_{x1}\alpha_1 & +2\zeta_1 & -2\zeta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2\chi_1 & +2\chi_1 & \eta_1 & \eta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
+l_{x1}\phi_{-11} & +l_{x1}\phi_{+11} & -\zeta_1\psi_{-11} & +\zeta_1\psi_{+11} & -l_{x2}\phi_{-21} & -l_{x2}\phi_{+21} & +\zeta_2\psi_{-21} & -\zeta_2\psi_{+21} & 0 & 0 \\
-\chi_1\phi_{-11} & +\chi_1\phi_{+11} & -m_{x1}\psi_{-11} & -m_{x1}\psi_{+11} & +\chi_2\phi_{-21} & -\chi_2\phi_{+21} & +m_{x2}\psi_{-21} & +m_{x2}\psi_{+21} & 0 & 0 \\
+l_{x1}\alpha_1\phi_{-11} & +l_{x1}\alpha_1\phi_{+11} & +2\zeta_1\psi_{-11} & -2\zeta_1\psi_{+11} & -\kappa_{12}\phi_{-21} & -\kappa_{12}\phi_{+21} & -2\rho_{12}\psi_{-21} & +2\rho_{12}\psi_{+21} & 0 & 0 \\
-2\chi_1\phi_{-11} & +2\chi_1\phi_{+11} & +\eta_1\psi_{-11} & +\eta_1\psi_{+11} & +2\lambda_{12}\phi_{-21} & -2\lambda_{12}\phi_{+21} & -\xi_1\eta_2\psi_{-21} & -\xi_1\eta_2\psi_{+21} & 0 & 0 \\
0 & 0 & 0 & 0 & +l_{x2}\phi_{-22} & +l_{x2}\phi_{+22} & -\zeta_2\psi_{-22} & +\zeta_2\psi_{+22} & -l_{xhs}\phi_{+hs\,2} & -\zeta_{hs}\psi_{+hs\,2} \\
0 & 0 & 0 & 0 & -\chi_2\phi_{-22} & +\chi_2\phi_{+22} & -m_{x2}\psi_{-22} & -m_{x2}\psi_{+22} & -\chi_{hs}\phi_{+hs\,2} & +m_{xhs}\psi_{+hs\,2} \\
0 & 0 & 0 & 0 & +l_{x2}\alpha_2\phi_{-22} & +l_{x2}\alpha_2\phi_{+22} & +2\zeta_2\psi_{-22} & -2\zeta_2\psi_{+22} & -\kappa_{2hs}\phi_{+hs\,2} & +2\rho_{2hs}\psi_{+hs\,2} \\
0 & 0 & 0 & 0 & -2\chi_2\phi_{-22} & +2\chi_2\phi_{+22} & +\eta_2\psi_{-22} & \eta_2\psi_{+22} & -2\lambda_{2hs}\phi_{+hs\,2} & -\xi_2\eta_{hs}\psi_{+hs\,2}
\end{bmatrix}
$$

$$
*\begin{Bmatrix} Ap_1 \\ Bp_1 \\ Asv_1 \\ Bsv_1 \\ Ap_2 \\ Bp_2 \\ Asv_2 \\ Bsv_2 \\ Bp_{hs} \\ Bsv_{hs} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix} \qquad \begin{Bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{Bmatrix} = \begin{bmatrix} +l_{xhs} & -\zeta_{hs} \\ -\chi_{hs} & -m_{xhs} \\ +\xi_2 l_{xhs}\alpha_{hs} & +2\xi_2\zeta_{hs} \\ -2\xi_2\chi_{hs} & +\xi_2\eta_{hs} \end{bmatrix} \begin{Bmatrix} Ap_{hs}\phi_{-hs\,2} \\ Asv_{hs}\psi_{-hs\,2} \end{Bmatrix}
$$

$$
\xi_j = (Cs^*_{j+1}/Cs^*_j)^2(\gamma_{j+1}/\gamma_j) \quad ; \quad \chi_j = \sqrt{1-l^2_{xj}} \quad ; \quad \zeta_j = \sqrt{1-m^2_{xj}} \quad ; \quad \eta_j = \frac{1-2m^2_{xj}}{m_{xj}} \quad ; \quad \alpha_j = \frac{1}{m^2_{xj}} - 2
$$

$$
\phi_{+jk} = e^{+iks_j z_k} \; ; \; \phi_{-jk} = e^{-iks_j z_k} \quad ; \quad \psi_{+jk} = e^{+ikt_j z_k} \; ; \; \psi_{-jk} = e^{-ikt_j z_k} \; ; \; \kappa_{jk} = \xi_j l_{xk}\alpha_k \; ; \; \lambda_{jk} = \xi_j\chi_k \; ; \; \rho_{jk} = \xi_j\zeta_k \qquad (3.23)
$$

The rock outcrop motion is calculated at $x_3 = H_{total}$ and assumes that there is no material above this level. Therefore, the rock outcrop motion is solved by considering the problem as a half-space problem at the level $x_3=H_{total}$ with stress free boundary conditions.

$$(\bar{u}_1)_{outcrop}(x,\omega) =$$
$$[l_{x\,hs}\{Ap_{hs}\exp(-iks_{hs}H_{total})$$
$$+Bp_{hs}\exp(+iks_{hs}H_{total})\} + \sqrt{1-m_{x\,hs}^2}\{-Asv_{hs}\exp(-ikt_{hs}H_{total})+Bsv_{hs}\exp(+ikt_{hs}H_{total})\}] \tag{3.24}$$

$$(\bar{u}_3)_{outcrop}(x,\omega) =$$
$$[\sqrt{1-l_{x\,hs}^2}\{-Ap_{hs}\exp(-iks_{hs}H_{total})$$
$$+ Bp_{hs}\exp(+iks_{hs}H_{total})\} + m_{x\,hs}\{-Asv_{hs}\exp(-ikt_{hs}H_{total}) - Bsv_{hs}\exp(+ikt_{hs}H_{total})\}] \tag{3.25}$$

$$(\bar{\sigma}_{33})_{outcrop}(x,\omega) =$$
$$+i\,\mu_{hs}^*[\{+Ap_{hs}\exp(-iks_{hs}H_{total}) + Bp_{hs}\exp(+iks_{hs}H_{total})\}kl_{x\,hs}(\frac{1}{m_{x\,hs}^2} - 2)+$$
$$\{+Asv_{hs}\exp(-ikt_{hs}H_{total}) - Bsv_{hs}\exp(+ikt_{hs}H_{total})\}(2k\sqrt{1-m_{x\,hs}^2})] \tag{3.26}$$

$$(\bar{\sigma}_{31})_{outcrop}(x,\omega) =$$
$$+i\,\mu_{hs}^*[\{-Ap_{hs}\exp(-iks_{hs}H_{total}) + Bp_{hs}\exp(+iks_{hs}H_{total})\}2\frac{\omega\, m_{x\,hs}}{Cs_{hs}^*}\sqrt{1-l_{x\,hs}^2}+$$
$$\{+Asv_{hs}\exp(-ikt_{hs}H_{total}) + Bsv_{hs}\exp(+ikt_{hs}H_{total})\}\frac{\omega}{Cs_{hs}^*}(\frac{1-2m_{x\,hs}^2}{m_{x\,hs}})] \tag{3.27}$$

Solution for the wave coefficients is found by setting $Ap_{hs} = 1.0$ or $Asv_{hs} = 1.0$, depending on the particular incident wave for which solutions are desired and solving the matrix equation (3.28):

$$\begin{bmatrix} \left(\frac{Cp^*_{hs}}{Cs^*_{hs}} - 2m_{x\,hs} l_{x\,hs}\right) & -2m_{x\,hs}\left[1-m^2_{x\,hs}\right]^{1/2} \\ 2m_{x\,hs}\left[1-l^2_{x\,hs}\right]^{1/2} & 1-2m^2_{x\,hs} \end{bmatrix} \begin{Bmatrix} Bp_{hs}\exp(+iks_{hs}H_{total}) \\ Bsv_{hs}\exp(+ikt_{hs}H_{total}) \end{Bmatrix} =$$

$$\begin{bmatrix} 2\,m_{x\,hs}\,l_{x\,hs} - \frac{Cp^*_{hs}}{Cs^*_{hs}} & -2m_{x\,hs}\left[1-m^2_{x\,hs}\right]^{1/2} \\ +2m_{x\,hs}\left[1-l^2_{x\,hs}\right]^{1/2} & 2m^2_{x\,hs}-1 \end{bmatrix} \begin{Bmatrix} Ap_{hs}\exp(-iks_{hs}H_{total}) \\ Asv_{hs}\exp(-ikt_{hs}H_{total}) \end{Bmatrix}$$

(3.28)

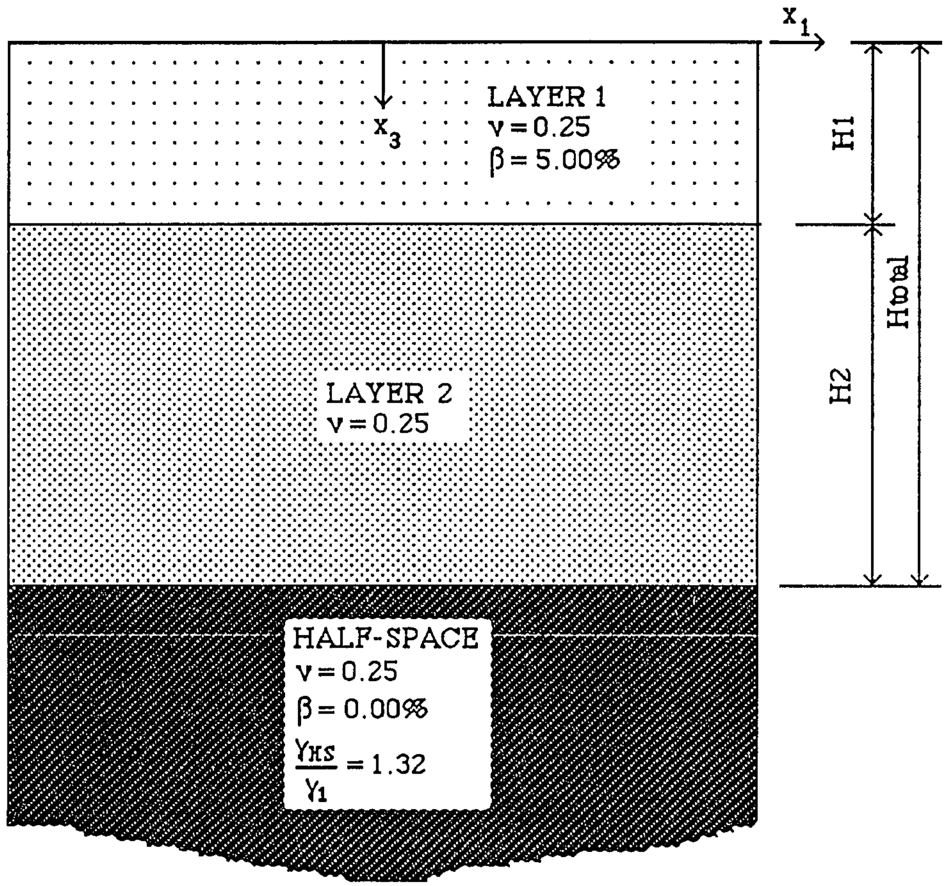
An oblique P wave will always reflect as a P wave and SV wave, however below a certain angle, the critical angle, an incident SV wave will not reflect as a P wave. Instead of a P wave, a surface wave is created which decays exponentially with depth. The critical angle is calculated by determining the angle in which the reflected P wave is 0. This can be found using equation (3.5) which results in the critical angle for the half-space being a function of the Poisson ratio of the half-space:

$$\theta_{SV} = \cos^{-1}(Cs/Cp) = \cos^{-1}\left[\sqrt{\frac{(1-2\nu)}{2(1-\nu)}}\right] \tag{3.29}$$

Chapter 4
Results of One-Dimensional Analysis

One-dimensional amplifications were determined for the various possible soil column configurations of Figure 1-2 if each side of the soil configuration ($\pm x_1$) were analyzed separately. This is equivalent to disregarding the scattering that the bedrock discontinuity causes. In Figure 1-2, the half-space below H_{total} is the same for both $\pm x_1$ and only the properties of layer 2 differ on the $-x_1$ and $+x_1$ sides. The one-dimensional results were used to initially determine what may be causes of significant amplification for the two-dimensional analysis. Figure 4-1 shows the assumed soil profile used in the 1-d analysis and the soil properties. Shear velocity and unit weight are given as a ratio to the soil layer 1 properties.

The Poisson ratio for the half-space is equal to 0.25 and therefore corresponds to a critical angle for the incident SV wave equal to 54.74 degrees. The half-space incidence angles used in this study are greater than this value. Therefore, only reflected P and SV waves can be expected for the in-plane motion when the angle of incidence is other than vertically incident. For vertical incidence there is no mode conversion, P waves reflect and refract as P waves, and SV waves reflect and refract as SV waves. For anti-plane motion, regardless of the half-space angle of incidence, there is no mode conversion and SH waves reflect and refract as SH waves.



LAYER 2 SOIL PROPERTIES:

Cs (layer 2) / Cs (layer 1)	$\frac{\gamma_2}{\gamma_1}$	DAMPING (%)
10.0	1.32	1.00
5.0	1.32	2.00
2.5	1.32	2.00
1.0	1.00	5.00

Figure 4-1: Soil Profile Used For The One-Dimensional Solution

Within each layer the angle of incidence is equal to the angle of reflection. In order to determine the angle of incidence for each layer "j", m_{xj} and l_{xj} can be determined for each layer from equation (3.18). Knowing these values, the P wave angle of incidence is the arc tangent of s_j and the SV angle of incidence the arc tangent of t_j:

$$\theta p_j = \tan^{-1}\left[\mathrm{Re}\left\{\sqrt{\frac{1}{l_{xj}^2} - 1}\right\}\right] \tag{4.1}$$

and

$$\theta sv_j = \tan^{-1}\left[\mathrm{Re}\left\{\sqrt{\frac{1}{m_{xj}^2} - 1}\right\}\right] \tag{4.2}$$

The incidence angles are given for each of the layers for each of the studied cases in Tables 4-A and 4-B. For an incident SH wave, the incidence angles are equal to θsv_j which are given in Table 4-A. Natural frequencies for the cases are given in Tables 4-C, 4-D, and 4-E .

$\theta sv_{HS} = 90.00^{\circ}$; $\theta p_{HS} = 90.00^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θsv (layer j) (degrees)	θp (layer j) (degrees)
10	1.0	90.00	90.00
5	2.0	90.00	90.00
2.5	2.0	90.00	90.00
1.0	5.0	90.00	90.00

$\theta sv_{HS} = 75.00^{\circ}$; $\theta p_{HS} = 63.37^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θsv (layer j) (degrees)	θp (layer j) (degrees)
10	1.0	75.00	63.36
5	2.0	82.56	77.04
2.5	2.0	86.29	83.56
1.0	5.0	88.51	87.42

$\theta sv_{HS} = 60.00^{\circ}$; $\theta p_{HS} = 30.00^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θsv (layer j) (degrees)	θp (layer j) (degrees)
10	1.0	60.00	30.00
5	2.0	75.51	64.33
2.5	2.0	82.82	77.49
1.0	5.0	87.12	85.01

Table 4-A: Angle of Incidence Due To An Incident SV Wave

$\theta p_{HS} = 90.00^{\circ}$; $\theta sv_{HS} = 90.00^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θp (layer j) (degrees)	θp (layer j) (degrees)
10	1.0	90.00	90.00
5	2.0	90.00	90.00
2.5	2.0	90.00	90.00
1.0	5.0	90.00	90.00

$\theta p_{HS} = 75.00^{\circ}$; $\theta sv_{HS} = 81.41^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θp (layer j) (degrees)	θp (layer j) (degrees)
10	1.0	75.00	81.41
5	2.0	82.56	85.71
2.5	2.0	86.29	87.86
1.0	5.0	88.51	89.14

$\theta p_{HS} = 60.00^{\circ}$; $\theta sv_{HS} = 73.22^{\circ}$

Cs (layer j) / Cs (layer 1)	DAMPING (%)	θp (layer i) (degrees)	θsv (layer i) (degrees)
10	1.0	60.00	73.22
5	2.0	75.51	81.70
2.5	2.0	82.82	85.86
1.0	5.0	87.12	88.34

Table 4-B: Angle of Incidence Due To An Incident P Wave

For a basis of comparison, the natural frequency of a soil layer on a half-space is given as:

For a vertically propagating shear wave:

$$\frac{\omega_n H1}{Cs_1} = \frac{(2n-1)}{4} * 2\pi \tag{4.3}$$

and for a vertically propagating compression wave:

$$\frac{\omega_n H1}{Cp_1} = \frac{(2n-1)}{4} * 2\pi \tag{4.4}$$

which is equivalent to:

$$\frac{\omega_n H1}{Cs_1} = \frac{(2n-1)}{4} * \frac{Cp_1}{Cs_1} * 2\pi \tag{4.5}$$

For the lowest natural frequency, or fundamental frequency, the dimensionless fundamental frequency is for a vertically propagating shear wave, 1.5708, and for a vertically propagating compression wave, 2.7207 . For these soil profiles, the natural frequencies at which maximum horizontal and vertical displacement are basically invariant for the chosen incidence angles.

4.1: Incident SH Wave

Results for an incident SH wave are given in Figures 4-2 through 4-4 for the various soil profiles and half-space incidence angles. The surface displacement is given as a function of the dimensionless frequency ks1*H1 where ks1 is the shear wave number and

H1 the thickness of layer 1. A summary of the maximum amplifications and natural frequencies are given in Table 4-C. Figures 4-2 through 4-4 indicate that the surface motion is not sensitive to the angle of incidence for the angles used. Maximum amplification occurs at the fundamental frequency for all the cases except for when the shear velocity ratio of layer 2 is equal to 2.5 and the thickness ratio is equal to 5.0 . For this particular case, maximum amplification occurs at the second natural frequency. The case in which the embedded rock layer is treated as soil is also included. This is equivalent to studying the dependency of the surface amplification to the thickness of the soil layer. As the thickness of layer 2 increases or the shear velocity ratio decreases, the soil profile becomes less stiff resulting in the natural frequencies decreasing. Maximum amplification, u_2=13.6, is caused by a vertically incident SH wave and occurs when the thickness of layer 2 is equal to layer 1 and the shear velocity ratio is equal to 2.5 .

Table 4-C: Natural Frequencies and Corresponding Amplitudes
Due To An Incident SH Wave

HORIZONTAL DISPLACEMENT

H2/H1	Cs2/Cs1	1st natural frequency n = 1 (fundamental frequency)				2nd natrural frequency n = 2			
		ks1*H1	θ incident			ks1*H1	θ incident		
			90.0	75.0	60.0		90.0	75.0	60.0
1.0	10.0	1.6	12.9	12.7	12.0	4.7	6.4	6.3	6.1
	5.0	1.6	13.0	12.8	12.1	4.6	7.2	7.1	6.7
	2.5	1.4	13.6	13.3	12.4	3.7	6.4	6.2	5.8
	1.0	0.8	12.7	12.5	11.8	2.4	6.4	6.3	6.1
2.0	10.0	1.6	12.9	12.7	12.0	4.7	6.4	6.3	6.1
	5.0	1.5	13.1	12.9	12.1	4.8	7.0	6.9	6.5
	2.5	1.2	12.6	12.3	11.4	2.3	7.7	7.5	7.0
	1.0	0.5	12.0	11.8	11.2	1.6	6.3	6.2	6.0
5.0	10.0	1.6	12.9	12.7	12.0	4.7	6.4	6.3	6.1
	5.0	1.5	10.5	10.3	10.0	4.7	6.8	6.7	6.4
	2.5	0.7	8.5	8.3	7.6	1.6	11.4	11.2	10.5
	1.0	0.3	12.0	11.8	11.2	0.8	6.2	6.1	6.0

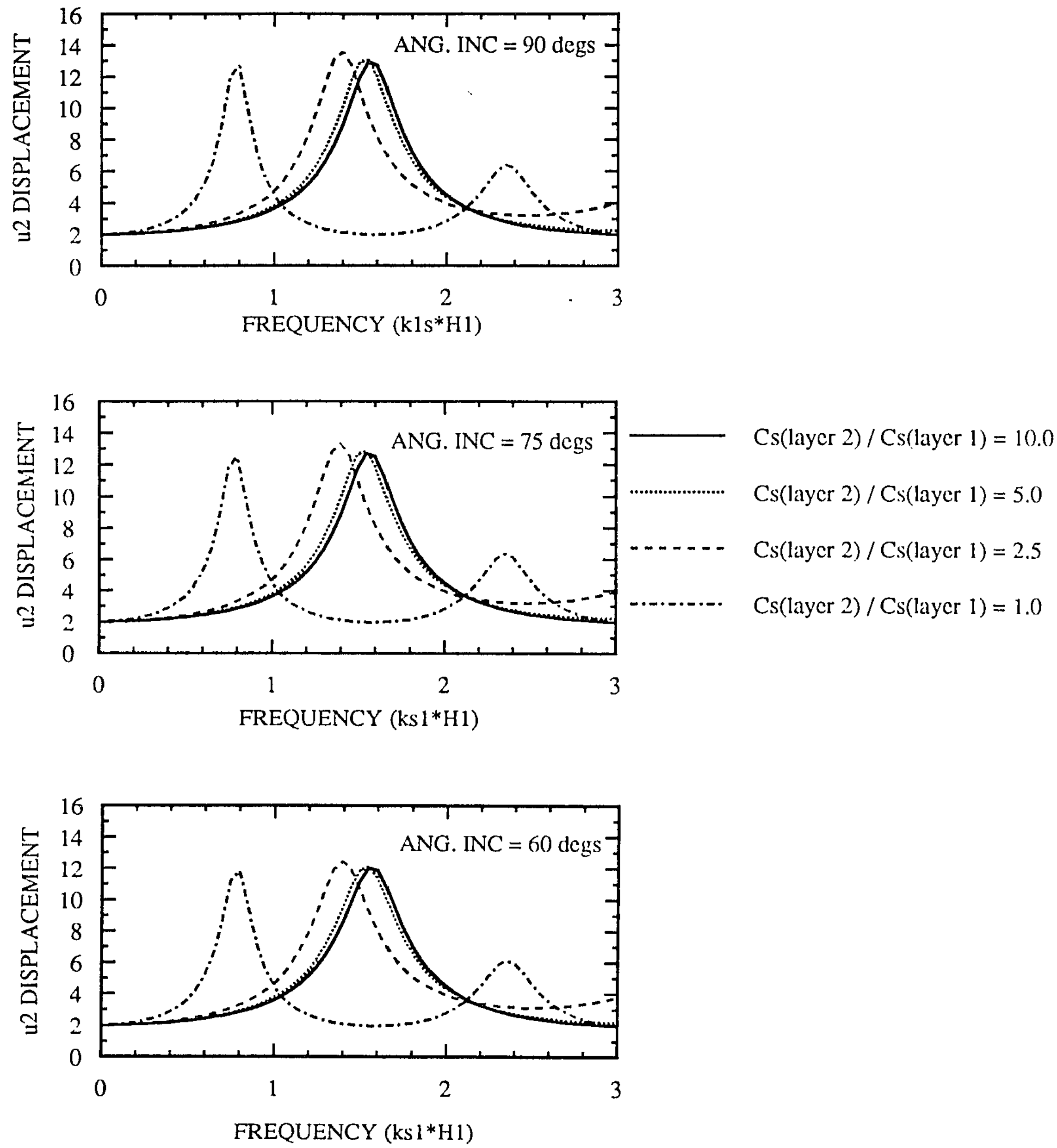


Figure 4-2: One-Dimensional Analysis Surface Displacement, u2,
Due To A Unit SH Wave,
H2 / H1 = 1.0

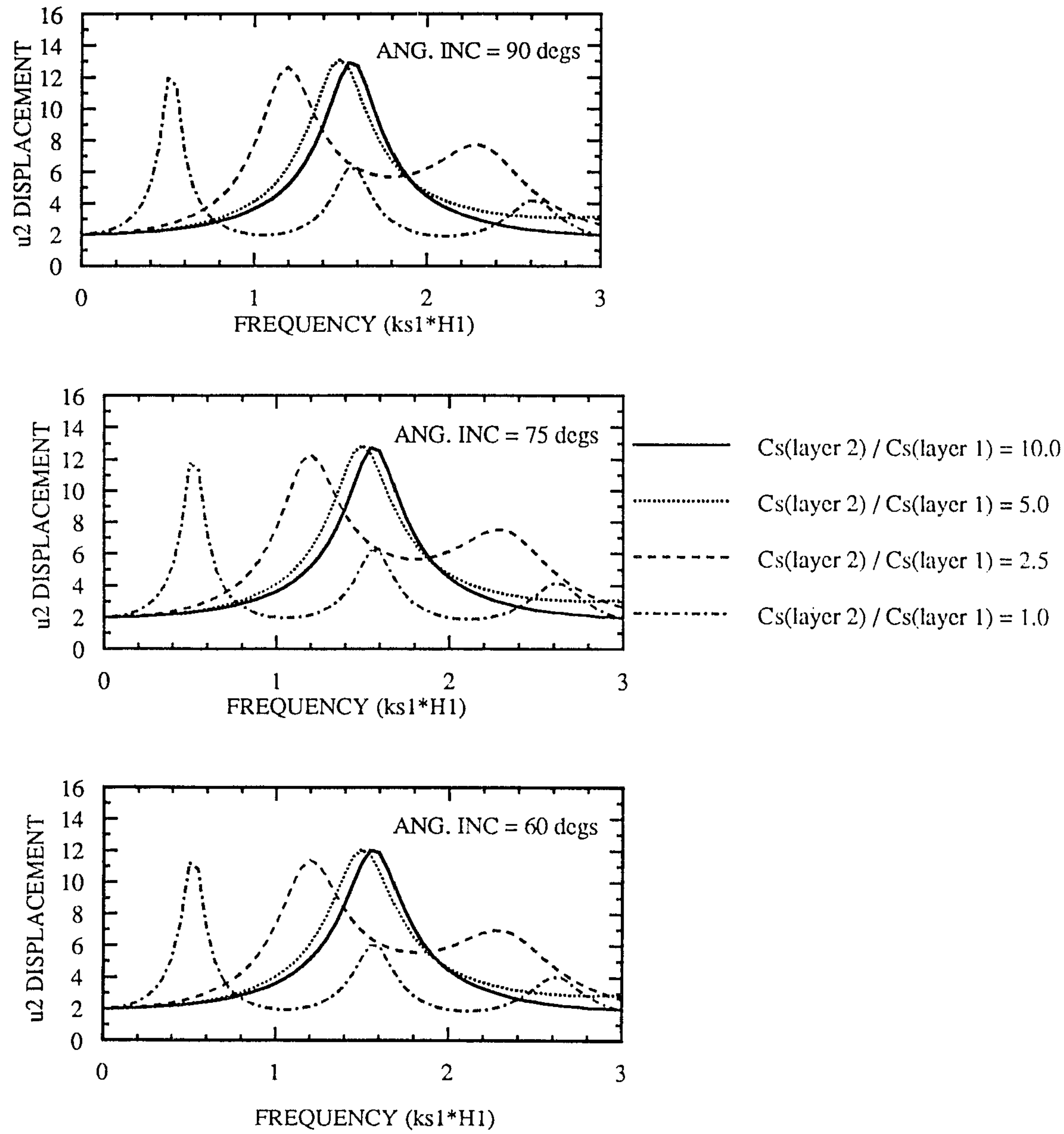


Figure 4-3: One-Dimensional Analysis Surface Displacement, u2,
Due To A Unit SH Wave,
H2 / H1 = 2.0

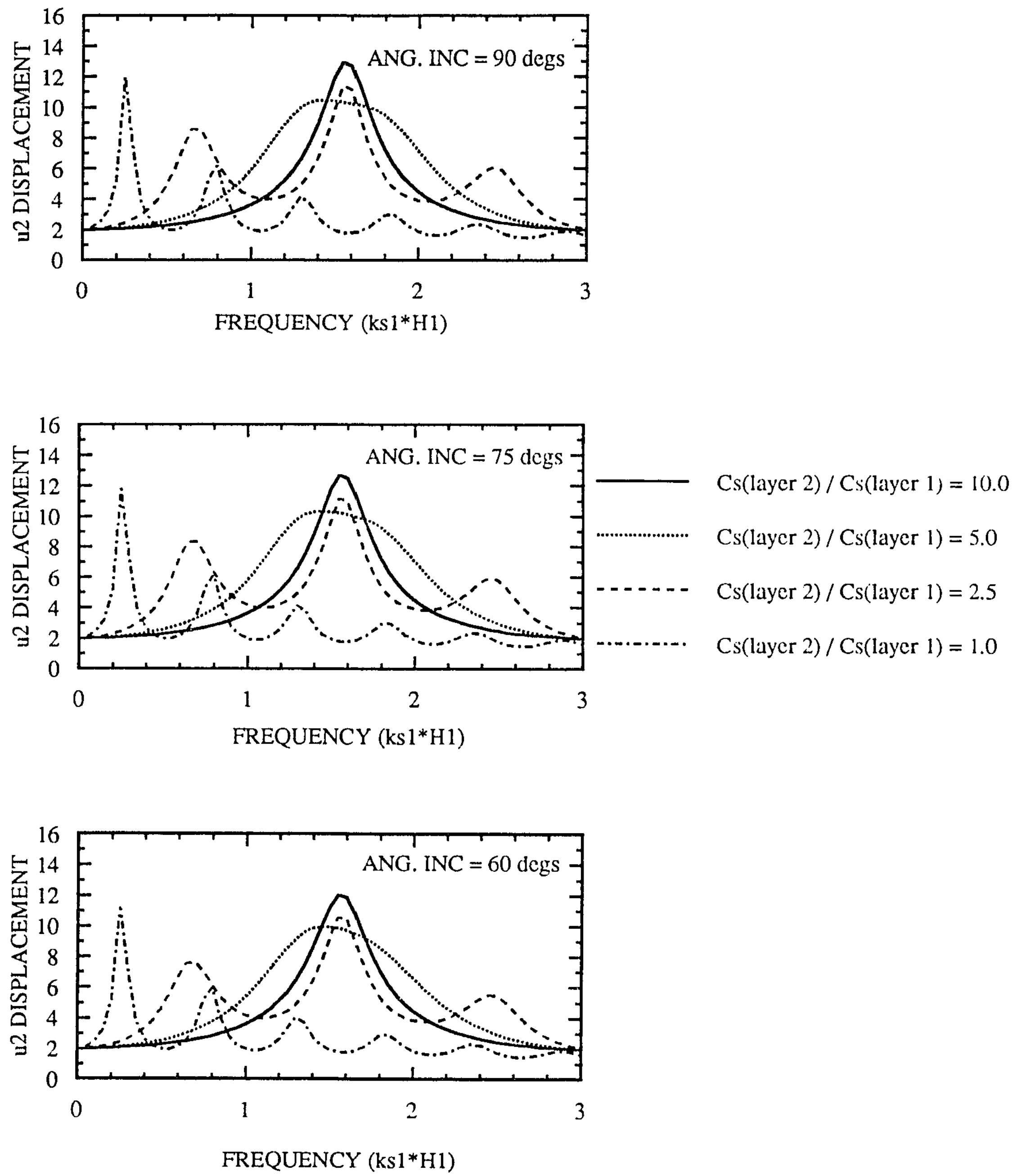


Figure 4-4: One-Dimensional Analysis Surface Displacement, u2,
Due To A Unit SH Wave,
H2 / H1 = 5.0

4.2: Incident SV Wave

Figures 4-5 through 4-8 are plots of the response for the various soil profiles and half-space incidence angles due to a half-space incident SV wave. For the case of vertical incidence, there is no mode conversion and the surface displacement is the same as a vertically incident SH wave. Also, for vertical incidence there is no vertical displacement developed. As the angle of incidence decreases, the horizontal displacement decreases and the vertical displacement increases. Maximum horizontal and vertical displacements occur for the case of Cs(layer 2) / Cs(layer 1) = 2.5 and H2 / H1 = 1.0 . Maximum horizontal displacement, u_1 = 13.6, occurs due to a vertically incident SV wave at a dimensionless frequency = 1.4 . Maximum vertical displacement , u_3 = 6.5, occurs for the same soil profile with a half-space incidence angle of 60 degrees and a dimensionless frequency = 2.4 .

Table 4-D: Natural Frequencies and Corresponding Amplitudes
Due To An Incident SV Wave

HORIZONTAL DISPLACEMENT

H2/H1	Cs2/Cs1	1st natural frequency n = 1 (fundamental frequency)				2nd natrural frequency n = 2			
		ks1*H1	θ incident			ks1*H1	θ incident		
			90.0	75.0	60.0		90.0	75.0	60.0
1.0	10.0	1.6	13.0	11.8	8.3	4.7	6.4	5.9	4.8
	5.0	1.5	13.1	11.9	8.3	4.6	7.2	6.6	5.0
	2.5	1.4	13.6	12.2	8.3	3.7	6.4	5.8	3.94
2.0	10.0	1.6	13.0	11.8	8.3	4.7	6.4	5.9	4.8
	5.0	1.5	13.1	11.8	8.3	4.8	7.0	6.6	5.6
	2.5	1.2	12.6	11.3	7.4	2.3	7.7	7.0	4.9

VERTICAL DISPLACEMENT

H2/H1	Cs2/Cs1	1st natural frequency n = 1 (fundamental frequency)				2nd natrural frequency n = 2			
		ks1*H1	θ incident			ks1*H1	θ incident		
			90.0	75.0	60.0		90.0	75.0	60.0
1.0	10.0	2.7	0.0	3.7	6.4	>5			
	5.0	2.7	0.0	3.8	6.4	>5			
	2.5	2.4	0.0	3.9	6.5	>5			
2.0	10.0	2.7	0.0	3.7	6.4	>5			
	5.0	2.6	0.0	3.6	6.1	>5			
	2.5	2.0	0.0	3.3	5.6	3.9	0.0	2.0	3.2

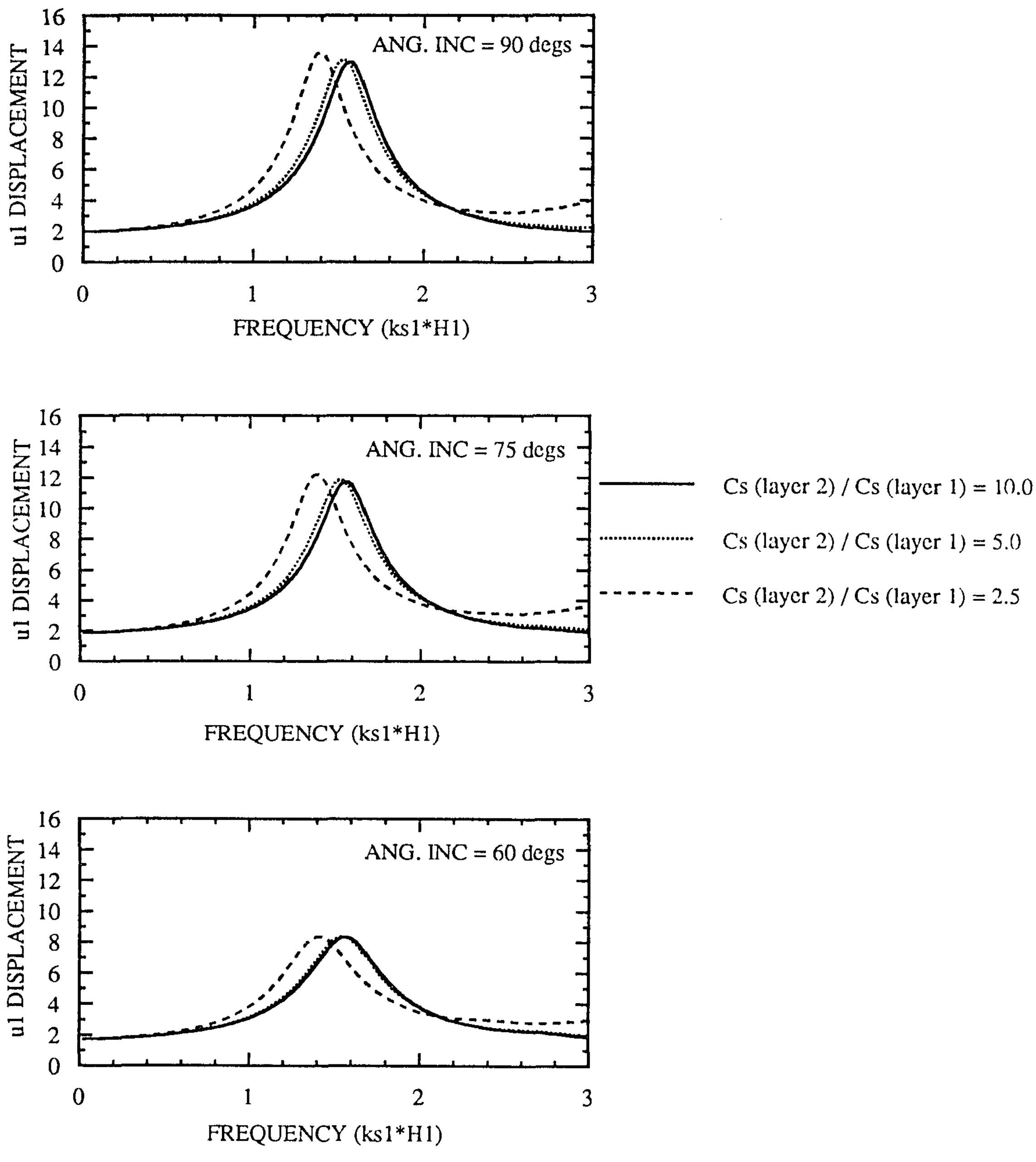


Figure 4-5: One-Dimensional Analysis Surface Displacement, u1,
Due To A Unit SV Wave
H2 / H1 = 1.0

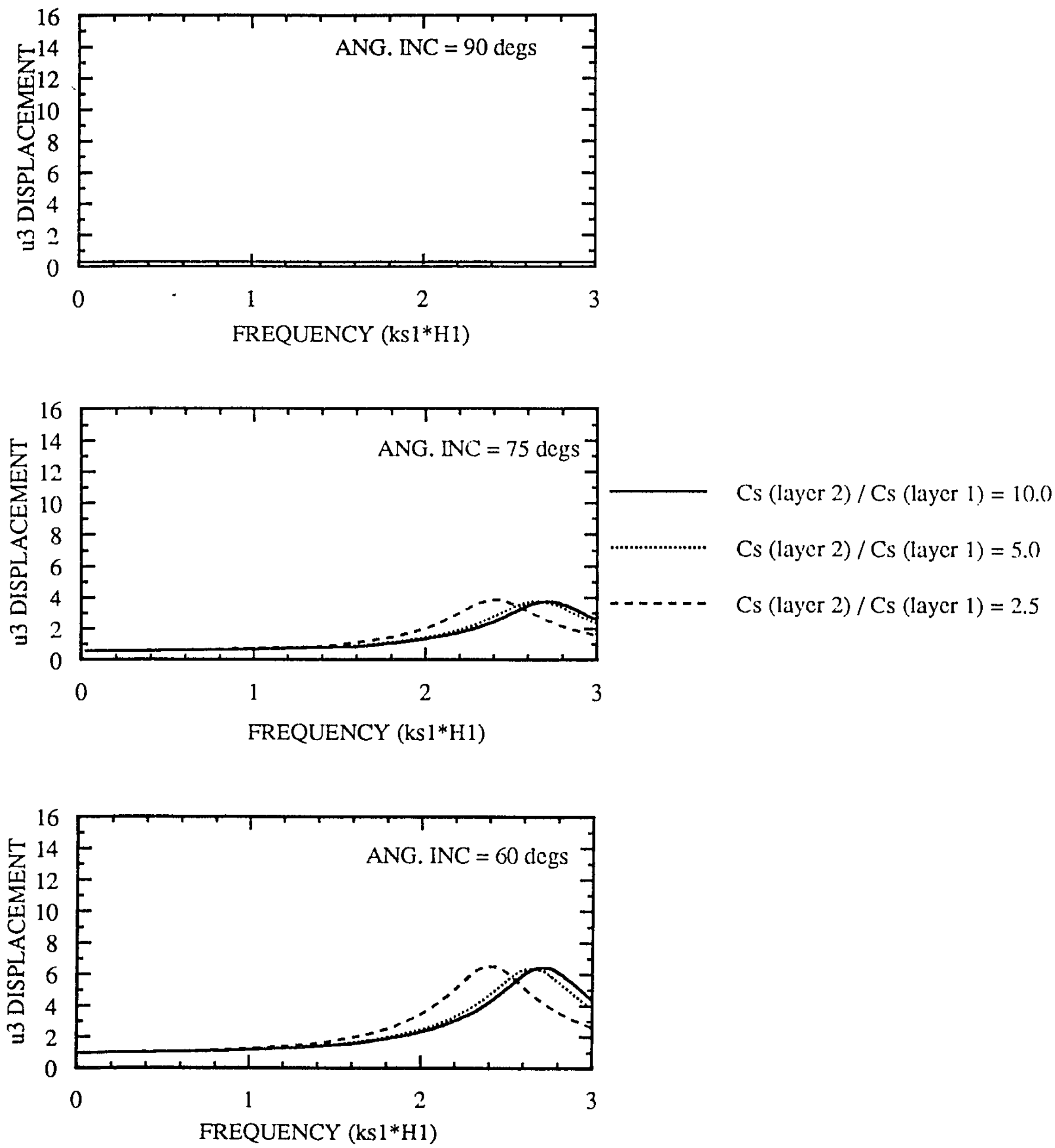


Figure 4-6: One-Dimensional Analysis Surface Displacement, u3,
Due To A Unit SV Wave,
H2 / H1 = 1.0

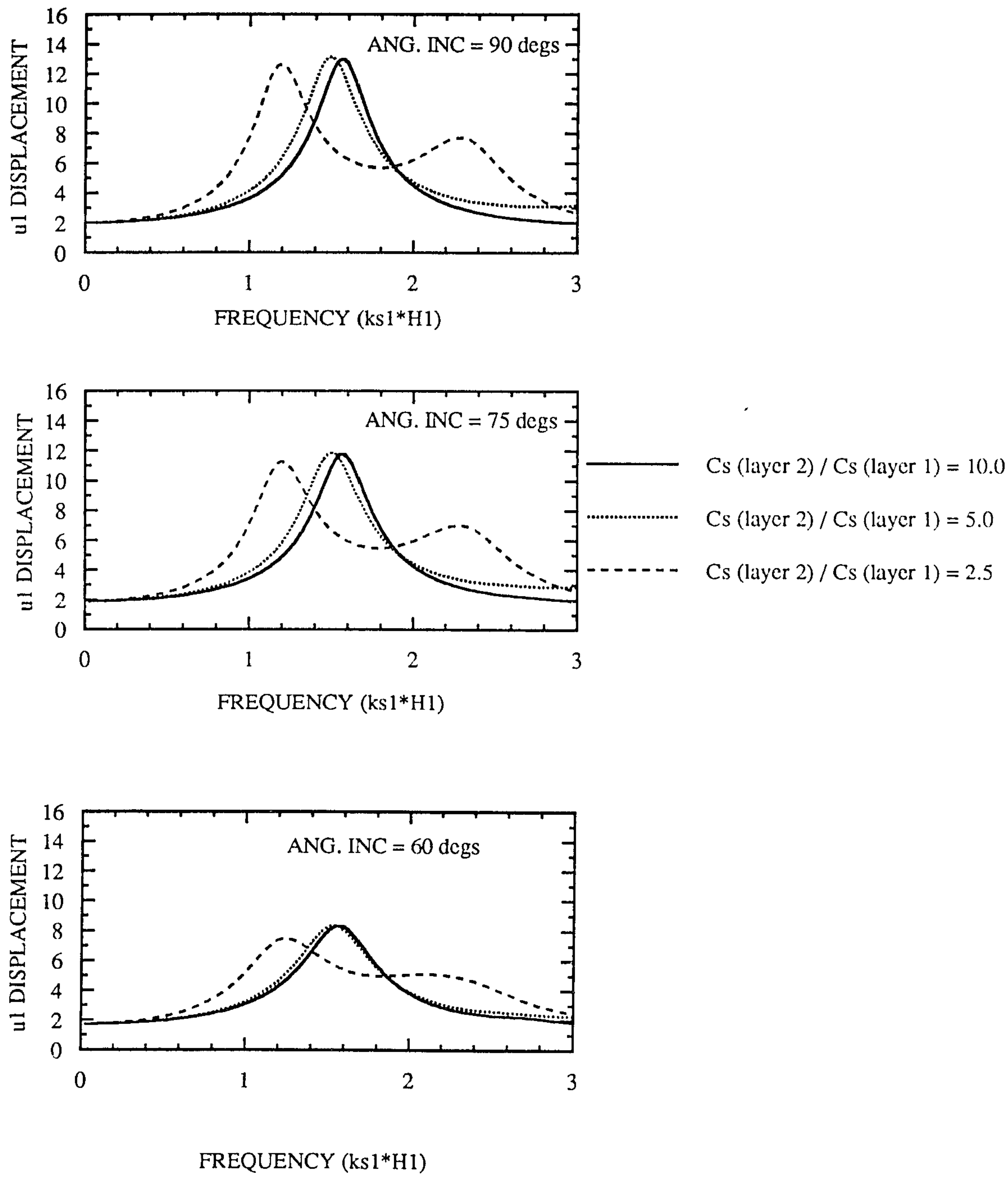


Figure 4-7: One-Dimensional Surface Displacement, u1,
Due To A Unit SV Wave,
H2 / H1 = 2.0

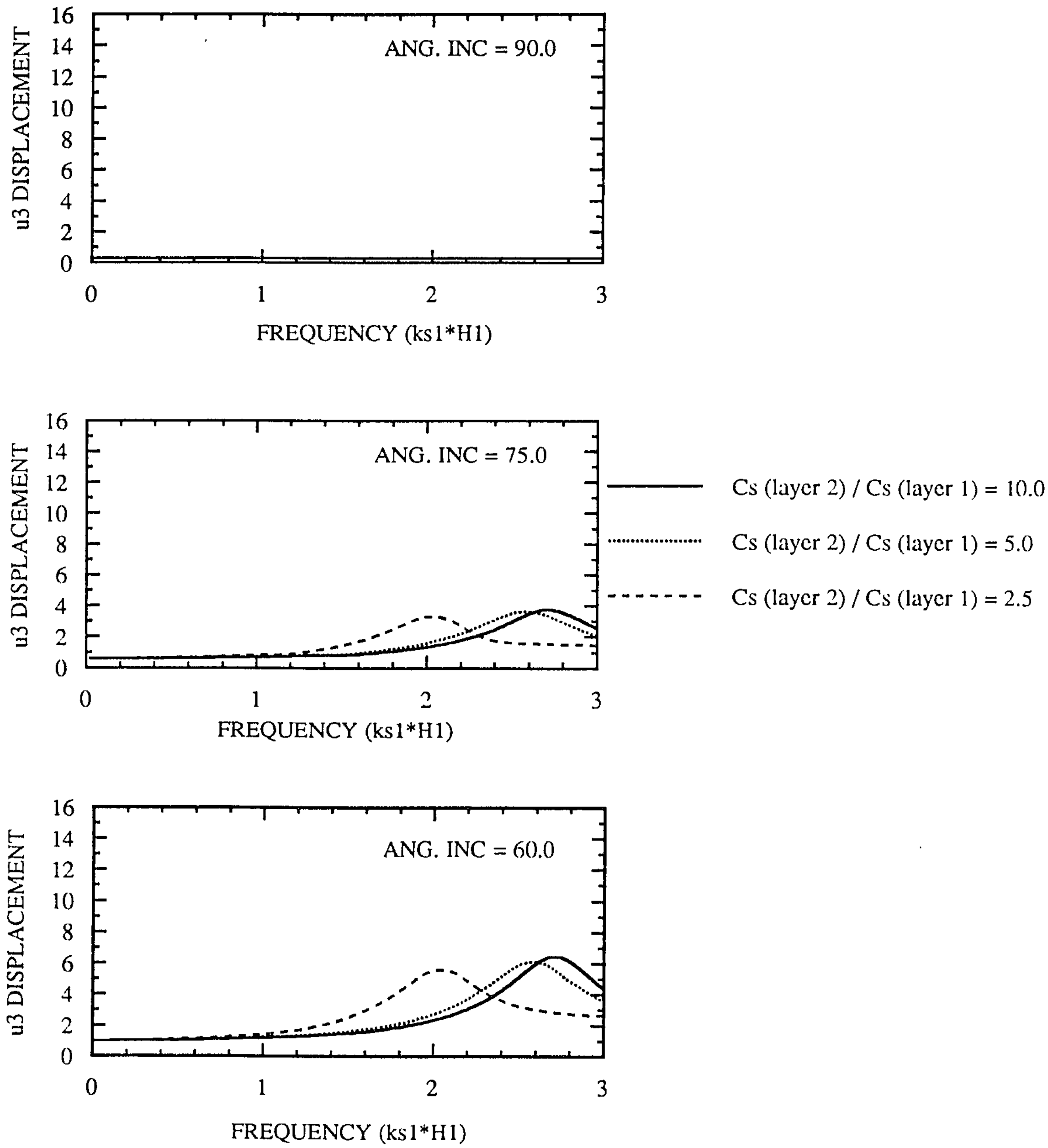


Figure 4-8: One-Dimensional Analysis Surface Displacement, u3,
Due To A Unit SV Wave,
H2 / H1 = 2.0

4.3: Incident P Wave

The surface amplifications for an incident P wave are shown in Figures 4-9 through 4-12 for the array of possible soil profiles and for incident angles equal to: 90, 75, and 60 degrees. No mode conversion occurs for a vertically incident P wave. Therefore, the maximum vertical displacement for this case is equal to the maximum horizontal displacement for a vertically propagating shear wave and coincides with zero horizontal displacement. As the half-space incidence angle of the P wave decreases, becomes shallower, the vertical displacement decreases and the horizontal displacement increases. Maximum displacements occur for the case of Cs(layer 2) / Cs(layer 1) = 2.5 and H2 / H1 = 1.0 . For this profile, the maximum vertical displacement, u_3 = 13.6, occurs at vertical incidence with a dimensionless frequency = 2.4. Maximum horizontal displacement, u_1 = 7.0, occurs for a half-space incidence angle of 60 degrees and a dimensionless frequency = 1.4 .

Table 4-E: Natural Frequencies and Corresponding Amplitudes
Due To An Incident P Wave

HORIZONTAL DISPLACEMENT

H2/H1	Cs2/Cs1	1st natural frequency n = 1 (fundamental frequency)				2nd natrural frequency n = 2			
		ks1*H1	θ incident			ks1*H1	θ incident		
			90.0	75.0	60.0		90.0	75.0	60.0
1.0	10.0	1.6	0.0	3.8	6.8	4.7	0.0	1.8	3.4
	5.0	1.5	0.0	3.8	6.8	4.6	0.0	2.0	3.6
	2.5	1.4	0.0	3.9	7.0	3.6	0.0	1.7	3.0
2.0	10.0	1.6	0.0	3.8	6.8	4.7	0.0	1.8	3.4
	5.0	1.5	0.0	3.8	6.8	4.7	0.0	1.5	2.7
	2.5	1.2	0.0	3.6	6.4	2.4	0.0	1.9	3.3

VERTICAL DISPLACEMENT

H2/H1	Cs2/Cs1	1st natural frequency n = 1 (fundamental frequency)				2nd natrural frequency n = 2			
		ks1*H1	θ incident			ks1*H1	θ incident		
			90.0	75.0	60.0		90.0	75.0	60.0
1.0	10.0	2.7	13.0	12.4	10.9	>5			
	5.0	2.7	13.1	12.6	11.1	>5			
	2.5	2.4	13.6	13.0	11.5	>5			
2.0	10.0	2.7	13.0	12.4	10.9	>5			
	5.0	2.6	13.1	12.6	11.1	>5			
	2.5	2.1	12.6	12.1	10.8	4.0	7.7	7.4	6.5

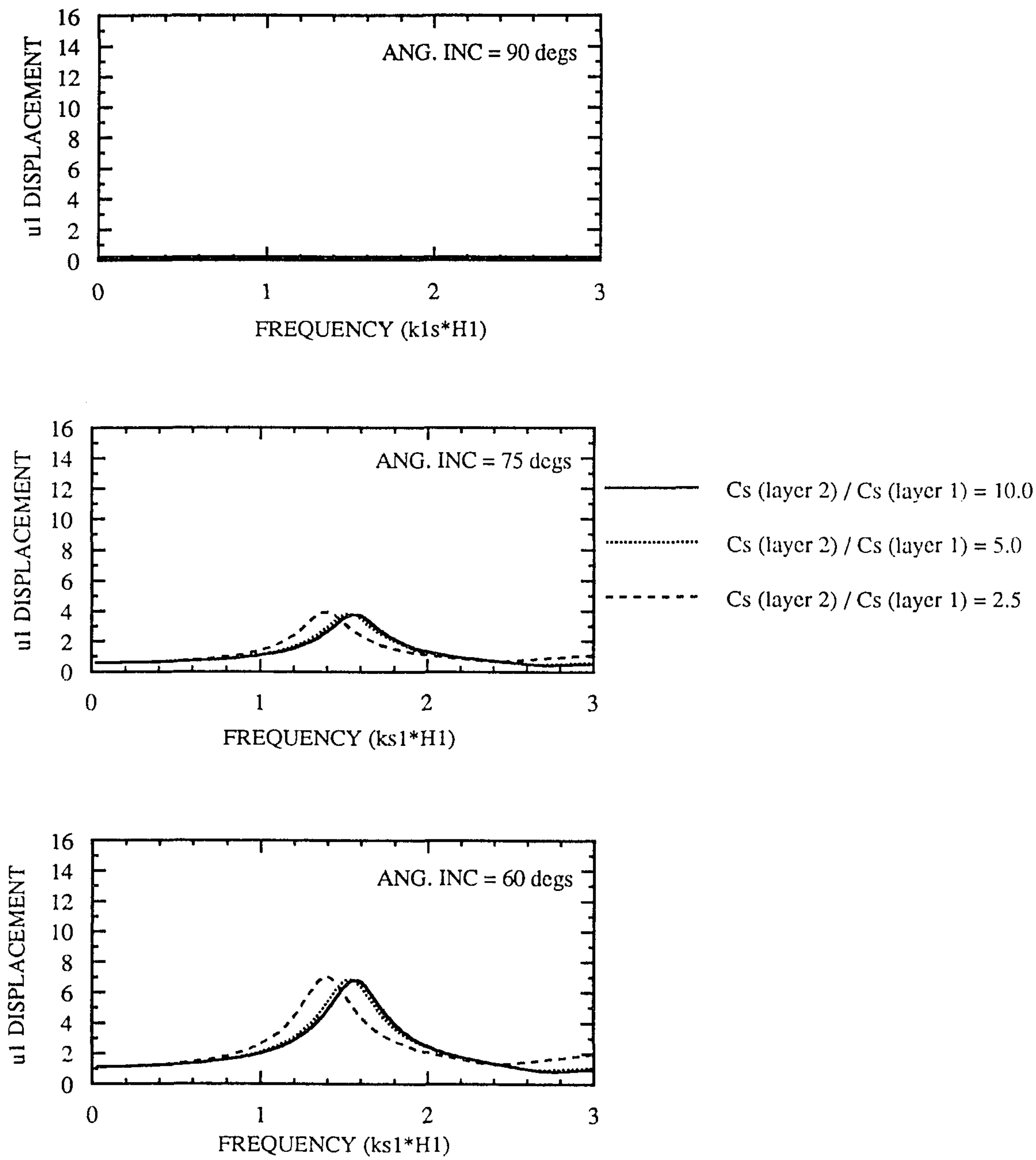


Figure 4-9: One-Dimensional Analysis Surface Displacement, u1,
Due To A Unit P Wave,
H2 / H1 = 1.0

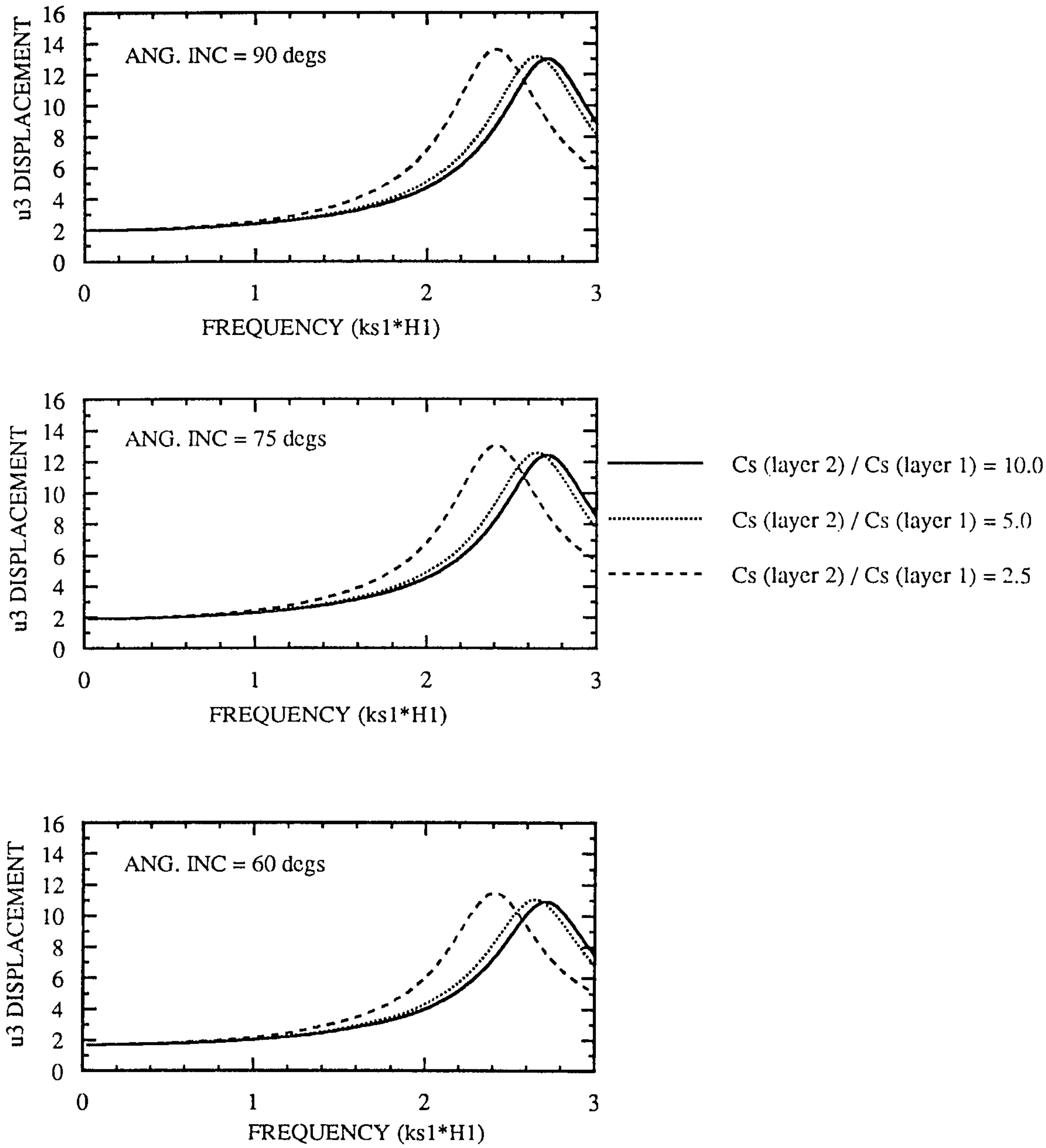


Figure 4-10: One-Dimensional Analysis Surface Displacement,u3,
Due To A Unit P Wave,
H2 / H1 = 1.0

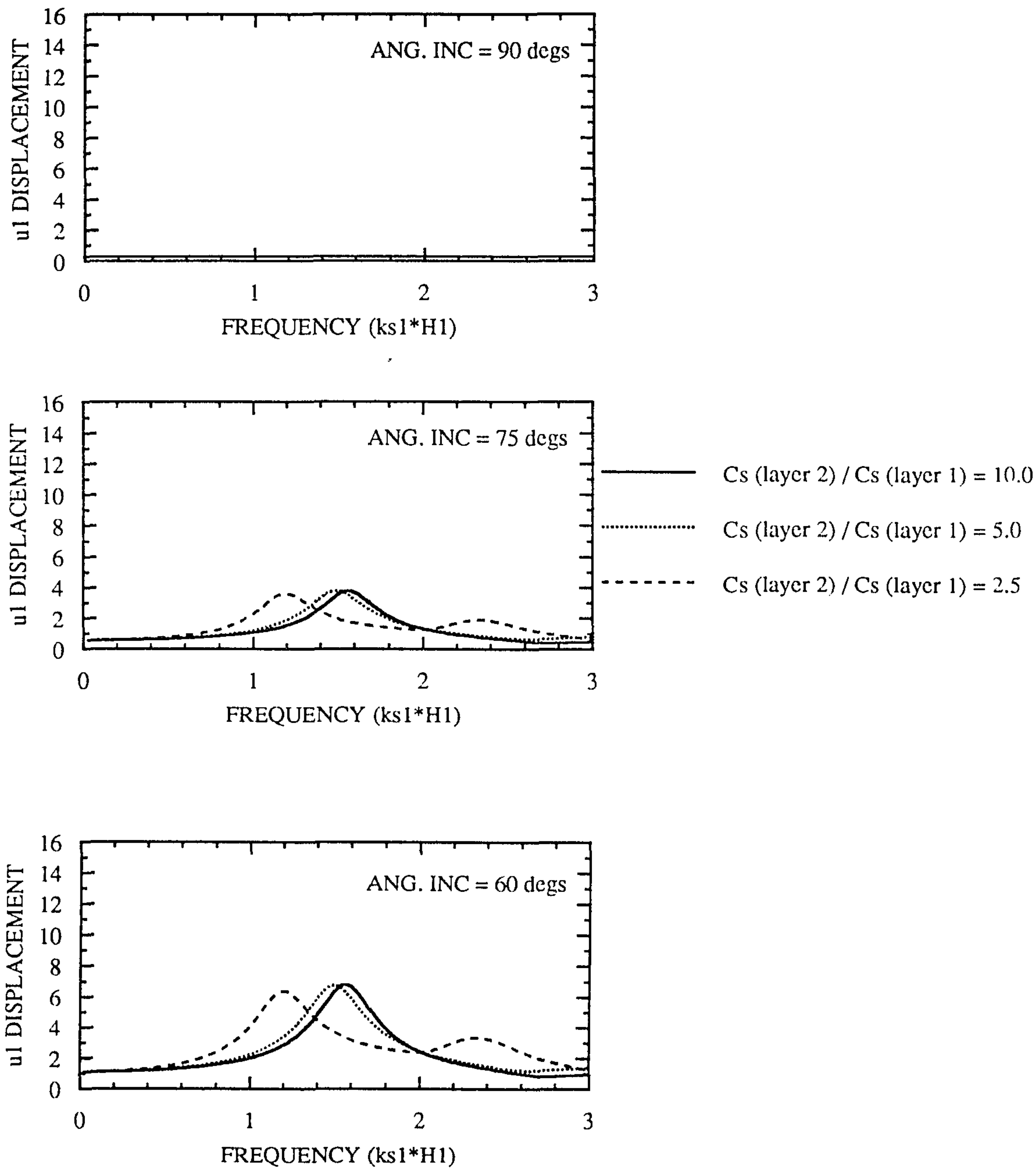


Figure 4-11: One-Dimensional Analysis Surface Displacement, u1,
Due To A Unit P Wave,
H2 / H1 = 2.0

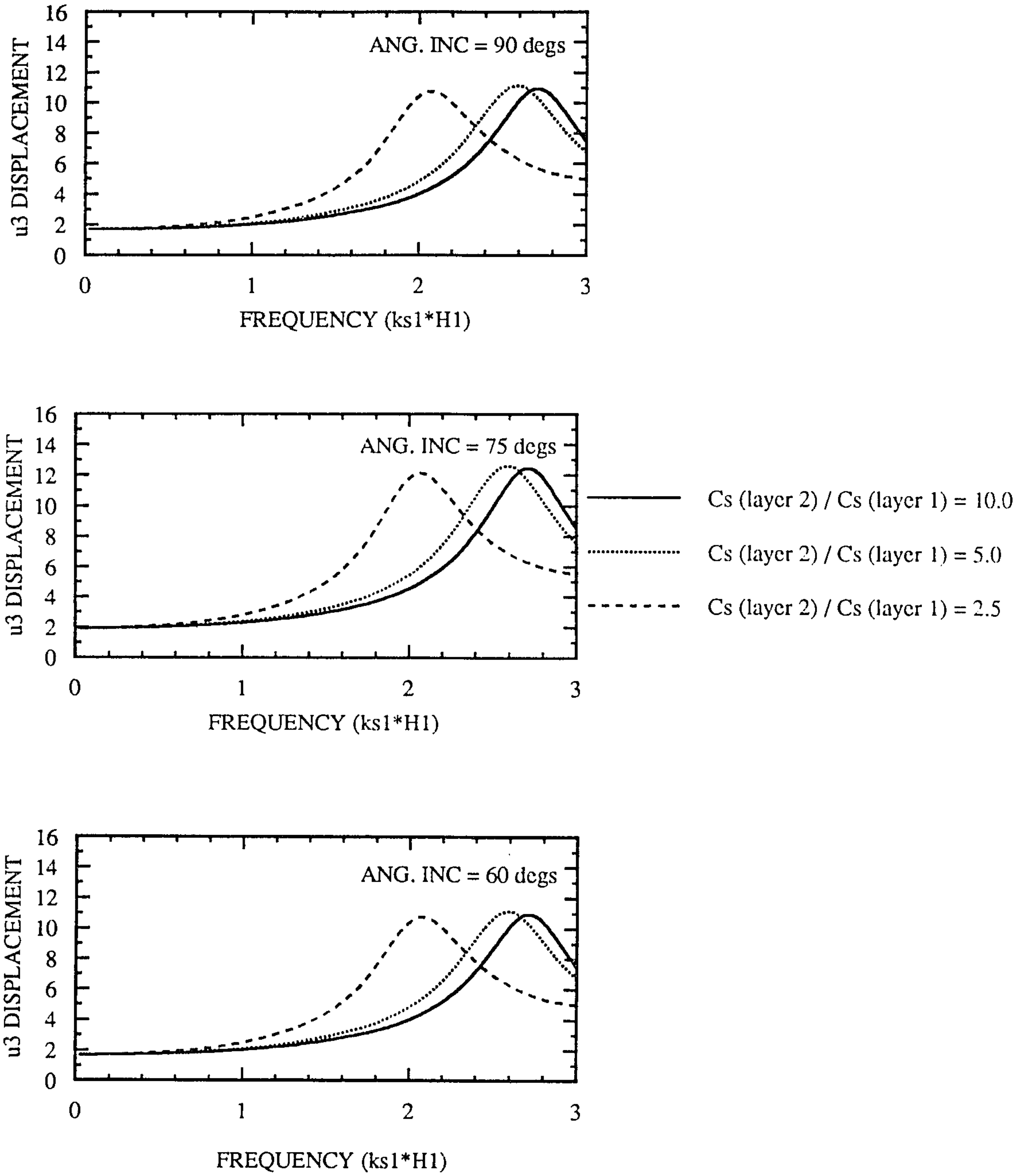


Figure 4-12: One-Dimensional Analysis Surface Displacement, u3,
Due To A Unit P Wave,
H2 / H1 = 2.0

4.4: Results to be Used in Two-Dimensional Analysis

For the cases of: Cs(layer 2)/ Cs(layer 1) = 10.0, 5.0, and 2.5, the dimensionless fundamental frequencies lie within the range of 0.6 and 3.0 . If layer 2 is considered to have the same soil properties as layer 1, this range is increased to 0.3 and 3.0 . What we are particularly interested in studying is a discontinuity in a rock half-space. Therefore, our primary interest is in the cases: Cs(layer 2)/ Cs(layer 1) = 10.0, 5.0, and 2.5. For the cases in which the shear velocity ratio of layer 2 is equal to 2.5, 5 or 10, the natural frequencies at which maximum horizontal displacement occurs fall in the range of 1.0 and 2.0 . For vertical displacement this range is between 2.0 and 3.0 . The lower frequency range of 1 and 2 is chosen to be studied most extensively in the two-dimensional analysis. In this suite of frequencies, an interval of 0.2 is used. A dimensionless frequency of 0.5 is included to examine the low frequency two-dimensional effect. Dimensionless frequencies of 2.5 and 3.0 are also included to examine both the P wave effects, which are significant in the range between 2.0 and 3.0, and the higher shear wave effects. This preliminary work of the one-dimensional study serves as a bench-mark to use for the scattering effects of the bedrock discontinuity in the two-dimensional study.

Chapter 5
Boundary Integral Equation Formulation

Because of its advantages, the boundary integral equation method (boundary element method) will be used to solve for the soil amplification for the various wave types. The boundary integral equation method is a weighted residual method which uses an influence function for the weighting function (Brebbia, 1989). The greatest advantage to the method is in the reduction of the domain, "V", problem to a boundary, "S", problem (Figure 5-1). Therefore, in our particular case, a two dimensional problem is reduced to a one dimensional problem which corresponds to a reduction in the numerical complexity of the problem.

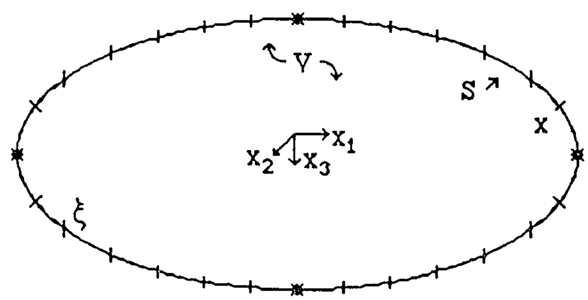


Figure 5-1: Boundary Integral Description

The methodology of the technique is to develop a boundary integral equation along the boundary "S". To develop the boundary integral equation, the displacement and traction influence functions are required. The displacement influence function is equal to the displacement at x due to a point load at ξ and the traction influence function is the traction at x due to a point load at ξ. The boundary is discretized into elements. In this work linear elements are used in which nodes are located at each end of the element. Within these elements, displacement and traction are assumed to vary linearly. At each node point within a domain there is, for anti-plane motion one boundary integral equation, and for in-plane motion two boundary integral equations. In the case of multi-domain problems, each domain is treated separately, however at interfacing boundaries the elements are discretized in a similar fashion in order that continuity equations can be established at node points. Therefore, at node points along an interfacing boundary there will be boundary integral equations from each of the boundary domains. After boundary integral equations are established, the resulting boundary integral equations are solved simultaneously through the use of boundary conditions and continuity conditions. In the following sections, boundary integral equations are developed for the anti-plane and in-plane wave motions.

5.1: Anti-Plane Boundary Integral Equation Formulation

The derivation of the boundary integral equation for anti-plane motion in an elastic material is available (Hadley et al, 1989). The boundary integral equation in this derivation will be expanded and extended to include materials with damping. Consider the two-dimensional problem with anti-plane motion, Figure 5-2 . Along the boundary, there are displacements and normal shear strains.

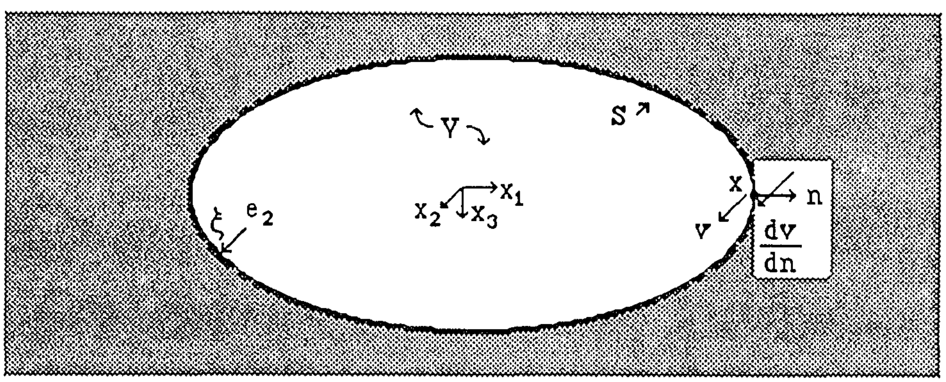


Figure 5-2: Anti-Plane Boundary Integral Description

Because this derivation is for anti-plane motion, in-plane motion is equal to zero and only out-of-plane motion exists:

$$u_1 = u_3 = 0,$$

and because the problem is plane strain:

$$\frac{\partial}{\partial x_2} = 0 \qquad (5.1)$$

With these conditions, the Navier steady state equation, equation (2.14), simplifies to the Helmholtz equation:

$$\nabla^2 \bar{u}_2(x,\omega) + \left(k_s^*\right)^2 \bar{u}_2(x,\omega) = 0 \qquad (5.2)$$

$\bar{u}_2(x,\omega)$ = Fourier transform of $u_2(x,t)$

k_s^*(complex shear wave number) = ω/C_s^*

In order to develop the boundary integral equation, equation (5.2), is multiplied by a weighting function, (u_{22}^*), and integrated over the domain:

$$\int_V \left[\nabla^2 \bar{u}_2(x,\omega) + \left(k_s^*\right)^2 \bar{u}_2(x,\omega)\right] u_{22}^*(x,\xi,\omega)\, dV(x) = 0 \qquad (5.3)$$

Using Green's theorem, equation (5.3) is transformed from a volume integral to a surface integral:

$$\int_V \bar{u}_2(x,\omega)\,[\nabla^2+\left(k_s^*\right)^2]\,u_{22}^*(x,\xi,\omega)\,dV(x) =$$
$$\int_S \bar{u}_2(x,\omega)\,\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)\,dS(x) - \int_S u_{22}^*(x,\xi,\omega)\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\,dS(x)$$

(5.4)

$\bar{u}_2(x,\omega)$ = displacement at x

$\frac{\partial \bar{u}_2}{\partial n}(x,\omega)$ = traction at x

$u_{22}^*(x,\xi)$ = anti-plane displacement influence function

$\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)$ = anti-plane traction influence function

In order to remove the volume integral in equation (5.4) and convert the equation into a boundary integral equation, u_{22}^* is chosen to satisfy equation (5.5):

$$\left[\nabla^2 + \left(k_s^*\right)^2\right] u_{22}^*(x,\xi) = -\delta(x - \xi) \tag{5.5}$$

$\delta(x - \xi)$ = Dirac delta function

The integral of the product of a function and the Dirac delta function is equivalent to the value of the function at the specified point where the Dirac delta function is specified:

$$\int_V f(x)\,\delta(x-\xi)\,dV(x) = f(\xi) \tag{5.6}$$

By using the Dirac delta function, the integration over the domain "V" of equation (5.4) is removed. After substituting the Dirac delta function, equation (5.4) becomes:

$$- \int_V \bar{u}_2(x,\omega)\, \delta(x - \xi)\, dV(x) = \int_S \bar{u}_2(x,\omega) \frac{\partial u^*_{22}}{\partial n}(x,\xi,\omega)\, dS(x) - \int_S u^*_{22}(x,\xi,\omega) \frac{\partial \bar{u}_2}{\partial n}(x,\omega)\, dS(x) \tag{5.7}$$

Total displacement in equation (5.7) is substituted for by incident and scattering wave displacements. The total displacement, $\bar{u}_2$, is the summation of the displacement due to the incident wave (i) and the displacement due to scattering (s). The incident wave displacement occurs in the half-space. Since the incident wave has no singularity in the volume it satisfies equation 5.8 in the volume "V" (Mow & Pao, 1973):

$$\int_S u^*_{22}(x,\xi,\omega) \frac{\partial \bar{u}^i_2}{\partial n}(x,\omega)\, dS(x) - \int_S \bar{u}^i_2(x,\omega) \frac{\partial u^*_{22}}{\partial n}(x,\xi,\omega)\, dS(x) = 0.0 \tag{5.8}$$

Scattering displacement originates at the boundary surface and is the displacement due to diffraction, reflection, and refraction caused by the incident wave impacting the boundary surface. The scattered wave has a singularity at the boundary surface and satisfies:

$$\bar{u}^s_2(\xi,\omega) = \int_S u^*_{22}(x,\xi,\omega) \frac{\partial \bar{u}^s_2}{\partial n}(x,\omega)\, dS(x) - \int_S \bar{u}^s_2(x,\omega) \frac{\partial u^*_{22}}{\partial n}(x,\xi,\omega)\, dS(x) \tag{5.9}$$

Adding equations (5.8) and (5.9) results in:

$$\bar{u}_2^s(\xi,\omega) = \int_S u_{22}^*(x,\xi,\omega)\,\frac{\partial \bar{u}_2^s}{\partial n}(x,\omega)\,dS(x) - \int_S \bar{u}_2^s(x,\omega)\,\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)\,dS(x) \tag{5.10}$$

Equation (5.10) is a function of scattered wave displacements, however, it is preferred to have equation (5.10) be in terms of the known incident wave displacements and the unknown total displacements. The displacement due to scattering in equation (5.10) is replaced by the difference of the total and incident displacements:

$$\bar{u}_2^s(\xi) = \bar{u}_2(\xi) - \bar{u}_2^i(\xi) = \int_S u_{22}^*(x,\xi,\omega)\,\frac{\partial \bar{u}_2}{\partial n}(x,\xi)\,dS(x) - \int_S \bar{u}_2(x,\xi)\,\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)\,dS(x) \tag{5.11}$$

Taking x to the boundary and allowing x to approach ξ causes singularities to occur in the integrals involving the influence functions. These singularities are considered in the boundary integral equation by modifying equation (5.11) to include the effect of these singularities as α and β for the singularity of the traction and displacement influence function respectively. Equation (5.11) becomes:

$$\bar{u}_2^i(\xi) = \int_S \bar{u}_2(x,\omega)\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)\,dS(x) - \int_S u_{22}^*(x,\xi,\omega)\,\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\,dS(x) + \beta\frac{\partial \bar{u}_2}{\partial n}(\xi) + \bar{u}_2(\xi)\,(1-\alpha) \tag{5.12}$$

This is the boundary integral equation used to solve for anti-plane motion. Integration in equation (5.12) is taken over the boundary of each of the domains. The integrals in equation (5.12) are taken equal to their Cauchy principal values.

5.1.1: Anti-Plane Influence Function

The boundary element method requires that the influence function $u_{22}^*(x,\xi)$, which satisfies equation (5.5), be known. To determine the displacement influence function, equation (5.5) is integrated about a small circular boundary (Hildebrand, 1976) . Allowing $x \to \xi$, results in

$$u_{22}^* = -\frac{1}{2\pi} \ln r \quad r \to 0$$

where

$$r = \sqrt{(x_1 - \xi_1)^2 + (x_3 - \xi_3)^2} \tag{5.13}$$

Considering $\xi \neq x$ (ie. $r \neq 0$), u_{22}^* is required to satisfy

$$\left[\nabla^2 + \left(k_S^*\right)^2\right] u_{22}^*(x,\xi,\omega) = 0 , \quad r \neq 0 . \tag{5.14}$$

u_{22}^* satisfies equation (5.14) if

$$u_{22}^*(x,\xi,\omega) = c_1 J_0 (k_S^* r) + c_2 Y_0 (k_S^* r) \tag{5.15}$$

The radiation condition requires that there is no incoming wave at $r = \infty$. In order to apply the radiation condition, u_{22}^* is transferred into complex form:

$$u_{22}^*(x,\xi,\omega) = a_1 H_0^{(1)} (k_S^* r) + a_2 H_0^{(2)} (k_S^* r) \tag{5.16}$$

where $H_0^{(n)}$ is the Hankel function of 0 order and "n" kind.

After taking the inverse Fourier transform, u_{22}^*, can be rewritten as:

$$F^{-1}\left[u_{22}^*(x,\xi,\omega)\right] = a_1 e^{-i\omega t} H_0^{(1)}(k_S^* r) + a_2 e^{-i\omega t} H_0^{(2)}(k_S^* r) \tag{5.17}$$

As $r \rightarrow \infty$, the first part of this summation represents an outgoing wave, while the second part, an incoming wave. Therefore, in order to satisfy the radiation condition, a_2 is set equal to 0 . In order to satisfy both conditions for $r \rightarrow 0$ and $r \rightarrow \infty$, it is required that:

$$u_{22}^{*}(x,\xi,\omega) = \frac{i}{4} H_0^{(1)}(k_s^{*} r) \tag{5.18}$$

The traction influence function is attained by taking the derivative of u_{22}^{*} with respect to n and using the chain rule.

5.1.1.1: Singularities

Along the boundary, singularities occur in the integration when r = 0 . Next, the singularities in anti-plane motion are determined by examining a small area, Γ, on the boundary and allowing $x \rightarrow \xi$, Figure 5-3.

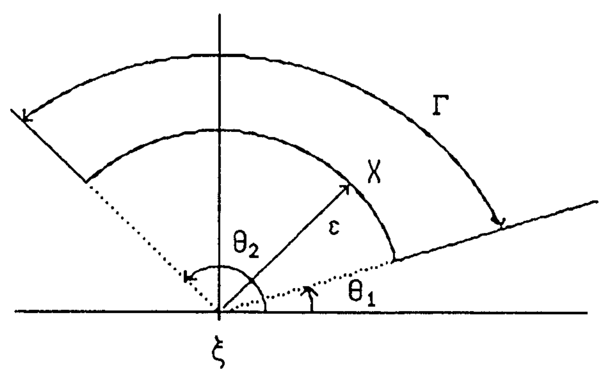


Figure 5-3: Singularity Analysis

To examine the singularity of u^*_{22} in the integral:

$$\int_\Gamma u^*_{22}(x,\xi,\omega)\,\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\,d\Gamma(x) \tag{5.19}$$

integration is performed in polar coordinates and the radius, ε, allowed to approach 0. Since the integration is taken oven an infinitesimal length, $\partial \bar{u}_2/\partial n$ is considered constant. After evaluating, the integral becomes:

$$\lim_{\varepsilon\to 0} \int_\Gamma u^*_{22}(x,\xi,\omega)\,\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\,d\Gamma(x) = 0 \tag{5.20}$$

The singularity in $\partial u^*_{22}/\partial n$ is examined next . Again a small area is examined and radius ε allowed to approach 0.

$$\lim_{\varepsilon\to 0} \int_\Gamma \frac{\partial u^*_{22}}{\partial n}(x,\xi,\omega)\,\bar{u}_2(x,\omega)\,d\Gamma(x) = -\frac{\bar{u}_2(x,\omega)\,\Delta\theta}{2\pi} \tag{5.21}$$

where $\Delta\theta$ represents the change in the exterior angle at ξ

Therefore, the boundary integral equation for anti-plane motion, equation (5.12), becomes:

$$\bar{u}^i_2(\xi) = \int_s \left[\bar{u}_2(x,\omega)\,\frac{\partial u^*_{22}}{\partial n}(x,\xi,\omega) - u^*_{22}(x,\xi,\omega)\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\right] dS(x) + \bar{u}_2(x,\omega)\left[1 - \frac{\Delta\theta}{2\Pi}\right] \tag{5.22}$$

where

$\Delta\theta = \theta_2 - \theta_1$ = exterior angle of the boundary at ξ

Therefore, $(1-\frac{\Delta\theta}{2\pi})$ is equal to the interior angle at ξ divided by 2π

5.2: In-Plane Boundary Integral Equation Formulation

Consider the two-dimension problem and in-plane motion shown in Figure 5-4:

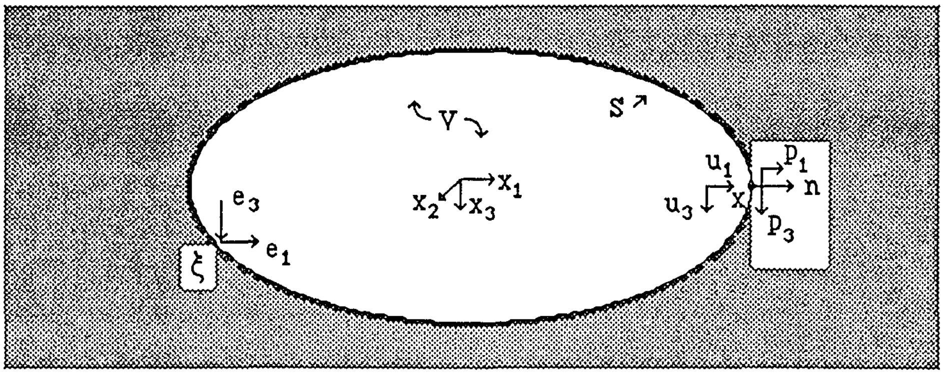


Figure 5-4: In-Plane Boundary Integral Description

The plane wave motion is due to P and SV waves. The elastodynamic boundary integral equation is developed here in a similar fashion as what is used to develop the static plane strain boundary integral equation which can be found elsewhere (Brebbia, 1989). The motion is in the plane and therefore $u_2 = 0$. The boundary integral equation for plane waves is formulated from the equilibrium equation, equation (2.1), which becomes after transferring to the frequency domain:

$$\bar{\sigma}_{mj,j} + \rho\omega^2\bar{u}_m + \bar{f}_m = 0 \tag{5.23}$$

The boundary integral equation is formulated using a weighted residual method in which the equilibrium equation in the frequency domain, equation (5.23), is multiplied by a weighting function, $U^*_m(x,\xi,\omega)$, and integrated over the domain. Body forces are excluded since our interest is with the change in displacement from the static position.

$$\int_V \left[\bar{\sigma}_{mj,j}(x,\omega) + \rho\omega^2 \bar{u}_m(x,\omega) \right] U_m^*(x,\xi,\omega)\, dV(x) = 0 \tag{5.24}$$

Green's theorem is applied to the stress term in equation (5.24) and further modified using Betti's reciprocal theorem. The integral of the stress term becomes:

$$\int_V \left[\bar{\sigma}_{mj,j}(x,\omega)\right] U_m^*(x,\xi,\omega)\, dV(x) = \int_S U_m^*(x,\xi,\omega)\, \bar{\sigma}_{mj}(x)\, n_j(x)\, dS(x) - \int_S \Sigma_{mj}^*(x,\xi,\omega)\, \bar{u}_m(x,\omega) n_j(x)\, dS(x) + \int_S \bar{u}_m(x,\omega)\, \Sigma_{mj,j}^*(x,\xi,\omega)\, dV(x) \tag{5.25}$$

$U_m^*(x,\xi,\omega)$ = displacement influence function

$\Sigma_{mj}^*(x,\xi,\omega)$ = stress influence function

nj(x) = normal at x in j direction

Substituting equation (5.25) into equation (5.24):

$$\int_S \left\{ (U_m^*(x,\xi,\omega)\, \bar{\sigma}_{mj}(x,\omega)\, n_j(x) - \Sigma_{mj}^*(x,\xi,\omega)\, \bar{u}_m(x,\omega)\, n_j(x) \right\} dS(x) + \int_V \left[\Sigma_{mj,j}^*(x,\xi,\omega) + \rho\omega^2 U_m^*(x,\xi,\omega) \right] \bar{u}_m(x,\omega)\, dV(x) = 0 \tag{5.26}$$

A weighting factor is selected in such a manner as to remove the volume integral of the displacement in equation (5.26):

$$\Sigma_{mj,j}^*(x,\xi,\omega) + \rho\omega^2 U_m^*(x,\xi,\omega) = -\delta(x-\xi)\, e_m \tag{5.27}$$

where

$e_m(\xi)$ = direction vector in "m" direction at ξ

and equation (5.26) becomes:

$$\int_S \left[(U^*_m(x,\xi,\omega)\, \bar{\sigma}_{mj}(x,\omega)\, n_j(x) - \Sigma^*_{mj}(x,\xi,\omega)\, \bar{u}_m(x,\omega)\, n_j(x) \right] dS(x) = \int_V \left[\delta(x-\xi)\, e_m(\xi) \right] \bar{u}_m(x,\omega)\, dV(x)$$

(5.28)

To remove the unit direction vector, e_m, the displacement and stress influence functions, $U^*_m(x,\xi,\omega)$ and $\Sigma^*_{mj}(x,\xi,\omega)$, are written as the summation of their components in each of the "k" directions:

$$U^*_m(x,\xi,\omega) = u^*_{mk}(x,\xi,\omega)\, e_k(\xi)$$

and

$$\Sigma^*_{mj}(x,\xi,\omega) = \sigma^*_{mjk}(x,\xi,\omega)\, e_k(\xi)$$

(5.29)

where

u^*_{mk} = displacement at x in the m direction due to a unit force at ξ in the k direction

σ^*_{mjk} = stress at x due to a unit force at ξ in the k direction

These terms from equation (5.29) are substituted into equation (5.28) and e_k is factored out:

$$\int_S \left[u^*_{mk}(x,\xi,\omega)\, \bar{\sigma}_{mj}(x,\omega)\, n_j(x) - \sigma^*_{mjk}(x,\xi,\omega) \bar{u}_m(x,\omega)\, n_j(x) \right] e_k\, dS\,(x) = \int_V \delta(x-\xi)\, \bar{u}_k(x,\omega)\, e_k\, dV(x)$$

(5.30)

The displacement in each of the "k" directions is taken independently and equation (5.30) becomes:

$$\int_S \left[u^*_{mk}(x,\xi,\omega)\,\bar{\sigma}_{mj}(x,\omega)\,n_j(x) - \sigma^*_{mjk}(x,\xi,\omega)\bar{u}_m(x,\omega)\,n_j(x)\right] dS\,(x) = \int_V \delta(x-\xi)\bar{u}_k(\xi,\omega)\,dV(x)$$

(5.31)

The equation is now transformed into an equation involving the known incident wave in a similar fashion as what was done for the anti-plane motion case.

$$\bar{u}^i_k(\xi,\omega) = \int_S \left[\sigma^*_{mjk}(x,\xi,\omega)\bar{u}_m(x,\omega) - u^*_{mk}(x,\xi,\omega)\bar{\sigma}_{mj}(x,\omega)\right] n_j(x)\,dS(x) + \bar{u}_k(\xi,\omega)$$

(5.32)

$\bar{u}^i_k(\xi,\omega)$ = Fourier transform of incident wave displacement at ξ

The integrals in equation (5.32) includes functions that have singularities as x approaches ξ; equation (5.33) is written to include the effect of these singularities:

$$\bar{u}^i_k(\xi,\omega) = \int_S \left[\sigma^*_{mjk}(x,\xi,\omega)\bar{u}_m(x,\omega) - u^*_{mk}(x,\xi,\omega)\bar{\sigma}_{mj}(x,\omega)\right] n_j(x)\,dS(x)$$
$$- \beta_{mk}\bar{\sigma}_{mj}(\xi,\omega)\,n_j(x) + \bar{u}_m(\xi,\omega)\left[\delta_{mk} + \alpha_{mk}\right]$$

(5.33)

β_{mk} = singularity in $u^*_{mk}(x,\xi,\omega)$

α_{mk} = singularity in $\sigma^*_{mjk}(x,\xi,\omega)\,n_j(x)$

The equation is further simplified by replacing the stress terms by tractions, where tractions are calculated from equation (5.34)

$$p_m(x) = \sigma_{mj}(x) * n_j(x) \tag{5.34}$$

$p_m(x)$ = traction in m direction

Equation (5.33) then becomes the in-plane boundary integral equation:

$$\bar{u}^i_k(\xi,\omega) = \int_S \left[p^*_{mk}(x,\xi,\omega)\bar{u}_m(x,\omega) - u^*_{mk}(x,\xi,\omega)\bar{p}_m(x,\omega) \right] dS(x) - \beta_{mk}\bar{p}_m(\xi,\omega) + \bar{u}_m(\xi,\omega)\left[\delta_{mk} + \alpha_{mk}\right] \tag{5.35}$$

5.2.1: In-Plane Influence Function

The Green's function for plane waves in an elastic material is (Banerjee and Butterfield, 1981):

$$u^*_{jk}(x,\xi,\omega) = \frac{i}{4\rho}\left(\frac{1}{C_s^2}\,\delta_{jk} H_0^{(1)}(k_s r) - \frac{1}{\omega^2}\,\frac{\partial^2}{\partial x_j \partial x_k}\left\{ H_0^{(1)}(k_p\, r) - H_0^{(1)}(k_s\, r) \right\} \right) \tag{5.36}$$

k_p = compression wave number (ω/C_p)

k_s = shear wave number (ω/C_s)

The influence function for a viscoelastic material is derived by applying the correspondence principle to the compression and shear velocities:

$$u^*_{jk}(x,\xi,\omega) = \frac{i}{4\rho}\left(\frac{1}{(C^*_s)^2}\,\delta_{jk}H_0^{(1)}(k^*_s r) - \frac{1}{\omega^2}\,\frac{\partial^2}{\partial x_j \partial x_k}\left\{H_0^{(1)}(k^*_p r) - H_0^{(1)}(k^*_s r)\right\}\right) \tag{5.37}$$

k^*_p = complex compression wave number (ω/C^*_p)

k^*_s = complex shear wave number (ω/C^*_s)

Equation (5.37) is further modified so that the displacement influence function is uncoupled into two components, P_{jk} and S_{jk}, where P_{jk} is the contribution of the compression wave and S_{jk} the contribution of the shear wave (Altay, 1986). Uncoupling equation (5.37) into the two components results in:

$$u^*_{jk}(x,\xi,\Omega) = \frac{1}{(\lambda + 2\mu)^*}\,P_{jk} + \frac{1}{\mu^*}\,S_{jk} \tag{5.38}$$

where

$$P_{jk} = \frac{-i}{4(k^*_p)^2}\left[H_0^1(k^*_p r)\right]_{,jk}$$

$$S_{jk} = \frac{i}{4}\left(\frac{1}{(k^*_s)^2}\left[H_0(k^*_s r)\right]_{,jk} + \delta_{jk}H_0^{(1)}(k^*_s r)\right)$$

After evaluating the derivatives in equation (5.38), the expressions for P_{jk} and S_{jk} are:

$$P_{jk} = \delta_{jk}\left[\frac{i}{8}\,(H_2^{(1)}(k^*_p r) + H_0^{(1)}(k^*_p r)) - \frac{1}{2\pi(k^*_p r)^2}\right] + \left[-\frac{i}{4}\,H_2^{(1)}(k^*_p r) + \frac{1}{\pi(k^*_p r)^2}\right]\frac{r_j r_k}{r^2} \tag{5.39}$$

$$S_{jk} = \delta_{jk}[\frac{i}{8}(H_0^{(1)}(k_s^*r) - H_2^{(1)}(k_s^*r)) + \frac{1}{2\pi(k_s^*r)^2}] + [+\frac{i}{4}H_2^{(1)}(k_s^*r) - \frac{1}{\pi(k_s^*r)^2}]\frac{r_j r_k}{r^2} \tag{5.40}$$

$$r_j = x_j - \xi_j$$
$$r_k = x_k - \xi_k$$
$$r = \left[(x_1 - \xi_1)^2 + (x_3 - \xi_3)^2\right]^{1/2}$$

P_{jk} and S_{jk} can further be simplified to:

$$P_{jk} = \delta_{jk}\, p_1 + p_2 \frac{r_j r_k}{r^2}$$

$$S_{jk} = \delta_{jk}\, s_1 + s_2 \frac{r_j r_k}{r^2} \tag{5.41}$$

where

$$p_1 = \frac{i}{8}(H_2^{(1)}(k_p^*r) + H_0^{(1)}(k_p^*r)) - \frac{1}{2\pi(k_p^*r)}$$

$$p_1 = \frac{i}{8}(H_2^{(1)}(k_p^*r) + H_0^{(1)}(k_p^*r)) - \frac{1}{2\pi(k_p^*r)^2}$$

$$s_1 = \frac{i}{8}(H_0^{(1)}(k_s^*r) - H_2^{(1)}(k_s^*r)) + \frac{1}{2\pi(k_s^*r)^2}$$

$$s_2 = +\frac{i}{4}H_2^{(1)}(k_s^*r) - \frac{1}{\pi(k_s^*r)^2}$$

The stress influence function is determined from the displacement influence function using Hooke's Law:

$$\sigma_{mj} = \lambda\delta_{mj}\varepsilon_{kk} + 2\mu\varepsilon_{mj} \tag{5.42}$$

Derivation of the stress influence function is based on an elastic material and later the correction for hysteretic damping is made through the use of complex material properties. The stress influence function is developed using the displacement influence function and Hooke's Law in terms of displacement (Altay, 1986). Since the stress influence function is equal to the stress due to a unit force, the displacement influence function is substituted for the displacement and the resulting stress is equal to the stress influence function.

$$\sigma^*_{mjk}(x,\xi,\omega) = \lambda^*\delta_{mj}u^*_{nk,n}(x,\xi,\omega) + \mu^*\left[u^*_{mk,j}(x,\xi,\omega) + u^*_{jk,m}(x,\xi,\omega)\right] \tag{5.43}$$

Carrying out the differentiation and simplifying:

$$\sigma^*_{mjk}(x,\xi,\omega) = \delta_{mj}\frac{r_k}{r}f_2 + \delta_{jk}\frac{r_m}{r}f_3 + \delta_{mk}\frac{r_j}{r}f_3 + \frac{r_m r_j r_k}{r^3}f_1 \tag{5.44}$$

$$f_1 = \frac{i}{r}\left\{\left[\frac{k_p^*}{k_s^*}\right]^2\left[2H_2^{(1)}(k_p^*r) - \frac{k_p^*r}{2}H_1^{(1)}(k_p^*r)\right] - 2H_2^{(1)}(k_s^*r) + \frac{k_s^*r}{2}H_1^{(1)}(k_s^*r)\right\}$$

$$f_2 = \frac{i}{r}\left\{\left(1 - 2\left[\frac{k_p^*}{k_s^*}\right]^2\right)\left[-\frac{k_p^*r}{4}H_1^{(1)}(k_p^*r)\right] - \frac{1}{2}\left[\frac{k_p^*}{k_s^*}\right]^2 H_2^{(1)}(k_p^*r) + \frac{1}{2}H_2^{(1)}(k_s^*r)\right\}$$

$$f_3 = \frac{i}{4r}\left\{\left(-2\left[\frac{k_p^*}{k_s^*}\right]^2\right)H_2^{(1)}(k_p^*r) + 2H_2^{(1)}(k_s^*r) - (k_s^*r)H_1^{(1)}(k_s^*r)\right\}$$

The traction influence function, which is equivalent to the traction at x in the m direction due to a point load at ξ in the k direction, can then determined as :

$$T_{mk}(x,\xi) = n_i\sigma^*_{imk} \tag{5.45}$$

Equation (5.44) then becomes:

$$T_{mk}(x,\xi,\omega) = n_m \frac{r_k}{r} f_2 + n_k \frac{r_m}{r} f_3 + \delta_{mk} \frac{n_j r_j}{r} f_3 + \frac{n_j r_m r_j r_k}{r^3} f_1 \tag{5.46}$$

5.2.1.1: Singularities

The singularity in the displacement influence function is analyzed by examining a small element, Γ, along the boundary and allowing $x \rightarrow \xi$ in a similar fashion as that for anti-plane motion (see Figure 5-3). Therefore, looking at the integral:

$$\int_{\Gamma} u^*_{mk}(x,\xi,\omega)\, \bar{p}_m(x,\omega)\, d\Gamma(x) \tag{5.47}$$

the integration is performed in polar coordinates and the radius allowed to approach 0. During the integration, the traction is assumed to remain constant . After evaluation, the integral becomes:

$$\lim \varepsilon \rightarrow 0 \int_{\Gamma} u^*_{mk}(x,\xi,\omega)\, \bar{p}_m(x,\omega)\, d\Gamma(x) = 0 \tag{5.48}$$

Next, the singularity in the integral containing the traction influence function is evaluated:

$$\lim \varepsilon \rightarrow 0 \int_{\Gamma} \bar{u}_m(x,\omega)\, p^*_{mk}(x,\xi,\omega)\, d\Gamma(x) \tag{5.49}$$

Equation (5.49) is more complicated because of the complexity of the $p^*_{mk}(x,\xi,\omega)$ term. Once again, integration is taken over a small area with radius ε allowed to approach 0 and $\bar{u}_m(x,\omega)$ held constant. The integral simplifies to

$$\lim_{\varepsilon\to 0} \int_\Gamma \bar{u}_m(x,\omega)\, p^*_{mk}(x,\xi,\omega)\, d\Gamma(x) = \frac{-\bar{u}_k(\xi,\omega)}{2\pi} \Delta\theta \tag{5.50}$$

where Δθ is the exterior angle of the boundary at ξ

Including the effect of the singularities, the boundary integral equation, equation (5.35), for in-plane body waves is:

$$\bar{u}^i_k(\xi,\omega) = \int_S \left[p^*_{mk}(x,\xi,\omega)\bar{u}_m(x,\omega) - u^*_{mk}(x,\xi,\omega)\bar{p}_m(x,\omega)\right] dS(x) + \bar{u}_k(\xi,\omega)\,(1 - \frac{\Delta\theta}{2\pi}) \tag{5.51}$$

$u^*_{mk}(x,\xi)$ = displacement at ξ in the m direction due to unit force at x in the k direction

$p^*_{mk}(x,\xi)$ = traction at ξ in the m direction due to a unit force at x in the k direction

The integrals in equation (5.51) are taken equal to their Cauchy principal values.

Chapter 6
Boundary Conditions

In the problem that is being investigated, the incident wave displacements are known and the boundary conditions along two media interfaces and at the intersection point of the three media are needed. The problem of boundary conditions for anti-plane motion has already been considered (Hadley et al., 1989). To analyze all the required boundary conditions, the three material configuration shown in Figure 6-1 is required to be evaluated. Within each medium, the material is considered to be homogeneous and satisfies the boundary integral equations for both anti-plane and in-plane motion. Although corners are indicated at the intersection point, mathematically the boundary at the intersection point is considered to have a small curvature.

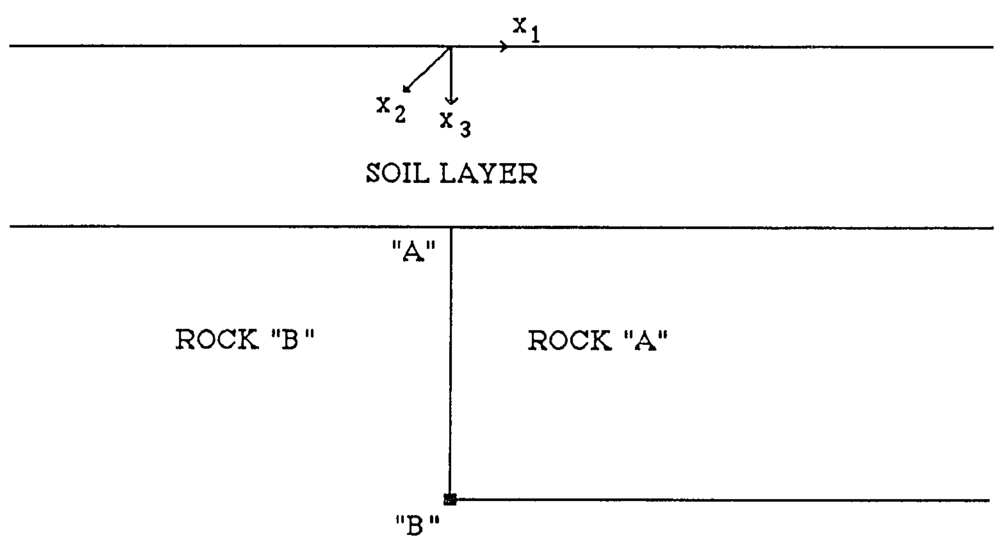


Figure 6-1: Boundary Conditions

6.1: Anti-Plane Motion

For anti-plane motion, the displacement due to the incident wave is known while the out-of-plane total displacement and stress are unknown. For the half-space domain, the incident wave displacement is:

$$u_{2_{half-space}}^{i}(x,\omega) = Ash_{half-space} \exp\{-ikt_{half-space} x_3\} \exp\{+ikx_1\} \tag{6.1}$$

For regions other than the half-space, the incident wave displacement is zero.

At the free surface, $x_3=0$, shear stress is 0. This results in one unknown, total displacement, with one equation along the free surface boundary. At interfaces between two media, welded conditions are assumed with continuous displacements and normal shear stresses, σ_{n2} . This results in along the interface of two media, two equations with two unknowns. At the intersection point of the three media, point "A" in Figure 6-1, the normal shear stress is separated into x_1 and x_3 components, σ_{12} and σ_{32} in Figure 6-2, in order to develop three unknowns, displacement and two shear stresses. The three boundary integral equations, one for each medium, and three unknowns make the problem solvable for anti-plane motion. Since rock "B" is divided into two areas along the horizontal at point "B", a similar method is used at point "B" as what is used at point "A". After the solution is attained for the stresses, the normal stress can be regained at the intersection by:

$$\sigma_{n2} = \sigma_{32} \cos\phi + \sigma_{12} \text{sine}\, \phi \tag{6.2}$$

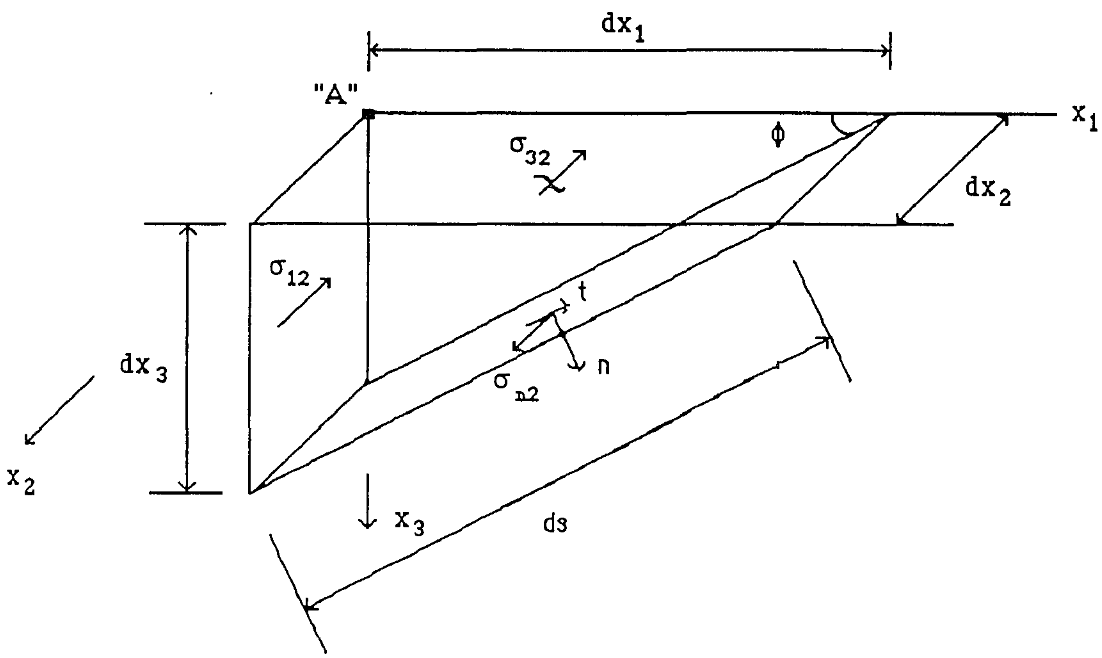


Figure 6-2: Anti-Plane Shear Stress

6.2: In-Plane Motion

Incident wave displacements required for the in-plane boundary integral equations are equal to:

$$(\bar{u}_1)^i_{half\text{-}space}(x,\omega) = \left[l_{x\,hs}\{Ap_{hs}\exp(-iks_{hs}x_3)\} + \sqrt{1-m^2_{x\,hs}}\,\{-Asv_{hs}\exp(-ikt_{hs}x_3)\}\right]\exp(ikx_1) \tag{6.3}$$

and

$$(\bar{u}_3)^i_{half\text{-}space}(x,\omega) = \left[\sqrt{1-l^2_{x\,hs}}\,\{-Ap_{hs}\exp(-iks_{hs}x_3)\} + m_{x\,hs}\,\{-Asv_{hs}\exp(-ikt_{hs}x_3)\}\right]\exp(ikx_1) \tag{6.4}$$

For in-plane motion, stresses at a point are replaced by tractions which are also known as stress resultants, equation (5.34). A boundary integral equation exists for each of the two directions (x_1, x_3) within each medium. Along the free surface, boundary conditions of zero stress are enforced ($p_1 = p_3 = 0$). At the interface of the two media, welded conditions are assumed with equal displacements and stresses for the two media. At point "A" (Figure 6-3), three media intersect .

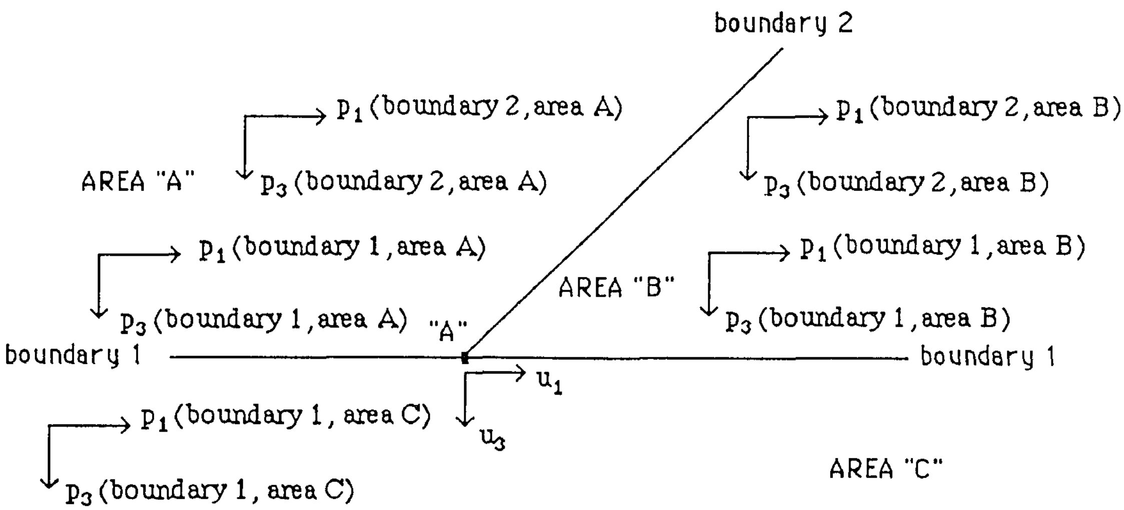


Figure 6-3: Plane Stress

Displacements at point "A" are considered equal for each of the three domains. At point "A" , tractions along a continuous straight line boundary are assumed equal in magnitude for areas bordering the boundary. Therefore, for the case shown in Figure 6-3,

$$p_1 \text{ (boundary 1, area A)} = p_1 \text{ (boundary 1, area B)} = -p_1 \text{ (boundary 1, area C)}$$
$$p_3 \text{ (boundary 1, area A)} = p_3 \text{ (boundary 1, area B)} = -p_3 \text{ (boundary 1, area C)}$$
$$p_1 \text{ (boundary 2, area A)} = -p_1 \text{ (boundary 2, area B)}$$
$$p_3 \text{ (boundary 2, area A)} = -p_3 \text{ (boundary 2, area B)} \qquad (6.5)$$

For the three areas, 6 boundary integral equations exist , 2 equations for each area, with 6 unknowns:

u_1(point A), u_3(point A), p_1(boundary 1), p_3(boundary 1), p_1(boundary 2), and p_3(boundary 2)

Therefore, the problem at point "A" and point "B" is well defined.

Chapter 7
Boundary Element Discretization

The boundary element method is a numerical method that requires the discretization of the boundary for each medium. Figure 7-1 shows how the soil bedrock configuration in Figure 1-2 is discretized for the anti-plane motion problem and Figure 7-2 the boundary discretization for the in-plane motion problem. Along the boundary $x_3=0$ and the vertical boundary at $x_1 = 0$, element segment lengths were set at 0.2*H1 which corresponds to one-tenth the minimum wave length. For the other horizontal boundaries, the element segment lengths are 0.25*H1 which is approximately one-eighth the minimum wave length. The shorter element lengths along the surface and the vertical boundary were used because it was thought that in these areas there would be either high amplifications or high variability. In the vicinity of the three area intersection points, the element length was further reduced to include for the high variance in the displacements and tractions in this area.

Within each medium, the boundary integral equations for in-plane and anti-plane motions are satisfied. For anti-plane motion, at each boundary point there is one equation for each region. For in-plane motion, at each boundary point there are two equations for each region, one in each displacement direction. Within each region, boundary points are numbered in a counter clock-wise fashion about a point within the region. Integration is taken theoretically along the entire boundary of the medium, however for the half-space the boundary integral equation along the surface at infinity is excluded since the influence function approaches 0 at these points. The truncation of the soil layers is next addressed in Chapter 8 .

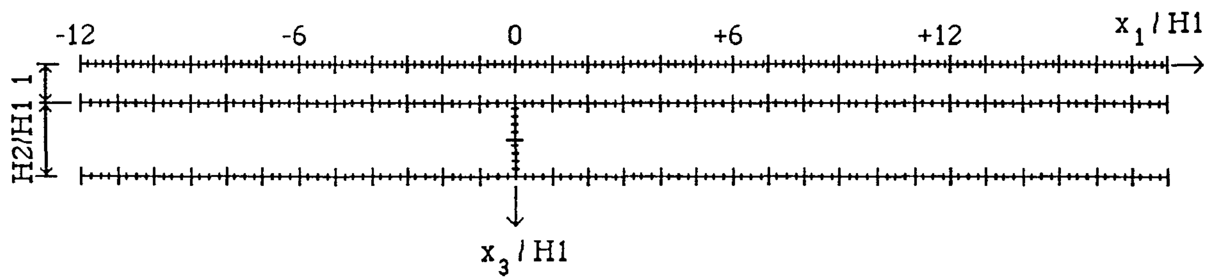


Figure 7-1: Boundary Element Discretization For Anti-Plane Motion

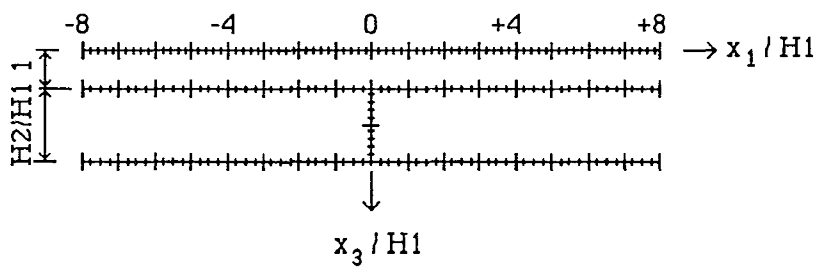


Figure 7-2: Boundary Element Discretization For In-Plane Motion

Chapter 8
Correction For Boundary Truncation

Within each medium, the boundary integral equations for plane and anti-plane motions are satisfied. Integration for the boundary integral equations is taken theoretically along the entire boundary of the medium. For a semi-infinite domain, this would require discretization of the domain to infinity or to a point which has insignificant influence to the interested area. A relaxation of this requirement is next looked at.

8.1: Layer Truncation

Figure 8-1 shows a medium which is unconfined in the x_1 direction. According to the boundary integral equations, integration should be taken for surfaces S_1 and S_2 from $x_1 = -\infty$ to $x_1 = +\infty$.

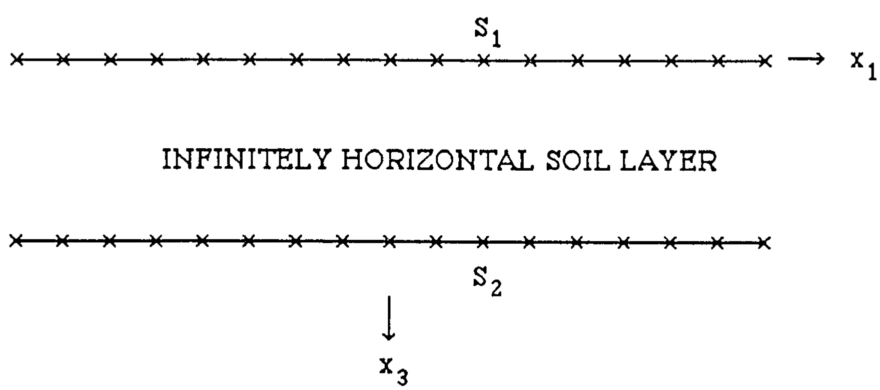


Figure 8-1: Discretization of an Infinitely Horizontal Soil Layer

In order to limit the area of discretization, imaginary boundaries are drawn at a distance far enough from the scattering area so that the displacements and tractions are not effected by the scatterer. The scatterer is an inhomogeneity in the soil profile which causes the motion to differ from the motion of a one-dimensional analysis. Therefore, along the imaginary boundaries the displacements and tractions are assumed to be equal to the results of a one-dimensional solution, Figure 8-2 (Hadley, et al., 1989). With this approach, the infinite domain is replaced by a closed domain and the boundary discretization is made only along S_1, S_2, S'_{side} and S''_{side} .

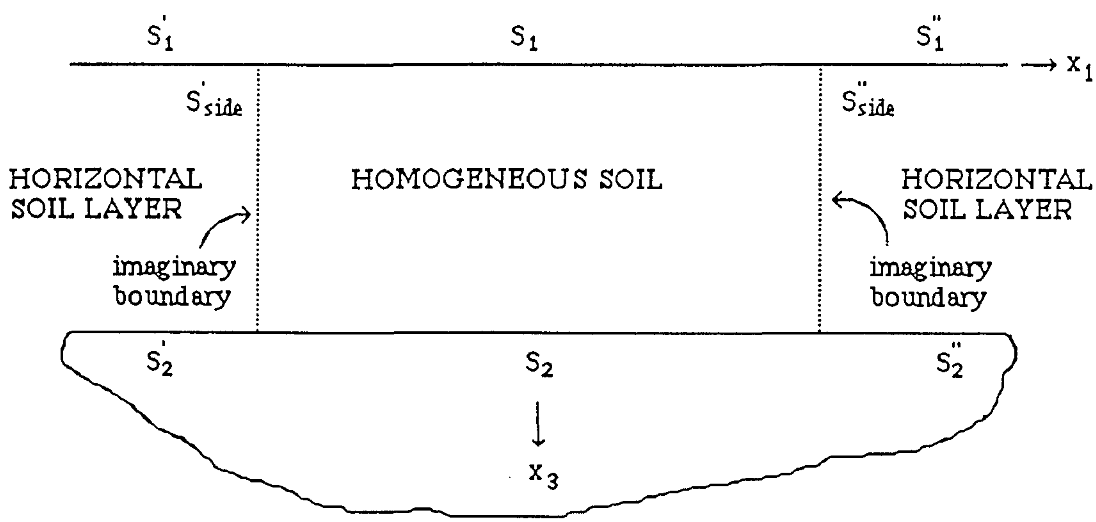


Figure 8-2: Discretization Limits

8.2: Half-Space Truncation

The boundary for the half-space should be theoretically discretized from $-\infty$ to $+\infty$, however in actuality the discretization of the boundary is limited to $x_1 = A$ to $x_1 = B$, Figure 8-3. For accurate results, A and B should be taken as large numbers in order to incorporate the influence of the outer regions to the interested area. The option of discretizing to large outer-limits is a time consuming calculation, however another alternative is used here to include the effect of this truncated region. The correction is primarily analytic and therefore a fast solution. The correction is made for the truncated region by setting the half-space boundary horizontal and assuming: (1) one-dimensional tractions and displacements in the truncated region and (2) an elastic half-space.

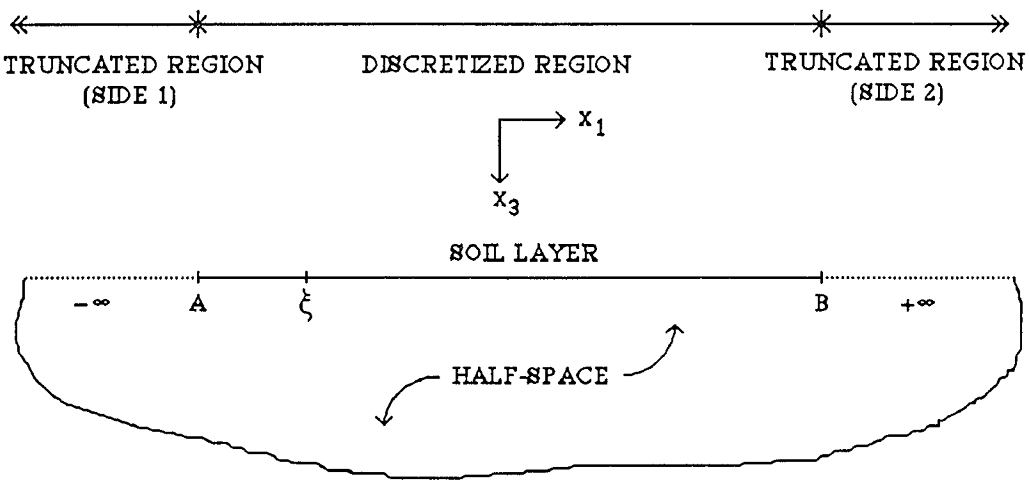


Figure 8-3: Discretization of a Half-Space

8.2.1: Anti-Plane Half-Space Correction

The boundary integral equation for the half-space is:

$$\bar{u}_2^i(\xi,\omega) = \int_{+\infty}^{-\infty} \bar{u}_2(x,\omega)\,\frac{\partial u_{22}^*}{\partial n}(x,\xi,\omega)\,dS(x) - \int_{+\infty}^{-\infty} u_{22}^*(x,\xi,\omega)\frac{\partial \bar{u}_2}{\partial n}(x,\omega)\,dS(x) + \bar{u}_2(\xi,\omega)\left(1-\frac{\Delta\theta}{2\pi}\right)$$

(8.1)

The boundary is chosen horizontal so that the traction influence function, which is a function of $\frac{dr}{dn}$, is equal to zero. Therefore, the first integral does not have to be considered. Also, because the normal is constant and in the direction of $-x_3$ the analysis is simplified. Due to this, the derivative with respect to its normal can now be taken with respect to x_3. The shear strain in the truncated regions is assumed equal to the one-dimensional shear strain which for a constant depth is equal to the product of a constant and $\exp(ikx_1)$. Since the geometries and/or properties of the soil profiles may be different for the + and - sides, the one-dimensional solutions for the tractions may be different from side to side. Therefore, any correction should consider the difference in the shear strains between the two sides. Equation (8-1) can be rewritten for the half-space as:

$$\bar{u}_2^i(\xi) =$$

$$\int_{+\infty}^{B} \frac{i}{4} H_0^{(1)}(k_{s\,hs} r) \frac{\partial \bar{u}_{2\,side\,2}(x_1=0, x_3=H_{total}, \omega)}{\partial x_3} \exp(ikx_1)\, dS(x) +$$

$$\int_{A}^{-\infty} \frac{i}{4} H_0^{(1)}(k_{s\,hs} r) \frac{\partial \bar{u}_{2\,side\,1}(x_1=0, x_3=H_{total}, \omega)}{\partial x_3} \exp(ikx_1)\, dS(x) -$$

$$\int_{B}^{A} \frac{i}{4} H_0^{(1)}(k_{s\,hs} r) \frac{\partial \bar{u}_2(x,\omega)}{\partial n}\, dS(x) + \bar{u}_2(\xi)\left(1 - \frac{\Delta\theta}{2\pi}\right)$$

(8.2)

So that the exponential terms have a similar argument as the Hankel function, the integrals for the truncated regions are put into terms of r and the wave number k is replaced by $k_{s\,hs} m_{x\,hs}$ which is the product of the half-space shear wave velocity and direction cosine.

HALF–SPACE CORRECTION =

$$+\frac{i}{4}*\left[\begin{aligned}&\frac{\partial\bar{u}_{2\,side\,1}}{\partial x_3}(x_1=0,x_3=H_{total},\omega)\exp(ik_{s\,hs}m_{x\,hs}\xi)\int_{\xi-A}^{\infty}H_0^{(1)}(k_{s\,hs}r)\exp(-ik_{s\,hs}m_{x\,hs}r)\,dr\\&+\frac{\partial\bar{u}_{2\,side\,2}}{\partial x_3}(x_1=0,x_3=H_{total},\omega)\exp(ik_{s\,hs}m_{x\,hs}\xi)\int_{B-\xi}^{\infty}H_0^{(1)}(k_{s\,hs}r)\exp(+ik_{s\,hs}m_{x\,hs}r)\,dr\end{aligned}\right]$$

(8.3)

Next, trigonometric terms are substituted for the exponential functions. For an elastic half-space the argument of the Hankel function is real. With these conditions the integral of the product of the Hankel function and trigonometric function can be analytically solved with the boundary limits of r=0 to infinity from integral solution tables (Gradshteyn and Ryzhik, 1994). This solution can then be corrected numerically for the bounds of the integrals specified in the half-space correction formula. Equation (8.3) becomes:

HALF–SPACE CORRECTION =

$$+\frac{i}{4}\exp(ik_{s\,hs}m_{x\,hs}\xi)*$$

$$\left[\begin{aligned}&\left[\frac{\partial(\bar{u}_2)_{side\,1}}{\partial x_3}(x_1=0,x_3=H_{total},\omega)\right]*\\&\left\{\frac{1}{k_{s\,hs}\sqrt{1-m_{x\,hs}^2}}\left(1+\frac{2(\pi/2-\theta(incident))}{\pi}\right)-\int_{r=0}^{\xi-A}H_0^{(1)}(k_{s\,hs}r)\exp(-ik_{s\,hs}m_{x\,hs}r)\,dr\right\}\\&+\left[\frac{\partial(\bar{u}_2)_{side\,2}}{\partial x_3}(x_1=0,x_3=H_{total},\omega)\right]*\\&\left\{\frac{1}{k_{s\,hs}\sqrt{1-m_{x\,hs}^2}}\left(1-\frac{2(\pi/2-\theta(incident))}{\pi}\right)-\int_{r=0}^{B-\xi}H_0^{(1)}(k_sr)\exp(+ik_sm_xr)\,dr\right\}\end{aligned}\right]$$

(8.4)

In order to evaluate the significance of the corrections for truncation, the problem of a single homogeneous soil layer resting on a half-space is investigated considering various half-space incidence angles and an incident SH wave at ks1*H1 = 1.5708 . The soil profile is the same as the one in the chapter 10, Figure 10-5, where the solution including both the layer and half-space truncation corrections is given . Figure 8-4 shows the results if no correction is included for truncation. The results at the surface die off which is understandable since no truncation correction is equivalent to saying that the values outside the area of discretization have zero displacement and strain. Figure 8-5 considers the solution if only layer correction is included. This solution no longer dies off at the ends of the discretized area, however the results are overestimated. The results including both corrections are shown in Figure 10-6 . These results which incorporate the truncation corrections show good agreement with the analytical solution and illustrate the need for these corrections.

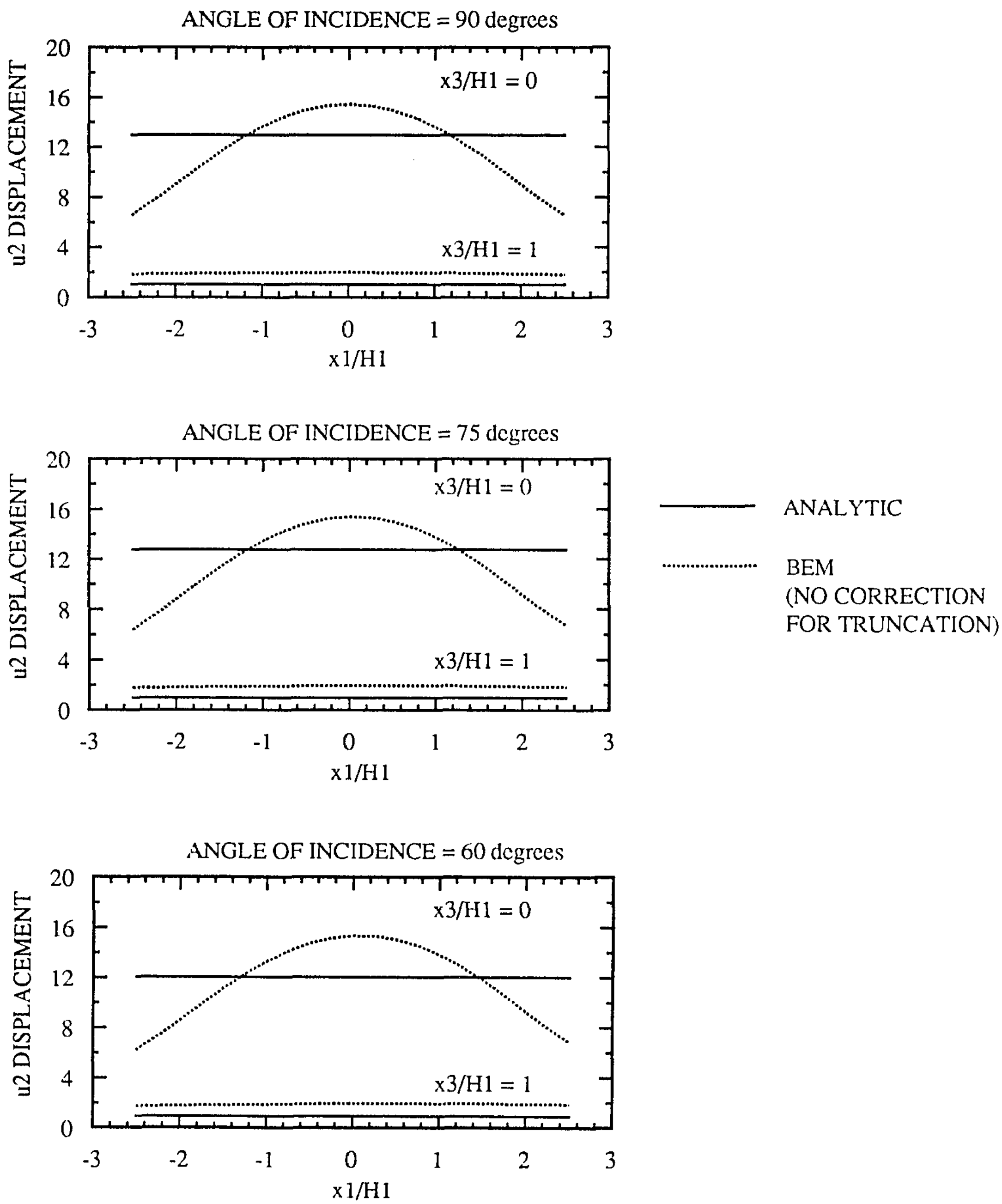


Figure 8-4: Single Layer on an Elastic Half-Space-
u2 Displacement Due to an Incident SH Wave using a
Dimensionless Frequency = 1.5708, and with
No Correction For Truncation

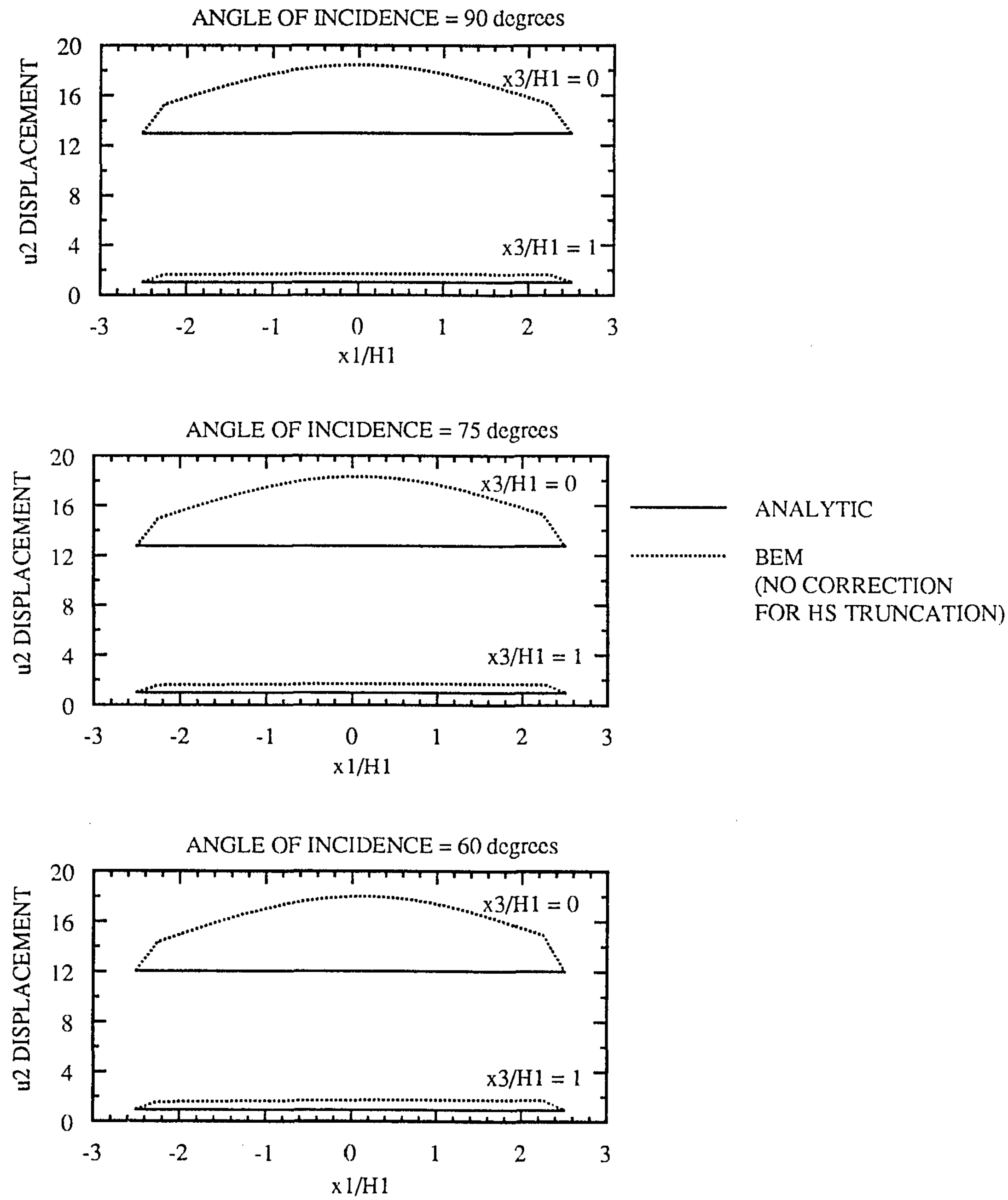


Figure 8-5: Single Layer on an Elastic Half-Space-
u2 Displacement Due to an Incident SH Wave using a
Dimensionless Frequency = 1.5708, and with
No Correction For Half-Space Truncation

8.2.2: In-Plane Half-Space Correction

The correction for truncation in the half-space for in-plane motion is made by setting the half-space boundary horizontal and assuming the displacements and tractions outside of the discretized boundary equal to the displacements and tractions from a one-dimensional analysis. Therefore, the discretized area needs to extend far enough away from the area which causes scattering so that the additional displacement due to scattering is minimal. The correction for in-plane half-space correction is more complicated than the one for anti-plane motion and requires modification of both the displacement and traction integrals.

Correction For Traction:

The traction integral,

$$\int_{+\infty}^{-\infty} \left[u_{ik}^{*}(x,\xi,\omega)\,\mu_{hs}\right]\left[\frac{\bar{p}_i(x,\omega)H1}{\mu_{hs}}\right] d(x_1/H1) \tag{8.5}$$

needs to be corrected for the area which is not included in the limits of the integration. Calling the boundary points of the discretized half-space boundary "A" ($-x_1$ side) and "B" ($+x_1$ side), the correction to the traction integral is:

$$\int_{+\infty}^{B} \left[u_{ik}^{*}(x,\xi,\omega)\,\mu_{hs}\right]\left[\frac{\bar{p}_i(x,\omega)H1}{\mu_{hs}}\right] d(x_1/H1) + \int_{A}^{-\infty} \left[u_{ik}^{*}(x,\xi,\omega)\,\mu_{hs}\right]\left[\frac{\bar{p}_i(x,\omega)H1}{\mu_{hs}}\right] d(x_1/H1) \tag{8.6}$$

Within these limits of integration, the one-dimensional solution for traction is assumed. Since the one-dimensional solution for a given depth is the product of a constant and $\exp(ikx_1)$, the constant can be taken outside of the integral:

$$\left[\frac{\bar{p}_{i\,side\,2}(x_1=0,x_3=H_{total},\omega)H1}{\mu_{hs}}\right]\int_{+\infty}^{B} \exp(ikx_1)\left[u_{ik}^{*}(x,\xi,\omega)\,\mu_{hs}\right] d(x_1/H1) +$$
$$\left[\frac{\bar{p}_{i\,side\,1}(x_1=0,x_3=H_{total},\omega)H1}{\mu_{hs}}\right]\int_{A}^{-\infty} \exp(ikx_1)\left[u_{ik}^{*}(x,\xi,\omega)\,\mu_{hs}\right] d(x_1/H1) \tag{8.7}$$

With the assumption of the horizontal boundary, u^*_{ik} becomes:

for i = 1:

$$u^*_{1k}(x,\xi,\omega)\,\mu = \delta_{1k}\left[\begin{array}{l}\dfrac{(1-2\nu_{hs})}{2(1-\nu_{hs})}*\left\{\dfrac{i}{8}\left[H_0^{(1)}(k_{p\,hs}r) - H_2^{(1)}(k_{p\,hs}r)\right] + \dfrac{1}{2\pi(k_{p\,hs}H1)^2\left(\dfrac{r}{H1}\right)^2}\right\} \\ \quad + \left\{\dfrac{i}{8}\left[H_0^{(1)}(k_{s\,hs}r) + H_2^{(1)}(k_{s\,hs}r)\right] - \dfrac{1}{2\pi(k_{s\,hs}H1)^2\left(\dfrac{r}{H1}\right)^2}\right\}\end{array}\right] \tag{8.8}$$

for i = 3:

$$u^*_{3k}(x,\xi,\omega)\,\mu = \delta_{3k}\left[\begin{array}{l}\dfrac{(1-2\nu_{hs})}{2(1-\nu_{hs})}*\left\{\dfrac{i}{8}\left[H_0^{(1)}(k_{p\,hs}r) + H_2^{(1)}(k_{p\,hs}r)\right] - \dfrac{1}{2\pi(k_{p\,hs}H1)^2\left(\dfrac{r}{H1}\right)^2}\right\} \\ \quad + \left\{\dfrac{i}{8}\left[H_0^{(1)}(k_{s\,hs}r) - H_2^{(1)}(k_{s\,hs}r)\right] + \dfrac{1}{2\pi(k_{s\,hs}H1)^2\left(\dfrac{r}{H1}\right)^2}\right\}\end{array}\right] \tag{8.9}$$

The integrals for the half-space correction due to traction are further modified in a manner similar to that discussed for the anti-plane half-space correction. The wave number is replaced by the shear wave number and the integrals are made to be a function of radial distance from the source point:

HALF-SPACE CORRECTION FOR TRACTION =

$$\exp(ik\xi) * \left[\begin{array}{l} + \left(\dfrac{\bar{p}_1(x_1=0,x_3=H_{total},\omega)\,H1}{\mu_{hs}}\right)_{side\,1}\displaystyle\int_{\xi-A}^{+\infty} e^{-ikH1\left(\frac{r}{H1}\right)}(u^*_{1k}\,\mu_{hs})\,d\left(\frac{r}{H1}\right) \\ + \left(\dfrac{\bar{p}_2(x_1=0,x_3=H_{total},\omega)\,H1}{\mu_{hs}}\right)_{side\,1}\displaystyle\int_{\xi-A}^{+\infty} e^{-ikH1\left(\frac{r}{H1}\right)}(u^*_{2k}\,\mu_{hs})\,d\left(\frac{r}{H1}\right) \\ + \left(\dfrac{\bar{p}_1(x_1=0,x_3=H_{total},\omega)\,H1}{\mu_{hs}}\right)_{side\,2}\displaystyle\int_{B-\xi}^{+\infty} e^{+ikH1\left(\frac{r}{H1}\right)}(u^*_{1k}\,\mu_{hs})\,d\left(\frac{r}{H1}\right) \\ + \left(\dfrac{\bar{p}_2(x_1=0,x_3=H_{total},\omega)\,H1}{\mu_{hs}}\right)_{side\,1}\displaystyle\int_{B-\xi}^{+\infty} e^{+ikH1\left(\frac{r}{H1}\right)}(u^*_{2k}\,\mu_{hs})\,d\left(\frac{r}{H1}\right)\end{array}\right] \tag{8.10}$$

Correction for Displacement:

The displacement integral is needed to be corrected for the area which is not discretized in the integral:

$$\int_{+\infty}^{-\infty} [p^*_{ik}(x,\xi,\omega)\, H1]\bar{u}_i(x)\, d(x_1/H1) \tag{8.11}$$

where

$$p^*_{ik}(x,\xi,\omega) = n_i\frac{r_k}{r}f_2 + n_k\frac{r_i}{r}f_3 + \delta_{ik}\frac{n_j r_j}{r}f_3 + \frac{n_j r_i r_j r_k}{r^3}f_1$$

Because of the horizontal boundary, the only non-zero terms for $p^*_{ik}(x,\xi,\omega)$ are $p^*_{31}(x,\xi,\omega)$ and $p^*_{13}(x,\xi,\omega)$. Therefore, the displacement half-space correction can be simplified to,

for k=1:

$$-\int_{+\infty}^{B} \frac{r_1}{r}[f_2 H1]\bar{u}_3(x,\omega)\, d(x_1/H1) - \int_{A}^{+\infty} \frac{r_1}{r}[f_2 H1]\bar{u}_3(x,\omega)\, d(x_1/H1) \tag{8.12}$$

for k=3:

$$-\int_{+\infty}^{B} \frac{r_1}{r}[f_3 H1]\bar{u}_1(x,\omega)\, d(x_1/H1) - \int_{A}^{+\infty} \frac{r_1}{r}[f_3 H1]\bar{u}_1(x,\omega)\, d(x_1/H1) \tag{8.13}$$

One-dimensional displacements are assumed in the truncated regions and the integrals are made to be a function of r, the radial distance from the source point ξ. Therefore, the half-space corrections become:

HALF-SPACE CORRECTION FOR DISPLACEMENT (k=1) =

$$\exp(ik\xi) * \left[\begin{array}{l} +\left(\bar{u}_3(x_1=0,x_3=H_{total},\omega)\right)_{side\,1} \displaystyle\int_{\xi-A}^{+\infty} e^{-ikH1\left(\frac{r}{H1}\right)}(f_2 H1)\, d\left(\frac{r}{H1}\right) \\ +\left(\bar{u}_3(x_1=0,x_3=H_{total},\omega)\right)_{side\,2} \displaystyle\int_{B-\xi}^{+\infty} e^{+ikH1\left(\frac{r}{H1}\right)}(f_2 H1)\, d\left(\frac{r}{H1}\right) \end{array} \right] \tag{8.14}$$

HALF-SPACE CORRECTION FOR DISPLACEMENT (k=3) =

$$\exp(ik\xi) * \left[\begin{array}{l} +\left(\bar{u}_1(x_1=0,x_3=H_{total},\omega)\right)_{side\,1} \displaystyle\int_{\xi-A}^{+\infty} e^{-ikH1\left(\frac{r}{H1}\right)}(f_3 H1)\, d\left(\frac{r}{H1}\right) \\ +\left(\bar{u}_1(x_1=0,x_3=H_{total},\omega)\right)_{side\,2} \displaystyle\int_{B-\xi}^{+\infty} e^{+ikH1\left(\frac{r}{H1}\right)}(f_3 H1)\, d\left(\frac{r}{H1}\right) \end{array} \right] \tag{8.15}$$

Integration for many of the functions involved in the half-space correction are analytically possible for limits of 0 and infinity from tables (Gradshteyn and Ryzhik, 1994)). The integration is then corrected for the actual limits using numerical methods for the limits of the integral. For the integration of functions that do not include an analytic solution, integration is made numerically using an integral upper bound of a large number instead of infinity.

The significance of including corrections for truncation can be observed if the problem of a single homogeneous soil layer resting on a half-space is calculated with and without the different truncation corrections. The soil profile is the same as the one given in chapter 10, Figure 10-5, with an incident SV wave at ks1*H1 = 1.5708. The solution in chapter 10 includes both the layer and half-space truncation. Figures 8-6 and 8-7 show the results if no corrections are included for truncation. The results at the surface are significantly below the analytic solution for u_1 displacement and vary significantly from the analytic u_3

displacement. Figures 8-8 and 8-9 consider the solution if only the layer correction is included. These results begin to approach the correct solution, however the results for u_1 displacement are overestimated and the u_3 displacements still show significant error. The results for the displacements which include both corrections are shown in Figure 10-7. These results show good agreement with the analytical solution and exemplify the need for the layer and half-space truncation corrections.

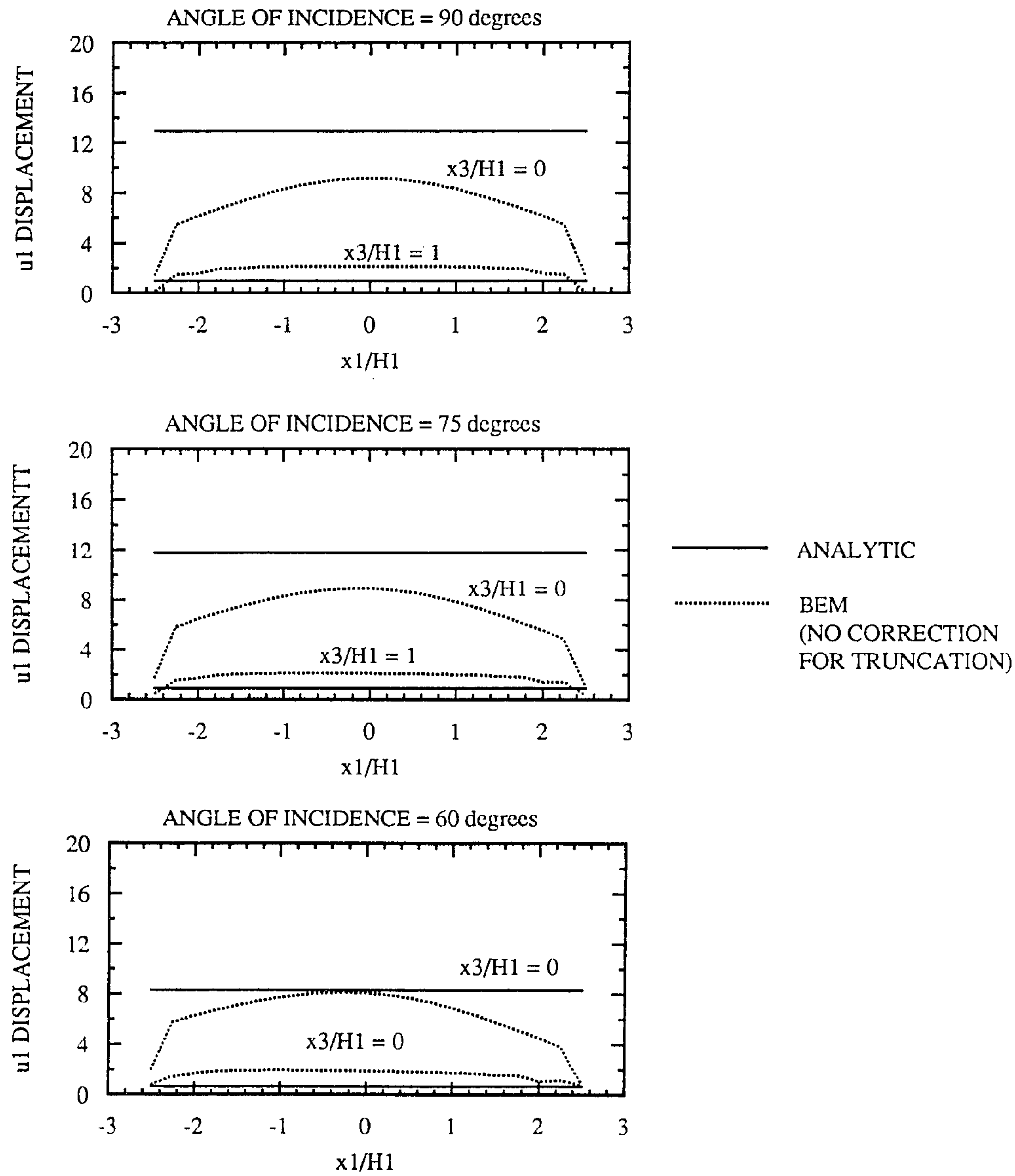


Figure 8-6: Single Layer on an Elastic Half-Space-
u1 Displacement Due to an Incident SV Wave using a
Dimensionless Frequency = 1.5708, and with
No Correction For Truncation

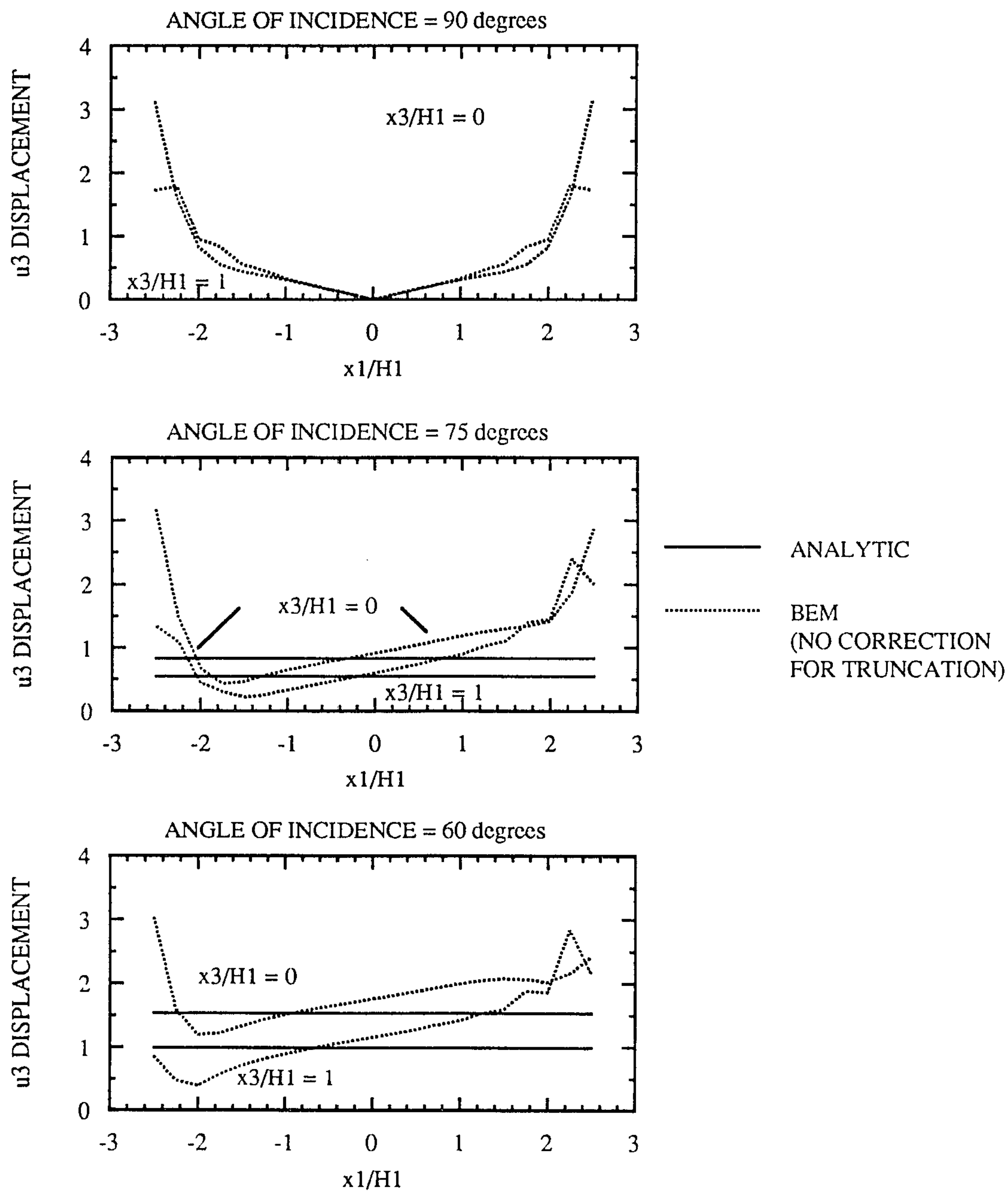


Figure 8-7: Single Layer on an Elastic Half-Space-
u3 Displacement Due to an Incident SV Wave using a Dimensionless Frequency = 1.5708, and with No Correction For Truncation

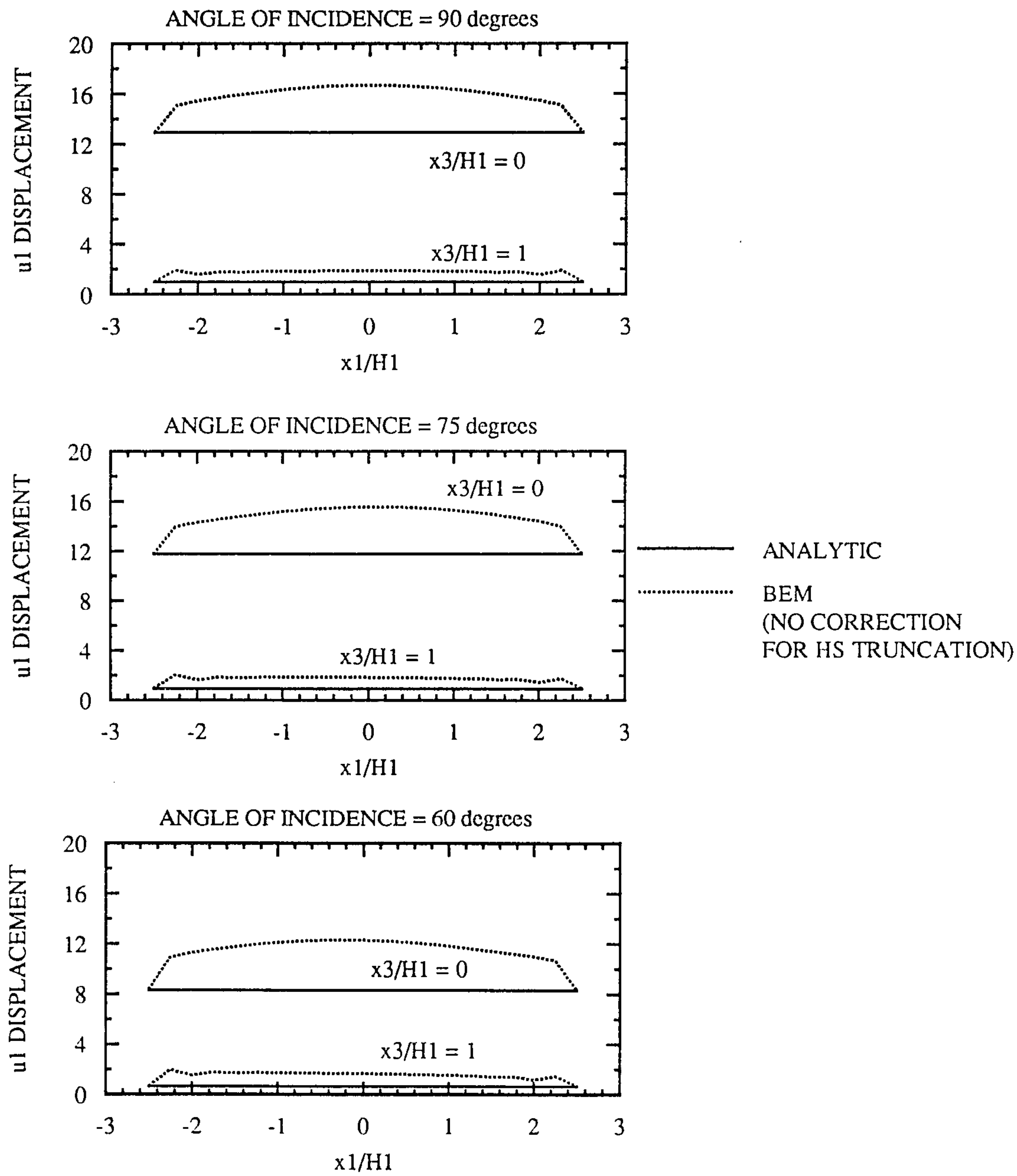


Figure 8-8: Single Layer on an Elastic Half-Space-
u1 Displacement Due to an Incident SV Wave using a
Dimensionless Frequency = 1.5708, and with
No Correction For Half-Space Truncation

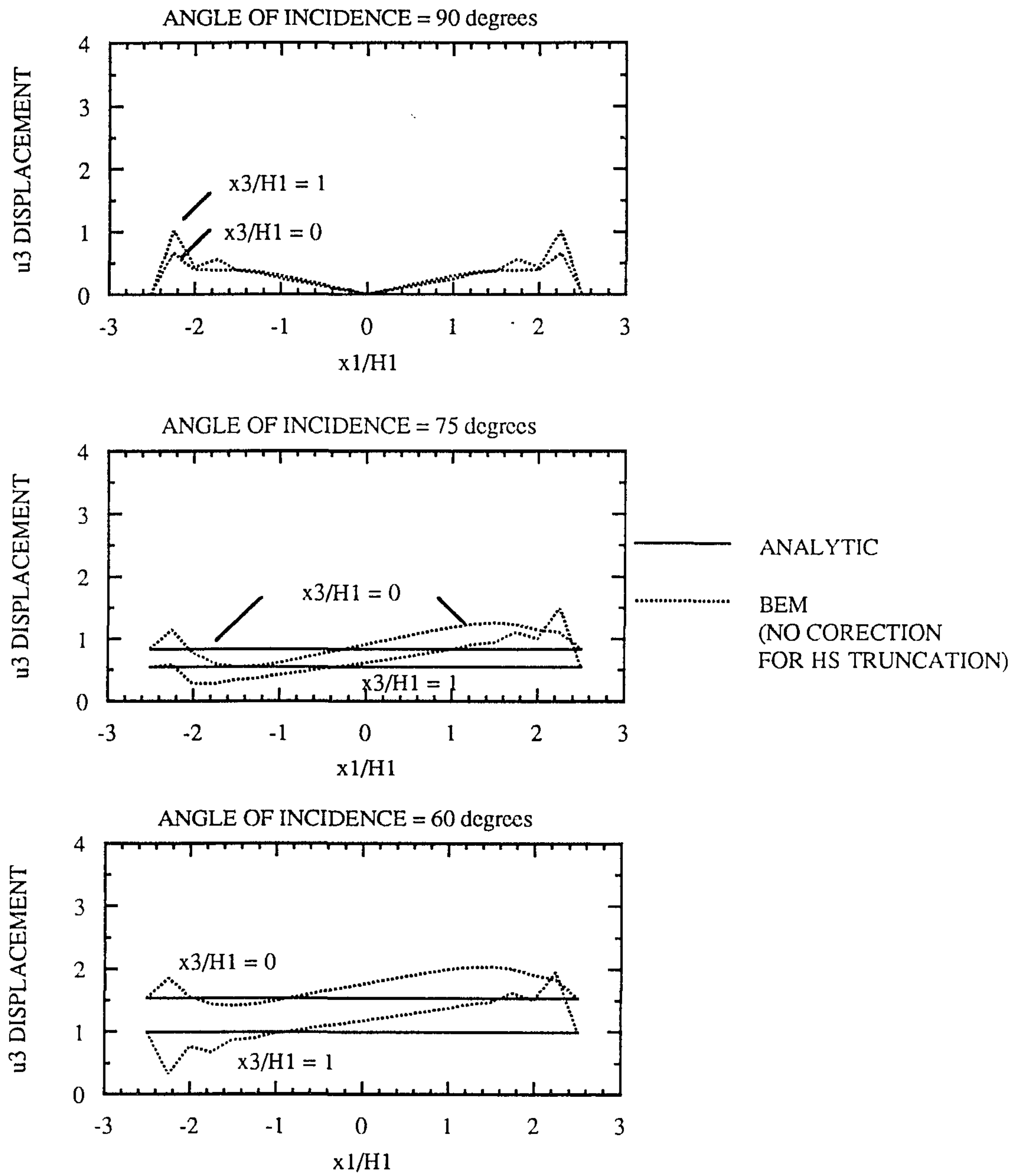


Figure 8-9: Single Layer on an Elastic Half-Space-
u3 Displacement Due to an Incident SV Wave using a
Dimensionless Frequency = 1.5708, and with
No Correction For Half-Space Truncation

Chapter 9
Numerical Techniques

In order to develop a solution using the boundary integral equation method various numerical techniques are employed. These numerical techniques are used for integration, the determination of the diagonal block elements of the traction influence matrix, and the determination of the Hankel functions of complex numbers.

9.1: Numerical Integration

The boundary integral equations include integrals that cannot in most cases be analytically solved. Instead, a numerical method must be used. The method used is the Gaussian quadrature method which allows for the integral of each boundary segment to be transformed into a summation of the products of weights and values of the function at prescribed points. The prescribed points are determined by solving for the roots of the Legendre polynomial.

The Gaussian quadrature method is defined as

$$\int_A^B f(x)\,dx \approx \sum_{i=0}^{n} f(x_i) \int_A^B l_i(x)\,dx \tag{9.1}$$

$l_i(x)$ = Lagrange polynomial

x_i = Gaussian quadrature points (roots of the Legendre polynomial)

The function f(x) in the boundary element method is equal to the product of the unknown displacement or traction and the influence function. If the unknown displacement or traction function is called u(x) and the influence function h(x), then f(x) = u(x) * h(x). The value for u(x) at the Gaussian quadrature points is assumed to be a linear function between the boundary element end points.

$$u(x_i) = u(x_A) + \left(\frac{u(x_B) - u(x_A)}{(x_B - x_A)}\right)(x_i - x_A) \tag{9.2}$$

By substituting the linear function into the integral of equation (9.1), integration of the function becomes numerically solvable.

$$\int_A^B f(x)\,dx \approx \sum_{i=0}^{n} \left[u(x_A) + \left(\frac{u(x_B) - u(x_A)}{(x_B - x_A)}\right)(x_i - x_A) \right] h(x_i) \int_A^B l_i(x)\,dx \tag{9.3}$$

or

$$\int_A^B f(x)\,dx \approx \sum_{i=0}^{n} \left[(u(x_A) + \left(\frac{u(x_B) - u(x_A)}{(x_B - x_A)}\right)(x_i - x_A) \right] h(x_i)\,(WT_i) \tag{9.4}$$

where

WT_i = Gaussian quadrature weights

A three point Gaussian quadrature is used for the numerical analysis of the boundary integral equations. The integrals involved in the boundary element method can now be put into a form of matrices. In matrix form, the boundary integral equations simplify to a matrix of influence functions and weights, [A], a vector of known incident wave displacements and truncation corrections, {y}, and a vector of unknown tractions and displacements, {x}.

$$[A]\{x\} = \{y\} \tag{9.5}$$

In order to allow for matrix [A] to be independent of the angle of incidence, all truncation corrections, which are dependent on the angle of incidence, are made to vector {y}. This allows for multiple incidence angles to be analyzed with only varying the vector {y} and leaving matrix [A] untouched.

9.2: Diagonal Traction Elements

The boundary integral equations for anti-plane motion, equation (5.22), and for in-plane motion, equation (5.51) include a singularity due to the traction influence function. In both the anti-plane and in-plane equations the singularity is equal to the exterior angle at the source point divided by 2π. Therefore, the function multiplying the total displacement in equations (5.22) and (5.51) is equal to the interior angle at the source point divided by 2π. Instead of calculating the singularity at each source point, another method is used. The values of the traction and displacement influence functions result in traction and displacement influence matrices. Each diagonal value of the traction influence matrix includes the singularity at the source point and the effect of the source point on the adjacent elements. The diagonal elements of the traction influence matrix can be determined by calculating the diagonal elements as though the problem was static and correcting for the dynamic effect of the adjacent elements (Ahmad and Banerjee, 1988). The matrix equation for the static problem is:

$$[p^*(\text{static})]\{u(\text{static})\} - [u^*(\text{static})]\{p(\text{static})\} = \{0\} \tag{9.6}$$

where

$\{u(\text{static})\}$ = static analysis displacement vector

$\{p(\text{static})\}$ = static analysis traction vector

$[u^*(\text{static})]$ = static displacement influence function matrix

$[p^*(\text{static})]$ = static traction influence function matrix

Since the singularity for each of the boundary integral equations, equations (5.22) and (5.51), is frequency independent, it is equal to the singularity of the static case. Although the singularity is frequency independent, the effect of the adjacent elements is still frequency dependent. Examining the static case, diagonal elements of the traction influence matrix can be calculated from equation (9.6) by assuming rigid body motion. For rigid body motion, the tractions are zero and equation (9.6) becomes:

$$[p^*(\text{static})]\{u(\text{static})\} = \{0\} \tag{9.7}$$

If a unit displacement is prescribed in each direction for the rigid displacement, the diagonal block elements related to source point ξ can be determined as;

if diagonal element "m" is odd:

$$p^*_{mm}(\text{static}) = -\sum_{\substack{n=1,3,5,..\\ n \neq m}}^{n=N} p^*_{mn}(\text{static}) \tag{9.8}$$

$$p^*_{m\,m+1}(\text{static}) = -\sum_{\substack{n=2,4,6,..\\ n \neq m+1}}^{n=N} p^*_{mn}(\text{static}) \tag{9.9}$$

if diagonal element "m" is even:

$$p^*_{mm}(\text{static}) = -\sum_{\substack{n=2,4,6,..\\ n \neq m}}^{n=N} p^*_{mn}(\text{static}) \tag{9.10}$$

$$p^*_{m\,m-1}(\text{static}) = -\sum_{\substack{n=1,3,5,..\\ n \neq m-1}}^{n=N} p^*_{mn}(\text{static}) \tag{9.11}$$

Where repeated subscripts in equations (9.8) through (9.11) does not imply summation. The static influence diagonal block elements include the singularity but also include the static influence of the adjacent elements to ξ, the source point. The singularity is found by reducing the static influence function by the static effect of the adjacent elements. For anti-plane motion, p^*_{mm}(static) is equal to p^*_{mm}(dynamic) since both the static and dynamic traction influence of the adjacent elements is equal to zero. However, for in-plane motion, the adjacent elements are influential. The singularity for the in-plane motion is;

for the diagonal elements:

$$[\text{ singularity @ } \xi]_{mm} = p^*_{mm}(\text{static}) \tag{9.12}$$

for the off-diagonal elements;

if m is odd:

$$[\text{ singularity @ } \xi]_{m,m+1} =$$

$$p^*_{m,m+1}(\text{static}) - \left[+C_1 C_2 u_3(r{=}S_{-1}) - C_3 C_4 \ln\left(\frac{S_{+1}}{S_{-1}}\right) u_3(r{=}0) - C_1 C_2 u_3(r{=}S_{+1}) \right] \tag{9.13}$$

if m is even:

$$[\text{ singularity @ } \xi]_{m,m-1} =$$

$$p^*_{m,m-1}(\text{static}) - \left[-C_1 C_2 u_1(r{=}S_{-1}) + C_1 C_2 \ln\left(\frac{S_{+1}}{S_{-1}}\right) u_1(r{=}0) + C_1 C_2 u_1(r{=}S_{+1}) \right] \tag{9.14}$$

where

$$C_1 = -\frac{1}{4\pi(1-\nu)}$$

$$C_2 = 1-2\nu$$

S_{-1} = length of the previous adjacent element to the source point

S_{+1} = length of the following adjacent element to the source point

The correction for the integration of the static influence function is carried out analytically in order to include the effect of unequal adjacent segment lengths. This effect is lost if Gaussian quadrature is used. The diagonal block elements are then found by adding to the singularity the dynamic influence of the adjacent elements.

$$
\begin{aligned}
&[p^*(\text{dynamic})] = [\text{ singularity @ } \xi] \; + \\
&\left[\bar{u}_i(r{=}0,\omega)\int_0^{S_{i+1}} p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) - \frac{\bar{u}_i(r{=}0,\omega)}{S_{i+1}}\int_0^{S_{i+1}} \frac{r}{H1}\, p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) \right. \\
&\qquad \left. + \frac{\bar{u}_i(r{=}S_{i+1},\omega)}{S_{i+1}}\int_0^{S_{i+1}} \frac{r}{H1}\, p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) \right] \\
&+ \\
&\left[\bar{u}_i(r{=}0,\omega)\int_0^{S_{i-1}} p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) - \frac{\bar{u}_i(r{=}0,\omega)}{S_{i-1}}\int_0^{S_{i-1}} \frac{r}{H1}\, p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) \right. \\
&\qquad \left. + \frac{\bar{u}_i(r{=}S_{i-1},\omega)}{S_{i-1}}\int_0^{S_{i-1}} \frac{r}{H1}\, p_{ik}^*(x,\xi,\omega)H1\, d\!\left(\frac{r}{H1}\right) \right]
\end{aligned}
\tag{9.15}
$$

where

k = 1 for odd row elements of p*(dynamic) matrix

k = 3 for even row elements of p*(dynamic) matrix

The Hankel functions for the adjacent elements are expanded in a series of trigonometric functions (Abramowitz and Stegun, 1972) Since only the adjacent elements are considered, not many terms are required in the series to gain reliable results.

9.3: Hankel Functions

The use of hysteretic damping in the soil amplification results in complex shear wave numbers. Therefore, a method to determine Hankel functions for complex variables is required. The Hankel function of the first kind is equal to:

$$H_n^{(1)}(z) = J_n(z) + Y_n(z) \tag{9.16}$$

where

$J_n(z)$ = Bessel function of the first kind and order n

$Y_n(z)$ =Bessel function of the second kind and order n

z = complex number (x + iy)

In lieu of the use of a series to determine the two kinds of Bessel functions, an integral method is used. The definition of the Bessel functions in terms of an integral is given as (Abramowitz & Stegun, 1974):

$$J_n(z) = \frac{1}{\pi}\int_0^{\pi} \cos(z\ \text{sine}(\theta) - n\theta)\, d\theta \tag{9.17}$$

and

$$Y_n(z) = \frac{1}{\pi}\int_0^{\pi} \text{sine}(z\ \text{sine}(\theta) - n\theta)\, d\theta - \frac{1}{\pi}\int_0^{+\infty} \left\{e^{nt} + e^{-nt}\cos(n\pi)\right\} e^{-z\sinh(t)}\, dt \qquad |\arg z| < \frac{\pi}{2} \tag{9.18}$$

The condition for the absolute argument of z to be less than π/2 is met if the hysteretic damping is less than 20% . This is satisfied for almost all soil amplification problems. Therefore, this condition is not considered to be a serious restriction to the calculation. The trigonometric functions in the integrands of equations (9.17) and (9.18) are further simplified into real (x) and imaginary (y) terms.

$$J_n(z) = \frac{1}{\pi}\int_0^{\pi} \cos\left(x\ \mathrm{sine}(\theta) - n\theta\right) \cosh\left(y\ \mathrm{sine}(\theta)\right) - i\ \mathrm{sine}\left(x\ \mathrm{sine}(\theta) - n\theta\right) \sinh\left(y\ \mathrm{sine}(\theta)\right) d\theta \tag{9.19}$$

$$Y_n(z) =$$

$$\frac{1}{\pi}\int_0^{\pi} \left[\mathrm{sine}\left(x\ \mathrm{sine}(\theta) - n\theta\right) \cosh\left(y\ \mathrm{sine}(\theta)\right) + i \cos\left(x\ \mathrm{sine}(\theta) - n\theta\right) \sinh\left(y\ \mathrm{sine}(\theta)\right)\right] d\theta -$$

$$\frac{1}{\pi}\int_0^{+\infty} \left[\left(e^{nt} + e^{-nt}\cos(n\pi)\right) e^{-x\ \sinh(t)} \cos\left(y\ \sinh(t)\right) - i\left(e^{nt} + e^{-nt}\cos(n\pi)\right) e^{-x\ \sinh(t)} \mathrm{sine}\left(y\ \sinh(t)\right)\right] dt \tag{9.20}$$

The integration is carried out using a 5 point Gaussian quadrature method. The upper limit of the second integral in equation (9.20) is set at 10.0 since this was found to be acceptable in the calculations for the Bessel function of the second kind. The Hankel function of a complex number, equation (9.16) is then the summation of the Bessel functions of the first and second kind, equations (9.19) and (9.20).

Chapter 10
Code Validation

To check the validity of the codes for in-plane and anti-plane amplification, comparisons are made with analytic solutions.

Figure 10-1 shows a comparison between the analytic and numeric amplification results of a vertically incident SH wave due to a cylindrical cavity in a full-space (Hadley et al., 1989). The cylindrical cavity has a radius R and a SH wave propagates at ks*R = 3.0 . The problem is solved numerically by dividing the cylindrical boundary into 20 equal segments. The comparison between the analytic results and the boundary element results are in the form of a polar graph where the radius is the amplification of the incident wave and the polar angle is the angular position on the cylindrical boundary. Results from the two procedures show excellent agreement.

An analytic solution is also available for the anti-plane problem of a semi-cylindrical canyon in a half-space (Trifunac, 1973). The comparison between the analytical solution and the numerical solution is shown in Figure 10-2. The half-space is discretized to $x_1/R = \pm 10.0$, where R is the radius of the semi-cylindrical canyon, and the circumference is divided into 5 degree intervals. For this problem, no correction is made for truncation. Various angles of incidence (90,75, and 60), were used with a dimensionless frequency of $ks^*R/\pi = 0.5$. Results for the different angles of incidence show good agreement. The case in which the angle of incidence is equal to 30 degrees shows the most variance from the exact solution. This is to be expected due to the greater amount of scattering that occurs in the $\pm x_1$ regions. Since the model used in the analysis is truncated at $x_1/H1 = \pm 10$, it excludes the contribution of the truncated regions which has some significance at the shallower incidence angles. Even with this truncation and shallow angle of incidence, the results are good.

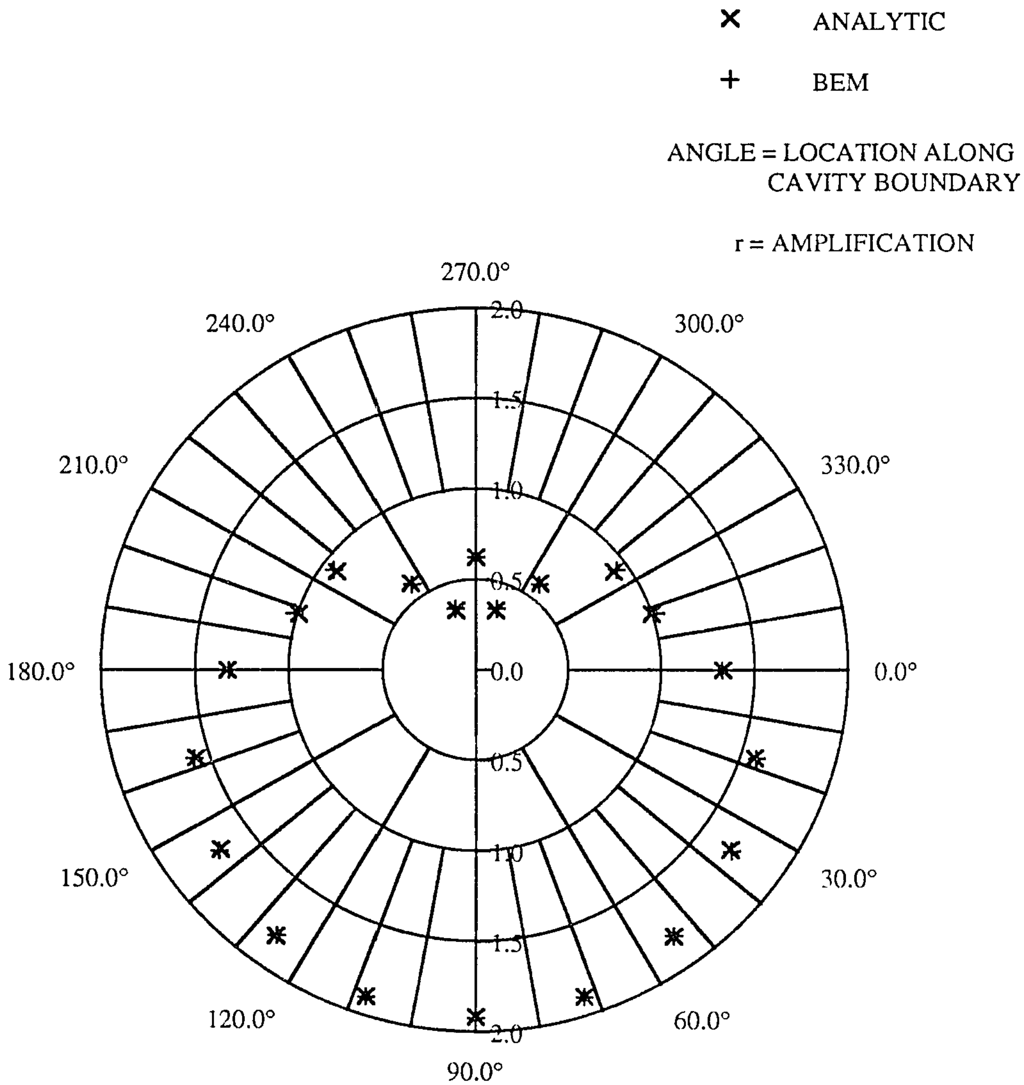


Figure 10-1: Cylindrical Cavity in an Elastic Full-Space,
u2 Displacement Due to a Vertically Incident SH Wave,
ks*R = 3.0

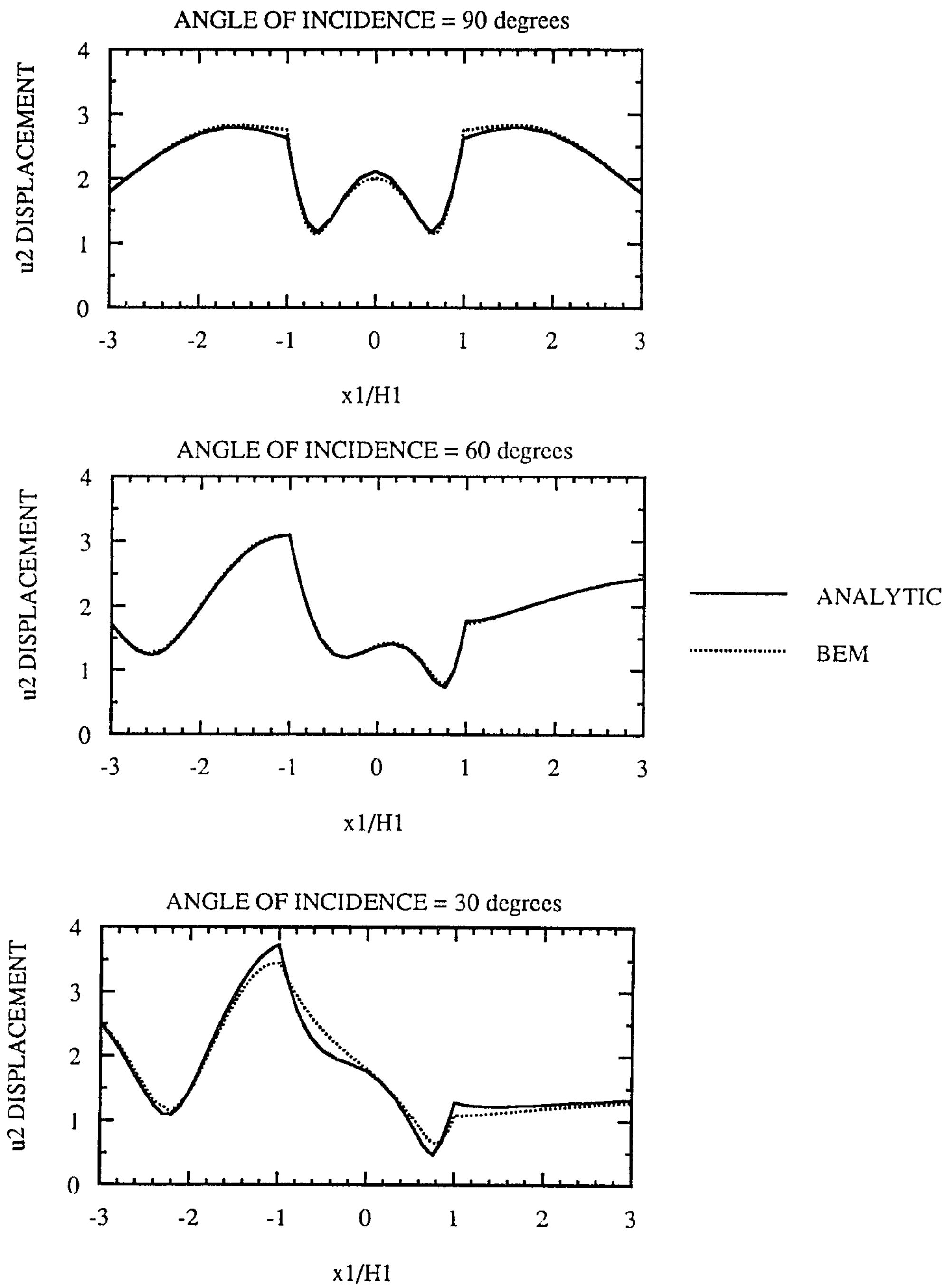


Figure 10-2: Semi-Cylindrical Canyon in an Elastic Half-Space, u2 Displacement Due to a Incident SH Wave with $ks^*R/\pi = 0.5$

Figure 10-3 is a check problem for in-plane motion in which an analytic solution is available. The figure shows an elastic inclusion in a elastic full-space subject to a horizontally propagating P wave. The inclusion is stiffer than the full-space and the material properties are given in the Figure 10-3 . The results of the analytic solution to the problem are given for a range of frequencies at specific points along the circumference of the inclusion (Dominguez and Abascal, 1989). The problem is solved numerically using two boundary element discretization schemes, one in which the circumference is divided into 28 equal elements and the other using 56 equal elements. Both cases show good agreement with the analytic results and show small variation between the two boundary element discretization schemes. Error increases at the higher frequencies, particularly for the case of $\theta = 167.14$ degrees, where more boundary elements are required to match the curvature of the inclusion boundary.

A single layer on a half-space provides a problem in which an analytic solution is available for both anti-plane (SH) and in-plane (P & SV) wave motions. The problem is shown in Figure 10-5 which also shows the boundary element discretization used to solve the problem. The problem is discretized to $x_1/H1 = \pm 2.5$ with boundary segment lengths equal to 0.25*H1 . The problem is solved for each of the wave types for a dimensionless frequency, ks(layer 1) *H1 = 1.5708, which is the natural frequency for a soil layer on a rock half-space subject to a vertically incident shear wave. Results are given for a range of angles of incidence (90, 75, 60 degrees). Soil amplification is given for an incident SH, SV, and P wave in Figures 10-6, 10-7, and 10-8, respectively and compared with the analytic solutions. Results between the analytic solution and the boundary element method are good. The boundary element results for this problem also exemplify the benefits of the truncation correction used in the solution.

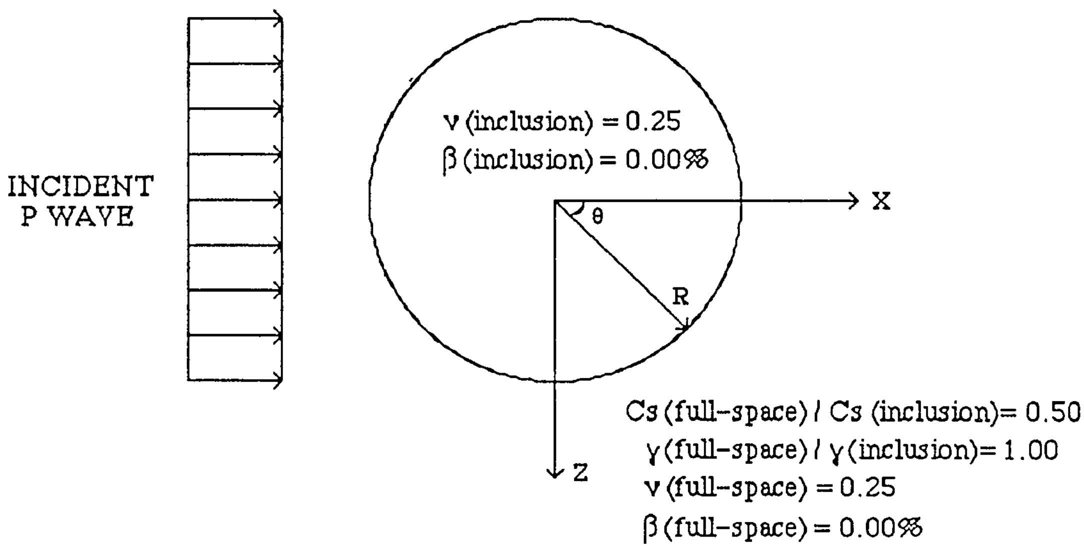


Figure 10-3: Elastic Cylindrical Inclusion
in an Elastic Full-Space

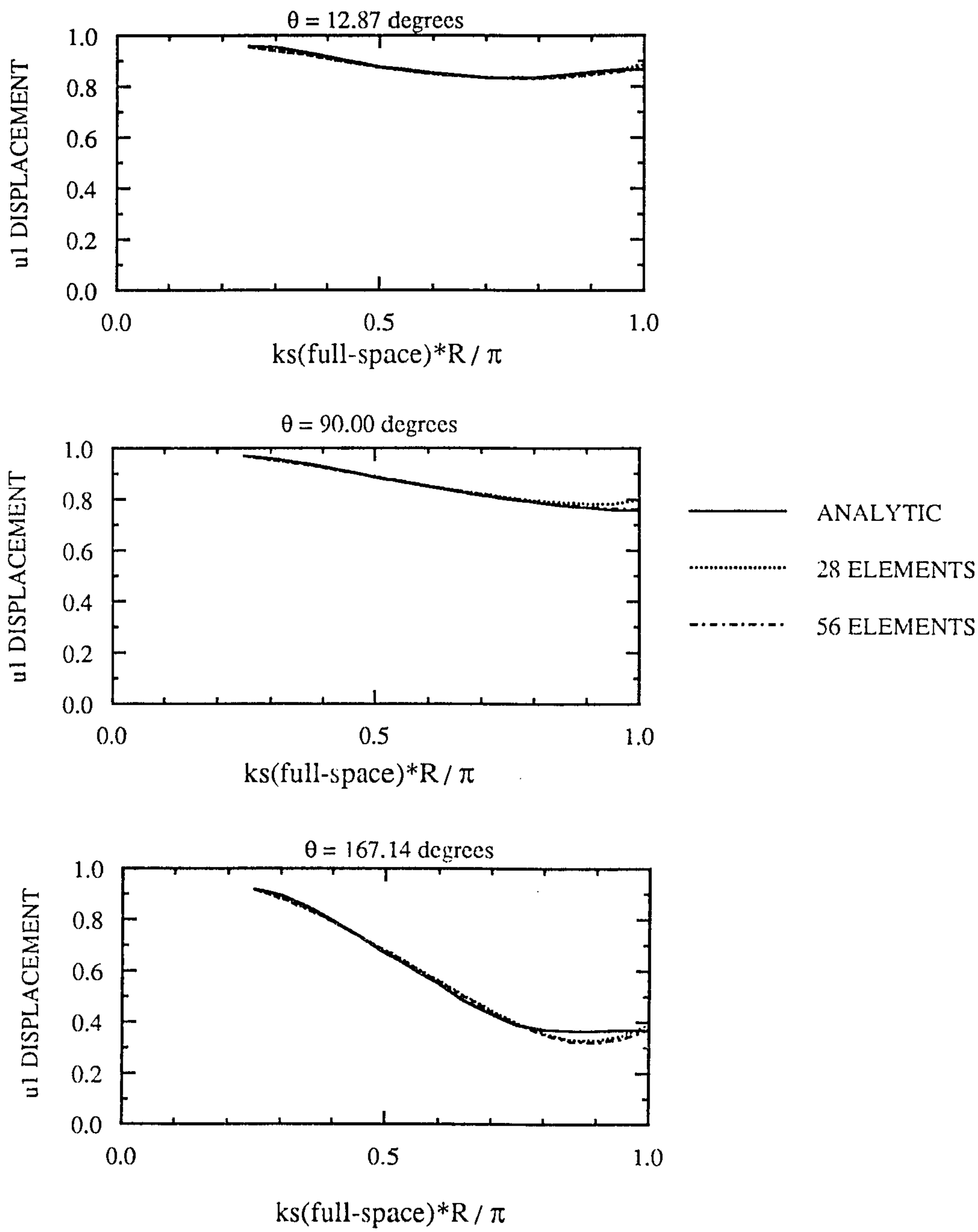
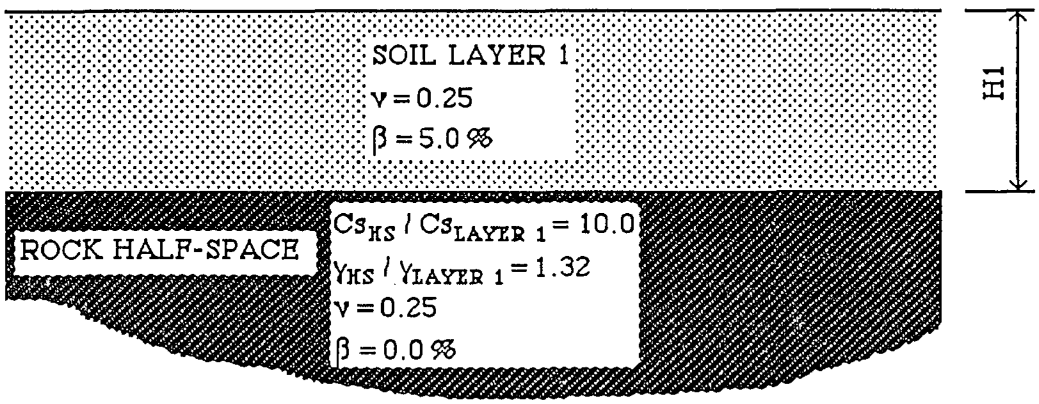
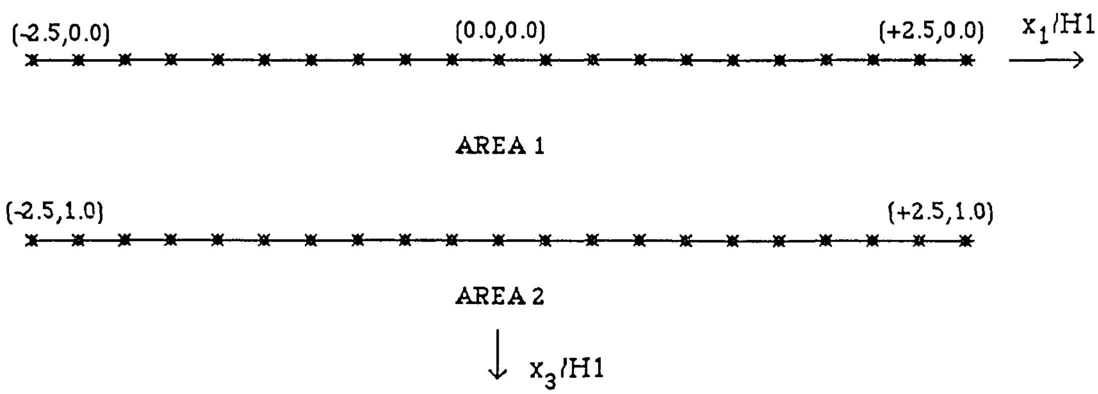


Figure 10-4: Elastic Inclusion in a Full-Space,
u1 Displacement at Specific Points
Due to an Incident P Wave



Soil Profile



Boundary Element Discretization

Figure 10-5: Single Soil Layer on a Half-Space

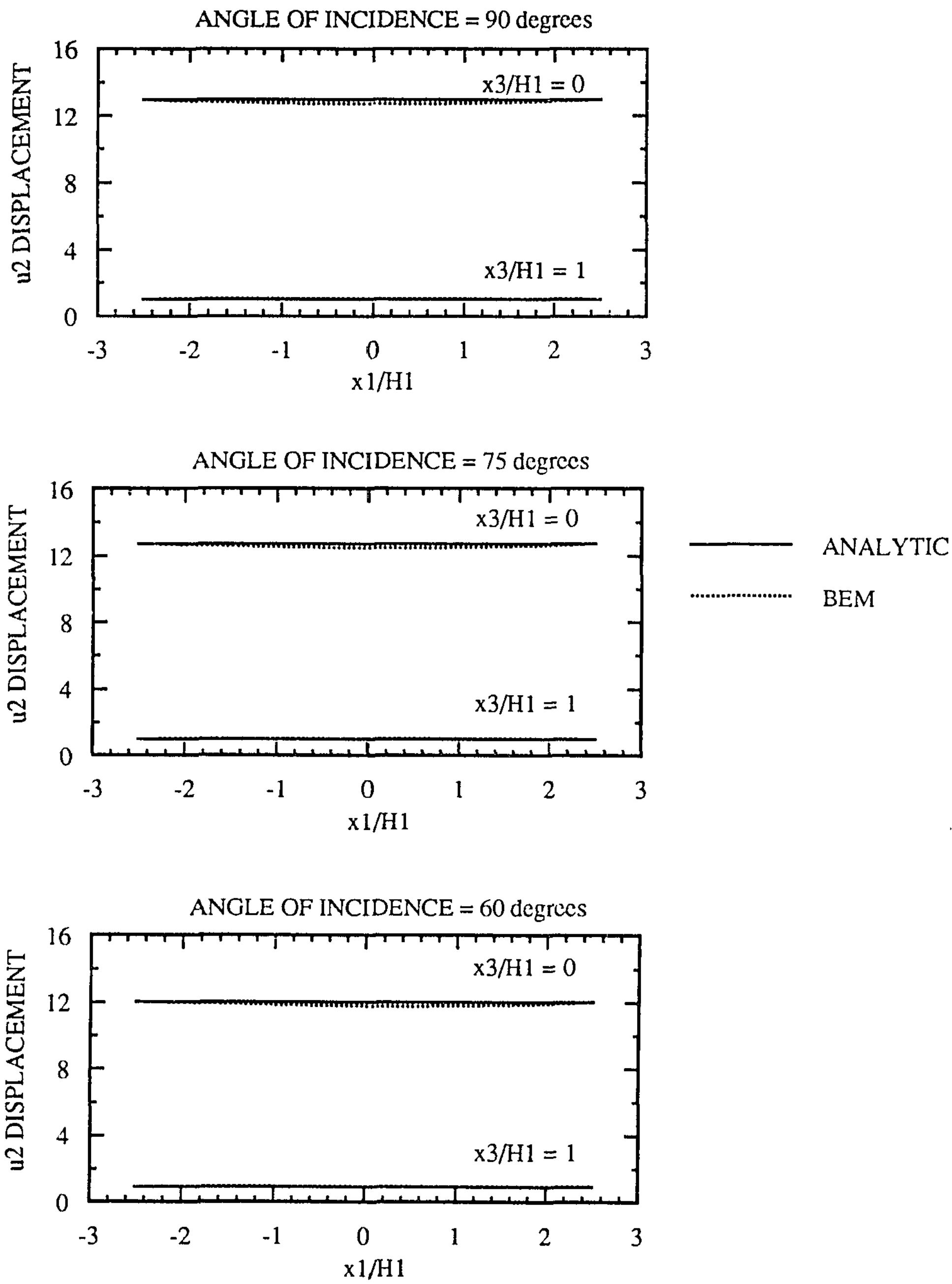


Figure 10-6: Single Layer on a Half-Space,
u2 Displacement Due to an Incident SH Wave with
ks1*H1 = 1.5708

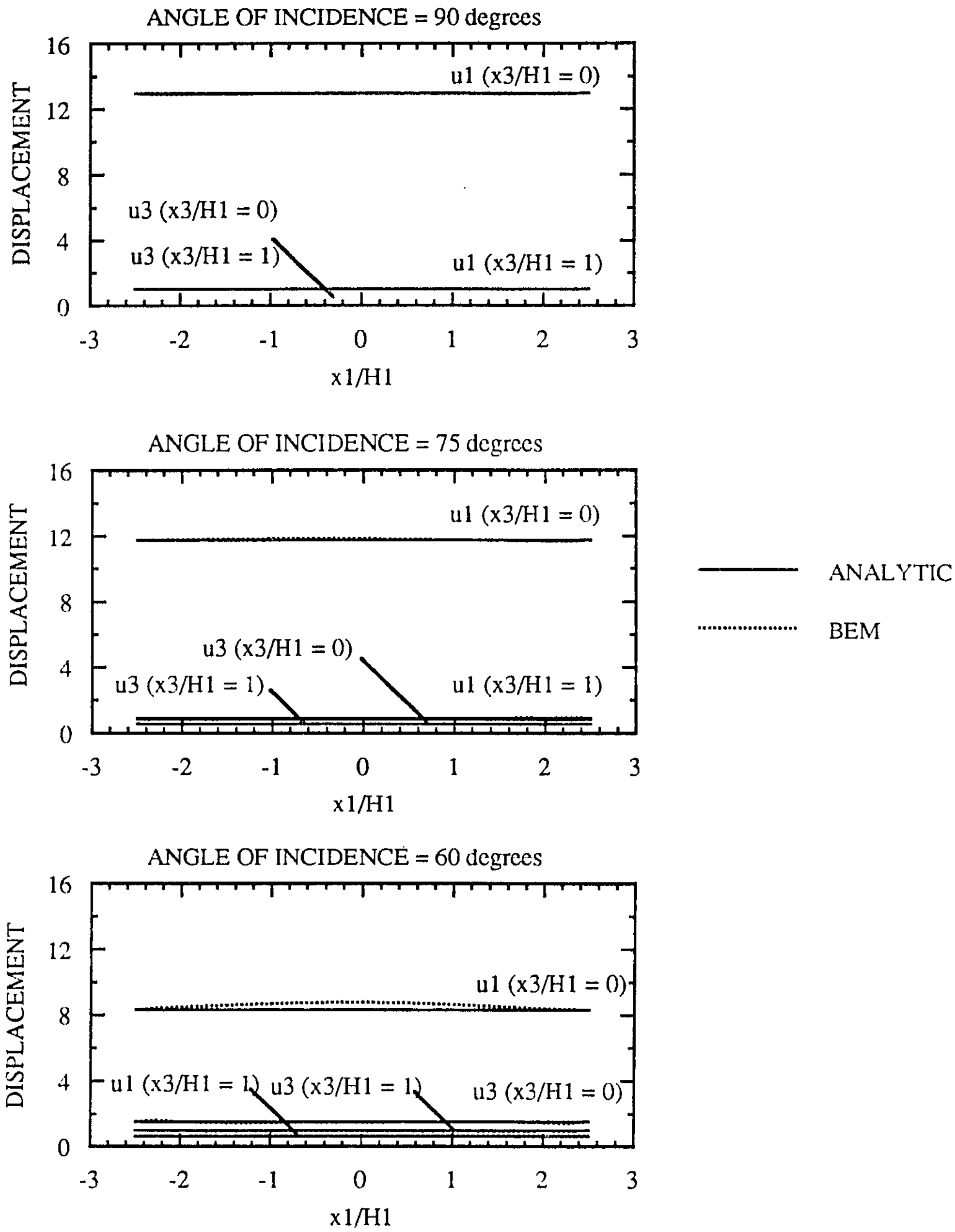


Figure 10-7: Single Layer on a Half-Space,
Displacement Due to an Incident SV Wave with
ks1*H1 = 1.5708

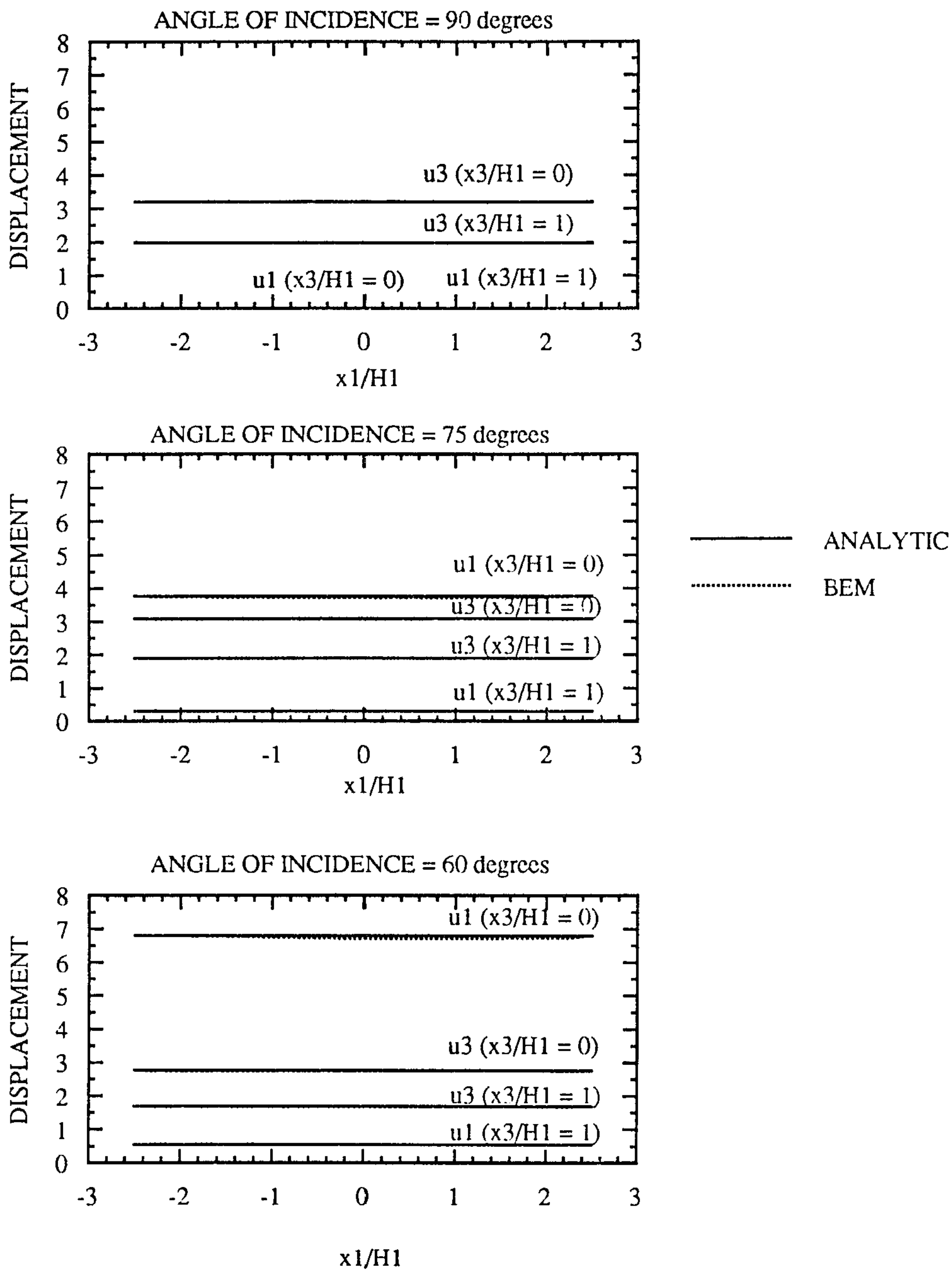


Figure 10-8: Single Layer on a Half-Space,
Displacement Due to an Incident P Wave with
ks1*H1 = 1.5708

The last check problem uses the soil-bedrock geometry shown in Figure 1-2, but using the same properties for rock "A" and rock "B". The regions are discretized in the same manner as what is used for the non-homogeneous rock problem and is solved for each of the wave types using ks(layer 1) * H1 = 1.5708 . Figures 10-9, 10-10, and 10-11, show that the boundary element results compare well with the analytic solution for each of the incident wave types.

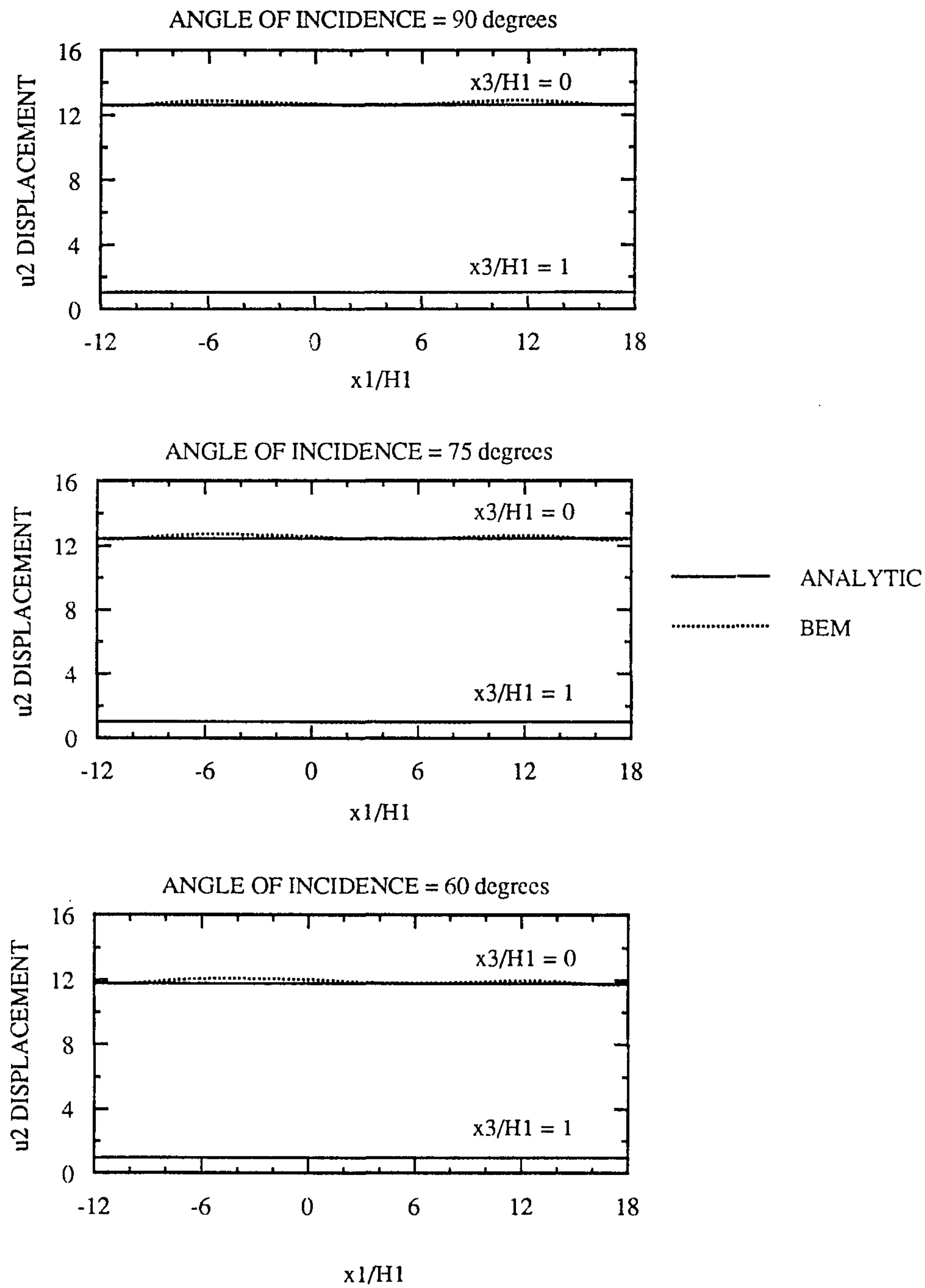


Figure 10-9: u2 Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 10,
H2 / H1 = 1.0,
Ks1*H1 = 1.5708

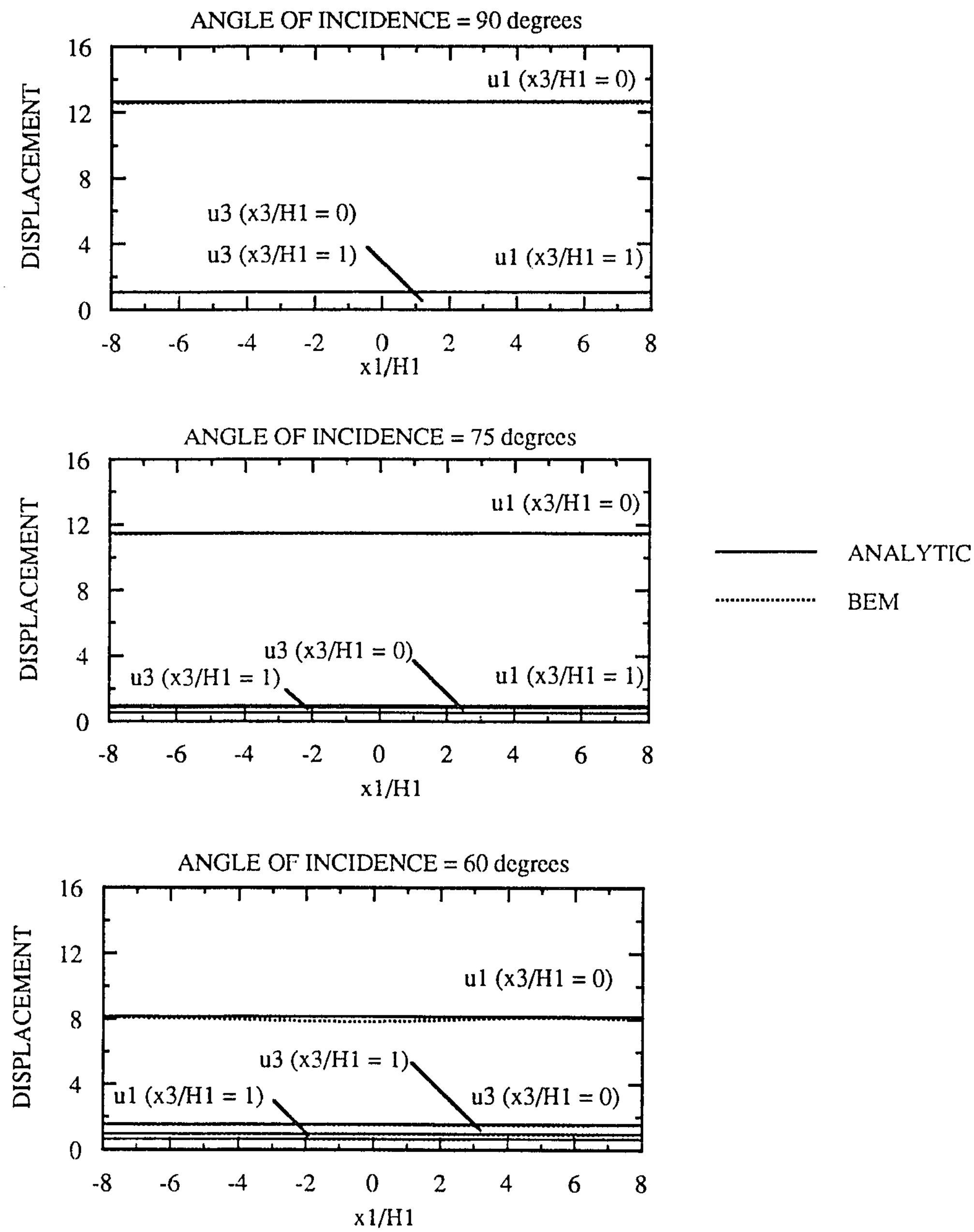


Figure 10-10: Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 10,
H2 / H1 = 1.0,
ks1*H1 = 1.5708

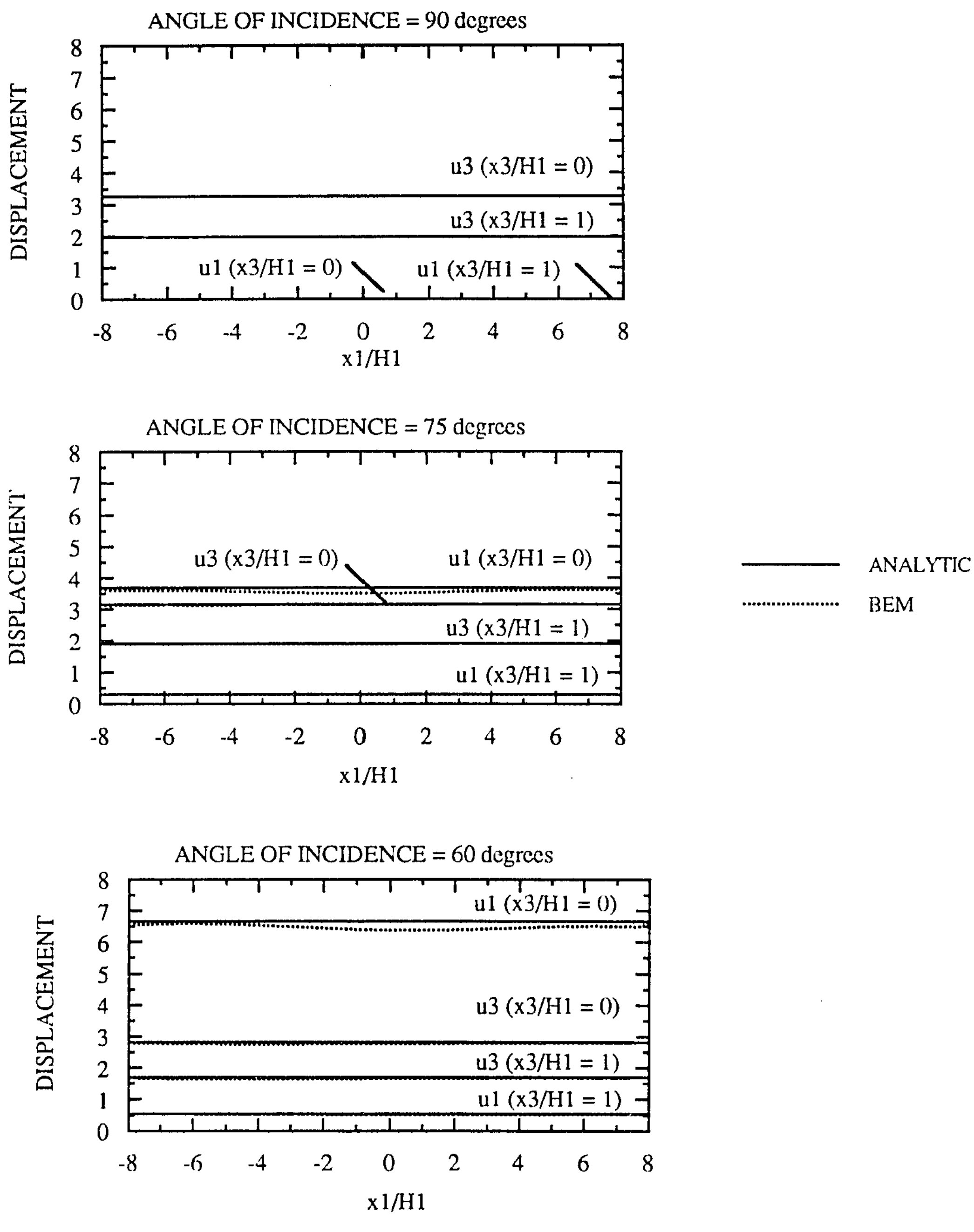


Figure 10-11: Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 10,
H2 / H1 = 1.0,
ks1*H1 = 1.5708

Chapter 11
Two-Dimensional Results

Surface amplification and scattering limits due to body waves using the boundary element method were calculated for dimensionless frequencies of 0.5, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.5, and 3.0 . The scattering limits indicate the distance away from the bedrock discontinuity ($x_1 = 0$) outside of which scattering is not significant in the surface amplification and simple one-dimensional calculations suffice. The scattering limits were developed by determining the point along the surface where the surface amplification varies by no more than a maximum of 10.0% from the one-dimensional solution. For surface amplifications below 1.0, the point was developed by using a maximum variance = ±0.10. Figure 11-1 illustrates for a particular case how the scattering limits were determined.

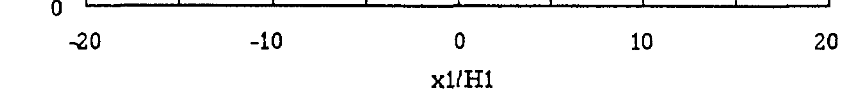


Figure 11-1: Determination of Scattering Limits

Within the scattering limits a two-dimensional solution is required in order to include the effects of scattering due to the discontinuity in layer 2 . Outside of these limits the discontinuity has only a nominal effect on surface amplification .

11.1: Anti-Plane Motion

Surface amplifications due to an incident SH wave are given in Figures 11-2 through 11-10. For each soil profile results are given for incidence angles of 90, 75, and 60 degrees. Three different shear velocity ratios were used for the embedded layer: 5, 2.5, and 1.0. The first two cases represent cases of an embedded rock layer. The third case in which the embedded layer shear velocity ratio is equal to 1.0 is equivalent to having a soil layer on a rock half-space in which there is a step in the rock half-space at $x_1 = 0.0$. For each soil profile, three embedded rock thicknesses were used: 1.0, 2.0, and 5.0 . Results for the shear velocity ratio equal to 5.0 are given in Figures 11-2 through 11-4. Results for the shear velocity ratio equal to 2.5 are given in Figures 11-5 through 11-7. And results for the shear velocity ratio equal to 1.0 are given in Figures 11-8 through 11-10. At the end points of these curves, displacements are forced to equal the 1-D solution displacements. The shape of the curves for these cases is independent of the angle of incidence. For any given shear velocity ratio, as the thickness of the embedded layer increases, scattering increases . Keeping the layer thickness constant and reducing the stiffness of the embedded material increases scattering. For any one soil profile, the waviness of the surface displacement increases as the frequency increases. This waviness indicates increased scattering and that two dimensional effects become more significant as the frequency increases.

Scattering limit plots summarize the scattering effects of the embedded layer and are displayed in Figures 11-11 through 11-13 for the various cases. Since one-dimension displacements are enforced at $x_1/H1 = -12$ and $x_1/H1 = +18$, the scattering limits are forced to be within this range. Scattering limits that approach these end points are artificial and indicate the need for a larger area to be discretized. The case of a embedded rock layer with the same thickness as the soil layer and a shear velocity half of the rock half-space has minimal effect on scattering. Scattering limits increase in the area above the embedded layer, $x_1 > 0$, as the thickness of the embedded layer increases; this is generally true for the surface area above rock "B", $x_1 < 0$. For most cases, a shallower incidence angle results in a larger scattering area. The scattering limits for the embedded layer having the same properties as the soil layer are included in Figure 11-13. The scattering limits in the $-x_1$ region fall within the discretized region for thickness ratios of 1 and 2, however are outside the discretized $+x_1$ region since the scattering limits approach the end points of the discretized surface boundary. For a thickness ratio of 5, the scattering limits approach the end points of the discretized surface boundary.

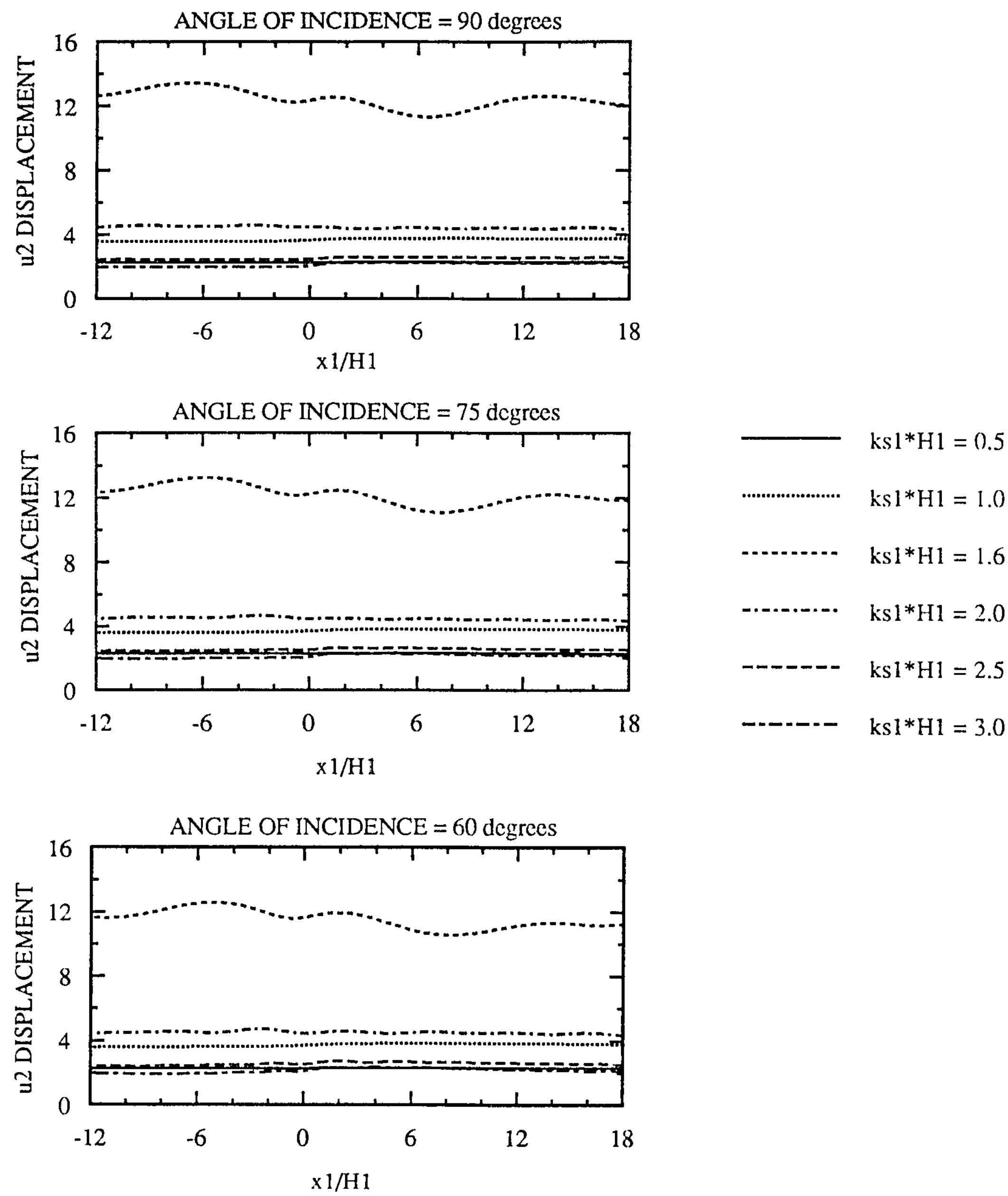


Figure 11-2: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 5
H2 / H1 = 1.0

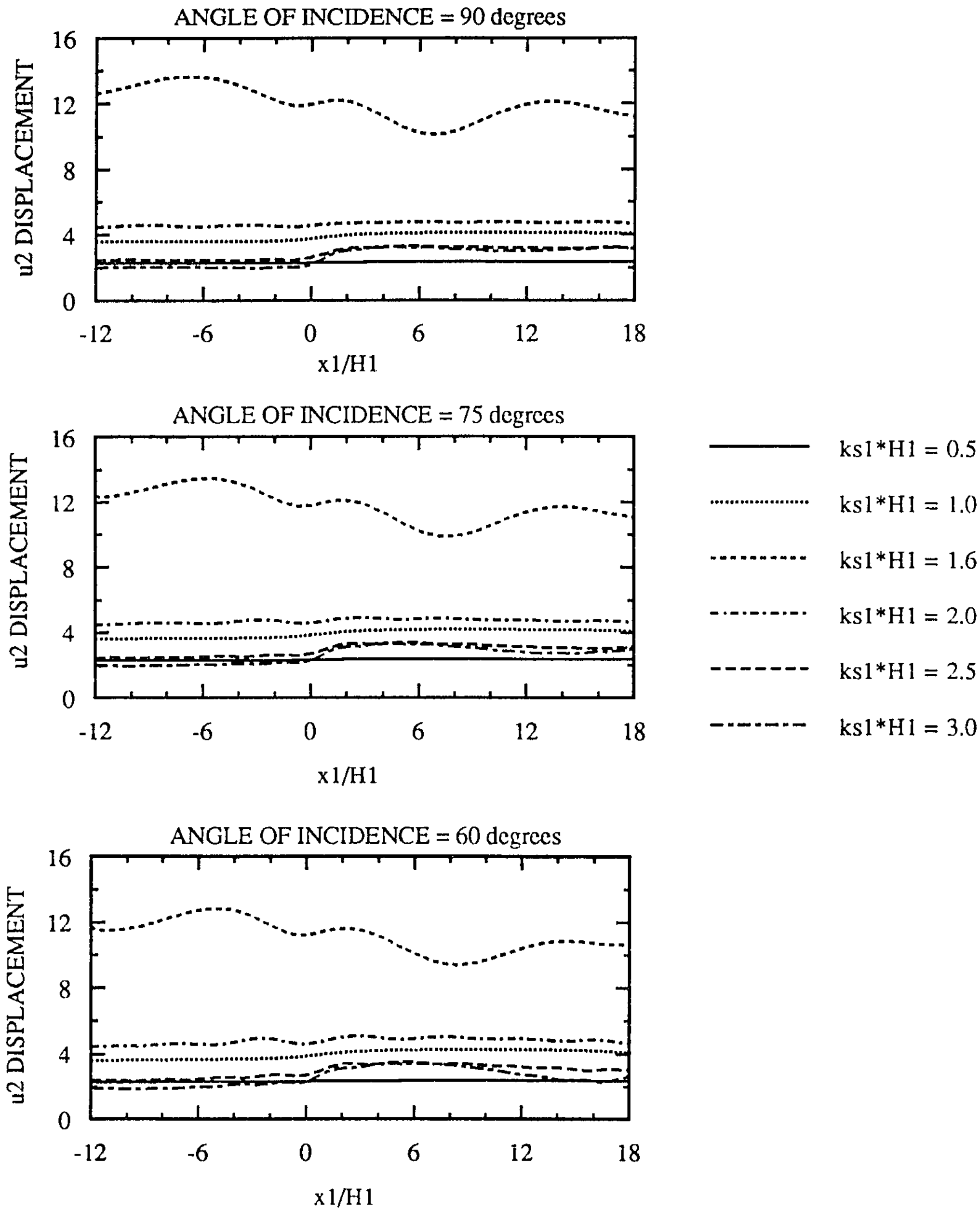


Figure 11-3: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 2.0

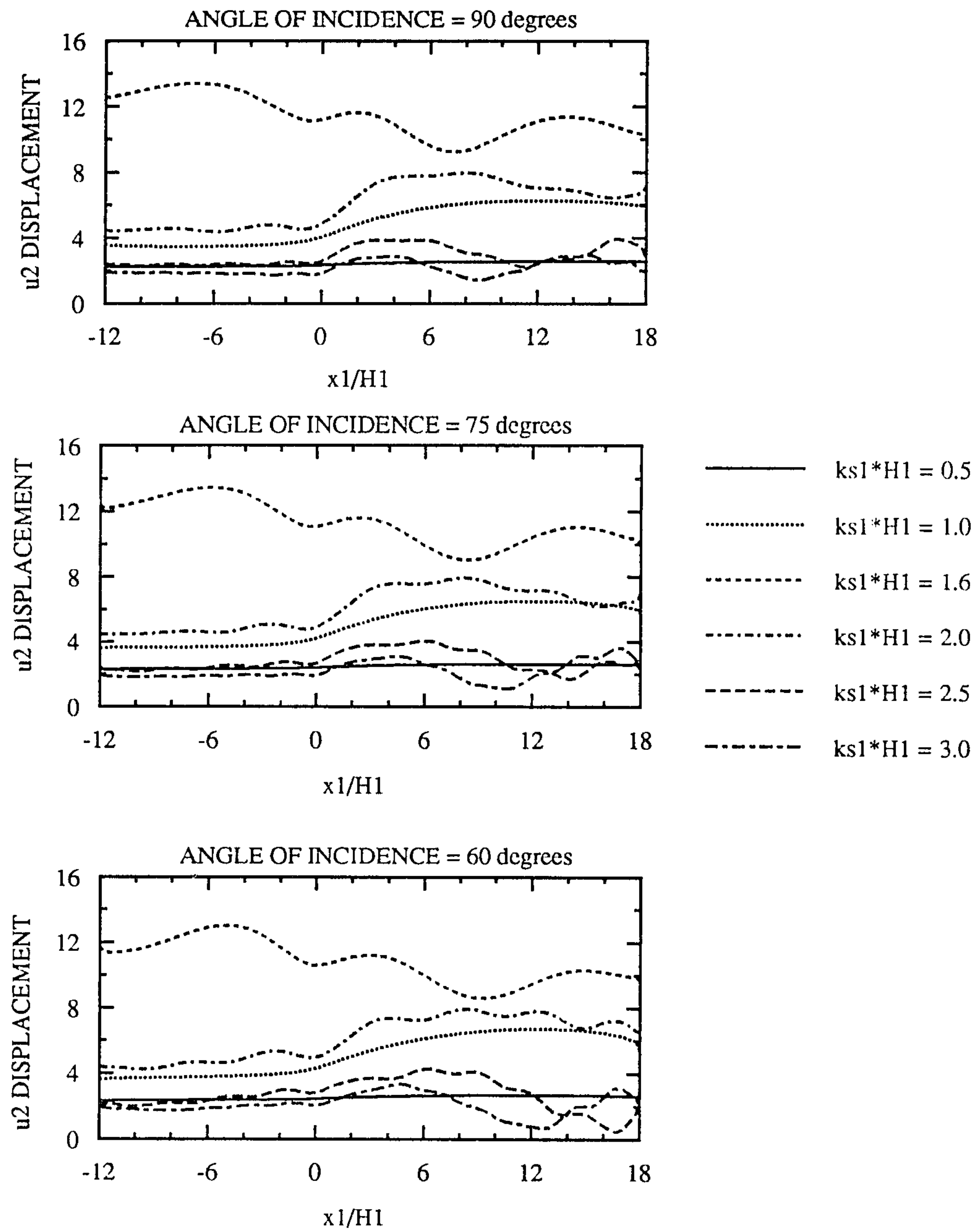


Figure 11-4: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 5.0

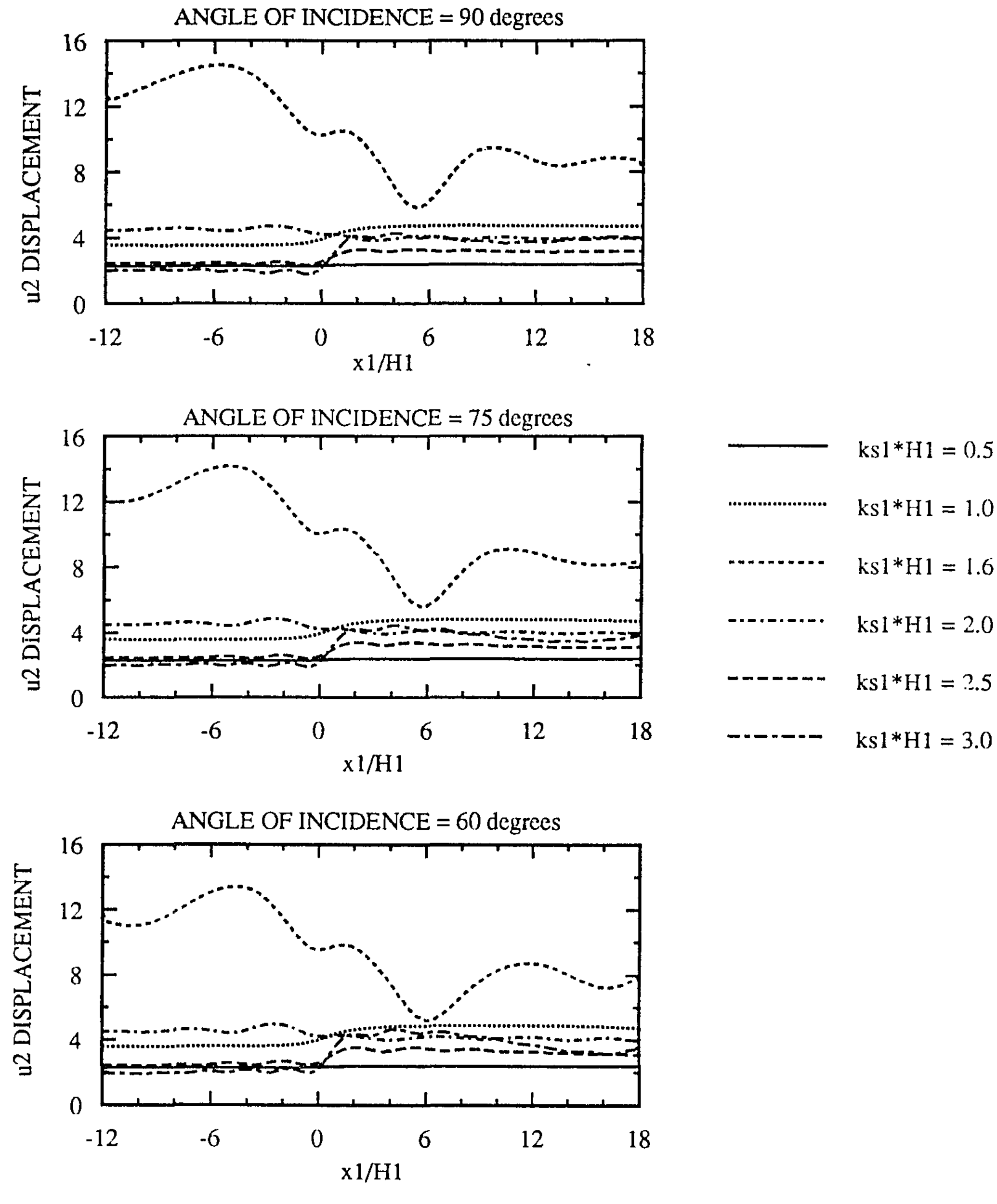


Figure 11-5: u2 Surface Displacment Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1.0

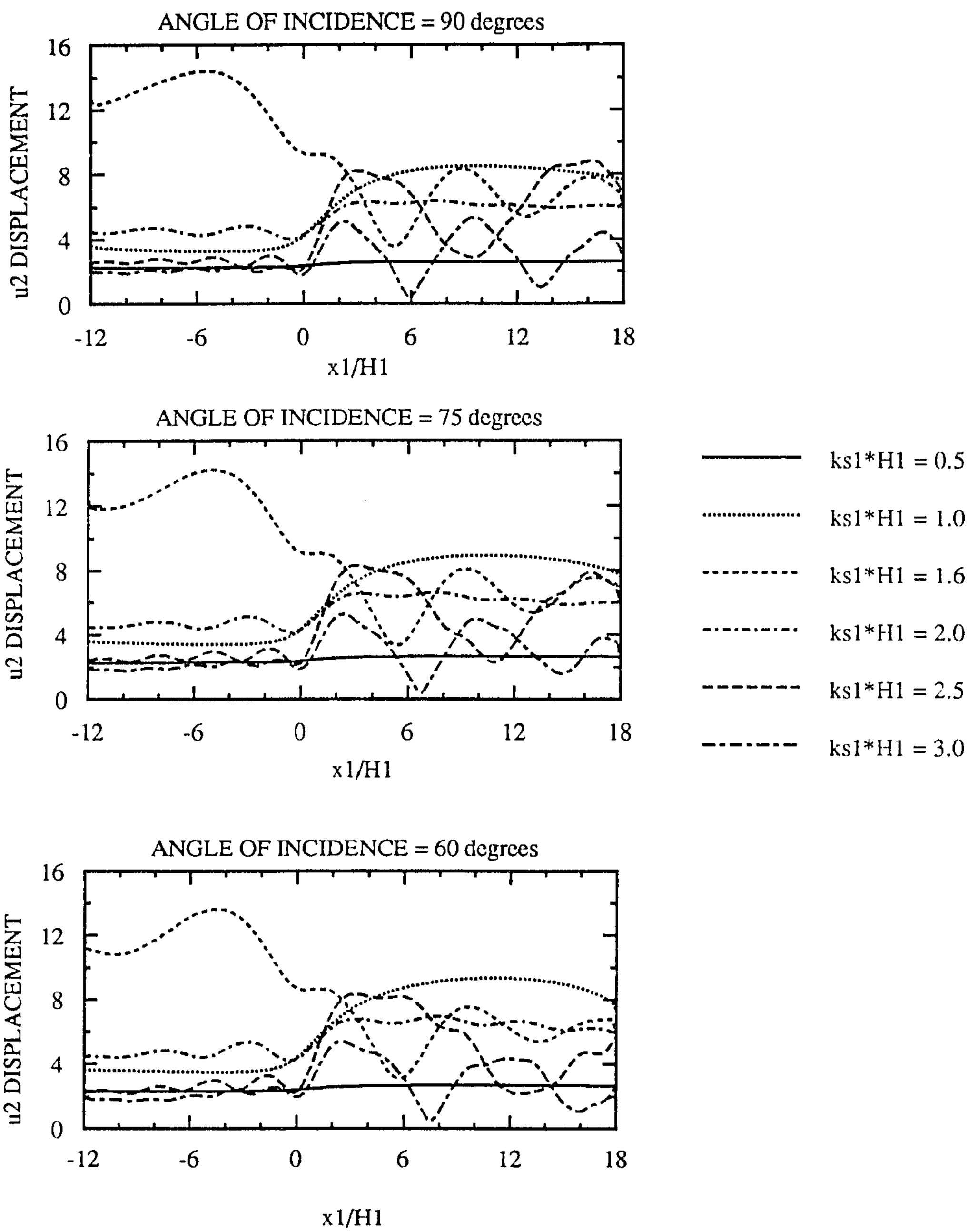


Figure 11-6: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 2.0

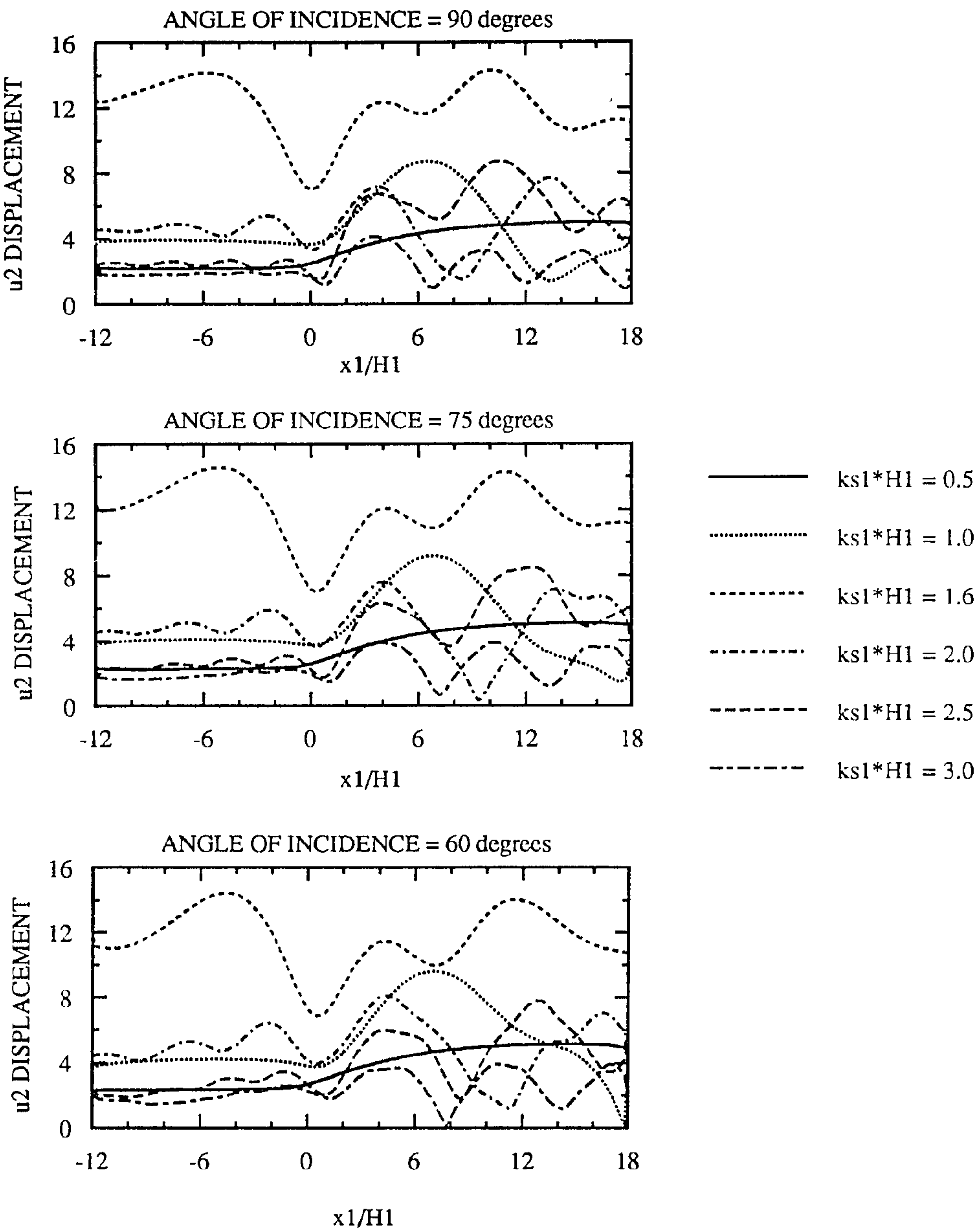


Figure 11-7: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 5.0

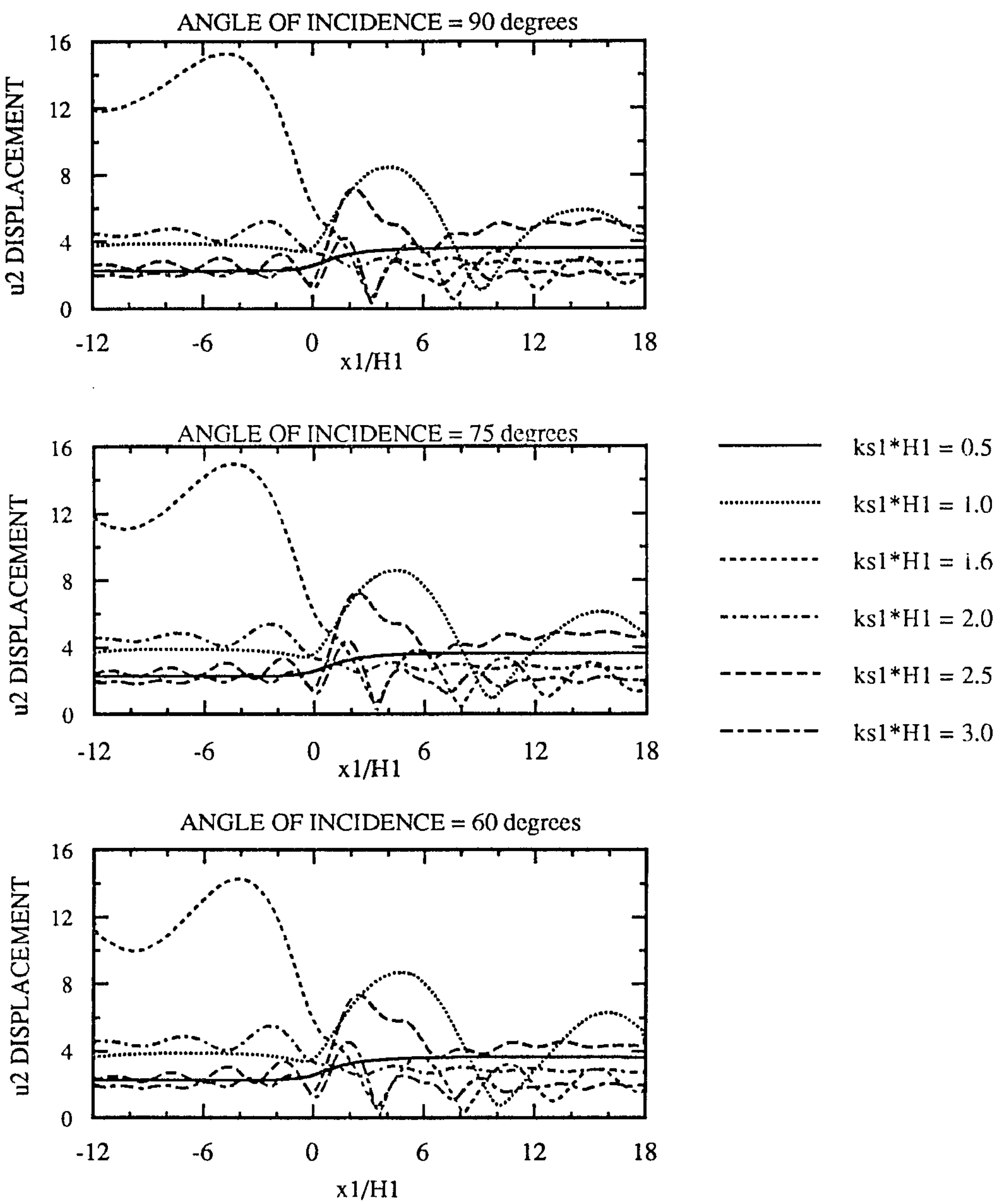


Figure 11-8: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 1,
H2 / H1 = 1.0

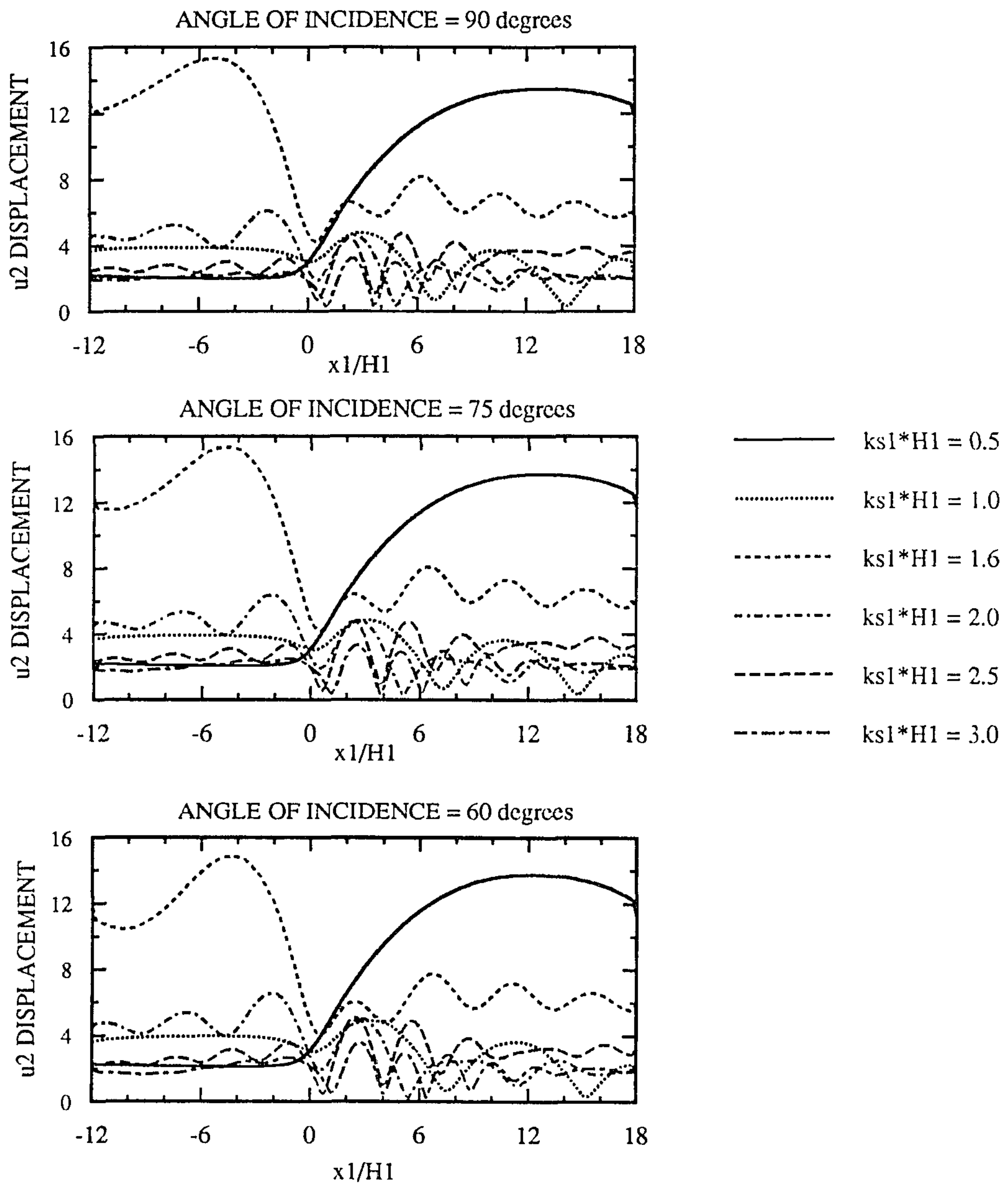


Figure 11-9: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 1,
H2 / H1 = 2.0

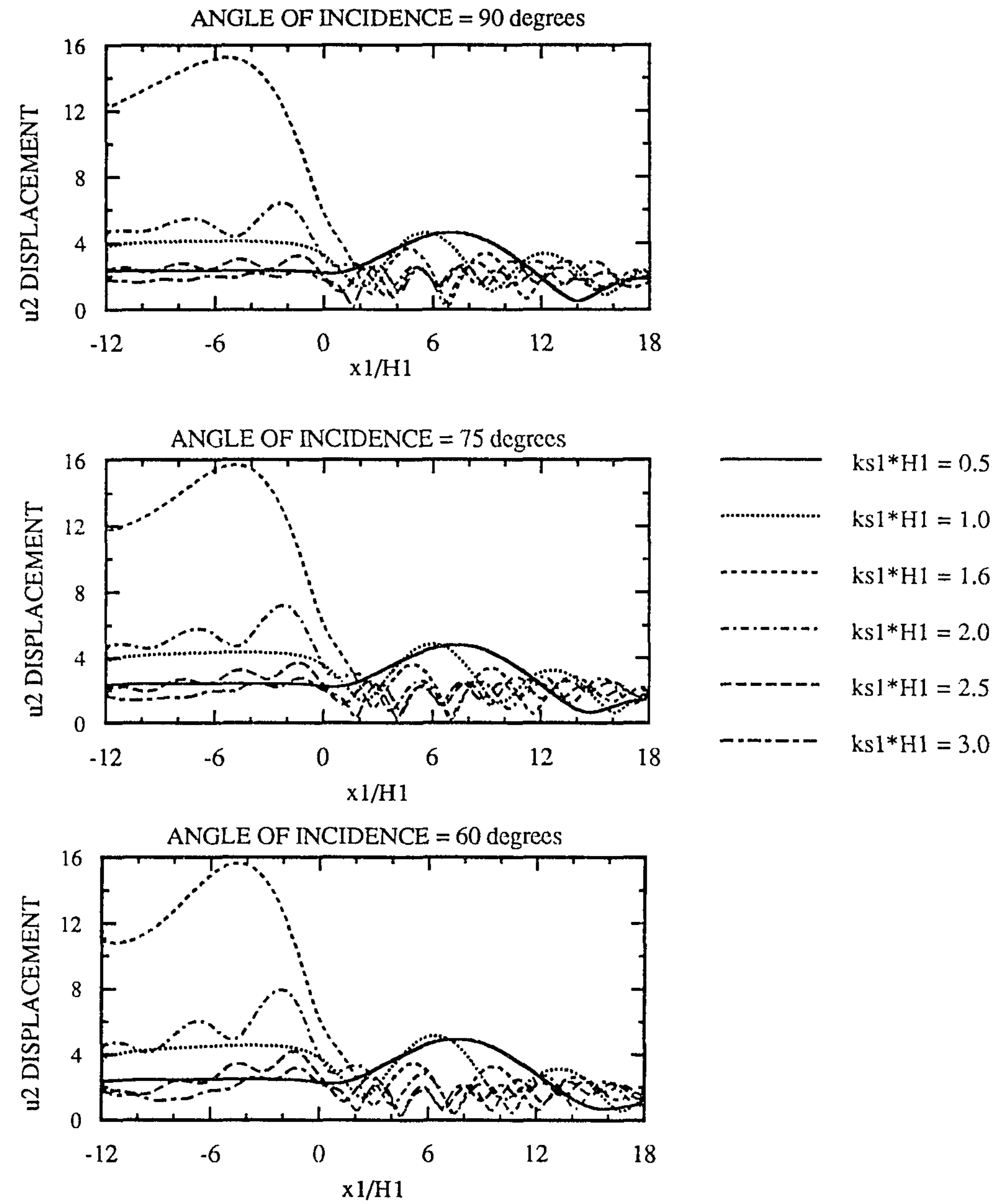


Figure 11-10: u2 Surface Displacement Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 1,
H2 / H1 = 5.0

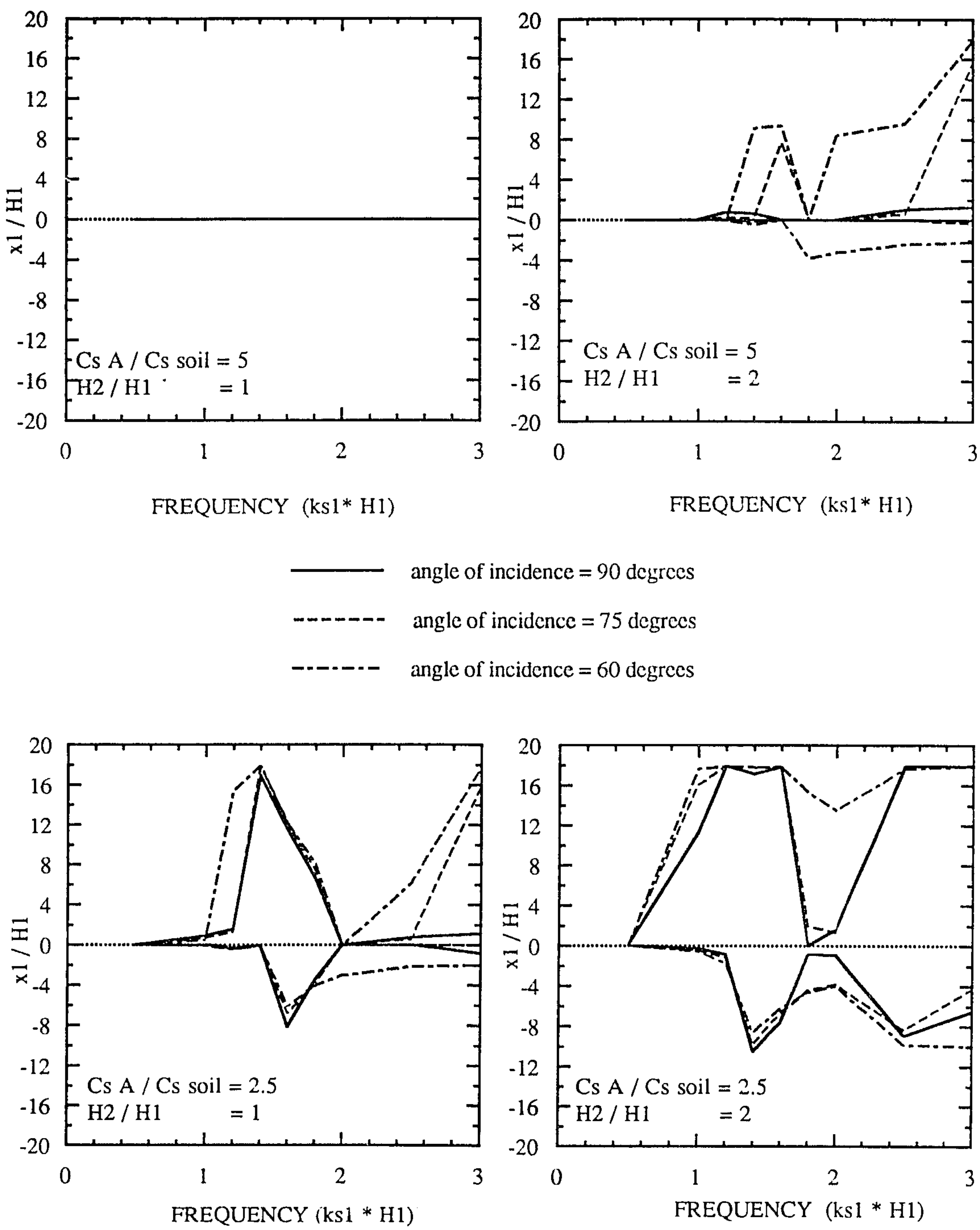


Figure 11-11: u2 Scattering Limits Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 5 and Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1 and H2 / H1 = 2

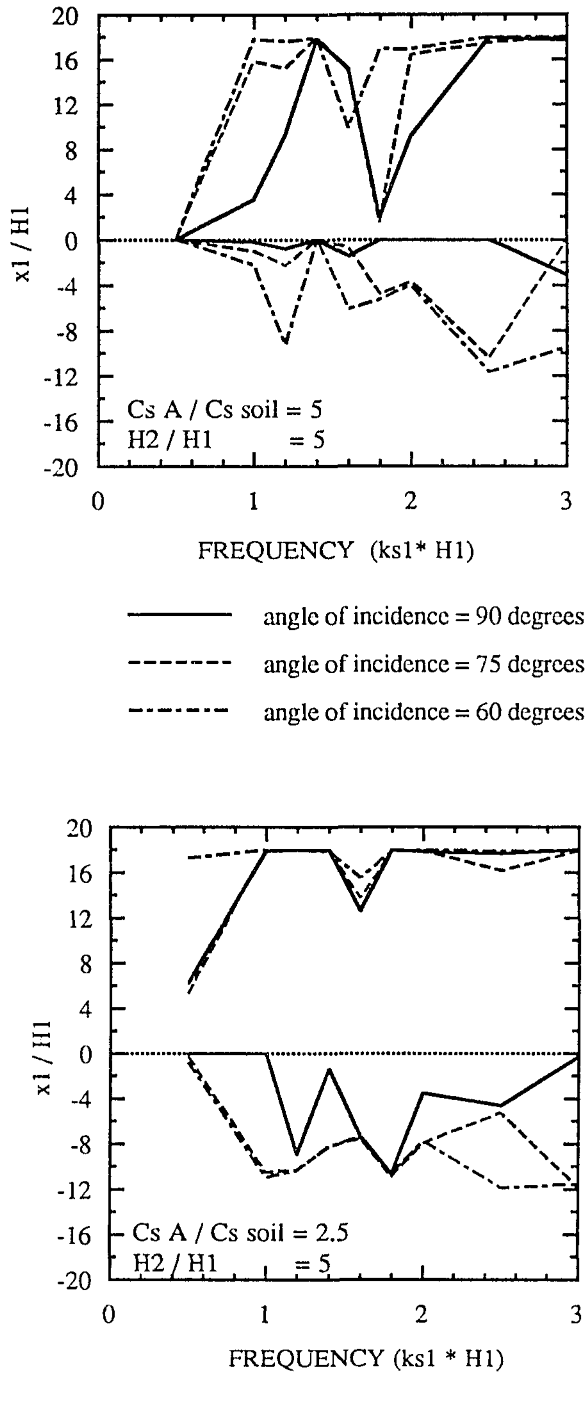


Figure 11-12: u2 Scattering Limits Due to a Unit Incident SH Wave, Cs rock "A" / Cs soil = 5 and Cs rock " A" / Cs soil = 2.5, H2 / H1 = 5

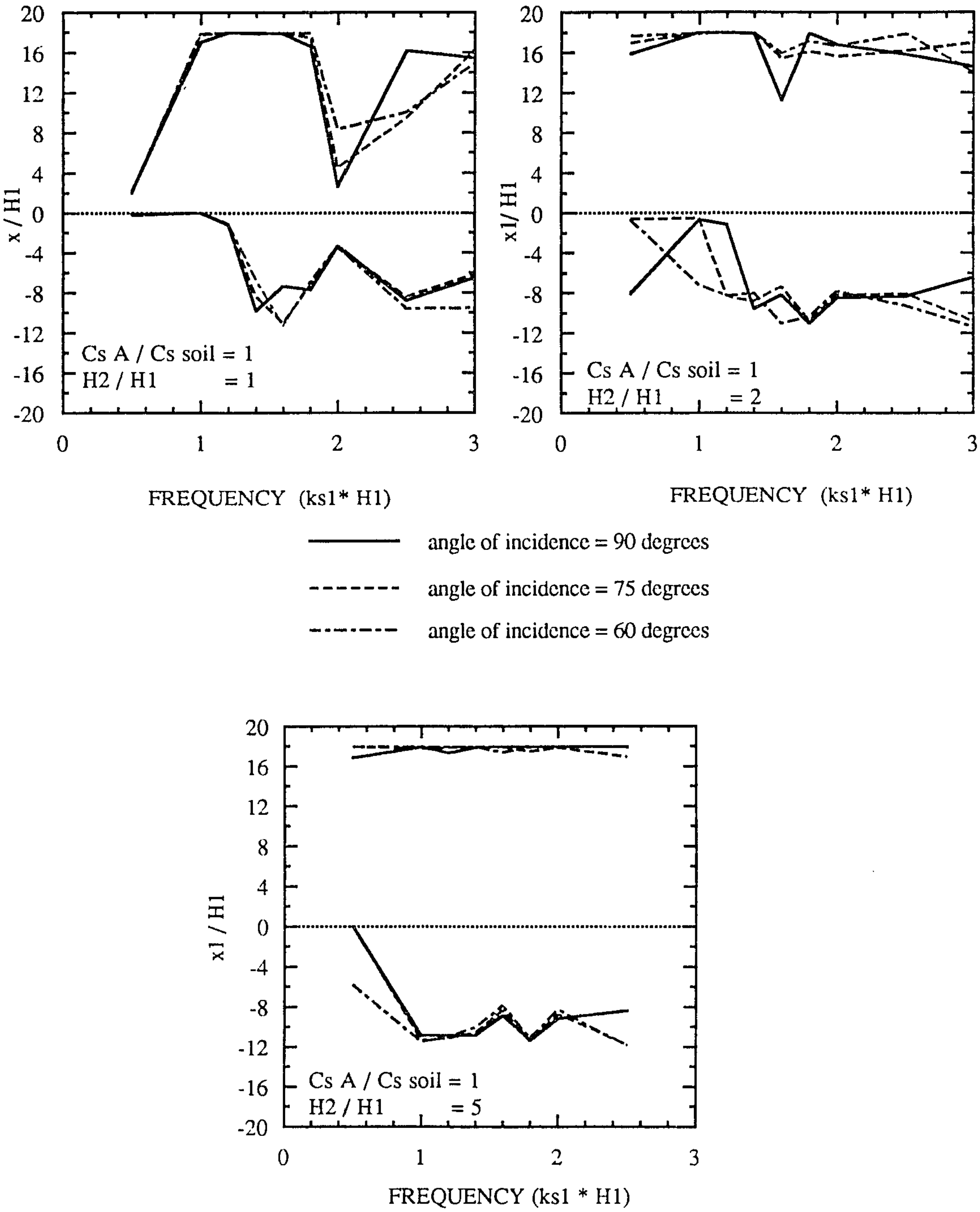


Figure 11-13: u2 Scattering Limits Due to a Unit Incident SH Wave,
Cs rock "A" / Cs soil = 1,
H2 / H1 = 1, H2 / H1 = 2, and H2 / H1 = 5

11.2: In-Plane Motion

Surface amplifications for in-plane body waves are given for various soil profiles and incidence angles. Two shear velocity ratios and two thickness ratios of the embedded rock layer were used for the analysis. For each of these soil profiles, a unit incident wave was applied at 90, 75, and 60 degrees. Results for incident SV and P waves are given as surface amplification plots in the horizontal and vertical directions. Scattering limit plots are also included for the horizontal and vertical displacements. These scattering limit plots indicate the dependency of the range of scattering to the frequency and the large area in which the scattering effect of the embedded rock layer encompasses. Because of the mode conversion included in plane wave calculations, the scattering profiles are more complicated than the ones for anti-plane motion.

11.2.1: Incident SV Wave

Results in the horizontal and vertical directions due to a unit incident SV wave are given in Figures 11-14 through 11-17 for a shear velocity ratio of 5.0 and in Figures 11-18 through 11-21 for a shear velocity ratio of 2.5 . For dimensionless frequencies below 2, the amplification varies smoothly from the $-x_1$ region to the $+x_1$ region . The length of this smooth transition can be determined from the scattering limit plots. For frequencies of 2 and above, surface amplification fluctuates resulting in peaks and valleys. Decreasing the angle of incidence, increasing the embedded rock thickness, or decreasing the embedded rock shear velocity results in larger deviation of the surface amplification from the one-dimensional solution. Increased scattering was found to occur at the frequency of 2.5 which is not the highest frequency in the study, but is the closest to the natural frequency of vertically propagating P waves.

The plots of scattering limits for horizontal and vertical displacement indicate the large scattering that occurs because of the embedded rock layer. These plots are shown in

Figures 11-22 and 11-23. For most cases, the scattering length approaches the length in which the surface is discretized and is therefore inapplicable. For the cases in which the discretized region is greater than the scattering length, the patterns are much more complicated than that found for the anti-plane solution. Scattering limits in the horizontal direction generally increase as the angle of incidence becomes shallower, however the scattering limits in the vertical direction do not follow this general pattern. The large scattering limits for vertical displacement due to vertically incident motion are misleading since for this case the vertical displacements of the one-dimensional solution are nonexistent and only a small variance is allowed between the two solutions.

For cases in which the scattering limits approach the discretization boundary end points, the solutions at the end points are incorrectly assumed to be the one-dimension solutions. In order to examine the significance of this incorrect assumption in the calculation of the surface displacement in the vicinity over the discontinuity ($x_1 = 0$), an additional case was run in which the discretized area is from $x_1/H1 = -6$ to +6 instead of -8 to +8 . The surface displacement for this additional case is calculated for an incident SV wave and for a soil profile in which the embedded rock layer has a shear velocity ratio equal to 2.5 and a thickness ratio equal to 2.0 . This case was chosen because of its large scattering limits. Comparison between the two discretization scenarios.is made in Figures 11-24 and 11-25 Figure 11-24 shows the comparison for horizontal displacement while Figure 11-25 for vertical displacement. The plots show that the difference in the surface displacements in the area between $x_1/H1 = -4.0$ and +4.0 does not differ significantly for incidence angles equal to 90 and 75 degrees. Differences become significant for the higher frequencies when the angle of incidence equals 60 degrees . These comparisons illustrate that even for the cases in which the scattering is not contained in the limits of discretization, reliable surface displacements can still be had over the local area above the discontinuity.

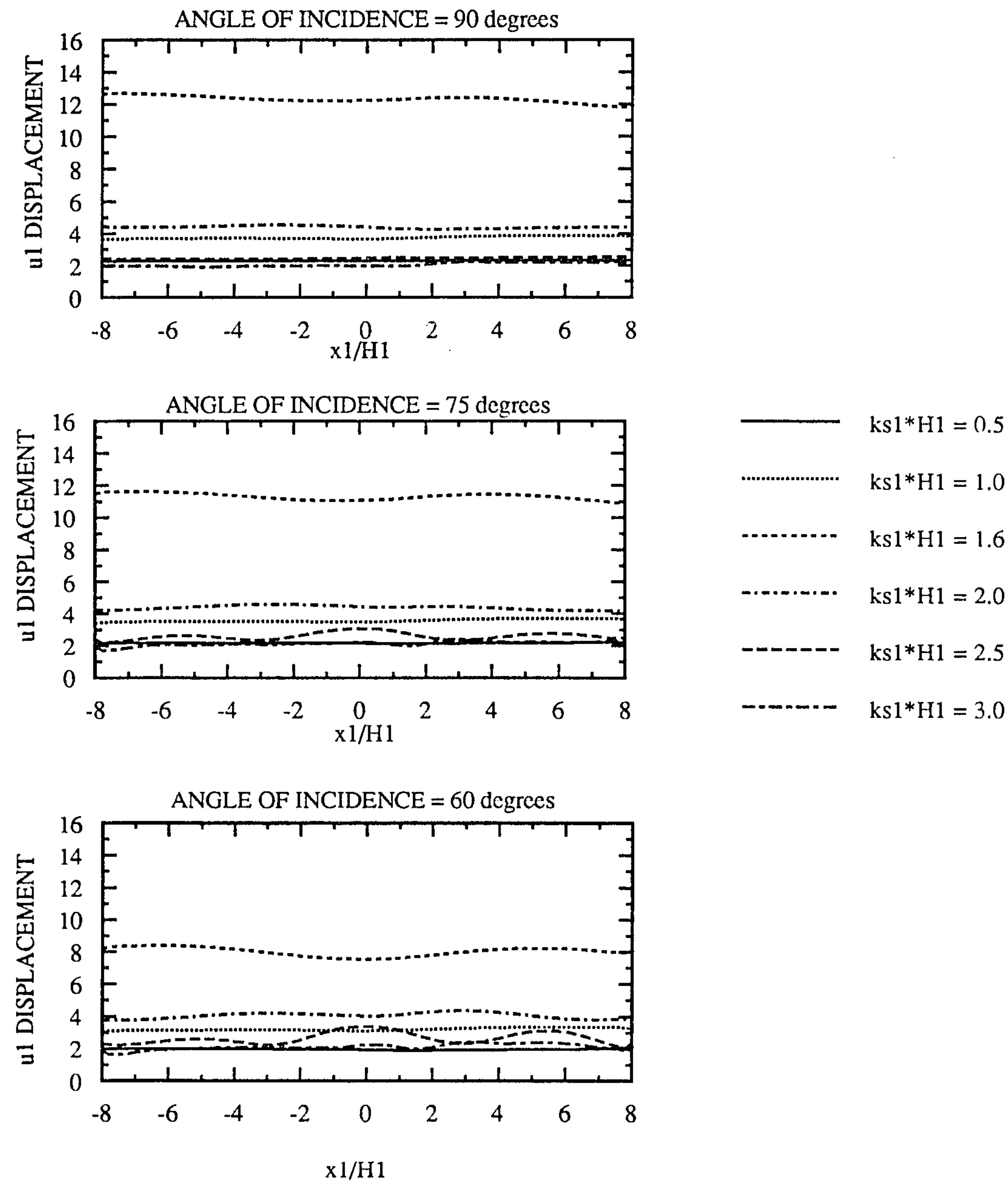


Figure 11-14: u1 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 1.0

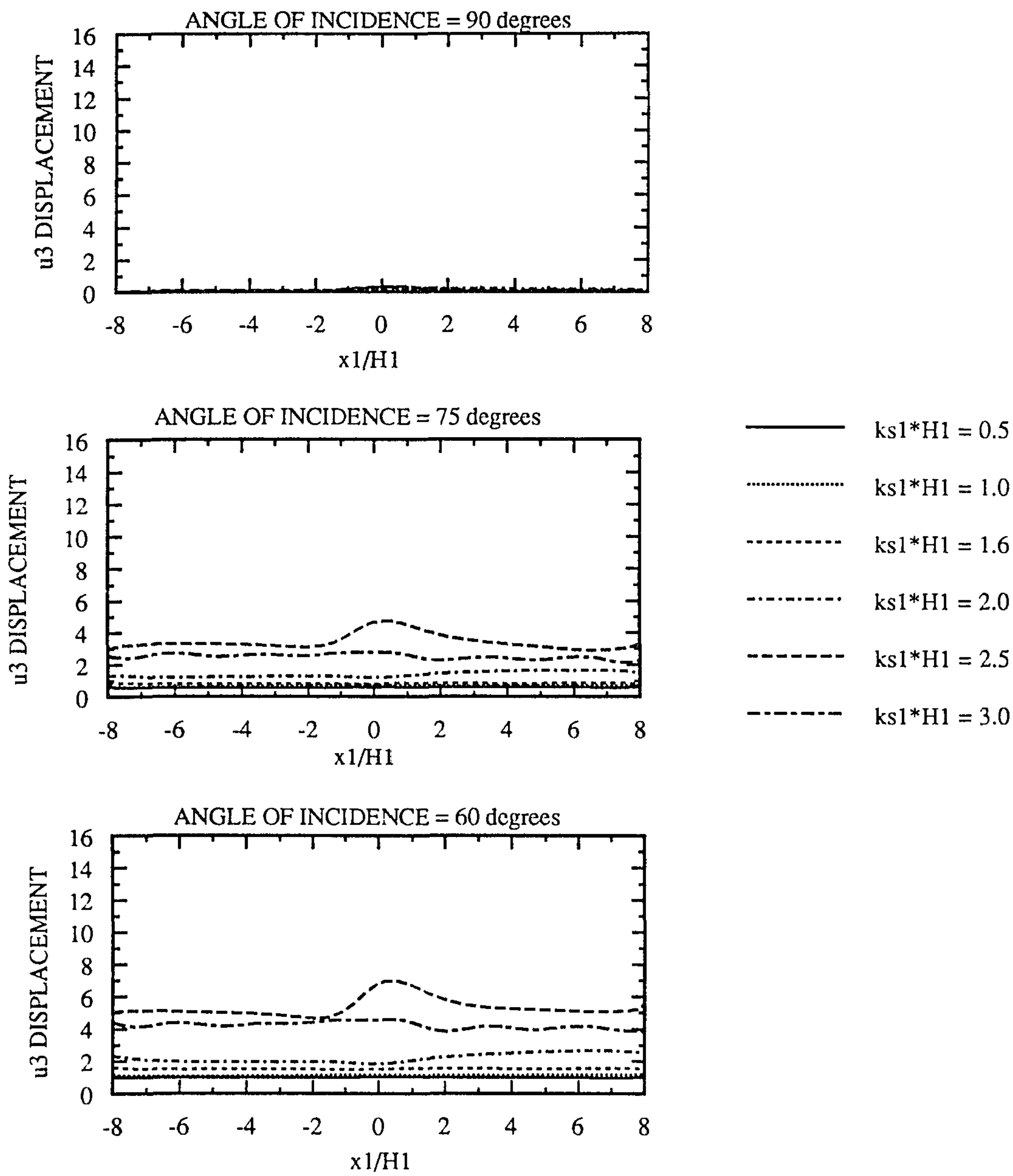


Figure 11-15: u3 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 1.0

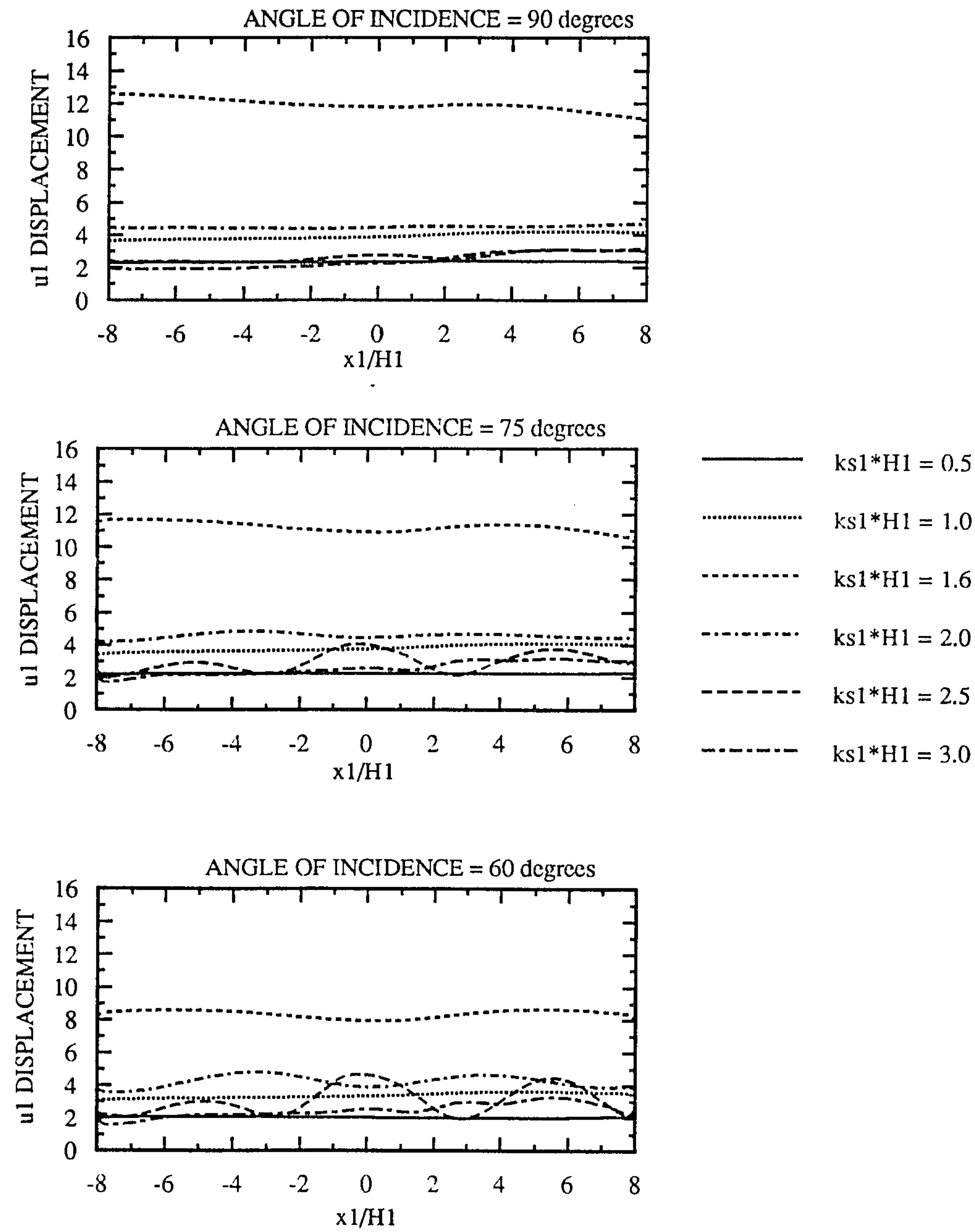


Figure 11-16: u1 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 2.0

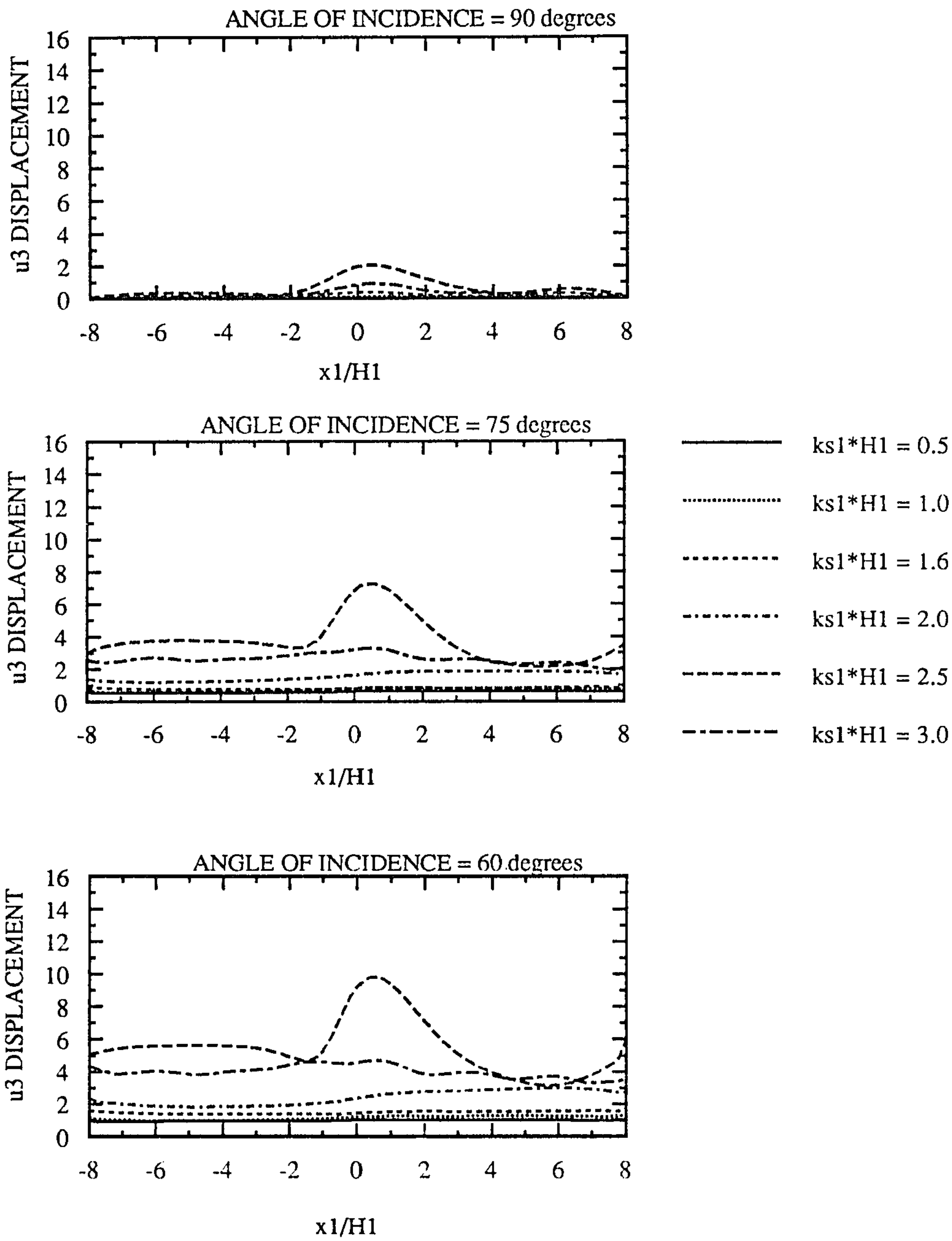


Figure 11-17: u3 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 2.0

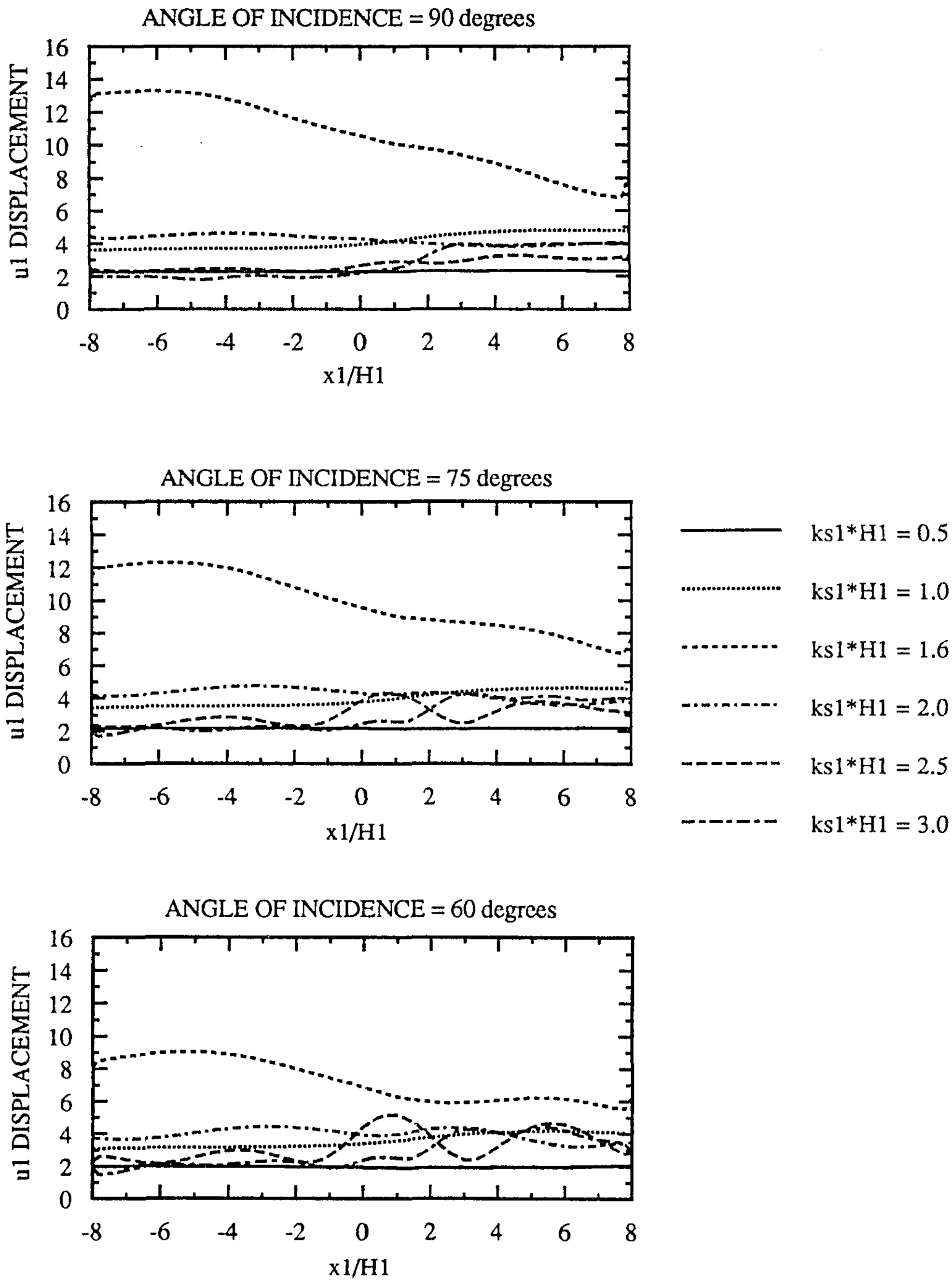


Figure 11-18: u1 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1.0

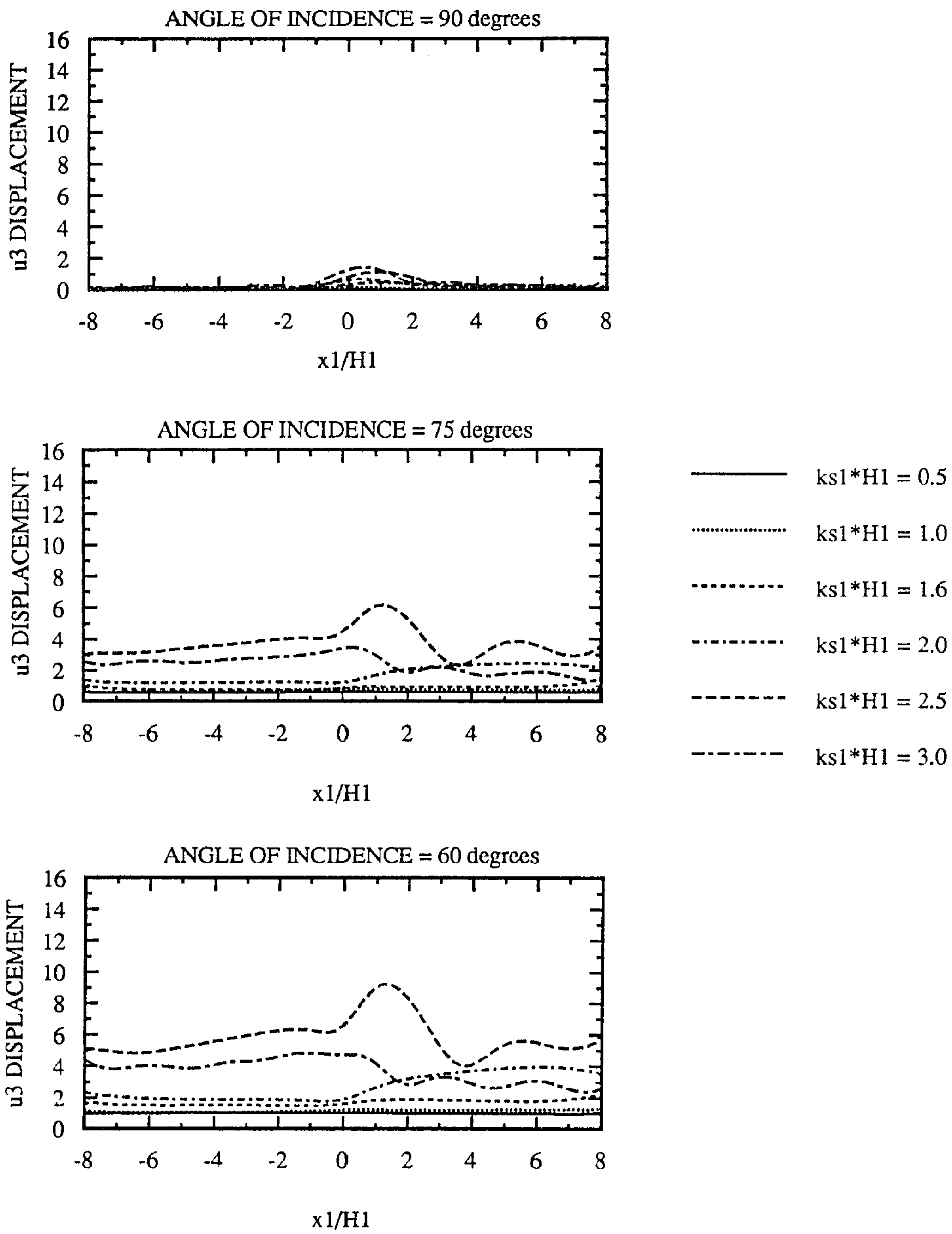


Figure 11-19: u3 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1.0

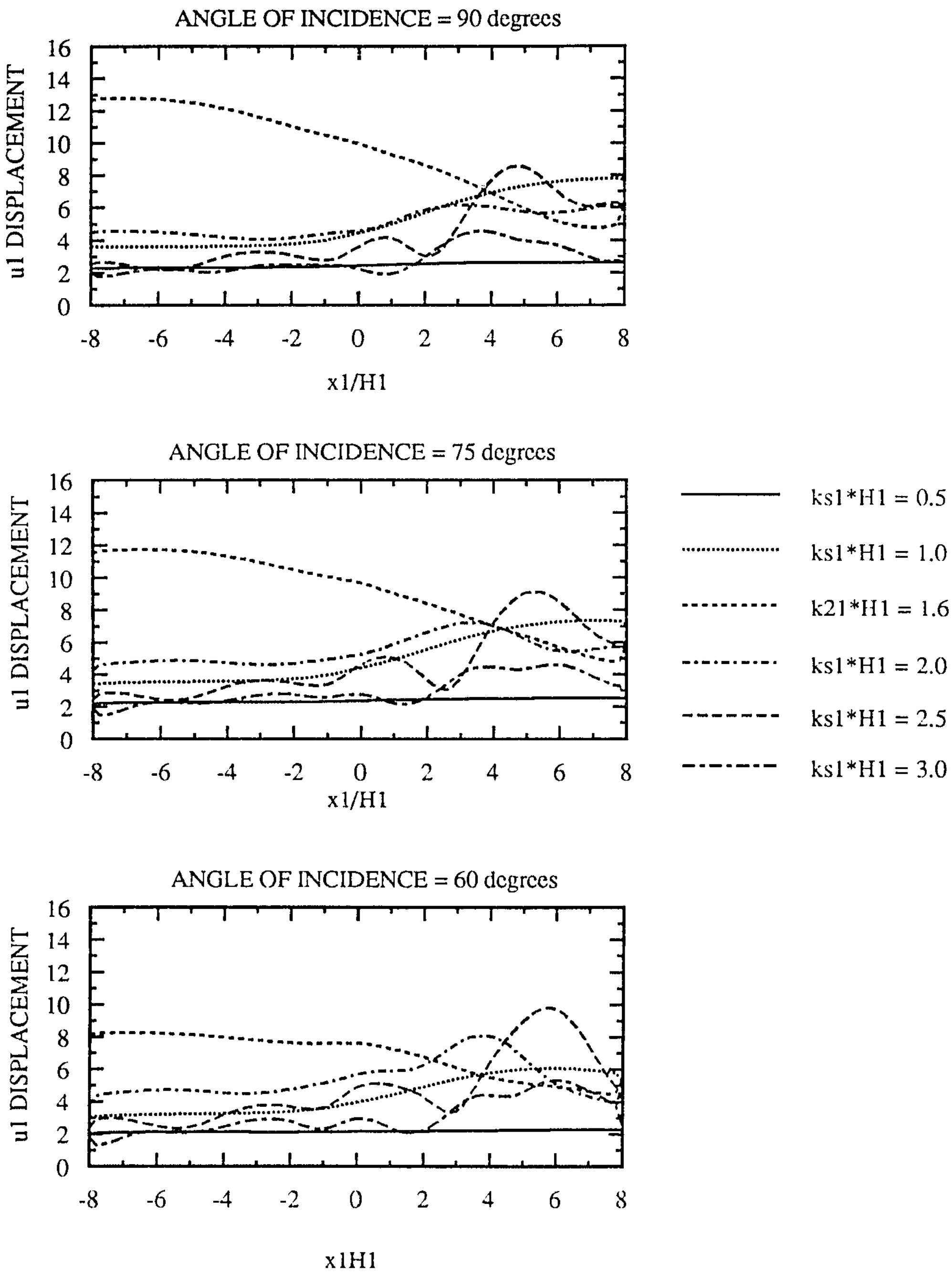


Figure 11-20: u1 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 2.0

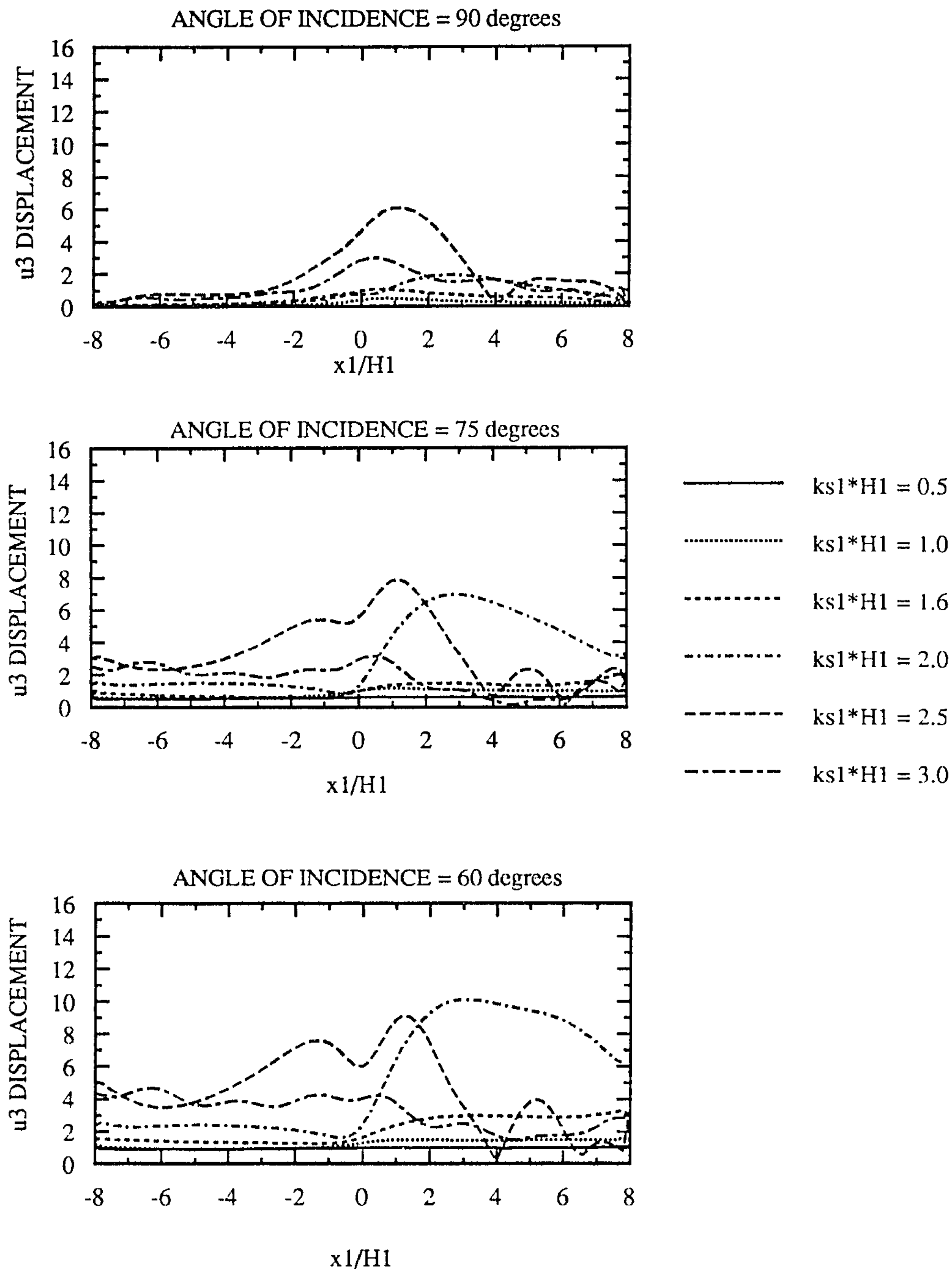


Figure 11-21: u3 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 2.0

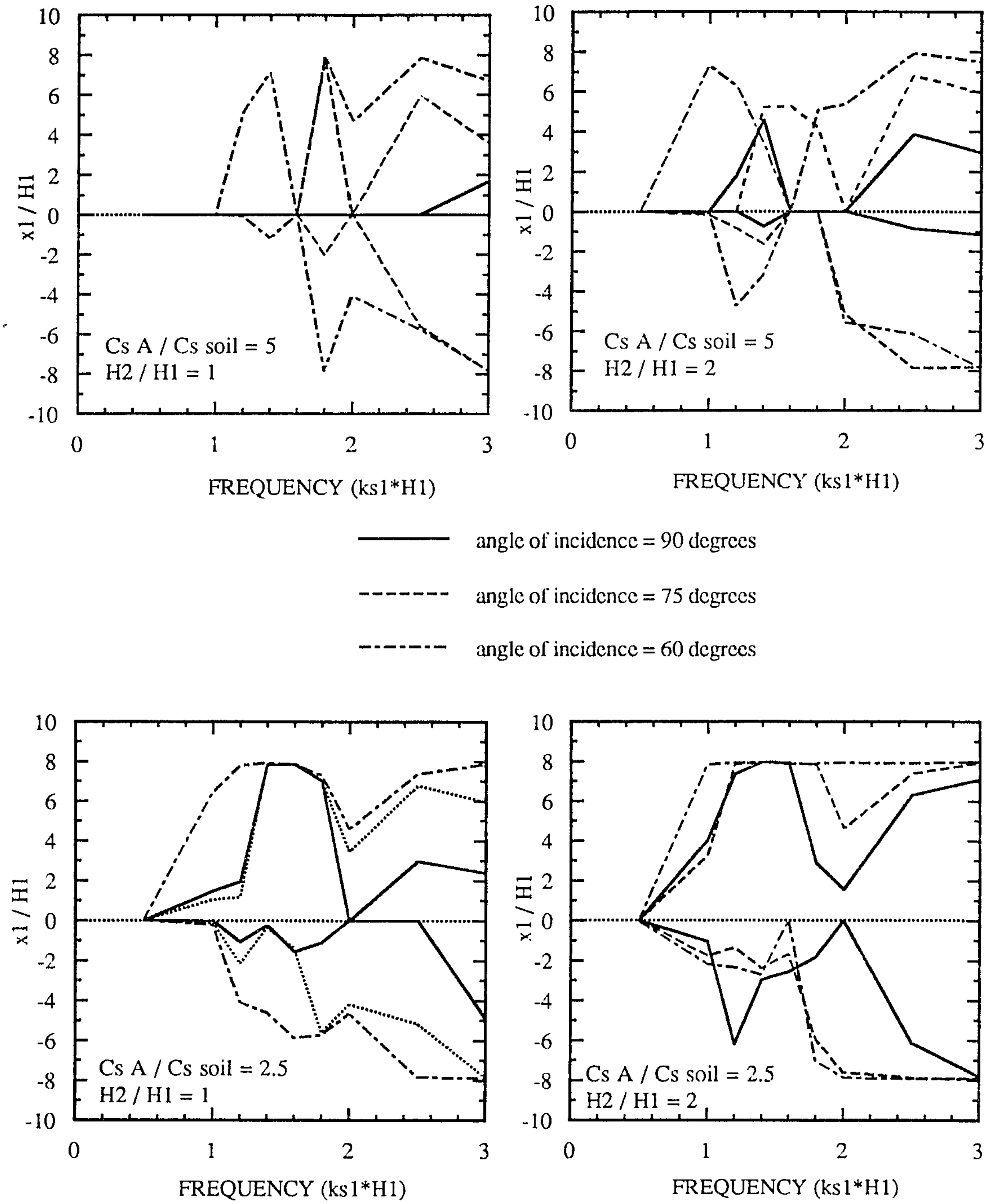


Figure 11-22: u1 Scattering Limits Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5 and Cs rock "A" / Cs soil = 5,
H2 / H1 = 1 and H2 / H1 = 2

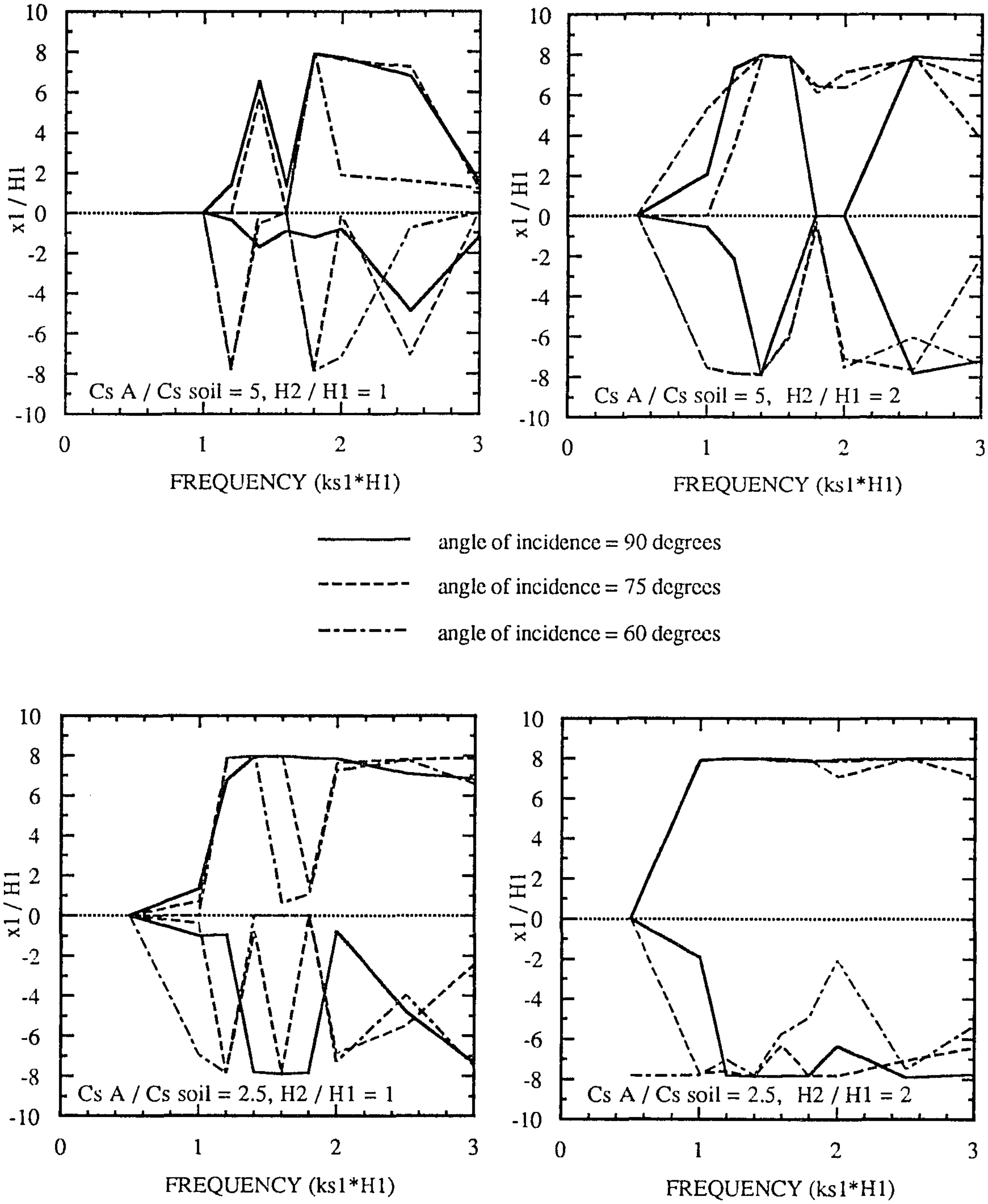


Figure 11-23: u3 Scattering Limits Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5 and Cs rock "A" / Cs soil = 5
H2 / H1 = 1 and H2 / H1 = 2

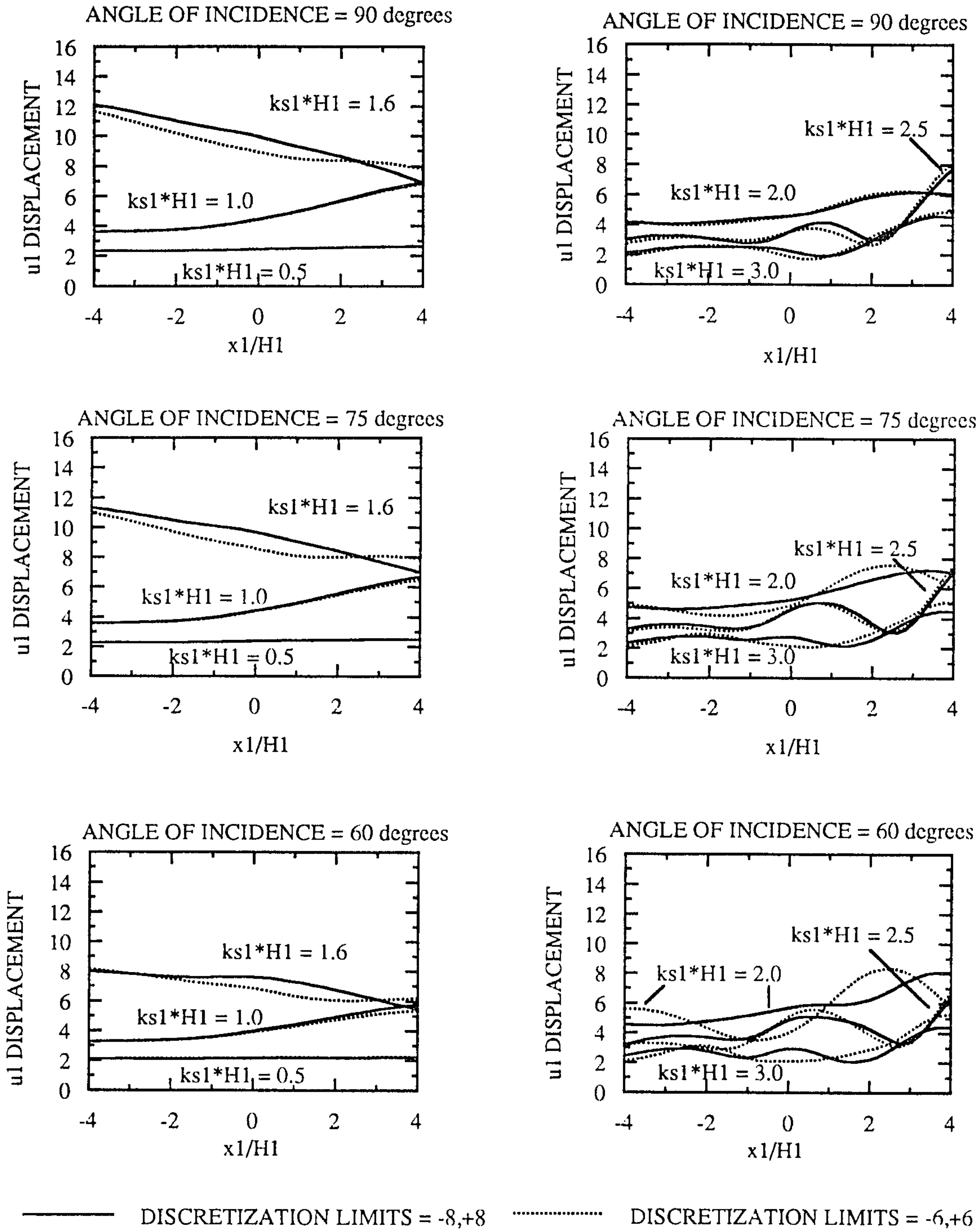


Figure 11-24: Comparison of Discretization Limits,
u1 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5, H2 / H1 = 2.0

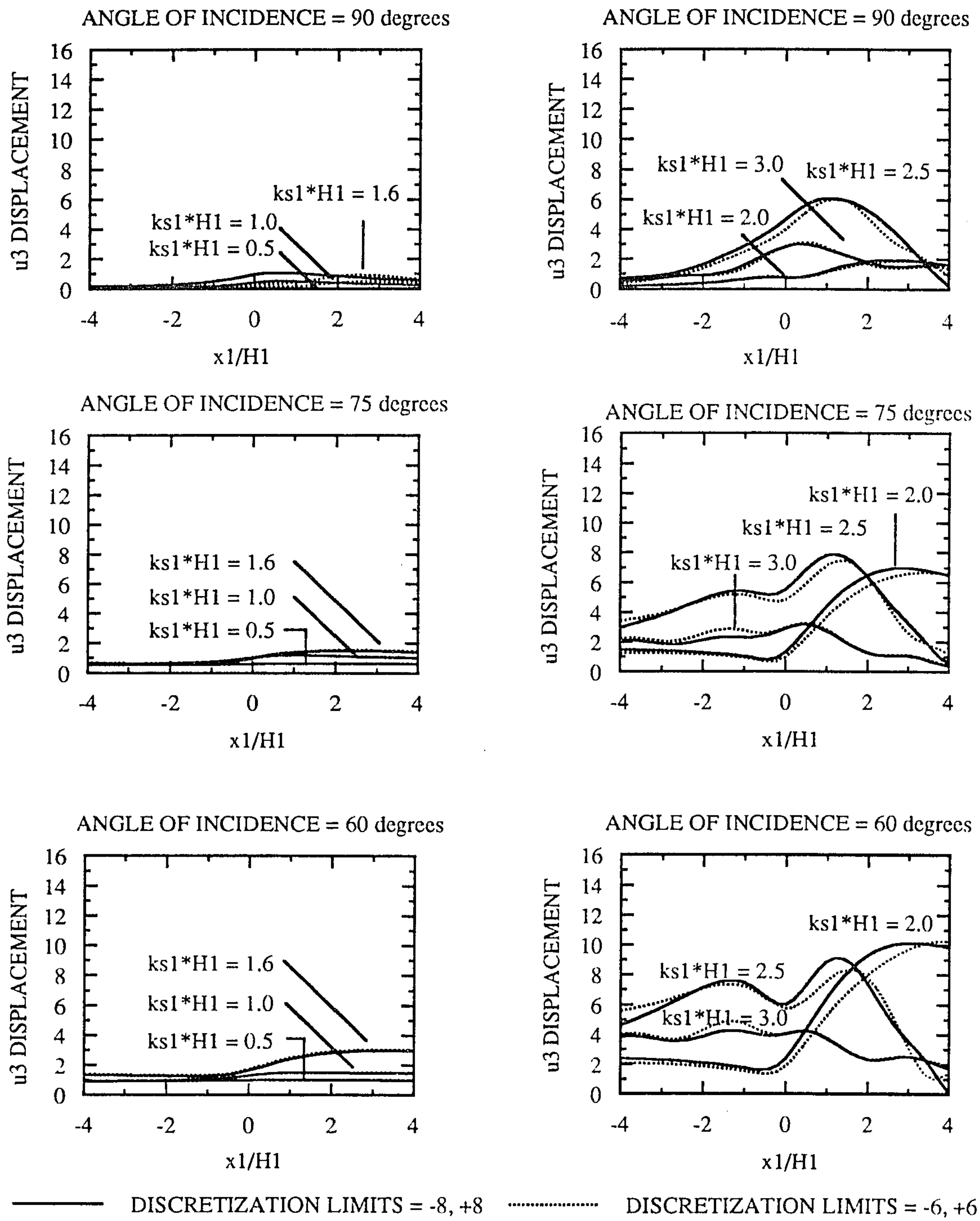


Figure 11-25: Comparison of Discretization Limits,
u3 Surface Displacement Due to a Unit Incident SV Wave,
Cs rock "A" / Cs soil = 2.5, H2 / H1 = 2.0

11.2.2: Incident P Wave

Results in the horizontal and vertical directions for an incident P wave are given in Figures 11-26 through 11-29 for a shear velocity ratio of 5.0 and in Figures 11-30 through 11-33 for a shear velocity ratio of 2.5 Increasing the thickness or decreasing the shear velocity of the embedded rock layer results in greater scattering to occur at the surface. The surface displacement plots indicate that as the angle of incidence decreases vertical displacement decreases, however for a given frequency the basic shape of each of the displacement curves does not significantly change. Except at ks1*H1 = 2.5, horizontal displacement increases as the angle of incidence decreases.

Plots of scattering limits due to the incident P wave are shown in Figures 11-34 and 11-35 for horizontal and vertical motion respectively. These scattering limit plots illustrate the increased area of scattering as the thickness of the embedded rock layer increases or the shear velocity of the embedded rock layer decreases. Large deviation from the one-dimensional solution begins at a lower frequency for the horizontal displacement than the vertical displacement. General trends for the scattering as a function of incidence angle are nonexistent.

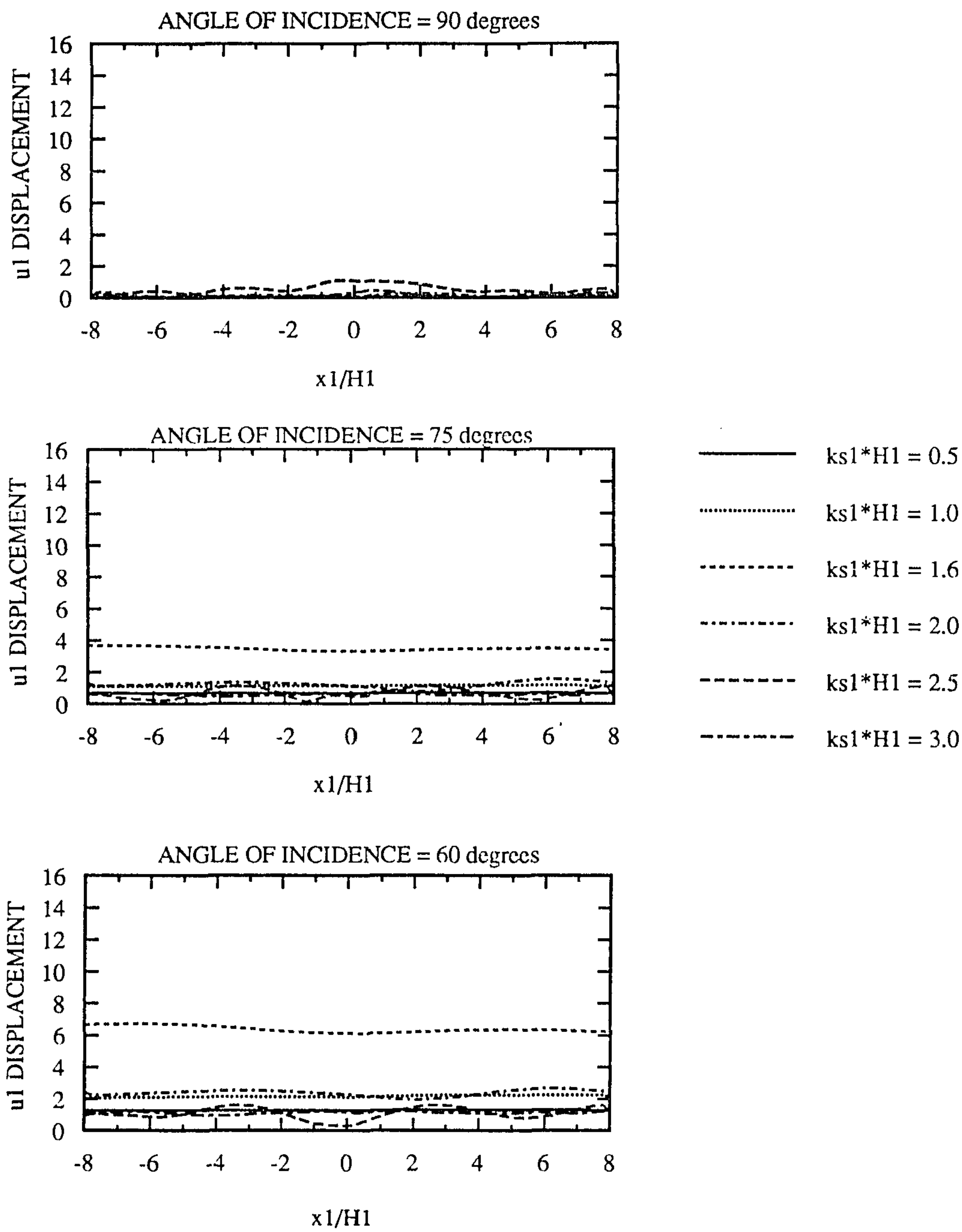


Figure 11-26: u1 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 1.0

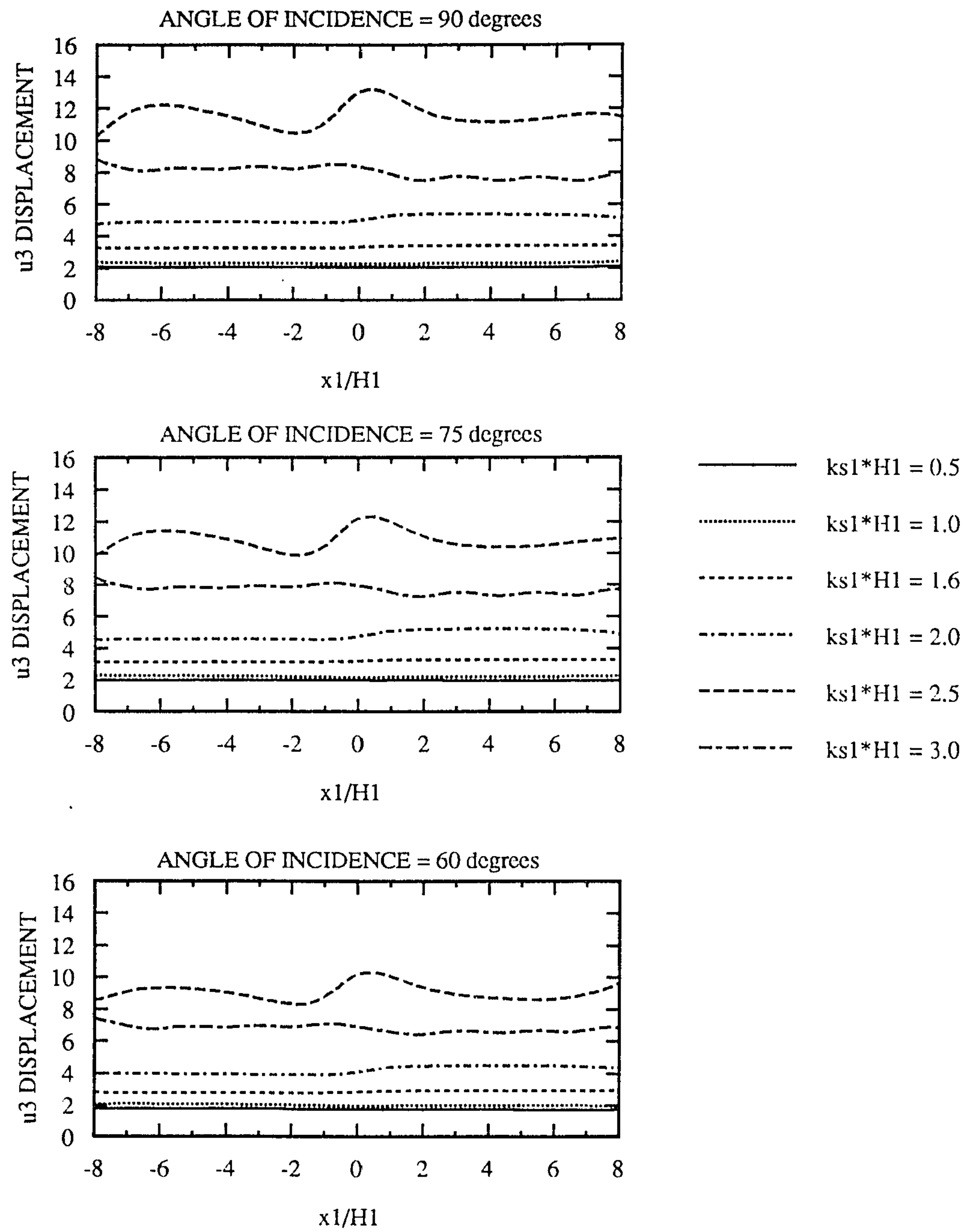


Figure 11-27: u3 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 1.0

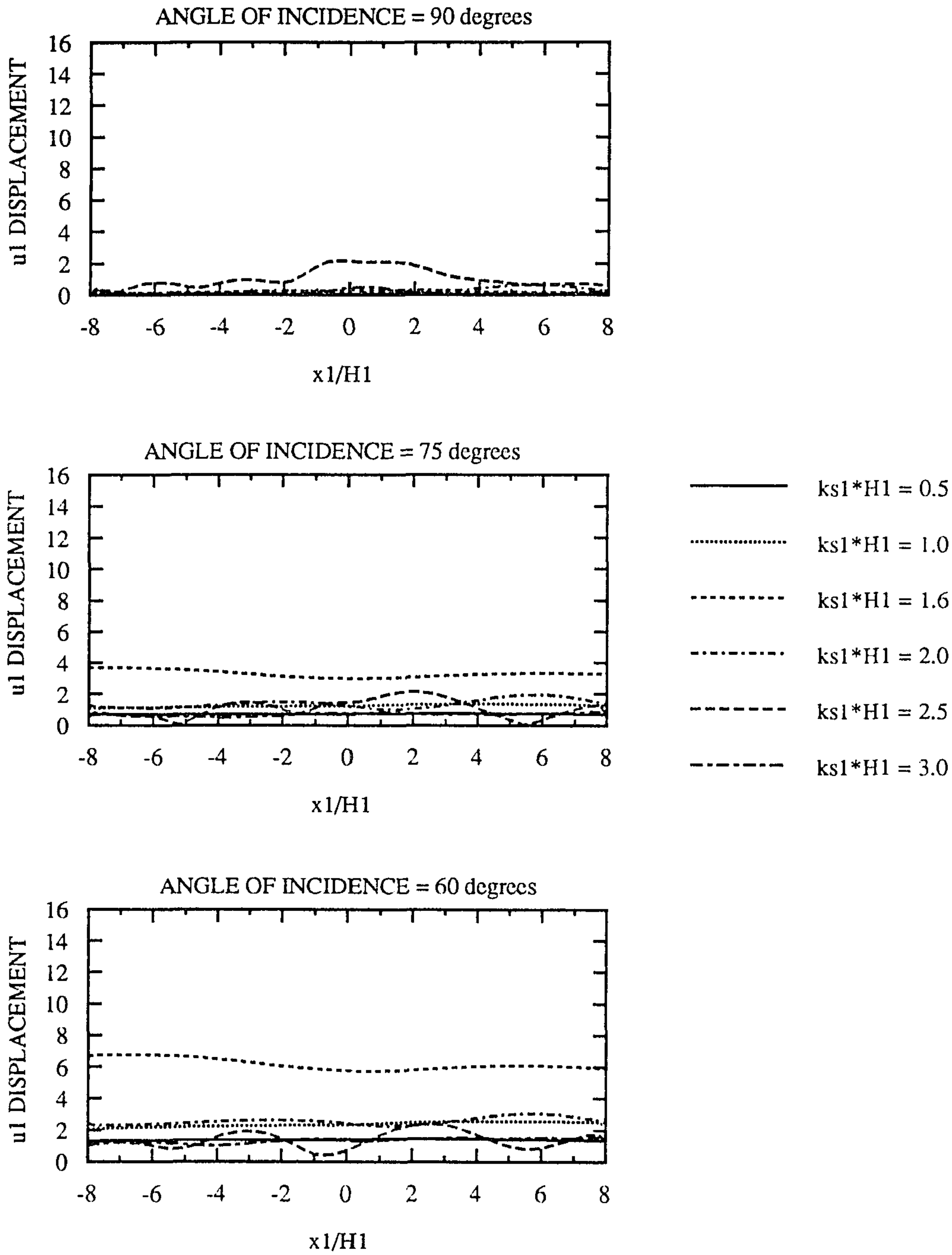


Figure 11-28: u1 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 2.0

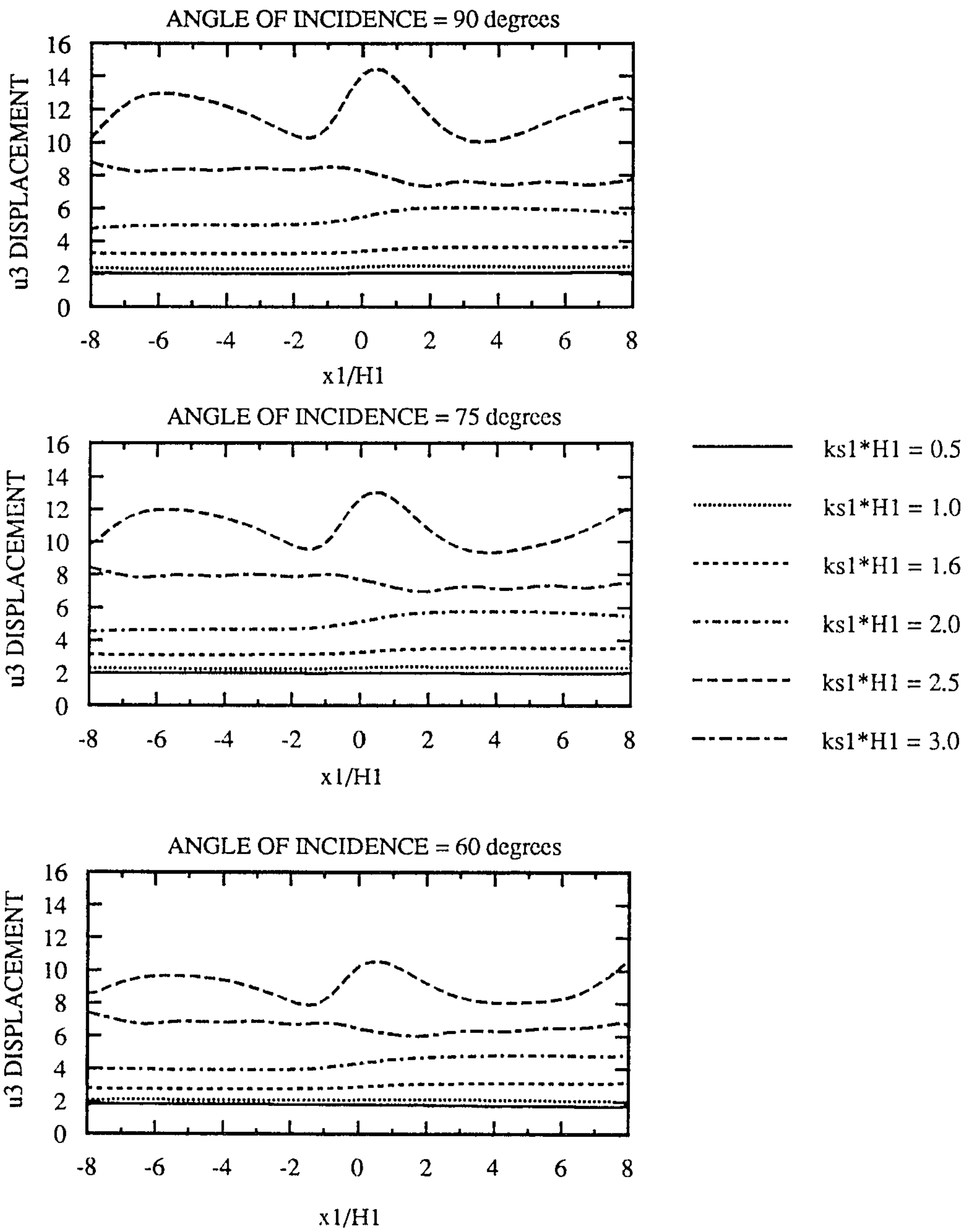


Figure 11-29: u3 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 5,
H2 / H1 = 2.0

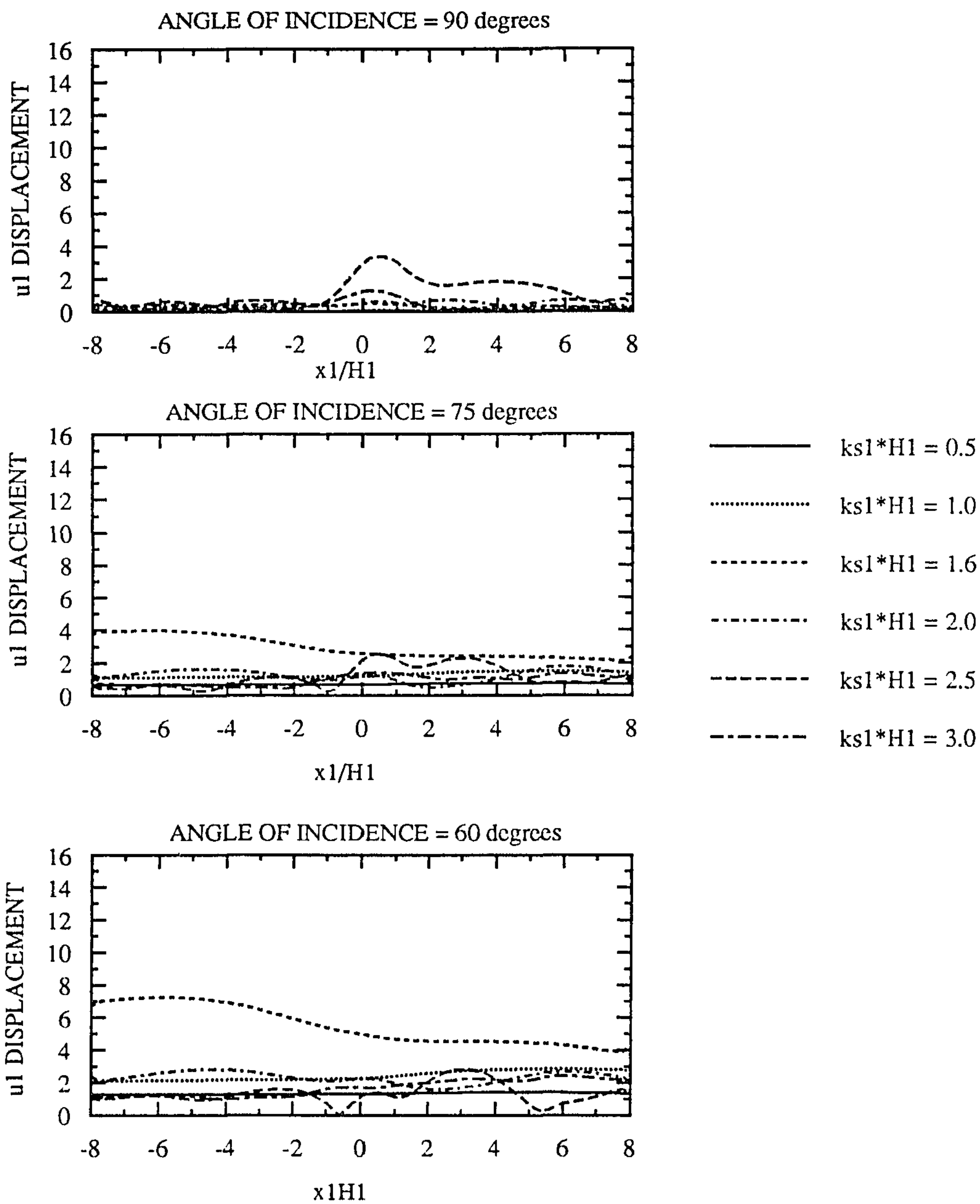


Figure 11-30: u1 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1.0

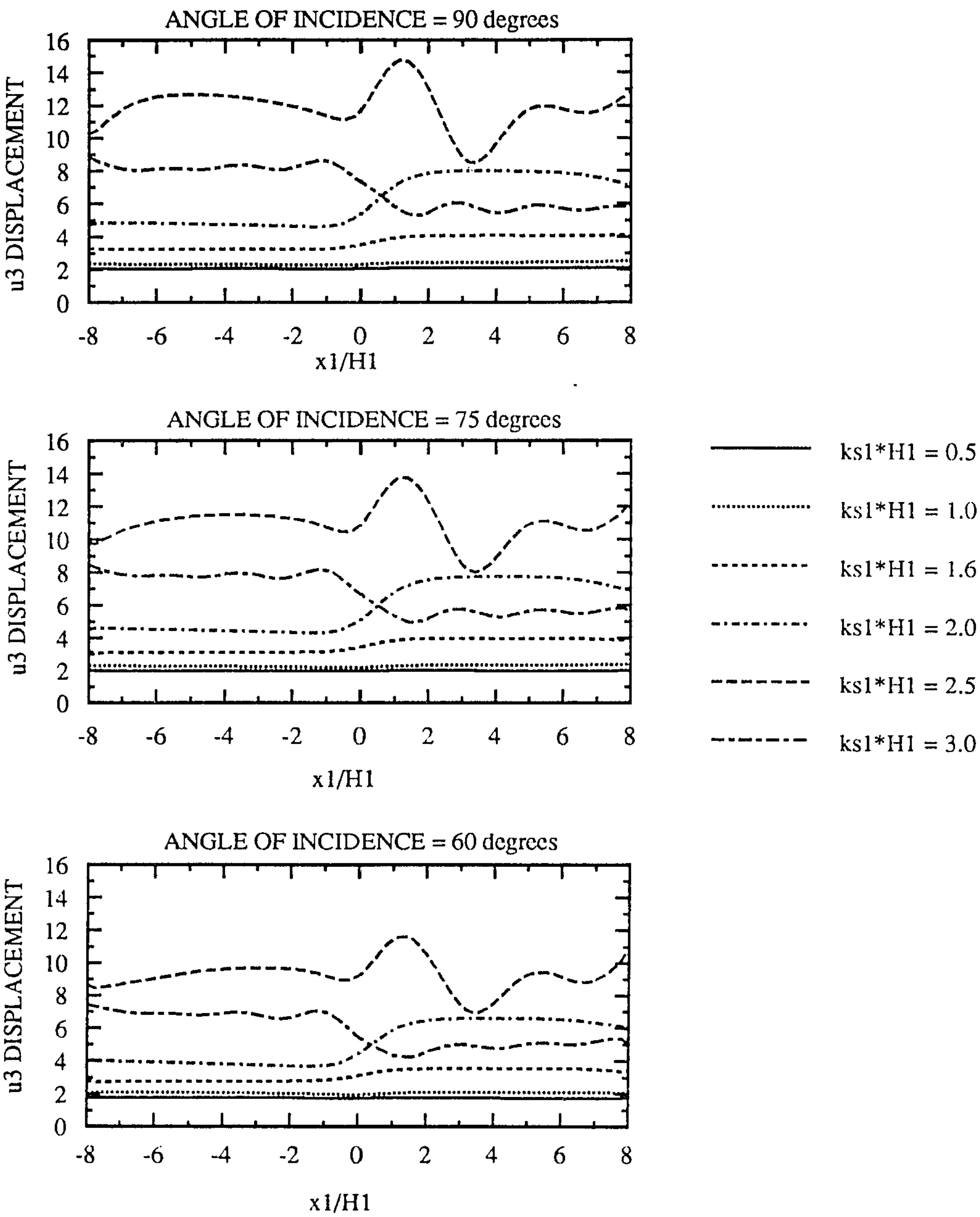


Figure 11-31: u3 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 1.0

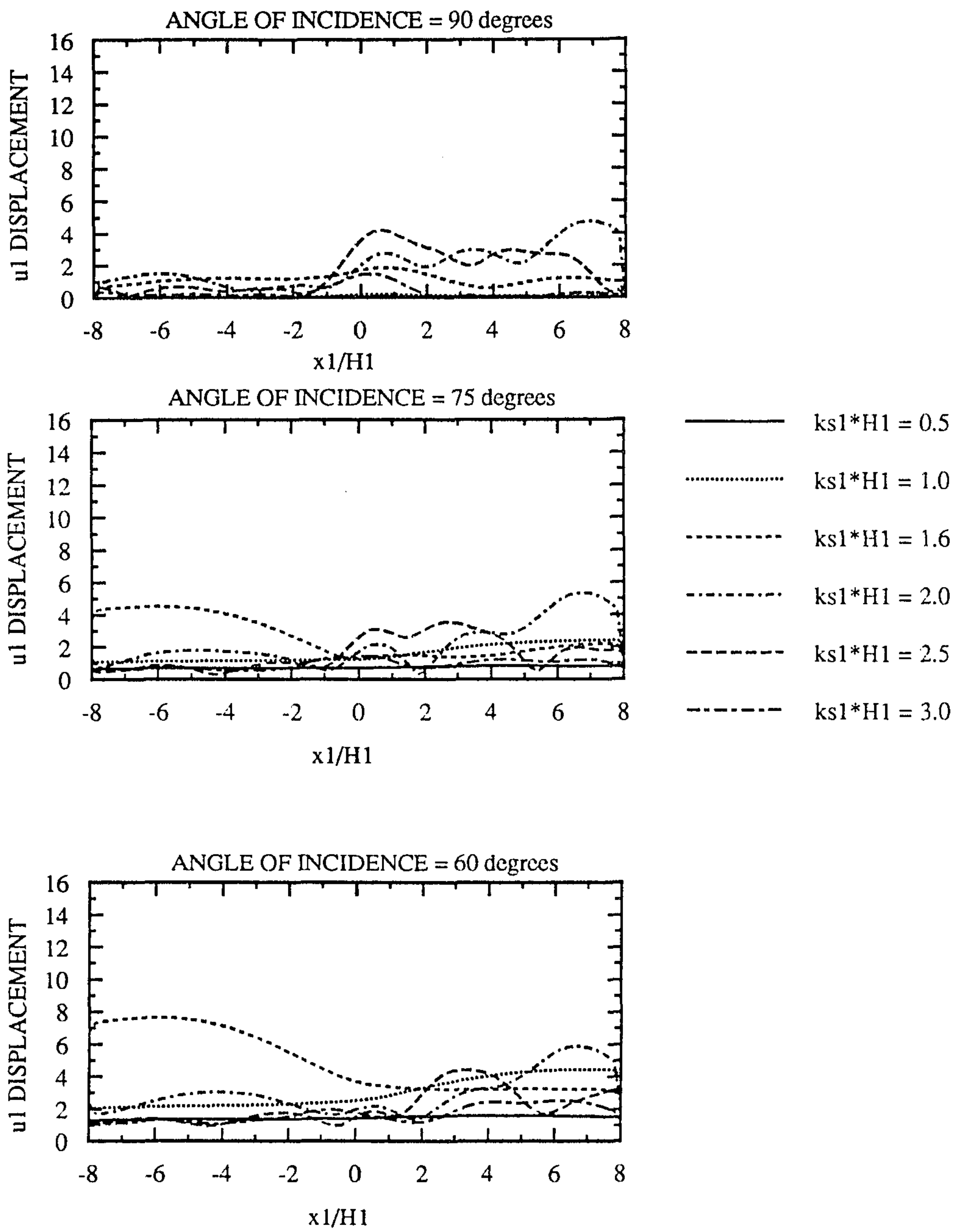


Figure 11-32: u1 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 2.0

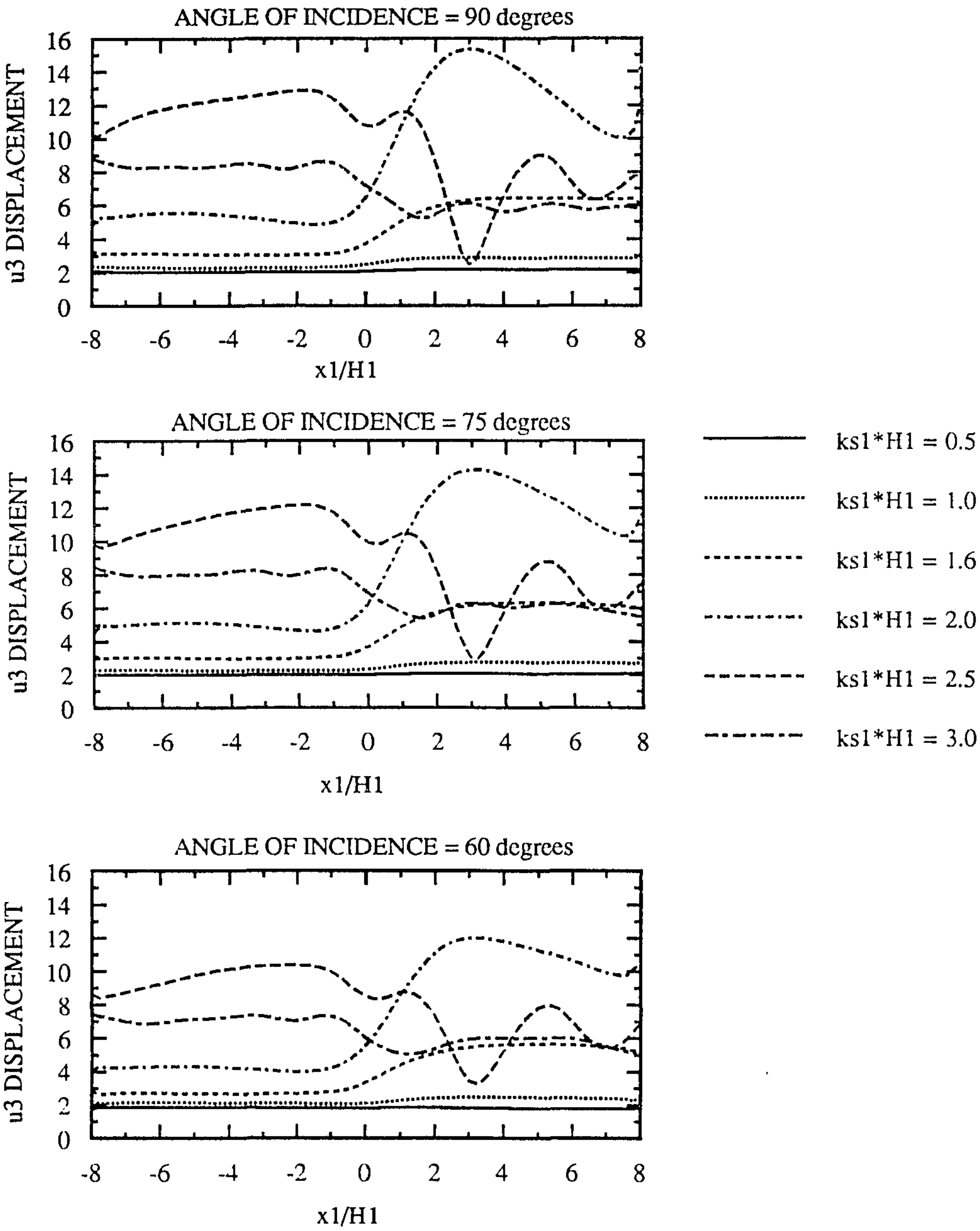


Figure 11-33: u3 Surface Displacement Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 2.5,
H2 / H1 = 2.0

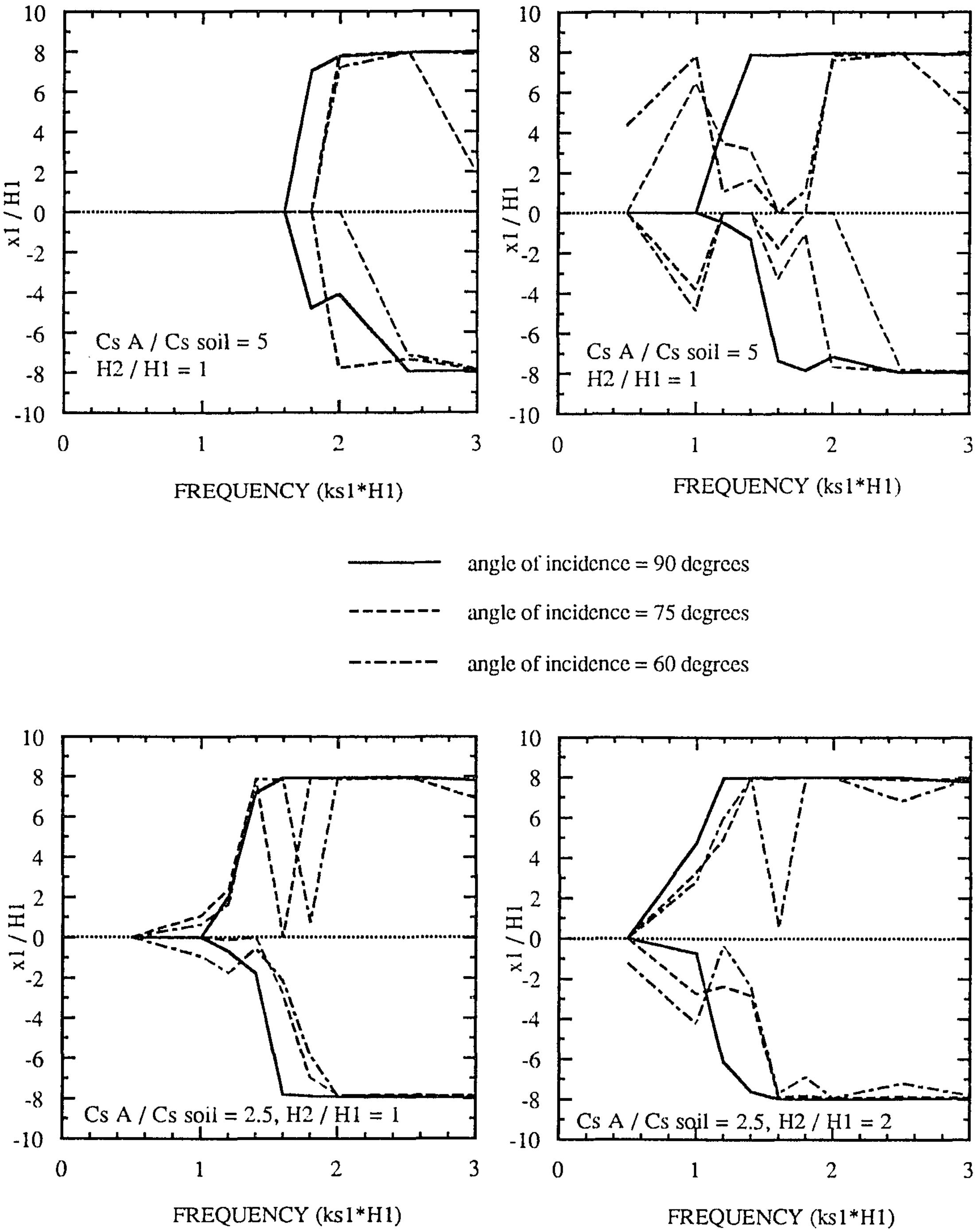


Figure 11-34: u1 Scattering Limits Due to a Unit Incident P Wave, Cs A / Cs soil = 2.5 and Cs A / Cs soil = 5, H2 / H1 = 1 and H2 / H1 = 2

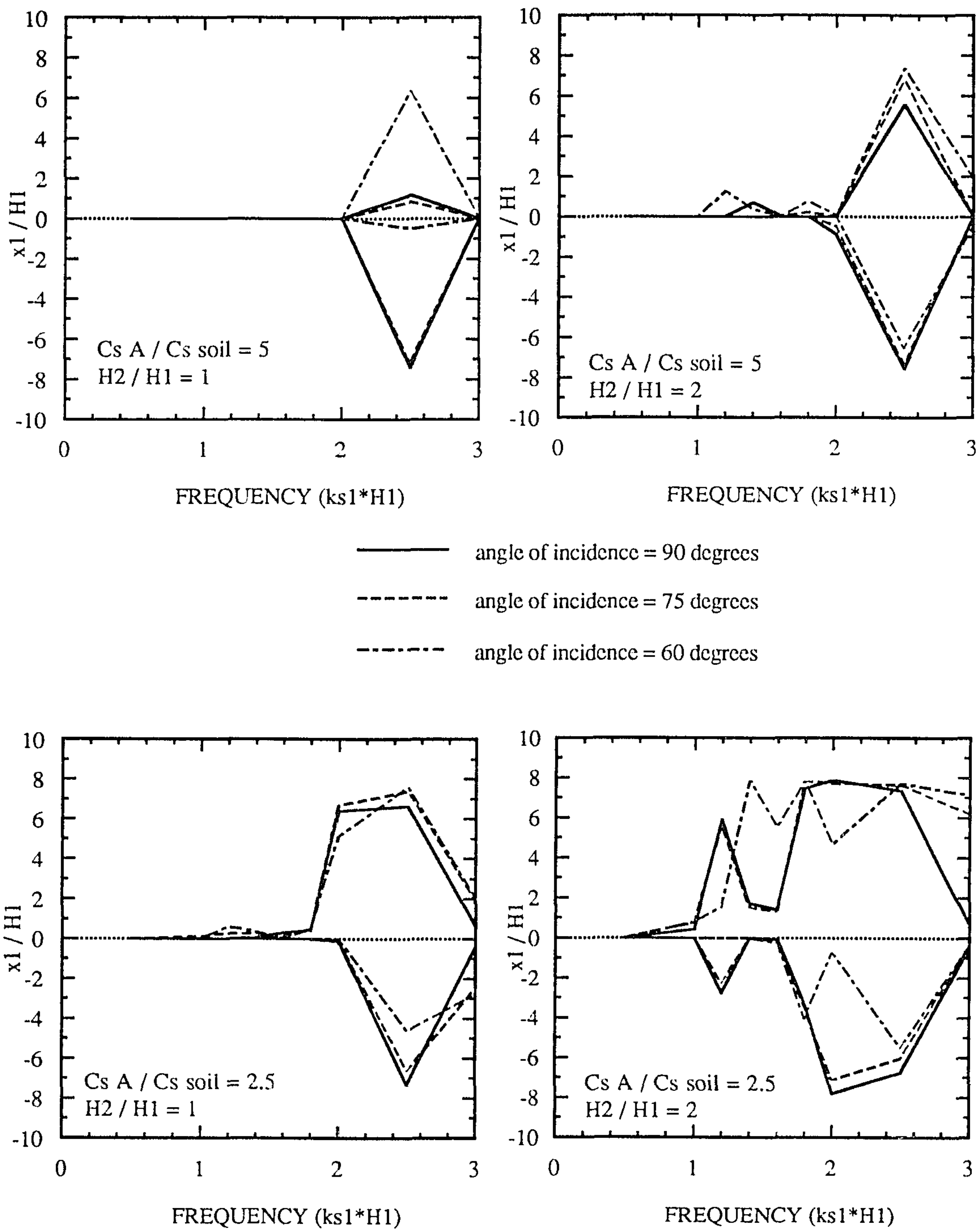


Figure 11-35: u3 Scattering Limits Due to a Unit Incident P Wave,
Cs rock "A" / Cs soil = 2.5 and Cs rock "A" / Cs soil = 5,
H2 / H1 = 1 and H2 / H1 = 2

Chapter 12
Conclusion

In this thesis a method has been described to determine the significance that an inhomogeneity in a rock half-space has on surface amplification due to earthquake motion. The particular soil-rock profile investigated is a soil layer resting on a rock half-space which includes a semi-infinite embedded rock layer. The upper surface of the embedded rock layer borders the soil layer. Surface displacements were determined for each type of body wave (SH, SV, and P) for a range of incidence angles using various shear velocity and thickness ratios of the embedded layer to the soil layer. For the case of SH wave motion, an additional case was studied in which the embedded layer was considered as a extension of the soil layer. Surface displacement plots along with scattering limit plots are included for each of the cases to summarize the effect that the discontinuity in the rock half-space has on surface amplification. An additional benefit of the scattering limit plots is the determination of the minimum distance away from the discontinuity, $x_1=0$, where a one-dimensional solution is satisfactory.

The solution to the two-dimensional problem was obtained using the boundary integral equation method. During the study, two boundary element codes, one for anti-plane motion and one for in-plane motion, were developed to solve for surface amplification of body waves due to input seismic motion. Although the truncation for elastic soil layers has been addressed elsewhere for anti-plane wave motion (Hadley et al., 1989), in this thesis the work has been extended to materials with damping and to in-plane motions. Also in this thesis, a new method has been developed for treating the truncation of a half-space.

Appendix
Computer Codes

The codes that were used for the two-dimensional analysis and the one-dimensional analysis are included in this appendix with their appropriate input files. The computer codes were programmed in terms of dimensionless properties to allow for parametric studies which can then be easily applied to a site specific study. The codes were generated by the writer.

A.1: Boundary Element Codes

For the two-dimensional study, the boundary integral equation method was used. Because of the difference in the influence functions used in the anti-plane and in-plane analysis, two codes were created: "shconvo" for the anti-plane motion due to SH waves and "psvconvo" for in-plane motion due to P and SV waves. The codes are independent of outside call libraries which therefore allows the programs to be used on any computer which has the capability to use a Fortran compiler. The codes include the ability to solve for complex Hankel functions and complex simultaneous equations. To solve simultaneous equations the Gaussian Elimination Method is used with maximum row and column pivoting.

Samples of input files are included for the boundary element codes. The input file for the two codes, "shconvo" and "psvconvo", are similar and only differ on the line indicating the type of half-space incident wave. The codes use free format so that the user does not have to be concerned with column justification. In the following, the input files for a cylindrical cavity and for a single soil layer on a half-space will be reviewed by describing the input information for each of the files.

Single Soil Layer on a Half-Space

Line 1 Output file name using a maximum of 20 characters and beginning at column 1.

Line 2 Title for output file using a maximum of 80 characters and beginning at column 1.

Line 3 Type of half-space incident wave. For the "shconvo" code the only option is to use SH. For the "psvconvo" code either P or SV should be used. The inclusion of this line for "shconvo" allows the same type of input file to be used for the two codes and for the possibility of the two codes to be meshed together in the future.

Line 4 Number of angle of incidences to be used and the values of the angles of incidences.
The values of the angles of incidence are given in degrees. Because of the method used in the formulation of the codes, only one angle of incidence is required to develop the influence matrices for the particular problem described in the input file.

Line 5 Dimensionless frequency to be used for the analysis. The transfer function and amplification of incident motion at each of the nodes is determined at this dimensionless frequency. The dimensionless frequency is equal to the product of the frequency of the incident wave (radians) and the characteristic length of the soil profile divided by the characteristic shear velocity of the soil profile. For a soil profile the characteristic properties would be the thickness and shear velocity for the top soil layer. For a half-space or a full-space with a cylindrical inclusion, the characteristic properties would be the radius of the cylinder and the shear velocity of the half-space.

Line 6 Number of homogeneous regions in the problem. A maximum of 4 can be used. Other than the half-space, the regions can include damping.

Begin input for region 1 properties:

Line 7 Shear velocity ratio with respect to area 1. Therefore, for region 1 this will equal 1.

Line 8 Poisson ratio for area 1. Although this is not required for anti-plane motion, this line is included to keep the same format as the in-plane motion code where it is required.

Line 9 Unit weight ratio for area 1 with respect to area 1. For area 1, this is equal to 1

Line 10 Hysteretic damping (%) for area 1. For cases in which the region is considered elastic, this value is equal to 0.0 .

Line 11 Number of boundaries that define region 1. For a soil layer without any inclusions, this value would be 2. For a soil layer with a single cylindrical inclusion, this value is equal to 3; the top boundary, the bottom boundary, and the boundary between the cylindrical inclusion and the soil layer.

Line 12 For boundary 1, the number of end points to define straight line segments that can be divided into equal subdivisions. This is not to be confused with the number of boundary elements. This foregoes the need for the user to input the coordinates of each element.

Line 13 x_1/H1 and x_3/H1 dimensionless coordinates for each of the two end-points for the first straight line segment. The coordinate system used follows the right hand rule with the x_3/H1 axis pointing down and the x_2/H1 axis pointing out of the paper. Coordinates are entered in a counter-clockwise fashion if you were standing inside of the particular domain.

Line 14 Number of equal elements to divide segment 1 into. The coordinates for each of the element end points is then determined by the code and defines the boundary elements in space.

Lines 13 and 14 are repeated for the number of straight line segments that are defined by the end points entered on line 12. For this case of a soil layer on a half-space only one straight line segment is included.

Line 15 For boundary 2, the number of end-points to define straight line segments that can be divided into equal subdivisions.

Line 16 x_1/H1 and x_3/H1 dimensionless coordinates of the end-points for the first straight line segment.

Line 17 Number of equal elements to divide segment 1 into. The coordinates for each of the end points is then determined by the code and defines the boundary elements in space.

Lines 16 and 17 are repeated for the number of straight line segments that are defined by the end points entered on line 15. For this case of a soil layer on a half-space only one straight line segment was used.

Input continues for the various regions in a similar fashion as that used for region 1 lines 7 through 17. In our case of a soil layer on rock, the next region to define is the half-space. For half-space problems that correct for truncation, the half-space must be elastic and its boundary a horizontal line. For full-space problems, which requires no truncation correction, or half-space regions which border the surface, the boundary can be non-horizontal since it will not include the truncation correction .

Line 18 Shear velocity ratio for the half-space. This represents the shear velocity of the half-space divided by the shear velocity of the soil layer.

Line 19 Poisson ratio for the half-space

Line 20 Unit weight ratio of the half-space. This is equal to the unit weight of the half-space divided by the unit weight of the soil layer.

Line 21 Hysteretic damping (%). When correction is made for a truncated half-space, the code assumes an elastic half-space. Therefore, the hysteretic damping is set equal to 0.0% .

Line 22 Number of boundaries for the half-space.

Line 23 Number of end points to subdivide the half-space boundary.

Line 24 End-point coordinates

Line 25 Number of elements to divide this straight segment defined by the above end-point coordinates into.

Individual region input information is completed.

Line 26 Number of homogeneous regions which have a traction free surface.

Line 27 For each of the regions of line 26, enter the number of boundaries segments which have a traction free surface. This line allows for the problem of an alluvial deposit with a traction free surface which would cause the area of the surrounding surface to have two traction free boundary segments.

Line 28 The first and last node of the first traction free boundary segment

Line 28 is repeated for the number of segments declared on line 27.

Lines 27 and 28 are repeated for the next homogeneous region with a traction free surface. This input procedure is reproduced for the number of homogeneous regions of line 26.

Line 29 Half-space region number. This defines which of the area input properties are to be assigned to the half-space region.

The next section identifies the node numbers that will be used for the correction for the soil layer truncation. The nodes define the soil profile that is used for the determination of displacements and tractions from a one-dimensional analysis. Because of the various configurations that are possible, a soil profile is needed to be defined at both limits of the $-x_1$ and $+x_1$ discretized areas.

Line 30 Number of areas along the $-x_1$ vertical imaginary boundary that will be used to define a one-dimensional soil profile. See section on correction for layer correction for description of imaginary boundary.

Line 31 Top node number and bottom node number of soil layer 1 in one-dimensional soil profile.

Line 31 input is repeated for the number of soil layers of line 30.

Line 32 Number of areas along the $+x_1$ vertical imaginary boundary that will be used to define a one-dimensional soil profile.

Line 33 Top node number and bottom node number of soil layer 1 for the one-dimensional soil profile needed to define $+x_1$ vertical imaginary boundary.

Line 33 input is repeated for the number of soil layers of line 32.

SAMPLE INPUT FILE FOR THE TWO-DIMENSIONAL ANLAYSIS
OF A SOIL LAYER ON A HALF-SPACE

LINE #	COLUMN # OF INPUT FILE 1
1	SH1layer.ouD
2	1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE
3	SH
4	4 90.0 75.0 60.0 45.0
5	1.5708
6	2
7	1.
8	.25
9	1.0
10	5.0
11	2
12	2
13	+2.5 0.0 -2.50 0.0
14	20
15	2
16	-2.5 1.0 +2.50 1.0
17	20
18	10.
19	.25
20	1.32
21	0.0
22	1
23	2
24	+2.5 1.0 -2.50 1.0
25	20
26	1
27	1
28	1 21
29	2
30	1
31	21 22
32	1
33	1 42

An example is included of an input file for a cylindrical surface in a full-space to identify the differences in the input when a cylindrical surface is used and/or a full-space is used.

Cylindrical Cavity in a Full-Space

Line 1	Output file name using a maximum of 20 characters and beginning at column 1.
Line 2	Title for output file using a maximum of 80 characters and beginning at column 1.
Line 3	Type of half-space incident wave.
Line 4	Number of incidence angles to be used and the values of the incidence angles in degrees.
Line 5	Dimensionless frequency to be used for the analysis.
Line 6	Number of homogeneous regions in the problem. A maximum of 4 can be used.
	Begin input for region 1 properties:
Line 7	Shear velocity ratio with respect to area 1. Therefore, for region 1 this will equal 1.
Line 8	Poisson ratio for area 1.
Line 9	Unit weight ratio for area 1 with respect to area 1. For area 1, this is equal to 1
Line 10	Hysteretic damping (%) for area 1.
Line 11	Number of boundaries that define region 1.
Line 12	For boundary 1 which is a cylindrical surface, the number of end-points is equal to 2
Line 13	$x_1/H1$ and $x_3/H1$ dimensionless coordinates of the end points . The values will be the coordinates of the first node of the cylindrical boundary. If the values of the second end point are equal to the values of the first, the boundary will be defined as being closed and cylindrical.
Line 14	For a cylindrical boundary, enter the center coordinates.
Line 15	Direction of the rotation in the direction of increasing node numbers if the center of the circle is used as reference and the number of equal segments to subdivide the boundary. For counter-clockwise rotation enter -1 ; this is what is used for the area within the cylindrical boundary. For clockwise rotation enter 1 ; this is what is used for the area outside of the cylindrical boundary.

Line 16	Number of homogeneous regions which have a traction free surface.
Line 17	For each of the regions of line 26, enter the number of boundaries segments which have a traction free surface.
Line 18	The first and last node of the first traction free boundary segment
Line 19	Half-space region number or in this case the full-space region number.

For full-space problems, there is no truncation correction and the one-dimensional soil profiles are not applicable.

SAMPLE INPUT FILE FOR A CYLINDRICAL CAVITY

LINE #	COLUMN # OF INPUT FILE 1
1	SHcylcav.ou
2	CYLINDRICAL CAVITY IN FULL SPACE
3	SH
4	1 90.0
5	3.0000
6	1
7	1.
8	0.33
9	1.0
10	0.0
11	1
12	2
13	+1.0000 +0.0000 +1.0000 +0.0000
14	+0.0000 +0.0000
15	+1 20
16	1
17	1
18	1 21
19	1

In order to allow for less user interaction a run file is used which permits for many problems to be run sequentially. A sample of a run file is include which includes the number of problems and the names of the input files for each of the problems. The output files from the boundary element codes include an echo of the input information. Node information is given for each node including its coordinates, displacement (complex), magnitude of displacement, and traction (complex). Output files are included for the two previously discussed cases

SAMPLE INPUT FOR A RUN FILE

LINE #	COLUMN # OF RUN FILE 1
1	3
2	INPUT.1
3	INPUT.2
4	INPUT.3

SAMPLE OUTPUT FILE FOR A SINGLE LAYER ON A HALF-SPACE

```
1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING

 TYPE OF WAVE = SH
 DIMENSIONLESS FREQUENCY (omega*H1/Cs1) =  1.5708

 ADMITTANCE FUNCTIONS ARE CALCULATED FOR 4 ANGLES
 INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs
 INCIDENT ANGLE (2) WITH HORIZONTAL = 75.00 degs
 INCIDENT ANGLE (3) WITH HORIZONTAL = 60.00 degs
 INCIDENT ANGLE (4) WITH HORIZONTAL = 45.00 degs
 NUMBER OF HOMOGENEOUS SUBREGIONS =  2


 HOMOGENEOUS SOIL REGION : 1

 Cs(area 1)/Cs(area 1) =     1.000
 UNIT WT(area 1)/UNIT WT(area 1) = 1.000
 POISSIONS RATIO 0.250
 DAMPING  5.00 %


 NUMBER OF BOUNDARIES  2

 NO. OF BOUNDARY ELEMENTS FOR BOUNDARY 1= 20
 NO. OF BOUNDARY ELEMENTS FOR BOUNDARY 2= 20

 BOUNDARY NODE COORDINATES        X/H(layer1)         Z/H(layer1)

 BOUNDARY =  1
                    1               2.5000          0.0000
                    2               2.2500          0.0000
                    3               2.0000          0.0000
                    4               1.7500          0.0000
                    5               1.5000          0.0000
                    6               1.2500          0.0000
                    7               1.0000          0.0000
                    8               0.7500          0.0000
                    9               0.5000          0.0000
                   10               0.2500          0.0000
                   11               0.0000          0.0000
                   12              -0.2500          0.0000
                   13              -0.5000          0.0000
                   14              -0.7500          0.0000
                   15              -1.0000          0.0000
                   16              -1.2500          0.0000
                   17              -1.5000          0.0000
                   18              -1.7500          0.0000
                   19              -2.0000          0.0000
                   20              -2.2500          0.0000
                   21              -2.5000          0.0000

 BOUNDARY =  2
                   22              -2.5000          1.0000
                   23              -2.2500          1.0000
                   24              -2.0000          1.0000
                   25              -1.7500          1.0000
                   26              -1.5000          1.0000
                   27              -1.2500          1.0000
                   28              -1.0000          1.0000
                   29              -0.7500          1.0000
                   30              -0.5000          1.0000
```

31	-0.2500	1.0000
32	0.0000	1.0000
33	0.2500	1.0000
34	0.5000	1.0000
35	0.7500	1.0000
36	1.0000	1.0000
37	1.2500	1.0000
38	1.5000	1.0000
39	1.7500	1.0000
40	2.0000	1.0000
41	2.2500	1.0000
42	2.5000	1.0000

BOUNDARY 1 IS OPEN
BOUNDARY 2 IS OPEN

HOMOGENEOUS SOIL REGION : 2

Cs(area 2)/Cs(area 1) = 10.000
UNIT WT(area 2)/UNIT WT(area 1) = 1.320
POISSIONS RATIO 0.250
DAMPING 0.00 %

NUMBER OF BOUNDARIES 1

NO. OF BOUNDARY ELEMENTS FOR BOUNDARY 1= 20

BOUNDARY NODE COORDINATES	X/H(layer1)	Z/H(layer1)
BOUNDARY = 1		
43	2.5000	1.0000
44	2.2500	1.0000
45	2.0000	1.0000
46	1.7500	1.0000
47	1.5000	1.0000
48	1.2500	1.0000
49	1.0000	1.0000
50	0.7500	1.0000
51	0.5000	1.0000
52	0.2500	1.0000
53	0.0000	1.0000
54	-0.2500	1.0000
55	-0.5000	1.0000
56	-0.7500	1.0000
57	-1.0000	1.0000
58	-1.2500	1.0000
59	-1.5000	1.0000
60	-1.7500	1.0000
61	-2.0000	1.0000
62	-2.2500	1.0000
63	-2.5000	1.0000

BOUNDARY 1 IS OPEN

INTERFACE NODES

22 - 63
23 - 62
24 - 61
25 - 60
26 - 59

```
        27 -  58
        28 -  57
        29 -  56
        30 -  55
        31 -  54
        32 -  53
        33 -  52
        34 -  51
        35 -  50
        36 -  49
        37 -  48
        38 -  47
        39 -  46
        40 -  45
        41 -  44
        42 -  43

    3 NODE INTERFACES

         NONE

    SURFACE NODES

    AREA 1: NODE   1 THROUGH NODE  21


  FREE-FIELD COLUMN DESCRIPTION

  -X REGION
  LAYER            d(i)/d(1)  Cs(layer i)/Cs(1)        WT(layer i)/WT(1)
DAMPING
                                                                               (%)

    1                1.00          1.00                      1.00
5.000
HALF-SPACE                        10.00                      1.32
0.000

  +X REGION
  LAYER            d(i)/d(1)  Cs(layer i)/Cs(1)        WT(layer i)/WT(1)
DAMPING
                                                                               (%)

    1                1.00          1.00                      1.00
5.000
HALF-SPACE                        10.00                      1.32
0.000

  SCATTERING BOUNDARY OCCURS AT X/H1 =    0.00


  1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING

  TYPE OF WAVE = SH
  DIMENSIONLESS FREQUENCY (omega*H1/Cs1) =  1.571
  INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs


  INCIDENT WAVE DISPLACEMENTS
    HALF-SPACE    NODE            INCDT V DISPL

                   43              0.988 -0.156
                   44              0.988 -0.156
```

45	0.988 -0.156
46	0.988 -0.156
47	0.988 -0.156
48	0.988 -0.156
49	0.988 -0.156
50	0.988 -0.156
51	0.988 -0.156
52	0.988 -0.156
53	0.988 -0.156
54	0.988 -0.156
55	0.988 -0.156
56	0.988 -0.156
57	0.988 -0.156
58	0.988 -0.156
59	0.988 -0.156
60	0.988 -0.156
61	0.988 -0.156
62	0.988 -0.156
63	0.988 -0.156

FREE-FIELD CALCULATION

-X BOUNDARY

LAYER	Ash		Bsh	
1	1.101	6.390	1.101	6.390
2	1.000	0.000	0.033	0.055

	REAL	IMAGINARY	MAGNITUDE
V(top)	2.203	12.779	12.968

LAYER = 1

NODE	X/H1	Z/H1	DISPLACEMENT V		STRESS SZY*H1/Gi		SXY*H1/Gi*	
1	-2.500	0.000	2.203	12.779	0.000	0.000	0.000	0.000
2	-2.500	0.200	2.158	12.149	-1.080	-6.202	0.000	0.000
3	-2.500	0.400	2.020	10.321	-2.114	-11.793	0.000	0.000
4	-2.500	0.600	1.786	7.473	-3.057	-16.220	0.000	0.000
5	-2.500	0.800	1.450	3.883	-3.860	-19.046	0.000	0.000
6	-2.500	1.000	1.011	-0.097	-4.472	-19.988	0.000	0.000

HALF-SPACE

DISPLACEMENT:
v/exp(ikx) = 1.011 -0.097

STRESS:
(Szy*H1/Ghs) /exp(ikx) = -0.034 -0.151
(Sxy*H1/Ghs*)/exp(ikx) = 0.000 0.000

ROCK OUTCROP

DISPLACEMENT:
u = 1.975 -0.313 |u| = 2.000

STRESS:
(Szy*H1/Ghs) /exp(ikx) = 0.000 0.000
(Sxy*H1/Ghs*)/exp(ikx) = 0.000 0.000

+X BOUNDARY

LAYER	Ash		Bsh	
1	1.101	6.390	1.101	6.390
2	1.000	0.000	0.033	0.055

	REAL	IMAGINARY	MAGNITUDE
V(top)	2.203	12.779	12.968

LAYER = 1

			DISPLACEMENT		STRESS			
NODE	X/H1	Z/H1	V		SZY*H1/Gi		SXY*H1/Gi*	
1	2.500	1.000	1.011	-0.097	-4.472	-19.988	0.000	0.000
2	2.500	0.800	1.450	3.883	-3.860	-19.046	0.000	0.000
3	2.500	0.600	1.786	7.473	-3.057	-16.220	0.000	0.000
4	2.500	0.400	2.020	10.321	-2.114	-11.793	0.000	0.000
5	2.500	0.200	2.158	12.149	-1.080	-6.202	0.000	0.000
6	2.500	0.000	2.203	12.779	0.000	0.000	0.000	0.000

HALF-SPACE

DISPLACEMENT:
v/exp(ikx) = 1.011 -0.097

STRESS:
(Szy*H1/Ghs) /exp(ikx) = -0.034 -0.151
(Sxy*H1/Ghs*)/exp(ikx) = 0.000 0.000

ROCK OUTCROP

DISPLACEMENT:
u = 1.975 -0.313 |u| = 2.000

STRESS:
(Szy*H1/Ghs) /exp(ikx) = 0.000 0.000
(Sxy*H1/Ghs*)/exp(ikx) = 0.000 0.000

OUTPUT

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING
TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs
NUMBER OF HOMOGENEOUS REGIONS = 2

BOUNDARY VALUES

HOMOGENEOUS REGION 1
NUMBER OF BOUNDARIES = 2

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL	|V|

1	2.500	0.000	2.203	12.779	12.968
2	2.250	0.000	2.513	12.659	12.906
3	2.000	0.000	2.561	12.609	12.867
4	1.750	0.000	2.605	12.562	12.829
5	1.500	0.000	2.646	12.519	12.795
6	1.250	0.000	2.682	12.481	12.766
7	1.000	0.000	2.713	12.449	12.741
8	0.750	0.000	2.738	12.424	12.722
9	0.500	0.000	2.756	12.405	12.708
10	0.250	0.000	2.767	12.394	12.699
11	0.000	0.000	2.771	12.390	12.696
12	-0.250	0.000	2.767	12.394	12.699
13	-0.500	0.000	2.756	12.405	12.708
14	-0.750	0.000	2.738	12.424	12.722
15	-1.000	0.000	2.713	12.449	12.741
16	-1.250	0.000	2.682	12.481	12.766
17	-1.500	0.000	2.646	12.519	12.795
18	-1.750	0.000	2.605	12.562	12.829
19	-2.000	0.000	2.561	12.609	12.867
20	-2.250	0.000	2.513	12.659	12.906
21	-2.500	0.000	2.203	12.779	12.968

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000
6	0.0000	0.0000
7	0.0000	0.0000
8	0.0000	0.0000
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0000	0.0000
12	0.0000	0.0000
13	0.0000	0.0000
14	0.0000	0.0000
15	0.0000	0.0000
16	0.0000	0.0000
17	0.0000	0.0000
18	0.0000	0.0000
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0000	0.0000

BOUNDARY = 2
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
22	-2.500	1.000	1.011	-0.097	1.016
23	-2.250	1.000	1.033	-0.090	1.037
24	-2.000	1.000	1.034	-0.091	1.038
25	-1.750	1.000	1.036	-0.091	1.040
26	-1.500	1.000	1.037	-0.092	1.041
27	-1.250	1.000	1.038	-0.093	1.042
28	-1.000	1.000	1.039	-0.094	1.043
29	-0.750	1.000	1.039	-0.094	1.044
30	-0.500	1.000	1.040	-0.095	1.044
31	-0.250	1.000	1.040	-0.095	1.045
32	0.000	1.000	1.040	-0.095	1.045

33	0.250	1.000	1.040	-0.095	1.045
34	0.500	1.000	1.040	-0.095	1.044
35	0.750	1.000	1.039	-0.094	1.044
36	1.000	1.000	1.039	-0.094	1.043
37	1.250	1.000	1.038	-0.093	1.042
38	1.500	1.000	1.037	-0.092	1.041
39	1.750	1.000	1.036	-0.091	1.040
40	2.000	1.000	1.034	-0.091	1.038
41	2.250	1.000	1.033	-0.090	1.037
42	2.500	1.000	1.011	-0.097	1.016

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
22	-6.2143	-19.7522
23	-4.7451	-20.0933
24	-5.1129	-19.7148
25	-5.1452	-19.6404
26	-5.2282	-19.5521
27	-5.2902	-19.4848
28	-5.3435	-19.4298
29	-5.3865	-19.3898
30	-5.4172	-19.3596
31	-5.4361	-19.3421
32	-5.4424	-19.3363
33	-5.4361	-19.3421
34	-5.4172	-19.3596
35	-5.3865	-19.3898
36	-5.3435	-19.4298
37	-5.2902	-19.4848
38	-5.2282	-19.5521
39	-5.1452	-19.6404
40	-5.1129	-19.7148
41	-4.7451	-20.0933
42	-6.2143	-19.7522

HOMOGENEOUS REGION 2
NUMBER OF BOUNDARIES = 1

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
43	2.500	1.000	1.011	-0.097	1.016
44	2.250	1.000	1.033	-0.090	1.037
45	2.000	1.000	1.034	-0.091	1.038
46	1.750	1.000	1.036	-0.091	1.040
47	1.500	1.000	1.037	-0.092	1.041
48	1.250	1.000	1.038	-0.093	1.042
49	1.000	1.000	1.039	-0.094	1.043
50	0.750	1.000	1.039	-0.094	1.044
51	0.500	1.000	1.040	-0.095	1.044
52	0.250	1.000	1.040	-0.095	1.045
53	0.000	1.000	1.040	-0.095	1.045
54	-0.250	1.000	1.040	-0.095	1.045
55	-0.500	1.000	1.040	-0.095	1.044
56	-0.750	1.000	1.039	-0.094	1.044
57	-1.000	1.000	1.039	-0.094	1.043
58	-1.250	1.000	1.038	-0.093	1.042
59	-1.500	1.000	1.037	-0.092	1.041
60	-1.750	1.000	1.036	-0.091	1.040

61	-2.000	1.000	1.034	-0.091	1.038
62	-2.250	1.000	1.033	-0.090	1.037
63	-2.500	1.000	1.011	-0.097	1.016

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
43	0.0471	0.1496
44	0.0359	0.1522
45	0.0387	0.1494
46	0.0390	0.1488
47	0.0396	0.1481
48	0.0401	0.1476
49	0.0405	0.1472
50	0.0408	0.1469
51	0.0410	0.1467
52	0.0412	0.1465
53	0.0412	0.1465
54	0.0412	0.1465
55	0.0410	0.1467
56	0.0408	0.1469
57	0.0405	0.1472
58	0.0401	0.1476
59	0.0396	0.1481
60	0.0390	0.1488
61	0.0387	0.1494
62	0.0359	0.1522
63	0.0471	0.1496

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING

TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (2) WITH HORIZONTAL = 75.00 degs

INCIDENT WAVE DISPLACEMENTS

HALF-SPACE	NODE	INCDT V DISPL	
	43	0.999	-0.050
	44	0.998	-0.060
	45	0.998	-0.070
	46	0.997	-0.080
	47	0.996	-0.091
	48	0.995	-0.101
	49	0.994	-0.111
	50	0.993	-0.121
	51	0.991	-0.131
	52	0.990	-0.141
	53	0.989	-0.151
	54	0.987	-0.161
	55	0.985	-0.171
	56	0.983	-0.181
	57	0.982	-0.191
	58	0.980	-0.201
	59	0.977	-0.211
	60	0.975	-0.221
	61	0.973	-0.231
	62	0.971	-0.241
	63	0.968	-0.251

```
   FREE-FIELD CALCULATION


   -X BOUNDARY

 LAYER        Ash             Bsh

   1     1.064  6.283   1.064  6.283
   2     1.000  0.000   0.017  0.063


                REAL          IMAGINARY       MAGNITUDE

 V(top)         2.127          12.567          12.745

   LAYER = 1
                      DISPLACEMENT                                     STRESS
NODE  X/H1    Z/H1           V                               SZY*H1/Gi      SXY*H1/Gi*

  1  -2.500  0.000  3.391 12.286                            0.000  0.000  -0.499  0.138
  2  -2.500  0.200  3.286 11.675                           -1.655 -5.958  -0.475  0.134
  3  -2.500  0.400  2.975  9.901                           -3.206-11.323  -0.403  0.121
  4  -2.500  0.600  2.471  7.139                           -4.556-15.561  -0.290  0.100
  5  -2.500  0.800  1.796  3.662                           -5.615-18.247  -0.149  0.073
  6  -2.500  1.000  0.982 -0.187                           -6.304-19.112   0.008  0.040

   HALF-SPACE

 DISPLACEMENT:
   v/exp(ikx) =                0.996  -0.086

 STRESS:
   (Szy*H1/Ghs) /exp(ikx) =  -0.033  -0.149
   (Sxy*H1/Ghs*)/exp(ikx) =   0.003   0.040

   ROCK OUTCROP

 DISPLACEMENT:
   u =                         1.977  -0.302    |u| =  2.000

 STRESS:
   (Szy*H1/Ghs) /exp(ikx) =   0.000   0.000
   (Sxy*H1/Ghs*)/exp(ikx) =   0.012   0.080


   +X BOUNDARY

 LAYER        Ash             Bsh

   1     1.064  6.283   1.064  6.283
   2     1.000  0.000   0.017  0.063


                REAL          IMAGINARY       MAGNITUDE

 V(top)         2.127          12.567          12.745

   LAYER = 1
                      DISPLACEMENT                                     STRESS
NODE  X/H1    Z/H1           V                               SZY*H1/Gi      SXY*H1/Gi*

  1   2.500  1.000  0.999  0.016                           -2.316-19.991  -0.001  0.041
  2   2.500  0.800  1.019  3.949                           -1.815-19.005  -0.161  0.041
  3   2.500  0.600  0.978  7.491                           -1.321-16.160  -0.305  0.040
```

4	2.500	0.400	0.915	10.297	-0.855	-11.737	-0.419	0.037
5	2.500	0.200	0.862	12.098	-0.418	-6.170	-0.492	0.035
6	2.500	0.000	0.841	12.718	0.000	0.000	-0.517	0.034

HALF-SPACE

DISPLACEMENT:
v/exp(ikx) = 0.996 -0.086

STRESS:
(Szy*H1/Ghs) /exp(ikx) = -0.033 -0.149
(Sxy*H1/Ghs*)/exp(ikx) = 0.003 0.040

ROCK OUTCROP

DISPLACEMENT:
u = 1.977 -0.302 |u| = 2.000

STRESS:
(Szy*H1/Ghs) /exp(ikx) = 0.000 0.000
(Sxy*H1/Ghs*)/exp(ikx) = 0.012 0.080

OUTPUT

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING
TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (2) WITH HORIZONTAL = 75.00 degs
NUMBER OF HOMOGENEOUS REGIONS = 2

BOUNDARY VALUES

HOMOGENEOUS REGION 1
NUMBER OF BOUNDARIES = 2

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
1	2.500	0.000	0.841	12.718	12.745
2	2.250	0.000	1.291	12.601	12.667
3	2.000	0.000	1.468	12.542	12.627
4	1.750	0.000	1.641	12.483	12.590
5	1.500	0.000	1.809	12.426	12.557
6	1.250	0.000	1.971	12.374	12.530
7	1.000	0.000	2.128	12.325	12.508
8	0.750	0.000	2.278	12.282	12.492
9	0.500	0.000	2.421	12.244	12.481
10	0.250	0.000	2.557	12.212	12.477
11	0.000	0.000	2.685	12.186	12.479
12	-0.250	0.000	2.806	12.166	12.486
13	-0.500	0.000	2.919	12.152	12.498
14	-0.750	0.000	3.025	12.144	12.515
15	-1.000	0.000	3.125	12.142	12.537
16	-1.250	0.000	3.219	12.144	12.564
17	-1.500	0.000	3.308	12.152	12.594
18	-1.750	0.000	3.393	12.163	12.628
19	-2.000	0.000	3.475	12.178	12.664
20	-2.250	0.000	3.554	12.194	12.701
21	-2.500	0.000	3.391	12.286	12.745

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000
6	0.0000	0.0000
7	0.0000	0.0000
8	0.0000	0.0000
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0000	0.0000
12	0.0000	0.0000
13	0.0000	0.0000
14	0.0000	0.0000
15	0.0000	0.0000
16	0.0000	0.0000
17	0.0000	0.0000
18	0.0000	0.0000
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0000	0.0000

BOUNDARY = 2
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
22	-2.500	1.000	0.982	-0.187	0.999
23	-2.250	1.000	1.005	-0.169	1.019
24	-2.000	1.000	1.008	-0.160	1.020
25	-1.750	1.000	1.011	-0.151	1.022
26	-1.500	1.000	1.013	-0.142	1.023
27	-1.250	1.000	1.016	-0.132	1.024
28	-1.000	1.000	1.018	-0.123	1.026
29	-0.750	1.000	1.020	-0.113	1.026
30	-0.500	1.000	1.022	-0.104	1.027
31	-0.250	1.000	1.023	-0.094	1.027
32	0.000	1.000	1.024	-0.084	1.028
33	0.250	1.000	1.025	-0.073	1.028
34	0.500	1.000	1.025	-0.063	1.027
35	0.750	1.000	1.026	-0.052	1.027
36	1.000	1.000	1.025	-0.042	1.026
37	1.250	1.000	1.025	-0.031	1.026
38	1.500	1.000	1.024	-0.020	1.025
39	1.750	1.000	1.023	-0.009	1.023
40	2.000	1.000	1.022	0.002	1.022
41	2.250	1.000	1.021	0.013	1.021
42	2.500	1.000	0.999	0.016	0.999

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
22	-7.9545	-18.8072
23	-6.3943	-19.2637
24	-6.5210	-18.9492
25	-6.3578	-18.9383
26	-6.2416	-18.9102
27	-6.1071	-18.9014
28	-5.9649	-18.9029

29	-5.8131	-18.9175
30	-5.6494	-18.9398
31	-5.4742	-18.9728
32	-5.2865	-19.0153
33	-5.0858	-19.0672
34	-4.8721	-19.1284
35	-4.6458	-19.1999
36	-4.4064	-19.2787
37	-4.1551	-19.3699
38	-3.8937	-19.4707
39	-3.6084	-19.5893
40	-3.3746	-19.6940
41	-2.7699	-20.0835
42	-4.0832	-19.8422

HOMOGENEOUS REGION 2
NUMBER OF BOUNDARIES = 1

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
43	2.500	1.000	0.999	0.016	0.999
44	2.250	1.000	1.021	0.013	1.021
45	2.000	1.000	1.022	0.002	1.022
46	1.750	1.000	1.023	-0.009	1.023
47	1.500	1.000	1.024	-0.020	1.025
48	1.250	1.000	1.025	-0.031	1.026
49	1.000	1.000	1.025	-0.042	1.026
50	0.750	1.000	1.026	-0.052	1.027
51	0.500	1.000	1.025	-0.063	1.027
52	0.250	1.000	1.025	-0.073	1.028
53	0.000	1.000	1.024	-0.084	1.028
54	-0.250	1.000	1.023	-0.094	1.027
55	-0.500	1.000	1.022	-0.104	1.027
56	-0.750	1.000	1.020	-0.113	1.026
57	-1.000	1.000	1.018	-0.123	1.026
58	-1.250	1.000	1.016	-0.132	1.024
59	-1.500	1.000	1.013	-0.142	1.023
60	-1.750	1.000	1.011	-0.151	1.022
61	-2.000	1.000	1.008	-0.160	1.020
62	-2.250	1.000	1.005	-0.169	1.019
63	-2.500	1.000	0.982	-0.187	0.999

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
43	0.0309	0.1503
44	0.0210	0.1521
45	0.0256	0.1492
46	0.0273	0.1484
47	0.0295	0.1475
48	0.0315	0.1467
49	0.0334	0.1461
50	0.0352	0.1455
51	0.0369	0.1449
52	0.0385	0.1444
53	0.0400	0.1441
54	0.0415	0.1437
55	0.0428	0.1435
56	0.0440	0.1433

57	0.0452	0.1432
58	0.0463	0.1432
59	0.0473	0.1433
60	0.0482	0.1435
61	0.0494	0.1436
62	0.0484	0.1459
63	0.0603	0.1425

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING

TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (3) WITH HORIZONTAL = 60.00 degs

INCIDENT WAVE DISPLACEMENTS

HALF-SPACE	NODE	INCDT V DISPL	
	43	0.998	0.060
	44	0.999	0.041
	45	1.000	0.021
	46	1.000	0.001
	47	1.000	-0.018
	48	0.999	-0.038
	49	0.998	-0.057
	50	0.997	-0.077
	51	0.995	-0.097
	52	0.993	-0.116
	53	0.991	-0.136
	54	0.988	-0.155
	55	0.985	-0.174
	56	0.981	-0.194
	57	0.977	-0.213
	58	0.973	-0.232
	59	0.968	-0.251
	60	0.963	-0.270
	61	0.957	-0.289
	62	0.951	-0.308
	63	0.945	-0.326

FREE-FIELD CALCULATION

-X BOUNDARY

LAYER	Ash		Bsh	
1	0.943	5.949	0.943	5.949
2	1.000	0.000	-0.034	0.087

	REAL	IMAGINARY	MAGNITUDE
V(top)	1.887	11.899	12.047

LAYER = 1

NODE	X/H1	Z/H1	DISPLACEMENT V		STRESS SZY*H1/Gi		SXY*H1/Gi*	
1	-2.500	0.000	4.172	11.302	0.000	0.000	-0.888	0.328
2	-2.500	0.200	4.025	10.736	-2.028	-5.471	-0.843	0.316

```
 3  -2.500  0.400  3.591  9.092             -3.913-10.393  -0.714  0.282
 4  -2.500  0.600  2.898  6.536             -5.519-14.273  -0.513  0.228
 5  -2.500  0.800  1.988  3.319             -6.728-16.719  -0.261  0.156
 6  -2.500  1.000  0.917 -0.238             -7.448-17.484   0.019  0.072

   HALF-SPACE

DISPLACEMENT:
   v/exp(ikx) =               0.946  -0.054

STRESS:
   (Szy*H1/Ghs) /exp(ikx) =  -0.030  -0.141
   (Sxy*H1/Ghs*)/exp(ikx) =   0.004   0.074

   ROCK OUTCROP

DISPLACEMENT:
   u =                        1.982  -0.271    |u| =  2.000

STRESS:
   (Szy*H1/Ghs) /exp(ikx) =   0.000   0.000
   (Sxy*H1/Ghs*)/exp(ikx) =   0.021   0.156


   +X BOUNDARY

LAYER       Ash            Bsh

  1    0.943  5.949   0.943  5.949
  2    1.000  0.000  -0.034  0.087


              REAL          IMAGINARY      MAGNITUDE

V(top)        1.887          11.899         12.047

   LAYER = 1
                   DISPLACEMENT                              STRESS
NODE  X/H1    Z/H1         V                       SZY*H1/Gi      SXY*H1/Gi*

  1   2.500  1.000  0.938  0.131             -0.190-19.004  -0.010  0.074
  2   2.500  0.800  0.566  3.827              0.182-18.021  -0.301  0.044
  3   2.500  0.600  0.176  7.147              0.363-15.298  -0.561  0.014
  4   2.500  0.400 -0.162  9.775              0.363-11.099  -0.768 -0.013
  5   2.500  0.200 -0.390 11.459              0.220 -5.830  -0.900 -0.031
  6   2.500  0.000 -0.471 12.038              0.000  0.000  -0.945 -0.037

   HALF-SPACE

DISPLACEMENT:
   v/exp(ikx) =               0.946  -0.054

STRESS:
   (Szy*H1/Ghs) /exp(ikx) =  -0.030  -0.141
   (Sxy*H1/Ghs*)/exp(ikx) =   0.004   0.074

   ROCK OUTCROP

DISPLACEMENT:
   u =                        1.982  -0.271    |u| =  2.000

STRESS:
   (Szy*H1/Ghs) /exp(ikx) =   0.000   0.000
   (Sxy*H1/Ghs*)/exp(ikx) =   0.021   0.156
```

OUTPUT

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING
TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (3) WITH HORIZONTAL = 60.00 degs
NUMBER OF HOMOGENEOUS REGIONS = 2

BOUNDARY VALUES

HOMOGENEOUS REGION 1
NUMBER OF BOUNDARIES = 2

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
1	2.500	0.000	-0.471	12.038	12.047
2	2.250	0.000	0.082	11.957	11.957
3	2.000	0.000	0.364	11.913	11.918
4	1.750	0.000	0.642	11.866	11.883
5	1.500	0.000	0.915	11.817	11.853
6	1.250	0.000	1.182	11.769	11.828
7	1.000	0.000	1.443	11.721	11.809
8	0.750	0.000	1.697	11.674	11.797
9	0.500	0.000	1.944	11.629	11.790
10	0.250	0.000	2.183	11.586	11.790
11	0.000	0.000	2.414	11.545	11.795
12	-0.250	0.000	2.638	11.506	11.805
13	-0.500	0.000	2.854	11.470	11.820
14	-0.750	0.000	3.063	11.436	11.839
15	-1.000	0.000	3.265	11.405	11.863
16	-1.250	0.000	3.462	11.374	11.889
17	-1.500	0.000	3.653	11.345	11.919
18	-1.750	0.000	3.841	11.317	11.951
19	-2.000	0.000	4.025	11.288	11.984
20	-2.250	0.000	4.206	11.258	12.018
21	-2.500	0.000	4.172	11.302	12.047

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000
6	0.0000	0.0000
7	0.0000	0.0000
8	0.0000	0.0000
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0000	0.0000
12	0.0000	0.0000
13	0.0000	0.0000
14	0.0000	0.0000
15	0.0000	0.0000
16	0.0000	0.0000
17	0.0000	0.0000

18	0.0000	0.0000
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0000	0.0000

BOUNDARY = 2
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
22	-2.500	1.000	0.917	-0.238	0.947
23	-2.250	1.000	0.941	-0.213	0.965
24	-2.000	1.000	0.946	-0.196	0.966
25	-1.750	1.000	0.951	-0.178	0.968
26	-1.500	1.000	0.956	-0.161	0.969
27	-1.250	1.000	0.960	-0.143	0.970
28	-1.000	1.000	0.963	-0.125	0.971
29	-0.750	1.000	0.966	-0.107	0.972
30	-0.500	1.000	0.969	-0.089	0.973
31	-0.250	1.000	0.971	-0.070	0.973
32	0.000	1.000	0.972	-0.052	0.974
33	0.250	1.000	0.973	-0.033	0.974
34	0.500	1.000	0.973	-0.014	0.974
35	0.750	1.000	0.973	0.005	0.973
36	1.000	1.000	0.972	0.024	0.973
37	1.250	1.000	0.971	0.043	0.972
38	1.500	1.000	0.969	0.062	0.971
39	1.750	1.000	0.967	0.082	0.970
40	2.000	1.000	0.964	0.101	0.969
41	2.250	1.000	0.961	0.120	0.968
42	2.500	1.000	0.938	0.131	0.947

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
22	-8.9477	-17.1458
23	-7.3841	-17.7001
24	-7.3077	-17.4824
25	-6.9897	-17.5485
26	-6.7120	-17.5924
27	-6.4178	-17.6497
28	-6.1156	-17.7118
29	-5.8039	-17.7812
30	-5.4800	-17.8530
31	-5.1443	-17.9298
32	-4.7957	-18.0103
33	-4.4338	-18.0944
34	-4.0583	-18.1818
35	-3.6696	-18.2735
36	-3.2671	-18.3664
37	-2.8517	-18.4653
38	-2.4249	-18.5674
39	-1.9727	-18.6799
40	-1.5705	-18.7761
41	-0.7813	-19.1198
42	-1.8992	-18.9502

HOMOGENEOUS REGION 2
NUMBER OF BOUNDARIES = 1

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
43	2.500	1.000	0.938	0.131	0.947
44	2.250	1.000	0.961	0.120	0.968
45	2.000	1.000	0.964	0.101	0.969
46	1.750	1.000	0.967	0.082	0.970
47	1.500	1.000	0.969	0.062	0.971
48	1.250	1.000	0.971	0.043	0.972
49	1.000	1.000	0.972	0.024	0.973
50	0.750	1.000	0.973	0.005	0.973
51	0.500	1.000	0.973	-0.014	0.974
52	0.250	1.000	0.973	-0.033	0.974
53	0.000	1.000	0.972	-0.052	0.974
54	-0.250	1.000	0.971	-0.070	0.973
55	-0.500	1.000	0.969	-0.089	0.973
56	-0.750	1.000	0.966	-0.107	0.972
57	-1.000	1.000	0.963	-0.125	0.971
58	-1.250	1.000	0.960	-0.143	0.970
59	-1.500	1.000	0.956	-0.161	0.969
60	-1.750	1.000	0.951	-0.178	0.968
61	-2.000	1.000	0.946	-0.196	0.966
62	-2.250	1.000	0.941	-0.213	0.965
63	-2.500	1.000	0.917	-0.238	0.947

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
43	0.0144	0.1436
44	0.0059	0.1448
45	0.0119	0.1422
46	0.0149	0.1415
47	0.0184	0.1407
48	0.0216	0.1399
49	0.0248	0.1391
50	0.0278	0.1384
51	0.0307	0.1377
52	0.0336	0.1371
53	0.0363	0.1364
54	0.0390	0.1358
55	0.0415	0.1352
56	0.0440	0.1347
57	0.0463	0.1342
58	0.0486	0.1337
59	0.0508	0.1333
60	0.0530	0.1329
61	0.0554	0.1324
62	0.0559	0.1341
63	0.0678	0.1299

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING

TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (4) WITH HORIZONTAL = 45.00 degs

INCIDENT WAVE DISPLACEMENTS

HALF-SPACE	NODE	INCDT V DISPL	
	43	0.986	0.166

44	0.990	0.138
45	0.994	0.111
46	0.997	0.083
47	0.998	0.056
48	1.000	0.028
49	1.000	0.000
50	1.000	-0.028
51	0.998	-0.056
52	0.997	-0.083
53	0.994	-0.111
54	0.990	-0.138
55	0.986	-0.166
56	0.981	-0.193
57	0.975	-0.220
58	0.969	-0.247
59	0.962	-0.274
60	0.954	-0.301
61	0.945	-0.327
62	0.936	-0.353
63	0.925	-0.379

FREE-FIELD CALCULATION

-X BOUNDARY

LAYER	Ash		Bsh	
1	0.728	5.335	0.728	5.335
2	1.000	0.000	-0.132	0.117

	REAL	IMAGINARY	MAGNITUDE
V(top)	1.457	10.670	10.769

LAYER = 1

			DISPLACEMENT		STRESS			
NODE	X/H1	Z/H1	V		SZY*H1/Gi		SXY*H1/Gi*	
1	-2.500	0.000	4.326	9.862	0.000	0.000	-1.095	0.480
2	-2.500	0.200	4.165	9.366	-2.095	-4.762	-1.040	0.463
3	-2.500	0.400	3.692	7.926	-4.034	-9.045	-0.880	0.410
4	-2.500	0.600	2.939	5.686	-5.671	-12.416	-0.632	0.326
5	-2.500	0.800	1.958	2.870	-6.880	-14.536	-0.319	0.217
6	-2.500	1.000	0.815	-0.242	-7.565	-15.187	0.027	0.091

HALF-SPACE

DISPLACEMENT:
v/exp(ikx) = 0.850 -0.009

STRESS:
(Szy*H1/Ghs) /exp(ikx) = -0.024 -0.126
(Sxy*H1/Ghs*)/exp(ikx) = 0.001 0.094

ROCK OUTCROP

DISPLACEMENT:
u = 1.988 -0.222 |u| = 2.000

STRESS:
(Szy*H1/Ghs) /exp(ikx) = 0.000 0.000

(Sxy*H1/Ghs*)/exp(ikx) = 0.025 0.221

+X BOUNDARY

LAYER	Ash		Bsh	
1	0.728	5.335	0.728	5.335
2	1.000	0.000	-0.132	0.117

	REAL	IMAGINARY	MAGNITUDE
V(top)	1.457	10.670	10.769

LAYER = 1

NODE	X/H1	Z/H1	DISPLACEMENT V		STRESS SZY*H1/Gi		SXY*H1/Gi*	
1	2.500	1.000	0.820	0.224	1.579	-16.894	-0.025	0.091
2	2.500	0.800	0.150	3.471	1.818	-15.979	-0.386	0.017
3	2.500	0.600	-0.501	6.381	1.728	-13.540	-0.709	-0.056
4	2.500	0.400	-1.042	8.681	1.341	-9.812	-0.964	-0.116
5	2.500	0.200	-1.400	10.154	0.731	-5.151	-1.128	-0.155
6	2.500	0.000	-1.524	10.661	0.000	0.000	-1.184	-0.169

HALF-SPACE

DISPLACEMENT:
v/exp(ikx) = 0.850 -0.009

STRESS:
(Szy*H1/Ghs) /exp(ikx) = -0.024 -0.126
(Sxy*H1/Ghs*)/exp(ikx) = 0.001 0.094

ROCK OUTCROP

DISPLACEMENT:
u = 1.988 -0.222 |u| = 2.000

STRESS:
(Szy*H1/Ghs) /exp(ikx) = 0.000 0.000
(Sxy*H1/Ghs*)/exp(ikx) = 0.025 0.221

OUTPUT

1 LAYER ON HALF-SPACE; HETEROGENEOUS PROFILE with DAMPING
TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 1.571
INCIDENT ANGLE (4) WITH HORIZONTAL = 45.00 degs
NUMBER OF HOMOGENEOUS REGIONS = 2

BOUNDARY VALUES

HOMOGENEOUS REGION 1
NUMBER OF BOUNDARIES = 2

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
1	2.500	0.000	-1.524	10.661	10.769
2	2.250	0.000	-0.935	10.634	10.675
3	2.000	0.000	-0.596	10.623	10.640
4	1.750	0.000	-0.260	10.605	10.608
5	1.500	0.000	0.071	10.582	10.582
6	1.250	0.000	0.397	10.553	10.561
7	1.000	0.000	0.717	10.521	10.546
8	0.750	0.000	1.031	10.486	10.537
9	0.500	0.000	1.338	10.448	10.533
10	0.250	0.000	1.638	10.408	10.536
11	0.000	0.000	1.930	10.365	10.543
12	-0.250	0.000	2.215	10.320	10.555
13	-0.500	0.000	2.492	10.273	10.571
14	-0.750	0.000	2.763	10.224	10.590
15	-1.000	0.000	3.027	10.172	10.613
16	-1.250	0.000	3.284	10.118	10.638
17	-1.500	0.000	3.537	10.061	10.665
18	-1.750	0.000	3.785	10.001	10.694
19	-2.000	0.000	4.029	9.937	10.723
20	-2.250	0.000	4.269	9.868	10.751
21	-2.500	0.000	4.326	9.862	10.769

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000
6	0.0000	0.0000
7	0.0000	0.0000
8	0.0000	0.0000
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0000	0.0000
12	0.0000	0.0000
13	0.0000	0.0000
14	0.0000	0.0000
15	0.0000	0.0000
16	0.0000	0.0000
17	0.0000	0.0000
18	0.0000	0.0000
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0000	0.0000

BOUNDARY = 2
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
22	-2.500	1.000	0.815	-0.242	0.850
23	-2.250	1.000	0.838	-0.213	0.865
24	-2.000	1.000	0.845	-0.191	0.867
25	-1.750	1.000	0.851	-0.169	0.868
26	-1.500	1.000	0.857	-0.146	0.869
27	-1.250	1.000	0.862	-0.123	0.870
28	-1.000	1.000	0.866	-0.100	0.871
29	-0.750	1.000	0.869	-0.077	0.872
30	-0.500	1.000	0.871	-0.054	0.873
31	-0.250	1.000	0.873	-0.030	0.873

32	0.000	1.000	0.874	-0.007	0.874
33	0.250	1.000	0.874	0.017	0.874
34	0.500	1.000	0.873	0.041	0.874
35	0.750	1.000	0.871	0.065	0.873
36	1.000	1.000	0.869	0.089	0.873
37	1.250	1.000	0.865	0.113	0.873
38	1.500	1.000	0.861	0.137	0.872
39	1.750	1.000	0.856	0.161	0.871
40	2.000	1.000	0.850	0.185	0.870
41	2.250	1.000	0.844	0.208	0.869
42	2.500	1.000	0.820	0.224	0.850

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
22	-8.8593	-14.8621
23	-7.3988	-15.4501
24	-7.1892	-15.3315
25	-6.7840	-15.4581
26	-6.4108	-15.5586
27	-6.0220	-15.6651
28	-5.6247	-15.7696
29	-5.2174	-15.8746
30	-4.7979	-15.9754
31	-4.3665	-16.0743
32	-3.9223	-16.1700
33	-3.4650	-16.2623
34	-2.9944	-16.3509
35	-2.5110	-16.4365
36	-2.0143	-16.5161
37	-1.5049	-16.5942
38	-0.9845	-16.6681
39	-0.4395	-16.7433
40	0.0589	-16.7997
41	0.9316	-17.0514
42	0.0292	-16.9316

HOMOGENEOUS REGION 2
NUMBER OF BOUNDARIES = 1

BOUNDARY = 1
NUMBER OF NODES = 21

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
43	2.500	1.000	0.820	0.224	0.850
44	2.250	1.000	0.844	0.208	0.869
45	2.000	1.000	0.850	0.185	0.870
46	1.750	1.000	0.856	0.161	0.871
47	1.500	1.000	0.861	0.137	0.872
48	1.250	1.000	0.865	0.113	0.873
49	1.000	1.000	0.869	0.089	0.873
50	0.750	1.000	0.871	0.065	0.873
51	0.500	1.000	0.873	0.041	0.874
52	0.250	1.000	0.874	0.017	0.874
53	0.000	1.000	0.874	-0.007	0.874
54	-0.250	1.000	0.873	-0.030	0.873
55	-0.500	1.000	0.871	-0.054	0.873
56	-0.750	1.000	0.869	-0.077	0.872
57	-1.000	1.000	0.866	-0.100	0.871
58	-1.250	1.000	0.862	-0.123	0.870
59	-1.500	1.000	0.857	-0.146	0.869

60	-1.750	1.000	0.851	-0.169	0.868
61	-2.000	1.000	0.845	-0.191	0.867
62	-2.250	1.000	0.838	-0.213	0.865
63	-2.500	1.000	0.815	-0.242	0.850

TRACTIONS

NODE	TAU(n,y)*H1/Gi	
43	-0.0002	0.1283
44	-0.0071	0.1292
45	-0.0004	0.1273
46	0.0033	0.1268
47	0.0075	0.1263
48	0.0114	0.1257
49	0.0153	0.1251
50	0.0190	0.1245
51	0.0227	0.1239
52	0.0262	0.1232
53	0.0297	0.1225
54	0.0331	0.1218
55	0.0363	0.1210
56	0.0395	0.1203
57	0.0426	0.1195
58	0.0456	0.1187
59	0.0486	0.1179
60	0.0514	0.1171
61	0.0545	0.1161
62	0.0561	0.1170
63	0.0671	0.1126

SAMPLE OUTPUT FILE FOR A CYLINCRICAL CAVITY IN A FULL-SPACE

```
cylindrical cavity IN FULL SPACE

 TYPE OF WAVE = SH
 DIMENSIONLESS FREQUENCY (omega*H1/Cs1) =  3.0000

 ADMITTANCE FUNCTIONS ARE CALCULATED FOR 1 ANGLES
 INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs
 NUMBER OF HOMOGENEOUS SUBREGIONS =  1


 HOMOGENEOUS SOIL REGION : 1

 Cs(area 1)/Cs(area 1) =    1.000
 UNIT WT(area 1)/UNIT WT(area 1) = 1.000
 POISSIONS RATIO 0.330
 DAMPING  0.00 %


 NUMBER OF BOUNDARIES  1

 NO. OF BOUNDARY ELEMENTS FOR BOUNDARY 1= 20

 BOUNDARY NODE COORDINATES       X/H(layer1)         Z/H(layer1)

 BOUNDARY =  1
                   1              1.0000          0.0000
                   2              0.9511          0.3090
                   3              0.8090          0.5878
                   4              0.5878          0.8090
                   5              0.3090          0.9511
                   6              0.0000          1.0000
                   7             -0.3090          0.9511
                   8             -0.5878          0.8090
                   9             -0.8090          0.5878
                  10             -0.9511          0.3090
                  11             -1.0000          0.0000
                  12             -0.9511         -0.3090
                  13             -0.8090         -0.5878
                  14             -0.5878         -0.8090
                  15             -0.3090         -0.9511
                  16              0.0000         -1.0000
                  17              0.3090         -0.9511
                  18              0.5878         -0.8090
                  19              0.8090         -0.5878
                  20              0.9511         -0.3090


    BOUNDARY 1 IS CLOSED

    INTERFACE NODES


    3 NODE INTERFACES

         NONE

    SURFACE NODES

    AREA 1: NODE   1 THROUGH NODE  21

 SCATTERING BOUNDARY OCCURS AT X/H1 =    0.00
```

cylindrical cavity IN FULL SPACE

TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 3.000
INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs

INCIDENT WAVE DISPLACEMENTS

HALF-SPACE	NODE	INCDT V DISPL	
	1	1.000	0.000
	2	0.600	-0.800
	3	-0.191	-0.982
	4	-0.755	-0.655
	5	-0.959	-0.284
	6	-0.990	-0.141
	7	-0.959	-0.284
	8	-0.755	-0.655
	9	-0.191	-0.982
	10	0.600	-0.800
	11	1.000	0.000
	12	0.600	0.800
	13	-0.191	0.982
	14	-0.755	0.655
	15	-0.959	0.284
	16	-0.990	0.141
	17	-0.959	0.284
	18	-0.755	0.655
	19	-0.191	0.982
	20	0.600	0.800

OUTPUT

CYLINDRICAL CAVITY IN FULL SPACE
TYPE OF WAVE = SH
DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = 3.000
INCIDENT ANGLE (1) WITH HORIZONTAL = 90.00 degs
NUMBER OF HOMOGENEOUS REGIONS = 1

BOUNDARY VALUES

HOMOGENEOUS REGION 1
NUMBER OF BOUNDARIES = 1

BOUNDARY = 1
NUMBER OF NODES = 20

DISPLACEMENTS

NODE	X/H1	Z/H1	V DISPL		|V|
1	1.000	0.000	1.308	-0.196	1.322
2	0.951	0.309	0.767	-1.386	1.584
3	0.809	0.588	-0.540	-1.610	1.698
4	0.588	0.809	-1.514	-0.997	1.813
5	0.309	0.951	-1.868	-0.335	1.898
6	0.000	1.000	-1.921	-0.082	1.923
7	-0.309	0.951	-1.868	-0.335	1.898
8	-0.588	0.809	-1.514	-0.997	1.813

9	-0.809	0.588	-0.540	-1.610	1.698
10	-0.951	0.309	0.767	-1.386	1.584
11	-1.000	0.000	1.308	-0.196	1.322
12	-0.951	-0.309	0.558	0.853	1.019
13	-0.809	-0.588	-0.446	0.828	0.940
14	-0.588	-0.809	-0.564	0.155	0.585
15	-0.309	-0.951	0.016	-0.357	0.358
16	0.000	-1.000	0.364	-0.506	0.623
17	0.309	-0.951	0.016	-0.357	0.358
18	0.588	-0.809	-0.564	0.155	0.585
19	0.809	-0.588	-0.446	0.828	0.940
20	0.951	-0.309	0.558	0.853	1.019

A.1.1: BOUNDARY ELEMENT CODE

shconvo

```
      PROGRAM SHMAIN
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C
C     OUTPUT DATA
      COMMON/COUT/VDISPL(720),SNYH1(720,2)
      COMPLEX VDISPL,SNYH1
C
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
      COMPLEX XIKTZ,XIKX
      CHARACTER*80 ANAME
C
```

```
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
      CHARACTER*20 ARUN
C
      DATA NGAUSS /5/
      DATA (WT(I),I=1,3)/0.236926885056189D0,0.478628670499366D0,
     & 0.568888888888889D0/
      DATA (XX(I),I=1,3)/-0.906179845938664D0,-0.538469310105683D0,
     & 0.0D0/
C
      DATA NGAUS3 /3/
      DATA (WT3(I),I=1,2)/0.555555555555556D0,0.888888888888888889D0/
      DATA (XX3(I),I=1,3)
     & /-0.77459666924148D0,0.0D0,+0.77459666924148D0/
C
C
      XX(4)= -1.0D0*XX(2)
      XX(5)= -1.0D0*XX(1)
      DO IGAUSS= 1,2
      WTFN(IGAUSS)= (1.0D0 - XX(IGAUSS))/2.0D0
      WTFN(NGAUSS-IGAUSS+1)= 1.0D0-WTFN(IGAUSS)
      END DO
      WTFN(3)= 0.50D0
C
      WTFN3(1)= (1.0D0 - XX3(1))/2.0D0
      WTFN3(3)= 1.0D0-WTFN3(1)
      WTFN3(2)= 0.50D0
C
C     OPEN(UNIT=10,FILE='CHECK.SHBEM',STATUS='UNKNOWN')
C
C
      WRITE(*,9000)
 9000 FORMAT(5X,'INPUT NAME OF RUNFILE')
      READ(*,9010)ARUN
 9010 FORMAT(A20)
      OPEN(UNIT=15,FILE=ARUN,STATUS='UNKNOWN')
      READ(15,*)NPROB
      DO 9999 IPROB=1,NPROB
      WRITE(*,9020)IPROB
 9020 FORMAT(/5X,'WORKING ON PROBLEM ',I2)
C
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON INPUT'
      WRITE(*,*)' '
      CALL INPUT(ANAME)
      DO 100 IANGLE=1,NANGLE
      WRITE(60,6000) ANAME
```

```
      WRITE(*,6000) ANAME
 6000 FORMAT(//2X,A80)
      WRITE(60,6010)ATYPE
      WRITE(*,6010)ATYPE
 6010 FORMAT(/2X,'TYPE OF WAVE = ',A2)
      X1=FBAR*CSQRT(1.0-(0.0,2.0)*BETA(1))
      WRITE(60,6020)X1
      WRITE(*,6020)X1
 6020 FORMAT(2X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F6.3)
      WRITE(60,6030)IANGLE,ANGLE(IANGLE)
      WRITE(*,6030)IANGLE,ANGLE(IANGLE)
 6030 FORMAT(2X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
C
C     CALC. INCIDENT WAVE DISPLACEMENTS
C     IF(ATYPE .EQ. 'SH')THEN
      XMX=COS(ANGLE(IANGLE)/57.29577951)
      ASH=1.0
C     END IF
      DO 10 INODE =JNODE(HALFSP,1),KNODE(HALFSP,NBOUND(HALFSP))
      IINODE=INODE-JNODE(HALFSP,1)+1
      XIKTZ=(0.0,1.0)*FBAR/CSRAT(HALFSP)*Z(INODE)*SQRT(1.-XMX*XMX)
      XIKX= (0.0,1.0)*FBAR*X(INODE)*XMX/CSRAT(HALFSP)
      VINCDT(IINODE)=ASH*CEXP(-XIKTZ)*CEXP(XIKX)
   10 CONTINUE
      WRITE(60,6040)
 6040 FORMAT(///2X,'INCIDENT WAVE DISPLACEMENTS ')
      WRITE(60,6050)
 6050 FORMAT(T5,'HALF-SPACE',T18,'NODE',T34,
     & 'INCDT V DISPL'/)
      DO 20INODE= JNODE(HALFSP,1),KNODE(HALFSP,NBOUND(HALFSP))
      IINODE=INODE-JNODE(HALFSP,1)+1
      WRITE(60,6060)INODE,VINCDT(IINODE)
 6060 FORMAT(T18,I3,T33,2F7.3)
   20 CONTINUE
C
      IF(NAREA .EQ. 1 .AND. ICLOSE(1,1) .EQ. 1)GO TO 40
C     CAVITY IN FULL SPACE
      IF(NNSURF .NE. 0)THEN
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON FREE-FIELD MOTION'
      DO 30IX=1,2
      CALL FFLD(IX)
   30 CONTINUE
      END IF
   40 CONTINUE
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON MATRIX'
      WRITE(*,*)' '
      CALL MATRIX
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON SOLVING MATRIX'
      WRITE(*,*)' '
      CALL SOLVE(2)
      WRITE(*,*)' '
      WRITE(*,*)'SORTING MATRICES'
      WRITE(*,*)' '
      CALL SORT
```

```
      WRITE(*,*)' '
      WRITE(*,*)'OUTPUT'
      CALL OUTPUT(ANAME)
  100 CONTINUE
      CLOSE(UNIT=60)
 9999 CONTINUE
      CLOSE(UNIT=15)
C     CLOSE(UNIT=10)
      STOP
      END
```

```
      SUBROUTINE INPUT(ANAME)
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
      INTEGER NELEM(4,2),NSURF1(3,3),NSURF2(3,3),NBND(3)
      REAL XENDPT(50),ZENDPT(50)
      CHARACTER*25 AINPT,AOUT
      CHARACTER*80 ANAME
C
C
C
      789112345678921234567893123456789412345678951234567896123456789712

      PI=4.*ATAN(1.0)
C
      READ(15,1500)AINPT
 1500 FORMAT(A25)
      WRITE(*,9000)AINPT
 9000 FORMAT(/2X,'WORKING ON INPUT FILE: ',A25/)
C
C     WRITE(*,9000)
```

```
C 9000 FORMAT(5X,'NAME OF INPUT FILE')
C      READ(*,9010)AINPT
C 9010 FORMAT(A20)
       OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
       READ(50,5000) AOUT
 5000  FORMAT(A25)
       READ(50,5010) ANAME
 5010  FORMAT(A80)
C      TYPE OF INCIDENT WAVE (SH)
       READ(50,5020)ATYPE
 5020  FORMAT(A2)
C      ANGLE OF INCIDENCE WITH HORIZONTAL (degrees)')
       READ(50,*)NANGLE,(ANGLE(I),I=1,NANGLE)
C      DIMENSIONLESS FREQUENCY (omega*H1/Cs1)')
       READ(50,*)X1
       FBAR=(1.0,0.0)*X1
C      # OF HOMOGENEOUS REGIONS
       READ(50,*)NAREA
C
C      INITIALIZE JSTART
       JSTART=0
C
       DO 50IAREA=1,NAREA
C      INPUT SOIL SHEAR VELOCITY RATIO Cs (area i)/Cs(area 1)
       READ(50,*)X1
       CSRAT(IAREA)=X1*(1.00,0.0)
C      INPUT POISSON'S RATIO
       READ(50,*)POISS(IAREA)
C      INPUT UNIT WEIGHT RATIO WT(layer i)/WT(layer 1)
       READ(50,*)UWTRAT(IAREA)
C      INPUT DAMPING (%)
       READ(50,*)BETA(IAREA)
       BETA(IAREA)=BETA(IAREA)/100.0
C      INPUT GEOMETRY DATA
       READ(50,*)NBOUND(IAREA)
C
C
       DO 40 IBOUND=1,NBOUND(IAREA)
       NELEM(IAREA,IBOUND)=0
       READ(50,*)NENDPT
C      STARTING NODE NUMBER
       JSTART=JSTART + 1
       JNODE(IAREA,IBOUND)=JSTART
       JNO= JNODE(IAREA,IBOUND)
       DO 30IENDPT=1,NENDPT-1
C      INPUT END POINTS THAT DEFINE BOUNDARY
       READ(50,*)XENDPT(IENDPT),ZENDPT(IENDPT),
     &  XENDPT(IENDPT+1),ZENDPT(IENDPT+1)
C      INPUT NUMBER OF SEGMENTS TO DIVIDE EACH PORTION OF BOUNDARY INTO
       DIST = SQRT((XENDPT(IENDPT+1)-XENDPT(IENDPT))**2+
     & (ZENDPT(IENDPT+1)-ZENDPT(IENDPT))**2)
C
C      CHECK IF CIRCULAR BOUNDARY
       IF(DIST .LT. 0.00001)THEN
C      CIRCULAR BOUNDARY
       ICLOSE(IAREA,IBOUND)=1
C      INPUT CENTER OF CIRCLE
       READ(50,*)X0,Z0
```

```
C     INPUT DIRECTION OF ROTATION +1=CCW, -1= CW AND NUMBER OF SEGMENTS
      READ(50,*)IROTAT,NSEG
      X1=XENDPT(1)
      Z1=ZENDPT(1)
      R=SQRT((X1-X0)**2+(Z1-Z0)**2)
C     DETERMINE ALPHA, ANGLE FOR Z1 & Z1
C
      IF(X1 .EQ. 0)THEN
      IF(Z1 .GE. 0)ALPHA=PI/2.
      IF(Z1 .LT. 0)ALPHA=3.*PI/2.
C
      ELSE
C
      ALPHA=ATAN((Z1-Z0)/(X1-X0))
      ALPHA=ABS(ALPHA)
      IF(X1 .LT. 0. .AND. Z1 .GE. 0)ALPHA=PI-ALPHA
      IF(X1 .LT. 0. .AND. Z1 .LT. 0)ALPHA=PI+ALPHA
      IF(X1 .GE. 0. .AND. Z1 .LT. 0)ALPHA=2.*PI-ALPHA
      END IF
C
      X(JNO)=X1
      Z(JNO)=Z1
      DTHETA=2.*PI/NSEG*IROTAT
      DO 10ISEG=1,NSEG-1
      INODE=JNO+ISEG
      THETA=ISEG*DTHETA
      X(INODE)=X0+R*COS(THETA+ALPHA)
      Z(INODE)=Z0+R*SIN(THETA+ALPHA)
   10 CONTINUE
      KNODE(IAREA,IBOUND)=JNO+NSEG-1
      JSTART=KNODE(IAREA,IBOUND)
      NELEM(IAREA,IBOUND)= NSEG
      GO TO 40
      END IF
C
C
      READ(50,*)NSEG
      ZDELTA=ZENDPT(IENDPT+1)-ZENDPT(IENDPT)
      XDELTA=XENDPT(IENDPT+1)-XENDPT(IENDPT)
      JSTOP=JSTART+NSEG
      DO 20INODE=JSTART,JSTOP
      X(INODE)=XENDPT(IENDPT)+XDELTA/NSEG*(INODE-JSTART)
      Z(INODE)=ZENDPT(IENDPT)+ZDELTA/NSEG*(INODE-JSTART)
   20 CONTINUE
      JSTART=JSTOP
      NELEM(IAREA,IBOUND)=NELEM(IAREA,IBOUND)+NSEG
   30 CONTINUE
      DIST=SQRT((X(JSTOP)-X(JNO))**2+(Z(JSTOP)-Z(JNO))**2)
      ICLOSE(IAREA,IBOUND)=0
      IF(DIST .LT. 0.00001) THEN
      ICLOSE(IAREA,IBOUND)=1
      JSTOP=JSTOP-1
      JSTART=JSTOP
      END IF
      KNODE(IAREA,IBOUND)=JSTOP
   40 CONTINUE
   50 CONTINUE
      DO 60INODE=1,KNODE(NAREA,NBOUND(NAREA))
```

```
      NSURF(INODE)=0
   60 CONTINUE
      NNSURF=0
C     NUMBER OF AREAS BORDERING SURFACE
      READ(50,*)NSURFA
      IF(NSURFA .NE. 0)THEN
      READ(50,*) (NBND(I),I=1,NSURFA)
      DO 67 ISURFA=1,NSURFA
C     SURFACE NODES
      DO 67IBND=1,NBND(ISURFA)
      READ(50,*)NSURF1(ISURFA,IBND), NSURF2(ISURFA,IBND)
      DO 65 INODE=NSURF1(ISURFA,IBND),NSURF2(ISURFA,IBND)
      NSURF(INODE)=1
   65 CONTINUE
      NNSURF= NNSURF+NSURF2(ISURFA,IBND)-NSURF1(ISURFA,IBND)+1
   67 CONTINUE
      END IF
C     HALF-SPACE REGION
      READ(50,*)HALFSP
C
      NLAYER(1)=0
      NLAYER(2)=0
C
      IF(NAREA .GT. 1 .AND. NNSURF .NE. 0)THEN
C     INPUT DESCRIPTION OF FREE-FIELD SOIL COLUMN
C     1: -X AREA   ,  2: +X AREA
      DO 90I=1,2
      READ(50,*)NLAYER(I)
      DO 80ILAYER=1,NLAYER(I)
      READ(50,*)NTOP(I,ILAYER),NBOTT(I,ILAYER)
      DO 70 IAREA=1,NAREA
      MARK1=0
      MARK2=0
      DO 70 INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF(NTOP(I,ILAYER) .EQ. INODE)MARK1=1
      IF(NBOTT(I,ILAYER) .EQ. INODE) MARK2=1
      IF(MARK1 .EQ. 1 .AND. MARK2 .EQ. 1)LTYPE(I,ILAYER) = IAREA
   70 CONTINUE
   80 CONTINUE
      LTYPE(I,NLAYER(I)+1)=HALFSP
   90 CONTINUE
      END IF
      CLOSE(UNIT=50)
      OPEN(UNIT=60,FILE=AOUT,STATUS='UNKNOWN')
      WRITE(60,6000) ANAME
 6000 FORMAT(2X,A80)
      WRITE(60,6010)ATYPE
 6010 FORMAT(/2X,'TYPE OF WAVE = ',A2)
      X1=REAL(FBAR)
      WRITE(60,6020)X1
 6020 FORMAT(2X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F7.4)
      WRITE(60,6025)NANGLE
 6025 FORMAT(
     & /2X,'ADMITTANCE FUNCTIONS ARE CALCULATED FOR ',I1,' ANGLES')
      DO 100 IANGLE=1,NANGLE
      WRITE(60,6030)IANGLE,ANGLE(IANGLE)
 6030 FORMAT(2X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
```

```
  100 CONTINUE
      WRITE(60,6040)NAREA
 6040 FORMAT(2X,'NUMBER OF HOMOGENEOUS SUBREGIONS = ',I2/)
      IELEM=0
      DO 190IAREA=1,NAREA
      X1=REAL(CSRAT(IAREA))
      WRITE(60,6050)IAREA
 6050 FORMAT(/2X,'HOMOGENEOUS SOIL REGION : ',I1/)
      WRITE(60,6060)IAREA,X1
 6060 FORMAT(2X,'Cs(area ',I1,')/Cs(area 1) = ',F8.3)
      WRITE(60,6070)IAREA,UWTRAT(IAREA)
 6070 FORMAT(2X,'UNIT WT(area ',I1,')/UNIT WT(area 1) = ',F5.3)
      WRITE(60,6080)POISS(IAREA)
 6080 FORMAT(2X,'POISSIONS RATIO ',F5.3)
      WRITE(60,6090)BETA(IAREA)*100.0
 6090 FORMAT(2X,'DAMPING ',F5.2,' %')
      WRITE(60,6100) NBOUND(IAREA)
 6100 FORMAT(//2X,'NUMBER OF BOUNDARIES ',I2/)
      DO 120 IBOUND=1,NBOUND(IAREA)
      WRITE(60,6120)IBOUND,NELEM(IAREA,IBOUND)
 6120 FORMAT(2X,'NO. OF BOUNDARY ELEMENTS FOR BOUNDARY',I2,'=',I3)
  120 CONTINUE
      WRITE(60,6130)
 6130 FORMAT(/2X,'BOUNDARY NODE COORDINATES', T35,'X/H(layer1)',T55,
     & 'Z/H(layer1)')
      DO 140 IBOUND=1,NBOUND(IAREA)
      WRITE(60,6140)IBOUND
 6140 FORMAT(/2X,'BOUNDARY = ',I2)
      DO 130 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      WRITE(60,6150)INODE,X(INODE),Z(INODE)
 6150 FORMAT(T20,I3,T36,F8.4,T51,F8.4)
  130 CONTINUE
  140 CONTINUE
C     NODE CONNECTIVITY
C     WRITE(60,6160)
 6160 FORMAT (//2X,'NODE CONNECTIVITY')
C     WRITE(60,6170)
 6170 FORMAT(/T6,'BOUNDARY',T17,'ELEMENT',T27,'NODE 1',T37,'NODE 2')
      DO 160 IBOUND=1,NBOUND(IAREA)
      IELEM=IELEM+1
C     WRITE(60,6180)IBOUND,IELEM,
C     & JNODE(IAREA,IBOUND),JNODE(IAREA,IBOUND)+1
 6180 FORMAT(T9,I2,T19,I3,T28,I3,T38,I3)
      DO 150 INODE=JNODE(IAREA,IBOUND)+1,KNODE(IAREA,IBOUND)-1
      IELEM=IELEM+1
C     WRITE(60,6190)IELEM,INODE,INODE+1
 6190 FORMAT(T19,I3,T28,I3,T38,I3)
  150 CONTINUE
      IF(ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
      IELEM=IELEM+1
C     WRITE(60,6190)IELEM,KNODE(IAREA,IBOUND),JNODE(IAREA,IBOUND)
      END IF
  160 CONTINUE
      WRITE(60,6230)
 6230 FORMAT(/)
      DO 6260IBOUND=1,NBOUND(IAREA)
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)WRITE(60,6240)IBOUND
      IF(ICLOSE(IAREA,IBOUND) .EQ. 1) WRITE(60,6250)IBOUND
```

```
6240 FORMAT(5X,'BOUNDARY',I2,' IS OPEN')
6250 FORMAT(5X,'BOUNDARY',I2,' IS CLOSED')
6260 CONTINUE
 190 CONTINUE
     DO 210 INODE=1,KNODE(NAREA,NBOUND(NAREA))
     IEQN(INODE)=1
     NCONN(INODE)=0
 210 CONTINUE
     DO 220 IAREA=1,NAREA-1
     DO 220 IBOUND=1,NBOUND(IAREA)
     DO 220 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
     IF(NCONN(INODE) .NE. 0 ) GO TO 220
     DO 217 IIAREA=IAREA+1,NAREA
     DO 215IINODE=JNODE(IIAREA,1),KNODE(NAREA,NBOUND(NAREA))
     IF(IEQN(IINODE) .EQ. 0) GO TO 215
     DIST=SQRT((X(INODE)-X(IINODE))**2+(Z(INODE)-Z(IINODE))**2)
     IF(DIST .LT. 0.00001) THEN
     NCONN(IINODE)=INODE
     IEQN(INODE)=IEQN(INODE)+1
     IEQN(IINODE)=0
     GO TO 217
     END IF
 215 CONTINUE
 217 CONTINUE
 220 CONTINUE
     WRITE(60,6270)
6270 FORMAT(/5X,'INTERFACE NODES'/)
     DO 240 INODE=1,KNODE(NAREA,NBOUND(NAREA))
     IF(IEQN(INODE) .EQ. 2)THEN
     DO 230 IINODE =INODE+1,KNODE(NAREA,NBOUND(NAREA))
     IF(NCONN(IINODE) .EQ. INODE) WRITE(60,6280)INODE,IINODE
6280 FORMAT(T9,I3,' - ',I3)
 230 CONTINUE
     END IF
 240 CONTINUE
     WRITE(60,6290)
6290 FORMAT(/5X,'3 NODE INTERFACES'/)
     ICOUNT=0
     DO 252 INODE=1,KNODE(NAREA,NBOUND(NAREA))-1
     IMARK=0
     IF(IEQN(INODE) .EQ. 3)THEN
     ICOUNT=ICOUNT+1
     DO 250 IINODE=INODE+1,KNODE(NAREA,NBOUND(NAREA))
     IF(NCONN(IINODE) .EQ. INODE .AND. IMARK .EQ. 0) THEN
     MARK1=IINODE
     IMARK=1
     END IF
     IF(NCONN(IINODE) .EQ. INODE .AND. IMARK .EQ. 1) THEN
     MARK2=IINODE
     END IF
 250 CONTINUE
     N3(ICOUNT,1)=INODE
     N3(ICOUNT,2)=MARK1
     N3(ICOUNT,3)=MARK2
     WRITE(60,6300)N3(ICOUNT,1),N3(ICOUNT,2),N3(ICOUNT,3)
6300 FORMAT(T9,I3,' - ',I3,' - ', I3)
     END IF
 252 CONTINUE
```

```
      NN3=ICOUNT
      IF(NN3 .EQ. 0) WRITE(60,6305)
 6305 FORMAT(T11,'NONE')
      WRITE(60,6307)
 6307 FORMAT(/5X,'SURFACE NODES')
      IF (NSURFA .EQ. 0)THEN
      WRITE(60,6305)
      ELSE
      DO 253 ISURFA=1,NSURFA
      DO 253IBND=1,NBND(ISURFA)
      WRITE(60,6308)ISURFA,NSURF1(ISURFA,IBND),NSURF2(ISURFA,IBND)
 6308 FORMAT(/5X,'AREA',I2,': NODE ',I3,' THROUGH NODE ',I3)
  253 CONTINUE
      END IF
C
C     NODE ORDER
C
      ICOUNT=0
      DO 255 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(IEQN(INODE) .NE. 0) THEN
      ICOUNT=ICOUNT+1
      NORDER(ICOUNT)=INODE
      END IF
  255 CONTINUE
      IF(NAREA .GT. 1 .AND. NNSURF .NE. 0)THEN
C     FREE-FIELD COLUMN DESCRIPTION
      WRITE(60,6310)
 6310 FORMAT(//2X,'FREE-FIELD COLUMN DESCRIPTION')
      DO 270 I=1,2
      IF(I .EQ. 1)WRITE(60,6320)
      IF(I .EQ. 2)WRITE(60,6330)
 6320 FORMAT(/2X,'-X REGION')
 6330 FORMAT(/2X,'+X REGION')
      WRITE(60,6340)
 6340 FORMAT(T3,'LAYER',T21,'d(i)/d(1)',T32,
     & 'Cs(layer i)/Cs(1)',T57,'WT(layer i)/WT(1)',
     & T82,'DAMPING')
      WRITE(60,6350)
 6350 FORMAT(T84,'(%)'/)
      DO 260ILAYER=1,NLAYER(I)
      ITYPE=LTYPE(I,ILAYER)
      X1=REAL(CSRAT(ITYPE))
      WRITE(60,6360)ILAYER,Z(NBOTT(I,ILAYER)),X1,UWTRAT(ITYPE),
     & BETA(ITYPE)*100.0
 6360 FORMAT(T4,I2,T22,F5.2,T37,F6.2,T62,F6.2,T82,F6.3)
  260 CONTINUE
      X1=REAL(CSRAT(HALFSP))
      WRITE(60,6370)X1,UWTRAT(HALFSP),
     & BETA(HALFSP)*100.0
 6370 FORMAT('HALF-SPACE',T37,F6.2,T62,F6.2,T82,F6.3)
  270 CONTINUE
      END IF
C
C
C
C
      FBAR=FBAR/CSQRT(1.0-(0.0,2.0)*BETA(1))
      DO 280 IAREA=1,NAREA
```

```
      CSRAT(IAREA)=CSRAT(IAREA)*CSQRT((1.0-(0.0,2.0)*BETA(IAREA))/
     & (1.0-(0.0,2.0)*BETA(1)))
 280  CONTINUE
      XSCATT=0.0
      DO 320 INODE=JNODE(HALFSP,1),KNODE(HALFSP,NBOUND(HALFSP))
      DO 320 IN3=1,NN3
      DO 320 IINODE=1,3
      IF(N3(IN3,IINODE) .EQ. INODE)THEN
      XSCATT=X(INODE)
      GO TO 330
      END IF
 320  CONTINUE
 330  CONTINUE
      IF(NNSURF .NE. 0)THEN
      WRITE(60,6410)XSCATT
6410  FORMAT(/2X,'SCATTERING BOUNDARY OCCURS AT X/H1 = ',F7.2)
      END IF
      RETURN
      END
```

```
      SUBROUTINE FFLD(IX)
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
      INTEGER NSEG(2)
      REAL HRATIO(5),ZLAYER(2),FFWTR(3),FFBETA(5)
      COMPLEX VTOP,ASH(5),BSH(5),FFCSR(3),THEATA(5),CALC,
     & XIKTH1,XIKTH2,XMX(5),XI,XIKTZ,XIKX,FFSZYH(100),HSSXYH(2),VOTCRP,
     & STOUT
C
      COMPLEX XTEMP1,CALC1
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C
      PI=3.141593
C
```

```
      IF(NAREA .EQ. 1)THEN
C     NAREA = 1
      FFCSR(1)=CSRAT(HALFSP)
      FFWTR(1)=UWTRAT(HALFSP)
      FFBETA(1)=BETA(HALFSP)
      THEATA(1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      ASH(1)=(1.0,0.0)
      XMX(1)=CCOS(THEATA(1))
C
      BSH(1)= +1.0*ASH(1)
C
      FFDIM=2
      DO 5 I=1,FFDIM
      FVECT(I)=(0.0,0.0)
      DO 5J =1,FFDIM
      STIFF(I,J)=(0.0,0.0)
    5 CONTINUE
      ZHS=0.0
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(1)*
     & CSQRT(1.0-(XMX(1)**2))*ZHS
C     XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)*XHSFF(IX,INODE)
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(1)*CSQRT(1.0-(XMX(1)**2))
      HSV(IX)=
     & (ASH(1)*CEXP(-XIKTZ)+ BSH(1)*CEXP(+XIKTZ))
C
C    & *CEXP(XIKX)
C
      HSSZYH(IX)= XIKTH1*
     & (-1.0*ASH(1)*CEXP(-XIKTZ)+BSH(1)*CEXP(+XIKTZ))
C     & *CEXP(XIKX)
C
      HSSXYH(IX)= (0.0,1.0)*FBAR*XMX(1)/FFCSR(1) *
     & (ASH(1)*CEXP(-XIKTZ)+ BSH(1)*CEXP(+XIKTZ))
C     &*CEXP(XIKX)
C
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)  *X(KNODE(HALFSP,1))
      FFV(1,1)=HSV(1)*CEXP(XIKX)
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)  *X(JNODE(HALFSP,1))
      FFV(2,1)=HSV(2)*CEXP(XIKX)
      WRITE(60,6090)
      WRITE(60,6100)
      WRITE(60,6110)HSV(IX)
      WRITE(60,6130)
C
      XTEMP1=HSSZYH(IX)*(1.0-(0.0,2.0)*FFBETA(1))
C
      WRITE(60,6140)XTEMP1
      WRITE(60,6150)HSSXYH(IX)
C     WRITE(60,6170)
C     WRITE(60,6180)HSPX(IX)
C     WRITE(60,6190)HSPZ(IX)
      GO TO 9999
      END IF
C
C     NUMBER OF FREE-FIELD SEGMENTS = NNSEG
C     NNSEG=5
C     CREATE 10 SEGMENTS PER SHEAR WAVE LENGTH
      NNSEG=REAL(FBAR)*10/(2.0*PI) +1
```

```
      IF (NNSEG .LT. 5)NNSEG=5
C
      DO 10 ILAYER=1,NLAYER(IX)
      ITYPE=LTYPE(IX,ILAYER)
      FFCSR(ILAYER)=CSRAT(ITYPE)
      FFWTR(ILAYER)=UWTRAT(ITYPE)
      HRATIO(ILAYER)=Z(NBOTT(IX,ILAYER))
      FFBETA(ILAYER)=BETA(ITYPE)
   10 CONTINUE
      FFCSR(NLAYER(IX)+1)=CSRAT(HALFSP)
      FFWTR(NLAYER(IX)+1)=UWTRAT(HALFSP)
      FFBETA(NLAYER(IX)+1)=BETA(HALFSP)
      THEATA(NLAYER(IX)+1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      ASH(NLAYER(IX)+1)=(1.0,0.0)
      XMX(NLAYER(IX)+1)=CCOS(THEATA(NLAYER(IX)+1))
      DO 25ILAYER=1,NLAYER(IX)
      XMX(ILAYER)=XMX(NLAYER(IX)+1)*FFCSR(ILAYER)/FFCSR(NLAYER(IX)+1)
   25 CONTINUE
      FFDIM=2*NLAYER(IX)+1
      DO 40I=1,FFDIM
      FVECT(I)=(0.0,0.0)
      DO 40J=1,FFDIM
      STIFF(I,J)=(0.0,0.0)
   40 CONTINUE
C
C
      STIFF(1,1)=(1.0,0.0)
      STIFF(1,2)=(-1.0,0.0)
      DO 50ILAYER=1,NLAYER(IX)
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER)**2))
      XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER+1)**2))
      I=(ILAYER-1)*2+1+1
      J=(ILAYER-1)*2+1
      XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
     & * FFWTR(ILAYER+1)/FFWTR(ILAYER)
      DIFF1= ABS(  REAL(XMX(ILAYER+1)-XMX(ILAYER)) )
      DIFF2= ABS( AIMAG(XMX(ILAYER+1)-XMX(ILAYER)) )
C
      IF(DIFF1 .LT. 0.0001 .AND. DIFF2 .LT. 0.0001)THEN
      CALC= XI * FFCSR(ILAYER)/FFCSR(ILAYER+1)
      ELSE
      CALC= XI* CSQRT(1.0-(XMX(ILAYER+1)**2))/
     & CSQRT(1.0-(XMX(ILAYER)**2)) * FFCSR(ILAYER)/FFCSR(ILAYER+1)
      END IF
C
      STIFF(I,J)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I,J+1)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+1,J)   = -1.0*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+1,J+1)   = +1.0*CEXP((+1.0,0.0)*XIKTH1)
      IF(ILAYER .LT. NLAYER(IX))THEN
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+3)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+2)   = +1.0*CALC*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+1,J+3)   = -1.0*CALC*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
```

```
      STIFF(I+1,J+2)   = -1.0*CALC*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)*ASH(NLAYER(IX)+1)
      FVECT(I+1)=(-1.0,0.0)*CALC*CEXP((-1.0,0.0)*XIKTH2)*
     & ASH(NLAYER(IX)+1)
      END IF
   50 CONTINUE
      CALL SOLVE(1)
      DO 70ILAYER=1,NLAYER(IX)
      IROW=(ILAYER-1)*2+1
      ASH(ILAYER)=XVECT(IROW)
      BSH(ILAYER)=XVECT(IROW+1)
   70 CONTINUE
      BSH(NLAYER(IX)+1)=XVECT(FFDIM)
      VTOP=(ASH(1)+BSH(1))
C
C
C
      IF(IX .EQ. 1)THEN
      WRITE(60,6000)
 6000 FORMAT(//5X,'FREE-FIELD CALCULATION')
      WRITE(60,6002)
 6002 FORMAT(//5X,'-X BOUNDARY')
      END IF
      IF(IX .EQ. 2)THEN
      WRITE(60,6004)
 6004 FORMAT(//5X,'+X BOUNDARY')
      END IF
      WRITE(60,6005)
 6005 FORMAT(/T3,'LAYER',T15,'Ash',T30,'Bsh'/)
      DO 80ILAYER=1,NLAYER(IX)+1
      WRITE(60,6010)ILAYER,ASH(ILAYER),BSH(ILAYER)
 6010 FORMAT(I5,T8,2F7.3,T23,2F7.3)
   80 CONTINUE
      VREAL=REAL(VTOP)
      VIMAG=AIMAG(VTOP)
      V=SQRT(VREAL*VREAL+VIMAG*VIMAG)
      WRITE(60,6020)
 6020 FORMAT(//T18,'REAL',T31,'IMAGINARY',T46,'MAGNITUDE')
      WRITE(60,6030)VREAL,VIMAG,V
 6030 FORMAT(/2X'V(top)',T15,F8.3,T30,F8.3,T45,F8.3)
      WRITE(*,6000)
      IF(IX .EQ. 1)THEN
      WRITE(*,6002)
      END IF
      IF(IX .EQ. 2)THEN
      WRITE(*,6004)
      END IF
      WRITE(*,6005)
      DO 90ILAYER=1,NLAYER(IX)+1
      WRITE(*,6010)ILAYER,ASH(ILAYER),BSH(ILAYER)
   90 CONTINUE
      WRITE(*,6020)
      WRITE(*,6030)VREAL,VIMAG,V
C
C
      TBIG=0.
      DO 100ILAYER=1,NLAYER(IX)
      T=Z(NBOTT(IX,ILAYER))-Z(NTOP(IX,ILAYER))
```

```
      IF(T .GT. TBIG)TBIG=T
      IF(ILAYER .EQ. 1) THEN
      ZLAYER(1)=T
      ELSE
      ZLAYER(ILAYER)=ZLAYER(ILAYER-1)+T
      END IF
  100 CONTINUE
      DO 110 ILAYER=1,NLAYER(IX)
      IF (ILAYER .EQ. 1)TLAYER=ZLAYER(1)
      IF (ILAYER .GT. 1)TLAYER=ZLAYER(ILAYER)-ZLAYER(ILAYER-1)
      NSEG(ILAYER)=    TLAYER/ZLAYER(1)*NNSEG
      NSEG1=REAL(FBAR)*10/(2.0*PI)*TLAYER/REAL(FFCSR(ILAYER))
      IF(NSEG1 .GT. NSEG(ILAYER))THEN
      NSEG(ILAYER)=NSEG1
      END IF
  110 CONTINUE
C
C
      INODE=0
      DO 130 ILAYER=1,NLAYER(IX)
      ZTOP=Z(NTOP(IX,ILAYER))
      ZBOTT=Z(NBOTT(IX,ILAYER))
      SEGLEN=(ZBOTT-ZTOP)/NSEG(ILAYER)
      IF(ILAYER .EQ. 1)JFF(IX,ILAYER)=1
      IF(ILAYER .GT. 1)JFF(IX,ILAYER)=KFF(IX,ILAYER-1)+1
      KFF(IX,ILAYER)=JFF(IX,ILAYER)+NSEG(ILAYER)
      XFF(IX,ILAYER)=X(NTOP(IX,ILAYER))
C
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)   *XFF(IX,ILAYER)
C
      DO  120 ISEG=1,NSEG(ILAYER)+1
      INODE=INODE+1
      IF(IX .EQ. 1)ZFF(IX,INODE)=ZTOP+(ISEG-1)*SEGLEN
      IF(IX .EQ. 2)ZFF(IX,INODE)=ZBOTT-(ISEG-1)*SEGLEN
C
C
C
C
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & CSQRT(1.0-(XMX(ILAYER)**2))*ZFF(IX,INODE)
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*CSQRT(1.0-(XMX(ILAYER)**2))
      FFV(IX,INODE)=
     &  (ASH(ILAYER)*CEXP(-XIKTZ)+ BSH(ILAYER)*CEXP(+XIKTZ))*CEXP(XIKX)
C
      FFSZYH(INODE)=XIKTH1*
     &  (-1.0*ASH(ILAYER)*CEXP(-XIKTZ)+BSH(ILAYER)*CEXP(+XIKTZ))*
     & CEXP(XIKX)
C
      FFSXYH(IX,INODE)=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)*
     &  (ASH(ILAYER)*CEXP(-XIKTZ)+ BSH(ILAYER)*CEXP(+XIKTZ))*CEXP(XIKX)
C
C
  120 CONTINUE
  130 CONTINUE
      DO 150 ILAYER=1,NLAYER(IX)
      WRITE(60,6050)ILAYER
 6050 FORMAT(/5X,'LAYER = ',I1)
      WRITE(60,6060)
```

```
 6060 FORMAT(T21,'DISPLACEMENT',T60,'STRESS')
      WRITE(60,6070)
 6070 FORMAT(T2,'NODE',T8,'X/H1',T15,'Z/H1',T27,'V',
     & T51,'SZY*H1/Gi',T65,'SXY*H1/Gi*'/)
      DO 140 INODE=JFF(IX,ILAYER),KFF(IX,ILAYER)
C
      XTEMP1=FFSZYH(INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
C
      WRITE(60,6080)INODE,XFF(IX,ILAYER),ZFF(IX,INODE),
     & FFV(IX,INODE),XTEMP1,FFSXYH(IX,INODE)
 6080 FORMAT(I4,T6,F7.3,T14,F6.3,
     & T20,2F7.3,T48,2F7.3,T63,2F7.3)
  140 CONTINUE
  150 CONTINUE
C
      NN=NLAYER(IX)+1
C
C
C     FREE-FIELD MOTION IS BASED ON HORIZONTAL HALF-SPACE INTERFACE
C
C
C
      ZHS=Z(KNODE(HALFSP,NBOUND(HALFSP)))
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XMX(NN)**2))*ZHS
C     XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)*XHSFF(IX,INODE)
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(NN)*CSQRT(1.0-(XMX(NN)**2))
      HSV(IX)=
     & (ASH(NN)*CEXP(-XIKTZ)+ BSH(NN)*CEXP(+XIKTZ))
C
C    & *CEXP(XIKX)
C
      HSSZYH(IX)= XIKTH1*
     & (-1.0*ASH(NN)*CEXP(-XIKTZ)+BSH(NN)*CEXP(+XIKTZ))
C     & *CEXP(XIKX)
C
      HSSXYH(IX)= (0.0,1.0)*FBAR*XMX(1)/FFCSR(1)*
     & (ASH(NN)*CEXP(-XIKTZ)+ BSH(NN)*CEXP(+XIKTZ))
C     &*CEXP(XIKX)
C
  160 CONTINUE
      WRITE(60,6090)
 6090 FORMAT(/5X,'HALF-SPACE')
      WRITE(60,6100)
 6100 FORMAT(/2X,'DISPLACEMENT: ')
      WRITE(60,6110)HSV(IX)
 6110 FORMAT(5X,'v/exp(ikx) =',T30,2F8.3)
      WRITE(60,6130)
 6130 FORMAT(/2X,'STRESS: ')
C
      XTEMP1=HSSZYH(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
C
      WRITE(60,6140)XTEMP1
 6140 FORMAT(5X,'(Szy*H1/Ghs) /exp(ikx) =',T30,2F8.3)
      WRITE(60,6150)HSSXYH(IX)
 6150 FORMAT(5X,'(Sxy*H1/Ghs*)/exp(ikx) =',T30,2F8.3)
C     WRITE(60,6170)
C 6170 FORMAT(/2X,'TRACTION: ')
```

```
C      WRITE(60,6180)HSPX(IX)
C 6180 FORMAT(5X,'(Px*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C      WRITE(60,6190)HSPZ(IX)
C 6190 FORMAT(5X,'(Pz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C
C      HALF-SPACE OUTCROP
       VOTCRP= 2.0*ASH(NN)*CEXP(-XIKTZ)
       VOTMAG=SQRT(REAL(VOTCRP)**2+AIMAG(VOTCRP)**2)
       STOUT=(0.0,0.0)
       WRITE(60,6200)
 6200  FORMAT(/5X,'ROCK OUTCROP')
       WRITE(60,6210)
 6210  FORMAT(/2X,'DISPLACEMENT: ')
       WRITE(60,6220)VOTCRP,VOTMAG
 6220  FORMAT(5X,'u =',T30,2F8.3,T50,'|u| = ',F6.3)
       WRITE(60,6240)
 6240  FORMAT(/2X,'STRESS: ')
       CALC1=(0.0,0.0)
C
       WRITE(60,6250)CALC1
 6250  FORMAT(5X,'(Szy*H1/Ghs) /exp(ikx) =',T30,2F8.3)
       CALC1=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)*VOTCRP
       WRITE(60,6260)CALC1
 6260  FORMAT(5X,'(Sxy*H1/Ghs*)/exp(ikx) =',T30,2F8.3)
C      WRITE(60,6280)
C 6280 FORMAT(/2X,'TRACTION: ')
C      WRITE(60,6290)CALC1
C 6290 FORMAT(5X,'(Px*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C      WRITE(60,6300)CALC1
C 6300 FORMAT(5X,'(Pz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C
 9999  CONTINUE
       RETURN
       END
```

```
      SUBROUTINE MATRIX
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
```

```
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
      COMPLEX DSTAT,DCORR
C     789112345678921234567893123456789412345678951234567896123456789712
C
      REAL XKH1,XKSH1
C
C**************************************************
C**************************************************
      XMXHS=COS(ANGLE(IANGLE)/57.29577951)
      XKSH1=FBAR/CSRAT(HALFSP)
      XKH1=XKSH1*XMXHS
C
      IF (IANGLE .EQ. 1)THEN
C
      PI=3.14159265
      TWOPI=2.0*PI
C
      HDIM=KNODE(NAREA,NBOUND(NAREA))
      GDIM=KNODE(NAREA,NBOUND(NAREA))+3*NN3
C     FORM "G" & "H" MATRIX FOR EACH SUBREGION
      DO 20IROW=1,KNODE(NAREA,NBOUND(NAREA))
      FVECT(IROW)=(0.0,0.0)
      DO 5JCOL=1,HDIM
      HMAT(IROW,JCOL)=(0.0,0.0)
    5 CONTINUE
      DO 10 JCOL=1,GDIM
      GMAT(IROW,JCOL)=(0.0,0.0)
   10 CONTINUE
   20 CONTINUE
C     OPEN(UNIT=80,FILE='TRUNC.CK',STATUS='UNKNOWN')
C
C     CHECK IF HALF-SPACE OR FULL-SPACE PROBLEM
      IFULSP=0
      ICOUNT=0
      DO IBOUND=1,NBOUND(HALFSP)
      ICOUNT=ICOUNT+ICLOSE(HALFSP,IBOUND)
      END DO
      IF (ICOUNT .EQ. NBOUND(HALFSP))IFULSP=1
C     IF IFULSP =1, FULL SPACE PROBLEM
C     IF IFULSP =0, HALF-SPACE PROBLEM
C
      JGMAT=0
      IN3=0
      JCOUNT=0
      DO 100IAREA=1,NAREA
C     WRITE(10,*)'REGION ',IAREA
C
            DO IBOUND =1,NBOUND(IAREA)
            DO ISRCE =JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
C
      JCOUNT=JCOUNT+1
C
      IF(JCOUNT .EQ. 10)THEN
      WRITE(*,*)'WORKING ON INFLU FOR SOURCE POINT  ',ISRCE
      JCOUNT=0
```

```
      END IF
C
      CALL INFLU(ISRCE,X(ISRCE),Z(ISRCE),IAREA,JGMAT)
C
C
C     FIND DIAGONAL TERMS OF HMAT
C
      IF(IAREA .EQ. HALFSP .AND. IFULSP .EQ. 0)THEN
C
C     DETERMINE INTERIOR ANGLES
      JNO=JNODE(IAREA,IBOUND)
      KNO=KNODE(IAREA,IBOUND)
      IROW=ISRCE
      JCOL=IROW
C
      IF(ISRCE .EQ. JNO .OR. ISRCE .EQ. KNO)THEN
C
            IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      ROTAT=PI
      HMAT(IROW,JCOL)=ROTAT/TWOPI+HMAT(IROW,JCOL)
            ELSE
      IF (ISRCE .EQ. JNO)THEN
      X1=X(KNO)
      Z1=Z(KNO)
      X2=X(ISRCE+1)
      Z2=Z(ISRCE+1)
      ELSE
      X1=X(ISRCE-1)
      Z1=Z(ISRCE-1)
      X2=X(JNO)
      Z2=Z(JNO)
      END IF
      XI=X(ISRCE)
      ZI=Z(ISRCE)
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      HMAT(IROW,JCOL)=ROTAT/(TWOPI)+HMAT(IROW,JCOL)
C
            END IF
            ELSE
C
      X1=X(ISRCE-1)
      X2=X(ISRCE+1)
      XI=X(ISRCE)
      Z1=Z(ISRCE-1)
      Z2=Z(ISRCE+1)
      ZI=Z(ISRCE)
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      HMAT(IROW,JCOL)=ROTAT/(2.*PI)+HMAT(IROW,JCOL)
   40 CONTINUE

C
            END IF
C

C
C     WRITE(*,*)'ISRCE = ',ISRCE
C     WRITE(*,9040)
C 9040 FORMAT(/'Cij')
```

```
C      DO I=1,2
C      WRITE(*,9050)((DSTAT(I,J)-DCORR(I,J)),J=1,2)
C 9050 FORMAT(2F10.3,5X,2F10.3)
C      END DO
C      WRITE(*,9052)
C 9052 FORMAT(/'Dij')
C      WRITE(*,9054)HMAT(IROW,JCOL)
C 9054 FORMAT(2F10.3,5X,2F10.3)
C      PAUSE
       ELSE
C
       CALL STATIC(ISRCE,X(ISRCE),Z(ISRCE),IAREA,DSTAT,DCORR)
       IROW=ISRCE
       JCOL=IROW
C      SINCE "DIST" = 0.0, DCORR =0.0
       HMAT(IROW,JCOL)=DSTAT+HMAT(IROW,JCOL)-DCORR
C
C
C      WRITE(*,*)'ISRCE = ',ISRCE
C      WRITE(*,9040)
C      WRITE(*,9050)DSTAT-DCORR
C      WRITE(*,9052)
C      WRITE(*,9054)HMAT(IROW,JCOL)
C
       END IF
C
             END DO
             END DO
C      WRITE(10,*)'GMAT'
       ISTART=JNODE(IAREA,1)
       ISTOP= KNODE(IAREA,NBOUND(IAREA))
       JSTART=JCOL1(JNODE(IAREA,1))
       JSTOP= JCOL1(KNODE(IAREA,NBOUND(IAREA)))
C      WRITE(10,*)'ISTART = ',ISTART
C      WRITE(10,*)'ISTOP = ',ISTOP
C      WRITE(10,*)'JSTART = ',JSTART
C      WRITE(10,*)'JSTOP = ',JSTOP
       DO 50 IROW=ISTART,ISTOP
C      WRITE(10,1010)(GMAT(IROW,J),J=JSTART,JSTOP)
 1010  FORMAT(8F10.5)
   50  CONTINUE
C      WRITE(10,1015)
 1015  FORMAT(/)
C      WRITE(10,*)'HMAT'
       DO 1020 I=ISTART,ISTOP
C      WRITE(10,1010)(HMAT(I,J),J=ISTART,ISTOP)
 1020  CONTINUE
  100  CONTINUE
C
       CALL FIXMAT
       CALL AFORM
       END IF
C*******************************************************
C*******************************************************
       JCOUNT=0
       DO 150IAREA=1,NAREA
C      WRITE(10,*)'REGION ',IAREA
       DO 140ISRCE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
```

```
      TCORR(ISRCE)=(0.0,0.0)
      IF(NAREA .EQ. 1 .OR. NNSURF .EQ. 0) GO TO 140
C
C
      JCOUNT=JCOUNT+1
      IF(JCOUNT .EQ. 10)THEN
      WRITE(*,*)'WORKING ON CORRCT FOR SOURCE POINT ',ISRCE
      JCOUNT=0
      END IF
      IF(IAREA .NE. HALFSP)CALL CORRCT(ISRCE,X(ISRCE),Z(ISRCE),IAREA)
      IF(IAREA .EQ. HALFSP .AND.
     & ISRCE .NE. JNODE(HALFSP,1) .AND.
     & ISRCE .NE. KNODE(HALFSP,NBOUND(HALFSP)))THEN
      CALL HSCORR(ISRCE,X(ISRCE),Z(ISRCE),IAREA)
      END IF
  140 CONTINUE
      DO 150INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IROW=INODE
      FVECT(IROW)=TCORR(INODE)
  150 CONTINUE
      DO 170 INODE=JNODE(HALFSP,1),
     &               KNODE(HALFSP,NBOUND(HALFSP))
      IROW=INODE
      IINODE=INODE-JNODE(HALFSP,1)+1
      FVECT(IROW)=FVECT(IROW)+VINCDT(IINODE)
  170 CONTINUE
C
      IF(NAREA .EQ. 1 .AND. ICLOSE(1,1) .EQ. 1) GO TO 190

                  IF(NNSURF .GT. 0 .AND. NAREA .GT. 1)THEN
C
      DO  ISIDE =1,2
C
            IF (NAREA .GT. 1)THEN
C
            DO ILAYER =1,NLAYER(ISIDE)
      IF(ISIDE .EQ. 1)INODE=JFF(ISIDE,ILAYER)
      IF(ISIDE .EQ. 2)INODE=KFF(ISIDE,ILAYER)
      IROW=NTOP(ISIDE,ILAYER)
      ICOUNT=0
      IF(NCONN(NTOP(ISIDE,ILAYER)) .EQ. 0)THEN
      NNODE=NTOP(ISIDE,ILAYER)
      ELSE
      NNODE=NCONN(NTOP(ISIDE,ILAYER))
      END IF
      DO I=1,NNODE
      IF(NCONN(I) .EQ. 0)ICOUNT=ICOUNT+1
      END DO
      JCOL=ICOUNT
      FVECT(IROW)=FFV(ISIDE,INODE)
      DO J=1,ADIM
      HMAT(IROW,J)=(0.0,0.0)
      END DO
      HMAT(IROW,JCOL)=(1.0,0.0)
C
            END DO
C
            END IF
```

```
C
      IF(ISIDE .EQ. 1)INODE=KNODE(HALFSP,NBOUND(HALFSP))
      IF(ISIDE .EQ. 2)INODE=JNODE(HALFSP,1)
      IROW=INODE
      IF(NCONN(INODE) .EQ. 0)THEN
      NNODE=INODE
      ELSE
      NNODE=NCONN(INODE)
      END IF
      ICOUNT=0
      DO I=1,NNODE
      IF(NCONN(I) .EQ. 0)ICOUNT=ICOUNT+1
      END DO
      JCOL=ICOUNT
      FVECT(IROW)=HSV(ISIDE)*CEXP((0.0,1.0)*XKH1*X(INODE))
      DO 180 J=1,ADIM
      HMAT(IROW,J)=(0.0,0.0)
  180 CONTINUE
      HMAT(IROW,JCOL)=(1.0,0.0)
C
            END DO
C
C
                  END IF
C
  190 CONTINUE
C     WRITE(10,1015)
C     WRITE(10,*)'FVECT'
C     DO 1050 I=1,2*KNODE(NAREA,NBOUND(NAREA))
C     WRITE(10,1040)I,FVECT(I)
C 1040 FORMAT(I5,5X,8F10.5)
C 1050 CONTINUE
C     CLOSE(UNIT=80)
      RETURN
      END

C


      SUBROUTINE AFORM
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/AMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX AMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
```

```
      INTEGER IEQN
C
      INTEGER ACOL
C
C     7891123456789212345678931234567894123456789512345678961234567897l2
C
C     [A]*{x} = {F}
C     FORM 'A' MATRIX
C     A [DISPLACEMENT/STRESS],F {DISPLACEMENT}
C
      ACOL=HDIM
      ADIM=ACOL
      IADD=0
      IF(NAREA .EQ. 1)GO TO 40
      DO 30 INODE=1,KNODE(NAREA,NBOUND(NAREA))
C     SURFACE NODE WITH NO INTERFACE
      IF(IEQN(INODE) .EQ. 1 .AND. NSURF(INODE) .EQ. 1)THEN
      GO TO 30
      END IF
      IF(NCONN(INODE) .EQ. 0 )THEN
      IADD=IADD+1
      ACOL=HDIM+IADD
      DO 10IROW=1,KNODE(NAREA,NBOUND(NAREA))
      AMAT(IROW,ACOL)= -1.*GMAT(IROW,JCOL1(INODE))
   10 CONTINUE
      IF(IEQN(INODE) .EQ. 3)THEN
      IADD=IADD+1
      ACOL=HDIM+IADD
      DO 20 IROW=1,KNODE(NAREA,NBOUND(NAREA))
      AMAT(IROW,ACOL)= -1.*GMAT(IROW,JCOL1(INODE)+1)
   20 CONTINUE
      END IF
      END IF
   30 CONTINUE
      ADIM=ACOL
   40 CONTINUE
      NDIM=ADIM
C
C     WRITE(10,*)'AMAT'
C     WRITE(10,*)'ADIM = ',ADIM
      DO 50 I=1,KNODE(NAREA,NBOUND(NAREA))
C     WRITE(10,1000)(AMAT(I,J),J=1,ADIM)
 1000 FORMAT(4F10.5)
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INFLU(ISRCE,XI,ZI,IAREA,JGMAT)
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5)
      COMPLEX G(5),T(5),XKSR,H01KSR,H11KSR
C
```

```
      PI=4.*ATAN(1.0)
      JADD=JGMAT
C     CALCULATE INFLUENCE FUNCTIONS
      DO 70  IBOUND=1,NBOUND(IAREA)
      JNO=JNODE(IAREA,IBOUND)
      JNDCOL= JNODE(IAREA,IBOUND)
C     WRITE(10,*)'INODE = ',INODE
C
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      NNODE=KNODE(IAREA,IBOUND)-1
      ELSE
      NNODE=KNODE(IAREA,IBOUND)
      END IF
C
      DO 60 INODE=JNODE(IAREA,IBOUND),NNODE
      X1=X(INODE)
      Z1=Z(INODE)
C
      IF(INODE .NE. KNODE(IAREA,IBOUND))THEN
      X2=X(INODE+1)
      Z2=Z(INODE+1)
      IINODE=INODE+1
      ELSE
      X2=X(JNO)
      Z2=Z(JNO)
      IINODE=JNO
      END IF
C
      SEGLEN=SQRT((X2-X1)**2+(Z2-Z1)**2)
      DIFF=ABS(X2-X1)
      IF(DIFF .LT. .00001)THEN
      DIST=X2-XI
      ELSE
      SLOPE=(Z2-Z1)/(X2-X1)
      DIST=(ZI-Z2-SLOPE*(XI-X2))/SQRT(SLOPE**2+1)
      END IF
      DIST=ABS(DIST)
C     WRITE(10,*)ISRCE,INODE,SEGLEN
      IF (DIST .LT. 0.000001) GO TO 10
C     IF ROTAT .GT. 0. CLOCKWISE , DIST/RADIUS = -
C     IF ROTAT .LT. 0. COUNTERCLOCKWISE, DIST/RADIUS = +
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      IF(ROTAT .GT. PI)ROTAT=-1.*(2.*PI-ROTAT)
      IF(ROTAT .GT. 0.) DIST=-1.*DIST
   10 CONTINUE
      DO 40IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(X2-X1)/2.0+X1
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
      XKSR=FBAR*RADIUS(IGAUSS)/CSRAT(IAREA)
C
C     SURFACE ELEMENT , G * traction =0
      IF(NSURF(INODE) .EQ. 1 .AND. NSURF(IINODE) .EQ. 1)THEN
C     IF(NSURF(INODE) .EQ. 1 .AND. NSURF(IINODE) .EQ. 1
C     & .AND. IEQN(INODE) .EQ. 1 .AND. IEQN(IINODE) .EQ. 1)THEN
      G(IGAUSS)=(0.0,0.0)
      ELSE
```

```
      CALL HANKEL(0,XKSR,H01KSR)
      G(IGAUSS)=(0.0,0.25)*H01KSR
      END IF
C
C     ELEMENT IS HORIZONTAL WITH ISRCE
      IF(ABS(DIST) .LT. 0.000001)THEN
      T(IGAUSS)=(0.0,0.0)
      ELSE
      CALL HANKEL(1,XKSR,H11KSR)
      T(IGAUSS)=
     & (0.0,-0.25)*FBAR/CSRAT(IAREA)*DIST/RADIUS(IGAUSS)*H11KSR
      END IF
C
   40 CONTINUE
C     IF INODE IS PART OF 3 NODE INTERFACE, INCLUDE P1 & P2 ON BOTH
SIDES OF NODE
      IF(NN3 .EQ. 0) GO TO 47
      DO 45I=1,NN3
      DO 45J=1,3
      IF(N3(I,J) .EQ. INODE) THEN
      JADD =JADD+1
      GO TO 47
      END IF
   45 CONTINUE
   47 CONTINUE
      JCOL=INODE+JADD
      JCOL1(INODE)=JCOL
      JCOL1(INODE+1)=JCOL+1
      JJCOL=INODE
      IROW= ISRCE
      IF(INODE .EQ. NNODE .AND. ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
C
      KCOL=JCOL1(JNO)
      KKCOL=JNDCOL
C
      ELSE
C
      KCOL=JCOL+1
      KKCOL=JJCOL+1
C
      END IF
C
C
C
      GMAT(IROW,JCOL)=GMAT(IROW,JCOL)
     &   + (WT3(1)*WTFN3(1)*G(1)+WT3(2)*WTFN3(2)*G(2)
     &   +  WT3(1)*WTFN3(3)*G(3))*SEGLEN/2.
C
      HMAT(IROW,JJCOL)= HMAT(IROW,JJCOL)
     &   + (WT3(1)*WTFN3(1)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(3)*T(3))*SEGLEN/2.
C
C
      GMAT(IROW,KCOL)=GMAT(IROW,KCOL)
     &   + (WT3(1)*WTFN3(3)*G(1)+WT3(2)*WTFN3(2)*G(2)
     &   +  WT3(1)*WTFN3(1)*G(3))*SEGLEN/2.
C
      HMAT(IROW,KKCOL)=HMAT(IROW,KKCOL)
```

```
  &   + (WT3(1)*WTFN3(3)*T(1)+WT3(2)*WTFN3(2)*T(2)
  &   +  WT3(1)*WTFN3(1)*T(3))*SEGLEN/2.
50 CONTINUE
60 CONTINUE
70 CONTINUE
   IF(ISRCE .EQ. KNODE(IAREA,NBOUND(IAREA)))THEN
   JGMAT=JADD
   END IF
   RETURN
   END
```

```
      SUBROUTINE HANKEL(IORDER,Z,HOK)
C     ********************************************
C     HANKEL FUNCTION OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C           0 KIND
C     ********************************************
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT(2),DIFF,ADELTA,PI,XREAL,YIMAG,CALC,
     &  AA,A,B
      COMPLEX HOK,Z,JN,YN
C
C     WRITE(*,9000)
C 9000 FORMAT(2X,'INPUT ORDER',I2)
C     READ(*,*)IORDER
C     WRITE(*,9010)
C 9010 FORMAT(2X,'REAL PART OF kr')
C     READ(*,*)XREAL
C     WRITE(*,9020)
C 9020 FORMAT(2X,'IMAGINARY PART OF kr')
C     READ(*,*)YIMAG
C     Z=CMPLX(XREAL,YIMAG)
C
      PI=3.1415926535D0
      TOLER=.000001D0
C
      XREAL=REAL(Z)
      YIMAG=AIMAG(Z)
C     CALCULATE BESSEL FUNCTION OF 1ST KIND (JN)
      A=0.0D0
      B=PI
      DO 25ITYPE=1,2
      HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   10 CONTINUE
      ICYCLE=ICYCLE+1
      ATOT(ITYPE)=0.0
      N=(B-A)/HDELTA
      DO 20I=1,N
      AA= (I-1)*HDELTA+A
      DO 15J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))*DCOSH(YIMAG* DSIN(X(J)))
      ELSE
      Y(J)=DSIN(XREAL* DSIN(X(J))-IORDER*X(J))* DSINH(YIMAG*DSIN(X(J)))
      END IF
   15 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
```

```
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   20 CONTINUE
C     N1=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 10
      END IF
   25 CONTINUE
      JN=(ATOT(1)-(0.0,1.0)*ATOT(2))/PI
C     CALCULATE BESSEL FUNCTION OF 2ND KIND, ORDER N (YN)
      A=0.
      B=PI
      DO 55ITYPE=1,2
      HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   30 CONTINUE
      ICYCLE=ICYCLE+1
      N=(B-A)/HDELTA
      ATOT(ITYPE)=0.0
      DO 50I=1,N
      AA= (I-1)*HDELTA+A
      DO 40J=1,NGAUSS
      X(J)=AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)= DSIN(XREAL* DSIN(X(J))-IORDER*X(J))*DCOSH(YIMAG* DSIN(X(J)))
      ELSE
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))*DSINH(YIMAG* DSIN(X(J)))
      END IF
   40 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   50 CONTINUE
C     N2=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 30
      END IF
   55 CONTINUE
      YN=ATOT(1)+(0.0,1.0)*ATOT(2)
      A=0.
      B=10.
      DO 90ITYPE=1,2
      HDELTA=1.0
C     HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   60 CONTINUE
      ICYCLE=ICYCLE+1
      ATOT(ITYPE)=0.0
      N=(B-A)/HDELTA
      DO 80I=1,N
```

```
      AA= (I-1)*HDELTA+A
      DO 70J=1,NGAUSS
      X(J)=AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALC=XREAL*DSINH(X(J))
      IF(CALC .GT. 40.)CALC=40.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)=(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)*DCOS(YIMAG*DSINH(X(J)))
      ELSE
      Y(J)=(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)*DSIN(YIMAG*DSINH(X(J)))
      END IF
   70 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   80 CONTINUE
C     N3=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 60
      END IF
   90 CONTINUE
      YN=(YN-ATOT(1)+(0.0,1.0)*ATOT(2))/PI
      HOK=JN+(0.0,1.0)*YN
C     WRITE(*,9030)
C 9030 FORMAT(5X,'HANKEL FUNCTION ')
C     WRITE(*,9040)IORDER
C 9040 FORMAT(5X,'ORDER = ',I1,3X,'KIND = 0')
C     WRITE(*,9050)HOK
C 9050 FORMAT(5X,F10.5,2X,' + i',F10.5)
C     WRITE(*,9060)N1
C 9060 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N1 = ',I4)
C     WRITE(*,9070)N2
C 9070 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N2 = ',I4)
C     WRITE(*,9080)N3
C 9080 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N3 = ',I4)
C     PAUSE
      RETURN
      END
```

```
      SUBROUTINE ROTATE (X1,Y1,X2,Y2,XI,YI,BETA)
C     CALCULATE ANGLE BETWEEN X1,Y1 & X2,Y2 ON DOMAIN SIDE
      DOUBLE PRECISION X1DIFF,Y1DIFF,X2DIFF,Y2DIFF,R,
     & ALPHA,PSI,X2NEW,Y2NEW
      PI=3.14159265
      X1DIFF=X1-XI
      X2DIFF=X2-XI
      Y1DIFF=Y1-YI
      Y2DIFF=Y2-YI
      IF (ABS(X1DIFF) .LT. 0.000001)X1DIFF=0.000001
      R=DSQRT(X2DIFF**2+Y2DIFF**2)
      ALPHA=DATAN(Y1DIFF/X1DIFF)
C     WRITE(10,*)XI,YI,X1,Y1,X1DIFF,Y1DIFF
C     WRITE(10,*)X2,Y2,X2DIFF,Y2DIFF
C     WRITE(10,*)R,ALPHA
      CALL QUAD(X1DIFF,Y1DIFF,ALPHA)
      X2NEW=X2DIFF*DCOS(ALPHA)+Y2DIFF*DSIN(ALPHA)
      Y2NEW=Y2DIFF*DCOS(ALPHA)-X2DIFF*DSIN(ALPHA)
      PSI=DASIN(Y2NEW/R)
      CALL QUAD(X2NEW,Y2NEW,PSI)
      BETA=PSI
      RETURN
      END


      SUBROUTINE QUAD(X,Y,ANGLE)
      DOUBLE PRECISION ANGLE,X,Y
C     DETERMINE QUADRANT IN WHICH ANGLE IS LOCATED
      PI=3.14159265
      ANGLE=ABS(ANGLE)
C     QUADRANT 2
      IF (X .LT. 0. .AND. Y .GE. 0.) ANGLE = PI-ANGLE
C     QUADRANT 3
      IF (X .LT. 0. .AND. Y .LT. 0.) ANGLE = PI+ANGLE
C     QUADRANT 4
      IF (X .GE. 0. .AND. Y .LT. 0.) ANGLE = 2.*PI-ANGLE
      RETURN
      END
```

```
      SUBROUTINE STATIC(ISRCE,XI,ZI,IAREA,DSTAT,DCORR)
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
```

```
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5),XN(2)
      COMPLEX T(5),DSTAT,DCORR

C
C
      DSTAT=(0.0,0.0)
      DCORR=(0.0,0.0)
      PI=4.*ATAN(1.0)
C     CALCULATE INFLUENCE FUNCTIONS
      DO 70  IBOUND=1,NBOUND(IAREA)
      JNO=JNODE(IAREA,IBOUND)
C     WRITE(10,*)'INODE = ',INODE
C
      IISRCE=ISRCE
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      NNODE=KNODE(IAREA,IBOUND)-1
      ELSE
      NNODE=KNODE(IAREA,IBOUND)
      IF(ISRCE .EQ. JNO)IISRCE=NNODE+1
      END IF
C
      DO 60 INODE=JNODE(IAREA,IBOUND),NNODE
      X1=X(INODE)
      Z1=Z(INODE)
C
      IF(INODE .NE. KNODE(IAREA,IBOUND))THEN
      X2=X(INODE+1)
      Z2=Z(INODE+1)
      ELSE
      X2=X(JNO)
      Z2=Z(JNO)
      END IF
C
      SEGLEN=SQRT((X2-X1)**2+(Z2-Z1)**2)
      DIFF=ABS(X2-X1)
      IF(DIFF .LT. .00001)THEN
      DIST=X2-XI
      ELSE
      SLOPE=(Z2-Z1)/(X2-X1)
      DIST=(ZI-Z2-SLOPE*(XI-X2))/SQRT(SLOPE**2+1)
      END IF
      DIST=ABS(DIST)
C     WRITE(10,*)ISRCE,INODE,SEGLEN
      IF (DIST .LT. 0.000001) GO TO 10
C     IF ROTAT .GT. 0. CLOCKWISE , DIST/RADIUS = -
C     IF ROTAT .LT. 0. COUNTERCLOCKWISE, DIST/RADIUS = +
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      IF(ROTAT .GT. PI)ROTAT=-1.*(2.*PI-ROTAT)
      IF(ROTAT .GT. 0.) DIST=-1.*DIST
   10 CONTINUE
C
      DO 40IGAUSS=1,NGAUS3
```

```
      XGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(X2-X1)/2.0+X1
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
C
      T(IGAUSS)=
     & -1.0/(2.0*PI)*DIST/(RADIUS(IGAUSS)*RADIUS(IGAUSS))
C
C
   40 CONTINUE
             IF (INODE .NE. ISRCE) THEN
      DSTAT= DSTAT
     &   + (WT3(1)*WTFN3(1)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(3)*T(3))*SEGLEN/2.
C
             ELSE
C
      DCORR= DCORR
     &   + (WT3(1)*WTFN3(1)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(3)*T(3))*SEGLEN/2.
      END IF
C
C
      IINODE = INODE + 1
             IF (IINODE .NE. ISRCE .AND.
     &                 IINODE .NE. IISRCE)THEN
      DSTAT=DSTAT
     &   + (WT3(1)*WTFN3(3)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(1)*T(3))*SEGLEN/2.
             ELSE
      DCORR=DCORR
     &   + (WT3(1)*WTFN3(3)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(1)*T(3))*SEGLEN/2.
             END IF
C
   50 CONTINUE
   60 CONTINUE
   70 CONTINUE
C
      IF (NLAYER(1) .GT. 0 .OR. NLAYER(2) .GT. 0)THEN
      DO ISIDE =1,2
      LAYER=0
C
                  IF(NBOUND(IAREA) .EQ. 1)THEN
      DO ILAYER=1,NLAYER(ISIDE)
      N1=NTOP(ISIDE,ILAYER)
      N2=NBOTT(ISIDE,ILAYER)
      IF(N1 .GT. N2)THEN
      NTEMP=N1
      N1=N2
      N2=NTEMP
      END IF
      IF(ISRCE .GE. N1 .AND. ISRCE .LE. N2)THEN
      LAYER=ILAYER
      GO TO 75
      END IF
      END DO
      GO TO 200
```

```
   75 CONTINUE
                  ELSE
C
            DO ILAYER =1,NLAYER(ISIDE)
      IF(LTYPE(ISIDE,ILAYER) .EQ. IAREA) LAYER=ILAYER
            END DO
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      IF(LAYER .EQ. 0) THEN
      WRITE(*,*)'PROBLEM IN SUBROUTINE CORRECT'
      PAUSE
      STOP
      END IF
                  END IF
C
      X1=XFF(ISIDE,LAYER)
      X2=XFF(ISIDE,LAYER)
      DO 150 INODE=JFF(ISIDE,LAYER),KFF(ISIDE,LAYER)-1
      SEGLEN=ABS(ZFF(ISIDE,INODE+1)-ZFF(ISIDE,INODE))
      XN(1)=(ZFF(ISIDE,INODE)-ZFF(ISIDE,INODE+1))/SEGLEN
      XN(2)=0.0
C
      Z1=ZFF(ISIDE,INODE)
      Z2=ZFF(ISIDE,INODE+1)
C
      SEGLEN=SQRT((X2-X1)**2+(Z2-Z1)**2)
      DIFF=ABS(X2-X1)
      IF(DIFF .LT. .00001)THEN
      DIST=X2-XI
      ELSE
      SLOPE=(Z2-Z1)/(X2-X1)
      DIST=(ZI-Z2-SLOPE*(XI-X2))/SQRT(SLOPE**2+1)
      END IF
      DIST=ABS(DIST)
C     WRITE(10,*)ISRCE,INODE,SEGLEN
      IF (DIST .GT. 0.000001) THEN
C     IF ROTAT .GT. 0. CLOCKWISE , DIST/RADIUS = -
C     IF ROTAT .LT. 0. COUNTERCLOCKWISE, DIST/RADIUS = +
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      IF(ROTAT .GT. PI)ROTAT=-1.*(2.*PI-ROTAT)
      IF(ROTAT .GT. 0.) DIST=-1.*DIST
      END IF
C
C
      DO 140IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=XFF(ISIDE,LAYER)
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
C
      T(IGAUSS)=
     & -1.0/(2.0*PI)*DIST/(RADIUS(IGAUSS)*RADIUS(IGAUSS))
C
C
  130 CONTINUE
  140 CONTINUE
C
      DIST=SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,INODE))**2)
```

```
            IF (DIST .GE. 0.00001) THEN
      DSTAT= DSTAT
     &   + (WT3(1)*WTFN3(1)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(3)*T(3))*SEGLEN/2.
C
            END IF
C
      IINODE = INODE + 1
      DIST=SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,IINODE))**2)
            IF (DIST .GE. 0.00001) THEN
      DSTAT=DSTAT
     &   + (WT3(1)*WTFN3(3)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +  WT3(1)*WTFN3(1)*T(3))*SEGLEN/2.
            END IF
C
  150 CONTINUE
C
C
  200 CONTINUE
            END DO
C
      END IF
      DSTAT= -DSTAT
      IF(IAREA .EQ. HALFSP)THEN
      DSTAT=(1.0,0.0)+DSTAT
      END IF
C
C
C     WRITE(*,*)'ISRCE = ',ISRCE
C     WRITE(*,9000)
C 9000 FORMAT('DSTAT')
C     WRITE(*,9010)DSTAT
C 9010 FORMAT(2F10.3,5X,2F10.3)
C
C
C     WRITE(*,9020)
C 9020 FORMAT(/'DCORR')
C     WRITE(*,9030)DCORR
C 9030 FORMAT(2F10.3,5X,2F10.3)
C     WRITE(*,9040)
C 9040 FORMAT(/'Cij')
C     WRITE(*,9050)(DSTAT-DCORR)
C 9050 FORMAT(2F10.3,5X,2F10.3)
C
C     PAUSE
C
C
C
C
      RETURN
      END
```

```
      SUBROUTINE CORRCT(ISRCE,XI,ZI,IAREA)
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5)
      COMPLEX G(5),T(5),XKSR,H01KSR,H11KSR,VCORR,STCORR
C
      PI=4.*ATAN(1.0)
      TCORR(ISRCE)=(0.0,0.0)
      STCORR=(0.0,0.0)
      VCORR=(0.0,0.0)
```

```
C     WRITE(80,*)'ISRCE',ISRCE
C
C
C
C
C     CALCULATE INFLUENCE FUNCTIONS
      DO 60  ISIDE=1,2
      LAYER=0
C
            IF(NBOUND(IAREA) .EQ. 1)THEN
      DO 4ILAYER=1,NLAYER(ISIDE)
      N1=NTOP(ISIDE,ILAYER)
      N2=NBOTT(ISIDE,ILAYER)
      IF(N1 .GT. N2)THEN
      NTEMP=N1
      N1=N2
      N2=NTEMP
      END IF
      IF(ISRCE .GE. N1 .AND. ISRCE .LE. N2)THEN
      LAYER=ILAYER
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      GO TO 6
      END IF
    4 CONTINUE
C     WRITE(*,*)'SIDE = ',ISIDE,' IS NOT INCLUDED IN CORRECTION'
      GO TO 60
    6 CONTINUE
            END IF
C
            IF(NBOUND(IAREA) .GT. 1)THEN
      DO 7ILAYER =1,NLAYER(ISIDE)
      IF(LTYPE(ISIDE,ILAYER) .EQ. IAREA) LAYER=ILAYER
    7 CONTINUE
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      IF(LAYER .EQ. 0) THEN
      WRITE(*,*)'PROBLEM IN SUBROUTINE CORRECT'
      PAUSE
      STOP
      END IF
            END IF
C
      DO 50 INODE=JFF(ISIDE,LAYER),KFF(ISIDE,LAYER)-1
      SEGLEN=ABS(ZFF(ISIDE,INODE+1)-ZFF(ISIDE,INODE))
C     CALCULATE INFLUENCE FUNCTIONS
C     WRITE(10,*)'INODE = ',INODE
      DIST=XFF(ISIDE,LAYER)-XI
      DIST=ABS(DIST)
C     WRITE(10,*)ISRCE,INODE,SEGLEN
      X1=XFF(ISIDE,LAYER)
      X2=XFF(ISIDE,LAYER)
      Z1=ZFF(ISIDE,INODE)
      Z2=ZFF(ISIDE,INODE+1)
      IF (DIST .LT. 0.000001) GO TO 10
C     IF ROTAT .GT. 0. CLOCKWISE , DIST/RADIUS = -
C     IF ROTAT .LT. 0. COUNTERCLOCKWISE, DIST/RADIUS = +
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,ROTAT)
      IF(ROTAT .GT. PI)ROTAT=-1.*(2.*PI-ROTAT)
      IF(ROTAT .GT. 0.) DIST=-1.*DIST
```

```
   10 CONTINUE
      DO 40IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(X2-X1)/2.0+X1
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
      XKSR=FBAR*RADIUS(IGAUSS)/CSRAT(IAREA)
      CALL HANKEL(0,XKSR,H01KSR)
      CALL HANKEL(1,XKSR,H11KSR)
C
      G(IGAUSS)=(0.0,0.25)*H01KSR
C
      T(IGAUSS)=
     & (0.0,-0.25)*FBAR/CSRAT(IAREA)*DIST/RADIUS(IGAUSS)*H11KSR
C
   40 CONTINUE

C     WRITE(80,*)'INODE',INODE
C     CORRECTION TO STRAIN
C
C     3 POINT GAUSSIAN QUADRATURE
      STCORR=
     &   (  (WT3(1)*WTFN3(1)*G(1)+WT3(2)*WTFN3(2)*G(2)
     &   +   WT3(1)*WTFN3(3)*G(3))* FFSXYH(ISIDE,INODE)
     &   +  (WT3(1)*WTFN3(3)*G(1)+WT3(2)*WTFN3(2)*G(2)
     &   +   WT3(1)*WTFN3(1)*G(3))* FFSXYH(ISIDE,INODE+1)  )
     &   *  SEGLEN/2.0
C
C     IN -X REGION, dv/dn = -dv/dx
      IF(ISIDE .EQ. 1)STCORR= -1.0*STCORR
C
C     3 POINT QUADRATURE
C     CORRECTION TO DISPLACEMENT
      VCORR=
     &   (  (WT3(1)*WTFN3(1)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +   WT3(1)*WTFN3(3)*T(3))*FFV(ISIDE,INODE)
     &   +  (WT3(1)*WTFN3(3)*T(1)+WT3(2)*WTFN3(2)*T(2)
     &   +   WT3(1)*WTFN3(1)*T(3))* FFV(ISIDE,INODE+1)  )
     &   *  SEGLEN/2.0
C
      TCORR(ISRCE)=TCORR(ISRCE)-VCORR+STCORR
C     WRITE(80,*)VCORR,STCORR,TCORR(ISRCE)
   50 CONTINUE
   60 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE HSCORR(ISRCE,XI,ZI,IAREA)
C     CALCULATE INTEGRAL OF HANKEL FUNCTION H01KSR
C     BETWEEN 0.0001 AND INFINITY
C23456789112345678921234567893123456789412345678951234567896123456789712
      DOUBLE PRECISION TOLER,PI
      COMPLEX AREA,XKH1
C
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      COMPLEX VCORR,STCORR
C
      TCORR(ISRCE)=(0.0,0.0)
      STCORR=(0.0,0.0)
      VCORR=(0.0,0.0)
C     WRITE(80,*)'ISRCE',ISRCE
      PI=3.1415926535D0
```

```
      TOLER=.0050D0
      XKSH1=FBAR/CSRAT(HALFSP)
      XKH1=COS(ANGLE(IANGLE)/57.29577951)*FBAR/CSRAT(HALFSP)
C
      XMX=COS(ANGLE(IANGLE)/57.29577951)
C
      IF(ANGLE(IANGLE) .LT. 0.0001)THEN
      XMX=0.999999
      END IF
C
      DO 30 ISIDE =1,2
      AREA=(0.0,0.0)
C     WRITE(*,9000)ISIDE
 9000 FORMAT(/5X,'WORKING ON SIDE ',I1)
      A=0.0001
      IF (ISIDE .EQ. 1)THEN
      B= XI-X(KNODE(HALFSP,NBOUND(HALFSP)))
      SIGN = -1.0
      END IF
      IF (ISIDE .EQ. 2)THEN
      B= X(JNODE(HALFSP,NBOUND(HALFSP)))-XI
      SIGN = +1.0
      END IF
C
C     ALPHA = Ks , ZETA = K
      ALPHA=XKSH1
      ZETA=XKH1
C
      CALL NUMINTGR(1,0,ALPHA,ZETA,A,B,AJSINE)
      CALL NUMINTGR(2,0,ALPHA,ZETA,A,B,AJCOS)
      CALL NUMINTGR(3,0,ALPHA,ZETA,A,B,AYSINE)
      CALL NUMINTGR(4,0,ALPHA,ZETA,A,B,AYCOS)
      AREA= (AJCOS-SIGN*AYSINE) + (0.0,1.0)*(AYCOS+SIGN*AJSINE)
      AREA= -AREA
C     INTEGRAL FROM 0 TO INFINITY
C     & +(1.0,0.0)/((FBAR/CSRAT(HALFSP))*SIN(ANGLE(IANGLE)/57.29577951))
     & +(1.0,0.0)/((FBAR/CSRAT(HALFSP))*SQRT(1.0-XMX*XMX) )
     & *(1.0-SIGN* 2.0/PI*(PI/2.-ANGLE(IANGLE)/57.29577951))
      AREA=AREA*CEXP((0.0,1.0)*XKH1*XI)
C     STCORR = i/4*AREA*STRAIN
      STCORR= (0.0,1.0)/4.0*AREA*HSSZYH(ISIDE)
C     APPLY CORRECTION TO FVECT ; STCORR= -STCORR
      STCORR= -1.0*STCORR
      VCORR=(0.0,0.0)
      TCORR(ISRCE)=TCORR(ISRCE)-VCORR+STCORR
C     WRITE(*,*)'ISRCE = ',ISRCE,' AREA  = ',AREA
C     WRITE(*,*)'ISRCE = ',ISRCE,' STCORR  = ',STCORR
C     WRITE(*,*)'ISRCE = ',ISRCE,' TOTAL CORR  = ',TCORR(ISRCE)
   30 CONTINUE
      RETURN
      END



      SUBROUTINE NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      DIMENSION R(5),Y(5)
```

```
      DOUBLE PRECISION AOLD,ATOT,ADELTA,DIFF
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      HDELTA=B-A
C
      AREA=0.0
      IF(ABS(HDELTA) .LT. 0.00001)THEN
      RETURN
      END IF
C
      IF(A .LT. 1.0 .AND. B .GT. 2.0)THEN
      AA=A
      BB=2.0
      NCYCLE =2
      ELSE
      AA=A
      BB=B
      NCYCLE=1
      END IF
      DO 50ICYCLE =1,NCYCLE
      HDELTA =BB-AA
      AOLD=9.99999D20
      TOLER=0.005
   10 CONTINUE
      ATOT=0.0D0
      N=(BB-AA)/HDELTA
C     WRITE(*,*)'N = ',N
      DO 40I=1,N
      AAA=(I-1)*HDELTA+AA
      DO 20J=1,NGAUSS
      R(J)=  AAA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALL YVALUE(IFUNC,NU,ALPHA,ZETA,R(J),Y(J))
   20 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   40 CONTINUE
      DIFF=DABS(ATOT-AOLD)
C     WRITE(*,*)'ATOT = ',ATOT
      IF (DIFF .GT. TOLER ) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 10
      END IF
      AREA=ATOT+AREA
      AA=BB
      BB=B
   50 CONTINUE
C     WRITE(*,*)'AREA = ',AREA
      RETURN
      END

      SUBROUTINE YVALUE(IFUNC,IORDER,ALPHA,ZETA,R,Y)
```

```
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION JN,YN
C
      GO TO (10,20,30,40), IFUNC
   10 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*SIN(ZETA*R)
      GO TO 9999
   20 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*COS(ZETA*R)
      GO TO 9999
   30 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
      Y=YN*SIN(ZETA*R)
      GO TO 9999
   40 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
      Y=YN*COS(ZETA*R)
C
 9999 CONTINUE
      RETURN
      END



      SUBROUTINE JNU(IORDER,XREAL,JN)
C     ********************************************
C     BESSEL FUNCTION (Jn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     ********************************************
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,AA,A,B
      DOUBLE PRECISION JN
C
      Z=XREAL
      PI=3.1415926535D0
      TOLER=.00001D0
C     CALCULATE BESSEL FUNCTION OF 1ST KIND (JN)
      A=0.0D0
      B=PI
```

```
      HDELTA=B-A
      AOLD=9.99999D20
   10 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 20I=1,N
      AA= (I-1)*HDELTA+A
      DO 15J=1,NGAUSS
      X(J)=    AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))
   15 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   20 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 10
      END IF
   25 CONTINUE
      JN= ATOT/PI
      RETURN
      END


      SUBROUTINE YNU(IORDER,XREAL,YN)
C     **********************************************
C     BESSEL FUNCTION (Yn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     **********************************************
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,CALC,AA,A,B
      DOUBLE PRECISION YN
C
C     CALCULATE BESSEL FUNCTION OF 2ND KIND, ORDER N (YN)
      PI=3.1415926535D0
      TOLER=0.00001
      A=0.
      B=PI
      HDELTA=B-A
      AOLD=9.99999D20
   30 CONTINUE
      N=(B-A)/HDELTA
      ATOT=0.0
      DO 50I=1,N
      AA= (I-1)*HDELTA+A
      DO 40J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DSIN(XREAL* DSIN(X(J))-IORDER*X(J))
```

```
   40 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   50 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 30
      END IF
   55 CONTINUE
      YN=ATOT
      A=0.
      B=10.
      HDELTA=1.0
C     HDELTA=B-A
      AOLD=9.99999D20
   60 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 80I=1,N
      AA= (I-1)*HDELTA+A
      DO 70J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALC=XREAL*DSINH(X(J))
      IF(CALC .GT. 40.)CALC=40.
      Y(J)=
     &(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)
   70 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   80 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 60
      END IF
   90 CONTINUE
      YN=(YN-ATOT)/PI
      RETURN
      END
```

```
      SUBROUTINE FIXMAT
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/HMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
      INTEGER ISKIP(6),JAREA(3)
      REAL XN1(6),XN2(6)
      COMPLEX GRATIO
C
C     7891123456789212345678931234567894123456789512345678961234567897l2
C
      KNO=KNODE(NAREA,NBOUND(NAREA))
C     OPEN(UNIT=70,FILE='GH.CHECK',STATUS='UNKNOWN')
C
            IF(NAREA .EQ. 1)THEN
            GO TO 205
            END IF
C
C     NAREA > 1
C
      JMARK=0
      DO 50IAREA =1,NAREA-1
      DO 50INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF (IEQN(INODE) .EQ. 0) GO TO 50
```

```
      JMARK=JMARK+1
      DO 20IROW=1,KNO
      HMAT(IROW,JMARK)=HMAT(IROW,INODE)
   20 CONTINUE
      IF (IEQN(INODE) .GE. 2)THEN
      DO 40 IIAREA=IAREA+1,NAREA
      DO 40 IINODE=JNODE(IIAREA,1),KNODE(IIAREA,NBOUND(IIAREA))
      IF (NCONN(IINODE) .EQ. INODE) THEN
      JJCOL2=IINODE
      DO 30 IROW=1,KNO
      HMAT(IROW,JMARK)=HMAT(IROW,JMARK)+HMAT(IROW,JJCOL2)
   30 CONTINUE
      END IF
   40 CONTINUE
      END IF
   50 CONTINUE
      HDIM=JMARK
C
      IN3=0
      JMARK=0
      DO 200IAREA =1,NAREA-1
      DO 190INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF(IEQN(INODE) .EQ. 0) GO TO 190
C
                  IF(IEQN(INODE) .EQ. 1 .OR. IEQN(INODE) .EQ. 2)THEN
      JMARK=JMARK+1
      DO 60IROW=1,KNO
      GMAT(IROW,JMARK)=GMAT(IROW,JCOL1(INODE))
   60 CONTINUE
C
            IF (IEQN(INODE) .EQ. 2)THEN
        DO 80 IIAREA=IAREA+1,NAREA
        DO 80 IINODE=JNODE(IIAREA,1),KNODE(IIAREA,NBOUND(IIAREA))
          IF (NCONN(IINODE) .EQ. INODE) THEN
          JJCOL1=JCOL1(INODE)
          JJCOL2=JCOL1(IINODE)
          GRATIO=((CSRAT(IAREA)/CSRAT(IIAREA))**2)*
     &     UWTRAT(IAREA)/UWTRAT(IIAREA)
          DO 70 IROW=1,KNO
          GMAT(IROW,JMARK)=GMAT(IROW,JMARK)-GMAT(IROW,JJCOL2)*GRATIO
   70   CONTINUE
          END IF
   80   CONTINUE
            END IF
            END IF
C
C
        IF (IEQN(INODE) .EQ. 3)THEN
        IN3=IN3+1
        DO 115 I=1,3
        IINODE=N3(IN3,I)
        IELEM=2*I-1
        SEGLEN=SQRT((X(IINODE)-X(IINODE-1))**2+
     &  (Z(IINODE)-Z(IINODE-1))**2)
        XN1(IELEM)=(Z(IINODE-1)-Z(IINODE))/SEGLEN
        XN2(IELEM)=(X(IINODE)-X(IINODE-1))/SEGLEN
        SEGLEN=SQRT((X(IINODE+1)-X(IINODE))**2+
     &  (Z(IINODE+1)-Z(IINODE))**2)
```

```
      XN1(IELEM+1)=(Z(IINODE)-Z(IINODE+1))/SEGLEN
      XN2(IELEM+1)=(X(IINODE+1)-X(IINODE))/SEGLEN
115   CONTINUE
      JAREA(1)=IAREA
      DO 130I=2,3
      DO 120 IIAREA=1,NAREA
      DO 120 IINODE=1,KNODE(IIAREA,NBOUND(IIAREA))
      IF(IINODE .EQ. N3(IN3,I))THEN
      JAREA(I)=IIAREA
      GO TO 130
      END IF
120   CONTINUE
130   CONTINUE
      DO 140 IELEM=1,6
      ISKIP(IELEM)=0
140   CONTINUE
      DO180 IELEM=1,5
      IF(ISKIP(IELEM) .EQ. 1)GO TO 180
C
      IF(IELEM .EQ. 1 .OR. IELEM .EQ. 2) THEN
      NODE1=INODE
      IAREA1=JAREA(1)
      END IF
      IF(IELEM .EQ. 3 .OR. IELEM .EQ. 4)THEN
      NODE1=N3(IN3,2)
      IAREA1=JAREA(2)
      END IF
      IF(IELEM .EQ. 5 .OR. IELEM .EQ. 6) THEN
      NODE1=N3(IN3,3)
      IAREA1=JAREA(3)
      END IF
C
C     DETERMINE IF ELEMENT IELEM IS EVEN (MARK1 = 0) OR ODD(MARK1=1)
C    IF IELEM IS EVEN, JJCOL=JCOL1(IINODE)
C    IF IELEM IS ODD,  JJCOL=JCOL1(IINODE)-1
C
      TEMP=IELEM
      XREAL =TEMP/2.
      I=IELEM/2
      DIFF=ABS(XREAL-I)
      MARK1=1
      IF(DIFF .LT. 0.01)MARK1=0
C
     JMARK=JMARK+1
     DO 150 IROW=1,KNO
     JJCOL=JCOL1(NODE1)-MARK1
     GMAT(IROW,JMARK)=GMAT(IROW,JJCOL)
150  CONTINUE
C
      DO 170 IIELEM=IELEM+1,6
      DIFF=ABS(ABS(XN1(IELEM))-ABS(XN1(IIELEM)))
        IF (DIFF .LT. 0.01)THEN
C    ELEMENTS ARE PARALLEL
        DIFF1=ABS(XN1(IELEM)-XN1(IIELEM))
        DIFF2=ABS(XN2(IELEM)-XN2(IIELEM))
        IF(DIFF1 .LT. 0.01 .AND. DIFF2 .LT. 0.01) THEN
C    ELEMENTS ARE PARALLEL WITH SAME NORMALS
        SIGN =+1.0
```

```
          ELSE
C     ELEMENTS ARE PARALLEL WITH OPPOSITE NORMALS
          SIGN = -1.0
          END IF
          ISKIP(IIELEM)=1
C         DETERMINE IF ELEMENT IIELEM IS EVEN (MARK2 = 0) OR
ODD(MARK2=1)
C     IF IIELEM IS EVEN, JJCOL=JCOL1(IINODE)
C     IF IIELEM IS ODD,  JJCOL=JCOL1(IINODE)-1
C
          TEMP=IIELEM
          XREAL =TEMP/2.
          I=IIELEM/2
          DIFF=ABS(XREAL-I)
          MARK2=1
          IF(DIFF .LT. 0.01)MARK2=0
C
C         DETERMINE AREA & NODE FOR PARALLEL ELEMENT
C         IN3 = NODE 3 INTERFACE THAT WE'RE WORKING ON
C
C
            IF(IIELEM .EQ. 2) THEN
            NODE2=INODE
            IAREA2=JAREA(1)
            END IF
            IF(IIELEM .EQ. 3 .OR. IIELEM .EQ. 4)THEN
            NODE2=N3(IN3,2)
            IAREA2=JAREA(2)
            END IF
            IF(IIELEM .EQ. 5 .OR. IIELEM .EQ. 6) THEN
            NODE2=N3(IN3,3)
            IAREA2=JAREA(3)
            END IF
          JJCOL1=JCOL1(NODE1)-MARK1
          JJCOL2=JCOL1(NODE2)-MARK2
          GRATIO=((CSRAT(IAREA1)/CSRAT(IAREA2))**2)*
     &    UWTRAT(IAREA1)/UWTRAT(IAREA2)
          DO 160 IROW=JNODE(IAREA2,1),KNODE(IAREA2,NBOUND(IAREA2))
          GMAT(IROW,JMARK)=GMAT(IROW,JMARK)+
     &           SIGN*GMAT(IROW,JJCOL2)*GRATIO
  160 CONTINUE
          END IF
  170   CONTINUE
  180   CONTINUE
        END IF
  190 CONTINUE
  200 CONTINUE
      GDIM=JMARK
  205 CONTINUE
C     WRITE(70,7000)
 7000 FORMAT(5X,'HMAT'/)
C     WRITE(70,*)'HDIM',HDIM
      JMAX=HDIM
C     WRITE(70,*)'ORDER',(NORDER(J),J=1,JMAX)
      DO 240I=1,KNODE(NAREA,NBOUND(NAREA))
C     WRITE(70,7005)I
 7005 FORMAT(I4)
C     WRITE(70,7010)(HMAT(I,J),J=1,HDIM)
```

```
 7010 FORMAT(4F12.6)
  240 CONTINUE
C     WRITE(70,7020)
 7020 FORMAT(/5X,'GMAT'/)
C     WRITE(70,*)'GDIM',GDIM
      DO 250I=1,KNODE(NAREA,NBOUND(NAREA))
C     WRITE(70,7005)I
C     WRITE(70,7010)(GMAT(I,J),J=1,GDIM)
  250 CONTINUE
C     ESTABLISH NEW JCOL1
      II=0
      DO 270 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(NCONN(INODE) .EQ. 0)THEN
C     NEXT NODE IN NORDER
      II=II+1
      JCOL1(INODE)=II
      IF(IEQN(INODE) .EQ. 3)II=II+1
      END IF
  270 CONTINUE
      NNODE=II-NN3
C     WRITE(70,7030)
 7030 FORMAT(/5X,'NODE,JCOL1')
      DO 280I=1,NNODE
      INODE=NORDER(I)
C     WRITE(70,7060)INODE,JCOL1(INODE)
 7060 FORMAT(2I5)
  280 CONTINUE
C     CLOSE(UNIT=70)
      RETURN
      END
```

```
      SUBROUTINE SOLVE(MARK)
C
C     MARK=1; FREE-FIELD CALCULATION
C     MARK=2; BEM CALCULATION
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/AMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX AMAT,GMAT,FVECT,XVECT
C
      DOUBLE PRECISION A(1440,1441),X(1440),PIVOT,TEMP,ANORM
      INTEGER IX(1440)
C
C     ******************************************************
C     ******************************************************
C     *****      SUBROUTINE FINDS THE SOLUTION OF      *****
C     *****      SIMULTANEOUS EQUATIONS USING          *****
C     *****      MAXIMUM PIVOT PROCEDURE               *****
C     ******************************************************
C     ******************************************************
C     II REPRESENTS THE REAL PART, II+1 REPRESENTS THE IMAGINARY PART
C     JJ REPRESENTS THE REAL PART, JJ+1 REPRESENTS THE IMAGINARY PART
C
            IF (MARK .EQ. 1)THEN
      N=2.*FFDIM
      NROW=FFDIM
      DO 10I=1,FFDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 10J=1,FFDIM
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(STIFF(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(STIFF(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(STIFF(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(STIFF(I,J))
   10 CONTINUE
            END IF
C
            IF (MARK .EQ. 2)THEN
      N=2.*NDIM
      NROW=NDIM
      DO 15I=1,NDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 15J=1,NDIM
```

```
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(AMAT(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(AMAT(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(AMAT(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(AMAT(I,J))
   15 CONTINUE
            END IF
C
C     OPEN(UNIT=20,FILE='MAT.OUT',STATUS='UNKNOWN')
C     WRITE(20,2000)ANAME
 2000 FORMAT(//5X,A80)
C     WRITE(20,2010)
 2010 FORMAT(/10X,'A matrix * X vector = C vector')
C     WRITE(20,2020)
 2020 FORMAT( 5X,'A matrix:')
      DO 20I=1,N
C     WRITE(20,2030)(A(I,J),J=1,N)
 2030 FORMAT(5X,8F10.4)
   20 CONTINUE
C     WRITE(20,2040)
 2040 FORMAT(/5X,'C vector:')
C     WRITE(20,2030)(A(J,N+1),J=1,N)
C
C     *****************************************************
C     *****************************************************
C     *SOLVE FOR X VECTOR USING GAUSS'S ELIMINATION METHOD *
C     ************   MAXIMUM PIVOT IS USED ****************
C     *****************************************************
C     *****************************************************
C
C
C
      DO 30I=1,N
      IX(I)=I
   30 CONTINUE
      DO 100IROW=1,N
      I=IROW/10
      XX=IROW
      XX=XX/10.
      DIFF=ABS(XX-I)
      IF(DIFF .LT. 0.0001)THEN
      WRITE(*,9000)IROW,N
 9000 FORMAT(5X,'WORKING ON COLUMN # ',I4,' / ',I4)
      END IF
      JCOL=IROW
      PIVOT=0.0D0
C     FIND PIVOT VALUE
      DO 40IIROW=IROW,N
      DO 40JJCOL=JCOL,N
      IF (ABS(A(IIROW,JJCOL)) .GT. ABS (PIVOT)) THEN
      PIVOT =A(IIROW,JJCOL)
      IMARK=IIROW
      JMARK=JJCOL
      END IF
   40 CONTINUE
      IF (ABS(PIVOT) .LT. 1.0D-18) THEN
      WRITE(*,9010)
 9010 FORMAT(////15X,'** MATRIX DOES NOT HAVE AN INVERSE,')
```

```
      WRITE(*,9020)
 9020 FORMAT(15X,'PUSH RETURN  **')
      PAUSE
      STOP
      END IF
C     INTERCHANGE ROWS
      DO 50 JJCOL=1,N+1
      TEMP=A(IROW,JJCOL)
      A(IROW,JJCOL)=A(IMARK,JJCOL)
      A(IMARK,JJCOL)=TEMP
   50 CONTINUE
C     INTERCHANGE COLUMNS
      ITEMP=IX(JCOL)
      IX(JCOL)=IX(JMARK)
      IX(JMARK)=ITEMP
      DO 60 IICOL=1,N
      TEMP=A(IICOL,JCOL)
      A(IICOL,JCOL)=A(IICOL,JMARK)
      A(IICOL,JMARK)=TEMP
   60 CONTINUE
C     WRITE(20,*)'IROW = ',IROW,'†PIVOT = ',PIVOT
C     DO 65 I=1,N
C     WRITE(20,2050)(A(I,J),J=1,N+1)
C   65 CONTINUE
C     NORMALIZE ROW OF PIVOT ELEMENT
      DO 70 JJCOL=JCOL,N+1
      A(IROW,JJCOL)=A(IROW,JJCOL)/PIVOT
   70 CONTINUE
C     SUBTRACT ROW FROM REMAINING ROWS
      DO 80 IIROW=IROW+1,N
      ANORM=-1.0D0*A(IIROW,JCOL)
      DO 80 JJCOL=JCOL,N+1
      A(IIROW,JJCOL)=A(IIROW,JJCOL)+ANORM *A(IROW,JJCOL)
   80 CONTINUE
C     WRITE(20,*)'IROW = ',IROW,'  PIVOT = ',PIVOT
C     DO 85 I=1,N
C     WRITE(20,2050)(A(I,J),J=1,N+1)
C 2050 FORMAT(5F10.3)
C   85 CONTINUE
  100 CONTINUE
C     USE BACKWARD SUBSTITUTION
      X(N)=A(N,N+1)/A(N,N)
      DO 120IROW=N-1,1,-1
      SUM=0.0
      DO 110 JCOL=IROW+1,N
      SUM=SUM+A(IROW,JCOL)*X(JCOL)
  110 CONTINUE
      X(IROW)=(A(IROW,N+1)-SUM)/A(IROW,IROW)
  120 CONTINUE
C     REORDER X VECTOR
      DO 130 IROW=1,N
      A(IROW,N+1)=X(IROW)
  130 CONTINUE
      DO 140 I=1,N
      IROW=IX(I)
      X(IROW)=A(I,N+1)
  140 CONTINUE
C
```

```
C      WRITE(20,2060)
 2060 FORMAT(//5X,'X vector:')
C       WRITE(20,2070)(X(J),J=1,N)
 2070 FORMAT(5X,8F10.4)
C      CLOSE (UNIT=20)
       DO 150J=1,NROW
       JJ=2*J-1
       XVECT(J)=X(JJ)+(0.0,1.0)*X(JJ+1)
  150 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE SORT
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(730)
      COMMON/CMATRIX/AMAT(720,720),GMAT(720,730),
     & FVECT(720),XVECT(720)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX AMAT,GMAT,FVECT,XVECT
C
C     OUTPUT DATA
      COMMON/COUT/VDISPL(720),SNYH1(720,2)
      COMPLEX VDISPL,SNYH1
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
      INTEGER ISKIP(6),JAREA(3)
      REAL XN1(6),XN2(6)
C
      COMPLEX GRATIO
C
C23456789112345678921234567893123456789412345678951234567896123456789712
      DO 10I=1,HDIM
      IROW=I
      INODE=NORDER(I)
      VDISPL(INODE)=XVECT(IROW)
   10 CONTINUE
      DO 20 IINODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF (IEQN(IINODE) .EQ. 0)THEN
      INODE=NCONN(IINODE)
      VDISPL(IINODE)=VDISPL(INODE)
      END IF
   20 CONTINUE
C
C
      ICOUNT= HDIM
      IN3=0
C
      DO 150 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(NSURF(INODE) .EQ. 1 .AND. IEQN(INODE) .EQ. 1)THEN
```

```
      SNYH1(INODE,1)=0.0
      END IF
      IF(NSURF(INODE) .EQ. 0 .AND. IEQN(INODE) .EQ. 1)THEN
      ICOUNT=ICOUNT+1
      IROW=ICOUNT
      SNYH1(INODE,1)=XVECT(IROW)
      END IF
      IF(IEQN(INODE) .EQ. 2)THEN
      ICOUNT=ICOUNT+1
      IROW=ICOUNT
      SNYH1(INODE,1)=XVECT(IROW)
      DO 30IINODE=INODE,KNODE(NAREA,NBOUND(NAREA))
      IF(NCONN(IINODE) .EQ. INODE) THEN
      NODE2=IINODE
      GO TO 40
      END IF
   30 CONTINUE
   40 CONTINUE
      DO 50 IAREA=1,NAREA
      DO 50 IINODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF(IINODE .EQ. INODE)JAREA(1)=IAREA
      IF(IINODE .EQ. NODE2)JAREA(2)=IAREA
   50 CONTINUE
      GRATIO=((CSRAT(JAREA(1))/CSRAT(JAREA(2)))**2)*
     & UWTRAT(JAREA(1))/UWTRAT(JAREA(2))
      SNYH1(NODE2,1)= -1.*SNYH1(INODE,1)*GRATIO
      END IF
      IF(IEQN(INODE) .EQ. 3)THEN
      IN3=IN3+1
      DO 60 I=1,3
      IINODE=N3(IN3,I)
      IELEM=2*I-1
        SEGLEN=SQRT((X(IINODE)-X(IINODE-1))**2+
     &  (Z(IINODE)-Z(IINODE-1))**2)
        XN1(IELEM)=(Z(IINODE-1)-Z(IINODE))/SEGLEN
        XN2(IELEM)=(X(IINODE)-X(IINODE-1))/SEGLEN
        SEGLEN=SQRT((X(IINODE+1)-X(IINODE))**2+
     &  (Z(IINODE+1)-Z(IINODE))**2)
        XN1(IELEM+1)=(Z(IINODE)-Z(IINODE+1))/SEGLEN
        XN2(IELEM+1)=(X(IINODE+1)-X(IINODE))/SEGLEN
   60 CONTINUE
      DO 80I=1,3
      DO 70 IAREA=1,NAREA
      DO 70 IINODE=1,KNODE(IAREA,NBOUND(IAREA))
      IF(IINODE .EQ. N3(IN3,I))THEN
      JAREA(I)=IAREA
      GO TO 80
      END IF
   70 CONTINUE
   80 CONTINUE
      DO 90 IELEM=1,6
      ISKIP(IELEM)=0
   90 CONTINUE
      DO 110 IELEM=1,5
      IF(ISKIP(IELEM) .EQ. 1)GO TO 110
C
C     DETERMINE IF ELEMENT IELEM IS EVEN (MARK1 = 2) OR ODD(MARK1=1)
      TEMP=IELEM
```

```
      XREAL =TEMP/2.
      I=IELEM/2
      DIFF=ABS(XREAL-I)
      MARK1=1
      IF(DIFF .LT. 0.01)MARK1=2
C
      IF(IELEM .EQ. 1 .OR. IELEM .EQ. 2) THEN
      NODE1=INODE
      IAREA1=JAREA(1)
      END IF
      IF(IELEM .EQ. 3 .OR. IELEM .EQ. 4)THEN
      NODE1=N3(IN3,2)
      IAREA1=JAREA(2)
      END IF
      IF(IELEM .EQ. 5 .OR. IELEM .EQ. 6) THEN
      NODE1=N3(IN3,3)
      IAREA1=JAREA(3)
      END IF
      ICOUNT=ICOUNT+1
      IROW=ICOUNT
      SNYH1(NODE1,MARK1)=XVECT(IROW)
C
      DO 100 IIELEM=IELEM+1,6
        DIFF=ABS(ABS(XN1(IELEM))-ABS(XN1(IIELEM)))
          IF (DIFF .LT. 0.01)THEN
          DIFF1=ABS(XN1(IELEM)-XN1(IIELEM))
          DIFF2=ABS(XN2(IELEM)-XN2(IIELEM))
          IF(DIFF1 .LT. 0.01 .AND. DIFF2 .LT. 0.01) THEN
          SIGN =+1.0
          ELSE
          SIGN = -1.0
          END IF
      ISKIP(IIELEM)=1
C     DETERMINE AREA & NODE FOR PARALLEL ELEMENT
C     IN3 = NODE 3 INTERFACE THAT WE'RE WORKING ON
C
C
C     DETERMINE IF ELEMENT IIELEM IS EVEN (MARK2 = 2) OR ODD(MARK2=1)
      TEMP=IIELEM
      XREAL =TEMP/2.
      I=IIELEM/2
      DIFF=ABS(XREAL-I)
      MARK2=1
      IF(DIFF .LT. 0.01)MARK2=2
C
      IF(IIELEM .EQ. 2) THEN
      NODE2=INODE
      IAREA2=JAREA(1)
      END IF
      IF(IIELEM .EQ. 3 .OR. IIELEM .EQ. 4)THEN
      NODE2=N3(IN3,2)
      IAREA2=JAREA(2)
      END IF
      IF(IIELEM .EQ. 5 .OR. IIELEM .EQ. 6) THEN
      NODE2=N3(IN3,3)
      IAREA2=JAREA(3)
      END IF
      GRATIO=((CSRAT(IAREA1)/CSRAT(IAREA2))**2)*
```

```
   & UWTRAT(IAREA1)/UWTRAT(IAREA2)
    SNYH1(NODE2,MARK2)=SNYH1(NODE1,MARK1)*GRATIO*SIGN
    END IF
100 CONTINUE
110 CONTINUE
    END IF
150 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE OUTPUT(ANAME)
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(720),NORDER(720),NNSURF,NSURF(720),N3(2,3),NN3
      COMMON/RGEOM/X(720),Z(720)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,VINCDT(130)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,VINCDT
      CHARACTER*2 ATYPE
C
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,100),ZFF(2,100)
      COMMON/CFFLD/FFV(2,100),TCORR(720),HSV(2),FFSXYH(2,100),
     & HSSZYH(2),STIFF(10,10)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFV,TCORR,HSV,FFSXYH,HSSZYH,STIFF
C
C     OUTPUT DATA
      COMMON/COUT/VDISPL(720),SNYH1(720,2)
      COMPLEX VDISPL,SNYH1
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(720)
      INTEGER IEQN
C
      CHARACTER*80 ANAME
C
      WRITE(60,6000)
 6000 FORMAT(//5X,'OUTPUT')
      WRITE(60,6001) ANAME
 6001 FORMAT(/5X,A80)
      WRITE(60,6002)ATYPE
 6002 FORMAT(5X,'TYPE OF WAVE = ',A2)
      X1=FBAR*CSQRT(1.0-(0.0,2.0)*BETA(1))
      WRITE(60,6003)X1
 6003 FORMAT(5X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F6.3)
```

```
      WRITE(60,6004)IANGLE,ANGLE(IANGLE)
 6004 FORMAT(5X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
C
      WRITE(60,6005)NAREA
 6005 FORMAT(5X,'NUMBER OF HOMOGENEOUS REGIONS = ',I1)
      WRITE(60,6006)
 6006 FORMAT(/5X,'BOUNDARY VALUES')
      DO 60IAREA=1,NAREA
      WRITE(60,6010)IAREA
 6010 FORMAT(//5X,'HOMOGENEOUS REGION ',I1)
      WRITE(60,6012)NBOUND(IAREA)
 6012 FORMAT(5X,'NUMBER OF BOUNDARIES = ',I1)
      DO 60IBOUND=1,NBOUND(IAREA)
      WRITE(60,6015)IBOUND
 6015 FORMAT(/5X,'BOUNDARY = ',I2)
      NNODE=KNODE(IAREA,IBOUND)-JNODE(IAREA,IBOUND)+1
      WRITE(60,6017)NNODE
 6017 FORMAT(5X,'NUMBER OF NODES = ',I3)
      WRITE(60,6018)
 6018 FORMAT(/2X,'DISPLACEMENTS')
      WRITE(60,6020)
 6020 FORMAT(/T3,'NODE',T13,'X/H1',T23,'Z/H1',T42,'V DISPL',T64,'|V|')
      DO 30 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      XREAL=REAL(VDISPL(INODE))
      XIMAG=AIMAG(VDISPL(INODE))
      AMPL=SQRT((XREAL**2)+(XIMAG**2))
      WRITE(60,6030)INODE,X(INODE),Z(INODE),VDISPL(INODE),AMPL
 6030 FORMAT(T4,I3,T11,F8.3,T21,F8.3,T37,2F8.3,T61,F8.3)
   30 CONTINUE
              IF(NAREA .GT. 1)THEN
      WRITE(60,6038)
 6038 FORMAT(/2X,'TRACTIONS')
      WRITE(60,6040)
 6040 FORMAT(/T3,'NODE',T18,'TAU(n,y)*H1/Gi')
      DO 55 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
C
      SNYH1(INODE,1)=SNYH1(INODE,1)*(1.0-(0.0,2.0)*BETA(IAREA))
C
      WRITE(60,6050)INODE,SNYH1(INODE,1)
 6050 FORMAT(T4,I3,T15,2F10.4)
      DO 50 IN3=1,NN3
      DO 40 J=1,3
      IF(N3(IN3,J) .EQ. INODE)THEN
C
      SNYH1(INODE,2)=SNYH1(INODE,2)*(1.0-(0.0,2.0)*BETA(IAREA))
C
      WRITE(60,6060)SNYH1(INODE,2)
 6060 FORMAT(T15,2F10.4)
      GO TO 55
      END IF
   40 CONTINUE
   50 CONTINUE
   55 CONTINUE
              END IF
   60 CONTINUE
      RETURN
      END
```

A.1.2: BOUNDARY ELEMENT CODE

psvconvo

```
      PROGRAM PSVMAIN
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
            COMPLEX HMAT,GMAT,FVECT,XVECT
C
C
C     OUTPUT DATA
      COMMON/COUT/UDISPL(390),WDISPL(390),PXH1(390,2),PZH1(390,2)
      COMPLEX UDISPL,WDISPL,PXH1,PZH1
C
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
C     HALF-SPACE INTEGRALS
      COMMON/RHSINT/ASTART
```

```
      COMMON/CHSINT/H0EXP(2,2),H1EXP(2,2),H2EXP(2,2),H2EXPR(2,2),
     & EXPR2(2),EXPR3(2)
      REAL ASTART
      COMPLEX H0EXP,H1EXP,H2EXP,H2EXPR,EXPR2,EXPR3
C
C
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
      COMMON/KHLFSP/XKH1,XKPH1,XKSH1
      REAL XKH1,XKPH1,XKSH1
C
      COMPLEX XIKSH,XIKTH,XIKX
      CHARACTER*80 ANAME
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
      CHARACTER*20 ARUN
C
      DATA NGAUSS /5/
      DATA (WT(I),I=1,3)/0.236926885056189D0,0.478628670499366D0,
     & 0.568888888888889D0/
      DATA (XX(I),I=1,3)/-0.906179845938664D0,-0.538469310105683D0,
     & 0.0D0/
C
      DATA NGAUS3 /3/
      DATA (WT3(I),I=1,2)/0.555555555555556D0,0.888888888888888889D0/
      DATA (XX3(I),I=1,3)
     & /-0.77459666924148D0,0.0D0,+0.77459666924148D0/
C
      XX(4)= -1.0D0*XX(2)
      XX(5)= -1.0D0*XX(1)
      DO IGAUSS= 1,2
      WTFN(IGAUSS)= (1.0D0 - XX(IGAUSS))/2.0D0
      WTFN(NGAUSS-IGAUSS+1)= 1.0D0-WTFN(IGAUSS)
      END DO
      WTFN(3)= 0.50D0
C
      WTFN3(1)= (1.0D0 - XX3(1))/2.0D0
      WTFN3(3)= 1.0D0-WTFN3(1)
      WTFN3(2)= 0.50D0
C
C     OPEN(UNIT=10,FILE='CHECK.BEM',STATUS='UNKNOWN')
      WRITE(*,9000)
 9000 FORMAT(5X,'INPUT NAME OF RUNFILE')
      READ(*,9010)ARUN
 9010 FORMAT(A20)
      OPEN(UNIT=15,FILE=ARUN,STATUS='UNKNOWN')
```

```
      READ(15,*)NPROB
      DO 9999 IPROB=1,NPROB
      WRITE(*,9020)IPROB
 9020 FORMAT(/5X,'WORKING ON PROBLEM ',I2)
C
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON INPUT'
      WRITE(*,*)' '
      CALL INPUT(ANAME)
      DO 100 IANGLE=1,NANGLE
      WRITE(60,6000) ANAME
      WRITE(*,6000) ANAME
 6000 FORMAT(//2X,A80)
      WRITE(60,6010)ATYPE
      WRITE(*,6010)ATYPE
 6010 FORMAT(/2X,'TYPE OF WAVE = ',A2)
      X1=FBAR*CSQRT(1.0-(0.0,2.0)*BETA(1))
      WRITE(60,6020)X1
      WRITE(*,6020)X1
 6020 FORMAT(2X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F6.3)
      WRITE(60,6030)IANGLE,ANGLE(IANGLE)
      WRITE(*,6030)IANGLE,ANGLE(IANGLE)
 6030 FORMAT(2X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
C     CALC. INCIDENT WAVE DISPLACEMENTS
      IF(ATYPE .EQ. 'P')THEN
      XLX=COS(ANGLE(IANGLE)/57.29577951)
      AP=1.0
      ASV=0.0
      RCPCS= SQRT((2.*(1.-POISS(HALFSP)))/(1.-2.*POISS(HALFSP)))
      XMX=XLX/RCPCS
      END IF
      IF(ATYPE .EQ. 'SV')THEN
      XMX=COS(ANGLE(IANGLE)/57.29577951)
      AP=0.0
      ASV=1.0
      END IF
      DO 10 INODE =JNODE(HALFSP,1),KNODE(HALFSP,NBOUND(HALFSP))
      IINODE=INODE-JNODE(HALFSP,1)+1
      IF (ATYPE .EQ. 'P')THEN
      XIKSH=(0.0,1.0)*FBAR/CSRAT(HALFSP)*Z(INODE)*SQRT(1.-XLX*XLX)/
     & RCPCS
      UINCDT(IINODE)= XLX*AP*CEXP(-XIKSH)
      WINCDT(IINODE)= -SQRT(1.-XLX*XLX)*AP*CEXP(-XIKSH)
      END IF
       IF (ATYPE .EQ. 'SV')THEN
      XIKTH=(0.0,1.0)*FBAR/CSRAT(HALFSP)*Z(INODE)*SQRT(1.-XMX*XMX)
      UINCDT(IINODE)= -SQRT(1.-XMX*XMX)*ASV*CEXP(-XIKTH)
      WINCDT(IINODE)= -XMX*ASV*CEXP(-XIKTH)
      END IF
      XIKX= (0.0,1.0)*FBAR*X(INODE)*XMX/CSRAT(HALFSP)
      UINCDT(IINODE)=UINCDT(IINODE)*CEXP(XIKX)
      WINCDT(IINODE)=WINCDT(IINODE)*CEXP(XIKX)
   10 CONTINUE
      WRITE(60,6040)
 6040 FORMAT(///2X,'INCIDENT WAVE DISPLACEMENTS ')
      WRITE(60,6050)
 6050 FORMAT(T5,'HALF-SPACE',T18,'NODE',T34,
```

```
     & 'INCDT U DISPL',T54,'INCDT W DISPL'/)
      DO 20INODE= JNODE(HALFSP,1),KNODE(HALFSP,1)
      IINODE=INODE-JNODE(HALFSP,1)+1
      WRITE(60,6060)INODE,UINCDT(IINODE),WINCDT(IINODE)
 6060 FORMAT(T18,I3,T33,2F7.3,T53,2F7.3)
   20 CONTINUE
C
      IF(NAREA .EQ. 1 .AND. ICLOSE(1,1) .EQ. 1)GO TO 40
C     CYLINDRICAL CAVITY IN FULL SPACE
      IF(NNSURF .NE. 0)THEN
      IXMAX=1
      IF(NAREA .GT. 1)IXMAX=2
      WRITE(*,*)'WORKING ON FREE-FIELD MOTION'
      DO 30IX=1,IXMAX
      WRITE(*,*)'SIDE = ',IX
      CALL FFLD(IX)
   30 CONTINUE
      END IF
   40 CONTINUE
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON MATRIX'
      WRITE(*,*)' '
      CALL MATRIX
      WRITE(*,*)' '
      WRITE(*,*)'WORKING ON SOLVING MATRIX'
      WRITE(*,*)' '
      CALL SOLVE(2)
      WRITE(*,*)' '
      WRITE(*,*)'SORTING MATRICES'
      WRITE(*,*)' '
      CALL SORT
      WRITE(*,*)' '
      WRITE(*,*)'OUTPUT'
      CALL OUTPUT(ANAME)
  100 CONTINUE
      CLOSE(UNIT=60)
 9999 CONTINUE
      CLOSE(UNIT=15)
C     CLOSE(UNIT=10)
      STOP
      END
```

```
      SUBROUTINE INPUT(ANAME)
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
      INTEGER NELEM(4,2),NSURF1(3,3),NSURF2(3,3),NBND(3)
      REAL XENDPT(70),ZENDPT(70)
      CHARACTER*25 AINPT,AOUT
      CHARACTER*80 ANAME
C
C
C
      7891123456789212345678931234567894123456789512345678961234567897l2

      PI=4.*ATAN(1.0)
C
      READ(15,1500)AINPT
 1500 FORMAT(A25)
      WRITE(*,9000)AINPT
 9000 FORMAT(/2X,'WORKING ON INPUT FILE: ',A25/)
```

```
C
C     WRITE(*,9000)
C 9000 FORMAT(5X,'NAME OF INPUT FILE')
C     READ(*,9010) AINPT
C 9010 FORMAT(A20)
      OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
      READ(50,5000) AOUT
 5000 FORMAT(A25)
      READ(50,5010) ANAME
 5010 FORMAT(A80)
C     TYPE OF INCIDENT WAVE (P or SV)
      READ(50,5020)ATYPE
 5020 FORMAT(A2)
C     ANGLE OF INCIDENCE WITH HORIZONTAL (degrees)')
      READ(50,*)NANGLE,(ANGLE(I),I=1,NANGLE)
C     DIMENSIONLESS FREQUENCY (omega*H1/Cs1)')
      READ(50,*)X1
      FBAR=(1.0,0.0)*X1
C     # OF HOMOGENEOUS REGIONS
      READ(50,*)NAREA
C
C     INITIALIZE JSTART
      JSTART=0
C
      DO 50IAREA=1,NAREA
C     INPUT SOIL SHEAR VELOCITY RATIO Cs (area i)/Cs(area 1)
      READ(50,*)X1
      CSRAT(IAREA)=X1*(1.00,0.0)
C     INPUT POISSON'S RATIO
      READ(50,*)POISS(IAREA)
C     INPUT UNIT WEIGHT RATIO WT(layer i)/WT(layer 1)
      READ(50,*)UWTRAT(IAREA)
C     INPUT DAMPING (%)
      READ(50,*)BETA(IAREA)
      BETA(IAREA)=BETA(IAREA)/100.0
C     INPUT GEOMETRY DATA
      READ(50,*)NBOUND(IAREA)
C
      DO 40 IBOUND=1,NBOUND(IAREA)
      NELEM(IAREA,IBOUND)=0
      READ(50,*)NENDPT
C     STARTING NODE NUMBER
      JSTART=JSTART + 1
      JNODE(IAREA,IBOUND)=JSTART
      JNO= JNODE(IAREA,IBOUND)
      DO 30IENDPT=1,NENDPT-1
C     INPUT END POINTS THAT DEFINE BOUNDARY
      READ(50,*)XENDPT(IENDPT),ZENDPT(IENDPT),
     &  XENDPT(IENDPT+1),ZENDPT(IENDPT+1)
C     INPUT NUMBER OF SEGMENTS TO DIVIDE EACH PORTION OF BOUNDARY INTO
      DIST = SQRT((XENDPT(IENDPT+1)-XENDPT(IENDPT))**2+
     & (ZENDPT(IENDPT+1)-ZENDPT(IENDPT))**2)
C
C     CHECK IF CIRCULAR BOUNDARY
      IF(DIST .LT. 0.00001)THEN
C     CIRCULAR BOUNDARY
      ICLOSE(IAREA,IBOUND)=1
C     INPUT CENTER OF CIRCLE
```

```
      READ(50,*)X0,Z0
C     INPUT DIRECTION OF ROTATION +1=CCW, -1= CW AND NUMBER OF SEGMENTS
      READ(50,*)IROTAT,NSEG
      X1=XENDPT(1)
      Z1=ZENDPT(1)
      R=SQRT((X1-X0)**2+(Z1-Z0)**2)
C     DETERMINE ALPHA, ANGLE FOR Z1 & Z1
C
      IF(X1 .EQ. 0)THEN
      IF(Z1 .GE. 0)ALPHA=PI/2.
      IF(Z1 .LT. 0)ALPHA=3.*PI/2.
C
      ELSE
C
      ALPHA=ATAN((Z1-Z0)/(X1-X0))
      ALPHA=ABS(ALPHA)
      IF(X1 .LT. 0. .AND. Z1 .GE. 0)ALPHA=PI-ALPHA
      IF(X1 .LT. 0. .AND. Z1 .LT. 0)ALPHA=PI+ALPHA
      IF(X1 .GE. 0. .AND. Z1 .LT. 0)ALPHA=2.*PI-ALPHA
      END IF
C
      X(JNO)=X1
      Z(JNO)=Z1
      DTHETA=2.*PI/NSEG*IROTAT
      DO 10ISEG=1,NSEG-1
      INODE=JNO+ISEG
      THETA=ISEG*DTHETA
      X(INODE)=X0+R*COS(THETA+ALPHA)
      Z(INODE)=Z0+R*SIN(THETA+ALPHA)
   10 CONTINUE
      KNODE(IAREA,IBOUND)=JNO+NSEG-1
      JSTART=KNODE(IAREA,IBOUND)
      NELEM(IAREA,IBOUND)= NSEG
      GO TO 40
      END IF
C
C
      READ(50,*)NSEG
      ZDELTA=ZENDPT(IENDPT+1)-ZENDPT(IENDPT)
      XDELTA=XENDPT(IENDPT+1)-XENDPT(IENDPT)
      JSTOP=JSTART+NSEG
      DO 20INODE=JSTART,JSTOP
      X(INODE)=XENDPT(IENDPT)+XDELTA/NSEG*(INODE-JSTART)
      Z(INODE)=ZENDPT(IENDPT)+ZDELTA/NSEG*(INODE-JSTART)
   20 CONTINUE
      JSTART=JSTOP
      NELEM(IAREA,IBOUND)=NELEM(IAREA,IBOUND)+NSEG
   30 CONTINUE
      DIST=SQRT((X(JSTOP)-X(JNO))**2+(Z(JSTOP)-Z(JNO))**2)
      ICLOSE(IAREA,IBOUND)=0
      IF(DIST .LT. 0.00001) THEN
      ICLOSE(IAREA,IBOUND)=1
      JSTOP=JSTOP-1
      JSTART=JSTOP
      END IF
      KNODE(IAREA,IBOUND)=JSTOP
   40 CONTINUE
   50 CONTINUE
```

```
      DO 60INODE=1,KNODE(NAREA,NBOUND(NAREA))
      NSURF(INODE)=0
   60 CONTINUE
      NNSURF=0
C     NUMBER OF AREAS BORDERING  SURFACE & SURFACE BOUNDARIES PER
BORDERING SURF.
      READ(50,*)NSURFA
      IF(NSURFA .NE. 0) THEN
      READ(50,*)(NBND(I),I=1,NSURFA)
      DO 67 ISURFA =1,NSURFA
C     SURFACE NODES
      DO 67IBND=1,NBND(ISURFA)
      READ(50,*)NSURF1(ISURFA,IBND), NSURF2(ISURFA,IBND)
      DO 65 INODE=NSURF1(ISURFA,IBND),NSURF2(ISURFA,IBND)
      NSURF(INODE)=1
   65 CONTINUE
      NNSURF=NNSURF+NSURF2(ISURFA,IBND)-NSURF1(ISURFA,IBND)+1
   67 CONTINUE
      END IF
C     HALF-SPACE REGION
      READ(50,*)HALFSP
C
      NLAYER(1)=0
      NLAYER(2)=0
C
      IF(NAREA .GT. 1 .AND. NNSURF .NE. 0)THEN
C     INPUT DESCRIPTION OF FREE-FIELD SOIL COLUMN
C     1: -X AREA  ,  2: +X AREA
      DO 90I=1,2
      READ(50,*)NLAYER(I)
      DO 80ILAYER=1,NLAYER(I)
      READ(50,*)NTOP(I,ILAYER),NBOTT(I,ILAYER)
      DO 70 IAREA=1,NAREA
      MARK1=0
      MARK2=0
      DO 70 INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF(NTOP(I,ILAYER) .EQ. INODE)MARK1=1
      IF(NBOTT(I,ILAYER) .EQ. INODE) MARK2=1
      IF(MARK1 .EQ. 1 .AND. MARK2 .EQ. 1)LTYPE(I,ILAYER) = IAREA
   70 CONTINUE
   80 CONTINUE
      LTYPE(I,NLAYER(I)+1) = HALFSP
   90 CONTINUE
      END IF
      CLOSE(UNIT=50)
      OPEN(UNIT=60,FILE=AOUT,STATUS='UNKNOWN')
      WRITE(60,6000) ANAME
 6000 FORMAT(2X,A80)
      WRITE(60,6010)ATYPE
 6010 FORMAT(/2X,'TYPE OF WAVE = ',A2)
      X1=REAL(FBAR)
      WRITE(60,6020)X1
 6020 FORMAT(2X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F7.4)
      WRITE(60,6025)NANGLE
 6025 FORMAT(
     & /2X,'ADMITTANCE FUNCTIONS ARE CALCULATED FOR ',I1,' ANGLES')
      DO 100 IANGLE=1,NANGLE
      WRITE(60,6030)IANGLE,ANGLE(IANGLE)
```

```
 6030 FORMAT(2X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
  100 CONTINUE
      WRITE(60,6040)NAREA
 6040 FORMAT(2X,'NUMBER OF HOMOGENEOUS SUBREGIONS = ',I2/)
      IELEM=0
      DO 190IAREA=1,NAREA
      X1=REAL(CSRAT(IAREA))
      WRITE(60,6050)IAREA
 6050 FORMAT(/2X,'HOMOGENEOUS SOIL REGION : ',I1/)
      WRITE(60,6060)IAREA,X1
 6060 FORMAT(2X,'Cs(area ',I1,')/Cs(area 1) = ',F8.2)
      WRITE(60,6070)IAREA,UWTRAT(IAREA)
 6070 FORMAT(2X,'UNIT WT(area ',I1,')/UNIT WT(area 1) = ',F5.2)
      WRITE(60,6080)POISS(IAREA)
 6080 FORMAT(2X,'POISSIONS RATIO ',F5.3)
      WRITE(60,6090)BETA(IAREA)*100.0
 6090 FORMAT(2X,'DAMPING ',F5.2,' %')
      WRITE(60,6100) NBOUND(IAREA)
 6100 FORMAT(//2X,'NUMBER OF BOUNDARIES ',I2/)
      DO 120 IBOUND=1,NBOUND(IAREA)
      WRITE(60,6120)IBOUND,NELEM(IAREA,IBOUND)
 6120 FORMAT(2X,'NO. OF BOUNDARY ELEMENTS FOR BOUNDARY',I2,'=',I3)
  120 CONTINUE
      WRITE(60,6130)
 6130 FORMAT(/2X,'BOUNDARY NODE COORDINATES', T35,'X/H(layer1)',T55,
     & 'Z/H(layer1)')
      DO 140 IBOUND=1,NBOUND(IAREA)
      WRITE(60,6140)IBOUND
 6140 FORMAT(/2X,'BOUNDARY = ',I2)
      DO 130 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      WRITE(60,6150)INODE,X(INODE),Z(INODE)
 6150 FORMAT(T20,I3,T36,F8.4,T51,F8.4)
  130 CONTINUE
  140 CONTINUE
C     NODE CONNECTIVITY
C     WRITE(60,6160)
C 6160 FORMAT (//2X,'NODE CONNECTIVITY')
C     WRITE(60,6170)
C 6170 FORMAT(/T6,'BOUNDARY',T17,'ELEMENT',T27,'NODE 1',T37,'NODE 2')
C     DO 160 IBOUND=1,NBOUND(IAREA)
C     IELEM=IELEM+1
C     WRITE(60,6180)IBOUND,IELEM,
C     & JNODE(IAREA,IBOUND),JNODE(IAREA,IBOUND)+1
C 6180 FORMAT(T9,I2,T19,I3,T28,I3,T38,I3)
C     DO 150 INODE=JNODE(IAREA,IBOUND)+1,KNODE(IAREA,IBOUND)-1
C     IELEM=IELEM+1
C     WRITE(60,6190)IELEM,INODE,INODE+1
C 6190 FORMAT(T19,I3,T28,I3,T38,I3)
C  150 CONTINUE
C     IF(ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
C     IELEM=IELEM+1
C     WRITE(60,6190)IELEM,KNODE(IAREA,IBOUND),JNODE(IAREA,IBOUND)
C     END IF
C  160 CONTINUE
      WRITE(60,6230)
 6230 FORMAT(/)
      DO 6260IBOUND=1,NBOUND(IAREA)
```

```
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)WRITE(60,6240)IBOUND
      IF(ICLOSE(IAREA,IBOUND) .EQ. 1) WRITE(60,6250)IBOUND
 6240 FORMAT(5X,'BOUNDARY',I2,' IS OPEN')
 6250 FORMAT(5X,'BOUNDARY',I2,' IS CLOSED')
 6260 CONTINUE
  190 CONTINUE
      DO 210 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IEQN(INODE)=1
      NCONN(INODE)=0
  210 CONTINUE
      DO 220 IAREA=1,NAREA-1
      DO 220 IBOUND=1,NBOUND(IAREA)
      DO 220 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      IF(NCONN(INODE) .NE. 0 ) GO TO 220
      DO 217 IIAREA=IAREA+1,NAREA
      DO 215IINODE=JNODE(IIAREA,1),KNODE(NAREA,NBOUND(NAREA))
      IF(IEQN(IINODE) .EQ. 0) GO TO 215
      DIST=SQRT((X(INODE)-X(IINODE))**2+(Z(INODE)-Z(IINODE))**2)
      IF(DIST .LT. 0.00001) THEN
      NCONN(IINODE)=INODE
      IEQN(INODE)=IEQN(INODE)+1
      IEQN(IINODE)=0
      GO TO 217
      END IF
  215 CONTINUE
  217 CONTINUE
  220 CONTINUE
C
C     WRITE(60,6270)
C 6270 FORMAT(/5X,'INTERFACE NODES'/)
C     DO 240 INODE=1,KNODE(NAREA,NBOUND(NAREA))
C     IF(IEQN(INODE) .EQ. 2)THEN
C     DO 230 IINODE =INODE+1,KNODE(NAREA,NBOUND(NAREA))
C     IF(NCONN(IINODE) .EQ. INODE) WRITE(60,6280)INODE,IINODE
C 6280 FORMAT(T9,I3,' - ',I3)
C  230 CONTINUE
C     END IF
C  240 CONTINUE
C
      WRITE(60,6290)
 6290 FORMAT(/5X,'3 NODE INTERFACES'/)
      ICOUNT=0
      DO 252 INODE=1,KNODE(NAREA,NBOUND(NAREA))-1
      IMARK=0
      IF(IEQN(INODE) .EQ. 3)THEN
      ICOUNT=ICOUNT+1
      DO 250 IINODE=INODE+1,KNODE(NAREA,NBOUND(NAREA))
      IF(NCONN(IINODE) .EQ. INODE .AND. IMARK .EQ. 0) THEN
      MARK1=IINODE
      IMARK=1
      END IF
      IF(NCONN(IINODE) .EQ. INODE .AND. IMARK .EQ. 1) THEN
      MARK2=IINODE
      END IF
  250 CONTINUE
      N3(ICOUNT,1)=INODE
      N3(ICOUNT,2)=MARK1
      N3(ICOUNT,3)=MARK2
```

```
      WRITE(60,6300)N3(ICOUNT,1),N3(ICOUNT,2),N3(ICOUNT,3)
 6300 FORMAT(T9,I3,' - ',I3,' - ', I3)
      END IF
  252 CONTINUE
      NN3=ICOUNT
      IF(NN3 .EQ. 0) WRITE(60,6305)
 6305 FORMAT(T11,'NONE')
      WRITE(60,6307)
 6307 FORMAT(/5X,'SURFACE NODES')
      IF (NSURFA .EQ. 0) THEN
      WRITE(60,6305)
      ELSE
      DO 253 ISURFA=1,NSURFA
      DO 253 IBND=1,NBND(ISURFA)
      WRITE(60,6308)ISURFA,NSURF1(ISURFA,IBND),NSURF2(ISURFA,IBND)
 6308 FORMAT(/5X,'AREA ',I2,': NODE ',I3,' THROUGH NODE ',I3)
  253 CONTINUE
      END IF
C
C     NODE ORDER
C
      ICOUNT=0
      DO 255 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(IEQN(INODE) .NE. 0) THEN
      ICOUNT=ICOUNT+1
      NORDER(ICOUNT)=INODE
      END IF
  255 CONTINUE
      IF(NAREA .GT. 1 .AND. NNSURF .NE. 0)THEN
C     FREE-FIELD COLUMN DESCRIPTION
      WRITE(60,6310)
 6310 FORMAT(//2X,'FREE-FIELD COLUMN DESCRIPTION')
      DO 270 I=1,2
      IF(I .EQ. 1)WRITE(60,6320)
      IF(I .EQ. 2)WRITE(60,6330)
 6320 FORMAT(/2X,'-X REGION')
 6330 FORMAT(/2X,'+X REGION')
      WRITE(60,6340)
 6340 FORMAT(T3,'LAYER',T21,'d(i)/d(1)',T32,
     & 'Cs(layer i)/Cs(1)',T57,'WT(layer i)/WT(1)',
     & T82,'DAMPING')
      WRITE(60,6350)
 6350 FORMAT(T84,'(%)'/)
      DO 260ILAYER=1,NLAYER(I)
      ITYPE=LTYPE(I,ILAYER)
      X1=REAL(CSRAT(ITYPE))
      WRITE(60,6360)ILAYER,Z(NBOTT(I,ILAYER)),X1,UWTRAT(ITYPE),
     & BETA(ITYPE)*100.0
 6360 FORMAT(T4,I2,T22,F5.2,T37,F6.2,T62,F6.2,T82,F6.3)
  260 CONTINUE
      X1=REAL(CSRAT(HALFSP))
      WRITE(60,6370)X1,UWTRAT(HALFSP),
     & BETA(HALFSP)*100.0
 6370 FORMAT('HALF-SPACE',T37,F6.2,T62,F6.2,T82,F6.3)
  270 CONTINUE
      END IF
C
C
```

```
C
C
      FBAR=FBAR/CSQRT(1.0-(0.0,2.0)*BETA(1))
      DO 280 IAREA=1,NAREA
      CSRAT(IAREA)=CSRAT(IAREA)*CSQRT((1.0-(0.0,2.0)*BETA(IAREA))/
     & (1.0-(0.0,2.0)*BETA(1)))
  280 CONTINUE
      XSCATT=0.0
      DO 320 INODE=JNODE(HALFSP,1),KNODE(HALFSP,NBOUND(HALFSP))
      DO 320 IN3=1,NN3
      DO 320 IINODE=1,3
      IF(N3(IN3,IINODE) .EQ. INODE)THEN
      XSCATT=X(INODE)
      GO TO 330
      END IF
  320 CONTINUE
  330 CONTINUE
      IF(NNSURF .NE. 0) THEN
      WRITE(60,6410)XSCATT
 6410 FORMAT(/2X,'SCATTERING BOUNDARY OCCURS AT X/H1 = ',F7.2)
      END IF
      RETURN
      END
```

```
      SUBROUTINE FFLD(IX)
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C
      INTEGER NSEG(2)
      REAL HRATIO(5),ZLAYER(2),FFBETA(5)
      COMPLEX FFSXXH(50),FFSZZH(50),FFSZXH(50),
     & HSSXXH(2),HSSZZH(2),HSSZXH(2)
      COMPLEX UTOP,WTOP,AP(5),BP(5),ASV(5),BSV(5),FFCSR(3),FFWTR(3),
     & FFPOIS(3),THEATA(5),CALC1,CALC2,CALC3,CALC4,CALC5,CALC6,CALC7,
     & CALC8,XIKSH1,XIKSH2,XIKTH1,XIKTH2,XLX(5),XMX(5),XI,XIKSZ,XIKTZ,
     & XIKX,RCPCS,APHS,BPHS,ASVHS,BSVHS,UOTCRP,WOTCRP,SXXOTC
C
      COMPLEX XTEMP1,XTEMP2,XTEMP3
C
```

```
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      PI=3.141593
C
      IF(NAREA .EQ. 1)THEN
C     NAREA = 1
      FFCSR(1)=CSRAT(HALFSP)
      FFWTR(1)=UWTRAT(HALFSP)
      FFPOIS(1)=POISS(HALFSP)
      FFBETA(1)=BETA(HALFSP)
      THEATA(1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      IF(ATYPE .EQ. 'P')THEN
      AP(1)=(1.0,0.0)
      ASV(1)=(0.0,0.0)
      XLX(1)=CCOS(THEATA(1))
      XMX(1)=XLX(1)*SQRT((1-2.*FFPOIS(1))/(2.*(1.-FFPOIS(1))))
      ELSE
      ASV(1)=(1.0,0.0)
      AP(1)=(0.0,0.0)
      XMX(1)=CCOS(THEATA(1))
      XLX(1)=XMX(1)* SQRT((2.*(1.-FFPOIS(1)))/(1-2.*FFPOIS(1)))
      END IF
      FFDIM=2
      DO 5I=1,FFDIM
      FVECT(I)=(0.0,0.0)
      DO 5J=1,FFDIM
      STIFF(I,J)=(0.0,0.0)
    5 CONTINUE
      DIFF=ABS(ANGLE(IANGLE)-90.0)
      IF(DIFF .LT. 0.0001)THEN
C
C     VERTICAL INCIDENCE
C
      BP(1)= -1.0*AP(1)
      BSV(1)= -1.0*ASV(1)
      ELSE
C
C     NON-VERTICAL INCIDENCE
C
      CALC1=CSQRT(1.0-(XLX(1)**2))
      CALC3=CSQRT(1.0-(XMX(1)**2))
      CALC5=(1.-2.*(XMX(1)**2))/XMX(1)
      CALC7=CALC5/XMX(1)
      STIFF(1,1)=XLX(1)*CALC7
      STIFF(1,2)= -2.*CALC3
      STIFF(2,1)= 2.*CALC1
      STIFF(2,2)=CALC5
      FVECT(1)= -1.0* STIFF(1,1)*AP(1)+STIFF(1,2)*ASV(1)
      FVECT(2)= STIFF(2,1)*AP(1)-STIFF(2,2)*ASV(1)
      CALL SOLVE(1)
      BP(1)=XVECT(1)
      BSV(1)=XVECT(2)
      END IF
      HSU(IX)=
     & XLX(1)*(AP(1)+BP(1))+CSQRT(1.0-(XMX(1)**2))*(-ASV(1)+BSV(1))
      HSW(IX)=
     & SQRT(1.0-(XLX(1)**2))*(-AP(1)+BP(1))+XMX(1)*(-ASV(1)-BSV(1))
      HSSZZH(IX)=(0.0,0.0)
```

```
      HSSZXH(IX)=(0.0,0.0)
C
      CALC1=SQRT( 2.*(1.-FFPOIS(1))/(1-2.*FFPOIS(1)))
      CALC3=2.*FBAR*XMX(1)/FFCSR(1) *SQRT(1.0-XMX(1)*XMX(1))
      CALC6=FBAR*(CALC1/FFCSR(1)+2.*(XLX(1)-1.0)/
     & (CALC1*FFCSR(1)))
C
      HSSXXH(IX)=(0.0,1.0)*
     & ((AP(1)+ BP(1))*CALC6-
     & (ASV(1)- BSV(1))*CALC3)
C
      HSPX(IX)=(0.0,0.0)
      HSPZ(IX)=(0.0,0.0)
      HSU(2)=HSU(1)
      HSW(2)=HSW(1)
      HSSXXH(2)=HSSXXH(1)
      HSSZZH(2)=(0.0,0.0)
      HSSZXH(2)=(0.0,0.0)
      HSPX(2)=(0.0,0.0)
      HSPZ(2)=(0.0,0.0)
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1) *X(KNODE(HALFSP,1))
      FFU(1,1)=HSU(1)*CEXP(XIKX)
      FFW(1,1)=HSW(1)*CEXP(XIKX)
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1) *X(JNODE(HALFSP,1))
      FFU(2,1)=HSU(2)*CEXP(XIKX)
      FFW(2,1)=HSW(2)*CEXP(XIKX)
C     WRITE(*,*)'FFU(1,1) = ',FFU(1,1)
C     WRITE(*,*)'FFW(1,1) = ',FFW(1,1)
C     WRITE(*,*)'FFU(2,1) = ',FFU(2,1)
C     WRITE(*,*)'FFW(2,1) = ',FFW(2,1)
      WRITE(60,6090)
      WRITE(60,6100)
      WRITE(60,6110)HSU(IX)
      WRITE(60,6120)HSW(IX)
      WRITE(60,6130)
C
      XTEMP1=HSSXXH(IX)*(1.0-(0.0,2.0)*FFBETA(1))
      XTEMP2=HSSZZH(IX)*(1.0-(0.0,2.0)*FFBETA(1))
      XTEMP3=HSSZXH(IX)*(1.0-(0.0,2.0)*FFBETA(1))
C
      WRITE(60,6140)XTEMP1
      WRITE(60,6150)XTEMP2
      WRITE(60,6160)XTEMP3
      WRITE(60,6170)
C
      XTEMP1=HSPX(IX)*(1.0-(0.0,2.0)*FFBETA(1))
      XTEMP2=HSPZ(IX)*(1.0-(0.0,2.0)*FFBETA(1))
C
      WRITE(60,6180)XTEMP1
      WRITE(60,6190)XTEMP2
C
      GO TO 9999
      END IF
C
C     NUMBER OF FREE-FIELD SEGMENTS = NNSEG
C     NNSEG=5
C     CREATE 10 SEGMENTS PER SHEAR WAVE LENGTH
      NNSEG=REAL(FBAR)*10/(2.0*PI) +1
```

```
      IF (NNSEG .LT. 5)NNSEG=5
C
      DO 10 ILAYER=1,NLAYER(IX)
      ITYPE=LTYPE(IX,ILAYER)
      FFCSR(ILAYER)=CSRAT(ITYPE)
      FFWTR(ILAYER)=UWTRAT(ITYPE)
      FFPOIS(ILAYER)=POISS(ITYPE)
      HRATIO(ILAYER)=Z(NBOTT(IX,ILAYER))
      FFBETA(ILAYER)= BETA(ITYPE)
   10 CONTINUE
      FFCSR(NLAYER(IX)+1)=CSRAT(HALFSP)
      FFWTR(NLAYER(IX)+1)=UWTRAT(HALFSP)
      FFPOIS(NLAYER(IX)+1)=POISS(HALFSP)
      FFBETA(NLAYER(IX)+1)=BETA(HALFSP)
      THEATA(NLAYER(IX)+1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      IF(ATYPE .EQ. 'P')THEN
      AP(NLAYER(IX)+1)=(1.0,0.0)
      ASV(NLAYER(IX)+1)=(0.0,0.0)
      XLX(NLAYER(IX)+1)=CCOS(THEATA(NLAYER(IX)+1))
      XMX(NLAYER(IX)+1)=XLX(NLAYER(IX)+1)*
     & SQRT((1-2.*FFPOIS(NLAYER(IX)+1))/
     & (2.*(1.-FFPOIS(NLAYER(IX)+1))))
      DO 20ILAYER=1,NLAYER(IX)
C
      XLX(ILAYER)=XLX(NLAYER(IX)+1)*FFCSR(ILAYER)/FFCSR(NLAYER(IX)+1)*
     & SQRT((1.-FFPOIS(ILAYER))*(1.0-(2.*FFPOIS(NLAYER(IX)+1)))/
     & ((1.-FFPOIS(NLAYER(IX)+1))*(1.-(2.*FFPOIS(ILAYER)))))
C
      XMX(ILAYER)=XLX(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
   20 CONTINUE
      ELSE
      ASV(NLAYER(IX)+1)=(1.0,0.0)
      AP(NLAYER(IX)+1)=(0.0,0.0)
      XMX(NLAYER(IX)+1)=CCOS(THEATA(NLAYER(IX)+1))
      XLX(NLAYER(IX)+1)=XMX(NLAYER(IX)+1)*
     & SQRT((2.*(1.-FFPOIS(NLAYER(IX)+1)))/
     & (1-2.*FFPOIS(NLAYER(IX)+1)))
      DO 25ILAYER=1,NLAYER(IX)
      XMX(ILAYER)=XMX(NLAYER(IX)+1)*FFCSR(ILAYER)/FFCSR(NLAYER(IX)+1)
      XLX(ILAYER)=XMX(ILAYER)*SQRT((2.*(1.-FFPOIS(ILAYER+1)))/
     & (1-2.*FFPOIS(ILAYER+1)))
   25 CONTINUE
      END IF
      FFDIM=4*NLAYER(IX)+2
      DO 40I=1,FFDIM
      FVECT(I)=(0.0,0.0)
      DO 40J=1,FFDIM
      STIFF(I,J)=(0.0,0.0)
   40 CONTINUE
      DIFF=ABS(ANGLE(IANGLE)-90.0)
      IF(DIFF .LT. 0.0001)THEN
C
C     VERTICAL INCIDENCE
C
      STIFF(1,1)=(1.0,0.0)
      STIFF(1,2)=(1.0,0.0)
      STIFF(2,3)=(1.0,0.0)
```

```
      STIFF(2,4)=(1.0,0.0)
      DO 50ILAYER=1,NLAYER(IX)
      XIKSH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
      XIKSH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)
      XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)
      I=(ILAYER-1)*4+3
      J=(ILAYER-1)*4+1
      XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
     & * FFWTR(ILAYER+1)/FFWTR(ILAYER)
      CALC1=XI*FFCSR(ILAYER)/FFCSR(ILAYER+1)
      CALC2=SQRT((2.*(1.-FFPOIS(ILAYER)))/(1.-2.*FFPOIS(ILAYER)))
      CALC3=CALC1*SQRT((2.*(1.-FFPOIS(ILAYER+1)))/
     & (1-2.*FFPOIS(ILAYER+1)))
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I,J+3)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+1,J)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+1,J+1)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+2,J)   =CALC2*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+2,J+1)=CALC2*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+3,J+2)=CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+3,J+3)=CEXP((+1.0,0.0)*XIKTH1)
      IF(ILAYER .LT. NLAYER(IX))THEN
      STIFF(I,J+6)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+7)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+4)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+1,J+5)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -CALC3*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+2,J+5)= -CALC3*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+3,J+6)=(-1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+3,J+7)=(-1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I+1,J+4)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -CALC3*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I,J+5)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+5)=(-1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER(IX)+1)
      FVECT(I+1)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER(IX)+1)
      FVECT(I+2)=(+1.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKSH2)*
     & AP(NLAYER(IX)+1)
      FVECT(I+3)=(+1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER(IX)+1)
      END IF
   50 CONTINUE
      ELSE
C
C     NON-VERTICAL INCIDENCE
C
      CALC1=CSQRT(1.0-(XLX(1)**2))
      CALC3=CSQRT(1.0-(XMX(1)**2))
      CALC5=(1.-2.*(XMX(1)**2))/XMX(1)
      CALC7=CALC5/XMX(1)
```

```
 STIFF(1,1)=XLX(1)*CALC7
 STIFF(1,2)=STIFF(1,1)
 STIFF(1,3)=2.*CALC3
 STIFF(1,4)=-1.*STIFF(1,3)
 STIFF(2,1)= -2.*CALC1
 STIFF(2,2)= 2.*CALC1
 STIFF(2,3)=CALC5
 STIFF(2,4)=CALC5
 DO 60ILAYER=1,NLAYER(IX)
 XIKSH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
& HRATIO(ILAYER)*CSQRT(1.0-(XLX(ILAYER)**2))*
& SQRT((1-2.*FFPOIS(ILAYER))/(2.*(1.-FFPOIS(ILAYER))))
 XIKSH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
& HRATIO(ILAYER)*CSQRT(1.0-(XLX(ILAYER+1)**2))*
& SQRT((1-2.*FFPOIS(ILAYER+1))/(2.*(1.-FFPOIS(ILAYER+1))))
 XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
& HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER)**2))
 XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
& HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER+1)**2))
 I=(ILAYER-1)*4+3
 J=(ILAYER-1)*4+1
 XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
& * FFWTR(ILAYER+1)/FFWTR(ILAYER)
 CALC1=CSQRT(1.0-(XLX(ILAYER)**2))
 CALC2=CSQRT(1.0-(XLX(ILAYER+1)**2))
 CALC3=CSQRT(1.0-(XMX(ILAYER)**2))
 CALC4=CSQRT(1.0-(XMX(ILAYER+1)**2))
 CALC5=(1.-2.*(XMX(ILAYER)**2))/XMX(ILAYER)
 CALC6=(1.-2.*(XMX(ILAYER+1)**2))/XMX(ILAYER+1)
 CALC7=CALC5/XMX(ILAYER)
 CALC8=CALC6/XMX(ILAYER+1)
 STIFF(I,J)=(+1.0,0.0)*XLX(ILAYER)*CEXP((-1.0,0.0)*XIKSH1)
 STIFF(I+1,J)=(-1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKSH1)
 STIFF(I+2,J)=XLX(ILAYER)*CALC7*CEXP((-1.0,0.0)*XIKSH1)
 STIFF(I+3,J)=(-2.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKSH1)
 STIFF(I,J+1)=(+1.0,0.0)*XLX(ILAYER)*CEXP((+1.0,0.0)*XIKSH1)
 STIFF(I+1,J+1)=(+1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKSH1)
 STIFF(I+2,J+1)=XLX(ILAYER)*CALC7*CEXP((+1.0,0.0)*XIKSH1)
 STIFF(I+3,J+1)=(+2.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKSH1)
 STIFF(I,J+2)=(-1.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKTH1)
 STIFF(I+1,J+2)=(-1.0,0.0)*XMX(ILAYER)*CEXP((-1.0,0.0)*XIKTH1)
 STIFF(I+2,J+2)=(+2.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKTH1)
 STIFF(I+3,J+2)=(+1.0,0.0)*CALC5*CEXP((-1.0,0.0)*XIKTH1)
 STIFF(I,J+3)=(+1.0,0.0)*CALC3*CEXP((+1.0,0.0)*XIKTH1)
 STIFF(I+1,J+3)=(-1.0,0.0)*XMX(ILAYER)*CEXP((+1.0,0.0)*XIKTH1)
 STIFF(I+2,J+3)=(-2.0,0.0)*CALC3*CEXP((+1.0,0.0)*XIKTH1)
 STIFF(I+3,J+3)=(+1.0,0.0)*CALC5*CEXP((+1.0,0.0)*XIKTH1)
 IF(ILAYER .LT. NLAYER(IX))THEN
 STIFF(I,J+4)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((-1.0,0.0)*XIKSH2)
 STIFF(I+1,J+4)=(+1.0,0.0)*CALC2*CEXP((-1.0,0.0)*XIKSH2)
 STIFF(I+2,J+4)= -XI*XLX(ILAYER+1)*CALC8*CEXP((-1.0,0.0)*XIKSH2)
 STIFF(I+3,J+4)=(+2.0,0.0)*XI*CALC2*CEXP((-1.0,0.0)*XIKSH2)
 STIFF(I,J+5)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((+1.0,0.0)*XIKSH2)
 STIFF(I+1,J+5)=(-1.0,0.0)*CALC2*CEXP((+1.0,0.0)*XIKSH2)
 STIFF(I+2,J+5)= -XI*XLX(ILAYER+1)*CALC8*CEXP((+1.0,0.0)*XIKSH2)
 STIFF(I+3,J+5)=(-2.0,0.0)*XI*CALC2*CEXP((+1.0,0.0)*XIKSH2)
 STIFF(I,J+6)=(+1.0,0.0)*CALC4*CEXP((-1.0,0.0)*XIKTH2)
 STIFF(I+1,J+6)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((-1.0,0.0)*XIKTH2)
```

```
      STIFF(I+2,J+6)=(-2.0,0.0)*XI*CALC4*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+3,J+6)=(-1.0,0.0)*XI*CALC6*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+7)=(-1.0,0.0)*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+7)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+2,J+7)=(+2.0,0.0)*XI*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+7)=(-1.0,0.0)*XI*CALC6*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I,J+4)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+1,J+4)=(-1.0,0.0)*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -XI*XLX(ILAYER+1)*CALC8*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+3,J+4)=(-2.0,0.0)*XI*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I,J+5)=(-1.0,0.0)*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+5)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+2,J+5)=(+2.0,0.0)*XI*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+5)=(-1.0,0.0)*XI*CALC6*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(+1.0,0.0)*XLX(ILAYER+1)*CEXP((-1.0,0.0)*XIKSH2)
     & *AP(NLAYER(IX)+1)-CALC4*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER(IX)+1)
      FVECT(I+1)=(-1.0,0.0)*CALC2*CEXP((-1.0,0.0)*XIKSH2)
     & *AP(NLAYER(IX)+1)-XMX(ILAYER+1)*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER(IX)+1)
      FVECT(I+2)=(+1.0,0.0)*XI*XLX(ILAYER+1)*CALC8*
     & CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER(IX)+1)
     & +(2.0,0.0)*XI*CALC4*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER(IX)+1)
      FVECT(I+3)=(-2.0,0.0)*XI*CALC2*
     & CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER(IX)+1)
     & +XI*CALC6*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER(IX)+1)
      END IF
   60 CONTINUE
      END IF
      CALL SOLVE(1)
      DO 70ILAYER=1,NLAYER(IX)
      IROW=(ILAYER-1)*4+1
      AP(ILAYER)=XVECT(IROW)
      BP(ILAYER)=XVECT(IROW+1)
      ASV(ILAYER)=XVECT(IROW+2)
      BSV(ILAYER)=XVECT(IROW+3)
   70 CONTINUE
      BP(NLAYER(IX)+1)=XVECT(FFDIM-1)
      BSV(NLAYER(IX)+1)=XVECT(FFDIM)
      UTOP=XLX(1)*(AP(1)+BP(1))+CSQRT(1.0-(XMX(1)**2))*(-ASV(1)+BSV(1))
      WTOP=SQRT(1.0-(XLX(1)**2))*(-AP(1)+BP(1))+XMX(1)*(-ASV(1)-BSV(1))
C
C
C
      IF(IX .EQ. 1)THEN
      WRITE(60,6000)
 6000 FORMAT(//5X,'FREE-FIELD CALCULATION')
      WRITE(60,6002)
 6002 FORMAT(/5X,'-X BOUNDARY')
      END IF
      IF(IX .EQ. 2)THEN
      WRITE(60,6004)
 6004 FORMAT(/5X,'+X BOUNDARY')
      END IF
      WRITE(60,6005)
 6005 FORMAT(/T3,'LAYER',T15,'Ap',T30,'Bp',T49,'Asv',T64,'Bsv'/)
      DO 80ILAYER=1,NLAYER(IX)+1
```

```
      WRITE(60,6010)ILAYER,AP(ILAYER),BP(ILAYER),ASV(ILAYER),
     & BSV(ILAYER)
 6010 FORMAT(I5,T8,2F7.3,T23,2F7.3,T43,2F7.3,T58,2F7.3)
   80 CONTINUE
      UREAL=REAL(UTOP)
      IF(ABS(UREAL) .LT. (1.0E-12) ) UREAL=0.0
      UIMAG=AIMAG(UTOP)
      IF(ABS(UIMAG) .LT. (1.0E-12) ) UIMAG=0.0
      WREAL=REAL(WTOP)
      IF(ABS(WREAL) .LT. (1.0E-12) ) WREAL=0.0
      WIMAG=AIMAG(WTOP)
      IF(ABS(WIMAG) .LT. (1.0E-12) ) WIMAG=0.0
      U=SQRT(UREAL*UREAL+UIMAG*UIMAG)
      W=SQRT(WREAL*WREAL+WIMAG*WIMAG)
      WRITE(60,6020)
 6020 FORMAT(//T18,'REAL',T31,'IMAGINARY',T46,'MAGNITUDE')
      WRITE(60,6030)UREAL,UIMAG,U
 6030 FORMAT(/2X'U(top)',T15,F8.3,T30,F8.3,T45,F8.3)
      WRITE(60,6040)WREAL,WIMAG,W
 6040 FORMAT(/2X'W(top)',T15,F8.3,T30,F8.3,T45,F8.3)
      WRITE(*,6000)
      IF(IX .EQ. 1)THEN
      WRITE(*,6002)
      END IF
      IF(IX .EQ. 2)THEN
      WRITE(*,6004)
      END IF
      WRITE(*,6005)
      DO 90ILAYER=1,NLAYER(IX)+1
      WRITE(*,6010)ILAYER,AP(ILAYER),BP(ILAYER),ASV(ILAYER),BSV(ILAYER)
   90 CONTINUE
      WRITE(*,6020)
      WRITE(*,6030)UREAL,UIMAG,U
      WRITE(*,6040)WREAL,WIMAG,W
C
C
      TBIG=0.
      DO 100ILAYER=1,NLAYER(IX)
      T=Z(NBOTT(IX,ILAYER))-Z(NTOP(IX,ILAYER))
      IF(T .GT. TBIG)TBIG=T
      IF(ILAYER .EQ. 1) THEN
      ZLAYER(1)=T
      ELSE
      ZLAYER(ILAYER)=ZLAYER(ILAYER-1)+T
      END IF
  100 CONTINUE
      DO 110 ILAYER=1,NLAYER(IX)
      IF (ILAYER .EQ. 1)TLAYER=ZLAYER(1)
      IF (ILAYER .GT. 1)TLAYER=ZLAYER(ILAYER)-ZLAYER(ILAYER-1)
      NSEG(ILAYER)=    TLAYER/ZLAYER(1)*NNSEG
      NSEG1=REAL(FBAR)*10/(2.0*PI)*TLAYER/REAL(FFCSR(ILAYER))
      IF(NSEG1 .GT. NSEG(ILAYER))THEN
      NSEG(ILAYER)=NSEG1
      END IF
  110 CONTINUE
C
C
      INODE=0
```

```
      DO 130 ILAYER=1,NLAYER(IX)
      ZTOP=Z(NTOP(IX,ILAYER))
      ZBOTT=Z(NBOTT(IX,ILAYER))
      SEGLEN=(ZBOTT-ZTOP)/NSEG(ILAYER)
      IF(ILAYER .EQ. 1)JFF(IX,ILAYER)=1
      IF(ILAYER .GT. 1)JFF(IX,ILAYER)=KFF(IX,ILAYER-1)+1
      KFF(IX,ILAYER)=JFF(IX,ILAYER)+NSEG(ILAYER)
      XFF(IX,ILAYER)=X(NTOP(IX,ILAYER))
C
      CALC1=SQRT((2.*(1.-FFPOIS(ILAYER)))/(1-2.*FFPOIS(ILAYER)))
      XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)  *XFF(IX,ILAYER)
      CALC2=FBAR/FFCSR(ILAYER)*CALC1*(1.0-2.*XMX(ILAYER)*XMX(ILAYER))
      CALC3=2.*FBAR*XMX(1)/FFCSR(1)*SQRT(1.0-XMX(ILAYER)*XMX(ILAYER))
      CALC4=2.*FBAR*XMX(1)/FFCSR(1)*SQRT(1.0-XLX(ILAYER)*XLX(ILAYER))
      CALC5=FBAR/FFCSR(ILAYER)*(1.0-2.*XMX(ILAYER)*XMX(ILAYER))
      CALC6=FBAR*(CALC1/FFCSR(ILAYER)+2.*(XLX(ILAYER)-1.0)/
     & (CALC1*FFCSR(ILAYER)))
C
C
      XNZ=0.0
      IF(IX .EQ. 1)XNX= -1.0
      IF(IX .EQ. 2)XNX= +1.0
      DO  120 ISEG=1,NSEG(ILAYER)+1
      INODE=INODE+1
      IF(IX .EQ. 1)ZFF(IX,INODE)=ZTOP+(ISEG-1)*SEGLEN
      IF(IX .EQ. 2)ZFF(IX,INODE)=ZBOTT-(ISEG-1)*SEGLEN
C
      XIKSZ=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & CSQRT(1.0-(XLX(ILAYER)**2))/CALC1*ZFF(IX,INODE)
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & CSQRT(1.0-(XMX(ILAYER)**2))*ZFF(IX,INODE)
      FFU(IX,INODE)=(XLX(ILAYER)*
     & (AP(ILAYER)*CEXP(-XIKSZ)+ BP(ILAYER)*CEXP(+XIKSZ))+
     & (CSQRT(1.0-XMX(ILAYER)**2))*
     & (-ASV(ILAYER)*CEXP(-XIKTZ)+ BSV(ILAYER)*CEXP(+XIKTZ)))*
     & CEXP(XIKX)
      FFW(IX,INODE)=(CSQRT(1.0-XLX(ILAYER)**2)*
     & (-AP(ILAYER)*CEXP(-XIKSZ)+ BP(ILAYER)*CEXP(+XIKSZ))+
     &  XMX(ILAYER)*
     & (-ASV(ILAYER)*CEXP(-XIKTZ)- BSV(ILAYER)*CEXP(+XIKTZ)))*
     & CEXP(XIKX)
C
      FFSZZH(INODE)=(0.0,1.0)*
     & ((AP(ILAYER)*CEXP(-XIKSZ)+ BP(ILAYER)*CEXP(+XIKSZ))*CALC2+
     & (ASV(ILAYER)*CEXP(-XIKTZ)- BSV(ILAYER)*CEXP(+XIKTZ))*CALC3)*
     & CEXP(XIKX)
C
      FFSZXH(INODE)=(0.0,1.0)*
     & ((-AP(ILAYER)*CEXP(-XIKSZ)+ BP(ILAYER)*CEXP(+XIKSZ))*CALC4+
     & (ASV(ILAYER)*CEXP(-XIKTZ)+ BSV(ILAYER)*CEXP(+XIKTZ))*CALC5)*
     & CEXP(XIKX)
C
      FFSXXH(INODE)=(0.0,1.0)*
     & ((AP(ILAYER)*CEXP(-XIKSZ)+ BP(ILAYER)*CEXP(+XIKSZ))*CALC6-
     & (ASV(ILAYER)*CEXP(-XIKTZ)- BSV(ILAYER)*CEXP(+XIKTZ))*CALC3)*
     & CEXP(XIKX)
      FFPX(IX,INODE)=FFSXXH(INODE)*XNX+FFSZXH(INODE)*XNZ
      FFPZ(IX,INODE)=FFSZXH(INODE)*XNX+FFSZZH(INODE)*XNZ
```

```
C
  120 CONTINUE
  130 CONTINUE
      DO 150 ILAYER=1,NLAYER(IX)
      WRITE(60,6050)ILAYER
 6050 FORMAT(/5X,'LAYER = ',I1)
      WRITE(60,6060)
 6060 FORMAT(T47,'DISPLACEMENT')
      WRITE(60,6070)
 6070 FORMAT(T3,'NODE',T13,'X/H1',T23,'Z/H1',T45,'U',T60,'W'/)
      DO 140 INODE=JFF(IX,ILAYER),KFF(IX,ILAYER)
      WRITE(60,6075)INODE,XFF(IX,ILAYER),ZFF(IX,INODE),
     & FFU(IX,INODE),FFW(IX,INODE)
 6075 FORMAT(T4,I2,T12,F7.3,T22,F7.3,
     & T37,2F7.3,T52,2F7.3,T67,2F7.3)
  140 CONTINUE
      WRITE(60,6077)
 6077 FORMAT(/T57,'STRESS')
      WRITE(60,6080)
 6080 FORMAT(T3,'NODE',T13,'X/H1',T23,'Z/H1',T41,'Sxx*H1/Gi' ,
     & T56,'Szz*H1/Gi', T71,'Szx*H1/Gi'  /)
      DO 142 INODE=JFF(IX,ILAYER),KFF(IX,ILAYER)
C
      XTEMP1=FFSXXH(INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
      XTEMP2=FFSZZH(INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
      XTEMP3=FFSZXH(INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
C
      WRITE(60,6075)INODE,XFF(IX,ILAYER),ZFF(IX,INODE),
C     & FFSXXH(INODE),FFSZZH(INODE),FFSZXH(INODE)
     & XTEMP1,XTEMP2,XTEMP3
C
  142 CONTINUE
      WRITE(60,6082)
 6082 FORMAT(/T48,'TRACTION')
      WRITE(60,6085)
 6085 FORMAT(T3,'NODE',T13,'X/H1',T23,'Z/H1',T41,'Px*H1/Gi',
     & T56,'PzH1/Gi'/)
      DO 145 INODE=JFF(IX,ILAYER),KFF(IX,ILAYER)
C
      XTEMP1=FFPX(IX,INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
      XTEMP2=FFPZ(IX,INODE)*(1.0-(0.0,2.0)*FFBETA(ILAYER))
C
      WRITE(60,6075)INODE,XFF(IX,ILAYER),ZFF(IX,INODE),
C     & FFPX(IX,INODE),FFPZ(IX,INODE)
     & XTEMP1,XTEMP2
C
  145 CONTINUE
  150 CONTINUE
C
      NN=NLAYER(IX)+1
C
      CALC1=SQRT( 2.*(1.-FFPOIS(NN))/(1-2.*FFPOIS(NN)))
      CALC2=FBAR/FFCSR(NN)*CALC1*(1.0-2.*XMX(NN)*XMX(NN))
      CALC3=2.*FBAR*XMX(1)/FFCSR(1)*SQRT(1.0-XMX(NN)*XMX(NN))
      CALC4=2.*FBAR*XMX(1)/FFCSR(1)*SQRT(1.0-XLX(NN)*XLX(NN))
      CALC5=FBAR/FFCSR(NN)*(1.0-2.*XMX(NN)*XMX(NN))
      CALC6=FBAR*(CALC1/FFCSR(NN)+2.*(XLX(NN)-1.0)/
     & (CALC1*FFCSR(NN)))
```

```
C
C
C     FREE-FIELD MOTION IS BASED ON HORIZONTAL HALF-SPACE INTERFACE
      XNX=  0.0
      XNZ= -1.0
      ZHS=Z(KNODE(HALFSP,NBOUND(HALFSP)))
C
      XIKSZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XLX(NN)**2))/CALC1*ZHS
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XMX(NN)**2))*ZHS
C     XIKX=(0.0,1.0)*FBAR*XMX(1)/FFCSR(1)  *X(INODE)
      HSU(IX)=(XLX(NN)*
     & (AP(NN)*CEXP(-XIKSZ)+ BP(NN)*CEXP(+XIKSZ))+
     &  CSQRT(1.0-XMX(NN)**2)*
     & (-ASV(NN)*CEXP(-XIKTZ)+ BSV(NN)*CEXP(+XIKTZ)))
C     & *CEXP(XIKX)
      HSW(IX)=(CSQRT(1.0-XLX(NN)**2)*
     & (-AP(NN)*CEXP(-XIKSZ)+ BP(NN)*CEXP(+XIKSZ))+
     &  XMX(NN)*
     & (-ASV(NN)*CEXP(-XIKTZ)- BSV(NN)*CEXP(+XIKTZ)))
C     & *CEXP(XIKX)
C
      HSSZZH(IX)= (0.0,1.0)*
     & ((AP(NN)*CEXP(-XIKSZ)+ BP(NN)*CEXP(+XIKSZ))*CALC2+
     & (ASV(NN)*CEXP(-XIKTZ)- BSV(NN)*CEXP(+XIKTZ))*CALC3)
C     & *CEXP(XIKX)
C
      HSSZXH(IX)= (0.0,1.0)*
     & ((-AP(NN)*CEXP(-XIKSZ)+ BP(NN)*CEXP(+XIKSZ))*CALC4+
     & (ASV(NN)*CEXP(-XIKTZ)+ BSV(NN)*CEXP(+XIKTZ))*CALC5)
C     & *CEXP(XIKX)
C
      HSSXXH(IX)= (0.0,1.0)*
     & ((AP(NN)*CEXP(-XIKSZ)+ BP(NN)*CEXP(+XIKSZ))*CALC6-
     & (ASV(NN)*CEXP(-XIKTZ)- BSV(NN)*CEXP(+XIKTZ))*CALC3)
C     & *CEXP(XIKX)
C
      HSPX(IX)=HSSXXH(IX)*XNX+HSSZXH(IX)*XNZ
      HSPZ(IX)=HSSZXH(IX)*XNX+HSSZZH(IX)*XNZ
  160 CONTINUE
      WRITE(60,6090)
 6090 FORMAT(/5X,'HALF-SPACE')
      WRITE(60,6100)
 6100 FORMAT(/2X,'DISPLACEMENT: ')
      WRITE(60,6110)HSU(IX)
 6110 FORMAT(5X,'u/exp(ikx)  =',T30,2F8.3)
      WRITE(60,6120)HSW(IX)
 6120 FORMAT(5X,'w/exp(ikx)  =',T30,2F8.3)
      WRITE(60,6130)
 6130 FORMAT(/2X,'STRESS: ')
C
      XTEMP1=HSSXXH(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
      XTEMP2=HSSZZH(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
      XTEMP3=HSSZXH(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
C
      WRITE(60,6140)XTEMP1
 6140 FORMAT(5X,'(Sxx*H1/Ghs)/exp(ikx) =',T30,2F8.3)
```

```
      WRITE(60,6150)XTEMP2
 6150 FORMAT(5X,'(Szz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6160)XTEMP3
 6160 FORMAT(5X,'(Szx*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6170)
 6170 FORMAT(/2X,'TRACTION: ')
C
      XTEMP1=HSPX(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
      XTEMP2=HSPZ(IX)*(1.0-(0.0,2.0)*FFBETA(NLAYER(IX)+1))
C
      WRITE(60,6180)HSPX(IX)
 6180 FORMAT(5X,'(Px*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6190)HSPZ(IX)
 6190 FORMAT(5X,'(Pz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C
C     ROCK OUTCROP
C
      RCPCS=SQRT((2.*(1.-FFPOIS(NN)))/(1.-2.*FFPOIS(NN)))
C
C     HALF-SPACE OUTCROP
      FFDIM=2
      APHS=AP(NN)*CEXP(-XIKSZ)
      ASVHS=ASV(NN)*CEXP(-XIKTZ)
      STIFF(1,1)= RCPCS-2.*XMX(NN)*XLX(NN)
      STIFF(1,2)= -2.*XMX(NN)*SQRT(1.-XMX(NN)*XMX(NN))
      STIFF(2,1)= 2.*XMX(NN)*SQRT(1.-XLX(NN)*XLX(NN))
      STIFF(2,2)= 1.0-2.*XMX(NN)*XMX(NN)
      FVECT(1)= (2.*XMX(NN)*XLX(NN)-RCPCS)*APHS
     & -2.*XMX(NN)*SQRT(1.-XMX(NN)*XMX(NN))*ASVHS
      FVECT(2)=2.*XMX(NN)*SQRT(1.-XLX(NN)*XLX(NN))*APHS
     &  +(2.*XMX(NN)*XMX(NN)-1)*ASVHS
      CALL SOLVE(1)
      BPHS=XVECT(1)
      BSVHS=XVECT(2)
      UOTCRP= XLX(NN)*(APHS+BPHS)+
     &           SQRT(1.-XMX(NN)*XMX(NN))* (-ASVHS+BSVHS)
      UOTMAG=SQRT(REAL(UOTCRP)**2+AIMAG(UOTCRP)**2)
      WOTCRP= SQRT(1.-XLX(NN)*XLX(NN))*(-APHS+BPHS)+
     &           XMX(NN)*(-ASVHS-BSVHS)
      WOTMAG=SQRT(REAL(WOTCRP)**2+AIMAG(WOTCRP)**2)
      SXXOTC= (0.0,1.0)*
     & ((APHS+ BPHS)*CALC6-
     & (ASVHS- BSVHS)*CALC3)
C
      WRITE(60,6200)
 6200 FORMAT(/5X,'ROCK OUTCROP')
      WRITE(60,6210)
 6210 FORMAT(/2X,'DISPLACEMENT: ')
      WRITE(60,6220)UOTCRP,UOTMAG
 6220 FORMAT(5X,'u =',T30,2F8.3,T50,'|u| = ',F6.3)
      WRITE(60,6230)WOTCRP,WOTMAG
 6230 FORMAT(5X,'w =',T30,2F8.3,T50,'|w| = ',F6.3)
      WRITE(60,6240)
 6240 FORMAT(/2X,'STRESS: ')
      CALC1=(0.0,0.0)
C
      XTEMP1=SXXOTC*(1.0-(0.0,2.0)*BETA(HALFSP))
C
```

```
      WRITE(60,6250)XTEMP1
 6250 FORMAT(5X,'(Sxx*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6260)CALC1
 6260 FORMAT(5X,'(Szz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6270)CALC1
 6270 FORMAT(5X,'(Szx*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6280)
 6280 FORMAT(/2X,'TRACTION: ')
      WRITE(60,6290)CALC1
 6290 FORMAT(5X,'(Px*H1/Ghs)/exp(ikx) =',T30,2F8.3)
      WRITE(60,6300)CALC1
 6300 FORMAT(5X,'(Pz*H1/Ghs)/exp(ikx) =',T30,2F8.3)
C
 9999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MATRIX
C234567891123456789212345678931234567894123456789512345678961234567897 12
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
C
C     HALF-SPACE INTEGRALS
      COMMON/RHSINT/ASTART
      COMMON/CHSINT/H0EXP(2,2),H1EXP(2,2),H2EXP(2,2),H2EXPR(2,2),
     & EXPR2(2),EXPR3(2)
      REAL ASTART
      COMPLEX H0EXP,H1EXP,H2EXP,H2EXPR,EXPR2,EXPR3
C
```

```
C
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
      COMMON/KHLFSP/XKH1,XKPH1,XKSH1
      REAL XKH1,XKPH1,XKSH1
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
      REAL ALPHA(2,2)
C
      COMPLEX DSTAT(2,2),DCORR(2,2)
C
C
C     789112345678921234567893123456789412345678951234567896123456789712
C
C
      IF(IANGLE .EQ. 1)THEN
C
      PI=3.14159265
      TWOPI=2.0*PI
C
      HDIM=2*KNODE(NAREA,NBOUND(NAREA))
      GDIM=2*KNODE(NAREA,NBOUND(NAREA))+6*NN3
C     FORM "G" & "H" MATRIX FOR EACH SUBREGION
      DO 20IROW=1,KNODE(NAREA,NBOUND(NAREA))*2
      FVECT(IROW)=(0.0,0.0)
      DO 5JCOL=1,HDIM
      HMAT(IROW,JCOL)=(0.0,0.0)
    5 CONTINUE
      DO 10 JCOL=1,GDIM
      GMAT(IROW,JCOL)=(0.0,0.0)
   10 CONTINUE
   20 CONTINUE
C     OPEN(UNIT=80,FILE='TRUNC.CK',STATUS='UNKNOWN')
C
C     CHECK IF HALF-SPACE OR FULL-SPACE PROBLEM
      IFULSP=0
      ICOUNT=0
      DO IBOUND=1,NBOUND(HALFSP)
      ICOUNT=ICOUNT+ICLOSE(HALFSP,IBOUND)
      END DO
      IF (ICOUNT .EQ. NBOUND(HALFSP))IFULSP=1
C     IF IFULSP =1, FULL SPACE PROBLEM
C     IF IFULSP =0, HALF-SPACE PROBLEM
C
      JGMAT=0
      IN3=0
```

```
      JCOUNT=0
      DO 100IAREA=1,NAREA
C     WRITE(10,*)'REGION ',IAREA
      CONST=1.0/(8.0*PI*(1.0-POISS(IAREA)))
C
            DO IBOUND = 1, NBOUND(IAREA)
            DO ISRCE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
C
      JCOUNT=JCOUNT+1
C
      IF(JCOUNT .EQ. 10)THEN
      WRITE(*,*)'WORKING ON INFLU FOR SOURCE POINT ',ISRCE
      JCOUNT=0
      END IF
C
      CALL INFLU(ISRCE,X(ISRCE),Z(ISRCE),IAREA,JGMAT)
C
      IF (IAREA .EQ. HALFSP .AND. IFULSP .EQ. 0)THEN
C
C     FIND DIAGONAL TERMS OF HMAT
C     DETERMINE INTERIOR ANGLES
      JNO=JNODE(IAREA,IBOUND)
      KNO=KNODE(IAREA,IBOUND)
      IROW=ISRCE*2 - 1
      JCOL=IROW
C
      IF (ISRCE .EQ. JNO .OR. ISRCE .EQ. KNO)THEN
            IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      ROTAT=PI
      HMAT(IROW,JCOL)=ROTAT/TWOPI+HMAT(IROW,JCOL)
      HMAT(IROW+1,JCOL+1)=ROTAT/TWOPI+HMAT(IROW+1,JCOL+1)
            ELSE
      IF (ISRCE .EQ. JNO)THEN
      X1=X(KNO)
      Z1=Z(KNO)
      X2=X(ISRCE+1)
      Z2=Z(ISRCE+1)
      ELSE
      X1=X(ISRCE-1)
      Z1=Z(ISRCE-1)
      X2=X(JNO)
      Z2=Z(JNO)
      END IF
      XI=X(ISRCE)
      ZI=Z(ISRCE)
      CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,PHI1,PHI2)
      ALPHA(1,1)= -(PHI1 -PHI2)/TWOPI
     &     +CONST*(SIN(2.0*PHI2)-SIN(2.0*PHI1))
      ALPHA(1,2)=
     &  CONST*(COS(2.0*PHI1)-COS(2.0*PHI2))
      ALPHA(2,1)=ALPHA(1,2)
      ALPHA(2,2)= -(PHI1 -PHI2)/TWOPI
     &     -CONST*(SIN(2.0*PHI2)-SIN(2.0*PHI1))
      HMAT(IROW,JCOL)=     (1.0+ALPHA(1,1))+HMAT(IROW,JCOL)
      HMAT(IROW,JCOL+1)=  ALPHA(2,1)+HMAT(IROW,JCOL+1)
      HMAT(IROW+1,JCOL)=  ALPHA(1,2)+HMAT(IROW+1,JCOL)
      HMAT(IROW+1,JCOL+1)=(1.0+ALPHA(2,2))+HMAT(IROW+1,JCOL+1)
            END IF
```

```
C
              ELSE
       X1=X(ISRCE-1)
       X2=X(ISRCE+1)
       Z1=Z(ISRCE-1)
       Z2=Z(ISRCE+1)
       XI=X(ISRCE)
       ZI=Z(ISRCE)
       CALL ROTATE(X1,Z1,X2,Z2,XI,ZI,PHI1,PHI2)
       ALPHA(1,1)= -(PHI1 -PHI2)/TWOPI
      &     +CONST*(SIN(2.0*PHI2)-SIN(2.0*PHI1))
       ALPHA(1,2)=
      &  CONST*(COS(2.0*PHI1)-COS(2.0*PHI2))
       ALPHA(2,1)=ALPHA(1,2)
       ALPHA(2,2)= -(PHI1 -PHI2)/TWOPI
      &     -CONST*(SIN(2.0*PHI2)-SIN(2.0*PHI1))
       HMAT(IROW,JCOL)=     (1.0+ALPHA(1,1))+HMAT(IROW,JCOL)
       HMAT(IROW,JCOL+1)=  ALPHA(2,1)+HMAT(IROW,JCOL+1)
       HMAT(IROW+1,JCOL)=  ALPHA(1,2)+HMAT(IROW+1,JCOL)
       HMAT(IROW+1,JCOL+1)=(1.0+ALPHA(2,2))+HMAT(IROW+1,JCOL+1)
C
   40 CONTINUE
C
              END IF
C
C
      DSTAT(1,1)=1.0+ALPHA(1,1)
      DSTAT(1,2)=ALPHA(2,1)
      DSTAT(2,1)=ALPHA(1,2)
      DSTAT(2,2)=1.0+ALPHA(2,2)
      DO I=1,2
      DO J=1,2
      DCORR(I,J)=(0.0,0.0)
      END DO
      END DO
C
C     WRITE(*,*)'ISRCE = ',ISRCE
C     WRITE(*,*)'PHI1,PHI2'
C     WRITE(*,*)PHI1,PHI2
C     WRITE(*,*)'CONST = ',CONST
C     WRITE(*,9040)
 9040 FORMAT(/'Cij')
C     DO I=1,2
C     WRITE(*,9050)((DSTAT(I,J)-DCORR(I,J)),J=1,2)
 9050 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C     WRITE(*,9052)
 9052 FORMAT(/'Dij')
C     DO I=IROW,IROW+1
C     WRITE(*,9054)(HMAT(I,J),J=JCOL,JCOL+1)
C 9054 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C     PAUSE
C
      ELSE
C
C
C     WRITE(*,*)'ISRCE = ',ISRCE
```

```
C     WRITE(*,*)'X = ',X(ISRCE)
C     WRITE(*,*)'Z = ',Z(ISRCE)
C     IROW=2.*ISRCE -1
C     JCOL=IROW
C     WRITE(*,*)'GMAT'
C     DO I=IROW,IROW+1
C
C     WRITE(*,9053)(GMAT(I,J),J=JCOL,JCOL+1)
C 9053 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C     WRITE(*,*)'HMAT'
C     DO I=IROW,IROW+1
C
C     WRITE(*,9054)(HMAT(I,J),J=JCOL,JCOL+1)
C 9054 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C
C     PAUSE
      CALL STATIC(ISRCE,X(ISRCE),Z(ISRCE),IAREA,DSTAT,DCORR)
C
      IROW=2.*ISRCE -1
      JCOL=IROW
C     CHECK IF CORNER POINT
C
C     IF (ISRCE .NE. JNODE(IAREA,IBOUND) .AND.
C     & ISRCE .NE. KNODE(IAREA,IBOUND))THEN
C     SGLN1=SQRT((X(ISRCE)-X(ISRCE-1))**2 + (Z(ISRCE)-Z(ISRCE-1))**2)
C     SGLN2=SQRT((X(ISRCE+1)-X(ISRCE))**2 + (Z(ISRCE+1)-Z(ISRCE))**2)
C
C     IF SEGMENT LENGTH IS SMALL ON BOTH SIDES OF ISRCE, USE STATIC
C
C     IF (SGLN1 .LT. 0.0999 .AND. SGLN2 .LT. 0.0999)THEN
C     HMAT(IROW,JCOL)=     (0.0,0.0)
C     HMAT(IROW,JCOL+1)=   (0.0,0.0)
C     HMAT(IROW+1,JCOL)=   (0.0,0.0)
C     HMAT(IROW+1,JCOL+1)= (0.0,0.0)
C     DCORR(1,1)= (0.0,0.0)
C     DCORR(1,2)= (0.0,0.0)
C     DCORR(2,1)= (0.0,0.0)
C     DCORR(2,2)= (0.0,0.0)
C     END IF
C     END IF
C
      HMAT(IROW,JCOL)=     DSTAT(1,1)+HMAT(IROW,JCOL)-DCORR(1,1)
      HMAT(IROW,JCOL+1)=   DSTAT(1,2)+HMAT(IROW,JCOL+1)-DCORR(1,2)
      HMAT(IROW+1,JCOL)=   DSTAT(2,1)+HMAT(IROW+1,JCOL)-DCORR(2,1)
      HMAT(IROW+1,JCOL+1)=DSTAT(2,2)+HMAT(IROW+1,JCOL+1)-DCORR(2,2)
C
C     WRITE(*,*)'ISRCE = ',ISRCE
C     WRITE(*,9040)
C     DO I=1,2
C     WRITE(*,9050)((DSTAT(I,J)-DCORR(I,J)),J=1,2)
C     END DO
C     WRITE(*,9052)
C     DO I=IROW,IROW+1
C     WRITE(*,9054)(HMAT(I,J),J=JCOL,JCOL+1)
C     END DO
C
```

```
      END IF
            END DO
            END DO
C     WRITE(10,*)'GMAT'
      ISTART=2*JNODE(IAREA,1)-1
      ISTOP= 2*KNODE(IAREA,NBOUND(IAREA))
      JSTART=JCOL1(JNODE(IAREA,1))
      JSTOP= JCOL1(KNODE(IAREA,NBOUND(IAREA)))+1
C     WRITE(10,*)'ISTART = ',ISTART
C     WRITE(10,*)'ISTOP = ',ISTOP
C     WRITE(10,*)'JSTART = ',JSTART
C     WRITE(10,*)'JSTOP = ',JSTOP
C     DO 50 IROW=ISTART,ISTOP
C     WRITE(10,1010)(GMAT(IROW,J),J=JSTART,JSTOP)
C 1010 FORMAT(8F10.5)
C    50 CONTINUE
C     WRITE(10,1015)
C 1015 FORMAT(/)
C     WRITE(10,*)'HMAT'
C     DO 1020 I=ISTART,ISTOP
C     WRITE(10,1010)(HMAT(I,J),J=ISTART,ISTOP)
C 1020 CONTINUE
  100 CONTINUE
      CALL FIXMAT
      CALL AFORM
      END IF
C**********************************************
C**********************************************
      IMARK=0
      JCOUNT=0
      DO 150 IAREA=1,NAREA
      DO 140 ISRCE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      TCORR(ISRCE,1)=(0.0,0.0)
      TCORR(ISRCE,2)=(0.0,0.0)
C
      IF( NAREA .EQ. 1 .OR. NNSURF .EQ. 0 )GO TO 140
C
      JCOUNT=JCOUNT+1
      IF(JCOUNT .EQ. 10)THEN
      WRITE(*,*)'WORKING ON CORRCT FOR SOURCE POINT',ISRCE
      JCOUNT=0
      END IF
C
      IF(IAREA .NE. HALFSP)CALL CORRCT(ISRCE,X(ISRCE),Z(ISRCE),IAREA)
            IF(IAREA .EQ. HALFSP .AND.
     & ISRCE .NE. JNODE(HALFSP,1) .AND.
     & ISRCE .NE. KNODE(HALFSP,NBOUND(HALFSP)))THEN
      IF(IMARK .EQ. 0)CALL HSINTGR
C
C
      CALL HSCORR(ISRCE,X(ISRCE),Z(ISRCE),IAREA)
C
      IMARK = 1
      END IF
  140 CONTINUE
      DO 145 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IROW=2*INODE-1
      FVECT(IROW)=TCORR(INODE,1)
```

```
      FVECT(IROW+1)=TCORR(INODE,2)
  145 CONTINUE
  150 CONTINUE
      DO 170 INODE=JNODE(HALFSP,1),
     &               KNODE(HALFSP,NBOUND(HALFSP))
      IINODE=INODE-JNODE(HALFSP,1)+1
      IROW=2*INODE-1
      FVECT(IROW)=  UINCDT(IINODE)+FVECT(IROW)
      FVECT(IROW+1)=WINCDT(IINODE)+FVECT(IROW+1)
  170 CONTINUE
C
      IF( NAREA .EQ. 1 .AND. ICLOSE(NAREA,NBOUND(NAREA)) .EQ. 1)THEN
      GO TO 190
      END IF
C
                  IF(NNSURF .GT. 0 .AND. NAREA .GT. 1)THEN
C
      DO  ISIDE =1,2
C
            IF (NAREA .GT. 1)THEN
C
            DO ILAYER =1,NLAYER(ISIDE)
      IF(ISIDE .EQ. 1)INODE=JFF(ISIDE,ILAYER)
      IF(ISIDE .EQ. 2)INODE=KFF(ISIDE,ILAYER)
      IROW=2*NTOP(ISIDE,ILAYER)-1
      ICOUNT=0
      IF(NCONN(NTOP(ISIDE,ILAYER)) .EQ. 0)THEN
      NNODE=NTOP(ISIDE,ILAYER)
      ELSE
      NNODE=NCONN(NTOP(ISIDE,ILAYER))
      END IF
      DO I=1,NNODE
      IF(NCONN(I) .EQ. 0)ICOUNT=ICOUNT+1
      END DO
      JCOL=2*ICOUNT - 1
      FVECT(IROW)=FFU(ISIDE,INODE)
      FVECT(IROW+1)=FFW(ISIDE,INODE)
      DO J=1,ADIM
      HMAT(IROW,J)=(0.0,0.0)
      HMAT(IROW+1,J)=(0.0,0.0)
      END DO
      HMAT(IROW,JCOL)=(1.0,0.0)
      HMAT(IROW+1,JCOL+1)=(1.0,0.0)
C
            END DO
C
            END IF
C
      IF(ISIDE .EQ. 1)INODE=KNODE(HALFSP,NBOUND(HALFSP))
      IF(ISIDE .EQ. 2)INODE=JNODE(HALFSP,1)
      IROW=2*INODE-1
      IF(NCONN(INODE) .EQ. 0)THEN
      NNODE=INODE
      ELSE
      NNODE=NCONN(INODE)
      END IF
      ICOUNT=0
      DO I=1,NNODE
```

```
      IF(NCONN(I) .EQ. 0)ICOUNT=ICOUNT+1
      END DO
      JCOL=2*ICOUNT - 1
      FVECT(IROW)=HSU(ISIDE)*CEXP((0.0,1.0)*XKH1*X(INODE))
      FVECT(IROW+1)=HSW(ISIDE)*CEXP((0.0,1.0)*XKH1*X(INODE))
      DO 180 J=1,ADIM
      HMAT(IROW,J)=(0.0,0.0)
      HMAT(IROW+1,J)=(0.0,0.0)
  180 CONTINUE
      HMAT(IROW,JCOL)=(1.0,0.0)
      HMAT(IROW+1,JCOL+1)=(1.0,0.0)
C
            END DO
C
C
                  END IF
C
  190 CONTINUE
C     WRITE(10,1015)
C     WRITE(10,*)'FVECT'
C     DO 1050 I=1,2*KNODE(NAREA,NBOUND(NAREA))
C     WRITE(10,1040)I,FVECT(I)
C 1040 FORMAT(I5,5X,8F10.5)
C 1050 CONTINUE
C     CLOSE(UNIT=80)
      RETURN
      END




      SUBROUTINE AFORM
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/AMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX AMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
      INTEGER ACOL
C
C     [A]*{x} = {F}
C     FORM 'A' MATRIX
C     A [DISPLACEMENT/STRESS],F {DISPLACEMENT}
```

```
C
      ACOL=HDIM
      ADIM=ACOL
      IADD=0
      IF(NAREA .EQ. 1)GO TO 40
      DO 30INODE=1,KNODE(NAREA,NBOUND(NAREA))
C     SURFACE NODE WITH NO INTERFACE
      IF(IEQN(INODE) .EQ. 1 .AND. NSURF(INODE) .EQ. 1 )GO TO 30
      IF(IEQN(INODE) .GT. 0 )THEN
      IADD=IADD+1
      ACOL=HDIM+IADD*2-1
      DO 10IROW=1,2*KNODE(NAREA,NBOUND(NAREA))
      AMAT(IROW,ACOL)= -1.*GMAT(IROW,JCOL1(INODE))
      AMAT(IROW,ACOL+1)= -1.*GMAT(IROW,JCOL1(INODE)+1)
   10 CONTINUE
      IF(IEQN(INODE) .EQ. 3)THEN
      IADD=IADD+1
      ACOL=HDIM+IADD*2-1
      DO 20 IROW=1,2*KNODE(NAREA,NBOUND(NAREA))
      AMAT(IROW,ACOL)= -1.*GMAT(IROW,JCOL1(INODE)+2)
      AMAT(IROW,ACOL+1)= -1.*GMAT(IROW,JCOL1(INODE)+3)
   20 CONTINUE
      END IF
      END IF
   30 CONTINUE
      ADIM=ACOL+1
   40 CONTINUE
      NDIM=ADIM
C
C     WRITE(10,*)'AMAT'
C     WRITE(10,*)'ADIM = ',ADIM
C     DO 50 I=1,2*KNODE(NAREA,NBOUND(NAREA))
C     INODE=I/2 + 1
C     WRITE(10,*)'ROW = ', ' INODE = ',INODE
C     WRITE(10,1000)(AMAT(I,J),J=1,ADIM)
C 1000 FORMAT(8F10.5)
C   50 CONTINUE
C     CLOSE(UNIT=80)
      RETURN
      END
```

```
      SUBROUTINE INFLU(ISRCE,XI,ZI,IAREA,JGMAT)
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5),R(2),XN(2)
      COMPLEX GMU(5,2,2),T(5,2,2),
     & PIK,SIK,F1H1,F2H1,F3H1,P1,P2,S1,S2,XKSR,XKPR,
     & H01KPR,H11KPR,H01KSR,H11KSR,H21KPR,H21KSR
C
      COMPLEX PIKR,SIKR,P1R,P2R,S1R,S2R,F2H1R,F3H1R
      PI=4.*ATAN(1.0)
      JADD=JGMAT
```

```
      CALC1=(1.-2.*POISS(IAREA))/(2.*(1.-POISS(IAREA)))
      CALC2=POISS(IAREA)/(1.-POISS(IAREA))
      CPCSR= 1.0/SQRT(CALC1)
C     CALCULATE INFLUENCE FUNCTIONS
      DO 70  IBOUND=1,NBOUND(IAREA)
      JNO=JNODE(IAREA,IBOUND)
      JNDCOL=2*(JNODE(IAREA,IBOUND)-1)+1
C     WRITE(10,*)'INODE = ',INODE
C
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      NNODE=KNODE(IAREA,IBOUND)-1
      ELSE
      NNODE=KNODE(IAREA,IBOUND)
      END IF
C
      DO 60 INODE=JNODE(IAREA,IBOUND),NNODE
      ISEG=INODE
      X1=X(INODE)
      Z1=Z(INODE)
C
      IF(INODE .NE. KNODE(IAREA,IBOUND))THEN
      X2=X(INODE+1)
      Z2=Z(INODE+1)
      ELSE
      X2=X(JNO)
      Z2=Z(JNO)
      END IF
C
      SEGLEN=SQRT((X2-X1)**2+(Z2-Z1)**2)
      XN(1)=(Z1-Z2)/SEGLEN
      XN(2)=(X2-X1)/SEGLEN
C
      MARK=0
      IF (ISEG .EQ. ISRCE -1 ) MARK = 1
      IF (ISEG .EQ. ISRCE ) MARK = 2
      IF (ISRCE .EQ. JNO .AND. ISEG .EQ. KNODE(IAREA,IBOUND) ) MARK=1
C     MARK = 1, ADJACENT ELEMENT PRIOR TO ISRCE
C     MARK = 2, ADJACENT ELEMENT AFTER ISRCE

      IF (MARK .EQ. 1 .OR. MARK .EQ. 2)THEN
      CALL AADJSEG(IAREA,SEGLEN,CPCSR,P1,P2,S1,S2,P1R,P2R,
     & S1R,S2R,F2H1,F3H1,F2H1R,F3H1R)
C
C     WRITE(*,*)'ISEG = ',ISEG
      IF (MARK .EQ. 1)THEN
      R(1)= -XN(2)
      R(2)= +XN(1)
      ELSE
      R(1)= +XN(2)
      R(2)= -XN(1)
      END IF
      DO I =1,2
      DO K = 1,2
      PIK=  P2*R(I)*R(K)
      SIK=  S2*R(I)*R(K)
      PIKR= P2R*R(I)*R(K)
      SIKR= S2R*R(I)*R(K)
C
```

```
      IF (I .EQ. K) THEN
      PIK = P1 + PIK
      SIK = S1 + SIK
      PIKR= P1R+ PIKR
      SIKR= S1R + SIKR
      END IF
      GMU(1,I,K)=CALC1*PIK+SIK
      GMU(2,I,K)=CALC1*PIKR+SIKR
C
      T(1,I,K)=XN(I)*R(K)*F2H1 + XN(K)*R(I)*F3H1
      T(2,I,K)=XN(I)*R(K)*F2H1R + XN(K)*R(I)*F3H1R

      END DO
      END DO

      ELSE


      DO 40IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(X2-X1)/2.0+X1
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
      XKSR=FBAR*RADIUS(IGAUSS)/CSRAT(IAREA)
      XKPR=FBAR*RADIUS(IGAUSS)/(CPCSR*CSRAT(IAREA))
C
      XMAG1=SQRT((REAL(XKSR))**2 +(AIMAG(XKSR))**2)
      IF(XMAG1 .LT. 0.010 )THEN
      P1= -1.0/(4.0*PI) * CLOG(XKPR) +
     & (0.0,0.125)*(XKPR*XKPR/8.0 + 1.0)
      P2 = -1.0*(0.0,1.0)/32.0*XKPR*XKPR
      S1= -1.0/(4.0*PI) * CLOG(XKSR) +
     & (0.0,0.125)*(1.0-XKSR*XKSR/8.0)
      S2 = +1.0*(0.0,1.0)/32.0*XKSR*XKSR
      F1H1= -1.0/PI * (CALC1 - 1.0)
      F2H1= 1.0/PI * (CALC1-0.5) +
     & (0.0,1.0)/16.0*(-2.0*XKPR*XKPR +3.0* CALC1*XKPR*XKPR+XKSR*XKSR)
      F3H1=-1.0/(2.0*PI) -
     & (0.0,1.0)/16.0*(CALC1*XKPR*XKPR+XKSR*XKSR)
      F1H1=F1H1/RADIUS(IGAUSS)
      F2H1=F2H1/RADIUS(IGAUSS)
      F3H1=F3H1/RADIUS(IGAUSS)
C
      ELSE
C
      CALL HANKEL(0,XKPR,H01KPR)
      CALL HANKEL(1,XKPR,H11KPR)
      CALL HANKEL(0,XKSR,H01KSR)
      CALL HANKEL(1,XKSR,H11KSR)
      H21KPR=2.*H11KPR/XKPR - H01KPR
      H21KSR=2.*H11KSR/XKSR - H01KSR
      P1=(0.0,+0.1250)*(H21KPR+H01KPR)-1./(2.0*PI*XKPR*XKPR)
      P2=(0.0,-0.250)*H21KPR + 1./(PI*XKPR*XKPR)
      S1=(0.0,+0.1250)*(-H21KSR+H01KSR)+1./(2.0*PI*XKSR*XKSR)
      S2=(0.0,+0.250)*H21KSR - 1./(PI*XKSR*XKSR)
C
      F1H1=(0.0,1.0)*(CALC1*(2.*H21KPR -0.5*XKPR*H11KPR)
     & -2.0*H21KSR + 0.5*XKSR*H11KSR)/RADIUS(IGAUSS)
```

```
      F2H1=(0.0,0.25)*((2.0*CALC1-1.0)*XKPR*H11KPR
     & - 2.0*CALC1*H21KPR + 2.0*H21KSR)/RADIUS(IGAUSS)
      F3H1=(0.0,0.25)*
     & (-2.0*CALC1*H21KPR + 2.0*H21KSR - XKSR*H11KSR)/RADIUS(IGAUSS)
C
      END IF
C
      R(1)=XGAUSS(IGAUSS)-XI
      R(2)=ZGAUSS(IGAUSS)-ZI
      RJNJ=0.0
      DO 20J=1,2
      RJNJ=R(J)*XN(J)+RJNJ
   20 CONTINUE
C     WRITE(10,*)'ISRCE = ',ISRCE
      DO 30 I=1,2
      DO 30 K=1,2
      PIK= P2*R(I)*R(K)/(RADIUS(IGAUSS)**2)
      SIK= S2*R(I)*R(K)/(RADIUS(IGAUSS)**2)
C
      IF(I .EQ. K)THEN
      PIK=PIK+P1
      SIK=SIK+S1
      END IF
C
      GMU(IGAUSS,I,K)=CALC1*PIK+SIK
C
      T(IGAUSS,I,K)=( XN(I)*R(K)*F2H1
     &               +    XN(K)*R(I)*F3H1
     &               +    RJNJ*R(I)*R(K)*F1H1/(RADIUS(IGAUSS)**2) )
C
        IF(I .EQ. K)THEN
      T(IGAUSS,I,K)=T(IGAUSS,I,K)+
     &               +     RJNJ*F3H1
      END IF
C
      T(IGAUSS,I,K)=T(IGAUSS,I,K)/RADIUS(IGAUSS)
C
   30 CONTINUE
   40 CONTINUE


      END IF


C     IF INODE IS PART OF 3 NODE INTERFACE, INCLUDE P1 & P2 ON BOTH
SIDES OF NODE
      IF(NN3 .EQ. 0) GO TO 47
      DO 45I=1,NN3
      DO 45J=1,3
      IF(N3(I,J) .EQ. INODE) THEN
      JADD =JADD+2
      GO TO 47
      END IF
   45 CONTINUE
   47 CONTINUE
      JCOL=2*(INODE-1)+1+JADD
      JCOL1(INODE)=JCOL
      JCOL1(INODE+1)=JCOL+2
```

```
      JJCOL=2*(INODE-1)+1
      IF(INODE .EQ. NNODE .AND. ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
C
      KCOL=JCOL1(JNO)
      KKCOL=JNDCOL
C
      ELSE
C
      KCOL=JCOL+2
      KKCOL=JJCOL+2
C
      END IF
C
      DO 50 K=1,2
      IROW=2*(ISRCE-1)+K
C
C
      IF (MARK .EQ. 0)THEN
C
      GMAT(IROW,JCOL)=GMAT(IROW,JCOL)
     &   + (WT3(1)*WTFN3(1)*GMU(1,1,K)+WT3(2)*WTFN3(2)*GMU(2,1,K)
     &   +  WT3(1)*WTFN3(3)*GMU(3,1,K))*SEGLEN/2.
C
      GMAT(IROW,JCOL+1)=GMAT(IROW,JCOL+1)
     &   + (WT3(1)*WTFN3(1)*GMU(1,2,K)+WT3(2)*WTFN3(2)*GMU(2,2,K)
     &   +  WT3(1)*WTFN3(3)*GMU(3,2,K))*SEGLEN/2.
C
      HMAT(IROW,JJCOL)= HMAT(IROW,JJCOL)
     &   + (WT3(1)*WTFN3(1)*T(1,1,K)+WT3(2)*WTFN3(2)*T(2,1,K)
     &   +  WT3(1)*WTFN3(3)*T(3,1,K))*SEGLEN/2.
C
      HMAT(IROW,JJCOL+1)=HMAT(IROW,JJCOL+1)+
     &   + (WT3(1)*WTFN3(1)*T(1,2,K)+WT3(2)*WTFN3(2)*T(2,2,K)
     &   +  WT3(1)*WTFN3(3)*T(3,2,K))*SEGLEN/2.
C
C
      GMAT(IROW,KCOL)=GMAT(IROW,KCOL)
     &   + (WT3(1)*WTFN3(3)*GMU(1,1,K)+WT3(2)*WTFN3(2)*GMU(2,1,K)
     &   +  WT3(1)*WTFN3(1)*GMU(3,1,K))*SEGLEN/2.
C
      GMAT(IROW,KCOL+1)=GMAT(IROW,KCOL+1)
     &   + (WT3(1)*WTFN3(3)*GMU(1,2,K)+WT3(2)*WTFN3(2)*GMU(2,2,K)
     &   +  WT3(1)*WTFN3(1)*GMU(3,2,K))*SEGLEN/2.
C
      HMAT(IROW,KKCOL)=HMAT(IROW,KKCOL)
     &   + (WT3(1)*WTFN3(3)*T(1,1,K)+WT3(2)*WTFN3(2)*T(2,1,K)
     &   +  WT3(1)*WTFN3(1)*T(3,1,K))*SEGLEN/2.
      HMAT(IROW,KKCOL+1)=HMAT(IROW,KKCOL+1)+
     &   + (WT3(1)*WTFN3(3)*T(1,2,K)+WT3(2)*WTFN3(2)*T(2,2,K)
     &   +  WT3(1)*WTFN3(1)*T(3,2,K))*SEGLEN/2.
C
      ELSE
      IF (MARK .EQ. 1)THEN
C     ELEMENT ADJACENT TO SOURCE AND BEFORE ISRCE
C
      GMAT(IROW,JCOL)=GMAT(IROW,JCOL)
     &   +1.0/SEGLEN * GMU(2,1,K)
      GMAT(IROW,JCOL+1)=GMAT(IROW,JCOL+1)
```

```
     &   +1.0/SEGLEN * GMU(2,2,K)
C
      HMAT(IROW,JJCOL)= HMAT(IROW,JJCOL)
     &   +1.0/SEGLEN * T(2,1,K)
      HMAT(IROW,JJCOL+1)=HMAT(IROW,JJCOL+1)
     &   +1.0/SEGLEN * T(2,2,K)
C
      GMAT(IROW,KCOL)=GMAT(IROW,KCOL)
     &   +GMU(1,1,K) -1.0/SEGLEN * GMU(2,1,K)
      GMAT(IROW,KCOL+1)=GMAT(IROW,KCOL+1)
     &   +GMU(1,2,K) -1.0/SEGLEN * GMU(2,2,K)
C
      HMAT(IROW,KKCOL)=HMAT(IROW,KKCOL)
     &   +T(1,1,K) -1.0/SEGLEN * T(2,1,K)
      HMAT(IROW,KKCOL+1)=HMAT(IROW,KKCOL+1)+
     &   +T(1,2,K) -1.0/SEGLEN * T(2,2,K)
C
      ELSE
C
C     MARK = 2, ELEMENT ADJACENT TO SOURCE AND AFTER ISRCE
C
      GMAT(IROW,JCOL)=GMAT(IROW,JCOL)
     &   +GMU(1,1,K) -1.0/SEGLEN * GMU(2,1,K)
      GMAT(IROW,JCOL+1)=GMAT(IROW,JCOL+1)
     &   +GMU(1,2,K) -1.0/SEGLEN * GMU(2,2,K)
C
      HMAT(IROW,JJCOL)= HMAT(IROW,JJCOL)
     &   +T(1,1,K) -1.0/SEGLEN * T(2,1,K)
      HMAT(IROW,JJCOL+1)=HMAT(IROW,JJCOL+1)
     &   +T(1,2,K) -1.0/SEGLEN * T(2,2,K)
C
C
      GMAT(IROW,KCOL)=GMAT(IROW,KCOL)
     &   +1.0/SEGLEN * GMU(2,1,K)
      GMAT(IROW,KCOL+1)=GMAT(IROW,KCOL+1)
     &   +1.0/SEGLEN * GMU(2,2,K)
C
      HMAT(IROW,KKCOL)=HMAT(IROW,KKCOL)
     &   +1.0/SEGLEN * T(2,1,K)
      HMAT(IROW,KKCOL+1)=HMAT(IROW,KKCOL+1)
     &   +1.0/SEGLEN * T(2,2,K)
C
      END IF
      END IF
C
   50 CONTINUE
   60 CONTINUE
   70 CONTINUE
      IF(ISRCE .EQ. KNODE(IAREA,NBOUND(IAREA)))THEN
      JGMAT=JADD
      END IF
      RETURN
      END
```

```
      SUBROUTINE HANKEL(IORDER,Z,HOK)
C     *********************************************
C     HANKEL FUNCTION OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C           0 KIND
C     *********************************************
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT(2),DIFF,ADELTA,PI,XREAL,YIMAG,CALC,
     &  AA,A,B
      COMPLEX HOK,Z,JN,YN
C
C     WRITE(*,9000)
C 9000 FORMAT(2X,'INPUT ORDER',I2)
C     READ(*,*)IORDER
C     WRITE(*,9010)
C 9010 FORMAT(2X,'REAL PART OF kr')
C     READ(*,*)XREAL
C     WRITE(*,9020)
C 9020 FORMAT(2X,'IMAGINARY PART OF kr')
C     READ(*,*)YIMAG
C     Z=CMPLX(XREAL,YIMAG)
C
      PI=3.1415926535D0
      TOLER=.000001D0
C
      XREAL=REAL(Z)
      YIMAG=AIMAG(Z)
C     CALCULATE BESSEL FUNCTION OF 1ST KIND (JN)
      A=0.0D0
      B=PI
      DO 25ITYPE=1,2
      HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   10 CONTINUE
      ICYCLE=ICYCLE+1
      ATOT(ITYPE)=0.0
      N=(B-A)/HDELTA
      DO 20I=1,N
      AA= (I-1)*HDELTA+A
      DO 15J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))*DCOSH(YIMAG* DSIN(X(J)))
      ELSE
      Y(J)=DSIN(XREAL* DSIN(X(J))-IORDER*X(J))* DSINH(YIMAG*DSIN(X(J)))
      END IF
   15 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
```

```
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   20 CONTINUE
C     N1=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 10
      END IF
   25 CONTINUE
      JN=(ATOT(1)-(0.0,1.0)*ATOT(2))/PI
C     CALCULATE BESSEL FUNCTION OF 2ND KIND, ORDER N (YN)
      A=0.
      B=PI
      DO 55ITYPE=1,2
      HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   30 CONTINUE
      ICYCLE=ICYCLE+1
      N=(B-A)/HDELTA
      ATOT(ITYPE)=0.0
      DO 50I=1,N
      AA= (I-1)*HDELTA+A
      DO 40J=1,NGAUSS
      X(J)=AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)= DSIN(XREAL* DSIN(X(J))-IORDER*X(J))*DCOSH(YIMAG* DSIN(X(J)))
      ELSE
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))*DSINH(YIMAG* DSIN(X(J)))
      END IF
   40 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   50 CONTINUE
C     N2=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 30
      END IF
   55 CONTINUE
      YN=ATOT(1)+(0.0,1.0)*ATOT(2)
      A=0.
      B=10.
      DO 90ITYPE=1,2
      HDELTA=1.0
C     HDELTA=B-A
      AOLD=9.99999D20
      ICYCLE=0
   60 CONTINUE
      ICYCLE=ICYCLE+1
      ATOT(ITYPE)=0.0
      N=(B-A)/HDELTA
      DO 80I=1,N
```

```
      AA= (I-1)*HDELTA+A
      DO 70J=1,NGAUSS
      X(J)=AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALC=XREAL*DSINH(X(J))
      IF(CALC .GT. 40.)CALC=40.0D0
      IF(ITYPE .EQ. 1) THEN
      Y(J)=(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)*DCOS(YIMAG*DSINH(X(J)))
      ELSE
      Y(J)=(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)*DSIN(YIMAG*DSINH(X(J)))
      END IF
   70 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT(ITYPE)=ATOT(ITYPE)+ADELTA
   80 CONTINUE
C     N3=N
      DIFF=ABS(ATOT(ITYPE)-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT(ITYPE)
      GO TO 60
      END IF
   90 CONTINUE
      YN=(YN-ATOT(1)+(0.0,1.0)*ATOT(2))/PI
      HOK=JN+(0.0,1.0)*YN
C     WRITE(*,9030)
C 9030 FORMAT(5X,'HANKEL FUNCTION ')
C     WRITE(*,9040)IORDER
C 9040 FORMAT(5X,'ORDER = ',I1,3X,'KIND = 0')
C     WRITE(*,9050)HOK
C 9050 FORMAT(5X,F10.5,2X,' + i',F10.5)
C     WRITE(*,9060)N1
C 9060 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N1 = ',I4)
C     WRITE(*,9070)N2
C 9070 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N2 = ',I4)
C     WRITE(*,9080)N3
C 9080 FORMAT(5X,'# OF INTEGRAL SEGMENTS,N3 = ',I4)
C     PAUSE
      RETURN
      END
```

```
      SUBROUTINE ROTATE (X1,Z1,X2,Z2,XI,ZI,PHI1,PHI2)
C     CALCULATE ANGLE BETWEEN X1,Z1 AND X AXIS (PHI1)
C     CALCULATE ANGLE BETWEEN X1,Z1 AND X AXIS (PHI2)
C
      REAL X1DIFF,Z1DIFF,X2DIFF,Z2DIFF,PHI1,PHI2
C
      PI=3.14159265
      X1DIFF=X1-XI
      X2DIFF=X2-XI
      Z1DIFF=Z1-ZI
      Z2DIFF=Z2-ZI
      IF (ABS(X1DIFF) .LT. 0.000001)THEN
            IF (Z1DIFF .GT. 0.0)THEN
            PHI1=PI/2.0
            ELSE
            PHI1=1.5*PI
            END IF
      ELSE
      PHI1 = ATAN(Z1DIFF/X1DIFF)
      CALL QUAD(X1DIFF,Z1DIFF,PHI1)
      END IF
C
      IF (ABS(X2DIFF) .LT. 0.000001)THEN
            IF (Z2DIFF .GT. 0.0)THEN
            PHI2=PI/2.0
            ELSE
            PHI2=1.5 *PI
            END IF
      ELSE
      PHI2 = ATAN(Z2DIFF/X2DIFF)
      CALL QUAD(X2DIFF,Z2DIFF,PHI2)
      END IF
      IF (PHI2 .GT. PHI1)PHI1=PHI1+2.0*PI
      RETURN
      END


      SUBROUTINE QUAD(X,Z,ANGLE)
      REAL ANGLE,X,Z
C     DETERMINE QUADRANT IN WHICH ANGLE IS LOCATED
      PI=3.14159265
      ANGLE=ABS(ANGLE)
C     QUADRANT 2
      IF (X .LT. 0. .AND. Z .GE. 0.) ANGLE = PI-ANGLE
C     QUADRANT 3
      IF (X .LT. 0. .AND. Z .LT. 0.) ANGLE = PI+ANGLE
C     QUADRANT 4
      IF (X .GE. 0. .AND. Z .LT. 0.) ANGLE = 2.*PI-ANGLE
      RETURN
      END
```

```
      SUBROUTINE STATIC(ISRCE,XI,ZI,IAREA,DSTAT,DCORR)
C
C     DETERMINE STATIC DIAGONAL TERMS
C
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
```

```
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5),R(2),XN(2)
      COMPLEX T(5,2,2),DSTAT(2,2),DCORR(2,2)
C
      PI=4.*ATAN(1.0)
      CALC1=  -1.0/((1.-POISS(IAREA))*4.0*PI)
      CALC2= 1.- 2.0* POISS(IAREA)
C
      DO I=1,2
      DO J=1,2
      DSTAT(I,J)=(0.0,0.0)
      DCORR(I,J)=(0.0,0.0)
      END DO
      END DO
C     CALCULATE INFLUENCE FUNCTIONS
      DO 70  IBOUND=1,NBOUND(IAREA)
      JNO=JNODE(IAREA,IBOUND)
      KNO=KNODE(IAREA,IBOUND)
C
      IF(ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      NELEM=KNO-JNO
      ELSE
      NELEM=KNO-JNO+1
      END IF
C
      ISKIP=0
      NSKIP = 0
      DO 60 IELEM=1,NELEM
C
      IF (ISKIP .EQ. 1)THEN
      ISKIP = 0
      GO TO 60
      END IF
      IF (NSKIP .EQ. 1 .AND. IELEM .EQ. NELEM)THEN
      NSKIP = 0
      GO TO 60
      END IF
C
      INODE=JNO+IELEM-1
      IF(ICLOSE(IAREA,IBOUND) .EQ. 1 .AND. IELEM .EQ. NELEM)THEN
      IINODE =JNO
      ELSE
      IINODE = INODE + 1
      END IF
C
      X1=X(INODE)
      Z1=Z(INODE)
C
      X2=X(IINODE)
      Z2=Z(IINODE)
C
      SEGLEN=SQRT((X2-X1)**2+(Z2-Z1)**2)
      XN(1)=(Z1-Z2)/SEGLEN
      XN(2)=(X2-X1)/SEGLEN
```

```
C
      IF (INODE .EQ. ISRCE .OR. IINODE .EQ. ISRCE)THEN
      IF (ISRCE .EQ. JNO .AND. ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      DSTAT(1,2)=DSTAT(1,2)-CALC1*CALC2
      DSTAT(2,1)=DSTAT(2,1)+CALC1*CALC2
      DCORR(1,2)= -1.0*CALC1*CALC2*LOG(SEGLEN)
      DCORR(2,1)= +1.0*CALC1*CALC2*LOG(SEGLEN)
      GO TO 60
      END IF
      IF (ISRCE .EQ. KNO .AND. ICLOSE(IAREA,IBOUND) .EQ. 0)THEN
      DSTAT(1,2)=DSTAT(1,2)+CALC1*CALC2
      DSTAT(2,1)=DSTAT(2,1)-CALC1*CALC2
      DCORR(1,2)= +1.0*CALC1*CALC2*LOG(SEGLEN)
      DCORR(2,1)= -1.0*CALC1*CALC2*LOG(SEGLEN)
      GO TO 60
      END IF
      IF (ISRCE .EQ. JNO .AND. ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
      SEGL1=SQRT((X(ISRCE)-X(KNO))**2+(Z(ISRCE)-Z(KNO) )**2)
      NSKIP = 1
      ELSE
      SEGL1=SQRT((X(ISRCE)-X(ISRCE-1))**2+(Z(ISRCE)-Z(ISRCE-1) )**2)
      ISKIP = 1
      END IF
      IF (ISRCE .EQ. KNO .AND. ICLOSE(IAREA,IBOUND) .EQ. 1)THEN
      SEGL2=SQRT((X(JNO)-X(ISRCE))**2+(Z(JNO)-Z(ISRCE))**2)
      ELSE
      SEGL2=SQRT((X(ISRCE+1)-X(ISRCE))**2+(Z(ISRCE+1)-Z(ISRCE) )**2)
      END IF
      DCORR(1,2)= -1.0*CALC1*CALC2*LOG(SEGL2/SEGL1)
      DCORR(2,1)= +1.0*CALC1*CALC2*LOG(SEGL2/SEGL1)
      GO TO 60
C
      ELSE

5     CONTINUE

      DO 40IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(X2-X1)/2.0+X1
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
C
      R(1)=XGAUSS(IGAUSS)-XI
      R(2)=ZGAUSS(IGAUSS)-ZI
C     WRITE(10,*)'ISRCE = ',ISRCE
C
      RJNJ=0.0
      DO 20J=1,2
      RJNJ=R(J)*XN(J)+RJNJ
   20 CONTINUE
C
      DO 30 I=1,2
      DO 30 K=1,2
      T(IGAUSS,I,K)=(-XN(I)*R(K)
     &                +   XN(K)*R(I))*CALC2
     &                +   (2.0*R(I)*R(K))/(RADIUS(IGAUSS)**2)*RJNJ
C
        IF(I .EQ. K)THEN
```

```
      T(IGAUSS,I,K)=T(IGAUSS,I,K)+
     &              +    CALC2*RJNJ
      END IF
C
      T(IGAUSS,I,K)=T(IGAUSS,I,K)*CALC1/(RADIUS(IGAUSS)**2 )
C
   30 CONTINUE
   40 CONTINUE
C
C
      DO  K=1,2
      DSTAT(K,1)= DSTAT(K,1)
     &  + (WT3(1)*T(1,1,K)+WT3(2)*T(2,1,K)
     &  +  WT3(1)*T(3,1,K))*SEGLEN/2.
C
      DSTAT(K,2)=DSTAT(K,2)
     &  + (WT3(1)*T(1,2,K)+WT3(2)*T(2,2,K)
     &  +  WT3(1)*T(3,2,K))*SEGLEN/2.
      END DO
C
   50 CONTINUE
      END IF
   60 CONTINUE
   70 CONTINUE
C
      IF (NLAYER(1) .GT. 0 .OR. NLAYER(2) .GT. 0)THEN
      DO ISIDE =1,2
      LAYER=0
C
                  IF(NBOUND(IAREA) .EQ. 1)THEN
      DO ILAYER=1,NLAYER(ISIDE)
      N1=NTOP(ISIDE,ILAYER)
      N2=NBOTT(ISIDE,ILAYER)
      IF(N1 .GT. N2)THEN
      NTEMP=N1
      N1=N2
      N2=NTEMP
      END IF
      IF(ISRCE .GE. N1 .AND. ISRCE .LE. N2)THEN
      LAYER=ILAYER
      GO TO 75
      END IF
      END DO
      GO TO 200
   75 CONTINUE
                  ELSE
C
            DO ILAYER =1,NLAYER(ISIDE)
      IF(LTYPE(ISIDE,ILAYER) .EQ. IAREA) LAYER=ILAYER
            END DO
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      IF(LAYER .EQ. 0) THEN
      WRITE(*,*)'PROBLEM IN SUBROUTINE CORRECT'
      PAUSE
      STOP
      END IF
                  END IF
C
```

```
C
      DO 150 INODE=JFF(ISIDE,LAYER),KFF(ISIDE,LAYER)-1
C
      DIST1=SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,INODE))**2)
      DIST2=SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,INODE+1))**2)
      IF (DIST1 .LT. 0.00001)THEN
      SEGLEN= SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,INODE+1))**2)
      DSTAT(1,2)=DSTAT(1,2)-CALC1*CALC2
      DSTAT(2,1)=DSTAT(2,1)+CALC1*CALC2
      DCORR(1,2)= DCORR(1,2)-CALC1*CALC2*LOG(SEGLEN)
      DCORR(2,1)= DCORR(2,1) +CALC1*CALC2*LOG(SEGLEN)
      GO TO 150
      END IF
C
      IF (DIST2 .LT. 0.00001)THEN
      SEGLEN= SQRT((XI-XFF(ISIDE,LAYER))**2 +(ZI-ZFF(ISIDE,INODE))**2)
      DSTAT(1,2)=DSTAT(1,2)+CALC1*CALC2
      DSTAT(2,1)=DSTAT(2,1)-CALC1*CALC2
      DCORR(1,2)= DCORR(1,2)+CALC1*CALC2*LOG(SEGLEN)
      DCORR(2,1)= DCORR(2,1)-CALC1*CALC2*LOG(SEGLEN)
      GO TO 150
      END IF
C
      SEGLEN=ABS(ZFF(ISIDE,INODE+1)-ZFF(ISIDE,INODE))
      XN(1)=(ZFF(ISIDE,INODE)-ZFF(ISIDE,INODE+1))/SEGLEN
      XN(2)=0.0
C
      Z1=ZFF(ISIDE,INODE)
      Z2=ZFF(ISIDE,INODE+1)
      DO 140IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=XFF(ISIDE,LAYER)
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
C
      R(1)=XGAUSS(IGAUSS)-XI
      R(2)=ZGAUSS(IGAUSS)-ZI
      RJNJ=0.0
      DO J=1,2
      RJNJ=R(J)*XN(J)+RJNJ
      END DO
C     WRITE(10,*)'ISRCE = ',ISRCE
      DO 130 I=1,2
      DO 130 K=1,2
      T(IGAUSS,I,K)=(-XN(I)*R(K)
     &              +   XN(K)*R(I))*CALC2
     &              +   (2.0*R(I)*R(K))/(RADIUS(IGAUSS)**2)*RJNJ
C
        IF(I .EQ. K)THEN
      T(IGAUSS,I,K)=T(IGAUSS,I,K)+
     &              +    CALC2*RJNJ
      END IF
C
      T(IGAUSS,I,K)=T(IGAUSS,I,K)*CALC1/(RADIUS(IGAUSS)**2 )
C
C
  130 CONTINUE
  140 CONTINUE
```

```
C
             DO 150K=1,2
       DSTAT(K,1)= DSTAT(K,1)
     &   + (WT3(1)*T(1,1,K)+WT3(2)*T(2,1,K)
     &   +  WT3(1)*T(3,1,K))*SEGLEN/2.
C
       DSTAT(K,2)=DSTAT(K,2)
     &   + (WT3(1)*T(1,2,K)+WT3(2)*T(2,2,K)
     &   +  WT3(1)*T(3,2,K))*SEGLEN/2.
C
  150 CONTINUE
C
C
  200 CONTINUE
            END DO
C
      END IF
C
      DO K=1,2
      DO J=1,2
      DSTAT(K,J)= -DSTAT(K,J)
      END DO
      END DO
      IF(IAREA .EQ. HALFSP)THEN
      DSTAT(1,1)=(1.0,0.0)+DSTAT(1,1)
      DSTAT(2,2)=(1.0,0.0)+DSTAT(2,2)
      END IF
C
C
C     WRITE(*,*)'ISRCE = ',ISRCE
C     WRITE(*,9000)
C 9000 FORMAT('DSTAT')
C     DO I=1,2
C     WRITE(*,9010)(DSTAT(I,J),J=1,2)
C 9010 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C
C
C     WRITE(*,9020)
 9020 FORMAT(/'DCORR')
C     WRITE(*,*)'SEGL1 = ',SEGL1
C     WRITE(*,*)'SEGL2 = ',SEGL2
C     DO I=1,2
C     WRITE(*,9030)(DCORR(I,J),J=1,2)
C 9030 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C     WRITE(*,9040)
C 9040 FORMAT(/'Cij')
C     DO I=1,2
C     WRITE(*,9050)((DSTAT(I,J)-DCORR(I,J)),J=1,2)
C 9050 FORMAT(2F10.3,5X,2F10.3)
C     END DO
C     PAUSE
C
C

      RETURN
      END
```

```
C23456789112345678921234567893123456789412345678951234567896123456789712
      SUBROUTINE AADJSEG(IAREA,SEGLEN,CPCSR,P1,P2,S1,S2,P1R,P2R,
     & S1R,S2R,F2H1,F3H1,F2H1R,F3H1R)
C
              COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMPLEX FBAR,UINCDT,WINCDT
C
      COMMON/CSOIL/CSRAT(4)
      COMPLEX CSRAT
C
      DOUBLE PRECISION PI,SEG(11),EPSLN
C     DOUBLE PRECISION PI,XREAL,XIMAG,XMAG,PHI,SEG(11),EPSLN
      COMPLEX*16 ALPHA(11),AJN,AYN,AH01(2),AH11(2),AH21(2),
     &  ARH01(2),ARH11(2),ARH21(2),AH21R(2),AKR2(2),ARKR2(2),
     & ZETA,A1,A2,A3
      COMPLEX P1,P2,S1,S2,P1R,P2R,S1R,S2R,F2H1,F3H1,F2H1R,F3H1R
C
      REAL KPKSR2
C
C
C     SOLVE ANALYTICALLY
C
      PI = 4.0*DATAN(1.0D0)
C
C     CPCSR=1.7321
C     IAREA=1
C     FBAR=1.6 +(0.0,1.0)*0.05
C     CSRAT(1)=1.0
C     SEGLEN = 0.05
      EPSLN=0.00001D0
      XLNS= LOG(SEGLEN)
      KPKSR2=1.0/(CPCSR*CPCSR)
C
C
      SEG(1)=SEGLEN
      DO IPOWER=2,11
      ALPHA(IPOWER)=ALPHA(1)**IPOWER
      SEG(IPOWER)=SEGLEN**IPOWER
      END DO
C

              DO IWAVE=1,2
C
      IF(IWAVE .EQ. 1)ALPHA(1)=FBAR/(CPCSR*CSRAT(IAREA))
      IF(IWAVE .EQ. 2)ALPHA(1)=FBAR/CSRAT(IAREA)
C
      DO IPOWER=2,11
      ALPHA(IPOWER)=ALPHA(1)**IPOWER
      END DO
C
C     INTEGRATION OF J0(alpha*r)
C
      AJN = SEG(1) -ALPHA(2)*SEG(3)/12.0D0 +ALPHA(4)*SEG(5)/320.0D0
     &       -ALPHA(6)*SEG(7)/16128.0D0 +ALPHA(8)*SEG(9)/1327104.0D0
C
C     INTEGRATION OF Y0(alpha*r)
      A1=
     &  SEG(1)*(XLNS-1.0D0)- ALPHA(2)*SEG(3)/12.0D0*(XLNS-1.0D0/3.0D0)
```

```
     & +ALPHA(4)*SEG(5)/320.0D0*(XLNS-1.0D0/5.0D0)
     & -ALPHA(6)*SEG(7)/16128.0D0*(XLNS-1.0D0/7.0D0)
C
      ZETA=-0.115931516D0+CDLOG(ALPHA(1))
      A2=ZETA*SEG(1)-ALPHA(2)*SEG(3)/12.0D0*ZETA
     & +ALPHA(4)*SEG(5)/1280.0D0*(4.0D0*ZETA-6.0D0)
     & -ALPHA(6)*SEG(7)/193536.0D0*(12.0D0*ZETA-22.0D0)
      AYN=2.0D0/PI * (A1+A2)
C
      AH01(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF J1(alpha*r)
C
      AJN=ALPHA(1)*SEG(2)/4.0D0-ALPHA(3)*SEG(4)/64.0D0
     & +ALPHA(5)*SEG(6)/2304.0D0-ALPHA(7)*SEG(8)/147456.0D0
     & +ALPHA(9)*SEG(10)/14745600.0D0
C
C
C     INTEGRATION OF Y1(alpha*r)
      A1= -1.0D0/ALPHA(1) *           (XLNS-DLOG(EPSLN))
      A2= ALPHA(1)*SEG(2)/4.0D0*      (XLNS-1.0D0/2.0D0)
     & -     ALPHA(3)*SEG(4)/64.0D0*     (XLNS-1.0D0/4.0D0)
     & +     ALPHA(5)*SEG(6)/2304.0D0*   (XLNS-1.0D0/6.0D0)
     & -     ALPHA(7)*SEG(8)/147456.0D0*(XLNS-1.0D0/8.0D0)
C
      ZETA=-1.115931516D0 +CDLOG(ALPHA(1))
C
      A3=ALPHA(1)*SEG(2)/8.0D0*        (2.0D0*ZETA+1.0D0)
     &     -ALPHA(3)*SEG(4)/256.0D0*      (4.0D0*ZETA-1.0D0)
     &     +ALPHA(5)*SEG(6)/6912.0D0*     (3.0D0*ZETA-2.0D0)
     &     -ALPHA(7)*SEG(8)/24772608.0D0*(168.0D0*ZETA-161.0D0)
      AYN=2.0D0/PI * (A1+A2+A3)
C
      AH11(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF J2(alpha*r)
C
      AJN=ALPHA(2)*SEG(3)/24.0D0-ALPHA(4)*SEG(5)/480.0D0
     & +ALPHA(6)*SEG(7)/21504.0D0
C
C     INTEGRATION OF Y2(alpha*r)
      A1= (0.0D0,0.0D0)
     & +2.0D0/ALPHA(2) * (1.0D0/SEG(1)-1.0D0/EPSLN)
      A2= ALPHA(1)*SEG(3)/24.0D0*     (XLNS-1.0D0/3.0D0)
     & -     ALPHA(4)*SEG(5)/480.0D0*    (XLNS-1.0D0/5.0D0)
     & +     ALPHA(6)*SEG(7)/21504.0D0*   (XLNS-1.0D0/7.0D0)
C
      ZETA=-1.615931516D0 +CDLOG(ALPHA(1))
C
      A3= -0.5D0*SEG(1)
     &     +ALPHA(2)*SEG(3)/96.0D0*        (4.0D0*ZETA+3.0D0)
     &     -ALPHA(4)*SEG(5)/5760.0D0*      (12.0D0*ZETA+1.0D0)
     &     +ALPHA(6)*SEG(7)/516096.0D0*    (24.0D0*ZETA-7.0D0)
      AYN=2.0D0/PI * (A1+A2+A3)
C
      AH21(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF r*J0(alpha*r)
```

```
C
      AJN = SEG(2)/2.0D0 -ALPHA(2)*SEG(4)/16.0D0
     & +ALPHA(4)*SEG(6)/384.0D0
     &      -ALPHA(6)*SEG(8)/18432.0D0 +ALPHA(8)*SEG(10)/1474560.0D0
C
C     INTEGRATION OF r*Y0(alpha*r)
      A1= SEG(2)/2.0D0 *                (XLNS-1.0D0/2.0D0)
      A2=-ALPHA(2)*SEG(4)/16.0D0*      (XLNS-1.0D0/4.0D0)
     & +    ALPHA(4)*SEG(6)/384.0D0*    (XLNS-1.0D0/6.0D0)
     & -    ALPHA(6)*SEG(8)/18432.0D0* (XLNS-1.0D0/8.0D0)
C
      ZETA=-0.115931516D0 +CDLOG(ALPHA(1))
C
      A3= ZETA*SEG(2)/2.0D0
     &    -ALPHA(2)*SEG(4)/16.0D0*         (ZETA)
     &    +ALPHA(4)*SEG(6)/1536.0D0*      (4.0D0*ZETA-6.0D0)
     &    -ALPHA(6)*SEG(8)/221184.0D0*    (12.0D0*ZETA-22.0D0)
      AYN=2.0D0/PI * (A1+A2+A3)
C
      ARH01(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF r*J1(alpha*r)
C
      AJN=ALPHA(1)*SEG(3)/6.0D0-ALPHA(3)*SEG(5)/80.0D0
     & +ALPHA(5)*SEG(7)/2688.0D0-ALPHA(7)*SEG(9)/165888.0D0
     & +ALPHA(9)*SEG(11)/16220160.0D0

C
C     INTEGRATION OF r*Y1(alpha*r)
      A1= -SEG(1)/ALPHA(1)
      A2= ALPHA(1)*SEG(3)/6.0D0*      (XLNS-1.0D0/3.0D0)
     & -    ALPHA(3)*SEG(5)/80.0D0*     (XLNS-1.0D0/5.0D0)
     & +    ALPHA(5)*SEG(7)/2688.0D0* (XLNS-1.0D0/7.0D0)
     & -    ALPHA(7)*SEG(9)/165888.0D0*(XLNS-1.0D0/9.0D0)
C
C
      ZETA=-1.115931516D0 +CDLOG(ALPHA(1))
C
      A3=ALPHA(1)*SEG(3)/12.0D0*        (2.0D0*ZETA+1.0D0)
     &    -ALPHA(3)*SEG(5)/320.0D0*       (4.0D0*ZETA-1.0D0)
     &    +ALPHA(5)*SEG(7)/8064.0D0*      (3.0D0*ZETA-2.0D0)
     &    -ALPHA(7)*SEG(9)/27869184.0D0* (168.0D0*ZETA-161.0D0)
      AYN=2.0D0/PI * (A1+A2+A3)
C
      ARH11(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF r*J2(alpha*r)
C
      AJN=ALPHA(2)*SEG(4)/32.0D0-ALPHA(4)*SEG(6)/576.0D0
     & +ALPHA(6)*SEG(8)/24576.0D0
C
C     INTEGRATION OF r*Y2(alpha*r)
      A1= (0.0D0,0.0D0)
     &  -2.0D0/ALPHA(2) * ( XLNS -DLOG(EPSLN))
      A2= ALPHA(2)*SEG(4)/32.0D0*     (XLNS-1.0D0/4.0D0)
     & -    ALPHA(4)*SEG(6)/576.0D0*    (XLNS-1.0D0/6.0D0)
     & +    ALPHA(6)*SEG(8)/24576.0D0* (XLNS-1.0D0/8.0D0)
C
```

```
      ZETA=-1.615931516D0 +CDLOG(ALPHA(1))
C
      A3= -SEG(2)/4.0D0
     &    +ALPHA(2)*SEG(4)/128.0D0*       (4.0D0*ZETA+3.0D0)
     &    -ALPHA(4)*SEG(6)/6912.0D0*      (12.0D0*ZETA+1.0D0)
     &    +ALPHA(6)*SEG(8)/589824.0D0*     (24.0D0*ZETA-7.0D0)
C
      AYN=2.0D0/PI * (A1+A2+A3)
C
      ARH21(IWAVE)=AJN+(0.0,1.0)*AYN
C
C     INTEGRATION OF J2(alpha*r) / r
C
      AJN=ALPHA(2)*SEG(2)/16.0D0-ALPHA(4)*SEG(4)/384.0D0
     & +ALPHA(6)*SEG(6)/18432.0D0
C
C     INTEGRATION OF Y2(alpha*r) / r
      A1= -1.0D0/2.0D0 * (XLNS-DLOG(EPSLN))
C     & +1.0D0/ALPHA(2) * (1.0D0/SEG(2)-1.0/(EPSLN*EPSLN) )
      A2= ALPHA(2)*SEG(2)/16.0D0*    (XLNS-1.0D0/2.0D0)
     & -    ALPHA(4)*SEG(4)/384.0D0*   (XLNS-1.0D0/4.0D0)
     & +    ALPHA(6)*SEG(6)/18432.0D0* (XLNS-1.0D0/6.0D0)
C
      ZETA=-1.615931516D0 +CDLOG(ALPHA(1))
C
      A3=
     &    +ALPHA(2)*SEG(2)/64.0D0*       (4.0D0*ZETA+3.0D0)
     &    -ALPHA(4)*SEG(4)/4608.0D0*     (12.0D0*ZETA+1.0D0)
     &    +ALPHA(6)*SEG(6)/442368.0D0*   (24.0D0*ZETA-7.0D0)
C
      AYN=2.0D0/PI * (A1+A2+A3)
C
      AH21R(IWAVE)=AJN+(0.0,1.0)*AYN
C
      AKR2(IWAVE)= -1.0D0/(ALPHA(1)*ALPHA(1))*(1.0D0/SEG(1) -
1.0D0/EPSLN)
      ARKR2(IWAVE)= 1.0D0/(ALPHA(1)*ALPHA(1))*( DLOG(SEG(1)) -
DLOG(EPSLN))
C
            END DO
C
C
C     DISPLACEMENT INFLUENCE FUNCTION
C
      P1 = (0.0D0,0.125D0)*(AH21(1)+AH01(1))
     & - AKR2(1)/(2.0D0*PI)
      P2 = (0.0D0,-0.25D0)*AH21(1)
     & + AKR2(1)/PI
      S1 = (0.0D0,0.125D0)*(AH01(2)-AH21(2))
     & + AKR2(2)/(2.0D0*PI)
      S2 = (0.0D0,+0.25D0)*AH21(2)
     & - AKR2(2)/PI
C
      P1R = (0.0D0,0.125D0)*(ARH21(1)+ARH01(1))
     & - ARKR2(1)/(2.0D0*PI)
      P2R = (0.0D0,-0.25D0)*ARH21(1)
     & + ARKR2(1)/PI
      S1R = (0.0D0,0.125D0)*(ARH01(2)-ARH21(2))
```

```
     & + ARKR2(2)/(2.0D0*PI)
      S2R = (0.0D0,+0.25D0)*ARH21(2)
     & - ARKR2(2)/PI
C
C     TRACTION INFLUENCE FUNCTION
C
      F2H1=(0.0D0,0.25D0)*
     & ((2.0D0*KPKSR2-1.0D0)*FBAR/(CPCSR*CSRAT(IAREA))*AH11(1)
     & - 2.0D0*KPKSR2*AH21R(1) + 2.0D0*AH21R(2))
      F3H1=(0.0D0,0.25D0)*
     & (-2.0D0*KPKSR2*AH21R(1) + 2.0D0*AH21R(2)
     & -FBAR/CSRAT(IAREA)*AH11(2))
C
      F2H1R=(0.0D0,0.25D0)*
     & ((2.0D0*KPKSR2-1.0D0)*FBAR/(CPCSR*CSRAT(IAREA))*ARH11(1)
     & - 2.0D0*KPKSR2*AH21(1) + 2.0D0*AH21(2))
      F3H1R=(0.0D0,0.25D0)*(-2.0D0*KPKSR2*AH21(1)+2.0D0*AH21(2)
     & - FBAR/CSRAT(IAREA)*ARH11(2))
C
C
C     WRITE(*,*)'SEGLEN = ',SEG(1)
C     WRITE(*,*)'FBAR = ',FBAR
C
C     WRITE(*,*)'P1 = ',P1
C     WRITE(*,*)'P2 = ',P2
C     WRITE(*,*)'S1 = ',S1
C     WRITE(*,*)'S2 = ',S2
C     WRITE(*,*)'P1R = ',P1R
C     WRITE(*,*)'P2R = ',P2R
C     WRITE(*,*)'S1R = ',S1R
C     WRITE(*,*)'S2R = ',S2R
C
C     WRITE(*,*)'F2H1 = ',F2H1
C     WRITE(*,*)'F3H1 = ',F3H1
C     WRITE(*,*)'F2H1R = ',F2H1R
C     WRITE(*,*)'F3H1R = ',F3H1R
C
      RETURN
      END
```

```
      SUBROUTINE CORRCT(ISRCE,XI,ZI,IAREA)
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      COMMON/IGAUS3/NGAUS3
      COMMON/DGAUS3/WT3(2),WTFN3(3),XX3(3)
      INTEGER NGAUS3
      DOUBLE PRECISION WT3,WTFN3,XX3
C
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
      REAL XGAUSS(5),ZGAUSS(5),RADIUS(5),R(2),XN(2)
      COMPLEX GMU(5,2,2),T(5,2,2),
     & PIK,SIK,F1H1,F2H1,F3H1,P1,P2,S1,S2,XKSR,XKPR,
     & H01KPR,H11KPR,H01KSR,H11KSR,H21KPR,H21KSR,WCORR,STCORR
C
```

```
      COMPLEX PIKR,SIKR,P1R,P2R,S1R,S2R,F2H1R,F3H1R
C
C
      PI=4.*ATAN(1.0)
      TCORR(ISRCE,1)=(0.0,0.0)
      TCORR(ISRCE,2)=(0.0,0.0)
      STCORR=(0.0,0.0)
      WCORR=(0.0,0.0)
      CALC1=(1.-2.*POISS(IAREA))/(2.*(1.-POISS(IAREA)))
      CALC2=POISS(IAREA)/(1.-POISS(IAREA))
      CPCSR= 1.0/SQRT(CALC1)
C     WRITE(80,*)'ISRCE',ISRCE
C     CALCULATE INFLUENCE FUNCTIONS
      DO 60  ISIDE=1,2
      LAYER=0
C
            IF(NBOUND(IAREA) .EQ. 1)THEN
      DO 4ILAYER=1,NLAYER(ISIDE)
      N1=NTOP(ISIDE,ILAYER)
      N2=NBOTT(ISIDE,ILAYER)
      IF(N1 .GT. N2)THEN
      NTEMP=N1
      N1=N2
      N2=NTEMP
      END IF
      IF(ISRCE .GE. N1 .AND. ISRCE .LE. N2)THEN
      LAYER=ILAYER
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      GO TO 6
      END IF
    4 CONTINUE
      WRITE(*,*)'SIDE = ',ISIDE,' IS NOT INCLUDED IN CORRECTION'
      GO TO 60
    6 CONTINUE
            END IF
C
            IF(NBOUND(IAREA) .GT. 1)THEN
      DO 7ILAYER =1,NLAYER(ISIDE)
      IF(LTYPE(ISIDE,ILAYER) .EQ. IAREA) LAYER=ILAYER
    7 CONTINUE
C     WRITE(80,*)'ISIDE,LAYER',ISIDE,LAYER
      IF(LAYER .EQ. 0) THEN
      WRITE(*,*)'PROBLEM IN SUBROUTINE CORRECT'
      PAUSE
      STOP
      END IF
            END IF
C
      DO 50 INODE=JFF(ISIDE,LAYER),KFF(ISIDE,LAYER)-1
      SEGLEN=ABS(ZFF(ISIDE,INODE+1)-ZFF(ISIDE,INODE))
      XN(1)=(ZFF(ISIDE,INODE)-ZFF(ISIDE,INODE+1))/SEGLEN
      XN(2)=0.0
C
      Z1=ZFF(ISIDE,INODE)
      Z2=ZFF(ISIDE,INODE+1)
C
      MARK=0
      IF (ISIDE .EQ. 1)THEN
```

```
      IF (ISRCE .EQ. NTOP(1,LAYER)
     &   .AND. INODE .EQ. JFF(1,LAYER)) MARK = 2
      IF (ISRCE .EQ. NBOTT(1,LAYER)
     &   .AND. INODE .EQ. KFF(1,LAYER)-1) MARK = 1
      END IF
      IF (ISIDE .EQ. 2)THEN
      IF (ISRCE .EQ. NTOP(2,LAYER)
     &   .AND. INODE .EQ. KFF(2,LAYER)-1) MARK = 1
      IF (ISRCE .EQ. NBOTT(2,LAYER)
     &   .AND. INODE .EQ. JFF(2,LAYER)) MARK = 2
      END IF
C
C     MARK = 1, ADJACENT ELEMENT PRIOR TO ISRCE
C     MARK = 2, ADJACENT ELEMENT AFTER ISRCE
C
      IF (MARK .EQ. 1 .OR. MARK .EQ. 2)THEN
      CALL AADJSEG(IAREA,SEGLEN,CPCSR,P1,P2,S1,S2,P1R,P2R,
     & S1R,S2R,F2H1,F3H1,F2H1R,F3H1R)
C
      IF (MARK .EQ. 1)THEN
      R(1)= -XN(2)
      R(2)= +XN(1)
      ELSE
      R(1)= +XN(2)
      R(2)= -XN(1)
      END IF
      DO I =1,2
      DO K = 1,2
      PIK=  P2*R(I)*R(K)
      SIK=  S2*R(I)*R(K)
      PIKR= P2R*R(I)*R(K)
      SIKR= S2R*R(I)*R(K)
C
      IF (I .EQ. K) THEN
      PIK = P1 + PIK
      SIK = S1 + SIK
      PIKR= P1R+ PIKR
      SIKR= S1R + SIKR
      END IF
      GMU(1,I,K)=CALC1*PIK+SIK
      GMU(2,I,K)=CALC1*PIKR+SIKR
C
      T(1,I,K)=XN(I)*R(K)*F2H1 + XN(K)*R(I)*F3H1
      T(2,I,K)=XN(I)*R(K)*F2H1R + XN(K)*R(I)*F3H1R
C
      END DO
      END DO
C
      ELSE
C
      DO 40IGAUSS=1,NGAUS3
      XGAUSS(IGAUSS)=XFF(ISIDE,LAYER)
      ZGAUSS(IGAUSS)=(XX3(IGAUSS)+1.0)*(Z2-Z1)/2.0+Z1
      RADIUS(IGAUSS)=SQRT((XGAUSS(IGAUSS)-XI)**2
     & +(ZGAUSS(IGAUSS)-ZI)**2)
      XKSR=FBAR*RADIUS(IGAUSS)/CSRAT(IAREA)
      XKPR=FBAR*RADIUS(IGAUSS)/(CPCSR*CSRAT(IAREA))
      CALL HANKEL(0,XKPR,H01KPR)
```

```
      CALL HANKEL(1,XKPR,H11KPR)
      CALL HANKEL(0,XKSR,H01KSR)
      CALL HANKEL(1,XKSR,H11KSR)
      H21KPR=2.*H11KPR/XKPR - H01KPR
      H21KSR=2.*H11KSR/XKSR - H01KSR
      P1=(0.0,+0.1250)*(H21KPR+H01KPR)-1./(2*PI*XKPR*XKPR)
      P2=(0.0,-0.250)*H21KPR + 1./(PI*XKPR*XKPR)
      S1=(0.0,+0.1250)*(-H21KSR+H01KSR)+1./(2*PI*XKSR*XKSR)
      S2=(0.0,+0.250)*H21KSR - 1./(PI*XKSR*XKSR)
C
      F1H1=(0.0,1.0)*(CALC1*(2.*H21KPR -0.5*XKPR*H11KPR)
     & -2.0*H21KSR + 0.5*XKSR*H11KSR)/RADIUS(IGAUSS)
      F2H1=(0.0,0.25)*((2.0*CALC1-1.0)*XKPR*H11KPR
     & - 2.0*CALC1*H21KPR + 2.0*H21KSR)/RADIUS(IGAUSS)
      F3H1=(0.0,0.25)*
     & (-2.0*CALC1*H21KPR + 2.0*H21KSR - XKSR*H11KSR)/RADIUS(IGAUSS)
C
C
      R(1)=XGAUSS(IGAUSS)-XI
      R(2)=ZGAUSS(IGAUSS)-ZI
      RJNJ=0.0
      DO 20J=1,2
      RJNJ=R(J)*XN(J)+RJNJ
   20 CONTINUE
C     WRITE(10,*)'ISRCE = ',ISRCE
      DO 30 I=1,2
      DO 30 K=1,2
      PIK= P2*R(I)*R(K)/(RADIUS(IGAUSS)**2)
      SIK= S2*R(I)*R(K)/(RADIUS(IGAUSS)**2)
C
      IF(I .EQ. K)THEN
      PIK=PIK+P1
      SIK=SIK+S1
      END IF
C
      GMU(IGAUSS,I,K)=CALC1*PIK+SIK
C
      T(IGAUSS,I,K)=( XN(I)*R(K)*F2H1
     &               +   XN(K)*R(I)*F3H1
     &               +   RJNJ*R(I)*R(K)*F1H1/(RADIUS(IGAUSS)**2) )
C
        IF(I .EQ. K)THEN
      T(IGAUSS,I,K)=T(IGAUSS,I,K)+
     &               +    RJNJ*F3H1
      END IF
C
      T(IGAUSS,I,K)=T(IGAUSS,I,K)/RADIUS(IGAUSS)
C
   30 CONTINUE
   40 CONTINUE
C
      END IF
C
C     WRITE(80,*)'INODE',INODE
      DO 45 K=1,2
C     CORRECTION TO STRAIN
      IF (MARK .EQ. 0)THEN
      STCORR=
```

```
C
     &  ( (WT3(1)*WTFN3(1)*GMU(1,1,K)+WT3(2)*WTFN3(2)*GMU(2,1,K)
     &  +  WT3(1)*WTFN3(3)*GMU(3,1,K))*FFPX(ISIDE,INODE)
C
     &  + (WT3(1)*WTFN3(1)*GMU(1,2,K)+WT3(2)*WTFN3(2)*GMU(2,2,K)
     &  +  WT3(1)*WTFN3(3)*GMU(3,2,K))*FFPZ(ISIDE,INODE)
C
     &  + (WT3(1)*WTFN3(3)*GMU(1,1,K)+WT3(2)*WTFN3(2)*GMU(2,1,K)
     &  +  WT3(1)*WTFN3(1)*GMU(3,1,K))* FFPX(ISIDE,INODE+1)
C
     &  + (WT3(1)*WTFN3(3)*GMU(1,2,K)+WT3(2)*WTFN3(2)*GMU(2,2,K)
     &  +  WT3(1)*WTFN3(1)*GMU(3,2,K))*FFPZ(ISIDE,INODE+1)    )
     &  * SEGLEN/2.0
C
C     CORRECTION TO DISPLACEMENT
      WCORR=
     &  ( (WT3(1)*WTFN3(1)*T(1,1,K)+WT3(2)*WTFN3(2)*T(2,1,K)
     &  +  WT3(1)*WTFN3(3)*T(3,1,K))* FFU(ISIDE,INODE)
C
     &  + (WT3(1)*WTFN3(1)*T(1,2,K)+WT3(2)*WTFN3(2)*T(2,2,K)
     &  +  WT3(1)*WTFN3(3)*T(3,2,K))* FFW(ISIDE,INODE)
C
     &  + (WT3(1)*WTFN3(3)*T(1,1,K)+WT3(2)*WTFN3(2)*T(2,1,K)
     &  +  WT3(1)*WTFN3(1)*T(3,1,K))* FFU(ISIDE,INODE+1)
C
     &  + (WT3(1)*WTFN3(3)*T(1,2,K)+WT3(2)*WTFN3(2)*T(2,2,K)
     &  +  WT3(1)*WTFN3(1)*T(3,2,K))* FFW(ISIDE,INODE+1)    )
     &  * SEGLEN/2.0
C
      ELSE
C
      IF (MARK .EQ. 1)THEN
C     ELEMENT ADJACENT TO SOURCE AND BEFORE ISRCE
C
      STCORR= (+1.0/SEGLEN * GMU(2,1,K) )*FFPX(ISIDE,INODE)
     &      +   (+1.0/SEGLEN * GMU(2,2,K) )*FFPZ(ISIDE,INODE)
     &   +   (GMU(1,1,K) -1.0/SEGLEN * GMU(2,1,K)) *FFPX(ISIDE,INODE+1)
     &   +   (GMU(1,2,K) -1.0/SEGLEN * GMU(2,2,K)) *FFPZ(ISIDE,INODE+1)
C
      WCORR    =  (+1.0/SEGLEN * T(2,1,K) ) * FFU(ISIDE,INODE)
     &       +           (+1.0/SEGLEN * T(2,2,K) ) * FFW(ISIDE,INODE)
     &   +     (T(1,1,K) -1.0/SEGLEN * T(2,1,K)) * FFU(ISIDE,INODE+1)
     &   +     (T(1,2,K) -1.0/SEGLEN * T(2,2,K)) * FFW(ISIDE,INODE+1)
C
      ELSE
C
C     MARK = 2, ELEMENT ADJACENT TO SOURCE AND AFTER ISRCE
C
      STCORR= (GMU(1,1,K) -1.0/SEGLEN * GMU(2,1,K))*FFPX(ISIDE,INODE)
     &       +  (GMU(1,2,K) -1.0/SEGLEN * GMU(2,2,K))*FFPZ(ISIDE,INODE)
     &       +  (1.0/SEGLEN * GMU(2,1,K))*FFPX(ISIDE,INODE+1)
     &       +  (1.0/SEGLEN * GMU(2,2,K))*FFPZ(ISIDE,INODE+1)
C
      WCORR = (T(1,1,K) -1.0/SEGLEN * T(2,1,K)) *  FFU(ISIDE,INODE)
     &       +  (T(1,2,K) -1.0/SEGLEN * T(2,2,K)) *  FFW(ISIDE,INODE)
     &       +  (1.0/SEGLEN * T(2,1,K)) * FFU(ISIDE,INODE+1)
     &       +  (1.0/SEGLEN * T(2,2,K)) * FFW(ISIDE,INODE+1)
C
```

```
      END IF
      END IF
C
C     WRITE(80,*)'ISRCE,ISIDE,K',ISRCE,ISIDE,K
      TCORR(ISRCE,K)=TCORR(ISRCE,K)-WCORR+STCORR
C     WRITE(80,*)'WCORR,STCORR,TCORR',
C    & WCORR,STCORR,TCORR(ISRCE,K)
   45 CONTINUE
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HSINTGR
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     HALF-SPACE INTEGRALS
      COMMON/RHSINT/ASTART
      COMMON/CHSINT/H0EXP(2,2),H1EXP(2,2),H2EXP(2,2),H2EXPR(2,2),
     & EXPR2(2),EXPR3(2)
      REAL ASTART
      COMPLEX H0EXP,H1EXP,H2EXP,H2EXPR,EXPR2,EXPR3
C
C
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
      COMMON/KHLFSP/XKH1,XKPH1,XKSH1
      REAL XKH1,XKPH1,XKSH1
C
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
```

```
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      DIMENSION SIGN(2)
C
      PI=4.0*ATAN(1.0)
C
      CPCSR= SQRT(2.*(1.-POISS(HALFSP)) / (1-2.*POISS(HALFSP))   )
C
      IF(ATYPE .EQ. 'P')THEN
      XLXHS=COS(ANGLE(IANGLE)/57.29577951)
      XMXHS=XLXHS/CPCSR
      END IF
      IF(ATYPE .EQ. 'SV')THEN
      XMXHS=COS(ANGLE(IANGLE)/57.29577951)
      XLXHS=XMXHS*CPCSR
      END IF
C
      XKSH1=FBAR/CSRAT(HALFSP)
      XKPH1=XKSH1/CPCSR
      XKH1=XKSH1*XMXHS
C
      JNO=JNODE(HALFSP,1)
      KNO=KNODE(HALFSP,NBOUND(HALFSP))
      SEGLN1=ABS(X(KNO)-X(KNO-1))
      SEGLN2=ABS(X(JNO+1)-X(JNO))
      IF(SEGLN1 .GT. SEGLN2) THEN
      ASTART = SEGLN2
      ELSE
      ASTART = SEGLN1
      END IF
      SIGN(1)=+1.0
      SIGN(2)=-1.0
C
C
      ZETA=XKH1
      DO 50 ITYPE=1,2
C     ITYPE=1 (P WAVE), ITYPE=2 (SV WAVE)
C     ISIDE=1 (-X), ISIDE=2 (+X)
C
C
C     DETERMINE ALPHA AND ZETA
      IF(ITYPE .EQ. 1)THEN
C     ALPHA= KpH1 , ZETA = KH1
      ALPHA= XKPH1
      END IF
      IF(ITYPE .EQ. 2)THEN
C     ALPHA= KsH1 , ZETA = KH1
      ALPHA=XKSH1
      END IF
C
      CALL ATRIG(PI,0,ALPHA,ZETA)
      CALL INTGJSINE(PI,0,ALPHA,ZETA,AJSINE)
      CALL INTGYSINE(PI,0,ALPHA,ZETA,AYSINE)
      CALL INTGJCOS(PI,0,ALPHA,ZETA,AJCOS)
      CALL INTGYCOS(PI,0,ALPHA,ZETA,AYCOS)
      DO 10ISIDE=1,2
```

```
      H0EXP(ITYPE,ISIDE)=(AJCOS+SIGN(ISIDE)*AYSINE)+
     &              (0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE)
   10 CONTINUE
C
C
      CALL ATRIG(PI,1,ALPHA,ZETA)
C
      CALL INTGJSINE(PI,1,ALPHA,ZETA,AJSINE)
      CALL INTGYSINE(PI,1,ALPHA,ZETA,AYSINE)
      CALL INTGJCOS(PI,1,ALPHA,ZETA,AJCOS)
      CALL INTGYCOS(PI,1,ALPHA,ZETA,AYCOS)
      DO 20ISIDE=1,2
      H1EXP(ITYPE,ISIDE)=(AJCOS+SIGN(ISIDE)*AYSINE)+
     &              (0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE)
   20 CONTINUE
C
      CALL ATRIG(PI,2,ALPHA,ZETA)
      CALL INTGJSINE(PI,2,ALPHA,ZETA,AJSINE)
      CALL INTGYSINE(PI,2,ALPHA,ZETA,AYSINE)
      CALL INTGJCOS(PI,2,ALPHA,ZETA,AJCOS)
      CALL INTGYCOS(PI,2,ALPHA,ZETA,AYCOS)
C
C
      CALL INTGJSINER(PI,2,ALPHA,ZETA,AJSINR)
      CALL INTGYSINER(PI,2,ALPHA,ZETA,AYSINR)
      CALL INTGJCOSR(PI,2,ALPHA,ZETA,AJCOSR)
      CALL INTGYCOSR(PI,2,ALPHA,ZETA,AYCOSR)
      DO 30ISIDE=1,2
      H2EXP(ITYPE,ISIDE)=(AJCOS+SIGN(ISIDE)*AYSINE)+
     &              (0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE)
      H2EXPR(ITYPE,ISIDE)=(AJCOSR+SIGN(ISIDE)*AYSINR)+
     &              (0.0,1.0)*(AYCOSR-SIGN(ISIDE)*AJSINR)
   30 CONTINUE
   50 CONTINUE
C     AREA FOR EXP(IKR)/(R*R)
      CALL INTGSINR2(PI,0,ALPHA,ZETA,ASINR2)
      CALL INTGCOSR2(PI,0,ALPHA,ZETA,ACOSR2)
C     CALL INTGSINR3(PI,0,ALPHA,ZETA,ASINR3)
C     CALL INTGCOSR3(PI,0,ALPHA,ZETA,ACOSR3)
      DO 60 ISIDE=1,2
      EXPR2(ISIDE)=ACOSR2-(SIGN(ISIDE)*(0.0,1.0)*ASINR2)
C     EXPR3(ISIDE)=ACOSR3-(SIGN(ISIDE)*(0.0,1.0)*ASINR3)
   60 CONTINUE
      RETURN
      END


      SUBROUTINE ATRIG(PI,NU,ALPHA,ZETA)
C
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C     DATA REQUIRED FOR GAUSSIAN QUADRATURE
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
```

```
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C     CALCULATE TRIG AREAS FROM 20 TO INFINITY
C
      DIFF = ABS(ALPHA - ZETA)
C
      IF (ZETA .GT. ALPHA)THEN
      SIGN=+1.0
      ELSE
      SIGN = -1.0
      END IF
C
C     FUNCTION = SINE(alpha*r)*SINE(beta*r)/SQRT(R)
      IFUNC=9
C
      IF(DIFF .LT. .001)THEN
      A=20.0
      B=1000000.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASSR12=AREA
      ELSE
      ASSR12=0.6266*(1./SQRT(ABS(ZETA-ALPHA)) -1./SQRT(ALPHA+ZETA) )
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASSR12=ASSR12-AREA
      END IF
C
C     FUNCTION = COS(alpha*r)*SINE(beta*r)/SQRT(R)
      IFUNC=10
C
      IF(DIFF .LT. .001)THEN
      A=20.0
      B=1000000.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACSR12=AREA
      ELSE
      ACSR12=0.6266*(1./SQRT(ALPHA+ZETA)
     & +1./SQRT(ABS(ALPHA-ZETA))*SIGN)
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACSR12=ACSR12-AREA
      END IF
C
C     FUNCTION = COS(alpha*r)*COS(beta*r)/SQRT(R)
      IFUNC=11
C
      IF(DIFF .LT. .001)THEN
      A=20.0
      B=1000000.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACCR12=AREA
      ELSE
      ACCR12=0.6266*(1./SQRT(ALPHA+ZETA) +1./SQRT(ABS(ALPHA-ZETA)))
      A=0.0001
```

```
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACCR12=ACCR12-AREA
      END IF
C
C
C     FUNCTION = SINE(alpha*r)*COS(beta*r)/SQRT(R)
      IFUNC=12
C
      IF(DIFF .LT. .001)THEN
      A=20.0
      B=1000000.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASCR12=AREA
      ELSE
      ASCR12=0.6266*(1./SQRT(ALPHA+ZETA)
     & -1./SQRT(ABS(ALPHA-ZETA))*SIGN)
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASCR12=ASCR12-AREA
      END IF
C
C     FUNCTION = SINE(alpha*r)*SINE(beta*r)/(R*SQRT(R))
      IFUNC=13
C
      IF(DIFF .LT. .001)THEN
      A=20.0
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASSR32=AREA
      ELSE
      ASSR32= -1.25331*(SQRT(ABS(ZETA-ALPHA))  - SQRT(ALPHA+ZETA) )
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASSR32=ASSR32-AREA
      END IF
C
C     FUNCTION = COS(alpha*r)*SINE(beta*r)/(R*SQRT(R))
      IFUNC=14
C
      ACSR32= +1.25331*(SQRT(ALPHA+ZETA) + SQRT(ABS(ALPHA-ZETA))*SIGN)
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACSR32=ACSR32-AREA
C
C     FUNCTION = COS(alpha*r)*COS(beta*r)/(R*SQRT(R))
      IFUNC=15
      A=20.0
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ACCR32=AREA
C
C
C     FUNCTION = SINE(alpha*r)*COS(beta*r)/(R*SQRT(R))
      IFUNC=16
```

```
      ASCR32= +1.25331*(SQRT(ALPHA+ZETA) - SQRT(ABS(ALPHA-ZETA))*SIGN)
      A=0.0001
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      ASCR32=ASCR32-AREA
      RETURN
      END




      SUBROUTINE INTGJSINE(PI,NU,ALPHA,ZETA,AJSINE)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= Jn(alpha*r)*SINE(beta*r)'
      IFUNC=1
C
            DIFF=ABS(ALPHA-ZETA)
            IF(DIFF .GT. 0.001)THEN
C
      IF(ALPHA .GT. ZETA)THEN
      AJSINE = SIN(NU*ASIN(ZETA/ALPHA))/SQRT(ALPHA**2-ZETA**2)
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AJSINE=(ALPHA**NU)*COS(NU*PI/2)
     & /(SQRT(ZETA**2-ALPHA**2)*((ZETA+SQRT(ZETA**2-ALPHA**2))**NU))
      END IF
C
      A=0.0
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJSINE=AJSINE-AREA
C
            ELSE
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJSINE=AREA
      IF (NU .EQ. 0 .OR. NU .EQ. 1)SIGN1=+1.0
      IF (NU .EQ. 2)SIGN1= -1.0
      IF (NU .EQ. 0 .OR. NU .EQ. 2)SIGN2=+1.0
      IF (NU .EQ. 1)SIGN2= -1.0
      AJSINE=AJSINE+SIGN1/(SQRT(PI*ALPHA))*(SIGN2*ACSR12+ASSR12)
C
            END IF
C
```

```
      RETURN
      END


      SUBROUTINE INTGJCOS(PI,NU,ALPHA,ZETA,AJCOS)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= Jn(alpha*r)*COS(beta*r)'
      IFUNC=2
C
            DIFF=ABS(ALPHA-ZETA)
            IF(DIFF .GT. 0.001)THEN
C
      IF(ALPHA .GT. ZETA)THEN
      AJCOS = COS(NU*ASIN(ZETA/ALPHA))/SQRT(ALPHA**2-ZETA**2)
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AJCOS= -1.0*(ALPHA**NU)*SIN(NU*PI/2)
     & /(SQRT(ZETA**2-ALPHA**2)*((ZETA+SQRT(ZETA**2-ALPHA**2))**NU))
      END IF
C
      A=0.0
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJCOS=AJCOS-AREA
C
            ELSE
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJCOS=AREA
      IF (NU .EQ. 0 .OR. NU .EQ. 1)SIGN1=+1.0
      IF (NU .EQ. 2)SIGN1= -1.0
      IF (NU .EQ. 0 .OR. NU .EQ. 2)SIGN2=+1.0
      IF (NU .EQ. 1)SIGN2= -1.0
      AJCOS=AJCOS+SIGN1/(SQRT(PI*ALPHA))*(ASCR12+SIGN2*ACCR12)
C
            END IF
C
      RETURN
      END


      SUBROUTINE INTGYSINE(PI,NU,ALPHA,ZETA,AYSINE)
      COMMON/RHSINT/ASTART
      REAL ASTART
```

```
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= Yn(alpha*r)*SINE(beta*r)'
      IFUNC=3
C
      DIFF=ABS(ALPHA-ZETA)
C
            IF(NU .EQ. 0 .AND. DIFF .GT. 0.001)THEN
C
C
      IF(ALPHA .GT. ZETA)THEN
      AYSINE = 2.0*ASIN(ZETA/ALPHA)/(PI*SQRT(ALPHA**2-ZETA**2))
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AYSINE = 2.0/(PI*SQRT(ZETA**2-ALPHA**2))
     & * LOG(ZETA/ALPHA - SQRT((ZETA/ALPHA)**2 -1.0))
      END IF
      A=0.0
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYSINE=AYSINE-AREA

C
            ELSE
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYSINE=AREA
      IF (NU .EQ. 0 )SIGN1=+1.0
      IF (NU .EQ. 1 .OR. NU .EQ. 2)SIGN1= -1.0
      IF (NU .EQ. 1)SIGN2= +1.0
      IF (NU .EQ. 0 .OR. NU .EQ. 2)SIGN2=-1.0
      AYSINE=AYSINE+SIGN1/(SQRT(PI*ALPHA)) * (ASSR12+SIGN2*ACSR12)
C
            END IF

      RETURN
      END

      SUBROUTINE INTGYCOS(PI,NU,ALPHA,ZETA,AYCOS)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
```

```
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= Yn(alpha*r)*COS(beta*r)'
      IFUNC=4
C
      DIFF=ABS(ALPHA-ZETA)
C
            IF(NU .EQ. 0 .AND. DIFF .GT. 0.001)THEN
C
C
      IF(ALPHA .GT. ZETA)THEN
      AYCOS = 0.0
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AYCOS = -1.0/ SQRT(ZETA**2-ALPHA**2)
      END IF
C
      A=0.0
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYCOS=AYCOS-AREA

            ELSE
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYCOS=AREA
      IF (NU .EQ. 0 )SIGN1=+1.0
      IF (NU .EQ. 1 .OR. NU .EQ. 2)SIGN1= -1.0
      IF (NU .EQ. 1)SIGN2= +1.0
      IF (NU .EQ. 0 .OR. NU .EQ. 2)SIGN2=-1.0
      AYCOS=AYCOS+SIGN1/(SQRT(PI*ALPHA)) * (ASCR12+SIGN2*ACCR12)
C
            END IF

      RETURN
      END



      SUBROUTINE INTGJSINER(PI,NU,ALPHA,ZETA,AJSINR)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C     FUNCTION= Jn(alpha*r)*SINE(beta*r)/r
      IFUNC=5
C
```

```
C
      IF(ALPHA .GE. ZETA)THEN
      AJSINR = SIN(NU*ASIN(ZETA/ALPHA))/NU
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AJSINR=(ALPHA**NU)*SIN(NU*PI/2.)
     & /(NU*((ZETA+SQRT(ZETA**2-ALPHA**2))**NU))
      END IF
C
      A=0.0001
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJSINR=AJSINR-AREA
C
C
      RETURN
      END



      SUBROUTINE INTGJCOSR(PI,NU,ALPHA,ZETA,AJCOSR)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C     FUNCTION= Jn(alpha*r)*COS(beta*r)/r
C     NU = 2
C
      IFUNC=6
C
      IF(ALPHA .GE. ZETA)THEN
      AJCOSR = COS(NU*ASIN(ZETA/ALPHA))/NU
      END IF
      IF(ALPHA .LT. ZETA)THEN
      AJCOSR= (ALPHA**NU)*COS(NU*PI/2.)
     & /(NU*((ZETA+SQRT(ZETA**2-ALPHA**2))**NU))
      END IF
C
      A=0.0001
      B=ASTART
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AJCOSR=AJCOSR-AREA
C
C
      RETURN
      END


      SUBROUTINE INTGYSINER(PI,NU,ALPHA,ZETA,AYSINR)
      COMMON/RHSINT/ASTART
```

```
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C     FUNCTION= Yn(alpha*r)*SINE(beta*r)/r
C     NU =2
      IFUNC=7
      SIGN=-1.0
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYSINR=AREA
      AYSINR=AYSINR+SIGN/(SQRT(PI*ALPHA)) * (ASSR32-ACSR32)
C
      RETURN
      END

      SUBROUTINE INTGYCOSR(PI,NU,ALPHA,ZETA,AYCOSR)
      COMMON/RHSINT/ASTART
      REAL ASTART
      COMMON /AREA/ASSR12,ACSR12,ASCR12,ACCR12,
     & ASSR32,ACSR32,ASCR32,ACCR32
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
C     FUNCTION= Yn(alpha*r)*COS(beta*r)/r
C     NU =2
      IFUNC=8
      SIGN=-1.0
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=20.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      AYCOSR=AREA
      AYCOSR=AYCOSR+SIGN/(SQRT(PI*ALPHA)) * (ASCR32-ACCR32)
C
      RETURN
      END

      SUBROUTINE INTGSINR2(PI,NU,ALPHA,ZETA,ASINR2)
C
      COMMON/RHSINT/ASTART
      REAL ASTART
```

```
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= 1./R2 * SINE(BETA*R)
      IFUNC=17
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,ASINR2)
C
      RETURN
      END


      SUBROUTINE INTGSINR3(PI,NU,ALPHA,ZETA,ASINR3)
C
      COMMON/RHSINT/ASTART
      REAL ASTART
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= 1./R3 * SINE(BETA*R)
      IFUNC=18
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,ASINR3)
C
      RETURN
      END

      SUBROUTINE INTGCOSR2(PI,NU,ALPHA,ZETA,ACOSR2)
C
      COMMON/RHSINT/ASTART
      REAL ASTART
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= 1./R2 * COS(BETA*R)
      IFUNC=19
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,ACOSR2)
C
```

```
      RETURN
      END


      SUBROUTINE INTGCOSR3(PI,NU,ALPHA,ZETA,ACOSR3)
C
      COMMON/RHSINT/ASTART
      REAL ASTART
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C     FUNCTION= 1./R3 * COS(BETA*R)
      IFUNC=20
C
C     SOLVE NUMERICALLY
      A=ASTART
      B=100.0
      CALL NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,ACOSR3)
C
      RETURN
      END




      SUBROUTINE NUMINTGR(IFUNC,NU,ALPHA,ZETA,A,B,AREA)
      DIMENSION R(5),Y(5)
      DOUBLE PRECISION AOLD,ATOT,ADELTA,DIFF
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
C
      HDELTA=B-A
C
      AREA=0.0
      IF(ABS(HDELTA) .LT. 0.00001)THEN
      RETURN
      END IF
C
      IF(A .LT. 1.0 .AND. B .GT. 2.0)THEN
      AA=A
      BB=2.0
      NCYCLE =2
      ELSE
      AA=A
      BB=B
      NCYCLE=1
      END IF
      DO 50ICYCLE =1,NCYCLE
      HDELTA =BB-AA
      AOLD=9.99999D20
      TOLER=0.005
```

```
   10 CONTINUE
      ATOT=0.0D0
      N=(BB-AA)/HDELTA
C     WRITE(*,*)'N = ',N
      DO 40I=1,N
      AAA=(I-1)*HDELTA+AA
      DO 20J=1,NGAUSS
      R(J)=  AAA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALL YVALUE(IFUNC,NU,ALPHA,ZETA,R(J),Y(J))
   20 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   40 CONTINUE
      DIFF=DABS(ATOT-AOLD)
C     WRITE(*,*)'ATOT = ',ATOT
      IF (DIFF .GT. TOLER ) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 10
      END IF
      AREA=ATOT+AREA
      AA=BB
      BB=B
   50 CONTINUE
C     WRITE(*,*)'AREA = ',AREA
      RETURN
      END

      SUBROUTINE YVALUE(IFUNC,IORDER,ALPHA,ZETA,R,Y)
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION JN,YN
C
      GO TO (10,20,30,40,50,60,70,80,90,100,
     &  110,120,130,140,150,160,170,180,190,200), IFUNC
   10 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*SIN(ZETA*R)
      GO TO 9999
   20 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*COS(ZETA*R)
      GO TO 9999
   30 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
      Y=YN*SIN(ZETA*R)
      GO TO 9999
   40 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
```

```
      Y=YN*COS(ZETA*R)
      GO TO 9999
   50 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*SIN(ZETA*R)/R
      GO TO 9999
   60 CONTINUE
      X=ALPHA*R
      CALL JNU(IORDER,X,JN)
      Y=JN*COS(ZETA*R)/R
      GO TO 9999
   70 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
      Y=YN*SIN(ZETA*R)/R
      GO TO 9999
   80 CONTINUE
      X=ALPHA*R
      CALL YNU(IORDER,X,YN)
      Y=YN*COS(ZETA*R)/R
      GO TO 9999
   90 CONTINUE
      Y=SIN(ALPHA*R)*SIN(ZETA*R)/SQRT(R)
      GO TO 9999
  100 CONTINUE
      Y=COS(ALPHA*R)*SIN(ZETA*R)/SQRT(R)
      GO TO 9999
  110 CONTINUE
      Y=COS(ALPHA*R)*COS(ZETA*R)/SQRT(R)
      GO TO 9999
  120 CONTINUE
      Y=SIN(ALPHA*R)*COS(ZETA*R)/SQRT(R)
      GO TO 9999
  130 CONTINUE
      Y=SIN(ALPHA*R)*SIN(ZETA*R)/(R**1.5)
      GO TO 9999
  140 CONTINUE
      Y=COS(ALPHA*R)*SIN(ZETA*R)/(R**1.5)
      GO TO 9999
  150 CONTINUE
      Y=COS(ALPHA*R)*COS(ZETA*R)/(R**1.5)
      GO TO 9999
  160 CONTINUE
      Y=SIN(ALPHA*R)*COS(ZETA*R)/(R**1.5)
      GO TO 9999
  170 CONTINUE
      Y= 1./(R**2)  * SIN(ZETA*R)
      GO TO 9999
  180 CONTINUE
      Y= 1./(R**3)  * SIN(ZETA*R)
      GO TO 9999
  190 CONTINUE
      Y= 1./(R**2)  * COS(ZETA*R)
      GO TO 9999
  200 CONTINUE
      Y= 1./(R**3)  * COS(ZETA*R)
C
```

```
 9999 CONTINUE
      RETURN
      END



      SUBROUTINE JNU(IORDER,XREAL,JN)
C     ****************************************************
C     BESSEL FUNCTION (Jn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     ****************************************************
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,AA,A,B
      DOUBLE PRECISION JN
C
      Z=XREAL
      PI=3.1415926535D0
      TOLER=.00001D0
C     CALCULATE BESSEL FUNCTION OF 1ST KIND (JN)
      A=0.0D0
      B=PI
      HDELTA=B-A
      AOLD=9.99999D20
   10 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 20I=1,N
      AA= (I-1)*HDELTA+A
      DO 15J=1,NGAUSS
      X(J)=   AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))
   15 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   20 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 10
      END IF
   25 CONTINUE
      JN= ATOT/PI
      RETURN
      END

      SUBROUTINE YNU(IORDER,XREAL,YN)
C     ****************************************************
```

```
C     BESSEL FUNCTION (Yn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     *****************************************
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,CALC,AA,A,B
      DOUBLE PRECISION YN
C
C     CALCULATE BESSEL FUNCTION OF 2ND KIND, ORDER N (YN)
      PI=3.1415926535D0
      TOLER=0.00001
      A=0.
      B=PI
      HDELTA=B-A
      AOLD=9.99999D20
   30 CONTINUE
      N=(B-A)/HDELTA
      ATOT=0.0
      DO 50I=1,N
      AA= (I-1)*HDELTA+A
      DO 40J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DSIN(XREAL* DSIN(X(J))-IORDER*X(J))
   40 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   50 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 30
      END IF
   55 CONTINUE
      YN=ATOT
      A=0.
      B=10.
      HDELTA=1.0
C     HDELTA=B-A
      AOLD=9.99999D20
   60 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 80I=1,N
      AA= (I-1)*HDELTA+A
      DO 70J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALC=XREAL*DSINH(X(J))
      IF(CALC .GT. 40.)CALC=40.
      Y(J)=
     &(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
```

```
     & DEXP(-CALC)
  70 CONTINUE
     ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
     ATOT=ATOT+ADELTA
  80 CONTINUE
     DIFF=ABS(ATOT-AOLD)
     IF (DIFF .GT. TOLER) THEN
     HDELTA=HDELTA/2.0
     AOLD=ATOT
     GO TO 60
     END IF
  90 CONTINUE
     YN=(YN-ATOT)/PI
     RETURN
     END
```

```
      SUBROUTINE HSCORR(ISRCE,XI,ZI,IAREA)
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
```

```
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C2345678911234567892123456789312345678941234567895123456789612345678971 2
C
C     HALF-SPACE INTEGRALS
      COMMON/RHSINT/ASTART
      COMMON/CHSINT/H0EXP(2,2),H1EXP(2,2),H2EXP(2,2),H2EXPR(2,2),
     & EXPR2(2),EXPR3(2)
      REAL ASTART
      COMPLEX H0EXP,H1EXP,H2EXP,H2EXPR,EXPR2,EXPR3
C
      COMMON/KHLFSP/XKH1,XKPH1,XKSH1
      REAL XKH1,XKPH1,XKSH1
C
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      COMPLEX WCORR,STCORR
      COMPLEX CH0EXP(2),CH1EXP(2),CH2EXP(2),CH2EPR(2),CEXPR2,
     & F2H1,F3H1,HSDISPL(2),HSTRAC(2),T(2,2),GMU(2,2),
     & P1,P2,S1,S2,PIK,SIK
C     COMPLEX CEXPR3
C
      DIMENSION SIGN(2),R(2),XN(2)
      SIGN(1)=1.0
      SIGN(2)=-1.0
      PI=4.*ATAN(1.0)
      TCORR(ISRCE,1)=(0.0,0.0)
      TCORR(ISRCE,2)=(0.0,0.0)
      CALC1=(1.-2.*POISS(IAREA))/(2.*(1.-POISS(IAREA)))
      CALC2=POISS(IAREA)/(1.-POISS(IAREA))
      CPCSR= 1.0/SQRT(CALC1)
C     WRITE(80,*)'ISRCE',ISRCE
C     AREA = HALF-SPACE
C
C
      A=ASTART
C
      ZETA=XKH1
      STCORR=(0.0,0.0)
      WCORR=(0.0,0.0)
      R(2)=0.0
      XN(1)=0.0
      XN(2)= -1.0
      DO 30ISIDE=1,2
      IF(ISIDE .EQ. 1)THEN
      B=XI-X(KNODE(HALFSP,NBOUND(HALFSP)))
      R(1)= -1.0
      END IF
      IF(ISIDE .EQ. 2)THEN
      B=X(JNODE(HALFSP,1))-XI
```

```
      R(1)= +1.0
      END IF
      DO 10 ITYPE=1,2
C     ITYPE=1 (P WAVE), ITYPE=2 (SV WAVE)
C     ISIDE=1 (-X), ISIDE=2 (+X)
C
C
C     DETERMINE ALPHA AND ZETA
      IF(ITYPE .EQ. 1)THEN
C     ALPHA= Kp , ZETA = K
      ALPHA= XKPH1
      END IF
      IF(ITYPE .EQ. 2)THEN
C     ALPHA= Ks , ZETA = K
      ALPHA=XKSH1
      END IF
C
      CALL NUMINTGR(1,0,ALPHA,ZETA,A,B,AJSINE)
      CALL NUMINTGR(3,0,ALPHA,ZETA,A,B,AYSINE)
      CALL NUMINTGR(2,0,ALPHA,ZETA,A,B,AJCOS)
      CALL NUMINTGR(4,0,ALPHA,ZETA,A,B,AYCOS)
      CH0EXP(ITYPE)=H0EXP(ITYPE,ISIDE)-
     & ((AJCOS+SIGN(ISIDE)*AYSINE)+(0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE))
C
      CALL NUMINTGR(1,1,ALPHA,ZETA,A,B,AJSINE)
      CALL NUMINTGR(3,1,ALPHA,ZETA,A,B,AYSINE)
      CALL NUMINTGR(2,1,ALPHA,ZETA,A,B,AJCOS)
      CALL NUMINTGR(4,1,ALPHA,ZETA,A,B,AYCOS)
      CH1EXP(ITYPE)=H1EXP(ITYPE,ISIDE)-
     & ((AJCOS+SIGN(ISIDE)*AYSINE)+(0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE))
C
      CALL NUMINTGR(1,2,ALPHA,ZETA,A,B,AJSINE)
      CALL NUMINTGR(3,2,ALPHA,ZETA,A,B,AYSINE)
      CALL NUMINTGR(2,2,ALPHA,ZETA,A,B,AJCOS)
      CALL NUMINTGR(4,2,ALPHA,ZETA,A,B,AYCOS)
C
C
      CALL NUMINTGR(5,2,ALPHA,ZETA,A,B,AJSINR)
      CALL NUMINTGR(7,2,ALPHA,ZETA,A,B,AYSINR)
      CALL NUMINTGR(6,2,ALPHA,ZETA,A,B,AJCOSR)
      CALL NUMINTGR(8,2,ALPHA,ZETA,A,B,AYCOSR)
      CH2EXP(ITYPE)=H2EXP(ITYPE,ISIDE)-
     & ((AJCOS+SIGN(ISIDE)*AYSINE)+(0.0,1.0)*(AYCOS-SIGN(ISIDE)*AJSINE))
      CH2EPR(ITYPE)=H2EXPR(ITYPE,ISIDE)-
     &((AJCOSR+SIGN(ISIDE)*AYSINR)+
     & (0.0,1.0)*(AYCOSR-SIGN(ISIDE)*AJSINR))
C
   10 CONTINUE
C     AREA FOR EXP(IKR)/(R*R)
      CALL NUMINTGR(17,2,ALPHA,ZETA,A,B,ASINR2)
      CALL NUMINTGR(19,2,ALPHA,ZETA,A,B,ACOSR2)
C     CALL NUMINTGR(18,2,ALPHA,ZETA,A,B,ASINR3)
C     CALL NUMINTGR(20,2,ALPHA,ZETA,A,B,ACOSR3)
      CEXPR2= EXPR2(ISIDE)-(ACOSR2-(SIGN(ISIDE)*(0.0,1.0)*ASINR2))
C     CEXPR3= EXPR3(ISIDE)-(ACOSR3-(SIGN(ISIDE)*(0.0,1.0)*ASINR3))
C
C
      HSDISPL(1)=HSU(ISIDE)*CEXP((0.0,1.0)*XKH1*XI)
```

```
      HSDISPL(2)=HSW(ISIDE)*CEXP((0.0,1.0)*XKH1*XI)
      HSTRAC(1)=HSPX(ISIDE)*CEXP((0.0,1.0)*XKH1*XI)
      HSTRAC(2)=HSPZ(ISIDE)*CEXP((0.0,1.0)*XKH1*XI)
C

C
      P1=(0.0,+0.1250)*(CH2EXP(1)+CH0EXP(1))
     & -1./(2*PI*XKPH1*XKPH1)*CEXPR2
      P2=(0.0,-0.250)*CH2EXP(1) + 1./(PI*XKPH1*XKPH1)*CEXPR2
      S1=(0.0,+0.1250)*(-CH2EXP(2)+CH0EXP(2))+
     & 1./(2*PI*XKSH1*XKSH1) * CEXPR2
      S2=(0.0,+0.250)*CH2EXP(2) - 1./(PI*XKSH1*XKSH1)*CEXPR2
C
      F2H1=(0.0,0.25)*((2.0*CALC1-1.0)*XKPH1*CH1EXP(1)
     & - 2.0*CALC1*CH2EPR(1) + 2.0*CH2EPR(2))
      F3H1=(0.0,0.25)*
     & (-2.0*CALC1*CH2EPR(1) + 2.0*CH2EPR(2)- XKSH1*CH1EXP(2))


      DO I=1,2
            DO K=1,2
C
      PIK=P2*R(I)*R(K)
      SIK=S2*R(I)*R(K)
C
      IF(I .EQ. K)THEN
      PIK=PIK+P1
      SIK=SIK+S1
      END IF
C
      GMU(I,K)=CALC1*PIK+SIK
C
      T(I,K)=XN(I)*R(K)*F2H1+XN(K)*R(I)*F3H1
C
C
      STCORR=HSTRAC(I)*GMU(I,K)
      WCORR= HSDISPL(I)*T(I,K)
      TCORR(ISRCE,K)=TCORR(ISRCE,K)-WCORR+STCORR
C
            END DO
C
      END DO
   30 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FIXMAT
C234567891123456789212345678931234567894123456789512345678961234567897l2
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
      INTEGER ISKIP(6),JAREA(3)
      REAL XN1(6),XN2(6)
      COMPLEX GRATIO
C
C     7891123456789212345678931234567894123456789512345678961234567897l2
C
      KNO=KNODE(NAREA,NBOUND(NAREA))
C     OPEN(UNIT=70,FILE='GH.CHECK',STATUS='UNKNOWN')
C
            IF(NAREA .EQ. 1)THEN
            GO TO 205
            END IF
C
C     NAREA > 1
C
      JMARK= -1
      DO 50IAREA =1,NAREA-1
      DO 50INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
C
```

```
            IF (IEQN(INODE) .GT. 0)THEN
      JMARK=JMARK+2
      JJCOL1=2*INODE-1
      DO 20 IROW=1,2*KNO
      HMAT(IROW,JMARK)=HMAT(IROW,JJCOL1)
      HMAT(IROW,JMARK+1)=HMAT(IROW,JJCOL1+1)
   20 CONTINUE
      IF(IEQN(INODE) .GE. 2)THEN
      DO 40 IIAREA=IAREA+1,NAREA
      DO 40 IINODE=JNODE(IIAREA,1),KNODE(IIAREA,NBOUND(IIAREA))
      IF (NCONN(IINODE) .EQ. INODE) THEN
      JJCOL2=IINODE*2 -1
      DO 30 IROW=1,2*KNO
      HMAT(IROW,JMARK)=HMAT(IROW,JMARK)+HMAT(IROW,JJCOL2)
      HMAT(IROW,JMARK+1)=HMAT(IROW,JMARK+1)+HMAT(IROW,JJCOL2+1)
   30 CONTINUE
      END IF
   40 CONTINUE
      END IF
            END IF
C
   50 CONTINUE
      HDIM=JMARK+1
C
      IN3=0
      JMARK= -1
      DO 200IAREA =1,NAREA-1
      DO 190INODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
        IF (IEQN(INODE) .EQ. 0)GO TO 190
C
            IF(IEQN(INODE) .EQ. 1 .OR. IEQN(INODE) .EQ. 2)THEN
      JMARK=JMARK+2
      DO 60 IROW=1,2*KNO
      GMAT(IROW,JMARK)=GMAT(IROW,JCOL1(INODE))
      GMAT(IROW,JMARK+1)=GMAT(IROW,JCOL1(INODE)+1)
   60 CONTINUE
C
            IF(IEQN(INODE) .EQ. 2) THEN
        DO 80 IIAREA=IAREA+1,NAREA
        DO 80 IINODE=JNODE(IIAREA,1),KNODE(IIAREA,NBOUND(IIAREA))
          IF (NCONN(IINODE) .EQ. INODE) THEN
          JJCOL1=JCOL1(INODE)
          JJCOL2=JCOL1(IINODE)
          GRATIO=((CSRAT(IAREA)/CSRAT(IIAREA))**2)*
     &     UWTRAT(IAREA)/UWTRAT(IIAREA)
          DO 70 IROW=1,2*KNO
          GMAT(IROW,JMARK)=GMAT(IROW,JMARK)-GMAT(IROW,JJCOL2)*GRATIO
          GMAT(IROW,JMARK+1)=GMAT(IROW,JMARK+1)-GMAT(IROW,JJCOL2+1)
     &    *GRATIO
   70 CONTINUE
      END IF
   80 CONTINUE
            END IF
            END IF
C
C
        IF (IEQN(INODE) .EQ. 3)THEN
        IN3=IN3+1
```

```
      DO 115 I=1,3
      IINODE=N3(IN3,I)
      IELEM=2*I-1
      SEGLEN=SQRT((X(IINODE)-X(IINODE-1))**2+
     & (Z(IINODE)-Z(IINODE-1))**2)
      XN1(IELEM)=(Z(IINODE-1)-Z(IINODE))/SEGLEN
      XN2(IELEM)=(X(IINODE)-X(IINODE-1))/SEGLEN
      SEGLEN=SQRT((X(IINODE+1)-X(IINODE))**2+
     & (Z(IINODE+1)-Z(IINODE))**2)
      XN1(IELEM+1)=(Z(IINODE)-Z(IINODE+1))/SEGLEN
      XN2(IELEM+1)=(X(IINODE+1)-X(IINODE))/SEGLEN
 115  CONTINUE
      JAREA(1)=IAREA
      DO 130I=2,3
      DO 120 IIAREA=1,NAREA
      DO 120 IINODE=1,KNODE(IIAREA,NBOUND(IIAREA))
      IF(IINODE .EQ. N3(IN3,I))THEN
      JAREA(I)=IIAREA
      GO TO 130
      END IF
 120  CONTINUE
 130  CONTINUE
      DO 140 IELEM=1,6
      ISKIP(IELEM)=0
 140  CONTINUE
      DO180 IELEM=1,5
      IF(ISKIP(IELEM) .EQ. 1)GO TO 180
C
      IF(IELEM .EQ. 1 .OR. IELEM .EQ. 2) THEN
      NODE1=INODE
      IAREA1=JAREA(1)
      END IF
      IF(IELEM .EQ. 3 .OR. IELEM .EQ. 4)THEN
      NODE1=N3(IN3,2)
      IAREA1=JAREA(2)
      END IF
      IF(IELEM .EQ. 5 .OR. IELEM .EQ. 6) THEN
      NODE1=N3(IN3,3)
      IAREA1=JAREA(3)
      END IF
C
C     DETERMINE IF ELEMENT IELEM IS EVEN (MARK1 = 0) OR ODD(MARK1=2)
C    IF IELEM IS EVEN, JJCOL=JCOL1(IINODE)
C    IF IELEM IS ODD, JJCOL=JCOL1(IINODE)-2
C
      TEMP=IELEM
      XREAL =TEMP/2.
      I=IELEM/2
      DIFF=ABS(XREAL-I)
      MARK1=2
      IF(DIFF .LT. 0.01)MARK1=0
C
     JMARK=JMARK+2
     DO 150 IROW=1,2*KNO
     JJCOL=JCOL1(NODE1)-MARK1
     GMAT(IROW,JMARK)=GMAT(IROW,JJCOL)
     GMAT(IROW,JMARK+1)=GMAT(IROW,JJCOL+1)
 150 CONTINUE
```

```
C
         DO 170 IIELEM=IELEM+1,6
         DIFF=ABS(ABS(XN1(IELEM))-ABS(XN1(IIELEM)))
           IF (DIFF .LT. 0.01)THEN
C      ELEMENTS ARE PARALLEL
           DIFF1=ABS(XN1(IELEM)-XN1(IIELEM))
           DIFF2=ABS(XN2(IELEM)-XN2(IIELEM))
           IF(DIFF1 .LT. 0.01 .AND. DIFF2 .LT. 0.01) THEN
C      ELEMENTS ARE PARALLEL WITH SAME NORMALS
           SIGN =+1.0
           ELSE
C      ELEMENTS ARE PARALLEL WITH OPPOSITE NORMALS
           SIGN = -1.0
           END IF
           ISKIP(IIELEM)=1
C          DETERMINE IF ELEMENT IIELEM IS EVEN (MARK2 = 0) OR
ODD(MARK2=2)
C      IF IIELEM IS EVEN, JJCOL=JCOL1(IINODE)
C      IF IIELEM IS ODD, JJCOL=JCOL1(IINODE)-1
C
           TEMP=IIELEM
           XREAL =TEMP/2.
           I=IIELEM/2
           DIFF=ABS(XREAL-I)
           MARK2=2
           IF(DIFF .LT. 0.01)MARK2=0
C
C          DETERMINE AREA & NODE FOR PARALLEL ELEMENT
C          IN3 = NODE 3 INTERFACE THAT WE'RE WORKING ON
C
C
             IF(IIELEM .EQ. 2) THEN
             NODE2=INODE
             IAREA2=JAREA(1)
             END IF
             IF(IIELEM .EQ. 3 .OR. IIELEM .EQ. 4)THEN
             NODE2=N3(IN3,2)
             IAREA2=JAREA(2)
             END IF
             IF(IIELEM .EQ. 5 .OR. IIELEM .EQ. 6) THEN
             NODE2=N3(IN3,3)
             IAREA2=JAREA(3)
             END IF
           JJCOL1=JCOL1(NODE1)-MARK1
           JJCOL2=JCOL1(NODE2)-MARK2
           GRATIO=((CSRAT(IAREA1)/CSRAT(IAREA2))**2)*
     &     UWTRAT(IAREA1)/UWTRAT(IAREA2)
       ISTART=2*JNODE(IAREA2,1)-1
       ISTOP=2*KNODE(IAREA2,NBOUND(IAREA2))
           DO 160 IROW=ISTART,ISTOP
           GMAT(IROW,JMARK)=GMAT(IROW,JMARK)+
     &                          SIGN*GMAT(IROW,JJCOL2)*GRATIO
           GMAT(IROW,JMARK+1)=GMAT(IROW,JMARK+1)+
     &                            SIGN*GMAT(IROW,JJCOL2+1)*GRATIO
  160 CONTINUE
      END IF
  170 CONTINUE
  180 CONTINUE
```

```
      END IF
  190 CONTINUE
  200 CONTINUE
      GDIM=JMARK+1
  205 CONTINUE
C     WRITE(70,7000)
 7000 FORMAT(5X,'HMAT'/)
C     WRITE(70,*)'HDIM',HDIM
      JMAX=HDIM/2
C     WRITE(70,*)'ORDER',(NORDER(J),J=1,JMAX)
      DO 240I=1,2*KNODE(NAREA,NBOUND(NAREA))
C     WRITE(70,7005)I
 7005 FORMAT(I4)
C     WRITE(70,7010)(HMAT(I,J),J=1,HDIM)
 7010 FORMAT(6F12.6)
  240 CONTINUE
C     WRITE(70,7020)
 7020 FORMAT(/5X,'GMAT'/)
C     WRITE(70,*)'GDIM',GDIM
      DO 250I=1,2*KNODE(NAREA,NBOUND(NAREA))
C     WRITE(70,7005)I
C     WRITE(70,7010)(GMAT(I,J),J=1,GDIM)
  250 CONTINUE
C     ESTABLISH NEW JCOL1
      II=0
      DO 270 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(NCONN(INODE) .EQ. 0)THEN
C     NEXT NODE IN NORDER
      II=II+1
      JCOL1(INODE)=II*2-1
      IF(IEQN(INODE) .EQ. 3)II=II+1
      END IF
  270 CONTINUE
      NNODE=II-NN3
C     WRITE(70,7030)
 7030 FORMAT(/5X,'NODE,JCOL1')
      DO 280I=1,NNODE
      INODE=NORDER(I)
C     WRITE(70,7060)INODE,JCOL1(INODE)
 7060 FORMAT(2I5)
  280 CONTINUE
C     CLOSE(UNIT=70)
      RETURN
      END
```

```
      SUBROUTINE SOLVE(MARK)
C
C     MARK=1; FREE-FIELD CALCULATION
C     MARK=2; BEM CALCULATION
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/AMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX AMAT,GMAT,FVECT,XVECT
C
      DOUBLE PRECISION A(1560,1561),X(1560),PIVOT,TEMP,ANORM
      INTEGER IX(1560)
C
C     ****************************************************
C     ****************************************************
C     *****      SUBROUTINE FINDS THE SOLUTION OF      *****
C     *****      SIMULTANEOUS EQUATIONS USING          *****
C     *****      MAXIMUM PIVOT PROCEDURE               *****
C     ****************************************************
C     ****************************************************
C     II REPRESENTS THE REAL PART, II+1 REPRESENTS THE IMAGINARY PART
C     JJ REPRESENTS THE REAL PART, JJ+1 REPRESENTS THE IMAGINARY PART
C
            IF(MARK .EQ. 1)THEN
      N=2.*FFDIM
      NROW=FFDIM
      DO 10I=1,FFDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 10J=1,FFDIM
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(STIFF(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(STIFF(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(STIFF(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(STIFF(I,J))
   10 CONTINUE
            END IF
C
            IF (MARK .EQ. 2)THEN
      N=2.*NDIM
      NROW=NDIM
      DO 15I=1,NDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
```

```
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 15J=1,NDIM
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(AMAT(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(AMAT(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(AMAT(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(AMAT(I,J))
   15 CONTINUE
            END IF
C
C     OPEN(UNIT=20,FILE='MAT.OUT',STATUS='UNKNOWN')
C     WRITE(20,2000)ANAME
C 2000 FORMAT(//5X,A80)
C     WRITE(20,2010)
C 2010 FORMAT(/10X,'A matrix * X vector = C vector')
C     WRITE(20,2020)
C 2020 FORMAT( 5X,'A matrix:')
C      DO 20I=1,N
C     WRITE(20,2030)(A(I,J),J=1,N)
C 2030 FORMAT(8F9.4)
C   20 CONTINUE
C     WRITE(20,2040)
C 2040 FORMAT(/5X,'C vector:')
C     WRITE(20,2030)(A(J,N+1),J=1,N)
C
C     ******************************************************
C     ******************************************************
C     *SOLVE FOR X VECTOR USING GAUSS'S ELIMINATION METHOD *
C     ************   MAXIMUM PIVOT IS USED *****************
C     ******************************************************
C     ******************************************************
C
C
C
      DO 30I=1,N
      IX(I)=I
   30 CONTINUE
      DO 100IROW=1,N
      I=IROW/10
      XX=IROW
      XX=XX/10.
      DIFF=ABS(XX-I)
      IF(DIFF .LT. 0.0001)THEN
      WRITE(*,9000)IROW,N
 9000 FORMAT(5X,'WORKING ON COLUMN # ',I4,' / ',I4)
      END IF
      JCOL=IROW
      PIVOT=0.0D0
C     FIND PIVOT VALUE
      DO 40IIROW=IROW,N
      DO 40JJCOL=JCOL,N
      IF (ABS(A(IIROW,JJCOL)) .GT. ABS (PIVOT)) THEN
      PIVOT =A(IIROW,JJCOL)
      IMARK=IIROW
      JMARK=JJCOL
      END IF
   40 CONTINUE
C     WRITE(*,*)'PIVOT = ',PIVOT
```

```
      IF (ABS(PIVOT) .LT. 1.0D-15) THEN
      WRITE(*,9010)
 9010 FORMAT(////15X,'** MATRIX DOES NOT HAVE AN INVERSE,')
      WRITE(*,9020)
 9020 FORMAT(15X,'PUSH RETURN  **')
      PAUSE
      STOP
      END IF
C     INTERCHANGE ROWS
      DO 50 JJCOL=1,N+1
      TEMP=A(IROW,JJCOL)
      A(IROW,JJCOL)=A(IMARK,JJCOL)
      A(IMARK,JJCOL)=TEMP
   50 CONTINUE
C     INTERCHANGE COLUMNS
      ITEMP=IX(JCOL)
      IX(JCOL)=IX(JMARK)
      IX(JMARK)=ITEMP
      DO 60 IICOL=1,N
      TEMP=A(IICOL,JCOL)
      A(IICOL,JCOL)=A(IICOL,JMARK)
      A(IICOL,JMARK)=TEMP
   60 CONTINUE
      DO 65 I=1,N
   65 CONTINUE
C     NORMALIZE ROW OF PIVOT ELEMENT
      DO 70 JJCOL=JCOL,N+1
      A(IROW,JJCOL)=A(IROW,JJCOL)/PIVOT
   70 CONTINUE
C     SUBTRACT ROW FROM REMAINING ROWS
      DO 80 IIROW=IROW+1,N
      ANORM=-1.0D0*A(IIROW,JCOL)
      DO 80 JJCOL=JCOL,N+1
      A(IIROW,JJCOL)=A(IIROW,JJCOL)+ANORM *A(IROW,JJCOL)
   80 CONTINUE
      DO 85 I=1,N
   85 CONTINUE
  100 CONTINUE
C     USE BACKWARD SUBSTITUTION
      X(N)=A(N,N+1)/A(N,N)
      DO 120IROW=N-1,1,-1
      SUM=0.0
      DO 110 JCOL=IROW+1,N
      SUM=SUM+A(IROW,JCOL)*X(JCOL)
  110 CONTINUE
      X(IROW)=(A(IROW,N+1)-SUM)/A(IROW,IROW)
  120 CONTINUE
C     REORDER X VECTOR
      DO 130 IROW=1,N
      A(IROW,N+1)=X(IROW)
  130 CONTINUE
      DO 140 I=1,N
      IROW=IX(I)
      X(IROW)=A(I,N+1)
  140 CONTINUE
C
C     WRITE(20,2060)
 2060 FORMAT(//5X,'X vector:')
```

```
C        WRITE(20,2070)(X(J),J=1,N)
 2070 FORMAT(5X,8F10.4)
C     CLOSE (UNIT=20)
      DO 150J=1,NROW
      JJ=2*J-1
      XVECT(J)=X(JJ)+(0.0,1.0)*X(JJ+1)
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SORT
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     MATRIX
      COMMON/IMATRIX/NDIM,GDIM,HDIM,ADIM,JCOL1(790)
      COMMON/CMATRIX/HMAT(780,780),GMAT(780,790),
     & FVECT(780),XVECT(780)
      INTEGER NDIM,GDIM,HDIM,ADIM,JCOL1
      COMPLEX HMAT,GMAT,FVECT,XVECT
C
C     OUTPUT DATA
      COMMON/COUT/UDISPL(390),WDISPL(390),PXH1(390,2),PZH1(390,2)
      COMPLEX UDISPL,WDISPL,PXH1,PZH1
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
      INTEGER ISKIP(6),JAREA(3)
      REAL XN1(6),XN2(6)
C
      COMPLEX GRATIO
C
C23456789112345678921234567893123456789412345678951234567896123456789712
      DO 10I=1,HDIM/2
      IROW=2*I-1
      INODE=NORDER(I)
      UDISPL(INODE)=XVECT(IROW)
      WDISPL(INODE)=XVECT(IROW+1)
   10 CONTINUE
      DO 20 IINODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF (IEQN(IINODE) .EQ. 0)THEN
      INODE=NCONN(IINODE)
      UDISPL(IINODE)=UDISPL(INODE)
      WDISPL(IINODE)=WDISPL(INODE)
      END IF
   20 CONTINUE
C
C
      ICOUNT= HDIM/2
      IN3=0
```

```
C
      DO 150 INODE=1,KNODE(NAREA,NBOUND(NAREA))
      IF(NSURF(INODE) .EQ. 1 .AND. IEQN(INODE) .EQ. 1)THEN
      PXH1(INODE,1)=0.0
      PZH1(INODE,1)=0.0
      END IF
      IF(NSURF(INODE) .EQ. 0 .AND. IEQN(INODE) .EQ. 1)THEN
      ICOUNT=ICOUNT+1
      IROW=ICOUNT*2-1
      PXH1(INODE,1)=XVECT(IROW)
      PZH1(INODE,1)=XVECT(IROW+1)
      END IF
      IF(IEQN(INODE) .EQ. 2)THEN
      ICOUNT=ICOUNT+1
      IROW=ICOUNT*2-1
      PXH1(INODE,1)=XVECT(IROW)
      PZH1(INODE,1)=XVECT(IROW+1)
      DO 30IINODE=INODE,KNODE(NAREA,NBOUND(NAREA))
      IF(NCONN(IINODE) .EQ. INODE) THEN
      NODE2=IINODE
      GO TO 40
      END IF
   30 CONTINUE
   40 CONTINUE
      DO 50 IAREA=1,NAREA
      DO 50 IINODE=JNODE(IAREA,1),KNODE(IAREA,NBOUND(IAREA))
      IF(IINODE .EQ. INODE)JAREA(1)=IAREA
      IF(IINODE .EQ. NODE2)JAREA(2)=IAREA
   50 CONTINUE
      GRATIO=((CSRAT(JAREA(1))/CSRAT(JAREA(2)))**2)*
     & UWTRAT(JAREA(1))/UWTRAT(JAREA(2))
      PXH1(NODE2,1)= -1.*PXH1(INODE,1)*GRATIO
      PZH1(NODE2,1)= -1.*PZH1(INODE,1)*GRATIO
      END IF
      IF(IEQN(INODE) .EQ. 3)THEN
      IN3=IN3+1
      DO 60 I=1,3
      IINODE=N3(IN3,I)
      IELEM=2*I-1
        SEGLEN=SQRT((X(IINODE)-X(IINODE-1))**2+
     &  (Z(IINODE)-Z(IINODE-1))**2)
        XN1(IELEM)=(Z(IINODE-1)-Z(IINODE))/SEGLEN
        XN2(IELEM)=(X(IINODE)-X(IINODE-1))/SEGLEN
        SEGLEN=SQRT((X(IINODE+1)-X(IINODE))**2+
     &  (Z(IINODE+1)-Z(IINODE))**2)
        XN1(IELEM+1)=(Z(IINODE)-Z(IINODE+1))/SEGLEN
        XN2(IELEM+1)=(X(IINODE+1)-X(IINODE))/SEGLEN
   60 CONTINUE
      DO 80I=1,3
      DO 70 IAREA=1,NAREA
      DO 70 IINODE=1,KNODE(IAREA,NBOUND(IAREA))
      IF(IINODE .EQ. N3(IN3,I))THEN
      JAREA(I)=IAREA
      GO TO 80
      END IF
   70 CONTINUE
   80 CONTINUE
      DO 90 IELEM=1,6
```

```
      ISKIP(IELEM)=0
   90 CONTINUE
      DO 110 IELEM=1,5
      IF(ISKIP(IELEM) .EQ. 1)GO TO 110
C
C     DETERMINE IF ELEMENT IELEM IS EVEN (MARK1 = 2) OR ODD(MARK1=1)
      TEMP=IELEM
      XREAL =TEMP/2.
      I=IELEM/2
      DIFF=ABS(XREAL-I)
      MARK1=1
      IF(DIFF .LT. 0.01)MARK1=2
C
      IF(IELEM .EQ. 1 .OR. IELEM .EQ. 2) THEN
      NODE1=INODE
      IAREA1=JAREA(1)
      END IF
      IF(IELEM .EQ. 3 .OR. IELEM .EQ. 4)THEN
      NODE1=N3(IN3,2)
      IAREA1=JAREA(2)
      END IF
      IF(IELEM .EQ. 5 .OR. IELEM .EQ. 6) THEN
      NODE1=N3(IN3,3)
      IAREA1=JAREA(3)
      END IF
      ICOUNT=ICOUNT+1
      IROW=ICOUNT*2-1
      PXH1(NODE1,MARK1)=XVECT(IROW)
      PZH1(NODE1,MARK1)=XVECT(IROW+1)
C
      DO 100 IIELEM=IELEM+1,6
        DIFF=ABS(ABS(XN1(IELEM))-ABS(XN1(IIELEM)))
          IF (DIFF .LT. 0.01)THEN
          DIFF1=ABS(XN1(IELEM)-XN1(IIELEM))
          DIFF2=ABS(XN2(IELEM)-XN2(IIELEM))
          IF(DIFF1 .LT. 0.01 .AND. DIFF2 .LT. 0.01) THEN
          SIGN =+1.0
          ELSE
          SIGN = -1.0
          END IF
      ISKIP(IIELEM)=1
C     DETERMINE AREA & NODE FOR PARALLEL ELEMENT
C     IN3 = NODE 3 INTERFACE THAT WE'RE WORKING ON
C
C
C     DETERMINE IF ELEMENT IIELEM IS EVEN (MARK2 = 2) OR ODD(MARK2=1)
      TEMP=IIELEM
      XREAL =TEMP/2.
      I=IIELEM/2
      DIFF=ABS(XREAL-I)
      MARK2=1
      IF(DIFF .LT. 0.01)MARK2=2
C
      IF(IIELEM .EQ. 2) THEN
      NODE2=INODE
      IAREA2=JAREA(1)
      END IF
      IF(IIELEM .EQ. 3 .OR. IIELEM .EQ. 4)THEN
```

```
      NODE2=N3(IN3,2)
      IAREA2=JAREA(2)
      END IF
      IF(IIELEM .EQ. 5 .OR. IIELEM .EQ. 6) THEN
      NODE2=N3(IN3,3)
      IAREA2=JAREA(3)
      END IF
      GRATIO=((CSRAT(IAREA1)/CSRAT(IAREA2))**2)*
     & UWTRAT(IAREA1)/UWTRAT(IAREA2)
      PXH1(NODE2,MARK2)=PXH1(NODE1,MARK1)*GRATIO*SIGN
      PZH1(NODE2,MARK2)=PZH1(NODE1,MARK1)*GRATIO*SIGN
      END IF
  100 CONTINUE
  110 CONTINUE
      END IF
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(ANAME)
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     GEOMETRY
      COMMON/IGEOM/NBOUND(4),JNODE(4,3),KNODE(4,3),NAREA,ICLOSE(4,3),
     & NCONN(390),NORDER(390),NNSURF,NSURF(390),N3(2,3),NN3
      COMMON/RGEOM/X(390),Z(390)
      INTEGER NBOUND,JNODE,KNODE,NAREA,ICLOSE,NCONN,NORDER,NNSURF,
     & NSURF,N3,NN3
      REAL X,Z
C
C     WAVE
      COMMON/IWAVE/IANGLE,NANGLE
      COMMON/RWAVE/ANGLE(4)
      COMMON/CWAVE/FBAR,UINCDT(200),WINCDT(200)
      COMMON/AWAVE/ATYPE
      REAL ANGLE
      COMPLEX FBAR,UINCDT,WINCDT
      CHARACTER*2 ATYPE
C
C
C     SOIL
      COMMON/ISOIL/HALFSP
      COMMON/RSOIL/UWTRAT(4),BETA(4),POISS(4)
      COMMON/CSOIL/CSRAT(4)
      INTEGER HALFSP
      REAL UWTRAT,BETA,POISS
      COMPLEX CSRAT
C
C     FREE-FIELD
      COMMON/IFFLD/LTYPE(2,3),NTOP(2,3),NBOTT(2,3),NLAYER(2),
     & JFF(2,3),KFF(2,3),FFDIM
      COMMON/RFFLD/XSCATT,XFF(2,50),ZFF(2,50)
      COMMON/CFFLD/FFU(2,50),FFW(2,50),TCORR(390,2),
     & HSU(2),HSW(2),FFPX(2,50),FFPZ(2,50),HSPX(2),HSPZ(2),
     & STIFF(20,20)
      INTEGER LTYPE,NTOP,NBOTT,NLAYER,JFF,KFF,FFDIM
      REAL XSCATT,XFF,ZFF
      COMPLEX FFU,FFW,TCORR,HSU,HSW,FFPX,FFPZ,HSPX,HSPZ,STIFF
C
C     OUTPUT DATA
      COMMON/COUT/UDISPL(390),WDISPL(390),PXH1(390,2),PZH1(390,2)
      COMPLEX UDISPL,WDISPL,PXH1,PZH1
C
C     NUMBER OF EQUATIONS AT EACH NODE
      COMMON/EQN/IEQN(390)
      INTEGER IEQN
C
      CHARACTER*80 ANAME
C
      WRITE(60,6000)
 6000 FORMAT(//5X,'OUTPUT')
      WRITE(60,6001) ANAME
 6001 FORMAT(/5X,A80)
      WRITE(60,6002)ATYPE
 6002 FORMAT(5X,'TYPE OF WAVE = ',A2)
      X1=FBAR*CSQRT(1.0-(0.0,2.0)*BETA(1))
      WRITE(60,6003)X1
```

```
 6003 FORMAT(5X,'DIMENSIONLESS FREQUENCY (omega*H1/Cs1) = ',F6.3)
      WRITE(60,6004)IANGLE,ANGLE(IANGLE)
 6004 FORMAT(5X,'INCIDENT ANGLE (',I1,') WITH HORIZONTAL = ',F5.2,
     & ' degs')
      WRITE(60,6005)NAREA
 6005 FORMAT(5X,'NUMBER OF HOMOGENEOUS REGIONS = ',I1)
      WRITE(60,6006)
 6006 FORMAT(/5X,'BOUNDARY VALUES')
      DO 60IAREA=1,NAREA
      WRITE(60,6010)IAREA
 6010 FORMAT(//5X,'HOMOGENEOUS REGION ',I1)
      WRITE(60,6012)NBOUND(IAREA)
 6012 FORMAT(5X,'NUMBER OF BOUNDARIES = ',I1)
      DO 60IBOUND=1,NBOUND(IAREA)
      WRITE(60,6015)IBOUND
 6015 FORMAT(/5X,'BOUNDARY = ',I2)
      NNODE=KNODE(IAREA,IBOUND)-JNODE(IAREA,IBOUND)+1
      WRITE(60,6017)NNODE
 6017 FORMAT(5X,'NUMBER OF NODES = ',I3)
      WRITE(60,6018)
 6018 FORMAT(/2X,'DISPLACEMENTS')
      WRITE(60,6020)
 6020 FORMAT( /T3,'NODE',T13,'X/H1',T23,'Z/H1',T42,'U DISPL',T64,'|U|')
      DO 30 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      XREAL=REAL(UDISPL(INODE))
      XIMAG=AIMAG(UDISPL(INODE))
      AMPL=SQRT((XREAL**2)+(XIMAG**2))
      WRITE(60,6030)INODE,X(INODE),Z(INODE),UDISPL(INODE),AMPL
 6030 FORMAT(T4,I3,T11,F8.3,T21,F8.3,T37,2F8.3,T61,F8.3)
   30 CONTINUE
      WRITE(60,6035)
 6035 FORMAT(/T3,'NODE',T13,'X/H1',T23,'Z/H1',T42,'W DISPL',T64,'|W|')
            DO INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
      XREAL=REAL(WDISPL(INODE))
      XIMAG=AIMAG(WDISPL(INODE))
      AMPL=SQRT((XREAL**2)+(XIMAG**2))
      WRITE(60,6037)INODE,X(INODE),Z(INODE),WDISPL(INODE),AMPL
 6037 FORMAT(T4,I3,T11,F8.3,T21,F8.3,T37,2F8.3,T61,F8.3)
            END DO
C
            IF(NAREA .GT. 1)THEN
      WRITE(60,6038)
 6038 FORMAT(/2X,'TRACTIONS')
      WRITE(60,6040)
 6040 FORMAT(/T3,'NODE',T22,'PxH1/Gi',T42,'PzH1/Gi')
      DO 55 INODE=JNODE(IAREA,IBOUND),KNODE(IAREA,IBOUND)
C
      PXH1(INODE,1)=PXH1(INODE,1)*(1.0-(0.0,2.0)*BETA(IAREA))
      PZH1(INODE,1)=PZH1(INODE,1)*(1.0-(0.0,2.0)*BETA(IAREA))
C
      WRITE(60,6050)INODE,PXH1(INODE,1),PZH1(INODE,1)
 6050 FORMAT(T4,I3,T15,2F10.4,T35,2F10.4)
      DO 50 IN3=1,NN3
      DO 40 J=1,3
      IF(N3(IN3,J) .EQ. INODE)THEN
C
      PXH1(INODE,2)=PXH1(INODE,2)*(1.0-(0.0,2.0)*BETA(IAREA))
      PZH1(INODE,2)=PZH1(INODE,2)*(1.0-(0.0,2.0)*BETA(IAREA))
```

```
C
      WRITE(60,6060)PXH1(INODE,2),PZH1(INODE,2)
 6060 FORMAT(T15,2F10.4,T35,2F10.4)
      GO TO 55
      END IF
   40 CONTINUE
   50 CONTINUE
   55 CONTINUE
            END IF
   60 CONTINUE
      RETURN
      END
```

A.2: One-Dimensional Codes

For the one-dimensional analysis, two codes, one for anti-plane and one for in-plane motion, were developed by the writer to calculate the surface transfer and amplification function of a unit incident wave. Dimensionless properties are used to define the soil column with characteristic properties based on soil layer 1 . The anti-plane code, "shffld", determines the surface values at increments of the dimensionless frequency equal to 0.05 for a range between 0 and 10.0 . Surface values from the in-plane code, "psvffld", are determined at increments of the dimensionless frequency equal to 0.025 for a range between 0 and 5.0 . The codes are independent of outside call libraries and are therefore "stand-alone" programs. Both the anti-plane code and in-plane codes solve the surface transfer function using the matrix method. Although the anti-plane motion could be solved more quickly using a ratio method similar to the method used in CARES (Costantino, Miller, et. al., 1991) or SHAKE (Schnabel, et. al., 1981) , the matrix method was used in order to have a similar approach for both the in-plane and anti-plane codes. For the number of layers considered in these calculations, the time difference is not considered significant between the ratio and matrix methods. To solve the simultaneous equations, the same routine is used as that used for the boundary element code.

A sample of an input file for "psvffld" is described in the following for a soil profile of 2 layers on a half-space.

Two Viscoelastic Soil Layers on a Elastic Half-Space

Line 1	Output file name using a maximum of 20 characters and beginning at column 1.
Line 2	Title for output file using a maximum of 80 characters and beginning at column 1.
Line 3	Number of layers

Since the dimensionless ratios are based on the shear velocity ratio, thickness, and unit weight of soil layer 1, only the Poisson ratio and hysteretic damping are required for layer 1

Line 4 Poisson ratio for layer 1

Line 5 Hysteretic damping for layer 1 (%)

If more than 1 layer is involved in the calculation, properties are inputted for the additional layers.

Line 6 Depth ratio to the bottom of soil layer 2 (x_3 (layer 2) / H1 (layer 1)).

Line 7 Shear velocity ratio of layer 2 (Cs (layer 2) / Cs (layer 1)) .

Line 8 Unit weight ratio of layer 2 (unit wt. (layer 2) / unit wt. (layer 1))

Line 9 Poisson ratio of layer 2.

Line 10 Hysteretic damping (%) for layer 2.

Lines 6 through 10 are repeated for the number of soil layers above 2 .

The half-space properties are then entered.

Line 11 Shear velocity ratio of the half-space (Cs (half-space) / Cs (layer 1)) .

Line 12 Unit weight ratio of the half-space: unit wt. (half-space) / unit wt. (layer 1)

Line 13 Poisson ratio of the half-space.

Line 14 Hysteretic damping (%) for the half-space.

The soil properties which define the soil column are completed. Next, the parameters which define the half-space incident motion are introduced.

Line 15 Type of incident wave. For "shffld" this would be SH (anti-plane shear wave). For "psvffld" this could either be SV (in-plane shear wave) or P (in-plane compression wave).

Line 16 Number of half-space angles of incidence to calculate for.

Line 17 Half-space angle of incidence in degrees.

Line 17 is repeated for the number of half-space angles of incidence indicated on line 16 .

To execute either "shffld" or "psvffld" a run file is used which includes the number of problems and the names of the input files similar to what is used for the boundary element codes.

SAMPLE INPUT FILE FOR THE ONE-DIMENSIONAL ANLAYSIS
OF 2 LAYERS ON A HALF-SPACE

LINE #	COLUMN # OF INPUT FILE
	1
1	Pffld1d.2lou
2	2 LAYERS ON HALF-SPACE; ONE-DIMENSIONAL ANALYSIS
3	2
4	0.25
5	5.0
6	2.0
7	2.5
8	1.32
9	0.25
10	2.0
11	10.0
12	1.32
13	0.25
14	0.0
15	P
16	4
17	90.0
18	60.0
19	45.0
20	10.0

A.2.1: ONE-DIMENSIONAL ANALYSIS CODE

shffld

```
      PROGRAM FFLD
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     MATRIX
      COMMON/IMATRIX/NDIM
      COMMON/CMATRIX/STIFF(100,100), FVECT(100),XVECT(100)
      INTEGER NDIM
      COMPLEX STIFF,FVECT,XVECT
C

      REAL HRATIO(10),FFWTR(10),BETA(10),ANGLE(5)
      COMPLEX VTOP,ASH(10),BSH(10),FFCSR(10),THEATA(10),CALC,
     & XIKTH1,XIKTH2,XMX(10),XI,XIKTZ,VOTCRP,FBAR

C
      CHARACTER*20 AOUT,ARUN,AINPT
      CHARACTER*80 ATITLE
      CHARACTER*2 ATYTPE
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C
      WRITE(*,9000)
 9000 FORMAT(/10X,'FREE-FIELD CALCULATION',/)
      WRITE(*,9010)
 9010 FORMAT(/5X,'ENTER RUN FILE NAME')
      READ(*,9020)ARUN
 9020 FORMAT(A20)
      OPEN(UNIT=10,FILE=ARUN,STATUS='UNKNOWN')
      READ(10,*)NPROB
      DO IPROB=1,NPROB
      READ(10,1000)AINPT
 1000 FORMAT(A20)
      OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
      READ(50,5000)AOUT
 5000 FORMAT(A20)
      READ(50,5010)ATITLE
 5010 FORMAT(A80)
      READ(50,*)NLAYER
      DO 10 ILAYER=1,NLAYER +1
      IF(ILAYER .EQ. 1)THEN
      HRATIO(1)=1.
      FFCSR(1)=(1.0,0.0)
      FFWTR(1)=1.0
      READ(50,*)POISS
      READ(50,*)BETA(1)
      BETA(1)=BETA(1)/100.0
      ELSE
      IF( ILAYER .LE. NLAYER)THEN
      READ(50,*)HRATIO(ILAYER)
      READ(50,*)X
      FFCSR(ILAYER)=X
      READ(50,*)FFWTR(ILAYER)
      READ(50,*)POISS
      READ(50,*)BETA(ILAYER)
      BETA(ILAYER)=BETA(ILAYER)/100.0
      ELSE
      READ(50,*)X
```

```
      FFCSR(ILAYER)=X
      READ(50,*)FFWTR(ILAYER)
      READ(50,*)POISS
      READ(50,*)BETA(ILAYER)
      BETA(ILAYER)=BETA(ILAYER)/100.0
      END IF
      END IF
   10 CONTINUE
      READ(50,5020)ATYTPE
 5020 FORMAT(A2)
      READ(50,*)NANGLE
      DO IANGLE = 1,NANGLE
      READ(50,*)ANGLE(IANGLE)
      END DO
      CLOSE(UNIT=50)
C
      ASH(NLAYER+1)=(1.0,0.0)
C
      OPEN(UNIT=60,FILE=AOUT, STATUS='UNKNOWN')
      WRITE(60,6000)
 6000 FORMAT(/10X,'ANTI-PLANE FREE-FIELD CALCULATION')
      WRITE(60,6002)ATITLE
 6002 FORMAT(/5X,A80,/)
      DO 25 ILAYER=1,NLAYER +1
      IF(ILAYER .LE. NLAYER)THEN
      WRITE(60,6005)ILAYER
 6005 FORMAT(/5X,'LAYER: ',I2)
      WRITE(60,6010)HRATIO(ILAYER)
 6010 FORMAT(2X,'Z bottom(i) / Z bottom(1) = ',T35,F7.3)
      X=REAL(FFCSR(ILAYER))
      WRITE(60,6020)X
 6020 FORMAT(2X,'Cs(i)/Cs(1) = ',T35,F7.3)
      X=REAL(FFWTR(ILAYER))
      WRITE(60,6030)X
 6030 FORMAT(2X,'Wt(i)/Wt(1) = ',T35,F7.3)
      WRITE(60,6040)BETA(ILAYER)*100.
 6040 FORMAT(2X,'DAMPING = ',T35,F7.3,' %')
      ELSE
      WRITE(60,6050)
 6050 FORMAT(/5X,'HALF-SPACE')
      X=REAL(FFCSR(ILAYER))
      WRITE(60,6060)X
 6060 FORMAT(2X,'Cs(half-space)/Cs(1) = ',T35,F7.3)
      X=REAL(FFWTR(ILAYER))
      WRITE(60,6070)X
 6070 FORMAT(2X,'Wt(half-space)/Wt(1) = ',T35,F7.3)
      WRITE(60,6040)BETA(ILAYER)*100.
      END IF
   25 CONTINUE
      WRITE(60,6075)
 6075 FORMAT(/2X,'INCIDENT WAVE TYPE = SH')
      WRITE(60,6077)NANGLE
 6077 FORMAT(2X,'NUMBER OF ANGLES OF INCIDENCE = ',I1)
      DO IANGLE=1,NANGLE
      WRITE(60,6078)IANGLE,ANGLE(IANGLE)
 6078 FORMAT(2X,'ANGLE (',I1,') = ',F7.3,' degs')
      END DO
C
```

```
      NFREQ=200
      DO 30 ILAYER=1,NLAYER+1
      FFCSR(ILAYER)=FFCSR(ILAYER)*CSQRT((1.0-(0.0,2.0)*BETA(ILAYER))/
     & (1.0-(0.0,2.0)*BETA(1)))
   30 CONTINUE
      NDIM=2*NLAYER+1
C
      DO IANGLE = 1,NANGLE
      THEATA(NLAYER+1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      XMX(NLAYER+1)=CCOS(THEATA(NLAYER+1))
            DO ILAYER=1,NLAYER
            XMX(ILAYER)=XMX(NLAYER+1)*FFCSR(ILAYER)/FFCSR(NLAYER+1)
            END DO
C
      WRITE(60,6080)ANGLE(IANGLE)
 6080 FORMAT(//2X,'ANGLE OF INCIDENCE = ',T35,F7.3,' degs')
      WRITE(60,6090)
 6090 FORMAT(/T3,'FREQ',T17,'V(top)',T31,'|V(top)|',
     &             T45,'V(outcrop)',T60,'|V(outcrop)|',T77,'V(ampl)'/)
      DO 120 IFREQ=1,NFREQ
      RFBAR=IFREQ*.05
      FBAR=RFBAR/CSQRT(1.0-(0.0,2.0)*BETA(1))
      DO 40I=1,NDIM
      FVECT(I)=(0.0,0.0)
      DO 40J=1,NDIM
      STIFF(I,J)=(0.0,0.0)
   40 CONTINUE
C
C
      STIFF(1,1)=(1.0,0.0)
      STIFF(1,2)=(-1.0,0.0)
      DO 50ILAYER=1,NLAYER
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER)**2))
      XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER+1)**2))
      I=(ILAYER-1)*2+1+1
      J=(ILAYER-1)*2+1
      XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
     & * FFWTR(ILAYER+1)/FFWTR(ILAYER)
      CALC= XI* CSQRT(1.0-(XMX(ILAYER+1)**2))/
     & CSQRT(1.0-(XMX(ILAYER)**2)) * FFCSR(ILAYER)/FFCSR(ILAYER+1)
      STIFF(I,J)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I,J+1)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+1,J)   = -1.0*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+1,J+1)   = +1.0*CEXP((+1.0,0.0)*XIKTH1)
      IF(ILAYER .LT. NLAYER)THEN
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+3)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+2)   = +1.0*CALC*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+1,J+3)   = -1.0*CALC*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+2)   = -1.0*CALC*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)*ASH(NLAYER+1)
      FVECT(I+1)=(-1.0,0.0)*CALC*CEXP((-1.0,0.0)*XIKTH2)*
     & ASH(NLAYER+1)
      END IF
```

```
   50 CONTINUE
      CALL SOLVE
      DO 70ILAYER=1,NLAYER
      IROW=(ILAYER-1)*2+1
      ASH(ILAYER)=XVECT(IROW)
      BSH(ILAYER)=XVECT(IROW+1)
   70 CONTINUE
      BSH(NLAYER+1)=XVECT(NDIM)
      VTOP=(ASH(1)+BSH(1))
      VTMAG=SQRT(REAL(VTOP)**2+AIMAG(VTOP)**2)
C     DO 90ILAYER=1,NLAYER+1
C     WRITE(*,6010)ILAYER,ASH(ILAYER),BSH(ILAYER)
C   90 CONTINUE
C
C
      NN=NLAYER+1
C
C
      ZHS=HRATIO(NLAYER)
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XMX(NN)**2))*ZHS
C
C     HALF-SPACE OUTCROP
      VOTCRP= 2.0*ASH(NN)*CEXP(-XIKTZ)
      VOTMAG=SQRT(REAL(VOTCRP)**2+AIMAG(VOTCRP)**2)
C
      VAMPL=VTMAG/VOTMAG
      WRITE(60,6100)RFBAR,VTOP,VTMAG,VOTCRP,VOTMAG,VAMPL
 6100 FORMAT(2X,F6.3,T13,2F7.2,T32,F7.2,T43,2F7.2,T62,F7.2,T77,F7.2)
  120 CONTINUE
      END DO
      CLOSE (UNIT=60)
      END DO
      CLOSE(UNIT=10)
      STOP
      END
```

```
      SUBROUTINE SOLVE
C
C     MATRIX
      COMMON/IMATRIX/NDIM
      COMMON/CMATRIX/AMAT(100,100), FVECT(100),XVECT(100)
      INTEGER NDIM
      COMPLEX AMAT,FVECT,XVECT
C
      DOUBLE PRECISION A(200,201),X(200),PIVOT,TEMP,ANORM
      INTEGER IX(200)
C
C     *****************************************************
C     *****************************************************
C     *****      SUBROUTINE FINDS THE SOLUTION OF      *****
C     *****      SIMULTANEOUS EQUATIONS USING          *****
C     *****      MAXIMUM PIVOT PROCEDURE               *****
C     *****************************************************
C     *****************************************************
C     II REPRESENTS THE REAL PART, II+1 REPRESENTS THE IMAGINARY PART
C     JJ REPRESENTS THE REAL PART, JJ+1 REPRESENTS THE IMAGINARY PART
C
      N=2.*NDIM
      DO 10I=1,NDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 10J=1,NDIM
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(AMAT(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(AMAT(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(AMAT(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(AMAT(I,J))
   10 CONTINUE
C     OPEN(UNIT=20,FILE='MAT.OUT',STATUS='UNKNOWN')
C     WRITE(20,2000)ANAME
 2000 FORMAT(//5X,A80)
C     WRITE(20,2010)
 2010 FORMAT(/10X,'A matrix * X vector = C vector')
C     WRITE(20,2020)
 2020 FORMAT( 5X,'A matrix:')
      DO 20I=1,N
C     WRITE(20,2030)(A(I,J),J=1,N)
 2030 FORMAT(5X,8F10.4)
   20 CONTINUE
C     WRITE(20,2040)
 2040 FORMAT(/5X,'C vector:')
C     WRITE(20,2030)(A(J,N+1),J=1,N)
C
C     *****************************************************
C     *****************************************************
C     *SOLVE FOR X VECTOR USING GAUSS'S ELIMINATION METHOD *
C     ************   MAXIMUM PIVOT IS USED *****************
C     *****************************************************
C     *****************************************************
C
C
C
      DO 30I=1,N
```

```
      IX(I)=I
   30 CONTINUE
      DO 100IROW=1,N
      I=IROW/10
      XX=IROW
      XX=XX/10.
      DIFF=ABS(XX-I)
C     IF(DIFF .LT. 0.0001)THEN
C     WRITE(*,9000)IROW,N
C 9000 FORMAT(5X,'WORKING ON COLUMN # ',I4,' / ',I4)
C     END IF
      JCOL=IROW
      PIVOT=0.0D0
C     FIND PIVOT VALUE
      DO 40IIROW=IROW,N
      DO 40JJCOL=JCOL,N
      IF (ABS(A(IIROW,JJCOL)) .GT. ABS (PIVOT)) THEN
      PIVOT =A(IIROW,JJCOL)
      IMARK=IIROW
      JMARK=JJCOL
      END IF
   40 CONTINUE
      IF (ABS(PIVOT) .LT. 1.0D-18) THEN
      WRITE(*,9010)
 9010 FORMAT(////15X,'** MATRIX DOES NOT HAVE AN INVERSE,')
      WRITE(*,9020)
 9020 FORMAT(15X,'PUSH RETURN  **')
      PAUSE
      STOP
      END IF
C     INTERCHANGE ROWS
      DO 50 JJCOL=1,N+1
      TEMP=A(IROW,JJCOL)
      A(IROW,JJCOL)=A(IMARK,JJCOL)
      A(IMARK,JJCOL)=TEMP
   50 CONTINUE
C     INTERCHANGE COLUMNS
      ITEMP=IX(JCOL)
      IX(JCOL)=IX(JMARK)
      IX(JMARK)=ITEMP
      DO 60 IICOL=1,N
      TEMP=A(IICOL,JCOL)
      A(IICOL,JCOL)=A(IICOL,JMARK)
      A(IICOL,JMARK)=TEMP
   60 CONTINUE
C     WRITE(20,*)'IROW = ',IROW,'†PIVOT = ',PIVOT
C     DO 65 I=1,N
C     WRITE(20,2050)(A(I,J),J=1,N+1)
C   65 CONTINUE
C     NORMALIZE ROW OF PIVOT ELEMENT
      DO 70 JJCOL=JCOL,N+1
      A(IROW,JJCOL)=A(IROW,JJCOL)/PIVOT
   70 CONTINUE
C     SUBTRACT ROW FROM REMAINING ROWS
      DO 80 IIROW=IROW+1,N
      ANORM=-1.0D0*A(IIROW,JCOL)
      DO 80 JJCOL=JCOL,N+1
      A(IIROW,JJCOL)=A(IIROW,JJCOL)+ANORM *A(IROW,JJCOL)
```

```
   80 CONTINUE
C      WRITE(20,*)'IROW = ',IROW,'  PIVOT = ',PIVOT
C      DO 85 I=1,N
C      WRITE(20,2050)(A(I,J),J=1,N+1)
C 2050 FORMAT(5F10.3)
C   85 CONTINUE
  100 CONTINUE
C      USE BACKWARD SUBSTITUTION
       X(N)=A(N,N+1)/A(N,N)
       DO 120IROW=N-1,1,-1
       SUM=0.0
       DO 110 JCOL=IROW+1,N
       SUM=SUM+A(IROW,JCOL)*X(JCOL)
  110 CONTINUE
       X(IROW)=(A(IROW,N+1)-SUM)/A(IROW,IROW)
  120 CONTINUE
C      REORDER X VECTOR
       DO 130 IROW=1,N
       A(IROW,N+1)=X(IROW)
  130 CONTINUE
       DO 140 I=1,N
       IROW=IX(I)
       X(IROW)=A(I,N+1)
  140 CONTINUE
C
C      WRITE(20,2060)
 2060 FORMAT(//5X,'X vector:')
C       WRITE(20,2070)(X(J),J=1,N)
 2070 FORMAT(5X,8F10.4)
C      CLOSE (UNIT=20)
       DO 150J=1,NDIM
       JJ=2*J-1
       XVECT(J)=X(JJ)+(0.0,1.0)*X(JJ+1)
  150 CONTINUE
       RETURN
       END
```

A.2.2: ONE-DIMENSIONAL ANALYSIS CODE

psvffld

```
      PROGRAM FFLD
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C     MATRIX
      COMMON/IMATRIX/NDIM
      COMMON/CMATRIX/STIFF(100,100), FVECT(100),XVECT(100)
      INTEGER NDIM
      COMPLEX STIFF,FVECT,XVECT
C

      REAL HRATIO(10),FFWTR(10),BETA(10),FFPOIS(10),RCPCS,ANGLE(5)
      COMPLEX UTOP,WTOP,AP(10),BP(10),ASV(10),BSV(10),FFCSR(10),
     & THEATA(10),CALC1,CALC2,CALC3,CALC4,CALC5,CALC6,CALC7,CALC8,
     & XIKSH1,XIKSH2,XIKTH1,XIKTH2,XLX(10),XMX(10),XI,XIKSZ,XIKTZ,
     & UOTCRP,WOTCRP,FBAR,APHS,BPHS,ASVHS,BSVHS
C
C
      CHARACTER*20 AOUT,ARUN,AINPT
      CHARACTER*80 ATITLE
      CHARACTER*2 ATYPE
C
C23456789112345678921234567893123456789412345678951234567896123456789712
C
C
C
      WRITE(*,9000)
 9000 FORMAT(/10X,'FREE-FIELD CALCULATION',/)
      WRITE(*,9010)
 9010 FORMAT(/5X,'ENTER RUN FILE NAME')
      READ(*,9020)ARUN
 9020 FORMAT(A20)
      OPEN(UNIT=10,FILE=ARUN,STATUS='UNKNOWN')
      READ(10,*)NPROB
      DO IPROB=1,NPROB
      READ(10,1000)AINPT
 1000 FORMAT(A20)
      OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
      READ(50,5000)AOUT
 5000 FORMAT(A20)
      READ(50,5010)ATITLE
 5010 FORMAT(A80)
      READ(50,*)NLAYER
      DO 10 ILAYER=1,NLAYER +1
      IF(ILAYER .EQ. 1)THEN
      HRATIO(1)=1.
      FFCSR(1)=(1.0,0.0)
      FFWTR(1)=1.0
      READ(50,*)FFPOIS(1)
      READ(50,*)BETA(1)
      BETA(1)=BETA(1)/100.0
      ELSE
      IF( ILAYER .LE. NLAYER)THEN
      READ(50,*)HRATIO(ILAYER)
      READ(50,*)X
      FFCSR(ILAYER)=X
      READ(50,*)FFWTR(ILAYER)
      READ(50,*)FFPOIS(ILAYER)
      READ(50,*)BETA(ILAYER)
```

```
      BETA(ILAYER)=BETA(ILAYER)/100.0
      ELSE
      READ(50,*)X
      FFCSR(ILAYER)=X
      READ(50,*)FFWTR(ILAYER)
      READ(50,*)FFPOIS(ILAYER)
      READ(50,*)BETA(ILAYER)
      BETA(ILAYER)=BETA(ILAYER)/100.0
      END IF
      END IF
   10 CONTINUE
      READ(50,5020)ATYPE
 5020 FORMAT(A2)
      READ(50,*)NANGLE
      DO IANGLE = 1,NANGLE
      READ(50,*)ANGLE(IANGLE)
      END DO
      CLOSE(UNIT=50)
C
      IF(ATYPE .EQ. 'P')THEN
      AP(NLAYER+1)=(1.0,0.0)
      ASV(NLAYER+1)=(0.0,0.0)
      ELSE
      AP(NLAYER+1)=(0.0,0.0)
      ASV(NLAYER+1)=(1.0,0.0)
      END IF
C
C
      THEATA(NLAYER+1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      XMX(NLAYER+1)=CCOS(THEATA(NLAYER+1))
      IF(NLAYER .NE. 0)THEN
      DO ILAYER=1,NLAYER
      XMX(ILAYER)=XMX(NLAYER+1)*FFCSR(ILAYER)/FFCSR(NLAYER+1)
      END DO
      END IF
      OPEN(UNIT=60,FILE=AOUT, STATUS='UNKNOWN')
      WRITE(60,6000)
 6000 FORMAT(/10X,'PLANE FREE-FIELD CALCULATION')
      WRITE(60,6002)ATITLE
 6002 FORMAT(/5X,A80,/)
      DO 15 ILAYER=1,NLAYER +1
      IF(ILAYER .LE. NLAYER)THEN
      WRITE(60,6005)ILAYER
 6005 FORMAT(/5X,'LAYER: ',I2)
      WRITE(60,6010)HRATIO(ILAYER)
 6010 FORMAT(2X,'Z bottom(i) / Z bottom(1) = ',T35,F7.3)
      X=REAL(FFCSR(ILAYER))
      WRITE(60,6020)X
 6020 FORMAT(2X,'Cs(i)/Cs(1)  = ',T35,F7.3)
      X=REAL(FFWTR(ILAYER))
      WRITE(60,6030)X
 6030 FORMAT(2X,'Wt(i)/Wt(1)  = ',T35,F7.3)
      WRITE(60,6035)FFPOIS(ILAYER)
 6035 FORMAT(2X,'POISSON RATIO = ',T35,F7.3)
      WRITE(60,6040)BETA(ILAYER)*100.
 6040 FORMAT(2X,'DAMPING = ',T35,F7.3,' %')
      ELSE
```

```
      WRITE(60,6050)
 6050 FORMAT(/5X,'HALF-SPACE')
      X=REAL(FFCSR(NLAYER+1))
      WRITE(60,6060)X
 6060 FORMAT(2X,'Cs(half-space)/Cs(1) = ',T35,F7.3)
      X=REAL(FFWTR(NLAYER+1))
      WRITE(60,6070)X
 6070 FORMAT(2X,'Wt(half-space)/Wt(1) = ',T35,F7.3)
      WRITE(60,6035)FFPOIS(NLAYER+1)
      WRITE(60,6040)BETA(NLAYER+1)*100.
      END IF
   15 CONTINUE
      WRITE(60,6075)ATYPE
 6075 FORMAT(/2X,'INCIDENT WAVE TYPE = ',A2)
      WRITE(60,6077)NANGLE
 6077 FORMAT(2X,'NUMBER OF ANGLES OF INCIDENCE = ',I1)
      DO IANGLE=1,NANGLE
      WRITE(60,6078)IANGLE,ANGLE(IANGLE)
 6078 FORMAT(2X,'ANGLE (',I1,') = ',F7.3,' degs')
      END DO
C
      NFREQ=200
      DO 17 ILAYER=1,NLAYER+1
      FFCSR(ILAYER)=FFCSR(ILAYER)*CSQRT((1.0-(0.0,2.0)*BETA(ILAYER))/
     & (1.0-(0.0,2.0)*BETA(1)))
   17 CONTINUE
C
      DO IANGLE =1,NANGLE
      THEATA(NLAYER+1)=(1.0,0.0)*ANGLE(IANGLE)/57.29577951
      XMX(NLAYER+1)=CCOS(THEATA(NLAYER+1))
            IF(NLAYER .NE. 0)THEN
            DO ILAYER=1,NLAYER
            XMX(ILAYER)=XMX(NLAYER+1)*FFCSR(ILAYER)/FFCSR(NLAYER+1)
            END DO
            END IF
C
      WRITE(60,6080)ANGLE(IANGLE)
 6080 FORMAT(//2X,'ANGLE OF INCIDENCE = ',T35,F7.3,' degs')
      WRITE(60,6090)
 6090 FORMAT(/T3,'FREQ',T11,'|U(top)|',T19,'|U(outcrop)|',T32,'U(ampl)'
     &                  ,T46,'|W(top)|',T54,'|W(outcrop)|',T67,'W(ampl)'
     & /)
      WRITE(*,6080)ANGLE(IANGLE)
      WRITE(*,6090)
C
C
      IF(ATYPE .EQ. 'P')THEN
      XLX(NLAYER+1)=CCOS(THEATA(NLAYER+1))
      XMX(NLAYER+1)=XLX(NLAYER+1)*
     & SQRT((1-2.*FFPOIS(NLAYER+1))/
     & (2.*(1.-FFPOIS(NLAYER+1))))
      IF(NLAYER .NE. 0)THEN
      DO 20ILAYER=1,NLAYER
      XLX(ILAYER)=XLX(NLAYER+1)*FFCSR(ILAYER)/FFCSR(NLAYER+1)*
     & SQRT((1.-FFPOIS(ILAYER))*(1.0-(2.*FFPOIS(NLAYER+1)))/
     & ((1.-FFPOIS(NLAYER+1))*(1.-(2.*FFPOIS(ILAYER)))))
      XMX(ILAYER)=XLX(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
```

```
   20 CONTINUE
      END IF
      ELSE
      XMX(NLAYER+1)=CCOS(THEATA(NLAYER+1))
      XLX(NLAYER+1)=XMX(NLAYER+1)*
     & SQRT((2.*(1.-FFPOIS(NLAYER+1)))/
     & (1-2.*FFPOIS(NLAYER+1)))
      IF(NLAYER .NE. 0)THEN
      DO 25ILAYER=1,NLAYER
      XMX(ILAYER)=XMX(NLAYER+1)*FFCSR(ILAYER)/FFCSR(NLAYER+1)
      XLX(ILAYER)=XMX(ILAYER)*SQRT((2.*(1.-FFPOIS(ILAYER+1)))/
     & (1-2.*FFPOIS(ILAYER+1)))
   25 CONTINUE
      END IF
      END IF
      DO 120IFREQ=1,NFREQ
      NDIM=4*NLAYER+2
      RFBAR=0.025*IFREQ
      FBAR=RFBAR/CSQRT(1.0-(0.0,2.0)*BETA(1))
      IF (NLAYER .EQ. 0)GO TO 95
      DO 40I=1,NDIM
      FVECT(I)=(0.0,0.0)
      DO 40J=1,NDIM
      STIFF(I,J)=(0.0,0.0)
   40 CONTINUE
      DIFF=ABS(ANGLE(IANGLE)-90.0)
      IF(DIFF .LT. 0.0001)THEN
C
C     VERTICAL INCIDENCE
C
      STIFF(1,1)=(1.0,0.0)
      STIFF(1,2)=(1.0,0.0)
      STIFF(2,3)=(1.0,0.0)
      STIFF(2,4)=(1.0,0.0)
      DO 50ILAYER=1,NLAYER
      XIKSH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
      XIKSH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*SQRT((1-2.*FFPOIS(ILAYER))/
     & (2.*(1.-FFPOIS(ILAYER))))
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)
      XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)
      I=(ILAYER-1)*4+3
      J=(ILAYER-1)*4+1
      XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
     & * FFWTR(ILAYER+1)/FFWTR(ILAYER)
      CALC1=XI*FFCSR(ILAYER)/FFCSR(ILAYER+1)
      CALC2=SQRT((2.*(1.-FFPOIS(ILAYER)))/(1.-2.*FFPOIS(ILAYER)))
      CALC3=CALC1*SQRT((2.*(1.-FFPOIS(ILAYER+1)))/
     & (1-2.*FFPOIS(ILAYER+1)))
      STIFF(I,J+2)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I,J+3)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+1,J)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+1,J+1)=(+1.0,0.0)*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+2,J)   =CALC2*CEXP((-1.0,0.0)*XIKSH1)
```

```
      STIFF(I+2,J+1)=CALC2*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+3,J+2)=CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+3,J+3)=CEXP((+1.0,0.0)*XIKTH1)
      IF(ILAYER .LT. NLAYER)THEN
      STIFF(I,J+6)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+7)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+4)=(+1.0,0.0)*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+1,J+5)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -CALC3*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+2,J+5)= -CALC3*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+3,J+6)=(-1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+3,J+7)=(-1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I+1,J+4)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -CALC3*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I,J+5)=(-1.0,0.0)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+5)=(-1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER+1)
      FVECT(I+1)=(-1.0,0.0)*CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER+1)
      FVECT(I+2)=(+1.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKSH2)*
     & AP(NLAYER+1)
      FVECT(I+3)=(+1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER+1)
      END IF
   50 CONTINUE
      ELSE
C
C     NON-VERTICAL INCIDENCE
C
      CALC1=CSQRT(1.0-(XLX(1)**2))
      CALC3=CSQRT(1.0-(XMX(1)**2))
      CALC5=(1.-2.*(XMX(1)**2))/XMX(1)
      CALC7=CALC5/XMX(1)
      STIFF(1,1)=XLX(1)*CALC7
      STIFF(1,2)=STIFF(1,1)
      STIFF(1,3)=2.*CALC3
      STIFF(1,4)=-1.*STIFF(1,3)
      STIFF(2,1)= -2.*CALC1
      STIFF(2,2)= 2.*CALC1
      STIFF(2,3)=CALC5
      STIFF(2,4)=CALC5
      DO 60ILAYER=1,NLAYER
      XIKSH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XLX(ILAYER)**2))*
     & SQRT((1-2.*FFPOIS(ILAYER))/(2.*(1.-FFPOIS(ILAYER))))
      XIKSH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XLX(ILAYER+1)**2))*
     & SQRT((1-2.*FFPOIS(ILAYER+1))/(2.*(1.-FFPOIS(ILAYER+1))))
      XIKTH1=(0.0,1.0)*FBAR/FFCSR(ILAYER)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER)**2))
      XIKTH2=(0.0,1.0)*FBAR/FFCSR(ILAYER+1)*
     & HRATIO(ILAYER)*CSQRT(1.0-(XMX(ILAYER+1)**2))
      I=(ILAYER-1)*4+3
      J=(ILAYER-1)*4+1
      XI=(FFCSR(ILAYER+1)/FFCSR(ILAYER))**2
     & * FFWTR(ILAYER+1)/FFWTR(ILAYER)
      CALC1=CSQRT(1.0-(XLX(ILAYER)**2))
      CALC2=CSQRT(1.0-(XLX(ILAYER+1)**2))
```

```
      CALC3=CSQRT(1.0-(XMX(ILAYER)**2))
      CALC4=CSQRT(1.0-(XMX(ILAYER+1)**2))
      CALC5=(1.-2.*(XMX(ILAYER)**2))/XMX(ILAYER)
      CALC6=(1.-2.*(XMX(ILAYER+1)**2))/XMX(ILAYER+1)
      CALC7=CALC5/XMX(ILAYER)
      CALC8=CALC6/XMX(ILAYER+1)
      STIFF(I,J)=(+1.0,0.0)*XLX(ILAYER)*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+1,J)=(-1.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+2,J)=XLX(ILAYER)*CALC7*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I+3,J)=(-2.0,0.0)*CALC1*CEXP((-1.0,0.0)*XIKSH1)
      STIFF(I,J+1)=(+1.0,0.0)*XLX(ILAYER)*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+1,J+1)=(+1.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+2,J+1)=XLX(ILAYER)*CALC7*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I+3,J+1)=(+2.0,0.0)*CALC1*CEXP((+1.0,0.0)*XIKSH1)
      STIFF(I,J+2)=(-1.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+1,J+2)=(-1.0,0.0)*XMX(ILAYER)*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+2,J+2)=(+2.0,0.0)*CALC3*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I+3,J+2)=(+1.0,0.0)*CALC5*CEXP((-1.0,0.0)*XIKTH1)
      STIFF(I,J+3)=(+1.0,0.0)*CALC3*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+1,J+3)=(-1.0,0.0)*XMX(ILAYER)*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+2,J+3)=(-2.0,0.0)*CALC3*CEXP((+1.0,0.0)*XIKTH1)
      STIFF(I+3,J+3)=(+1.0,0.0)*CALC5*CEXP((+1.0,0.0)*XIKTH1)
      IF(ILAYER .LT. NLAYER)THEN
      STIFF(I,J+4)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+1,J+4)=(+1.0,0.0)*CALC2*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -XI*XLX(ILAYER+1)*CALC8*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I+3,J+4)=(+2.0,0.0)*XI*CALC2*CEXP((-1.0,0.0)*XIKSH2)
      STIFF(I,J+5)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+1,J+5)=(-1.0,0.0)*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+5)= -XI*XLX(ILAYER+1)*CALC8*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+3,J+5)=(-2.0,0.0)*XI*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I,J+6)=(+1.0,0.0)*CALC4*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+1,J+6)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+2,J+6)=(-2.0,0.0)*XI*CALC4*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I+3,J+6)=(-1.0,0.0)*XI*CALC6*CEXP((-1.0,0.0)*XIKTH2)
      STIFF(I,J+7)=(-1.0,0.0)*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+7)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+2,J+7)=(+2.0,0.0)*XI*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+7)=(-1.0,0.0)*XI*CALC6*CEXP((+1.0,0.0)*XIKTH2)
      ELSE
      STIFF(I,J+4)=(-1.0,0.0)*XLX(ILAYER+1)*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+1,J+4)=(-1.0,0.0)*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+2,J+4)= -XI*XLX(ILAYER+1)*CALC8*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I+3,J+4)=(-2.0,0.0)*XI*CALC2*CEXP((+1.0,0.0)*XIKSH2)
      STIFF(I,J+5)=(-1.0,0.0)*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+1,J+5)=(+1.0,0.0)*XMX(ILAYER+1)*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+2,J+5)=(+2.0,0.0)*XI*CALC4*CEXP((+1.0,0.0)*XIKTH2)
      STIFF(I+3,J+5)=(-1.0,0.0)*XI*CALC6*CEXP((+1.0,0.0)*XIKTH2)
      FVECT(I)=(+1.0,0.0)*XLX(ILAYER+1)*CEXP((-1.0,0.0)*XIKSH2)
     & *AP(NLAYER+1)-CALC4*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER+1)
      FVECT(I+1)=(-1.0,0.0)*CALC2*CEXP((-1.0,0.0)*XIKSH2)
     & *AP(NLAYER+1)-XMX(ILAYER+1)*CEXP((-1.0,0.0)*XIKTH2)*
     & ASV(NLAYER+1)
      FVECT(I+2)=(+1.0,0.0)*XI*XLX(ILAYER+1)*CALC8*
     & CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER+1)
     & +(2.0,0.0)*XI*CALC4*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER+1)
      FVECT(I+3)=(-2.0,0.0)*XI*CALC2*
```

```
     & CEXP((-1.0,0.0)*XIKSH2)*AP(NLAYER+1)
     & +XI*CALC6*CEXP((-1.0,0.0)*XIKTH2)*ASV(NLAYER+1)
      END IF
   60 CONTINUE
      END IF
      CALL SOLVE
      DO 70ILAYER=1,NLAYER
      IROW=(ILAYER-1)*4+1
      AP(ILAYER)=XVECT(IROW)
      BP(ILAYER)=XVECT(IROW+1)
      ASV(ILAYER)=XVECT(IROW+2)
      BSV(ILAYER)=XVECT(IROW+3)
   70 CONTINUE
      BP(NLAYER+1)=XVECT(NDIM-1)
      BSV(NLAYER+1)=XVECT(NDIM)
      UTOP=
     & XLX(1)*(AP(1)+BP(1))+CSQRT(1.0-(XMX(1)**2))*(-ASV(1)+BSV(1))
      WTOP=
     & SQRT(1.0-(XLX(1)**2))*(-AP(1)+BP(1))+XMX(1)*(-ASV(1)-BSV(1))
C
C
C
      UTMAG=SQRT(REAL(UTOP)**2+AIMAG(UTOP)**2)
      WTMAG=SQRT(REAL(WTOP)**2+AIMAG(WTOP)**2)
C
C     DO 90ILAYER=1,NLAYER+1
C     WRITE(*,*)ILAYER,AP(ILAYER),BP(ILAYER)
C     WRITE(*,*)ILAYER,ASV(ILAYER),BSV(ILAYER)
C   90 CONTINUE
C
C
   95 CONTINUE
      NN=NLAYER+1
C
C
      RCPCS=SQRT((2.*(1.-FFPOIS(NN)))/(1.-2.*FFPOIS(NN)))
      IF(NLAYER .EQ. 0)THEN
      ZHS=0.0
      ELSE
      ZHS=HRATIO(NLAYER)
      END IF
      XIKSZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XLX(NN)**2))*ZHS/RCPCS
C
      XIKTZ=(0.0,1.0)*FBAR/FFCSR(NN)*
     & CSQRT(1.0-(XMX(NN)**2))*ZHS
C
C     HALF-SPACE OUTCROP
      NDIM=2
      APHS=AP(NN)*CEXP(-XIKSZ)
      ASVHS=ASV(NN)*CEXP(-XIKTZ)
      STIFF(1,1)= RCPCS-2.*XMX(NN)*XLX(NN)
      STIFF(1,2)= -2.*XMX(NN)*SQRT(1.-XMX(NN)*XMX(NN))
      STIFF(2,1)= 2.*XMX(NN)*SQRT(1.-XLX(NN)*XLX(NN))
      STIFF(2,2)= 1.0-2.*XMX(NN)*XMX(NN)
      FVECT(1)= (2.*XMX(NN)*XLX(NN)-RCPCS)*APHS
     & -2.*XMX(NN)*SQRT(1.-XMX(NN)*XMX(NN))*ASVHS
      FVECT(2)=2.*XMX(NN)*SQRT(1.-XLX(NN)*XLX(NN))*APHS
```

```
     &   +(2.*XMX(NN)*XMX(NN)-1)*ASVHS
      CALL SOLVE
      BPHS=XVECT(1)
      BSVHS=XVECT(2)
      UOTCRP= XLX(NN)*(APHS+BPHS)+
     &          SQRT(1.-XMX(NN)*XMX(NN))*  (-ASVHS+BSVHS)
      UOTMAG=SQRT(REAL(UOTCRP)**2+AIMAG(UOTCRP)**2)
      WOTCRP= SQRT(1.-XLX(NN)*XLX(NN))*(-APHS+BPHS)+
     &          XMX(NN)*(-ASVHS-BSVHS)
      WOTMAG=SQRT(REAL(WOTCRP)**2+AIMAG(WOTCRP)**2)
C
      IF(NLAYER .EQ. 0)THEN
      UAMPL=1.0
      WAMPL=1.0
      ELSE
      UAMPL=UTMAG/UOTMAG
      WAMPL=WTMAG/WOTMAG
      END IF
      WRITE(60,6100)RFBAR,UTMAG,UOTMAG,UAMPL,WTMAG,WOTMAG,WAMPL
 6100 FORMAT(2X,F6.3,T13,F5.2,T23,F5.2,T33,F5.2,
     &               T48,F5.2,T58,F5.2,T68,F5.2)
C     WRITE(*,6100)RFBAR,UTMAG,UOTMAG,UAMPL,WTMAG,WOTMAG,WAMPL
  120 CONTINUE
      END DO
      CLOSE (UNIT=60)
      END DO
      CLOSE(UNIT=10)
      STOP
      END
```

```
      SUBROUTINE SOLVE
C
C     MATRIX
      COMMON/IMATRIX/NDIM
      COMMON/CMATRIX/AMAT(100,100), FVECT(100),XVECT(100)
      INTEGER NDIM
      COMPLEX AMAT,FVECT,XVECT
C
      DOUBLE PRECISION A(200,201),X(200),PIVOT,TEMP,ANORM
      INTEGER IX(200)
C
C     ****************************************************
C     ****************************************************
C     *****      SUBROUTINE FINDS THE SOLUTION OF     *****
C     *****      SIMULTANEOUS EQUATIONS USING          *****
C     *****      MAXIMUM PIVOT PROCEDURE               *****
C     ****************************************************
C     ****************************************************
C     II REPRESENTS THE REAL PART, II+1 REPRESENTS THE IMAGINARY PART
C     JJ REPRESENTS THE REAL PART, JJ+1 REPRESENTS THE IMAGINARY PART
C
      N=2.*NDIM
      DO 10I=1,NDIM
      II=2 *I-1
      A(II,N+1)= 1.0D0*REAL(FVECT(I))
      A(II+1,N+1)=1.0D0*AIMAG(FVECT(I))
      DO 10J=1,NDIM
      JJ=2*J-1
      A(II,JJ)= 1.0D0*REAL(AMAT(I,J))
      A(II+1,JJ)=1.0D0*AIMAG(AMAT(I,J))
      A(II,JJ+1)= -1.0D0*AIMAG(AMAT(I,J))
      A(II+1,JJ+1)= 1.0D0*REAL(AMAT(I,J))
   10 CONTINUE
C     OPEN(UNIT=20,FILE='MAT.OUT',STATUS='UNKNOWN')
C     WRITE(20,2000)ANAME
 2000 FORMAT(//5X,A80)
C     WRITE(20,2010)
 2010 FORMAT(/10X,'A matrix * X vector = C vector')
C     WRITE(20,2020)
 2020 FORMAT( 5X,'A matrix:')
      DO 20I=1,N
C     WRITE(20,2030)(A(I,J),J=1,N)
 2030 FORMAT(5X,8F10.4)
   20 CONTINUE
C     WRITE(20,2040)
 2040 FORMAT(/5X,'C vector:')
C     WRITE(20,2030)(A(J,N+1),J=1,N)
C
C     *****************************************************
C     *****************************************************
C     *SOLVE FOR X VECTOR USING GAUSS'S ELIMINATION METHOD *
C     ************   MAXIMUM PIVOT IS USED ****************
C     *****************************************************
C     *****************************************************
C
C
C
      DO 30I=1,N
```

```
      IX(I)=I
   30 CONTINUE
      DO 100IROW=1,N
      I=IROW/10
      XX=IROW
      XX=XX/10.
      DIFF=ABS(XX-I)
C     IF(DIFF .LT. 0.0001)THEN
C     WRITE(*,9000)IROW,N
C 9000 FORMAT(5X,'WORKING ON COLUMN # ',I4,' / ',I4)
C     END IF
      JCOL=IROW
      PIVOT=0.0D0
C     FIND PIVOT VALUE
      DO 40IIROW=IROW,N
      DO 40JJCOL=JCOL,N
      IF (ABS(A(IIROW,JJCOL)) .GT. ABS (PIVOT)) THEN
      PIVOT =A(IIROW,JJCOL)
      IMARK=IIROW
      JMARK=JJCOL
      END IF
   40 CONTINUE
      IF (ABS(PIVOT) .LT. 1.0D-18) THEN
      WRITE(*,9010)
 9010 FORMAT(////15X,'** MATRIX DOES NOT HAVE AN INVERSE,')
      WRITE(*,9020)
 9020 FORMAT(15X,'PUSH RETURN  **')
      PAUSE
      STOP
      END IF
C     INTERCHANGE ROWS
      DO 50 JJCOL=1,N+1
      TEMP=A(IROW,JJCOL)
      A(IROW,JJCOL)=A(IMARK,JJCOL)
      A(IMARK,JJCOL)=TEMP
   50 CONTINUE
C     INTERCHANGE COLUMNS
      ITEMP=IX(JCOL)
      IX(JCOL)=IX(JMARK)
      IX(JMARK)=ITEMP
      DO 60 IICOL=1,N
      TEMP=A(IICOL,JCOL)
      A(IICOL,JCOL)=A(IICOL,JMARK)
      A(IICOL,JMARK)=TEMP
   60 CONTINUE
C     WRITE(20,*)'IROW = ',IROW,'†PIVOT = ',PIVOT
C     DO 65 I=1,N
C     WRITE(20,2050)(A(I,J),J=1,N+1)
C   65 CONTINUE
C     NORMALIZE ROW OF PIVOT ELEMENT
      DO 70 JJCOL=JCOL,N+1
      A(IROW,JJCOL)=A(IROW,JJCOL)/PIVOT
   70 CONTINUE
C     SUBTRACT ROW FROM REMAINING ROWS
      DO 80 IIROW=IROW+1,N
      ANORM=-1.0D0*A(IIROW,JCOL)
      DO 80 JJCOL=JCOL,N+1
      A(IIROW,JJCOL)=A(IIROW,JJCOL)+ANORM *A(IROW,JJCOL)
```

```
   80 CONTINUE
C       WRITE(20,*)'IROW = ',IROW,'  PIVOT = ',PIVOT
C       DO 85 I=1,N
C       WRITE(20,2050)(A(I,J),J=1,N+1)
C 2050 FORMAT(5F10.3)
C    85 CONTINUE
  100 CONTINUE
C       USE BACKWARD SUBSTITUTION
        X(N)=A(N,N+1)/A(N,N)
        DO 120IROW=N-1,1,-1
        SUM=0.0
        DO 110 JCOL=IROW+1,N
        SUM=SUM+A(IROW,JCOL)*X(JCOL)
  110 CONTINUE
        X(IROW)=(A(IROW,N+1)-SUM)/A(IROW,IROW)
  120 CONTINUE
C       REORDER X VECTOR
        DO 130 IROW=1,N
        A(IROW,N+1)=X(IROW)
  130 CONTINUE
        DO 140 I=1,N
        IROW=IX(I)
        X(IROW)=A(I,N+1)
  140 CONTINUE
C
C       WRITE(20,2060)
 2060 FORMAT(//5X,'X vector:')
C        WRITE(20,2070)(X(J),J=1,N)
 2070 FORMAT(5X,8F10.4)
C       CLOSE (UNIT=20)
        DO 150J=1,NDIM
        JJ=2*J-1
        XVECT(J)=X(JJ)+(0.0,1.0)*X(JJ+1)
  150 CONTINUE
        RETURN
        END
```

A.3: Sort Codes

Sort codes were developed to reformat the information from the output files to a column format appropriate for the input to Cricket Graph (Computer Associates). The Cricket Graph program was used to generate the plots included in this thesis. The code "bemsort" is used to reformat the output files from the boundary element codes to give in column format the node coordinates and their corresponding surface amplifications The code "ffldsort" is used to reformat the output files from the one-dimensional programs and to give in column format the dimensionless frequencies and surface amplifications. Both of the sort programs use a run-file which includes:

Line 1	Number of problems
Line 2	Input file name using a maximum of 20 characters.
Line 3	Output file name which is the name of the sorted input file. A maximum of 20 characters can be used for this name.

Lines 2 and 3 are repeated for the number of problems above 2 indicated on line 1 .

A.3.1: SORT CODE

bemsort

```
      PROGRAM BEMSORT
      DIMENSION X(500),Z(500),DISPL(500),ANGLE(5),XFFLD(2)
      CHARACTER*80 ANAME
      CHARACTER*20 AINPT,ASORT,ARUN
      CHARACTER*1 ADUM
      CHARACTER*6 ASTART
      CHARACTER*2 ATYPE
C
      WRITE(*,9000)
 9000 FORMAT(2X,'ENTER BEM SORT RUN FILE NAME')
      READ(*,9010)ARUN
      OPEN(UNIT=10,FILE=ARUN,STATUS='UNKNOWN')
      READ(10,*)NPROB
C
      DO IPROB=1,NPROB

C     WRITE(*,9005)
C 9005 FORMAT(2X,'ENTER BEM OUTPUT FILE NAME')
      READ(10,9010)AINPT
 9010 FORMAT(A20)
C     WRITE(*,9020)
C 9020 FORMAT(2X,'ENTER BEM SORT FILE NAME')
      READ(10,9010)ASORT
      WRITE(*,9020)AINPT
 9020 FORMAT(/2X,'SORTING: ',A20)
      OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
      OPEN(UNIT=60,FILE=ASORT,STATUS='UNKNOWN')
      READ(50,5000)NANGLE
 5000 FORMAT(/////,T43,I1)
      DO IANGLE =1,NANGLE
      READ(50,5002)ANGLE(IANGLE)
 5002 FORMAT(T39,F6.0)
      END DO
C
      DO 9999 IANGLE =1,NANGLE
      WRITE(*,*)'IANGLE= ',IANGLE,' OUT OF ',NANGLE
C
C     FIND WHERE OUTPUT SECTION BEGINS
   10 CONTINUE
      READ(50,5005)ASTART
 5005 FORMAT(5X,A6)
      IF(ASTART .EQ. 'OUTPUT')GO TO 20
      GO TO 10
   20 CONTINUE
      READ(50,5010)ANAME
 5010 FORMAT(/,A80)
      IF(IANGLE .EQ. 1)WRITE(60,6000)ANAME
      IF(IANGLE .GT. 1)WRITE(60,6005)ANAME
 6000 FORMAT(5X,A80)
 6005 FORMAT(/5X,A80)
      READ(50,5020)ATYPE
 5020 FORMAT(T21,A2)
      WRITE(60,6010)ATYPE
 6010 FORMAT(5X,'TYPE OF WAVE = ',A2)
      READ(50,5030)ANAME
 5030 FORMAT(A80)
      WRITE(60,6000)ANAME
      READ(50,5030)ANAME
```

```
      WRITE(60,6000)ANAME
      READ(50,5040)NAREA
 5040 FORMAT(T38,I1)
      WRITE(60,6020)NAREA
 6020 FORMAT(5X,'NUMBER OF HOMOGENEOUS REGIONS = 'I1)
      WRITE(60,6030)
 6030 FORMAT(/5X,'BOUNDARY VALUES')
      READ(50,5070)ADUM
      READ(50,5070)ADUM
      DO IAREA =1,NAREA
      IF(IAREA .EQ. 1)THEN
      WRITE(60,6040)IAREA
 6040 FORMAT(/5X,'HOMOGENEOUS REGION : ',I1)
      END IF
C
C
C     FIND WHERE HOMOGENEOUS REGION BEGINS
   30 CONTINUE
      READ(50,5005)ASTART
      IF(ASTART .EQ. 'HOMOGE')GO TO 40
      GO TO 30
   40 CONTINUE
      READ(50,5050)NBOUND
 5050 FORMAT(T29,I1)
      DO IBOUND=1,NBOUND
C     FIND WHERE BOUNDARY BEGINS
   50 CONTINUE
      READ(50,5005)ASTART
      IF(ASTART .EQ. 'BOUNDA')GO TO 60
      GO TO 50
   60 CONTINUE
C
      READ(50,5060)NNODE
 5060 FORMAT(T24,I3)
      IF (IAREA .EQ. 1 .AND. IBOUND .EQ. 1)THEN
      WRITE(60,6050)IBOUND
 6050 FORMAT(/2X,'BOUNDARY #',I1)
      WRITE(60,6060)NNODE
 6060 FORMAT(2X,'NUMBER OF NODES = ',I3)
      END IF
            DO I=1,4
      READ(50,5070)ADUM
 5070 FORMAT(A1)
            END DO
C     READ DISPLACEMENTS
C
      DO INODE=1,NNODE
      READ(50,5080)X(INODE),Z(INODE),DISPL(INODE)
 5080 FORMAT(T11,F8.3,T21,F8.3,T61,F8.3)
      END DO
      IF (IAREA .EQ. 1 .AND. IBOUND .EQ. 1)THEN
C
      CALL FFLDLGTH(X,DISPL,NNODE,XFFLD)
C
      IF (IANGLE .EQ. 1)THEN
      WRITE(60,6070)
 6070 FORMAT(/2X,'X COORDINATE')
      DO INODE =1,NNODE
```

```
      WRITE(60,6080)X(INODE)
 6080 FORMAT(F8.3)
      END DO
      WRITE(60,6090)
 6090 FORMAT(/2X,'Z COORDINATE')
      DO INODE =1,NNODE
      WRITE(60,6100)Z(INODE)
 6100 FORMAT(F8.3)
      END DO
      END IF
C
      IF(ATYPE .EQ. 'P' .OR. ATYPE .EQ. 'SV')
     & WRITE(60,6110)ANGLE(IANGLE)
      IF(ATYPE .EQ. 'SH')WRITE(60,6120)ANGLE(IANGLE)
 6110 FORMAT(/2X,'|U|  INCIDENT ANGLE = ',F6.2,' degs')
 6120 FORMAT(/2X,'|V|  INCIDENT ANGLE = ',F6.2,' degs')
      DO INODE =1,NNODE
      WRITE(60,6130)DISPL(INODE)
 6130 FORMAT(F8.3)
      END DO
C
      WRITE(60,6135)XFFLD(1),XFFLD(2)
 6135 FORMAT(/5X,'DISTANCE TO FREE-FIELD CALC: ',
     &       /5X,'(-X SIDE) = ',F7.3,' X/H1',
     &       /5X,'(+X SIDE) = ',F7.3,' X/H1'//)
C
      END IF
C
      IF(ATYPE .EQ. 'P' .OR. ATYPE .EQ. 'SV')THEN
            DO I=1,2
      READ(50,5070)ADUM
            END DO

C     READ DISPLACEMENTS
      DO INODE=1,NNODE
      READ(50,5080)X(INODE),Z(INODE),DISPL(INODE)
      END DO

      IF (IAREA .EQ. 1 .AND. IBOUND .EQ. 1)THEN
C
      CALL FFLDLGTH(X,DISPL,NNODE,XFFLD)
C
      WRITE(60,6140)ANGLE(IANGLE)
 6140 FORMAT(/2X,'|W|  INCIDENT ANGLE = ',F6.2,' degs')
      DO INODE =1,NNODE
      WRITE(60,6080)DISPL(INODE)
      END DO
C
      WRITE(60,6135)XFFLD(1),XFFLD(2)

C
      END IF
      END IF
C
C
      END DO
      END DO
 9999 CONTINUE
```

```
      CLOSE(UNIT=50)
      CLOSE(UNIT=60)
      END DO
C
      STOP
      END


      SUBROUTINE FFLDLGTH(X,DISPL,NNODE,XFFLD)
C
      DIMENSION X(500),DISPL(500),XFFLD(2)
C
C
      IF(ABS(DISPL(1) ) .LT. 1.0)THEN
      XHIGH = DISPL(1) +0.1
      XLOW = DISPL(1) -0.1
      ELSE
      XHIGH = 1.10*DISPL(1)
      XLOW =  0.90*DISPL(1)
      END IF
C
      IF (XLOW .LT. 0.0)XLOW = 0.0
      INODE =1
   10 CONTINUE
      INODE =INODE+1
      MARK =0
      IF (DISPL(INODE) .GT. XHIGH)THEN
      MARK = 1
      GO TO 20
      END IF
      IF (DISPL(INODE) .LT. XLOW)THEN
      MARK =2
      GO TO 20
      END IF
      IF (ABS(X(INODE)) .GT. 0.001)GO TO 10
   20 CONTINUE
      IF (MARK .EQ. 0)THEN
      XFFLD(2)=0.0
      ELSE
      IF (MARK .EQ. 1)THEN
      XFFLD(2)=     (XHIGH-DISPL(INODE-1))*(X(INODE)-X(INODE-1))
     &  /(DISPL(INODE)-DISPL(INODE-1))  +X(INODE-1)
      END IF
      IF (MARK .EQ. 2)THEN
      XFFLD(2)=     (XLOW-DISPL(INODE-1))*(X(INODE)-X(INODE-1))
     &  /(DISPL(INODE)-DISPL(INODE-1))  +X(INODE-1)
      END IF
      END IF
C
      IF(ABS(DISPL(NNODE) ) .LT. 1.0)THEN
      XHIGH = DISPL(NNODE) +0.1
      XLOW = DISPL(NNODE) -0.1
      ELSE
      XHIGH = 1.10*DISPL(NNODE)
      XLOW =  0.90*DISPL(NNODE)
      END IF
C
```

```
      IF (XLOW .LT. 0.0)XLOW = 0.0
C
      INODE =NNODE
   30 CONTINUE
      INODE =INODE-1
      MARK =0
      IF (DISPL(INODE) .GT. XHIGH)THEN
      MARK = 1
      GO TO 40
      END IF
      IF (DISPL(INODE) .LT. XLOW)THEN
      MARK =2
      GO TO 40
      END IF
      IF (ABS(X(INODE)) .GT. 0.001)GO TO 30
   40 CONTINUE
      IF (MARK .EQ. 0)THEN
      XFFLD(1)=0.0
      ELSE
      IF (MARK .EQ. 1)THEN
      XFFLD(1)=     (XHIGH-DISPL(INODE+1))*(X(INODE)-X(INODE+1))
     &  /(DISPL(INODE)-DISPL(INODE+1))  +X(INODE+1)
      END IF
      IF (MARK .EQ. 2)THEN
      XFFLD(1)=     (XLOW-DISPL(INODE+1))*(X(INODE)-X(INODE+1))
     &  /(DISPL(INODE)-DISPL(INODE+1))  +X(INODE+1)
      END IF
      END IF

      RETURN
      END
```

A.3.2: SORT CODE

ffldsort

```
      PROGRAM FFLDSORT
      CHARACTER*20 AINPT, ASORT,ARUN
      CHARACTER*4 ASTART
      CHARACTER*1 ADUM
      CHARACTER*2 ATYPE
      CHARACTER*80 ATITLE
      DIMENSION X(1000,5),ANGLE(5)
C
      WRITE(*,9000)
 9000 FORMAT(2X,'ENTER FREE-FIELD RUN SORT FILE')
      READ(*,9010)ARUN
      OPEN(UNIT=10,FILE=ARUN,STATUS='UNKNOWN')
      READ(10,*)NPROB
      DO IPROB=1,NPROB
C     WRITE(*,9000)
C 9000 FORMAT(2X,'ENTER FREE-FIELD FILE TO SORTED')
      READ(10,9010)AINPT
 9010 FORMAT(A20)
C     WRITE(*,9020)
C 9020 FORMAT(2X,'ENTER FILE NAME OF SORTED FILE')
      READ(10,9010)ASORT
      OPEN(UNIT=50,FILE=AINPT,STATUS='UNKNOWN')
      OPEN(UNIT=60,FILE=ASORT,STATUS='UNKNOWN')
      READ(50,5000)ATITLE
 5000 FORMAT(/A80)
      WRITE(60,5000)ATITLE
      READ(50,5000)ATITLE
      WRITE(60,5000)ATITLE
   10 CONTINUE
      READ(50,5010)ASTART
 5010 FORMAT(T6,A4)
      IF(ASTART .EQ. 'HALF')GO TO 20
      GO TO 10
   20 CONTINUE
   30 CONTINUE
      READ(50,5020)ASTART
 5020 FORMAT(T3,A4)
      IF(ASTART .EQ. 'DAMP')GO TO 40
      GO TO 30
   40 CONTINUE
      READ(50,5030)ATYPE
 5030 FORMAT(/T24,A2)
      READ(50,5040)NANGLE
 5040 FORMAT(T35,I1)
      DO IANGLE=1,NANGLE
      READ(50,5050)ANGLE(IANGLE)
 5050 FORMAT(T15,F7.3)
      END DO
C
            DO IANGLE =1,NANGLE
C
      READ(50,5060)ATITLE
 5060 FORMAT(//A80///)
      NFREQ=200
      IF (ATYPE .EQ. 'P' .OR. ATYPE .EQ. 'SV')THEN
      DO IFREQ=1,NFREQ
      READ(50,5070)(X(IFREQ,I),I=1,5)
 5070 FORMAT(T3,F6.0,T13,F5.2,T33,F5.2,T48,F5.2,T68,F5.2)
```

```
      END DO
      DO I=1,5
      WRITE(60,6010)ATITLE
 6010 FORMAT(//A80)
      IF(I .EQ. 1)WRITE(60,6020)
      IF(I .EQ. 2)WRITE(60,6030)
      IF(I .EQ. 3)WRITE(60,6040)
      IF(I .EQ. 4)WRITE(60,6050)
      IF(I .EQ. 5)WRITE(60,6060)
 6020 FORMAT(/'FREQ (cps)'/)
 6030 FORMAT(/'|U(top)|'/)
 6040 FORMAT(/'|U(ampl)|'/)
 6050 FORMAT(/'|W(top)|'/)
 6060 FORMAT(/'|W(ampl)|'/)
      DO IFREQ=1,NFREQ
      WRITE(60,6070)X(IFREQ,I)
 6070 FORMAT(F6.3)
      END DO
      END DO
      END IF
      IF (ATYPE .EQ. 'SH' )THEN
      DO IFREQ=1,NFREQ
      READ(50,5080)(X(IFREQ,I),I=1,3)
 5080 FORMAT(T3,F6.0,T32,F7.2,T77,F7.2)
      END DO
      DO I=1,3
      WRITE(60,6010)ATITLE
      IF(I .EQ. 1)WRITE(60,6020)
      IF(I .EQ. 2)WRITE(60,6080)
      IF(I .EQ. 3)WRITE(60,6090)
 6080 FORMAT(/'|V(top)|'/)
 6090 FORMAT(/'|V(ampl)|'/)
      DO IFREQ=1,NFREQ
      WRITE(60,6070)X(IFREQ,I)
      END DO
      END DO
      END IF
C
            END DO
      CLOSE(UNIT=50)
      CLOSE(UNIT=60)
      END DO
      CLOSE(UNIT=10)
      STOP
      END
```

A.4: Half-Space Angle of Incidence Codes:

Two codes, "svhsinc" and "phsinc", were developed to determine the incident and reflected angles in various soil layers from a given incident wave type and half-space incidence angle. The code "svhsinc" is used for an incident shear wave and "phsinc" is used for an incident compression wave. The codes are developed based on the wave number being constant for each of the soil layers and half-space. Input to these codes is interactive.

A.4.1: ANGLE OF INCIDENCE CODE

svhsinc

```
      PROGRAM SVHSINC
      COMPLEX XMXJ,TJ,XLXJ,SJ
      WRITE(*,9000)
 9000 FORMAT(2X,'INPUT SV WAVE HALF-SPACE ANGLE OF INCIDENCE (DEGREES)')
      READ(*,*)HSANG
      IF(HSANG .GE. 89.999 .AND. HSANG .LE. 90.0001)HSANG= 89.9999
      HSANG=HSANG/57.29578
      WRITE(*,9010)
 9010 FORMAT(2X,'INPUT SHEAR VELOCITY RATIO')
      READ(*,*)CSRAT
      WRITE(*,9020)
 9020 FORMAT(2X,'INPUT DAMPING (%)')
      READ(*,*)BETA
      BETA=BETA/100.0
      WRITE(*,9025)
 9025 FORMAT(2X,'INPUT POISSON RATIO')
      READ(*,*)POISS
C
      RCPCS=SQRT(2.0*(1.0-POISS)/(1.0-2.0*POISS))
      XMXJ=CSRAT/10.0*CSQRT(1.0-(0.0,1.0)*2.0*BETA) *COS(HSANG)
      TJ=CSQRT(1.0/(XMXJ*XMXJ) -1.0)
      REALTJ=REAL(TJ)
      THEATA=ATAN(REALTJ)
      THEATA=THEATA*57.29578
      WRITE(*,9030)THEATA
 9030 FORMAT
     & (2X,'ANGLE OF INCOMING SV WAVE = ',F9.3,' degrees')
      XLXJ=CSRAT*RCPCS/10.0*CSQRT(1.0-(0.0,1.0)*2.0*BETA) *COS(HSANG)
      SJ=CSQRT(1.0/(XLXJ*XLXJ) -1.0)
      REALSJ=REAL(SJ)
      THEATA=ATAN(REALSJ)
      THEATA=THEATA*57.29578
      WRITE(*,9040)THEATA
 9040 FORMAT
     & (2X,'ANGLE OF INCOMING P WAVE  = ',F9.3 ,' degrees')
      PAUSE
      STOP
      END
```

A.4.2: ANGLE OF INCIDENCE CODE

phsinc

```
      PROGRAM PHSINC
      COMPLEX XMXJ,TJ,XLXJ,SJ
      WRITE(*,9000)
 9000 FORMAT(2X,'INPUT P WAVE HALF-SPACE ANGLE OF INCIDENCE (DEGREES)')
      READ(*,*)HSANG
      IF(HSANG .GE. 89.999 .AND. HSANG .LE. 90.0001)HSANG= 89.9999
      HSANG=HSANG/57.29578
      WRITE(*,9010)
 9010 FORMAT(2X,'INPUT SHEAR VELOCITY RATIO')
      READ(*,*)CSRAT
      WRITE(*,9020)
 9020 FORMAT(2X,'INPUT DAMPING (%)')
      READ(*,*)BETA
      BETA=BETA/100.0
      WRITE(*,9025)
 9025 FORMAT(2X,'INPUT POISSON RATIO')
      READ(*,*)POISS
C
      RCPCS=SQRT(2.0*(1.0-POISS)/(1.0-2.0*POISS))
      XLXJ=CSRAT/10.0*CSQRT(1.0-(0.0,1.0)*2.0*BETA) *COS(HSANG)
      SJ=CSQRT(1.0/(XLXJ*XLXJ) -1.0)
      REALSJ=REAL(SJ)
      THEATA=ATAN(REALSJ)
      THEATA=THEATA*57.29578
      WRITE(*,9030)THEATA
 9030 FORMAT
     & (2X,'ANGLE OF INCOMING P WAVE  = ',F9.3,' degrees')
      XMXJ=CSRAT/(RCPCS*10.0)*CSQRT(1.0-(0.0,1.0)*2.0*BETA)*COS(HSANG)
      TJ=CSQRT(1.0/(XMXJ*XMXJ) -1.0)
      REALTJ=REAL(TJ)
      THEATA=ATAN(REALTJ)
      THEATA=THEATA*57.29578
      WRITE(*,9040)THEATA
 9040 FORMAT
     & (2X,'ANGLE OF INCOMING SV WAVE  = ',F9.3 ,' degrees')
      PAUSE
      STOP
      END
```

A.5: Analytic Solution for a Semi-Cylindrical Cavity in a Half-Space

A code was developed to calculate the surface amplification due to a semi-cylindrical cavity in a half-space subject to an incident SH wave. Incidence angles of 90.0, 75.0, 60.0, and 30.0 degrees were used. The formulation of the code comes from the analytic solution (Trifunac, 1973). Output including node coordinates and surface amplification are given along the horizontal boundary and semi-cylindrical surface. The output is given in two formats for each of the angles of incidence. The first format is in table form and allows for easy reading. The second format is column form and allows for easy input to the Cricket Graph program for reprocessing. The program is interactive and asks for the output file name and the dimensionless frequency to analyze for. The dimensionless frequency uses the radius of the cylinder for the characteristic length.

A.5.1: SEMI-CYLINDRICAL CAVITY IN A HALF-SPACE CODE

sccanyon

```
      PROGRAM SCCANYON
C234567891123456789212345678931234567894123456789512345678961234567897l
2
      DOUBLE PRECISION JN0,JN(20),YN,JN2N,JN2N1
      COMPLEX HNK02,HNK2(20),A0,B0,FFLD,DISPL(60),
     & HNK2N,HNK2N1
      COMPLEX*16 AN(20),BN(20)
      DIMENSION ANGINC(4),THEATA(60),X(60),Y(60)
      CHARACTER*20 AOUT
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DATA NGAUSS /5/
      DATA (WT(I),I=1,3)/0.236926885056189D0,0.478628670499366D0,
     & 0.568888888888889D0/
      DATA (XX(I),I=1,3)/-0.906179845938664D0,-0.538469310105683D0,
     & 0.0D0/
C
      XX(4)= -1.0D0*XX(2)
      XX(5)= -1.0D0*XX(1)
C
            DO IGAUSS= 1,2
      WTFN(IGAUSS)= (1.0D0 - XX(IGAUSS))/2.0D0
      WTFN(NGAUSS-IGAUSS+1)= 1.0D0-WTFN(IGAUSS)
            END DO
      WTFN(3)= 0.50D0
C
      PI=3.141593
      NANGLE=4
      NSTEP=58
      ANGINC(1)=90.0
      ANGINC(2)=75.0
      ANGINC(3)=60.0
      ANGINC(4)=30.0
C
C
      WRITE(*,9000)
 9000 FORMAT(2X,'ENTER DIMENSIONLESS FREQUENCY')
      READ(*,*)FBAR
      WRITE(*,9010)
 9010 FORMAT(/2X,'ENTER OUTPUT FILE NAME')
      READ(*,9020)AOUT
 9020 FORMAT(A20)
      CALL JNU(0,FBAR,JN0)
      CALL YNU(0,FBAR,YN)
      HNK02=JN0-(0.0,1.0)*YN
            DO IORDER =1,15
      WRITE(*,*)'IORDER = ',IORDER
      CALL JNU(IORDER,FBAR,JN(IORDER))
      CALL YNU(IORDER,FBAR,YN)
      HNK2(IORDER)=JN(IORDER)-(0.0,1.0)*YN
            END DO
C
      OPEN(UNIT=60,FILE=AOUT,STATUS= 'UNKNOWN')
      WRITE(60,6000)
```

```
 6000 FORMAT(5X,'SEMICIRCULAR CANYON DISPLACEMENTS DUE TO SH WAVE')
      WRITE(60,6010)FBAR
 6010 FORMAT(/2X,'OMEGA*H1/(Cs) = ',F5.3)
C
                              DO IANGLE = 1,NANGLE
      WRITE(*,*)'WORKING ON ANGLE = ',IANGLE
      PHI=90.0-ANGINC(IANGLE)
      PHI=PHI/57.29577951
      A0=-2.0*JN(1)/HNK2(1)
      B0=(0.0,4.0)*SIN(PHI)*(FBAR*JN0-JN(1))/(FBAR*HNK02-HNK2(1))
            DO N=1,7
      AN(N)=-4.0*(-1.0**N)*COS(2.0*N*PHI)*
     & (FBAR*JN(2*N-1)-2.0*N*JN(2*N))/
     & (FBAR*HNK2(2*N-1)-2.0*N*HNK2(2*N))
      BN(N)=(0.0,4.0)*
     &  (-1.0**N)*SIN((2.0*N*+1)*PHI)*
     & (FBAR*JN(2*N)-(2.0*N+1)*JN(2*N+1))/
     & (FBAR*HNK2(2*N)-(2.0*N+1)*HNK2(2*N+1))
C     WRITE(*,*)'N,AN,BN',N,AN(N),BN(N)
            END DO
                  DO I=1,NSTEP+1
      IF(I .LT. 21)THEN
      THEATA(I)=0.0
      X(I)=3.0-(I-1)*0.1
      Y(I)=0.0
      END IF
      IF(I .GE. 21 .AND. I .LE. 39)THEN
      THEATA(I)=(I-21)*PI/(18)
      X(I)=COS(THEATA(I))
      Y(I)=SIN(THEATA(I))
      END IF
      IF(I .GT. 39)THEN
      THEATA(I)=PI
      X(I)=-1.0-(I-39)*0.1
      Y(I)=0.0
      END IF
      XKX=FBAR*SIN(PHI)
      XKY=FBAR*COS(PHI)

C
      FFLD=2.0*CEXP((0.0,-1.0)*XKX*X(I))*COS(XKY*Y(I))
C     WRITE(*,*)'X,Y,FFLD',X(I),Y(I),FFLD
      PSI=(PI/2.0) -THEATA(I)
C
      IF(I .LT. 21 .OR. I .GT. 39)THEN
      XKR=FBAR*ABS(X(I))
      CALL JNU(0,XKR,JN2N)
      CALL YNU(0,XKR,YN)
      HNK2N=JN2N-(0.0,1.0)*YN
      CALL JNU(1,XKR,JN2N1)
      CALL YNU(1,XKR,YN)
      HNK2N1=JN2N1-(0.0,1.0)*YN
      DISPL(I)=FFLD+A0*HNK2N+B0*HNK2N1*SIN(PSI)
            DO N=1,7
      CALL JNU(2*N,XKR,JN2N)
      CALL YNU(2*N,XKR,YN)
      HNK2N=JN2N-(0.0,1.0)*YN
      CALL JNU(2*N+1,XKR,JN2N1)
```

```
      CALL YNU(2*N+1,XKR,YN)
      HNK2N1=JN2N1-(0.0,1.0)*YN
      DISPL(I)=DISPL(I)+
     & AN(N)*HNK2N*COS(2*N*PSI)+BN(N)*HNK2N1*SIN((2*N+1)*PSI)
            END DO
C
      ELSE
C
      DISPL(I)=
     & FFLD+A0*HNK02+B0*HNK2(1)*SIN(PSI)
      DISPL(I)=DISPL(I)
            DO N=1,7
      DISPL(I)=DISPL(I)+
     & AN(N)*HNK2(2*N)*COS(2*N*PSI)+BN(N)*HNK2(2*N+1)*SIN((2*N+1)*PSI)
      DISPL(I)=DISPL(I)
C     WRITE(*,*)'DISPL(I)',DISPL(I)
            END DO
      END IF
C
                  END DO
      WRITE(60,6050)ANGINC(IANGLE)
 6050 FORMAT(/5X,'ANGLE OF INCIDENCE = ',F5.2,' degrees'/)
      WRITE(60,6055)
 6055 FORMAT(T2,'THEATA',T14,'X/a',T24,'Y/a',
     & T43,'DISPL',T56,'MAGNITUDE'/)
            DO I=1,NSTEP+1
      THEATA(I)=THEATA(I)*57.29578
      XREAL=REAL(DISPL(I))
      XIMAG=AIMAG(DISPL(I))
      XMAG=SQRT(XREAL*XREAL + XIMAG*XIMAG)

      WRITE(60,6060)THEATA(I),X(I),Y(I),DISPL(I),XMAG
 6060 FORMAT(T2,F6.2,T11,F8.4,T21,F8.4,T37,2F8.4,T56,F8.4)
            END DO
C
      IF(IANGLE .EQ. 1)THEN
            DO ISTEP=1,NSTEP+1
      WRITE(60,6070)X(ISTEP)
 6070 FORMAT(F10.3)
            END DO
      END IF
C

      WRITE(60,6080)
 6080 FORMAT(/5X,'MAGNITUDE'/)
            DO I = 1,NSTEP+1
      XREAL=REAL(DISPL(I))
      XIMAG=AIMAG(DISPL(I))
      XMAG=SQRT(XREAL*XREAL + XIMAG*XIMAG)
      WRITE(60,6090)XMAG
 6090 FORMAT(F10.3)
            END DO
                  END DO

      CLOSE(UNIT=60)
      STOP
      END
```

```
      SUBROUTINE JNU(IORDER,XREAL,JN)
C     *******************************************
C     BESSEL FUNCTION (Jn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     *******************************************
C23456789112345678921234567893123456789412345678951234567896123456789712
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,AA,A,B
      DOUBLE PRECISION JN
C
      Z=XREAL
      PI=3.1415926535D0
      TOLER=.00001D0
C     CALCULATE BESSEL FUNCTION OF 1ST KIND (JN)
      A=0.0D0
      B=PI
      HDELTA=B-A
      AOLD=9.99999D20
   10 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 20I=1,N
      AA= (I-1)*HDELTA+A
      DO 15J=1,NGAUSS
      X(J)=    AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DCOS(XREAL* DSIN(X(J))-IORDER*X(J))
   15 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   20 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 10
      END IF
   25 CONTINUE
      JN= ATOT/PI
C     WRITE(*,*)'JN = ',JN
      RETURN
      END


      SUBROUTINE YNU(IORDER,XREAL,YN)
C     *******************************************
C     BESSEL FUNCTION (Yn) OF A COMPLEX ARGUMENT
C     INTERGRATION USING 5 POINT GAUSSIAN QUADRATURE
C     *******************************************
C23456789112345678921234567893123456789412345678951234567896123456789712
```

```
C
      COMMON/IGAUS5/NGAUSS
      COMMON/DGAUS5/WT(3),WTFN(5),XX(5)
      INTEGER NGAUSS
      DOUBLE PRECISION WT,WTFN,XX
C
      DOUBLE PRECISION TOLER,HDELTA,AOLD,
     & X(5),Y(5),ATOT,DIFF,ADELTA,PI,CALC,AA,A,B
      DOUBLE PRECISION YN
C
C     CALCULATE BESSEL FUNCTION OF 2ND KIND, ORDER N (YN)
      PI=3.1415926535D0
      TOLER=0.00001
      A=0.
      B=PI
      HDELTA=B-A
      AOLD=9.99999D20
   30 CONTINUE
      N=(B-A)/HDELTA
      ATOT=0.0
      DO 50I=1,N
      AA= (I-1)*HDELTA+A
      DO 40J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      Y(J)= DSIN(XREAL* DSIN(X(J))-IORDER*X(J))
   40 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
      ATOT=ATOT+ADELTA
   50 CONTINUE
      DIFF=ABS(ATOT-AOLD)
      IF (DIFF .GT. TOLER) THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 30
      END IF
   55 CONTINUE
      YN=ATOT
      A=0.
      B=10.
      HDELTA=1.0
C     HDELTA=B-A
      AOLD=9.99999D20
   60 CONTINUE
      ATOT=0.0
      N=(B-A)/HDELTA
      DO 80I=1,N
      AA= (I-1)*HDELTA+A
      DO 70J=1,NGAUSS
      X(J)= AA +(XX(J)+1.0D0)*HDELTA/2.0D0
      CALC=XREAL*DSINH(X(J))
      IF(CALC .GT. 40.)CALC=40.
      Y(J)=
     &(DEXP(IORDER*X(J))+DEXP((-1.)*IORDER*X(J))*DCOS(IORDER*PI))*
     & DEXP(-CALC)
   70 CONTINUE
      ADELTA=HDELTA/2.0D0*(WT(1)*Y(1)+WT(2)*Y(2)+WT(3)*Y(3)+
     & WT(2)*Y(4)+WT(1)*Y(5))
```

```
      ATOT=ATOT+ADELTA
   80 CONTINUE
      DIFF=ABS(ATOT-AOLD)
C     WRITE(*,*)'DIFF = ',DIFF
C     WRITE(*,*)'TOLER = ',TOLER
      DIFPER=ABS((ATOT-AOLD)/AOLD *100.0)
C     IF (DIFF .GT. TOLER) THEN
      IF(DIFPER .GT. 0.01)THEN
      HDELTA=HDELTA/2.0
      AOLD=ATOT
      GO TO 60
      END IF
   90 CONTINUE
      YN=(YN-ATOT)/PI
      ISIGN=(YN+.00001)/ABS(YN)
      IF(ABS(YN) .GT. 10.0**10)THEN
      YN=ISIGN*(10.0**10)
      END IF
C
C     WRITE(*,*)'YN = ',YN
      RETURN
      END
```

References

Abramowitz, M., Stegun, I, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards, 1972

Altay, Selcuk, *Explicit Integration of Boundary Integral Equations in the Frequency Domain for Wave Scattering*, Doctoral Dissertation, Princeton University, 1986

Ahmad, S., & Banerjee, P.K., "Multi-Domain BEM For Two Dimensional Problems of Elastodynamics", International Journal for Numerical Methods in Engineering , Vol 26, 891-911, 1988

Banerjee, P.K. & Butterfield, R., *Boundary Element Methods in Engineering Science*, McGraw Hill Book Company, London, 1981

Brebbia, C.A. & Dominguez, J., *Boundary Elements An Introductory Course*, McGraw-Hill Book Company, New York, 1989

Chopra, A.K., *Dynamics of Structures : Theory and Applications to Earthquake Engineering*, Prentice-Hall, Inc., Englewood Cliffs, 1995

Costantino, C.J., Miller, C.A., Heymsfield, E., McClean, J.L., & Zhu, Y., "CARES (Computer Analysis for Rapid Evaluation of Structures)", Version 1.1 for U.S. NRC, Fin. No. E20883, City College of New York, 1991

Computer Associates, *CA-Cricket Graph III, Version 1.0*, Computer Associates International, Inc., 1992

Dominguez, J. & Abascal, R., "Seismic Response of Strip Footings on Zoned Viscoelastic Soils", Journal of Engineering Mechanics, Vol 115, No. 5, May 1989

Dravinski, M., "Scattering of SH Waves by Subsurface Topography", Journal of the Engineering Mechanics Division, ASCE, Vol. 108, No. EM1, Feb. 1982, pp 1-17

Dravinski, M,"Scattering of Elastic Waves by an Alluvial Valley", Journal of the Engineering Mechanics Division, ASCE, Vol 108, No. EM1, Feb. 1982, pp. 19-31.

Eshraghi, H. & Dravinski, M., "Transient Scattering of Elastic Waves by Dipping Layers of Arbitrary Shape. Part 1: Antiplane Strain Model", Earthquake Engineering and Structural Dynamics. Vol. 18, 1989, pp 397-415

Eshraghi, H. & Dravinski, M., "Transient Scattering of Elastic Waves by Dipping Layers of Arbitrary Shape. Part 2: Plane Strain Model", Earthquake Engineering and Structural Dynamics. Vol. 18, 1989, pp 417-434

Das, M.D., *Fundamentals of Soil Dynamics*, Elsevier, New York, 1983

Gradshteyn, I.S. and Ryzhik, I.M., *Table of Integrals, Series, and Products*, Academic Press, Inc., New York, 1994

Hadley, P.K., Askar, A., & Cakmak, A.S., "Scattering of Waves by Inclusions in a Nonhomogeneous Elastic Half Space Solved by Boundary Element Methods", National Center For Earthquake Engineering Research, Technical Report NCEER-89-0027, June 15, 1989

Hildebrand, F.B, *Advanced Calculus for Applications*, Prentice-Hall, Inc., Englewood Cliffs, NJ., 1976

Idriss, I.M. & Seed, H.B., "Seismic Response of Horizontal Soil Layers", Journal of the Soil Mechanics and Foundations Division, ASCE, Vol. 94, No. SM4, July 1968, pp 1003-1031

James, M.L., Smith, G.M., Wolford, J.C., *Applied Numerical Methods For Digital Computation With Fortran and CSMP*, Harper and Row, Publishers, Inc., New York, NY, 1977

Roesset, J.M., *Fundamentals of Soil Amplification*, Department of Civil Engineering, Massachusetts Institute of Technology, March 1969

Sanchez-Sesma, F.J., "Site Effects on Strong Ground Motion", Soil Dynamics and Earthquake Engineering, Vol. 6, 1987

Schnabel, P.B., Lysmer, J., Seed, H.B., *SHAKE* , University of California, Berkeley, 1972

Trifunac, M.D., "Surface Motion of a Semi-cylindrical Alluvial Valley for Incident Plane SH Waves", Bulletin of the Seismological Society of America, Vol. 61, 1971, pp. 1755-1770

Trifunac, M.D., "Scattering of Plane SH Waves By A Semi-Cylindrical Canyon", Earthquake Engineering and Structural Dynamics, Vol. 1, 1973, pp. 267-281

Vogt, R.F., Wolf, J.P., & Bachmann, H., "Wave scattering by a Canyon of Arbitrary Shape in a Layered Half-Space", Earthquake Engineering and Structural Dynamics, Vol 16, 1988, pp 803-812

Wolf, J.P., *Dynamic Soil-Structure Interaction*, Prentice-Hall,Inc., N.J., 1985, pp 114-165

Wong, H.L., &Trifunac, M.D., "Surface Motion of a Semielliptical Alluvial Valley for Incident Plane SH waves", Bulletin of the Seismological Society of America Vol. 64, 1974, pp 1389-1408