2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

WAVE PROPAGATION IN ELASTIC SOLIDS

by

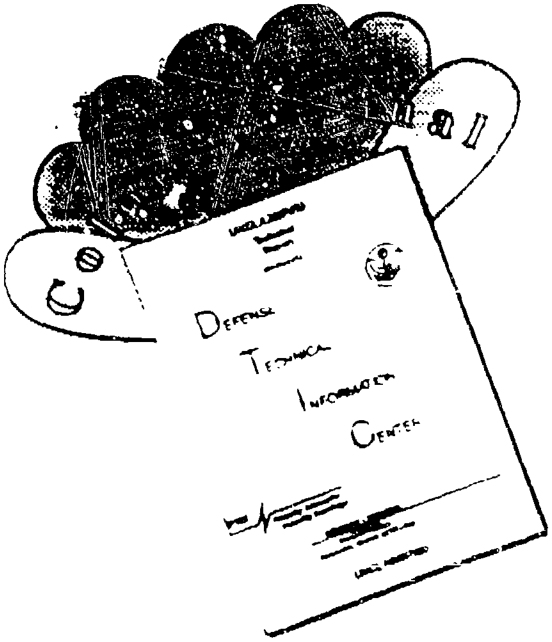
Hugh Joseph McBride

June 1992

Thesis Advisor: Clyde L. Scandrett
Co-Advisor: Van Emden Henson

Approved for public release; distribution is unlimited.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b. OFFICE SYMBOL (If applicable)
MA

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943

7b. ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943-5006

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Naval Postgraduate School

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
Monterey, CA 93943

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Wave Propagation in Elastic Solids

12. PERSONAL AUTHOR(S)
Hugh Joseph McBride

13a. TYPE OF REPORT
Master's thesis

13b. TIME COVERED
FROM______TO______

14. DATE OF REPORT (Year, month day)
June, 1992

15. PAGE COUNT
135

16. SUPPLEMENTARY NOTATION
The views expressed in this paper are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Finite difference approximation of irregularly shaped domains; wave propagation in solids; wave propagation in fluids; fluid structure interaction; finite difference approximations of a nonlocal radiation boundary condition

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis presents a model which simulates the scattering from a fluid loaded I-beam and the resultant behavior due to fluid-structure interaction. Chapter I gives an overview of the problem and describes the characteristics of the solid and fluid, the aspects of periodicity, boundary conditions and the coupling of the two media.

The governing equations of motion are scaled in Chapter II. In Chapter III, the finite-difference formulae for these equations are derived, as is the non-local radiation boundary condition. Difference formulas for typical boundary points of the solid and corner nodes are also derived. All finite difference formulae used are presented in Appendix C. Chapter IV contains numerical results. Conclusions are drawn and areas of the problem that would require further study are in Chapter V.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSICIATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Clyde Scandrett

22b. TELEPHONE (Include Area Code)
(408) 646-2027

22c. OFFICE SYMBOL
MA/Sd

Wave Propagation in Elastic Solids

by

Hugh Joseph McBride
Lieutenant, United States Naval Reserve
B.Sc., University College Galway, Republic of Ireland, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

From the

NAVAL POSTGRADUATE SCHOOL
June 1992

Author: ____________________
Hugh Joseph McBride

Approved by: ____________________
Clyde L. Scandrett, Thesis Advisor

____
Van Emden Henson, Co-Advisor

____
Young W. Kwon, Second Reader

Richard Franke
____
Richard Franke, Chairman
Department of Mathematics

ABSTRACT

This thesis presents a model which simulates the scattering from a fluid loaded I-beam and the resultant behavior due to fluid-structure interaction. Chapter I gives an overview of the problem and describes the characteristics of the solid and fluid, the aspects of periodicity, boundary conditions and the coupling of the two media.

The governing equations of motion are scaled in Chapter II. In Chapter III, the finite-difference formulae for these equations are derived, as is the non-local radiation boundary condition. Difference formulas for typical boundary points of the solid and corner nodes are also derived. All finite difference formulae used are presented in Appendix C. Chapter IV contains numerical results. Conclusions are drawn and areas of the problem that would require further study are in Chapter V.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to acknowledge the contribution of my thesis advisor and co-advisor, Professors Clyde Scandrett and Van Emden Henson, whose infinite patience and careful guidance let me achieve what I once thought impossible.

Of my father and mother, Conal and Nora McBride, and sister Dr. Lynne McBride whose continuing support saw me through the hardest of times.

Of all those families too numerous to mention who showed such generous hospitality to a stranger in a strange land. I am eternally grateful.

And Paddy, this is a big deal!

I. PROBLEM OVERVIEW AND DOMAIN DESCRIPTION

The problem we consider is where an acoustic pressure wave emitted by an active sonar impinges on a submarine. In our case this is a double-hulled submarine. Active sonar works on the principle of detecting the reflection off a solid object of an acoustic pressure wave emitted by a source, by which one can calculate distance and direction to that object as in Figure 1, but instead of concentrating on the reflected wave we look at the interaction between the structure and the incoming wave. This generates scattered pressure waves. The scattered pressure waves include waves which decay as they travel through the fluid (evanescent modes), and waves which do not (propagating modes). Since propagating modes do not decay they can be detected. The main thrust of this thesis is to determine the characteristics of the propagating modes, such as amplitude and energy. The optimum situation would be to perturb the double-hulled structure at a resonance frequency which would increase the amplitude of the propagating modes making them easier to detect. We investigate the steady state behavior of the propagating modes and the resultant shear strain field in the I-beam. At steady state the characteristics of the propagating modes stabilize which might be used as an acoustic signature of the structure. In addition when the shear strain field reaches steady state its value throughout the solid may be used to isolate areas of the I-beam where large stresses occur. This information could be useful in the design of double-hulled vessels.

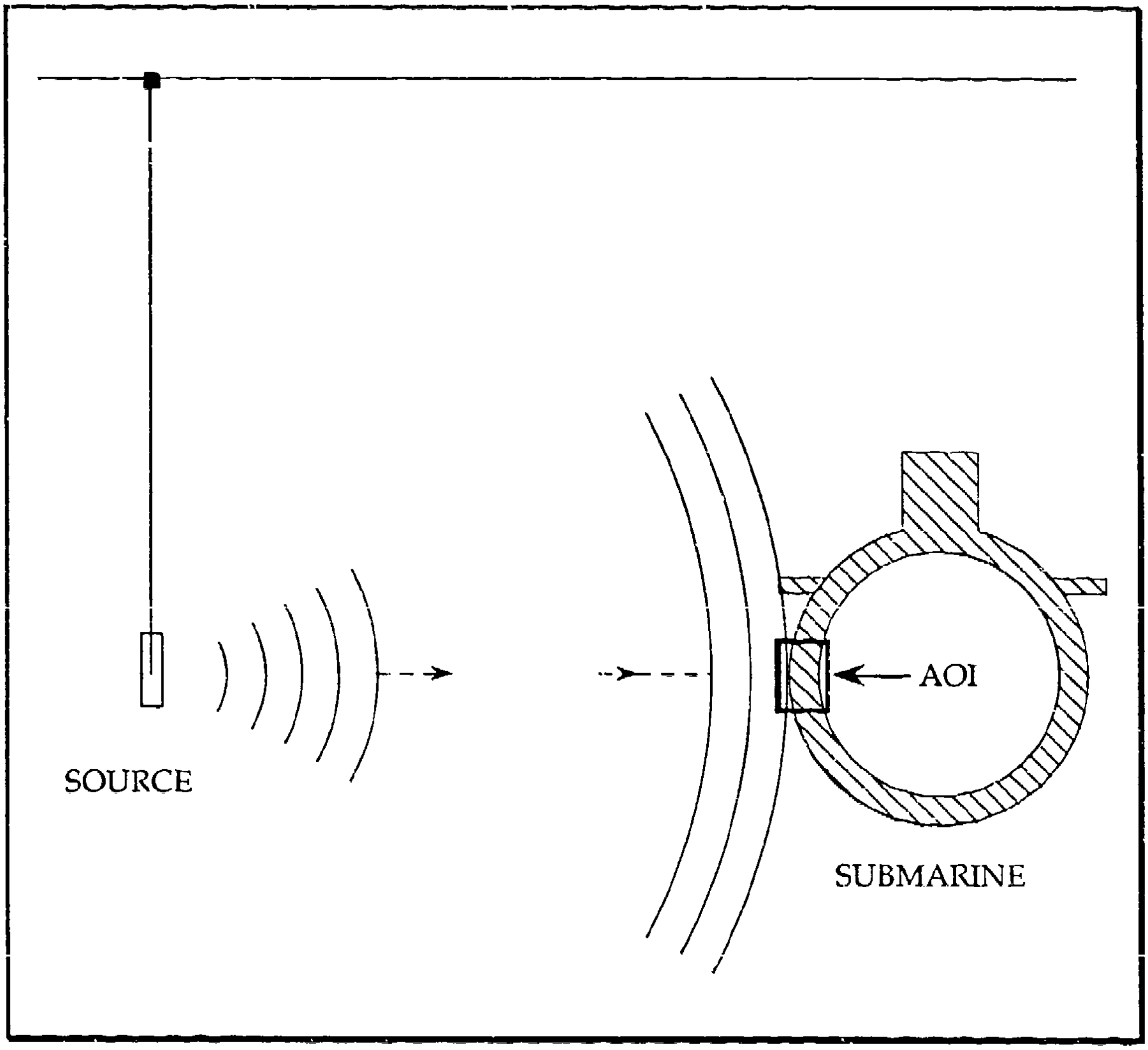


Figure 1. Physical Situation

To reduce the model to one where we can apply numerical techniques to simulate its behavior we must make simplifying assumptions which are discussed below.

A. ASSUMPTIONS

- The source of the pressure waves is far enough away so that the impinging wave can be approximated by a normally incident plane wave.
- There are no other sources present such as might occur with bottom bounce, surface reflection and the like.
- The area of interest--AOI as shown in Figure 2 where the wave impinges is where the inner and outer hulls are joined or connected by a supporting spar forming an I-beam shaped domain.
- The I beam shaped domain is a uniformly continuous linear isotropic elastic medium with no cracks, welds or other deformities.
- The dimensions of our AOI are such that any curvature of the surfaces can be ignored.
- The center spar or support beam occurs at regular intervals through the structure as shown in Figure 3 allowing us to truncate the domain to the left and right of center spar using periodic boundary conditions.
- The incident wave does not displace the submarine.
- The cavities A and B in Figure 2 enclosed by the inner and outer hulls, and the center spar is void and contain no sources.
- We are only interested in displacement in the x and y directions as given in Figure 2 which reduces our problem to two dimensions.
- The fluid is seawater, the solid is steel.

Our model is now reduced to one where we have two coupled media--fluid and solid, and the characteristics of each will now be discussed in turn.

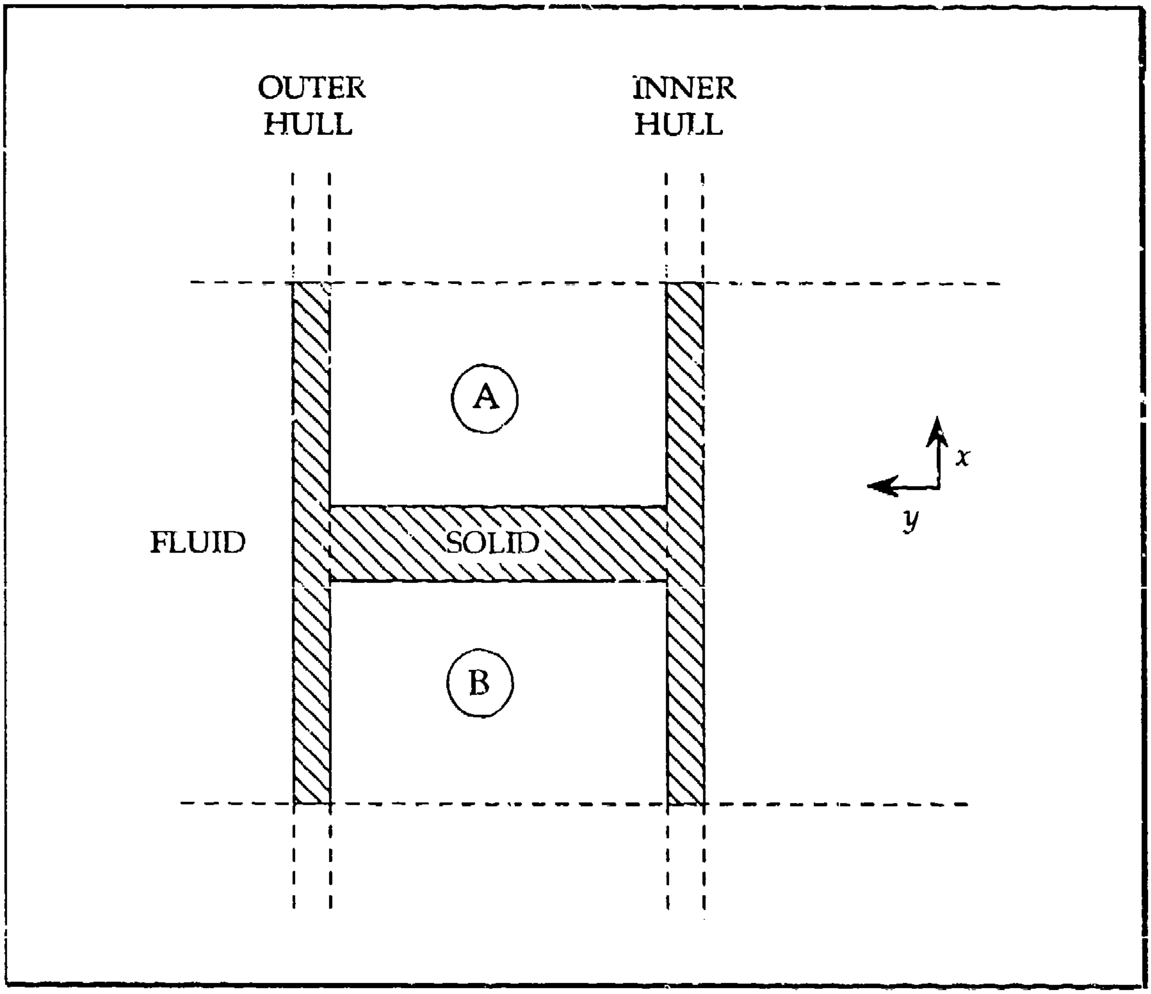


Figure 2. Area of Investigation

A. THE SOLID

A cross-section of the double hull of length 8a is shown in Figure 3. Note *a* is a scaling constant for distance. The domain is essentially infinite in extent in the x (length) and z (depth) directions. Since it is of uniform shape in the z direction, we can neglect this coordinate and reduce our problem to two dimensions. Our area of investigation is outlined in Figure 3, and we are now faced with the problem of providing boundary conditions in the x domain.

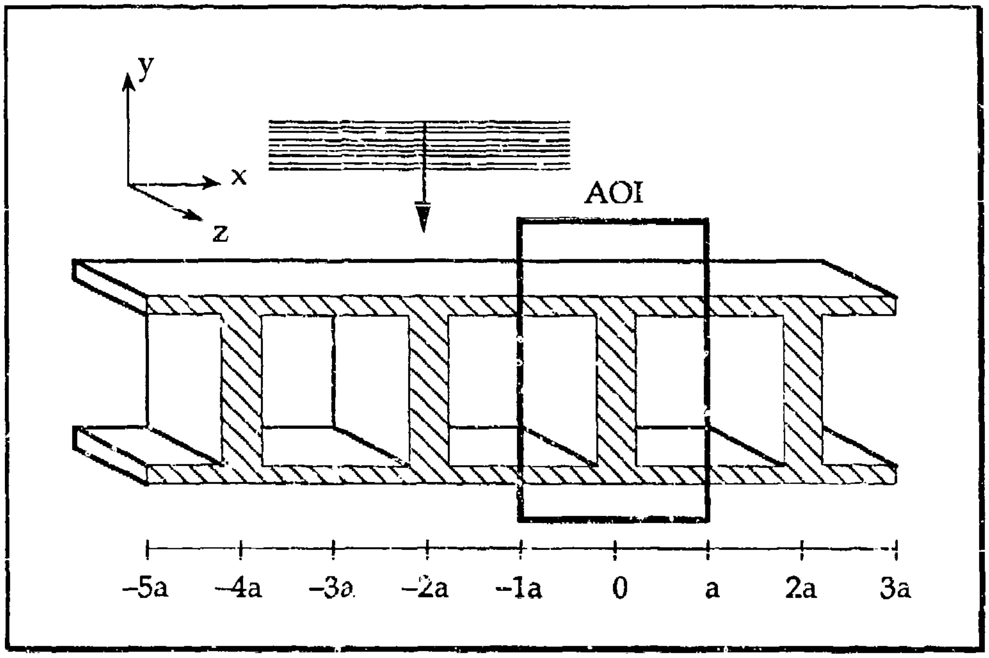


Figure 3. Cross-section of Submarine Hull

We have a normally incident plane wave impinging over one boundary of the domain. Thus the pressure is the same at all points (there is no phase shift) along the fluid/structure boundary. Due to the periodic nature of the structure we expect the solid to behave the same over given intervals of 2a, that is

$$f(x,y,t)=f(x+2aN,y,t) \tag{1}$$

where N is an integer and f a function describing the state of the solid such as displacement, velocity and so on. Using this periodicity we can truncate our domain at a and $-a$ and use the following periodic boundary conditions

$$f(-a,y,t)=f(a,y,t) \tag{2}$$

and

$$f^k(-a,y,t)=f^k(a,y,t) \tag{3}$$

where k is a positive integer and can signify the derivative with respect to x, y or t, depending on the function f.

The solid has now been reduced to a two-dimensional linear isotropic elastic medium whose governing equation of motion for points interior to the domain is given by the plane strain elastic wave equation which is

$$\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+(\lambda+\mu)\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x \partial y}\right)=\rho_s\frac{\partial^2 u}{\partial t^2} \tag{4}$$

$$\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+(\lambda+\mu)\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 u}{\partial x \partial y}\right)=\rho_s\frac{\partial^2 v}{\partial t^2} \tag{5}$$

u and v are displacements in the lateral (x) and transverse (y) direction, μ and λ are Lamé constants, and ρ_S is the density of the solid. The I-beam is deformed due to the imposed fluid pressure. Balancing normal forces at the fluid/solid interface we obtain the boundary condition

$$\tau_{yy} = \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + 2\mu\frac{\partial v}{\partial y} = -p^{\text{total}} \qquad (6)$$

where p^{total} is the total pressure, and τ_{yy} the normal stress component. All other boundary conditions are either traction free or periodic and are summarized below.

$$\left.\begin{aligned} \tau_{xx} &= 0 \\ \tau_{xy} &= 0 \end{aligned}\right\} \text{on surface DH and EI in Figure 4,} \qquad (7)$$

$$\left.\begin{aligned} \tau_{yy} &= 0 \\ \tau_{xy} &= 0 \end{aligned}\right\} \text{on CD, EF, GH, IJ, and KL in Figure 4,} \qquad (8)$$

$$\left.\begin{aligned} \tau_{yy} &= -p^{\text{total}} \\ \tau_{xy} &= 0 \end{aligned}\right\} \text{on AB in Figure 4, and} \qquad (9)$$

$$\left.\begin{matrix} \text{periodic} \\ \text{conditions} \end{matrix}\right\} \text{elsewhere.} \qquad (10)$$

τ_{xx}, τ_{yy} and τ_{xy} are the normal and shear components of stress and are represented by

$$\tau_{yy} = \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + 2\mu\left(\frac{\partial v}{\partial y}\right), \qquad (11)$$

$$\tau_{xx} = \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + 2\mu\left(\frac{\partial u}{\partial x}\right), \text{ and} \qquad (12)$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right). \qquad (13)$$

Figure 4 is the unscaled domain, in which s, and k are constants used to vary the size of the cavities A and B of Figure 2. Note: $0 \leq s < 1$, $0 \leq k < 1$.

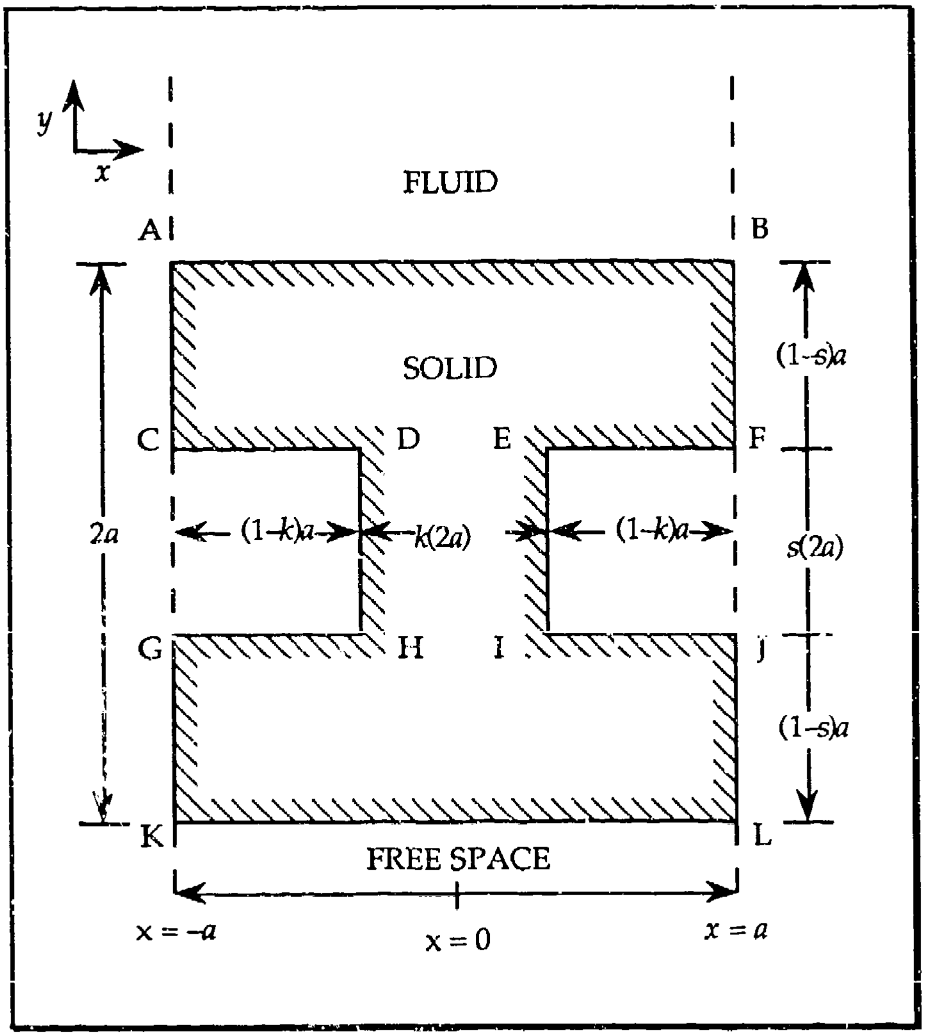


Figure 4. Unscaled Domain

B. THE FLUID

We are interested in the scattered pressure waves generated by perturbations at the fluid/solid interface. The partial differential equation governing the propagation of these waves in the fluid is the linear two-dimensional wave equation given by

$$\frac{1}{c_f^2}\frac{\partial^2 p}{\partial t^2}=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2} \tag{14}$$

where c_f is the speed of sound in the fluid.

These perturbations arise when a normal plane wave of magnitude P and frequency ω impinges on the surface of the solid, causing it to deform. Since the pressure wave is of normal incidence (there is no phase shift at points along the surface of the solid) and the solid is a periodic structure we can expect the behavior of the scattered pressure waves to be the same in given intervals of 2a, thus we can use periodic boundary conditions in the same manner as was used for the solid.

Ideally, if the solid were acoustically hard our total pressure would only have two components, incident and reflected. In reality the solid is perturbed by the incident wave, generating scattered pressure waves which propagate out into the fluid domain. The total pressure in the fluid can be represented by

$$p^{\text{total}} = p^{\text{incident}} + p^{\text{reflected}} + p^{\text{scattered}} \tag{14}$$

where $p^{\text{incident}} = e^{-ik_f y-it}$; $p^{\text{reflected}} = e^{ik_f y-it}$ and $p^{\text{scattered}}$ is to be determined. To avoid cavitation at the fluid-solid boundary we employ the inviscid form of the Navier-Stokes equation which we refer to as the compatibility condition and is given by

$$\left.\frac{\partial p^s}{\partial y}\right|_{y=0} = -\rho_f \frac{\partial^2 v}{\partial t^2} \tag{15}$$

where ρ_f is the density of the fluid. Note that only the scattered term of the pressure is included in this equation since

$$\left.\frac{\partial p^I}{\partial y}\right|_{y=0} = -ik_f e^{-it} \quad \text{and} \quad \left.\frac{\partial p^R}{\partial y}\right|_{y=0} = ik_f e^{-it} \tag{16}$$

thus

$$\left.\left(\frac{\partial p^I}{\partial y} + \frac{\partial p^R}{\partial y}\right)\right|_{y=0} = 0. \tag{17}$$

The scattered pressure waves generated at the fluid/solid interface are composed of propagating (non-decaying) and evanescent (decaying) modes which must be allowed to propagate off to infinity in the positive y direction. To do this we must employ a non-local radiation boundary condition whose implementation will be discussed in greater detail in the discretization section of this paper.

II. NON-DIMENSIONALIZATION OF THE GOVERNING EQUATIONS OF MOTION AND BOUNDARY CONDITIONS

The problem addressed models the propagation of a scattered pressure wave in two dimensions. This is described by the two-dimensional scalar wave equation

$$\frac{1}{c_f^2}\frac{\partial^2 p}{\partial t^2}=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2} \tag{18}$$

where p is the pressure and c_f the speed of sound in the fluid. To facilitate implementation and to free ourselves from the requirement of using a given system of measurement such as metric or imperial we scale or non-dimensionalize Equation 18 as follows. Let

$$\frac{1}{c_f^2}\frac{\partial^2 \overline{p}}{\partial \overline{t}^2}=\frac{\partial^2 \overline{p}}{\partial \overline{x}^2}+\frac{\partial^2 \overline{p}}{\partial \overline{y}^2} \tag{19}$$

represent the unscaled wave equation. We now use the following relationships:

$$t=\omega\overline{t};\quad x=\frac{\overline{x}}{a};\quad y=\frac{\overline{y}}{a};\quad p=\frac{\overline{p}}{P}. \tag{20}$$

Here ω is the scaling constant for time and the frequency of the incident plane wave, a is the half length of the I-beam and the scaling constant for distance and P is the scaling constant for pressure. Note that when taking derivatives with respect to $\overline{x}$ we get

$$\frac{d}{d\bar{x}}=\frac{1}{a}\frac{d}{dx} \quad \text{and} \quad \frac{d^2}{d\bar{x}^2}=\frac{1}{a^2}\frac{d^2}{dx^2}. \tag{21}$$

This also holds true for derivatives with respect to $\bar{y}$, and a similar relation results when taking derivatives with respect to $\bar{t}$. With the above relationships, Equation 19 is now written as

$$\frac{\omega^2 P}{c_f^2}\frac{\partial^2 p}{\partial t^2}=\frac{P}{a^2}\frac{\partial^2 p}{\partial x^2}+\frac{P}{a^2}\frac{\partial^2 p}{\partial y^2}. \tag{22}$$

Cancelling common factors and multiplying Equation 22 by a^2 reduces it to

$$\frac{\omega^2 a^2}{c_f^2}\frac{\partial^2 p}{\partial t^2}=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}. \tag{23}$$

Defining

$$k_f=\frac{\omega a}{c_f}, \tag{24}$$

Equation 23 is now written as

$$k_f^2\frac{\partial^2 p}{\partial t^2}=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}. \tag{25}$$

The elastic wave equation, which is a vector wave equation is given by

$$\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+(\lambda+\mu)\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x\partial y}\right)=\rho_s\frac{\partial^2 u}{\partial t^2} \tag{26}$$

$$\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+(\lambda+\mu)\left(\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 u}{\partial x\partial y}\right)=\rho_s\frac{\partial^2 v}{\partial t^2}. \tag{27}$$

The following relationships allow us to write Equations 26 and 27 in a more convenient form,

$$\frac{\mu}{\rho_s} = c_T^2 \quad \text{and} \quad \frac{\lambda + 2\mu}{\rho_s} = c_L^2. \tag{28}$$

The constants c_L and c_T are the lateral and transverse velocities of the solid. The unscaled forms of Equations 26 and 27 are now written as

$$c_T^2\left(\frac{\partial^2 \bar{u}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{u}}{\partial \bar{y}^2}\right) + \left(c_L^2 - c_T^2\right)\left(\frac{\partial^2 \bar{u}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{v}}{\partial \bar{x}\partial \bar{y}}\right) = \frac{\partial^2 \bar{u}}{\partial \bar{t}^2} \tag{29}$$

$$c_T^2\left(\frac{\partial^2 \bar{v}}{\partial \bar{x}^2} + \frac{\partial^2 \bar{v}}{\partial \bar{y}^2}\right) + \left(c_L^2 - c_T^2\right)\left(\frac{\partial^2 \bar{v}}{\partial \bar{y}^2} + \frac{\partial^2 \bar{u}}{\partial \bar{x}\partial \bar{y}}\right) = \frac{\partial^2 \bar{v}}{\partial \bar{t}^2}. \tag{30}$$

We use the same scaling relationships as before for x, y and t (Equation 20) as well as

$$u = \frac{\bar{u}}{D} \quad \text{and} \quad v = \frac{\bar{v}}{D}, \tag{31}$$

which are the scaling relationships for the displacement, and rewrite Equations 29 and 30 as

$$c_T^2\left(\frac{D}{a^2}\frac{\partial^2 u}{\partial x^2} + \frac{D}{a^2}\frac{\partial^2 u}{\partial y^2}\right) + \left(c_L^2 - c_T^2\right)\left(\frac{D}{a^2}\frac{\partial^2 u}{\partial x^2} + \frac{D}{a^2}\frac{\partial^2 v}{\partial x \partial y}\right) = D\omega^2 \frac{\partial^2 u}{\partial t^2} \tag{32}$$

$$c_T^2\left(\frac{D}{a^2}\frac{\partial^2 v}{\partial x^2} + \frac{D}{a^2}\frac{\partial^2 v}{\partial y^2}\right) + \left(c_L^2 - c_T^2\right)\left(\frac{D}{a^2}\frac{\partial^2 v}{\partial y^2} + \frac{D}{a^2}\frac{\partial^2 u}{\partial x \partial y}\right) = D\omega^2 \frac{\partial^2 v}{\partial t^2}. \tag{33}$$

Defining

$$k_L = \frac{\omega a}{c_L} \quad \text{and} \quad k_T = \frac{\omega a}{c_T}, \tag{34}$$

and dividing Equations 32 and 33 through by ω^2 and cancelling common factors they reduce to

$$\frac{1}{k_T^2}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x \partial y}\right)=\frac{\partial^2 u}{\partial t^2} \tag{35}$$

$$\frac{1}{k_T^2}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 u}{\partial x \partial y}\right)=\frac{\partial^2 v}{\partial t^2}. \tag{36}$$

Collecting like terms gives us the final form

$$\frac{1}{k_L^2}\frac{\partial^2 u}{\partial x^2}+\frac{1}{k_T^2}\frac{\partial^2 u}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(\frac{\partial^2 v}{\partial x \partial y}\right)=\frac{\partial^2 u}{\partial t^2} \tag{37}$$

$$\frac{1}{k_T^2}\frac{\partial^2 v}{\partial x^2}+\frac{1}{k_L^2}\frac{\partial^2 v}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(\frac{\partial^2 u}{\partial x \partial y}\right)=\frac{\partial^2 v}{\partial t^2}. \tag{38}$$

The surfaces in contact with free space are stress free. (The fluid/solid boundary is dealt with separately). Therefore the stresses τ_{xx} and τ_{xy} on surfaces EI and DH and τ_{yy} and τ_{xy} on surfaces CD, EF, GH, IJ and KL of Figure 5 are zero.

These components of the stress tensor can be written

$$\tau_{xy}=\mu\left(\frac{\partial \overline{u}}{\partial \overline{y}}+\frac{\partial \overline{v}}{\partial \overline{x}}\right)=0 \tag{39}$$

$$\tau_{xx}=\lambda\left(\frac{\partial \overline{u}}{\partial \overline{x}}+\frac{\partial \overline{v}}{\partial \overline{y}}\right)+2\mu\frac{\partial \overline{u}}{\partial \overline{x}}=0 \tag{40}$$

$$\tau_{yy}=\lambda\left(\frac{\partial \overline{u}}{\partial \overline{x}}+\frac{\partial \overline{v}}{\partial \overline{y}}\right)+2\mu\frac{\partial \overline{v}}{\partial \overline{y}}=0 \tag{41}$$

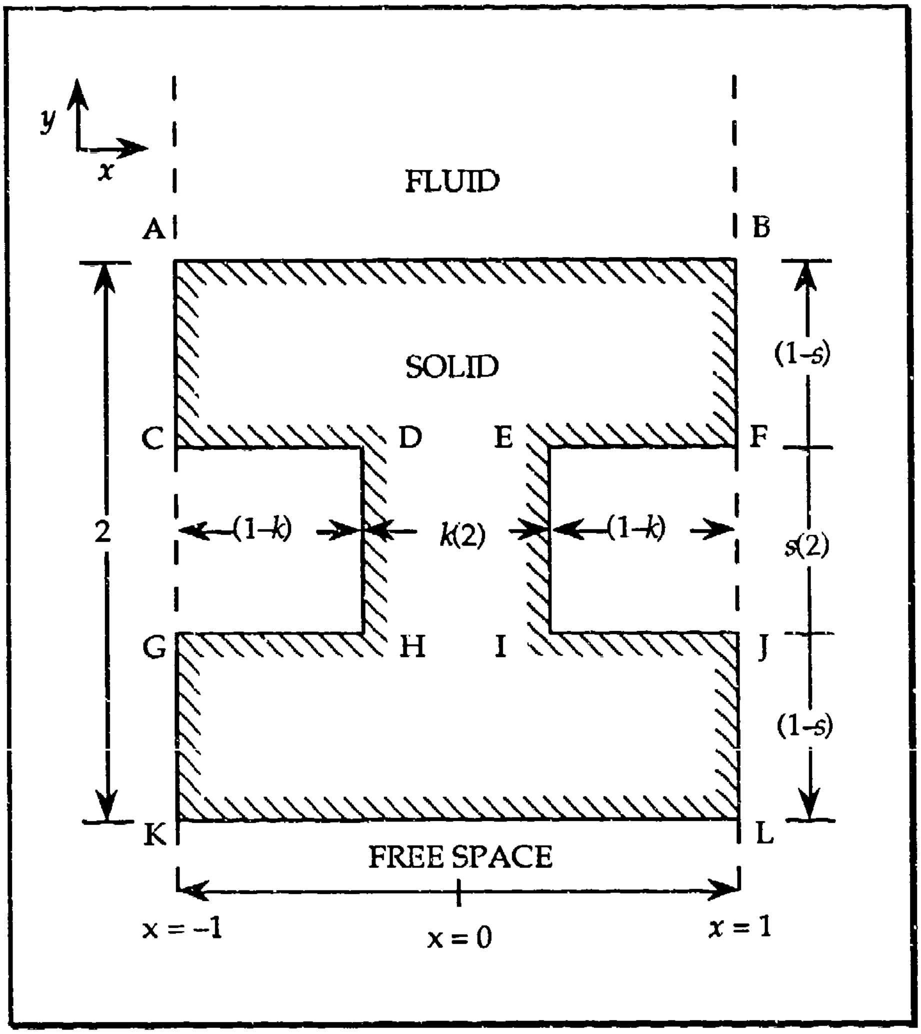


Figure 5. Scaled Domain

where $\bar{x}$, $\bar{y}$, $\bar{u}$ and $\bar{v}$ are the unscaled components of distance and displacement and μ and λ are the Lamé constants. We will now scale Equations 39 through 41 in turn.

For Equation 39 using the same scaling relationships as before (see Equations 20 and 31) gives

$$\mu\left(\frac{D}{a}\frac{\partial u}{\partial y}+\frac{D}{a}\frac{\partial v}{\partial x}\right)=0. \tag{42}$$

Cancelling common factors reduces it to

$$\tau_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}=0. \tag{43}$$

For Equation 40 we first divide through by ρ_s the density of the solid, and using the same scaling relationships we get

$$\frac{\lambda}{\rho_s}\left(\frac{D}{a}\frac{\partial u}{\partial x}+\frac{D}{a}\frac{\partial v}{\partial y}\right)+\frac{2\mu}{\rho_s}\frac{D}{a}\frac{\partial u}{\partial x}=0. \tag{44}$$

We know that

$$\mu/\rho_s=c_T^2. \tag{45}$$

Similarly it can be shown that

$$\frac{\lambda}{\rho_s}=c_L^2-2c_T^2. \tag{46}$$

Using Equations 45 and 46, substituting into Equation 44, and cancelling common factors gives

$$\left(c_L^2-2c_T^2\right)\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+2c_T^2\frac{\partial u}{\partial x}=0 \tag{47}$$

or

$$\left(\frac{c_L^2}{c_T^2}-2\right)\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+2\frac{\partial u}{\partial x}=0. \tag{48}$$

Using the previous definitions oı k_L and k_T it can be shown that

$$\frac{c_L^2}{c_T^2} = \frac{k_T^2}{k_L^2}. \tag{49}$$

and when used in Equation 48 it reduces to our final form

$$\frac{k_T^2}{k_L^2}\frac{\partial u}{\partial x} + \left(\frac{k_T^2}{k_L^2} - 2\right)\frac{\partial u}{\partial y} = 0. \tag{50}$$

Following the same procedure for Equation 41 its final form is

$$\left(\frac{k_T^2}{k_L^2} - 2\right)\frac{\partial u}{\partial x} + \frac{k_T^2}{k_L^2}\frac{\partial v}{\partial y} = 0. \tag{51}$$

At the interface the fluid cannot exert a force tangential to the boundary, hence the shear component of stress is zero there, and is given by

$$\tau_{xy} = \mu\left(\frac{\partial \overline{u}}{\partial \overline{x}} + \frac{\partial \overline{v}}{\partial \overline{y}}\right) = 0. \tag{52}$$

For the normal component of stress at the interface we refer back to Equation 6, Section B and Equation 15 in Section C of Chapter I to see that the normal component of stress is given by

$$\tau_{yy} = \lambda\left(\frac{\partial \overline{u}}{\partial \overline{x}} + \frac{\partial \overline{v}}{\partial \overline{y}}\right) + 2\mu\frac{\partial \overline{u}}{\partial \overline{x}} = -\left(p^I + p^R + p^S\right), \tag{53}$$

where the superscripts I, R and S signify incident, reflected and scattered pressures. The incident and reflected pressures are given by

$$p^I = e^{-ik_f y - it} \text{ and } p^R = e^{ik_f y - it}$$

τ_{xy} is scaled the same as before. To scale τ_{yy} properly we will need the compatibility condition, whose unscaled form is

$$-\frac{\partial \bar{p}^s}{\partial \bar{y}}\bigg|_{y=0} = \rho_f \frac{\partial^2 \bar{v}}{\partial \bar{t}^2}. \tag{54}$$

We refer back to Equation 17, Chapter I, Section C to see that only the scattered pressure need be considered here.

Scaling Equation 54 we get

$$-\frac{P}{a}\frac{\partial p^s}{\partial y}\bigg|_{y=0} = \rho_f \omega^2 D \frac{\partial^2 v}{\partial t^2}, \tag{55}$$

or

$$-P\frac{\partial p^s}{\partial y}\bigg|_{y=0} = \rho_f \omega^2 a D \frac{\partial^2 v}{\partial t^2}. \tag{56}$$

Equation 56 gives us a convenient choice for the scaling constant for pressure of

$$P = \omega^2 a D \rho_f \tag{57}$$

and when substituted back into Equation 56 cancels as a common factor to give

$$-\frac{\partial p^s}{\partial y}\bigg|_{y=0} = \frac{\partial^2 v}{\partial t^2}. \tag{58}$$

Dividing Equation 53 by ρ_s and scaling gives

$$\frac{\lambda D}{\rho_s a}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \frac{2\mu D}{\rho_s a}\frac{\partial v}{\partial y} = \frac{-P}{\rho_s}\left(p^I + p^R + p^s\right)\Big|_{y=0}. \tag{59}$$

Substituting the value of P from Equation 57 we obtain

$$\frac{\lambda D}{\rho_s a}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+\frac{2\mu D}{\rho_s a}\frac{\partial v}{\partial y}=-\frac{\rho_f}{\rho_s}\omega^2 aD\left(p^I+p^R+p^S\right)\Big|_{y=0} \tag{60}$$

or

$$\left(\frac{k_T^2}{k_L^2}-2\right)\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+2\frac{\partial v}{\partial y}=-\varepsilon k_T^2\left(p^I+p^R+p^S\right)\Big|_{y=0} \tag{61}$$

where $\varepsilon = \rho_f/\rho_s$.

Evaluating the incident and reflected pressures at $y = 0$ we get $p^I = p^R = e^{-it}$, and when substituted into Equation 61 gives

$$\left(\frac{k_T^2}{k_L^2}-2\right)\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+2\frac{\partial v}{\partial y}=-2\varepsilon k_T^2 e^{-it}-\varepsilon k_T^2 p^s. \tag{62}$$

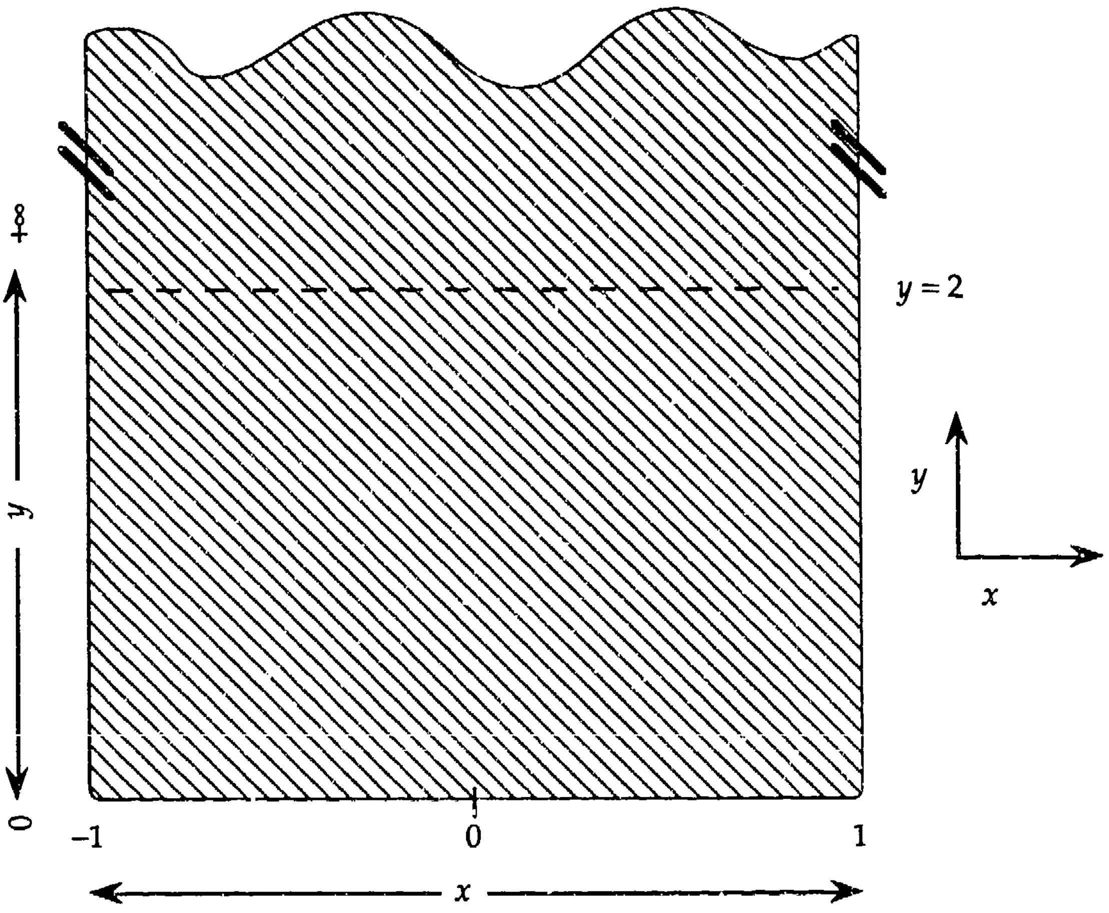
III. DERIVATION OF THE FINITE DIFFERENCE FORMULAE FOR THE GOVERNING EQUATIONS

Throughout the derivation of the finite difference approximations we identify gridpoints of the scaled domain with the subscripts i and j and superscript n. Variation in the x direction is denoted by i, $1 \le i \le N$, where N is the total number of subdivisions (since we are using a square grid the number of subdivisions in the y direction is the same), variation in y with j, $1 \le j \le N$, and time with n, $0 \le n < \infty$. Lower case indices denote varying quantities, while upper case letters denote fixed quantities. $p^{n}_{i,K}$ would be the value of the scattered pressure for all values of i at the grid level $y = K\Delta y$, thus $p^{n}_{i,K}$ is a vector, while $p^{n}_{i,j}$ is an element.

As [illegible] discussed above we use the letter i to denote variation in the x direction. This is a common convention and we do not wish to deviate from it. To avoid confusion with the complex quantity, $\sqrt{-1}$, also commonly denoted by i, we state the following rule, that whenever the letter i appears as a superscript it denotes the complex quantity $\sqrt{-1}$, and when appearing as a subscript it is an index denoting variation in the x direction.

A. FINITE DIFFERENCE APPROXIMATIONS FOR THE EQUATIONS GOVERNING THE BEHAVIOR OF THE FLUID

The scaled domain of consideration as shown in Figure 6 has periodic boundary conditions applied at $x = 1$ and -1 and a radiation boundary condition for the propagating modes at $y = 2$. To model the two-dimensional



$$k_f^2 \frac{\partial p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$$

$$\left.\begin{aligned} p(1,y,t) &= p(-1,y,t) \\ \left.\frac{\partial p}{\partial x}\right|_{(1,y,t)} &= \left.\frac{\partial p}{\partial x}\right|_{(-1,y,t)} \end{aligned}\right\} \text{Periodic Boundary Conditions}$$

$$B_k(p) = \frac{\partial p}{\partial y} + \beta_k \frac{\partial p}{\partial t} \Big\} \text{ Radiation Boundary Condition for the } k^{th} \text{ propagating mode}$$

Figure 6. Fluid Domain

wave equation numerically we must derive an equivalent finite difference formula, from which we can solve for the pressure for all subsequent time levels as follows:

$$k_f^2 \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \qquad (63)$$

is the two-dimensional wave equation as derived in Chapter II Section A. Using central difference approximation for all second order partial derivatives the equivalent finite difference equation for Equation 63 is

$$\frac{k_f^2}{\Delta t^2}\left[p_{i,j}^{n+1} - 2p_{i,j}^{n} + p_{i,j}^{n-1}\right] = \frac{1}{h^2}\left[p_{i+1,j}^{n} - 2p_{i,j}^{n} + p_{i-1,j}^{n}\right] + \frac{1}{h^2}\left[p_{i,j+1}^{n} - 2p_{i,j}^{n} + p_{i,j-1}^{n}\right]$$

$$(64)$$

where $h = \Delta x = \Delta y$ is the step size and Δt is the increment in time. The truncation error for Equation 64 is $O(h^2)$ in space and $O(\Delta t^2)$ in time. Solving for $p_{i,j}^{n+1}$ explicitly we have

$$p_{i,j}^{n+1} = \left(2 - \frac{4\Delta t^2}{k_f^2 h^2}\right) p_{i,j}^{n} + \frac{\Delta t^2}{k_f^2 h^2}\left[p_{i+1,j}^{n} + p_{i-1,j}^{n} + p_{i,j+1}^{n} + p_{i,j-1}^{n}\right] - p_{i,j}^{n-1} \qquad (65)$$

Letting $\rho^2 = \frac{\Delta t^2}{k_f^2 h^2}$ Equation 65 can be written as

$$p_{i,j}^{n+1} = 2\left(1 - 2\rho^2\right) p_{i,j}^{n} + \rho^2\left[p_{i+1,j}^{n} + p_{i-1,j}^{n} + p_{i,j+1}^{n} + p_{i,j-1}^{n}\right] - p_{i,j}^{n-1}. \qquad (66)$$

The Von Neumann stability criterion

$$\rho \le \frac{1}{\sqrt{2}} \qquad (67)$$

must be satisfied to ensure the stability of Equation 66.[1]

Special attention must be paid when applying the wave equation along the boundaries at $x = 1$ and -1 for it is here that we make use of the periodic boundary conditions. Applying Equation 66 at $x = -1$, $i = 0$ and $y = jh$ (see Figure 7) yields

$$p_{0,j}^{n+1} = 2\left(1-2\rho^2\right)p_{0,j}^{n} - \rho^2\left[p_{i,j}^{n} - p_{-1,j}^{n} + p_{0,j+1}^{n} + p_{0,j-1}^{n}\right] - p_{0,j}^{n-1} \qquad (68)$$

This requires the value of p at the point $(-1, jh, n\Delta t)$ which lies outside the domain. By using the periodic boundary conditions

$$p(1, y, t) = p(-1, y, t) \qquad (69)$$

$$\left.\frac{\partial p}{\partial x}\right|_{(+1,y,t)} = \left.\frac{\partial p}{\partial x}\right|_{(-1,y,t)}. \qquad (70)$$

we can substitute the value of $((N-1)h, jh, n\Delta t)$ (where N is the total number of subdivisions in the x direction) for the value at $(-1, jh, n\Delta t)$, allowing us to evaluate the wave equation at the boundary.

B. APPLICATION OF THE RADIATION BOUNDARY CONDITION IN THE NUMERICAL SCHEME

As was mentioned at the end of Chapter I (Section C), we apply a non-local radiation boundary condition (referred to as *nlrb*) to the fluid to simulate an infinite domain in the positive y direction. Our domain is truncated at x=1 and -1, forcing our fluid domain to act as a waveguide. The scattered pressure can be represented as a series of plane waves which take the form

[1] For treatment of the von Neumann Stability Criterion see Appendix A.

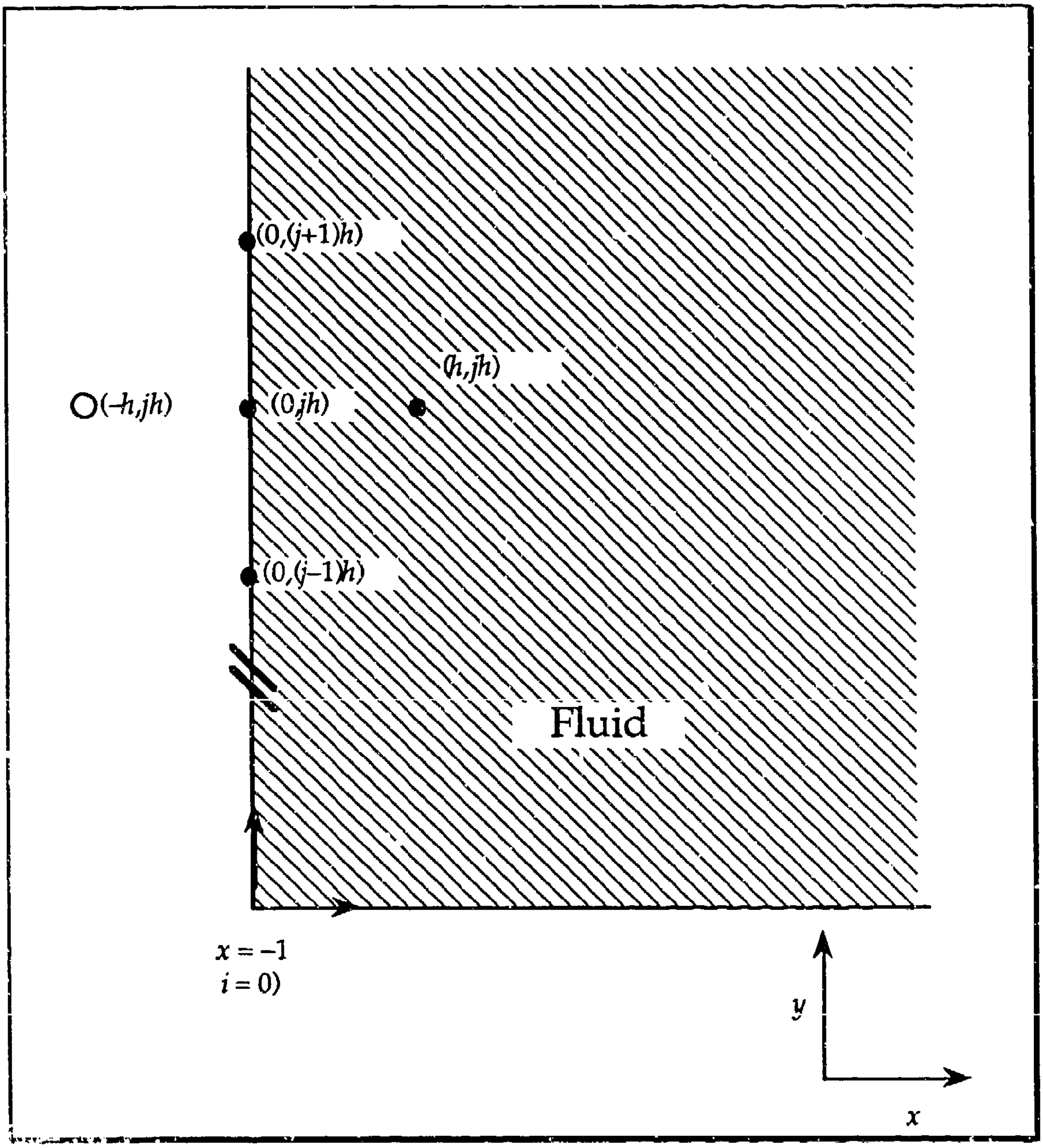


Figure 7. Left Boundary for the Fluid

$$e^{i(\gamma_k x+\beta_k y-t)} \quad \text{where} \quad \beta_k = \begin{cases} \sqrt{k_f^2-\gamma_k^2} & \text{for propagating modes} \\ \mathrm{i}\sqrt{\gamma_k^2-k_f^2} & \text{for evanescent modes} \end{cases} \tag{71}$$

and $\gamma_k = k\pi$. We note that when $\beta_k = i\sqrt{\gamma_k - k_f^2}$ Equation 71 yields

$$e^{-\beta_k y+i(\gamma_k x-t)}$$

which is an exponentially decaying quantity for positive values of y. This fact allows us to neglect evanescent modes when applying the radiation boundary condition, at $y = 2$.

The total scattered pressure is given by

$$p = \sum_{k=-\infty}^{\infty} a_k e^{i(\gamma_k x+\beta_k y-t)}. \tag{72}$$

This is composed of propagating and evanescent modes. Far from the fluid solid interface where only the propagating modes are assumed present (for reasons given above) the scattered pressure is written

$$p = \sum_{k=-M}^{M} a_k e^{i(\gamma_k x+\beta_k y-t)}, \tag{73}$$

where M is the total number of modes (positive or negative) under consideration. At the boundary $y = 2$, we apply the *nlrb* operator (Scandrett and Kriegsmann, 1992, unpublished paper).

$$B_k(p) = \frac{\partial p}{\partial y} + \beta_k \frac{\partial p}{\partial y} \tag{74}$$

to the individual modes of the scattered pressure of Equation 73 which yields

$$B(p) = \sum_{k=-M}^{M} \frac{\partial}{\partial y}\left(a_k e^{i(\gamma_k x+\beta_k y-t)} \right) + \sum_{k=-M}^{M} \beta_k \left(\frac{\partial}{\partial t}\left(a_k e^{i(\gamma_k x+\beta_k y-t)} \right) \right). \tag{75}$$

Equation 75 reduces to

$$B(p) = \sum_{k=-M}^{M} (i\beta_k - i\beta_k) a_k e^{i(\gamma_k x+\beta_k y-t)}. \tag{76}$$

The right-hand side of Equation 76 is identically zero. The boundary operator has annihilated the propagating modes and since the evanescent modes are assumed to have negligible magnitude there, any scattered pressure waves reaching the boundary experience no reflection, simulating an infinite domain in the positive y direction.

To apply the *nlrb* operator at the boundary $y = 2$, we rewrite Equation 75 as

$$\left.\frac{\partial p}{\partial y}\right|_{\substack{y=Jh \\ t=T}} + \sum_{k=-M}^{M} \beta_k \frac{\partial}{\partial t}\left(\alpha_k(T) e^{i(\beta_k Jh+\gamma_k x)} \right) \cong 0 \tag{77}$$

where we evaluate the pressure at a constant time T and along the boundary $y = Jh$ (i.e. at $y = 2$). We incorporate a_k and e^{-iT} as $a_k e^{-iT}$ and define this to be a new constant $\alpha_k(T)$. Note that $\|\alpha_k(T)\| = \|a_k\|$ since $\|e^{-iT}\| = 1$ for all values of T. $\alpha_k(T)$ is unknown so we must derive an alternative expression to be able to evaluate Equation 77. The k^{th} propagating mode can be written as

$$p_k = a_k e^{i(\gamma_k x+\beta_k y-t)} \tag{78}$$

and when evaluated at $y=Jh$ and at $t = T$ and employing our new constant $\alpha_k(T)$ we obtain

$$p_k\Big|_{\substack{y=Jh\\t=T}} = \alpha_k(T)e^{i\beta_k Jh}e^{i\gamma_k x}. \qquad (79)$$

To isolate $\alpha_k(T)e^{i\beta_k Jh}$ we multiply Equation 79 by $e^{-i\gamma_k x}$ (we apply orthogonality) and integrate over the x domain to obtain

$$\int_{-1}^{1} p(x,Jh,T)e^{-i\gamma_k x}dx = 2\alpha_k(T)e^{i\beta_k Jh} \qquad (80)$$

or

$$\alpha_k(T)e^{i\beta_k Jh} = \frac{1}{2}\int_{-1}^{1} p(\xi,Jh,T)e^{-i\gamma_k \xi}d\xi, \qquad (81)$$

where ξ is a dummy variable of integration. We substitute this value of $\alpha_k(T)e^{i\beta_k Jh}$ back into Equation 77 to obtain

$$\frac{\partial p}{\partial y}\Big|_{\substack{y=Jh\\t=T}} + \frac{1}{2}\sum_{k=-M}^{M}\beta_k\int_{-1}^{1}\frac{\partial p}{\partial t}(\xi,Jh,T)e^{i\gamma_k(x-\xi)}d\xi = 0 \qquad (82)$$

which allows us to apply the *nlrb* at the boundary y = 2 and from Equation 81 we will be able to evaluate the amplitudes of the propagating modes of the scattered pressure.

We now proceed with deriving the finite difference approximation for the radiation boundary condition. Using a central difference approximation for $\frac{\partial p}{\partial y}\Big|_{\substack{y=Jh\\t=T}}$, Equation 82 is written as

$$\frac{p^n_{i,J+1} + p^n_{i,J-1}}{2h} + \frac{1}{2}\sum_{k=-M}^{M} \beta_k \int_{-1}^{1} \frac{\partial p}{\partial t}(\xi, Jh, T) e^{i\gamma_k(x-\xi)} d\xi = 0. \tag{83}$$

The trapezoidal rule for integration is

$$\int_a^b f(x)dx \cong \frac{h}{2}(f_1 + 2f_2 + 2f_3 \ldots + 2f_{n-1} + f_n) \tag{84}$$

and is used for the integral in Equation 83, however it can be more compactly expressed as

$$\int_a^b f(x)dx = h\sum_{r=1}^{I} \delta_r f_r \quad \text{where} \quad \delta_r = \begin{cases} \frac{1}{2} & r = 1 \text{ or } I \\ 1 & \text{elsewhere,} \end{cases} \tag{85}$$

where I is the total number of nodes in the x direction. When substituted into Equation 83, using a central difference approximation for $\frac{\partial p}{\partial t}(\xi, Jh, t)$, Equation 83 can be written

$$\frac{p^n_{i,J+1} - p^n_{i,J-1}}{2h} + \frac{1}{2}\sum_{k=-M}^{M} \beta_k \cdot \frac{h}{2\Delta t}\sum_{r=1}^{I} \delta_r e^{i\gamma_k(x_i - \xi_r)}\left(p^{n+1}_{r,J} - p^{n-1}_{r,J}\right) = 0. \tag{86}$$

Multiplying by $\frac{2\Delta t}{h}$, we have

$$\frac{2\Delta t}{h^2}\left(p^n_{i,J+1} - p^n_{i,J-1}\right) + \sum_{r=1}^{I} \sum_{k=-M}^{M} \beta_k \delta_r e^{i\gamma_k(x_i - \xi_k)}\left(p^{n+1}_{r,J} - p^{n-1}_{r,J}\right) = 0. \tag{87}$$

Define A to be a matrix whose i,r entry is given by

$$A(i,r) = \sum_{k=-M}^{M} \beta_k \delta_r e^{i\gamma_k(x_i - \xi_r)}. \tag{88}$$

Upon substitution into Equation 87 we obtain

$$\frac{2\Delta t}{h^2}\left(p_{i,J+1}^{n}-p_{i,J-1}^{n}\right)+Ap_{r,J}^{n+1}-Ap_{r,J}^{n-1}=0. \tag{89}$$

We note that $p_{i,J+1}^{n}, p_{i,J-1}^{n}, p_{r,J}^{n+1}$ and $p_{r,J}^{n-1}$ are all vectors due to the J subscript as described at the beginning of this chapter. The radiation boundary condition and the wave equation must both be satisfied at the artificial boundary $y = 2$. Applying either of the two conditions will require the use of pseudonodes which lie outside the domain. Through the combination of the two equations we will be able to eliminate this requirement. Reproducing the two-dimensional wave equation and the radiation boundary condition, (substitute k for all x indices in the wave equation and radiation boundary condition, since these are dummy indices), we have

$$p_{k,J}^{n+1}-2p_{k,J}^{n}+p_{k,J}^{n-1}-\frac{\Delta t^2}{k_f^2h^2}\left(p_{k-1,J}^{n}+p_{k+1,J}^{n}+p_{k,J+1}^{n}+p_{k,J-1}^{n}-4p_{k,J}^{n}\right)=0 \tag{90}$$

$$\frac{2\Delta t}{h^2}\left(p_{k,J+1}^{n}-p_{k,J-1}^{n}\right)+Ap_{k,J}^{n+1}-Ap_{k,J}^{n-1}=0. \tag{91}$$

Note that both equations are being evaluated at $y = 2$ and that the vectors $p_{k-1,J}^{n}$ and $p_{k+1,J}^{n}$ have a circular shift and are of the same dimension as the rest of the vector in Equation 91. Collecting like terms in Equations 90 and 91 we obtain

$$-\frac{\Delta t^2}{k_f^2h^2}p_{k,J+1}^{n}-\frac{\Delta t^2}{k_f^2h^2}p_{k,J-1}^{n}+p_{k,J}^{n+1}+p_{k,J}^{n-1}-\frac{\Delta t^2}{k_f^2h^2}\left[p_{k-1,J}^{n}+p_{k+1,J}^{n}-4p_{k,J}^{n}\right]-2p_{k,J}^{n}=0 \tag{92}$$

$$\frac{2\Delta t}{h^2} p^n_{k,J+1} - \frac{2\Delta t}{h^2} p^n_{k,J-1} + A p^{n+1}_{k,J} - A p^{n-1}_{k,J} = 0. \tag{93}$$

Adding $\frac{\Delta t}{2k_f^2}$ times Equation 93 to Equation 92 yields

$$-2p^n_{k,J-1}\left(\frac{\Delta t^2}{k_f^2 h^2}\right) + p^{n+1}_{k,J}\left(I + \frac{\Delta t}{2k_f^2}A\right) + p^{n-1}_{k,J}\left(I - \frac{\Delta t}{2k_f^2}A\right)$$

$$+\left(\frac{4\Delta t^2}{k_f^2 h^2} - 2\right)p^n_{k,J} - \frac{\Delta t^2}{k_f^2 h^2}p^n_{k-1,J} - \frac{\Delta t^2}{k_f^2 h^2}p^n_{k+1,J} = 0. \tag{94}$$

Solving for $p^{n+1}_{k,J}$ in Equation 94 we obtain

$$p^{n+1}_{k,J} = \left(I + \frac{\Delta t}{2k_f^2}A\right)^{-1}\left[\frac{2\Delta t^2}{k_f^2 h^2}p^n_{k,J-1} + \frac{\Delta t^2}{k_f^2 h^2}p^n_{k-1,J} + \left(2 - \frac{4\Delta t^2}{k_f^2 h^2}\right)p^n_{k,J}\right.$$

$$\left. + \frac{\Delta t^2}{k_f^2 h^2}p^n_{k+1,J} - \left(I - \frac{\Delta t}{2k_f^2}A\right)p^{n-1}_{k,J}\right] \tag{95}$$

The terms differenced in x, that is $p^n_{k-1,J}$, $p^n_{k,J}$, $p^n_{k+1,J}$ can be more compactly expressed as $Tp^n_{k,J}$ where T is a tridiagonal matrix with $2 - \frac{4\Delta t^2}{k_f^2 h^2}$ on the main diagonal and $\frac{\Delta t^2}{k_f^2 h^2}$ on the sub and super diagonals. We must also allow for the periodic boundary conditions when constructing T. To do this we replace the (N, 2) and the (1, N−1) elements of T with $\frac{\Delta t^2}{k_f^2 h^2}$ where T is an N by N matrix. The general form of T can be seen from Equation 16 Appendix C. Equation 95 is now written as

$$p_{k,J}^{n+1} = \left(I + \frac{\Delta t}{2k_f^2} A \right)^{-1} \left[\frac{2\Delta t^2}{k_f^2 h^2} p_{k,J-1}^{n} + T p_{k,J}^{n} - \left(I - \frac{\Delta t}{2k_f^2} A \right) p_{k,J}^{n-1} \right] \qquad (96)$$

which satisfies both the wave equation and the radiation condition at the boundary.

C. FINITE DIFFERENCE APPROXIMATION FOR THE ELASTIC WAVE EQUATION

To investigate the propagation of disturbances in an "I-beam"-shaped domain as shown in Figure 5, it will be necessary to apply the elastic wave equation to points in the interior of the domain. (The boundaries will be dealt with in a separate section.) By only considering displacements in the x (lateral) and y (transverse) directions, the problem becomes one of plane strain in two dimensions. Reproducing the scaled equations derived earlier for motion in the lateral and transverse directions

$$\frac{1}{k_L^2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{k_T^2}\frac{\partial^2 u}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2} \right) \frac{\partial^2 v}{\partial x \partial y} = \frac{\partial^2 u}{\partial t^2} \qquad (97)$$

$$\frac{1}{k_T^2}\frac{\partial^2 v}{\partial x^2} + \frac{1}{k_L^2}\frac{\partial^2 v}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2} \right) \frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 v}{\partial t^2}, \qquad (98)$$

we use central difference formulas for the partial derivatives in Equation 97 to get the equivalent finite difference equation

$$\frac{1}{h^2 k_L^2}\left[u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n} \right] + \frac{1}{k_T^2 h^2}\left[u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n} \right]$$

$$+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left[\frac{1}{4h^2}\left(v^n_{i+1,j+1}-v^n_{i-1,j+1}-v^n_{i+1,j-1}+v^n_{i-1,j-1}\right)\right]$$

$$=\frac{1}{\Delta t^2}\left[u^{n+1}_{i,j}-2u^n_{i,j}+u^{n-1}_{i,j}\right]. \tag{99}$$

The truncation error for Equation 99 is $O(h^2)$ in space and $O(\Delta t^2)$ in time. Solving for $u^{n+1}_{i,j}$ explicitly in Equation 99 we obtain

$$u^{n+1}_{i,j}=\frac{\Delta t^2}{k_L^2h^2}\left[u^n_{i+1,j}+u^n_{i-1,j}\right]+\frac{\Delta t^2}{k_T^2h^2}\left[u^n_{i,j+1}+u^n_{i,j-1}\right]$$

$$+\frac{\Delta t^2}{4h^2}\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left[v^n_{i+1,j+1}-v^n_{i-1,j+1}-v^n_{i+1,j-1}+v^n_{i-1,j-1}\right]$$

$$+\left[2-\frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2}+\frac{1}{k_T^2}\right)\right]u^n_{i,j}-u^{n-1}_{i,j}. \tag{100}$$

Using the same method for Equation 98 we solve for $v^{n+1}_{i,j}$ to obtain

$$v^{n+1}_{i,j}=\frac{\Delta t^2}{k_T^2h^2}\left(v^n_{i+1,j}+v^n_{i-1,j}\right)+\frac{\Delta t^2}{k_L^2h^2}\left(v^n_{i,j+1}+v^n_{i,j-1}\right)$$

$$+\frac{\Delta t^2}{4h^2}\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(u^n_{i+1,j+1}-u^n_{i-1,j+1}-u^n_{i+1,j-1}+u^n_{i-1,j-1}\right)$$

$$+\left(2-\frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2}+\frac{1}{k_T^2}\right)\right)v^n_{i,j}-v^{n-1}_{i,j}. \tag{101}$$

To ensure the stability of Equations 100 and 101 the Von Neumann stability condition of

$$\Delta t < \frac{h}{\sqrt{\frac{1}{k_T^2} + \frac{1}{k_L^2}}} \qquad (102)$$

must be satisfied.[2]

D. APPLICATION OF THE STRESS-FREE BOUNDARY CONDITIONS TO THE SOLID

The boundary conditions of the two-dimensional domain are broken down into two major categories, those whose normal vector is $\begin{pmatrix}\pm 1\\ 0\end{pmatrix}$, where $\tau_{xx} = \tau_{xy} = 0$, and those whose normal vector is $\begin{pmatrix}0\\ \pm 1\end{pmatrix}$ where $\tau_{yy} = \tau_{xy} = 0$. These are in turn divided into two classes. For $\hat{n} = \begin{pmatrix}\pm 1\\ 0\end{pmatrix}$ they are

a1. The boundary whose unit normal is $\begin{pmatrix}1\\ 0\end{pmatrix}$, that is facing in the positive x-direction, the surface EI in Figure 5 and

a2. The boundary whose unit normal is $\begin{pmatrix}-1\\ 0\end{pmatrix}$, facing in the negative x direction, the surface DH in Figure 5.

Similarly for $\hat{n} = \begin{pmatrix}0\\ \pm 1\end{pmatrix}$

b1. Boundary whose unit normal is $\begin{pmatrix}0\\ 1\end{pmatrix}$, the surfaces AB, GH and IJ in Figure 5 and

b2. Boundary whose unit normal is $\begin{pmatrix}0\\ -1\end{pmatrix}$, the surfaces CD, EF and KL in Figure 5.

The application of the stress-free boundary conditions for cases a1 and b1 is discussed below.

[2] For a brief treatment of the Von Neumann stability criteria see Appendix B.

1. Application of the Stress-free Boundary Condition for the Case of $\hat{n} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

The boundary under investigation is identified in Figure 8 as XY.

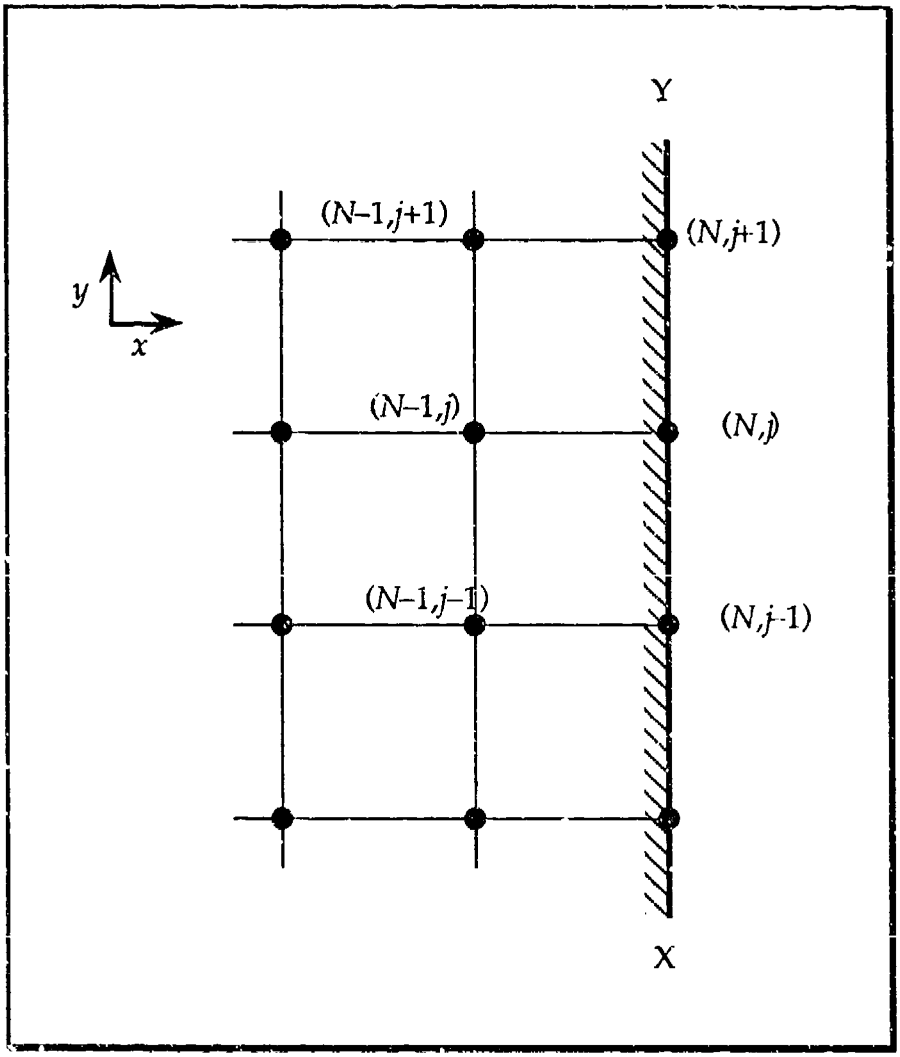


Figure 8. Boundary with Normal $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

The governing equations as derived in Chapter II Section A

$$\tau_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \tag{103}$$

$$\tau_{xx} = \left(\frac{k_T^2}{k_L^2}\right)\frac{\partial u}{\partial x} + \left(\frac{k_T^2}{k_L^2} - 2\right)\frac{\partial v}{\partial y} = 0 \tag{104}$$

$$\frac{1}{k_L^2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{k_T^2}\frac{\partial^2 u}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\frac{\partial^2 v}{\partial x \partial y} = \frac{\partial^2 u}{\partial t^2} \tag{105}$$

$$\frac{1}{k_T^2}\frac{\partial^2 v}{\partial x^2} + \frac{1}{k_L^2}\frac{\partial^2 v}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 v}{\partial t^2} \tag{106}$$

Concentrating on displacements in the lateral (x) direction, a typical boundary point as shown in Figure 8 must satisfy the boundary conditions, Equations 103 and 104 and the governing equations of motion, Equations 105 and 106.

If we apply Equations 105 and 106 at the node (N,j) in Figure 8 we will require values for $u^n_{N+1,j}, v^n_{N+1,j+1}, v^n_{N+1,j-1}, v^n_{N+1,j}, u^n_{N+1,j+1}$ and $u^n_{N+1,j-1}$, which lie outside of the domain and are called pseudonodes. To eliminate this reliance the technique as developed by Ilan and Lowenthal (Ilan, 1976, pp. 431-453) is followed, and is presented here.

The lateral displacement (u) at the node ($N-1,j$) in Figure 8 can be expanded in a Taylor series as

$$u^n_{N-1,j} = u^n_{N,j} - h\left(\frac{\partial u}{\partial x}\right)_{N,j,n} + \frac{h^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{N,j,n} + \begin{array}{c}\text{higher}\\ \text{order}\\ \text{terms}\end{array}, \tag{107}$$

where $u^n_{N-1,j}$ denotes the value of u at the node ($N-1,j$) and at time level n. Alternate expressions for $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$ are given by equations 104 and 105, we have from Equation 104

$$\frac{\partial u}{\partial x} = -\frac{k_L^2}{k_T^2}\left(\frac{k_T^2}{k_L^2} - 2\right)\frac{\partial v}{\partial y} = \left(2\frac{k_L^2}{k_T^2} - 1\right)\frac{\partial v}{\partial y}. \tag{108}$$

From Equation 105,

$$\frac{\partial^2 u}{\partial x^2} = k_L^2\frac{\partial^2 u}{\partial t^2} - \frac{k_L^2}{k_T^2}\frac{\partial^2 u}{\partial y^2} - k_L^2\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\frac{\partial^2 v}{\partial x \partial y} \tag{109}$$

$$= k_L^2\frac{\partial^2 u}{\partial t^2} - \frac{k_L^2}{k_T^2}\frac{\partial^2 u}{\partial y^2} + \left(\frac{k_L^2}{k_T^2} - 1\right)\frac{\partial^2 v}{\partial x \partial y}. \tag{110}$$

Thus the lower order terms of Equation 107 can be written as

$$u_{N-1,j}^n = u_{N,j}^n - h\left(\frac{2k_L^2}{k_T^2} - 1\right)\frac{\partial v}{\partial y} + \frac{h^2}{2}\left(k_L^2\frac{\partial^2 u}{\partial t^2} - \frac{k_L^2}{k_T^2}\frac{\partial^2 u}{\partial y^2} + \left(\frac{k_L^2}{k_T^2} - 1\right)\frac{\partial^2 v}{\partial x \partial y}\right). \tag{111}$$

Using centered differences for $\frac{\partial v}{\partial y}$, $\frac{\partial^2 u}{\partial t^2}$, $\frac{\partial^2 u}{\partial y^2}$ and the following difference formula for the cross derivative term

$$\frac{\partial^2 v}{\partial x \partial y} = \frac{1}{2h^2}\left(v_{N,j+1}^n - v_{N,j-1}^n - v_{N-1,j+1}^n + v_{N-1,j-1}^n\right) \tag{112}$$

the finite difference approximation for Equation 111 is

$$\begin{aligned} u_{N-1,j}^n = u_{N,j}^n &- h\left(\frac{2k_L^2}{k_T^2} - 1\right)\frac{1}{2h}\left(v_{N,j+1}^n - v_{N,j-1}^n\right) + \frac{h^2}{2}\frac{k_L^2}{\Delta t^2}\left(u_{N,j}^{n+1} - 2u_{N,j}^n + u_{N,j}^{n-1}\right) \\ &- \frac{h^2 k_L^2}{2k_T^2}\frac{1}{h^2}\left(u_{N,j+1}^n - 2u_{N,j}^n + u_{N,j-1}^n\right) \\ &+ \frac{h^2}{2}\left(\frac{k_L^2}{k_T^2} - 1\right)\frac{1}{2h^2}\left(v_{N,j+1}^n - v_{N,j-1}^n - v_{N-1,j+1}^n + v_{N-1,j-1}^n\right). \end{aligned} \tag{113}$$

Cancelling common factors Equation 113 reduces to

$$u^n_{N-1,j} = u^n_{N,j} + \left(\frac{1}{2} - \frac{k_L^2}{k_T^2}\right)\left(v^n_{N,j+1} - v^n_{N,j-1}\right) + \frac{h^2 k_L^2}{2\Delta t^2}\left(u^{n+1}_{N,j} - 2u^n_{N,j} + u^{n-1}_{N,j}\right)$$

$$-\frac{k_L^2}{2k_T^2}\left(u^n_{N,j+1} - 2u^n_{N,j} + u^n_{N,j-1}\right) + \left(\frac{k_L^2}{4k_T^2} - \frac{1}{4}\right)\left(v^n_{N,j+1} - v^n_{N,j-1} - v^n_{N-1,j+1} + v^n_{N-1,j-1}\right). \tag{114}$$

Solving for $u^{n+1}_{N,j}$ explicitly we obtain

$$u^{n+1}_{N,j} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u^n_{N,j} - u^{n-1}_{N,j} + \frac{2\Delta t^2}{h^2 k_L^2}u^n_{N-1,j}$$

$$+\frac{\Delta t^2}{h^2 k_T^2}\left(u^n_{N,j+1} + u^n_{N,j-1}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v^n_{N-1,j-1} - v^n_{N-1,j+1}\right)$$

$$-\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(v^n_{N,j+1} - v^n_{N,j-1}\right). \tag{115}$$

The truncation error for Equation 115 is $O(h^3)$ (Ilan, 1976, pp. 431-453). Using the same procedure for the transverse displacement an explicit expression $v^{n+1}_{N,j}$ is given as

$$v^{n+1}_{N,j} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v^n_{N,j} - v^{n-1}_{N,j} + \frac{2\Delta t^2}{h^2 k_T^2}v^n_{N-1,j}$$

$$+\frac{\Delta t^2}{h^2 k_L^2}\left(v^n_{N,j+1} + v^n_{N,j-1}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u^n_{N-1,j-1} - u^n_{N-1,j+1}\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(u^n_{N,j+1} - u^n_{N,j-1}\right). \tag{116}$$

2. Application of the Stress Free Boundary Condition for the Case of $\hat{n} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

In the case of b2, the governing equations

$$\tau_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \tag{117}$$

$$\tau_{yy} = \left(\frac{k_T^2}{k_L^2} - 2 \right) \frac{\partial u}{\partial x} + \frac{k_T^2}{k_L^2} \frac{\partial v}{\partial y} = 0 \tag{118}$$

$$\frac{1}{k_L^2} \frac{\partial^2 u}{\partial x^2} + \frac{1}{k_T^2} \frac{\partial^2 u}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2} \right) \frac{\partial^2 v}{\partial x \partial y} = \frac{\partial^2 u}{\partial t^2} \tag{119}$$

$$\frac{1}{k_T^2} \frac{\partial^2 v}{\partial x^2} + \frac{1}{k_L^2} \frac{\partial^2 v}{\partial y^2} + \left(\frac{1}{k_L^2} - \frac{1}{k_T^2} \right) \frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 v}{\partial t^2} \tag{120}$$

and a typical boundary point is depicted in Figure 9. Expanding in the vertical direction at the node $(i, M+1)$ and at time level n, and ignoring higher order terms

$$u_{i,M+1}^n = u_{i,M}^n + h \left(\frac{\partial u}{\partial y} \right) \bigg|_{i,M,n} + \frac{h^2}{2} \left(\frac{\partial^2 u}{\partial y^2} \right) \bigg|_{i,M,n} \tag{121}$$

using the substitution

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{122}$$

from Equation 117 and

$$\frac{\partial^2 u}{\partial y^2} = k_T^2 \frac{\partial^2 u}{\partial t^2} - \frac{k_T^2}{k_L^2} \frac{\partial^2 u}{\partial x^2} + \left(\frac{k_T^2}{k_L^2} - 1 \right) \frac{\partial^2 v}{\partial x \partial y} \tag{123}$$

from Equation 119, Equation 120 now becomes

$$u_{i,M+1}^{n} = u_{i,M}^{n} + h\left(-\frac{\partial v}{\partial x}\right) + \frac{h^2 k_T^2}{2}\frac{\partial^2 u}{\partial t^2} - \frac{h^2}{2}\frac{k_T^2}{k_L^2}\frac{\partial^2 u}{\partial x^2} + \frac{h^2}{2}\left(\frac{k_T^2}{k_L^2} - 1\right)\frac{\partial^2 v}{\partial x \partial y} \qquad (124)$$

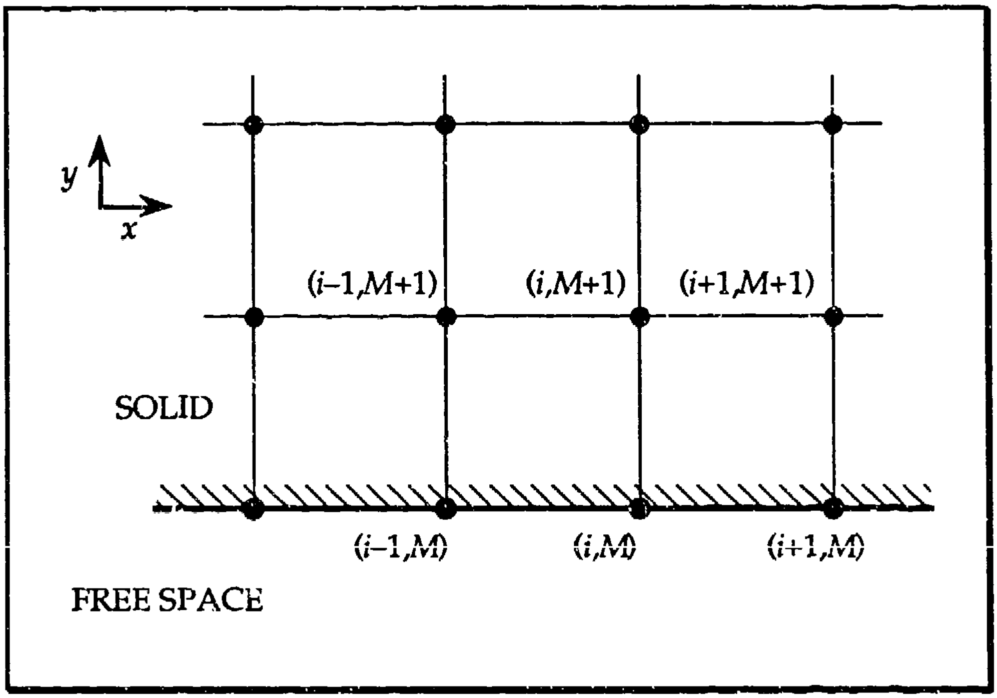


Figure 9. Boundary with Normal $\begin{pmatrix} 0 \\ -1 \end{pmatrix}$

We use central difference formulas for $\frac{\partial v}{\partial x}$, $\frac{\partial^2 u}{\partial t^2}$, $\frac{\partial^2 u}{\partial x^2}$ and for the cross derivative term $\frac{\partial^2 v}{\partial x \partial y}$ the following finite difference formula is used

$$\frac{\partial^2 v}{\partial x \partial y} = \frac{1}{2h^2}\left(v_{i-1,M}^{n} - v_{i+1,M}^{n} - v_{i-1,M+1}^{n} + v_{i+1,M+1}^{n}\right) \qquad (125)$$

The analogous finite difference equation is now

$$u_{i,M+1}^{n} = u_{i,M}^{n} - h\left(\frac{1}{2h}\left(v_{i+1,M}^{n} - v_{i-1,M}^{n}\right)\right) + \frac{h^2 k_T^2}{2\Delta t^2}\left(u_{i,M}^{n+1} - 2u_{i,M}^{n} + u_{i,M}^{n-1}\right)$$

$$-\frac{h^2}{2}\frac{k_T^2}{k_L^2}\frac{1}{h^2}\left(u_{i+1,M}^n - 2u_{i,M}^n + u_{i-1,M}^n\right)$$

$$+\frac{h^2}{2}\left(\frac{k_T^2}{k_L^2}-1\right)\frac{1}{2h^2}\left(v_{i-1,M}^n - v_{i+1,M}^n - v_{i-1,M+1}^n + v_{i+1,M+1}^n\right). \qquad (126)$$

Cancelling common factors yields

$$u_{i,M+1}^n = u_{i,M}^n - \frac{1}{2}\left(v_{i+1,M}^n - v_{i-1,M}^n\right) + \frac{h^2 k_T^2}{2\Delta t^2}\left(u_{i,M}^{n+1} - 2u_{i,M}^n + u_{i,M}^{n-1}\right)$$

$$-\frac{k_T^2}{2k_L^2}\left(u_{i+1,M}^n - 2u_{i,M}^n + u_{i-1,M}^n\right) + \left(\frac{1}{4}-\frac{k_T^2}{4k_L^2}\right)\left(v_{i-1,M}^n + v_{i+1,M}^n - v_{i-1,M+1}^n + v_{i+1,M+1}^n\right)$$

$$(127)$$

and solving for $u_{i,M}^{n+1}$ explicitly, we obtain

$$u_{i,M}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2}+\frac{1}{k_T^2}\right)\right)u_{M,j}^n - u_{M,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_T^2}u_{i,M+1}^n$$

$$+\frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,M}^n + u_{i-1,M}^n\right) - \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(v_{i-1,M+1}^n - v_{i+1,M+1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2}-\frac{3}{k_T^2}\right)\left(v_{i+1,M}^n - v_{i-1,M}^n\right). \qquad (128)$$

Similarly for $v_{i,M}^{n+1}$ we have

$$v_{i,M}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2}+\frac{1}{k_T^2}\right)\right)v_{i,M}^n - v_{i,M}^{n-1} + \frac{2\Delta t^2}{h^2 k_L^2}v_{i,M+1}^n$$

$$+\frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,M}^n + v_{i-1,M}^n\right) - \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(u_{i-1,M+1}^n - u_{i+1,M+1}^n\right)$$

$$-\frac{\Delta t^2}{h^2}\left(\frac{1}{k_L^2}-\frac{3}{k_T^2}\right)\left(u^n_{i+1,M}-u^n_{i-1,M}\right) \tag{129}$$

E. FINITE DIFFERENCE APPROXIMATIONS FOR THE CORNER NODES

The I beam shaped domain has four 270° corners which are identified in Figure 10 as 1, 2, 3 and 4. The treatment of these corners falls into two categories, a) corners 1,3, and b) corners 2,4. Within each category the finite difference formula applied is identical, the difference between them comes in the cross derivative terms of the elastic wave equation. Each category will be discussed below. It is important to note that only the governing equations of motion are applied. The stress free boundary conditions are omitted due to the complexity that arises in trying to apply stress conditions at the corner node. We assume as in Fuyuki and Matsumoto that the consequences of neglecting these boundary conditions is minimal. (Fuyuki, 1980, pp. 2051-2069)

Category a (Corners 1,3): An arbitrary numerical mesh with $\Delta x = \Delta y = h$ about corner 1 is presented in Figure 11. The governing equations of motion are

$$\frac{1}{k_L^2}\frac{\partial^2 u}{\partial x^2}+\frac{1}{k_T^2}\frac{\partial^2 u}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\frac{\partial^2 v}{\partial x \partial y}=\frac{\partial^2 u}{\partial t^2} \tag{130}$$

$$\frac{1}{k_T^2}\frac{\partial^2 v}{\partial x^2}+\frac{1}{k_L^2}\frac{\partial^2 v}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\frac{\partial u}{\partial x \partial y}=\frac{\partial^2 v}{\partial t^2}. \tag{131}$$

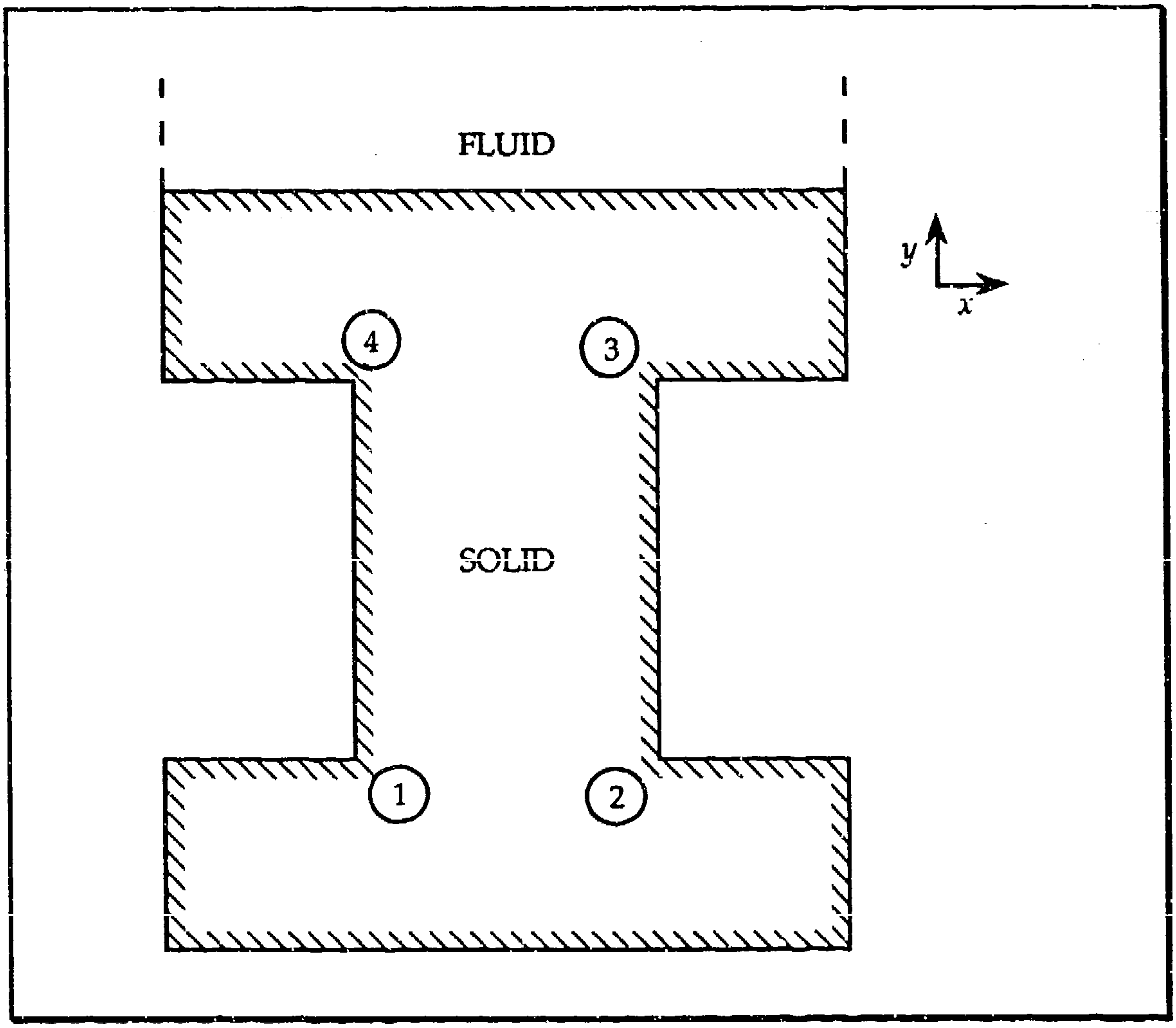


Figure 10. The Corner Nodes

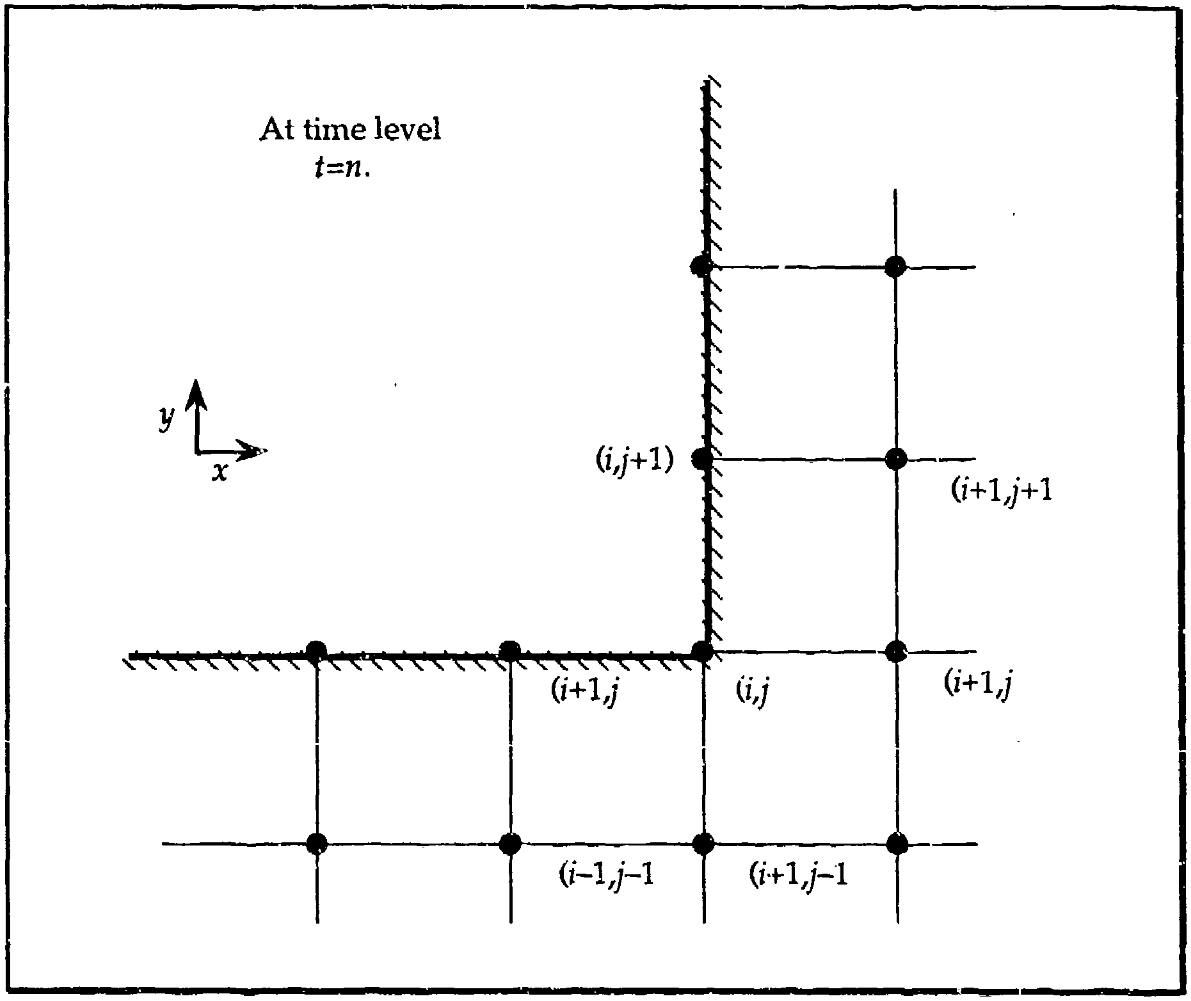


Figure 11. Corner Node—Category *a*

In Equation 124 which describes lateral motion, the normal central difference formulas are used for the $\frac{\partial^2 u}{\partial t^2}$, $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$, terms and for the cross derivative term $\frac{\partial^2 v}{\partial x \partial y}$ the difference formula of Fuyuki and Matsumoto (Fuyuki, 1980, pp. 2051-2069) is applied at the node (*i,j*) which is

$$\frac{\partial^2}{\partial x \partial y} \approx -\frac{1}{2}\left[D_+^x D_+^y + D_-^x D_-^y\right]. \tag{132}$$

Where D_+^x is the forward difference formula

$$D_+^x = \frac{1}{h}\left[u_{i+1,j}^n - u_{i,j}^n\right] \tag{133}$$

and D_-^x is the backward difference formula

$$D_-^x = \frac{1}{h}\left[u_{i,j}^n - u_{i-1,j}^n\right]. \tag{134}$$

The resultant difference formula for the cross derivative term in v at node i,j is

$$\frac{\partial^2 v}{\partial x \partial y} \approx \frac{1}{2h^2}\left[v_{i+1,j}^n + v_{i-1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n - v_{i+1,j+1}^n - v_{i-1,j-1}^n - 2v_{i,j}^n\right] \tag{135}$$

which is $O(h^2)$. The finite difference equivalent of Equation 124 is

$$\frac{1}{k_L^2 h^2}\left[u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n\right] + \frac{1}{h^2 k_T^2}\left[u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n\right]$$

$$+\frac{1}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i+1,j}^n + v_{i-1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n - v_{i+1,j+1}^n - v_{i-1,j-1}^n - 2v_{i,j}^n\right)$$

$$= \frac{1}{\Delta t^2}\left(u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}\right). \tag{136}$$

Solving explicitly for u_{ij}^{n+1} from Equation 130 we obtain

$$u_{i,j}^{n+1} = \frac{\Delta t^2}{k_L^2 h^2}\left(u_{i+1,j}^n + u_{i-1,j}^n\right) + \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^n + \frac{\Delta t^2}{k_T^2 h^2}\left(u_{i,j+1}^n + u_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i+1,j}^n + v_{i-1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n - v_{i+1,j+1}^n - v_{i-1,j-1}^n - 2v_{i,j}^n\right) - u_{i,j}^{n-1} \tag{137}$$

Similarly for Equation 125 an explicit expression for $v_{i,j}^{n+1}$ is

$$v_{i,j}^{n+1} = \frac{\Delta t^2}{k_T^2 h^2}\left(v_{i+1,j}^n + v_{i-1,j}^n\right) + \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^n + \frac{\Delta t^2}{k_L^2 h^2}\left(v_{i,j+1}^n + v_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - u_{i+1,j+1}^n - u_{i-1,j-1}^n - 2u_{i,j}^n\right) - v_{i,j}^{n-1}. \tag{138}$$

For Category b), the governing equations of motion are the same. The finite difference approximation of all terms with the exception of the cross derivative are done in the same manner. For the cross derivative term in the case of Equation 124, the difference formula applied at the node *(i,j)* is

$$\frac{\partial^2}{\partial x \partial y} \approx -\frac{1}{2}\left[D_+^x D_-^y + D_-^x D_+^y\right] \tag{139}$$

Thus $\frac{\partial^2 v}{\partial x \partial y}$ at that node in Figure 12 is represented by

$$\frac{1}{2h^2}\left[2v_{i,j}^n + v_{i+1,j-1}^n + v_{i-1,j+1}^n - v_{i+1,j}^n - v_{i-1,j}^n - v_{i,j+1}^n - v_{i,j-1}^n\right]. \tag{140}$$

The finite difference approximation to Equation 124 is now

$$\frac{1}{h^2 k_L^2}\left(u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n\right) + \frac{1}{h^2 k_T^2}\left(u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n\right)$$

$$+\frac{1}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(2v_{i,j}^n + v_{i+1,j-1}^n + v_{i-1,j+1}^n - v_{i+1,j}^n - v_{i-1,j}^n - v_{i,j+1}^n - v_{i,j-1}^n\right)$$

$$= \frac{1}{\Delta t^2}\left(u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}\right) \tag{141}$$

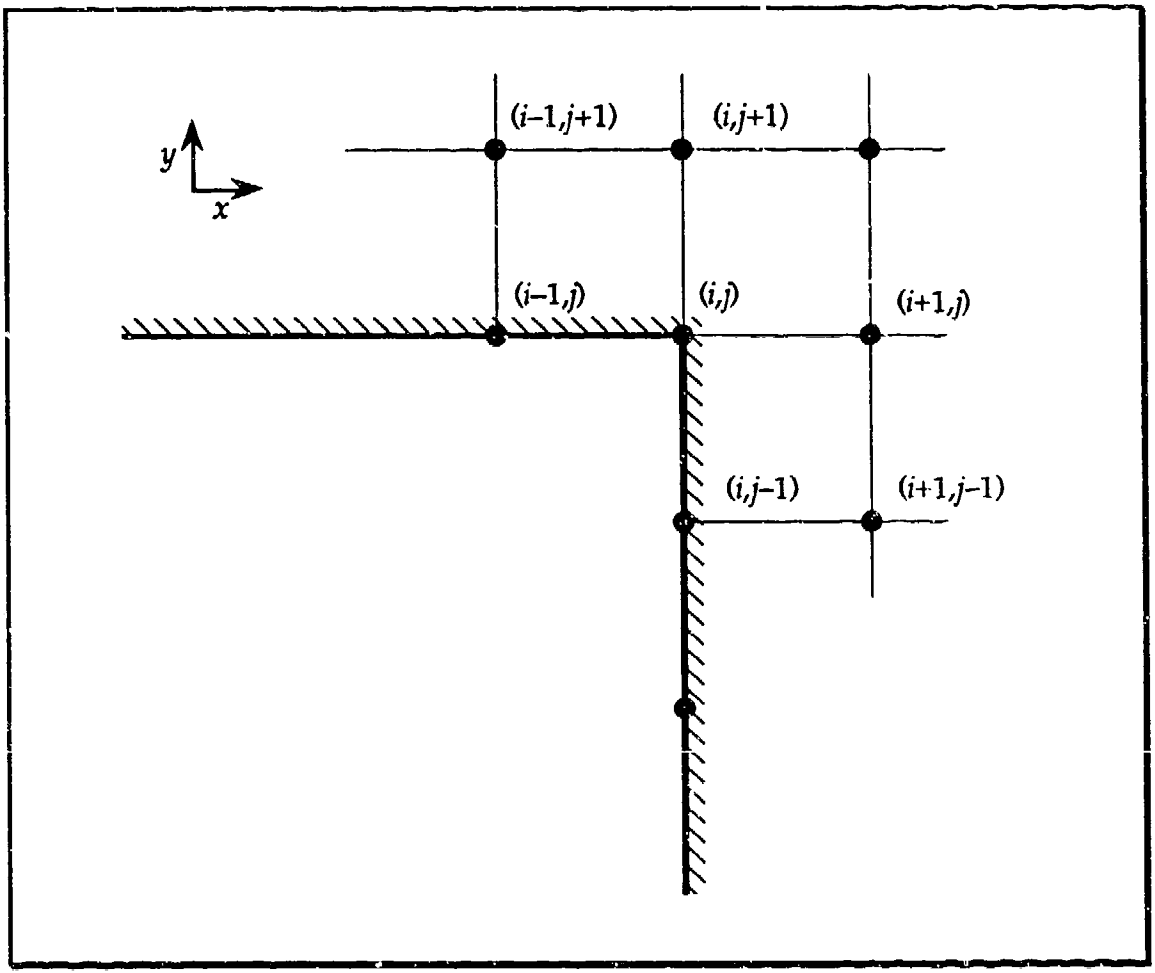


Figure 12. Corner Node—Cateogory *b*

Solving explicitly for $u_{i,j}^{n+1}$ we obtain

$$u_{i,j}^{n+1} = \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,j}^n + u_{i-1,j}^n\right) + \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\right)u_{i,j}^n + \frac{\Delta t^2}{h^2 k_T^2}\left(u_{i,j+1}^n + u_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(2v_{i,j}^n + v_{i+1,j-1}^n + v_{i-1,j+1}^n - v_{i+1,j}^n - v_{i-1,j}^n - v_{i,j+1}^n - v_{i,j-1}^n\right) - u_{i,j}^{n-1}, \tag{142}$$

in the same manner an explicit expression for $v_{i,j}^{n+1}$ is

$$v_{i,j}^{n+1}=\frac{\Delta t^2}{h^2k_T^2}\left(v_{i+1,j}^n+v_{i-1,j}^n\right)+\left(2-\frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2}+\frac{1}{k_T^2}\right)\right)v_{i,j}^n+\frac{\Delta t^2}{h^2k_L^2}\left(v_{i,j+1}^n+v_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\left(2u_{i,j}^n+u_{i+1,j-1}^n+u_{i-1,j+1}^n-u_{i+1,j}^n-u_{i-1,j}^n-u_{i,j+1}^n-u_{i,j-1}^n\right)-v_{i,j}^{n-1} \tag{143}$$

F. BOUNDARY CONDITIONS AT THE FLUID/SOLID INTERFACE

The boundary conditions at the fluid/solid interface are modified by the introduction of a normal plane wave incident on the surface of the solid. As a result the normal component of stress is no longer zero. We must allow for the effect of the scattered pressure at the interface, which is a result of the compliance of the I-beam structure and reflections of displacement waves from internal boundaries. This is done by use of the compatibility condition as derived earlier. Our necessary equations are

$$\tau_{xy}=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{144}$$

$$\tau_{yy}=\left(\frac{k_T^2}{k_L^2}-2\right)\frac{\partial u}{\partial x}+\frac{k_T^2}{k_L^2}\frac{\partial v}{\partial y}=-2\varepsilon k_T^2e^{-it}-\varepsilon k_T^2p^s \tag{145}$$

$$\frac{1}{k_L^2}\frac{\partial^2u}{\partial x^2}+\frac{1}{k_T^2}\frac{\partial^2u}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\frac{\partial^2v}{\partial x\partial y}=\frac{\partial^2u}{\partial t^2} \tag{146}$$

$$\frac{1}{k_T^2}\frac{\partial^2v}{\partial x^2}+\frac{1}{k_L^2}\frac{\partial^2v}{\partial y^2}+\left(\frac{1}{k_L^2}-\frac{1}{k_T^2}\right)\frac{\partial^2u}{\partial x\partial y}=\frac{\partial^2v}{\partial t^2} \tag{147}$$

$$-\frac{\partial p}{\partial y} = \rho_f \frac{\partial^2 v}{\partial t^2} \tag{148}$$

$$k_f^2 \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \tag{149}$$

where $\varepsilon = \rho_f / \rho_s$ and p^s is the scattered pressure along the boundary $y = Kh$ (at the interface, see Figure 13), and at time level n.

Explicit expression for $u_{i,K}^{n+1}$ and $v_{i,K}^{n+1}$ are derived in the same manner as before and they are

$$u_{i,K}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right) u_{i,K}^n - u_{i,K}^{n-1} + \frac{2\Delta t^2}{h^2 k_T^2} u_{i,K-1}^n$$

$$+ \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,K}^n - u_{i-1,K}^n\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i-1,K-1}^n - v_{i+1,K-1}^n\right)$$

$$+ \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(v_{i+1,K}^n - v_{i-1,K}^n\right) \tag{150}$$

$$v_{i,K}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right) v_{i,K}^n - v_{i,K}^{n-1} + \frac{2\Delta t^2}{h^2 k_L^2} v_{i,K-1}^n$$

$$+ \frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,K}^n - v_{i-1,K}^n\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i-1,K-1}^n - u_{i+1,K-1}^n\right)$$

$$- \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(u_{i+1,K}^n - u_{i-1,K}^n\right) - \frac{2\Delta t^2 \varepsilon}{h}\left(2e^{-it} + p_{i,K}^n\right). \tag{151}$$

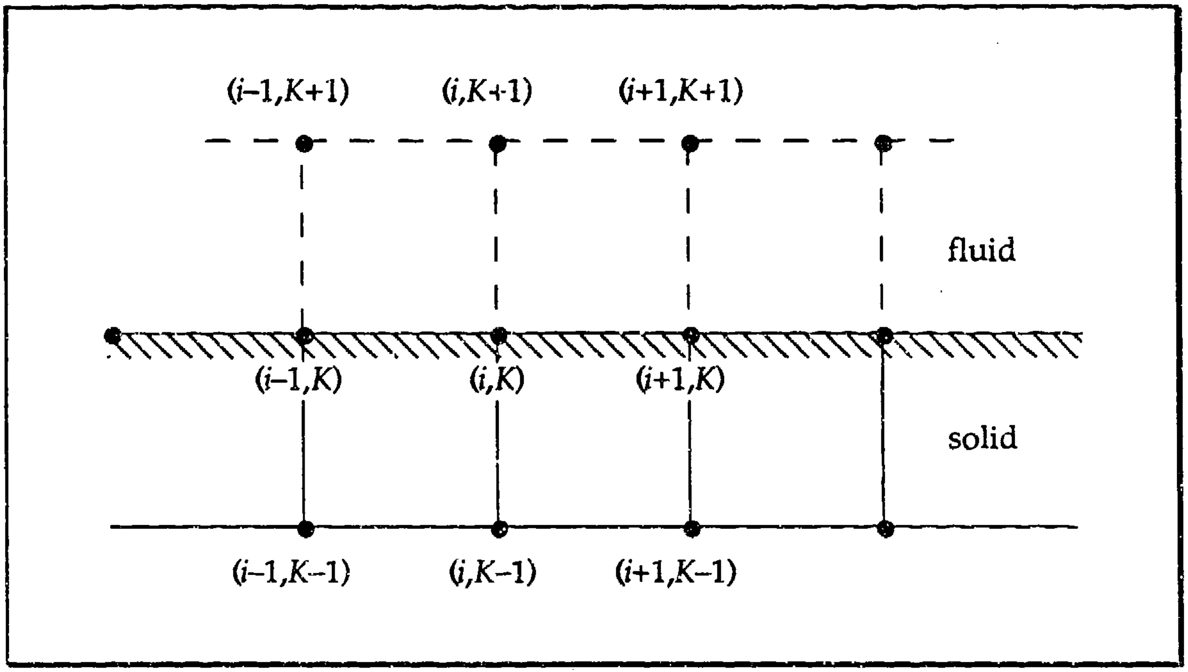


Figure 13. Fluid-Solid Interface

Looking at the compatibility condition, Equation 142 we use central difference approximations to obtain

$$\frac{p^n_{i,K+1} - p^n_{i,K-1}}{2h} = -\frac{1}{\Delta t^2}\left(v^{n+1}_{i,K} - 2v^n_{i,K} + v^{n-1}_{i,K}\right) \tag{152}$$

assuming that it is applied at the boundary $y = Kh$ in Figure 13.

Solving for $p^n_{i,K-1}$ we obtain

$$p^n_{i,K-1} = \frac{2h}{\Delta t^2}\left(v^{n+1}_{i,K} - 2v^n_{i,K} + v^{n-1}_{i,K}\right) + p^n_{i,K+1} \tag{153}$$

all quantities on the right hand side are known. We now have a method of calculating $p^n_{i,K-1}$ which is required when applying Equation 149 at the boundary $y = Kh$.

Solving for the p^{n+1} term of Equation 149 we have

$$p_{i,K}^{n+1} = \left(2 - \frac{4\Delta t^2}{k_f^2 h^2}\right) p_{i,K}^n + \frac{\Delta t^2}{k_f^2 h^2}\left(p_{i-1,K}^n + p_{i+1,K}^n + p_{i,K+1}^n + p_{i,K-1}^n\right) - p_{i,K}^{n-1}. \quad (15.4)$$

The components of the right-hand side with the exception of $p_{i,K-1}^n$ lie in the interior or on the boundary of the fluid domain and can be evaluated, however $p_{i,K-1}^n$ is essentially a pseudonode for the fluid and is determined from Equation 153 above. By this method we have now generated the scattered pressure waves caused by the vibration of the solid at the fluid/solid surface.

G. PROGRAMMING CONSIDERATIONS

The program for this thesis was written entirely in *Matlab 4.0 Beta version* for two reasons,

- Ease of programming
- The ability to generate quality graphics.

Although originally intended as a linear algebra toolkit the above features have caused it to be used more and more as a high level programming language. In our case Matlab was convenient since the fluid and solid domains are square matrices, which are easily manipulated in Matlab. Updating values is done somewhat differently than in FORTRAN or C and is discussed below.

Equation 66 of this section updates the interior points of the fluid domain and is given by

$$p_{i,j}^{n+1} = 2\left(1 - 2\rho^2\right)p_{i,j}^n + \rho^2\left[p_{i+1,j}^n + p_{i-1,j}^n + p_{i,j+1}^n + p_{i,j-1}^n\right] - p_{i,j}^{n-1},$$

an equivalent FORTRAN statement might look like

```
      DO 10 I = 2,K
       DO 20 J = 2,K
        P(I,J,N+1) = 2.0*(1.0-2.0*(RHO**2))*P(I,J,N)
&        +(RHO**2)*(P(I-1,J,N)+P(I+1,J,N)+P(I,J-1,N)+P(I,J+1,N))
&       -P(I,J,N-1))
:
:
  20   CONTINUE
  10   CONTINUE
```

Here each value is updated individually by using a double DO loop. In Matlab this is done by "shifting" a grid the size of the interior around the appropriate matrix and weighting terms. The equivalent code in MATLAB would be

```
PNEW(2:K,2:K) = 2*(1-2*RHO^2)*PCURR(2:K,2:K)
+(RHO^2)*(PCURR(1:K-1,2:K)+PCURR(3:K+1,2:K)+PCURR(2:K,1:K-1)
+PCURR(2:K,3:K+1))-POLD(2:K,2:K).
```

where PNEW contains the new values, PCURR the current values and so on. All updating is done in this manner eliminating the requirement for multiple do loops.

IV. NUMERICAL RESULTS

In an effort to verify our code we checked the behavior of the fluid and employed energy conservation methods to check for the consistency of the coupled domain. For the fluid waveguide we want to ensure that the propagating modes behave as expected, that is, they should not reflect from the artificial boundary. This was done by placing a driving force of the form

$$p = A_n e^{i(\beta_n y + \gamma_n x - t)} \tag{155}$$

where $\beta_n = \sqrt{k_f^2 - \gamma_n^2}$ and $\gamma_n = n\pi$ and $k_f = \frac{\omega a}{c_f}$, (Ref. Equation 24, Chapter 1, Section A) at the boundary $y = 0$, which excited the fundamental, first and second modes ($n = 0, 1, 2$). The coefficients A_n had value 1 for all n. The amplitude of each mode was measured at the boundary. As can be seen from Figures 14 a, b, and c the fundamental and first mode approach the value 1 with the second mode showing the same behavior but at a much slower rate.

To check that the coupling of the two domains was working correctly we eliminated the cavities of the I-beam which reduced our problem to one which could be solved analytically. For a normally incident plane wave only the fundamental mode was excited. Since the incident wave displaces the solid only in the y direction v has no x dependence and u is identically zero. The values of A_0 and v are given by (Scandrett, 1992, interview),

$$A_0 = c_1 \left[\frac{-i\left(1 + e^{-4ik_L}\right)}{k_f} \right] \quad \text{and} \quad v = e^{-it}\left(c_1 e^{ik_L y} + c_2 e^{-ik_L y}\right).$$

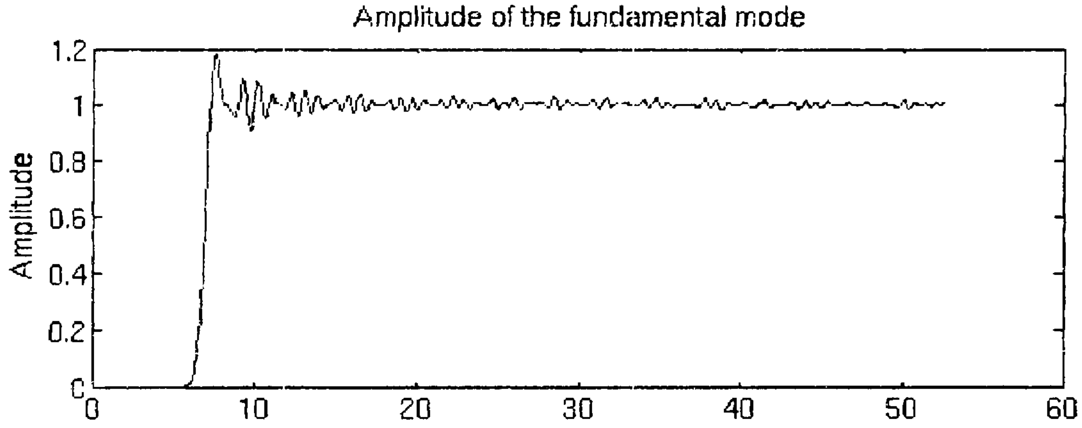


Fig 14a

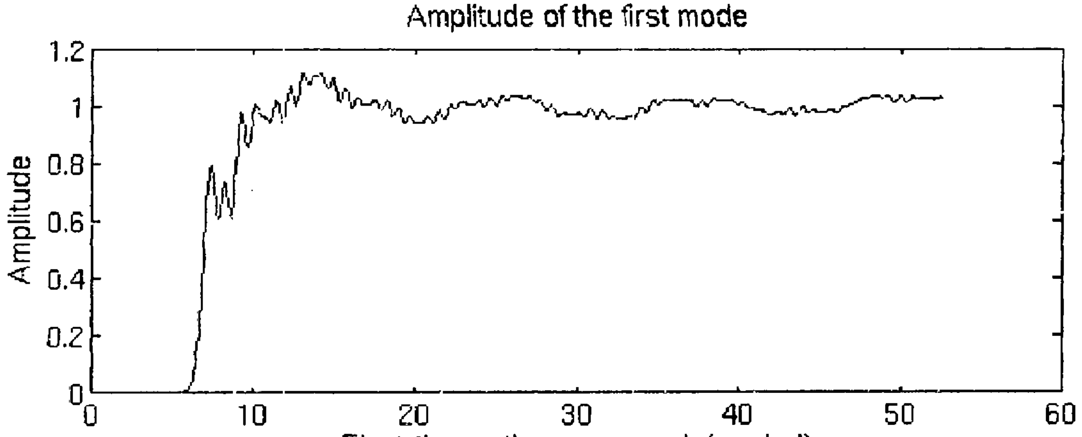


Fig 14b

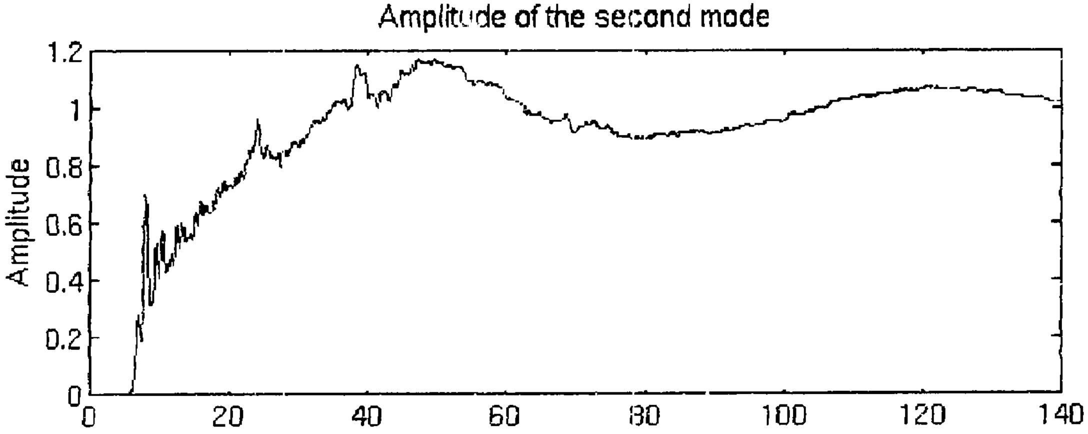


Fig 14c

For $k_f = 1$ and $k_L = 0.2542$, c_1 and c_2 were given by $0.0625 + 0.03i$ and $0.0585 - 0.0374i$ respectively. In this case the magnitude of $\|A_0\|$ of 0.1211 compared favorably with the numerical solution of 0.1255. A major discrepancy lay in the numerical and analytical values of v (transverse displacement). The average absolute value for the numerical solution was 13.12 while the average absolute value for the analytical solution was 0.0965. They exhibited similar behavior but the numerical solution was translated by a constant term. We believe this to be a result of there being no displacement term to compensate for the effect of imposing a periodic pressure instantaneously in time which has caused our solid domain to drift or displace, violating one of our initial assumptions (see Chapter I, Section A). To nullify this effect we take the time derivative of our steady state solution for v and then compare with our analytic value as can be seen in Figure 15. Again the numerical and analytical solutions give close agreement. A second check was to apply energy conservation methods to our steady state solution, from which it can be shown (Scandrett, 1992) that the propagating modes must satisfy

$$\sum_{n=-M}^{M} \beta_n \|A_n\|^2 = -2\beta_0 \operatorname{Re}(A_0) = -\frac{1}{2} \operatorname{Im} \int_{-1}^{1} p\Big|_{y=0} \bar{v}\Big|_{y=0} dx. \tag{156}$$

The quantities $\sum_{n=-M}^{M} \beta_n \|A_n\|^2$ and $-2\beta_0 \operatorname{Re}(A_0)$ for the various values of k_f are listed in Table 1. Considerable discrepancies exist throughout, which may indicate either an error in the code or the inability of our simulation to model the high frequency components which exist at the interface. Another factor which may contribute to the failure of the integral is the translation of v which was discussed previously. With this in mind we must possibly consider the following results as being inaccurate.

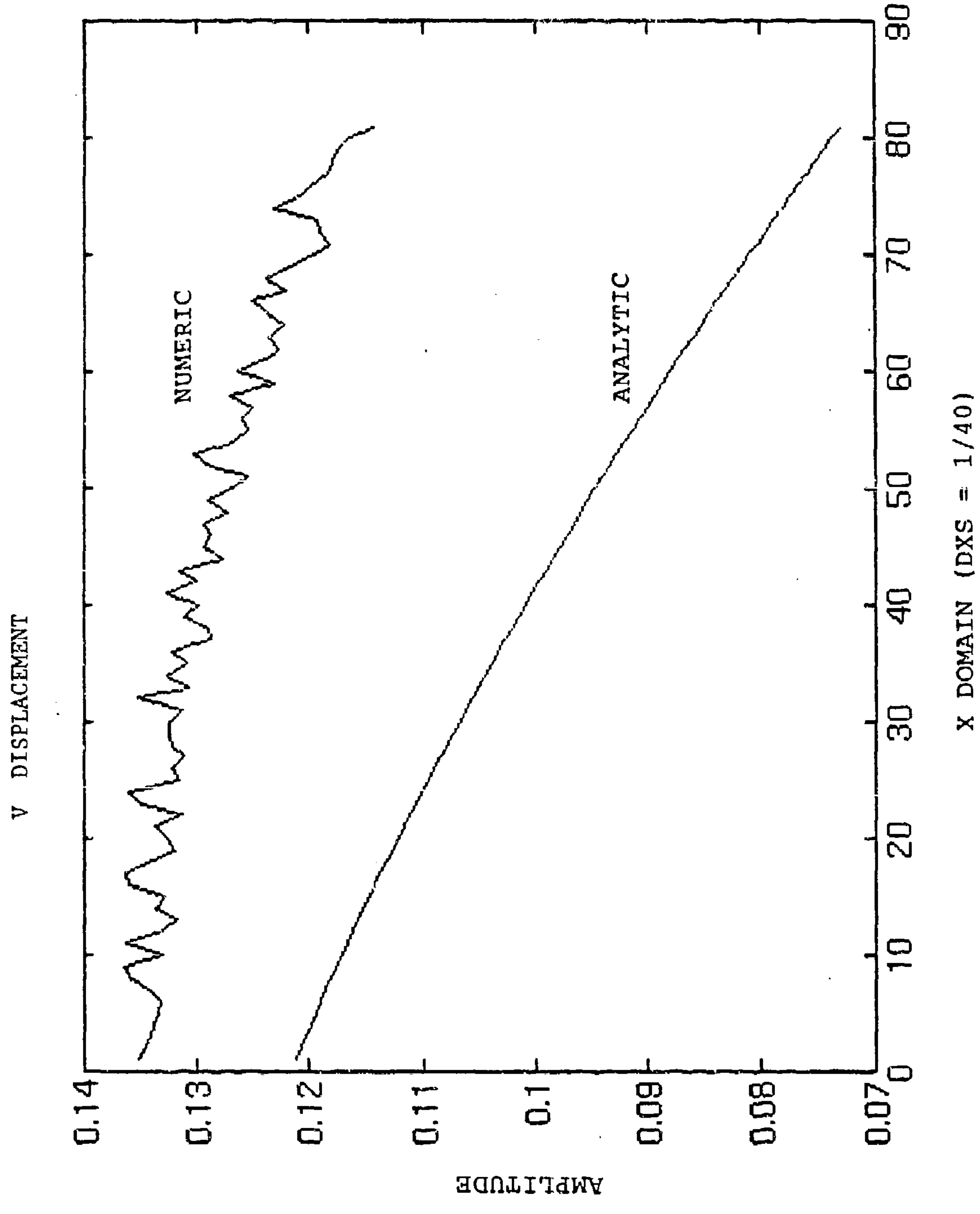

V DISPLACEMENT
NUMERIC
ANALYTIC
AMPLITUDE
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0
10
20
30
40
50
60
70
80
90
X DOMAIN (DXS = 1/40)

Fig 15

The first aspect we look at is the amplitude of the propagating modes of the scattered pressure. For $k_f = 7$, (this we call our original system) the time history of the amplitudes for the fundamental, first and second modes are shown in Figures 16, 17, and 18 respectively. These indicate roughly the rate of convergence of the numerical method.

TABLE 1. ENERGY CONSIDERATIONS

k_f	$-\frac{1}{2}\mathrm{Im}\left(\int_{-1}^{1} p\big\vert_{y=0}\,\overline{v}\big\vert_{y=0}\,dx\right)$	$\sum_{n=-M}^{M}\beta_n\Vert A_n\Vert^2$	$-2\beta_0\,\mathrm{Re}(A_0)$
0.5065	–0.1735	0.022	0.0003
1	0.475	0.0279	–0.2138
1.2516	0.12965	0.0006	0.055
2.026	40.5886	2.135	3.782
3.3446	–0.6093	0.5380	–1.3844
4	2.6288	0.2179	–0.6506
4.5585	7.6195	0.2967	0.5685
6.5572	-5.72	0.0835	–0.4053
7	4.0212	0.0644	0.0667
7.5	0.18715	0.0846	0.6723
8	4.13916	0.5225	0.2738
9	4.8779	0.0742	–1.5974

To gain insight into the characteristics of the solid, we treated the flange at the fluid-solid interface as a thin plate and tried different values of k_f corresponding to the resonant frequencies of a thin plate with prescribed boundary conditions. The first case was a plate with the fixed (referred to as FF) boundary conditions given by

$$v(-1) = v(1) = v''(-1) = v''(1) = 0.$$

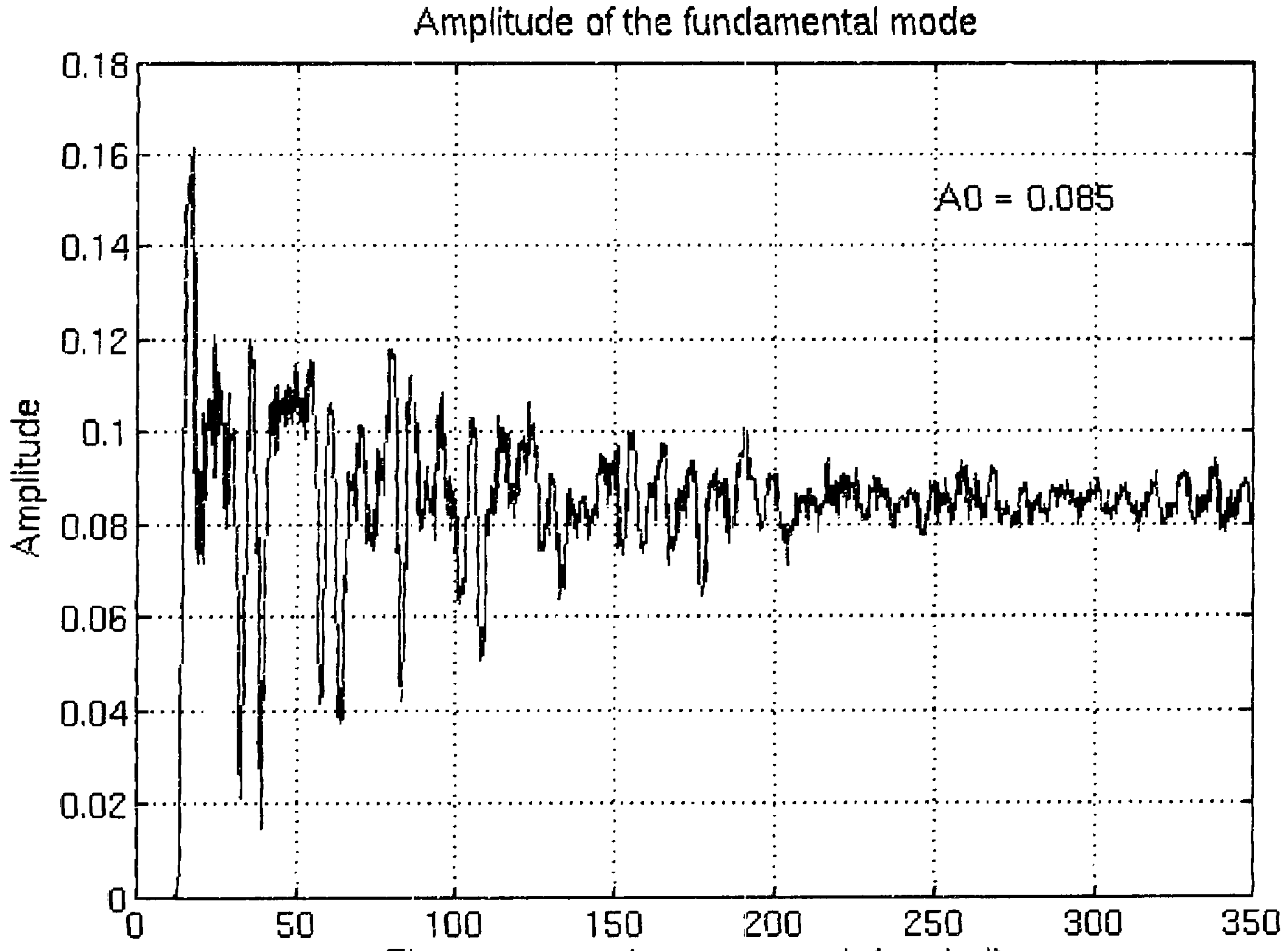

Amplitude of the fundamental mode
A0 = 0.085
Amplitude
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
250
300
350
time – seconds(scaled)

Fig 16

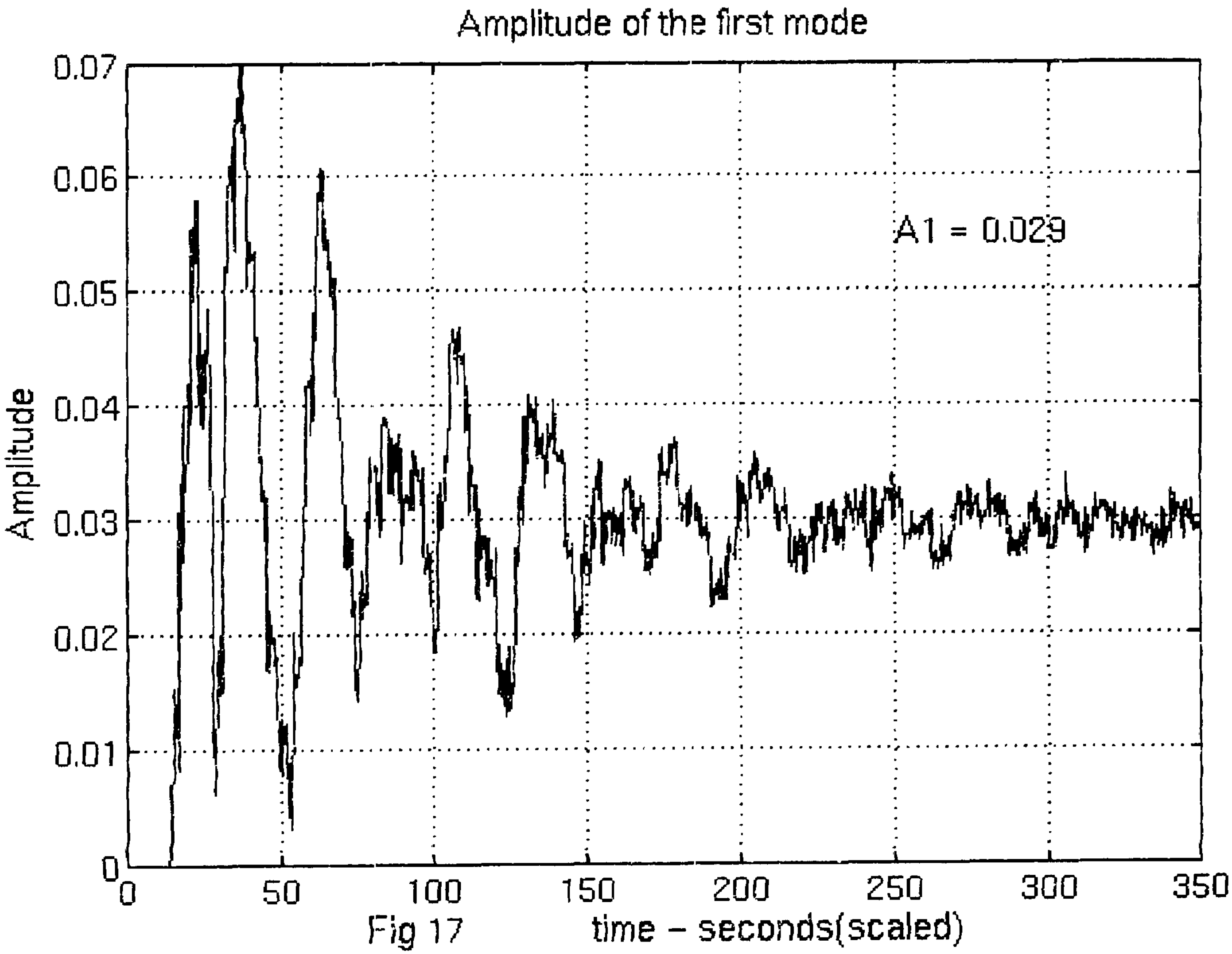

Amplitude of the first mode
A1 = 0.029
Amplitude
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
200
250
300
350
time - seconds(scaled)

Fig 17

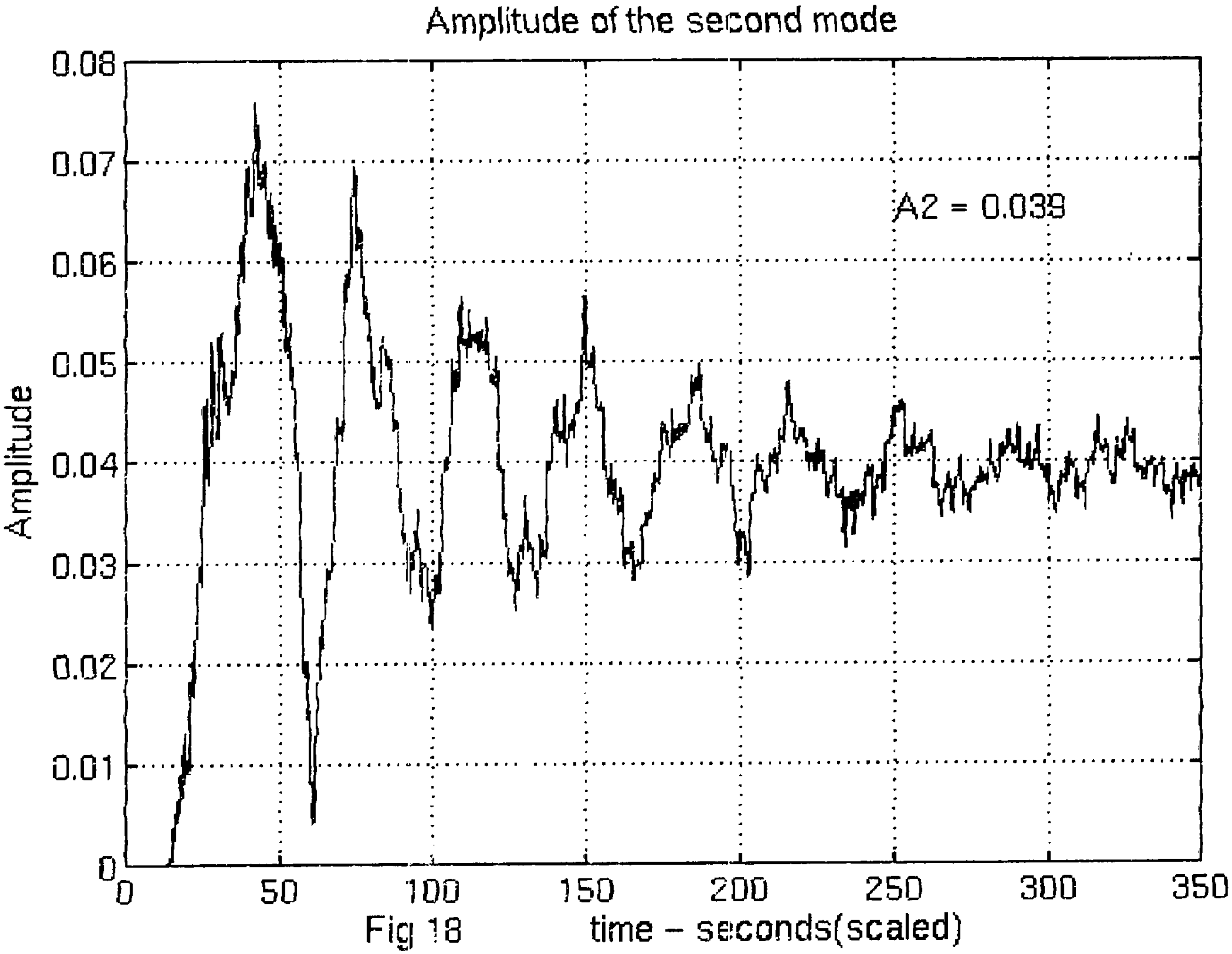

Amplitude of the second mode
A2 = 0.039
Amplitude
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
200
250
300
350
time – seconds(scaled)

Fig 18

The resonant frequencies are

$$\omega_n = \frac{c_T h n^2 \pi^2}{4a\sqrt{6(1-\upsilon)}}, \tag{157}$$

(Lalanne, 1982, p. 103) where υ is Poisson's ratio, h is the thickness of the plate, c_T is the transverse velocity of the solid and a is a scaling constant for distance.

For the second case, clamped (referred to as CC) boundary conditions were prescribed, which are given by

$$v(-1) = v(1) = v'(-1) = v'(1) = 0.$$

The resonant frequencies of this system are given by

$$\omega_n = \frac{c_T h X_n^2}{4a\sqrt{6(1-\upsilon)}} \tag{158}$$

where $X_1^2 = 22.37$, $X_2^2 = 61.67$ and $X_3^2 = 120.9$. (Lalanne, 1982, p. 103)

The purpose of this experiment is to determine whether the I-beam exhibits behavior similar to a clamped or fixed plate, since it would be advantageous (numerically) if we could substitute a periodically placed boundary condition for the remaining portion (lower flange and center spar) of the I-beam rather than having to calculate finite difference approximations for the entire I-beam.

We do this by plotting the amplitudes versus the corresponding values of k_f for the fundamental, first, and second modes. The presence of any peaks would indicate a resonant type behavior. If any of these peaks corresponds to a particular value of k_f for a clamped or fixed plate we say that for that value of k_f (and hence ω), the I-beam behaves in a manner similar to a plate with clamped or fixed boundary conditions. This gives us a possible range for k_f values on which to concentrate when looking for resonant frequencies.

The values of k_f are given in Table 2. The quantities in parentheses are the modes that propagated for a particular value of k_f which is determined by the maximum integer value n can take such that β_n is real (Ref. Equation 71).

TABLE 2. k_f VALUES

	SYSTEM		
CATEGORY	ORIGINAL	C–C	F–F
1	1 (0)	1.2516 (0)	0.5265 (0)
2	4 (0, ±1)	3.3446 (0, ±1)	2.026 (0)
3	7 (0, ±1, ±2)	6.5572 (0, ±1, ±2)	4.558 (0, ±1)

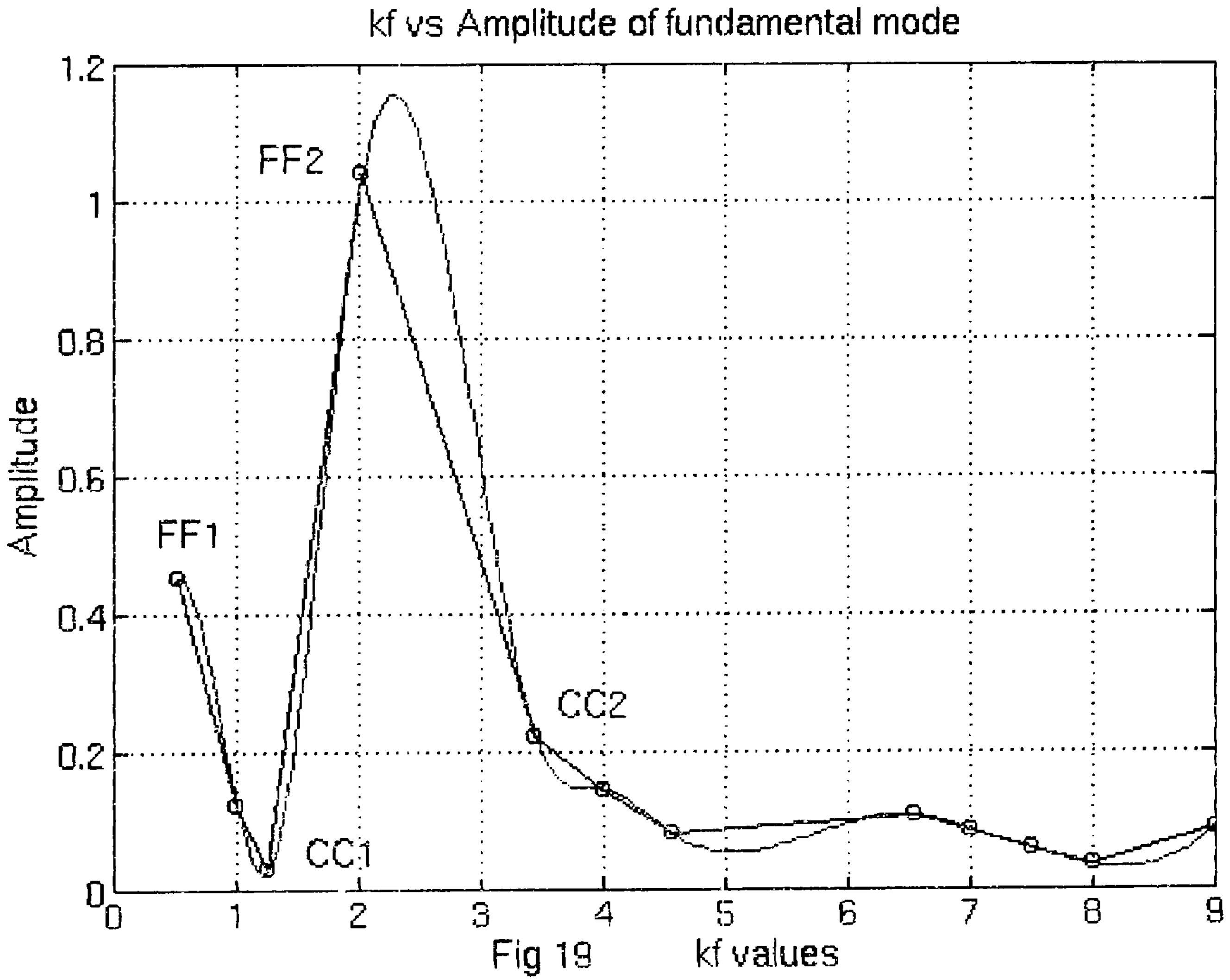
We divide our values of k_f into three categories (for identification purposes) In the first category we include the values of k_f corresponding to the first resonant frequency of the clamped and fixed plates and the values of k_f for our original system which allows only the fundamental mode to propagate. In the second category are values of k_f that correspond to the second resonant frequency of the clamped and fixed plates as well as the value of k_f which allow only the fundamental and first modes to propagate. The third category corresponds to the third resonance of the clamped and fixed plates and the first three modes in our original system. Also included in this final category are the values of $k_f = 7.5$, 8 and 9 which will allow us to study the behavior of the second mode at higher frequencies in greater detail.

In Figure 19 we show a plot of k_f values versus the amplitude of the fundamental mode. Computed values are shown in black. A cubic spline of the points involved is shown in red. We must note however that this spline is only a possible representation of the behavior of the amplitude for different values of k_f, since time constraints did not allow us to conduct a more comprehensive set of simulations from which we could obtain an accurate picture of the behavior.

It can be seen that for $k_f = 0.5065$ and 2.026 (the points labeled FF1 and FF2, the I-beam is exhibiting a resonant type behavior. These values correspond to the first and second resonant frequencies of a fixed plate. It can also be seen that for values of k_f greater than 3.5 there is little variation in the amplitude of the fundamental model since we are now past the first cutoff value (after which three modes propagate) of $k_f = \pi$. This leads us to believe that at these higher frequencies most of the activity takes place in the higher modes. However the total energy calculation shifts from fundamental to first and back to fundamental as can be seen in Table 3.

In Figure 20 the only resonant behavior is exhibited for the value $k_f = 3.446$ (the point labelled CC2). This frequency is just past the first cutoff value (π) and the amplitude of the first mode is double that of the fundamental (compared with the corresponding value in Figure 19) which confirms that at higher frequencies energy is being propagated in the higher modes.

In Figure 21 we plot the amplitude of the second mode against k_f. The only resonant type behavior which exists here is for the $k_f = 8$, but since the amplitudes involved are so small it would be difficult to draw an accurate



kf vs Amplitude of fundamental mode
Amplitude
kf values
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
FF1
FF2
CC1
CC2

Fig 19

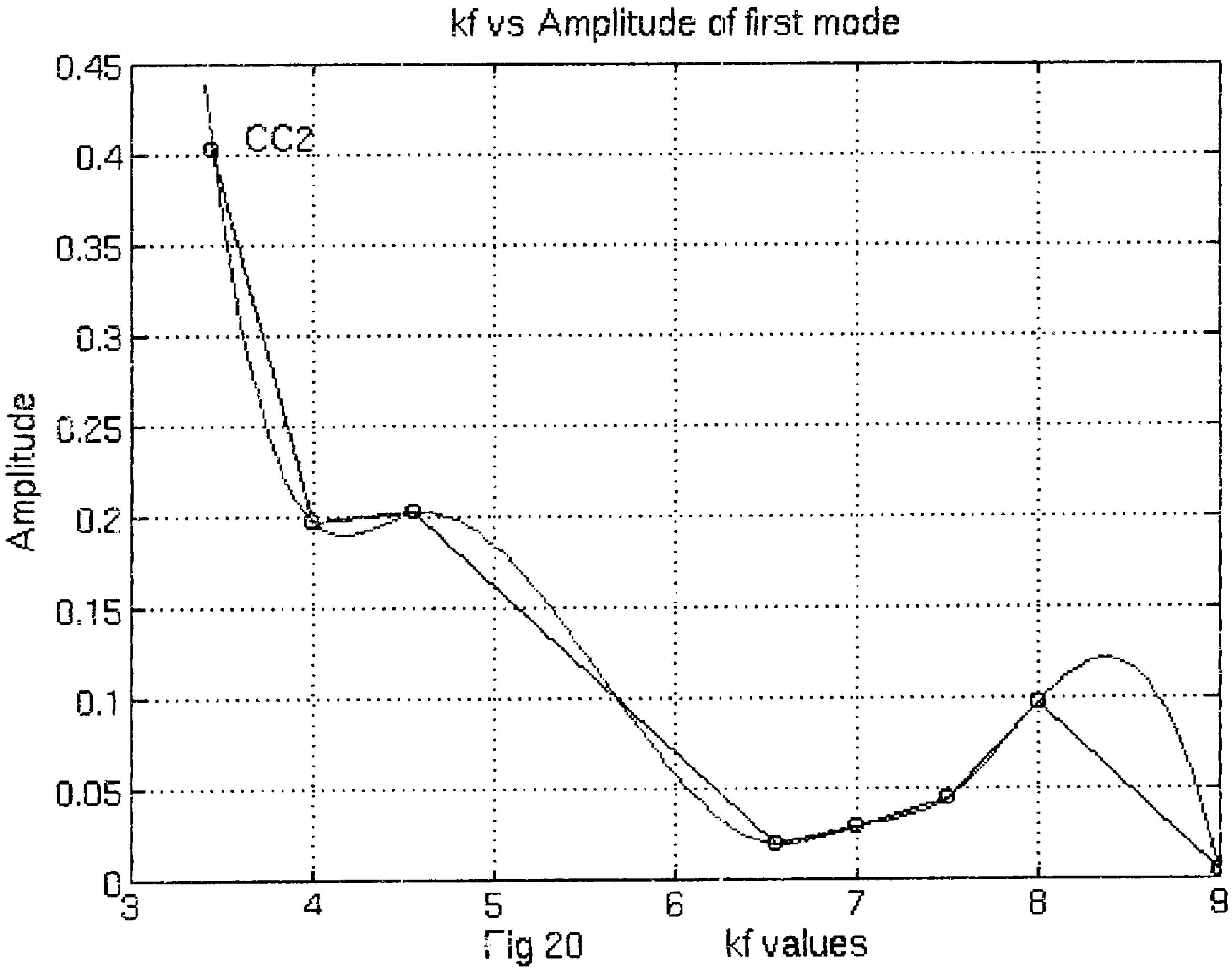

kf vs Amplitude of first mode
Amplitude
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
3
4
5
6
7
8
9
kf values
CC2

Fig 20

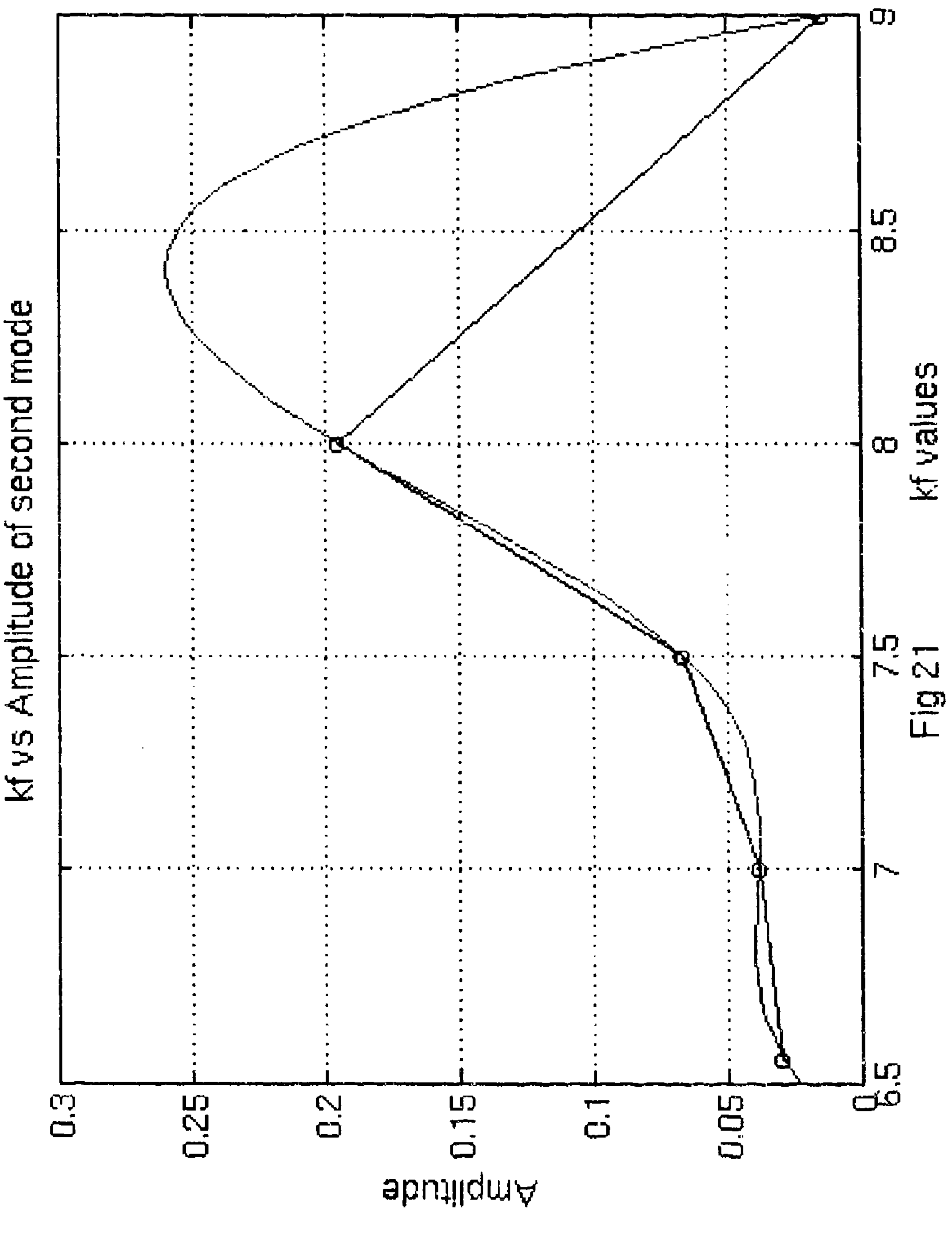

kf vs Amplitude of second mode
Amplitude
0
0.05
0.1
0.15
0.2
0.25
0.3
6.5
7
7.5
8
8.5
9
kf values

Fig 21

conclusion about the existence of a resonant frequency. Another aspect we investigate is the proportion of the total energy carried by each propagating mode which is given by the formula

$$E_n = \frac{\|A_n\|^2}{\sum_{p=-M}^{M} \|A_p\|^2} \tag{159}$$

where E_n is the energy of the n^{th} mode and A_n the amplitude of the n^{th} mode (Kinsler, 1982, p. 110). These quantities are summarized in Table 3.

TABLE 3. ENERGY CONSIDERATIONS

	$E_n(\%)$				
k_f	E_0	E_1	E_2	E_{-1}	E_{-2}
0.5056	100	N/A	N/A	N/A	N/A
1	100	N/A	N/A	N/A	N/A
1.2516	100	N/A	N/A	N/A	N/A
2.026	100	N/A	N/A	N/A	N/A
3.3446	13.04	43.47	N/A	43.47	N/A
4	13.36	40.82	N/A	40.82	N/A
4.5535	7.47	46.26	N/A	46.26	N/A
6.5572	82.43	2.29	6.5	2.29	6.5
7	58.23	8.08	12.81	8.08	12.81

We can see that the proportion of energy carried by the fundamental mode varies from 100% to 12.04% as k_f goes from 2.026 to 3.3446. This can be accounted for as a redistribution of energy that takes place as the frequency of the waves passing through the fluid wave guide pass the first cutoff value of $k_f = \pi$. Similar arguments hold for higher values of k_f.

Finally we take a brief look at the shear strain field generated in the solid. We do this for two reasons, one to check the behavior at the corner nodes and two, to find out where the maximum shear strain occurs.

When treating the corner points we neglected applying the stress free boundary conditions there assuming this would have no adverse effect on the behavior of the solid. Were this assumption invalid we would expect to observe singularities or other irregular behavior. Provided in Figures 22, 23 and 24 are snapshots of the solid at different time levels which display no unusual behavior at the corner nodes, leading us to accept the assumption as valid.

From an engineering standpoint we are interested in where the maximum shear strain occurs for possible failure analysis. As can be seen from Figure 25 the maximum occurs along the transverse borders of the I-beam cavity. We must note that the amplitudes and frequencies of the acoustic pressure waves involved in our study are not comparable to those which would cause permanent deformation or failure, but which could result in fatigue cracking if subjected to prolonged periods of stress.

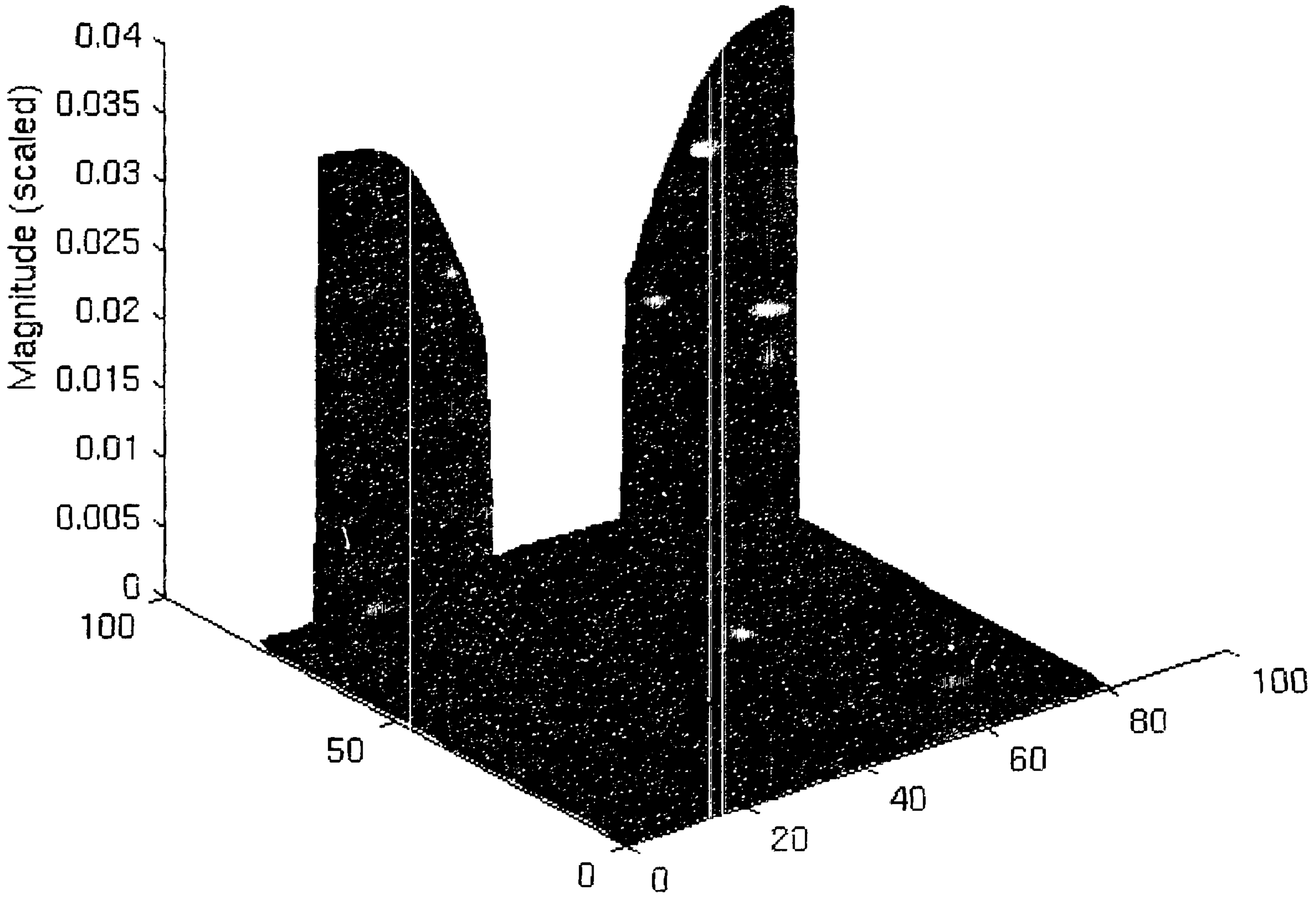


Fig 22 Shear Strain at time t = 3s(scaled)

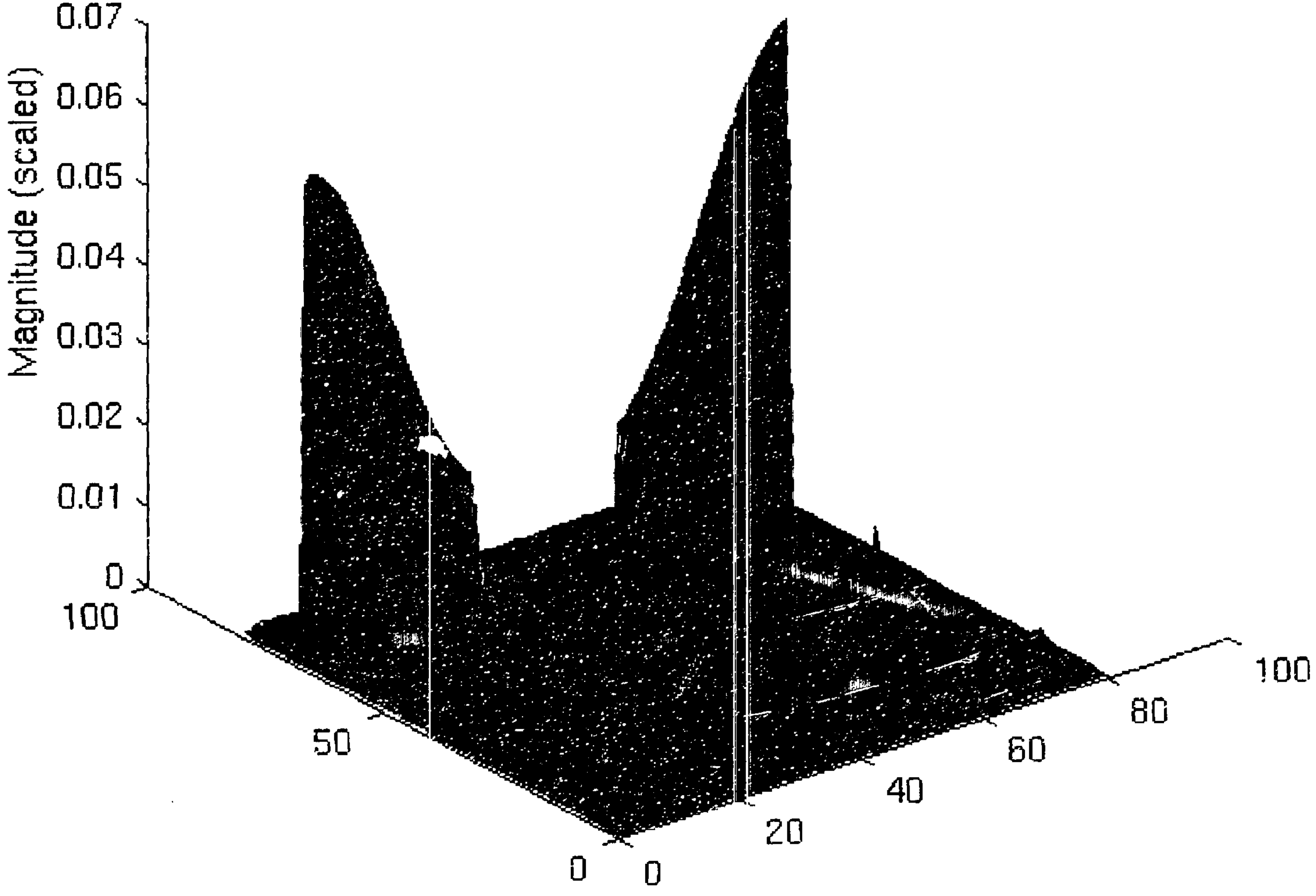


Fig 23 Shear Strain at time t = 6s(scaled)

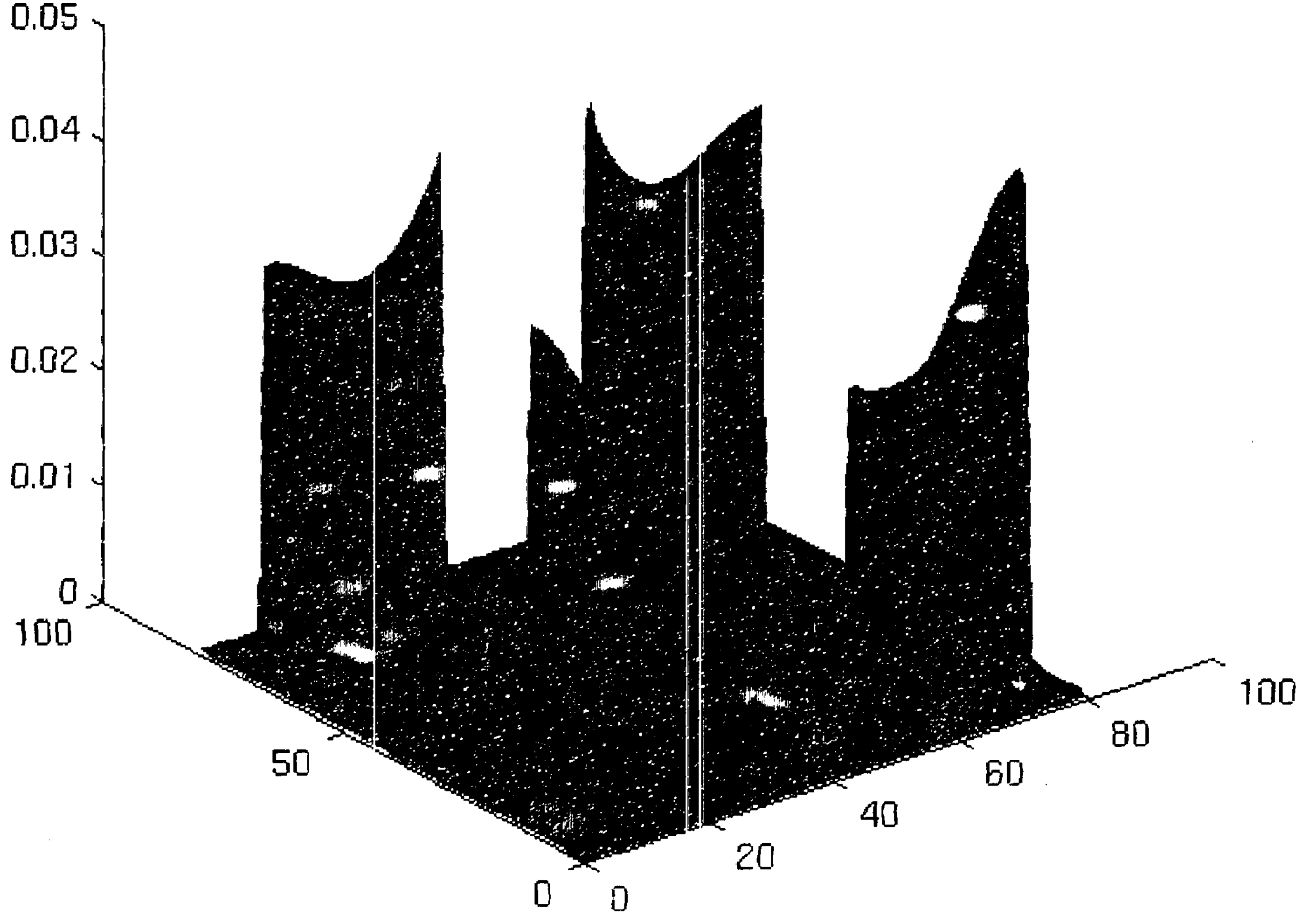


Fig 24 Shear Strain at time t = 12s(scaled)

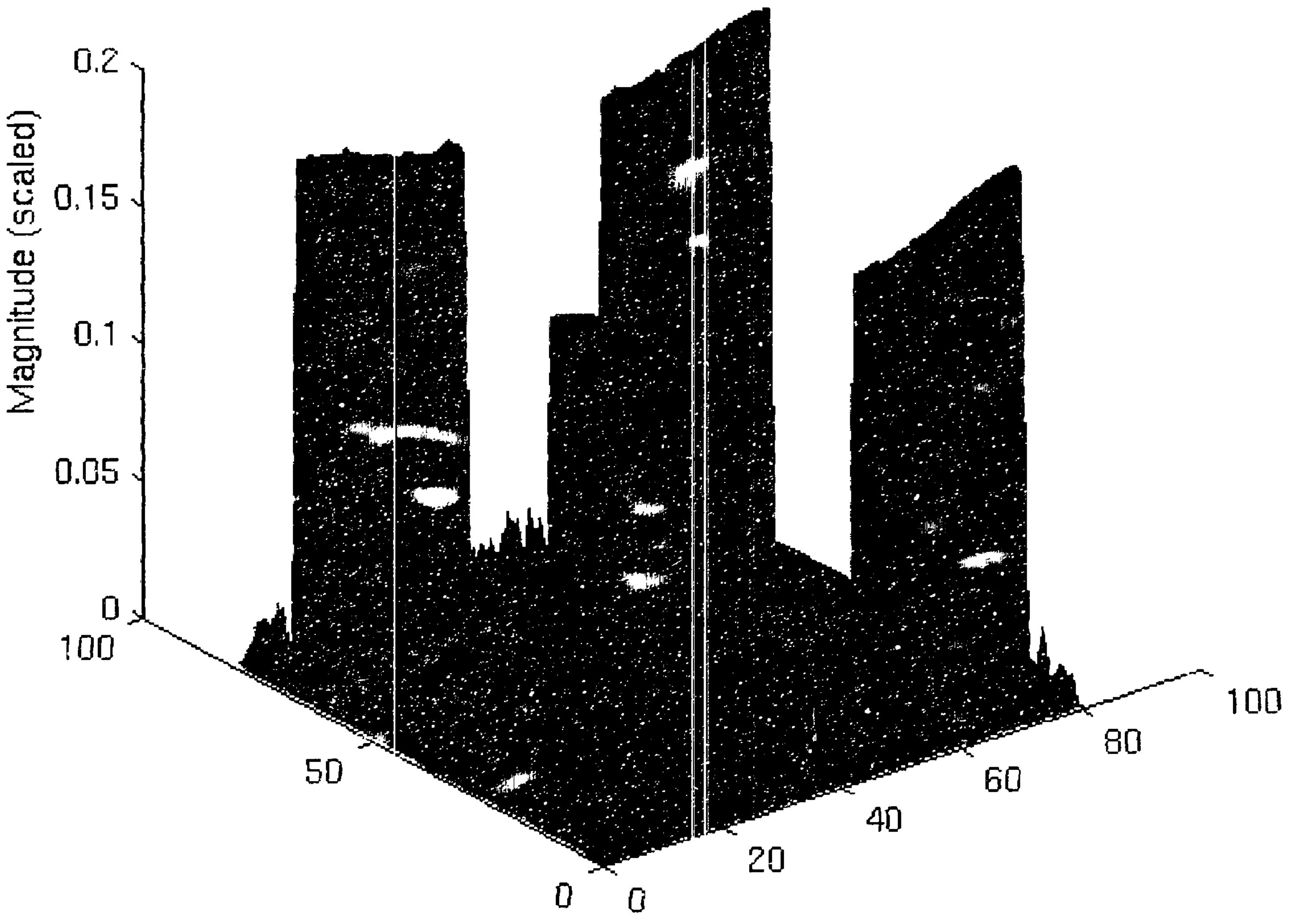


Fig 25 Shear Strain at steady state

V. CONCLUSIONS

Several problems arose in implementing the numerical scheme. It was found that the non-local radiation boundary condition did not completely annihilate the initial wavefront and a small amount of reflection took place, but over time the effect of this reflection became negligible. Evanescent modes did not decay sufficiently and were reflected at the boundary adding to the overall noise of the problem.

When calculating the amplitudes of the propagating modes at steady state we applied the formula

$$A_k = \int_{-1}^{1} p\Big|_{y=2} e^{ik\pi x} dx \tag{161}$$

which should yield consistent results for any y values in the fluid domain. In the neighborhood of the fluid/solid interface this was not the case as can be seen from Figure 26. We believe this is due to the presence of high frequency components in this region and having too coarse a mesh to effectively evaluate the integral there.

With a stepsize h of 1/40 our truncation error was on the order of 10^{-3}. To increase the accuracy we can perform Romberg extrapolation or decrease the stepsize, thereby increasing the dimensions of the matrices involved. The latter was not an option due to the construction of the code and machine limitations, in that a simulation which involved 10 to 15 thousand time steps took from 10 to 12 hours to perform. Increasing the size of the matrices involved, thereby increasing the required time, would not make it a suitable code for experimentation and timely results.

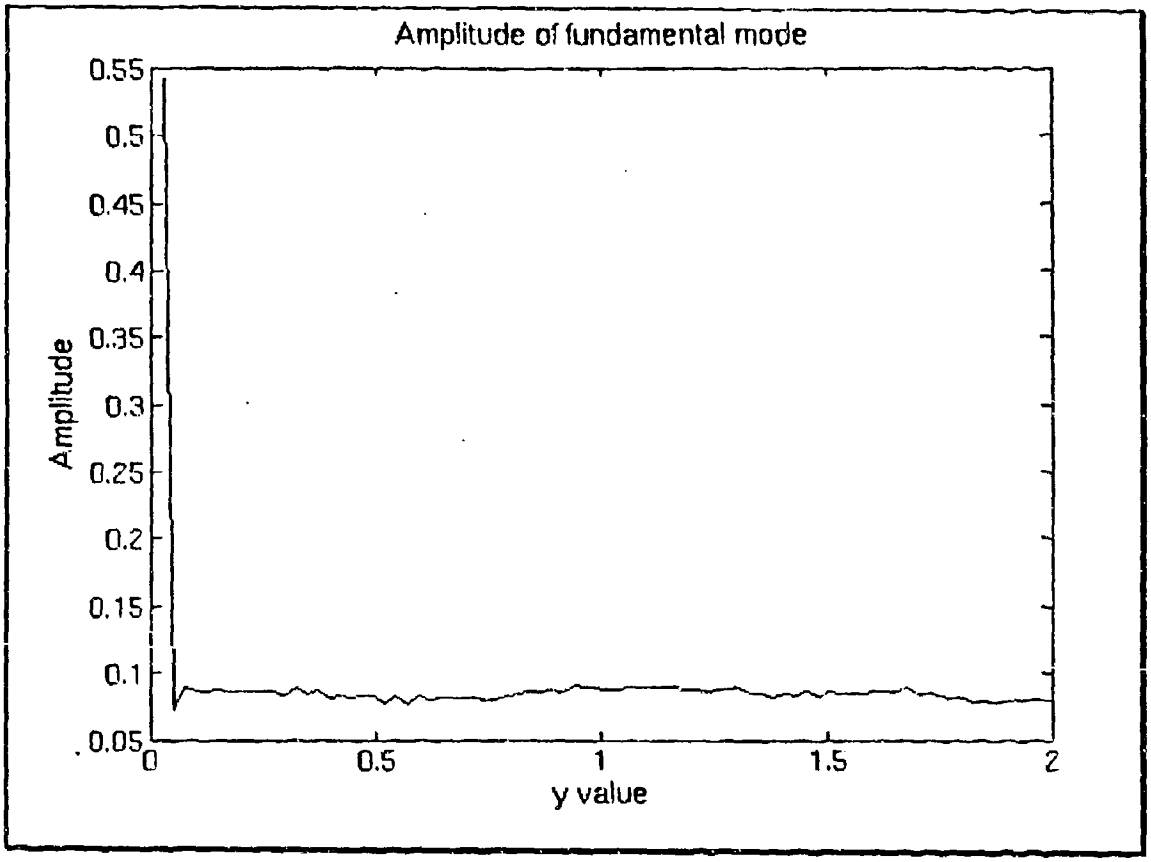


Figure 26. Amplitude of Propagating Modes in the Y-domain

Aspects of the problem which deserve further study would be the use of obliquely incident waves vice normal as was used here. Variation of the cavity size and its effect on the amplitude of the propagating modes should also be considered. As a simplifying assumption the cavities were considered to be void. It would be realistic to expect that they may contain fluid (seawater) or an acoustic dampening material. Modifying the model to account for such conditions warrants further study. A final aspect that could be considered is a resonance analysis.

Throughout all of our finite difference approximations we were able to maintain second order accuracy in space and time. We did not have to resort to the use of pseudonodes outside of our fluid-solid domain. Finally the possibility of being able to establish an acoustic signature for a double-hulled structure could open up a new avenue of submarine detection for which this thesis could be considered a starting point

APPENDIX A. VON NEUMANN STABILITY ANALYSIS FOR THE 2-D SCALAR WAVE EQUATION

The general form of the scalar wave equation being used is

$$\frac{1}{c_f^2}\frac{\partial^2 u}{\partial t^2}=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2} \tag{A-1}$$

whose equivalent difference equation, given $\Delta x = \Delta y = h$ is

$$\frac{1}{c_f^2\Delta t^2}\left(u_{i,j}^{n+1}-2u_{i,j}^{n}+u_{i,j}^{n-1}\right)=\frac{1}{h^2}\left(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}\right)+\frac{1}{h^2}\left(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}\right). \tag{A-2}$$

The error function takes the form

$$u=E\xi^{r}e^{i(\beta p+\gamma q)h}. \tag{A-3}$$

This is substituted into Equation A-2 and common terms are cancelled to give

$$\frac{1}{c^2\Delta t^2}\left(\xi-2+\xi^{-1}\right)=\frac{1}{h^2}\left(e^{i\beta h}-2+e^{-i\beta h}\right)+\frac{1}{h^2}\left(e^{i\gamma h}-2+e^{-i\gamma h}\right). \tag{A-4}$$

Using the following identity,

$$\cos\alpha x=\frac{e^{i\alpha x}+e^{-i\alpha x}}{2} \tag{A-5}$$

letting $\beta h = \gamma h$ and defining $\rho = c\Delta t/h$, Equation A-4 reduces to

$$\xi-2+\xi^{-1}=\rho^2(4\cos\beta h-4). \tag{A-6}$$

Multiplying Equation A-6 through by ξ and collecting like terms gives

$$\xi^2 - 2\xi\left(1 - 2\rho^2(\cos\beta p - 1)\right) + 1 = 0 \tag{A-7}$$

or equivalently

$$\xi^2 - 2\xi\left(1 - 4\rho^2 \sin^2\left(\frac{\beta h}{2}\right)\right) + 1 = 0. \tag{A-8}$$

The roots of this quadratic are

$$\xi_1, \xi_2 = 1 - 4\rho^2 \sin^2\left(\frac{\beta h}{2}\right) \pm \frac{1}{2}\sqrt{4\left(1 - 4\rho^2 \sin^2\left(\frac{\beta h}{2}\right)\right)^2 - 4}\ . \tag{A-9}$$

For stability we require

$$\xi \leq 1$$

which forces

$$4\left(1 - 4\rho^2 \sin^2\left(\frac{\beta h}{2}\right)\right)^2 - 4 < 0. \tag{A-10}$$

Solving for ρ^2 in Equation A-10 gives

$$\rho^2 \leq \frac{1}{2\sin^2\left(\frac{\beta h}{2}\right)} \tag{A-11}$$

which reduces to our final stability criterion of

$$\rho \leq \frac{1}{\sqrt{2\sin^2\left(\frac{\beta h}{2}\right)}} \tag{A-12}$$

or

$$\rho \leq \frac{1}{\sqrt{2}}$$

APPENDIX B. VON NEUMANN STABILITY ANALYSIS FOR THE ELASTIC WAVE EQUATION

In this section we give a brief outline of the stability analysis required for the elastic wave equation, whose general form is

$$c_L^2\frac{\partial^2 u}{\partial x^2}+c_T^2\frac{\partial^2 u}{\partial y^2}+\left(c_L^2-c_T^2\right)\frac{\partial^2 v}{\partial x\partial y}=\frac{\partial^2 u}{\partial t^2} \tag{B-1}$$

$$c_T^2\frac{\partial^2 v}{\partial x^2}+c_L^2\frac{\partial^2 v}{\partial y^2}+\left(c_L^2-c_T^2\right)\frac{\partial^2 u}{\partial x\partial y}=\frac{\partial^2 v}{\partial t^2} \tag{B-2}$$

The finite difference approximation for Equation B-1 is

$$\frac{c_L^2}{h^2}\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)+\frac{c_T^2}{h^2}\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right)$$

$$+\frac{\left(c_L^2-c_T^2\right)}{4h^2}\left(v_{i+1,j+1}^n-v_{i+1,j-1}^n-v_{i-1,j+1}^n+v_{i-1,j-1}^n\right)=\frac{1}{\Delta t^2}\left(u_{i,j}^{n+1}-2u_{i,j}^n+u_{i,j}^{n-1}\right) \tag{B-3}$$

We use the following error functions

$$u=U\xi^r e^{i(\beta p+\gamma q)h} \quad \text{and} \quad v=V\xi^r e^{i(\beta p+\gamma q)}h \tag{B-4}$$

which are substituted into Equation B-3, common terms are cancelled and complex exponentials are gathered into trigonometric quantities to give

$$-4r^2\left(c_L^2\sin^2\frac{\beta h}{2}+c_T^2\sin^2\frac{\gamma h}{2}\right)U-r^2\left(c_L^2-c_T^2\right)(\sin\beta h\sin\gamma h)V=\left(\xi-2+\xi^{-1}\right)\cdot U.\text{(B-5)}$$

Following the same procedure for Equation B-2 we obtain

$$-4r^2\left(c_T^2\sin^2\frac{\beta h}{2}+c_L^2\sin^2\frac{\gamma h}{2}\right)V-r^2\left(c_L^2-c_T^2\right)(\sin\beta h\sin\gamma h)U=\left(\xi-2+\xi^{-1}\right)\cdot V.\text{(B-6)}$$

Equations B-5 and B-6 can be written in matrix form which is

$$\begin{pmatrix}-4r^2\left(c_L^2\sin^2\frac{\beta h}{2}+c_T^2\sin^2\frac{\gamma h}{2}\right) & -r^2\left(c_L^2-c_T^2\right)\sin\beta h\sin\gamma h\\ -r^2\left(c_L^2-c_T^2\right)\sin\beta h\sin\gamma h & -4r^2\left(c_T^2\sin^2\frac{\beta h}{2}+c_L^2\sin^2\frac{\gamma h}{2}\right)\end{pmatrix}\begin{pmatrix}U\\V\end{pmatrix}=\Lambda\begin{pmatrix}U\\V\end{pmatrix}$$

(B-7).

where $\Lambda = \xi - 2 + \xi^{-1}$. The eigenvalues of matrix in Equation B-7 are

$$2r^2\left(c_L^2+c_T^2\right)(\cos\beta h+\cos\gamma h-2)\pm\frac{r^2\left(c_L^2-c_T^2\right)}{2}\left((\cos(\beta h-\gamma h)+\cos(\beta h+\gamma h)-2)^2\right)^{\frac{1}{2}}$$

(B-8).

they take on a maximum value when $\beta h = \gamma h = \pi$, and we obtain

$$\lambda_1=\lambda_2=-4r^2\left(c_L^2+c_T^2\right). \qquad \text{(B-9)}$$

We are now left with the identity

$$\xi-2+\xi^{-1}=-4r^2\left(c_L^2+c_T^2\right) \qquad \text{(B-10)}$$

or

$$\xi^2+\left(4r^2\left(c_L^2+c_T^2\right)-2\right)\xi+1=0. \qquad \text{(B-11)}$$

For stability we require

$$\xi\le 1,$$

which forces

$$\left(4r^2\left(c_L^2 + c_T^2\right) - 2\right)^2 - 4 \le 0 \qquad \text{(B-12)}$$

or

$$\Delta t \le \frac{h}{\sqrt{c_L^2 + c_T^2}} \qquad \text{(B-13)}$$

APPENDIX C. FINITE DIFFERENCE FORMULAE FOR THE EQUATION OF MOTION AND BOUNDARY CONDITIONS

This appendix contains the finite difference approximations of all the equations required for the simulation.

Solid

1) Elastic Wave Equation

$$\begin{aligned} u_{i,j}^{n+1} &= \frac{\Delta t^2}{k_L^2 h^2}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n}\right) + \frac{\Delta t^2}{k_T^2 h^2}\left(u_{i,j+1}^{n} + u_{i,j-1}^{n}\right) \\ &+ \frac{\Delta t^2}{4h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i+1,j+1}^{n} - v_{i-1,j+1}^{n} - v_{i+1,j-1}^{n} + v_{i-1,j-1}^{n}\right) \\ &+ \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\right)u_{i,j}^{n} - u_{i,j}^{n-1} \end{aligned} \qquad \text{(C - 1)}$$

$$\begin{aligned} v_{i,j}^{n+1} &= \frac{\Delta t^2}{k_T^2 h^2}\left(v_{i+1,j}^{n} + v_{i-1,j}^{n}\right) + \frac{\Delta t^2}{k_L^2 h^2}\left(v_{i,j+1}^{n} + v_{i,j-1}^{n}\right) \\ &+ \frac{\Delta t^2}{4h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i+1,j+1}^{n} - u_{i-1,j+1}^{n} - u_{i+1,j-1}^{n} + u_{i-1,j-1}^{n}\right) \\ &+ \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\right)v_{i,j}^{n} - v_{i,j}^{n-1}. \end{aligned} \qquad \text{(C - 2)}$$

Stress-free Boundary Conditior :

Surfaces with $\hat{n} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^{n} - u_{i,j}^{n-1} + \frac{2\Delta t^2}{k_L^2 h^2}u_{i-1,j}^{n} + \frac{\Delta t^2}{k_T^2 h^2}\left(u_{i,j+1}^{n} + u_{i,j-1}^{n}\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i-1,j-1}^{n} - v_{i+1,j-1}^{n}\right) - \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(v_{i,j+1}^{n} - v_{i,j-1}^{n}\right) \qquad (C-3)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^{n} - v_{i,j}^{n-1} + \frac{2\Delta t^2}{k_T^2 h^2}v_{i-1,j}^{n} + \frac{\Delta t^2}{h^2 k_L^2}\left(v_{i,j+1}^{n} + v_{i,j-1}^{n}\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i-1,j-1}^{n} - u_{i-1,j+1}^{n}\right) - \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(u_{i,j+1}^{n} - u_{i,j-1}^{n}\right) \qquad (C-4)$$

Surfaces with $\hat{n} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^{n} - u_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_L^2}u_{i+1,j}^{n} + \frac{\Delta t^2}{h^2 k_T^2}\left(u_{i,j+1}^{n} + u_{i,j-1}^{n}\right)$$

$$-\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i+1,j-1}^{n} - v_{i+1,j+1}^{n}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(v_{i,j+1}^{n} - v_{i,j-1}^{n}\right) \qquad (C-5)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^{n} - v_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_T^2}v_{i+1,j}^{n} + \frac{\Delta t^2}{h^2 k_L^2}\left(v_{i,j+1}^{n} + v_{i,j-1}^{n}\right)$$

$$-\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i+1,j-1}^{n} - u_{i+1,j+1}^{n}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{3}{k_T^2} - \frac{1}{k_L^2}\right)\left(u_{i,j+1}^{n} - u_{i,j-1}^{n}\right). \qquad (C-6)$$

Surfaces with $\hat{n} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^{n} - u_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_T^2}u_{i,j-1}^{n} + \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n}\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i-1,j-1}^{n} - v_{i+1,j-1}^{n}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(v_{i+1,j}^{n} - v_{i-1,j}^{n}\right) \qquad (C-7)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^{n} - v_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_L^2}v_{i,j-1}^{n} + \frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,j}^{n} + v_{i-1,j}^{n}\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i-1,j-1}^{n} - u_{i+1,j-1}^{n}\right) - \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(u_{i+1,j}^{n} - u_{i-1,j}^{n}\right) \qquad (C-8)$$

Surfaces with $\hat{n} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^{n} - u_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_T^2}u_{i,j+1}^{n} + \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,j}^{n} + u_{i-1,j}^{n}\right)$$

$$-\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i-1,j+1}^{n} - v_{i+1,j+1}^{n}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{3}{k_T^2} - \frac{1}{k_L^2}\right)\left(v_{i+1,j}^{n} - v_{i-1,j}^{n}\right) \qquad (C-9)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^{n} - v_{i,j}^{n-1} + \frac{2\Delta t^2}{h^2 k_L^2}v_{i,j+1}^{n} + \frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,j}^{n} + v_{i+1,j}^{n}\right)$$

$$-\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i-1,j+1}^{n} - u_{i+1,j+1}^{n}\right) + \frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{3}{k_T^2}\right)\left(u_{i+1,j}^{n} - u_{i-1,j}^{n}\right) \qquad (C-10)$$

Corners 2 and 4

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^n - u_{i,j}^{n-1} + \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,j}^n + u_{i-1,j}^n\right) + \frac{\Delta t^2}{k_T^2 h^2}\left(u_{i,j+1}^n - u_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(2v_{i,j}^n + v_{i+1,j-1}^n + v_{i-1,j+1}^n - v_{i+1,j}^n - v_{i-1,j}^n - v_{i,j+1}^n - v_{i,j-1}^n\right). \quad (\mathrm{C}-11)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^n - v_{i,j}^{n-1} + \frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,j}^n + v_{i-1,j}^n\right) + \frac{\Delta t^2}{k_L^2 h^2}\left(v_{i,j+1}^n - v_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(2u_{i,j}^n + u_{i+1,j-1}^n + u_{i-1,j+1}^n - u_{i+1,j}^n - u_{i-1,j}^n - u_{i,j+1}^n - u_{i,j-1}^n\right). \quad (\mathrm{C}-12)$$

Corner nodes 1, and 3

$$u_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)u_{i,j}^n - u_{i,j}^{n-1} + \frac{\Delta t^2}{h^2 k_L^2}\left(u_{i+1,j}^n + u_{i-1,j}^n\right) + \frac{\Delta t^2}{k_T^2 h^2}\left(u_{i,j+1}^n - u_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(v_{i+1,j}^n + v_{i-1,j}^n + v_{i,j+1}^n + v_{i,j-1}^n - v_{i-1,j-1}^n - v_{i+1,j+1}^n - 2v_{i,j}^n\right). \quad (\mathrm{C}-13)$$

$$v_{i,j}^{n+1} = \left(2 - \frac{2\Delta t^2}{h^2}\left(\frac{1}{k_L^2} + \frac{1}{k_T^2}\right)\right)v_{i,j}^n - v_{i,j}^{n-1} + \frac{\Delta t^2}{h^2 k_T^2}\left(v_{i+1,j}^n + v_{i-1,j}^n\right) + \frac{\Delta t^2}{k_L^2 h^2}\left(v_{i,j+1}^n - v_{i,j-1}^n\right)$$

$$+\frac{\Delta t^2}{2h^2}\left(\frac{1}{k_L^2} - \frac{1}{k_T^2}\right)\left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - u_{i-1,j-1}^n - u_{i+1,j+1}^n - 2u_{i,j}^n\right). \quad (\mathrm{C}-14)$$

Fluid

1) Wave Equation

$$p_{i,j}^{n+1} = \left(2 - \frac{4\Delta t^2}{k_f^2 h^2}\right) p_{i,j}^{n} - p_{i,j}^{n-1} + \frac{\Delta t^2}{k_f^2 h^2}\left(p_{i+1j}^{n} + p_{i-1,j}^{n} + p_{i,j+1}^{n} + p_{i,j-1}^{n}\right). \qquad \text{(C-15)}$$

Radiation Boundary Condition

$$p_{i,j}^{n+1} = \left(I + \frac{\Delta t}{2k_f^2} A\right)^{-1}\left(\frac{2\Delta t^2}{h^2 k_f^2} p_{i,j-1}^{n} + T p_{i,j}^{n} - \left(I - \frac{\Delta t}{2k_f^2} A\right) p_{i,j}^{n-1}\right). \qquad \text{(C-16)}$$

$$T = \begin{bmatrix} 2 - \frac{4\Delta t^2}{k_f^2 h^2} & \frac{\Delta t^2}{k_f^2 h^2} & 0 & \cdots & \frac{\Delta t^2}{k_f^2 h^2} & 0 \\ \frac{\Delta t^2}{k_f^2 h^2} & \ddots & \ddots & & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \frac{\Delta t^2}{k_f^2 h^2} \\ 0 & \frac{\Delta t^2}{k_f^2 h^2} & 0 & \cdots & \frac{\Delta t^2}{k_f^2 h^2} & 2 - \frac{4\Delta t^2}{k_f^2 h^2} \end{bmatrix}. \qquad \text{(C-17)}$$

APPENDIX D. COMPUTER CODE

```
% function base1 - base for the 'I - beam ' shaped domain
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% constructs a grid in the shape of an I beam

function [b,rows, pcoll,pcolr,fill,coll,colr] = base1(n)
% varisbles
% rows - identifies the rows to be zeroed to form the I beam
% coll - identifies the columns to be zeroed to the left of the center spar
% colr - identifies the columns to be zeroed to the right of the center spar
% pcolr - identifies the columns to be zeroed to the right cf the center spar
% including the boundary values.
% pcoll -  identifies the columns to be zeroed to the left of the center spar
% including the boundary values.
% bb - building block of the correct size
% cc - dimension of the blocks to be zeroed out on either side of the center
% spar
% s - variable to adjust the size of the zero blocks and keep it symmetric
% fill - block of zeros of appropiate size  to fill in the spaces on
% either side of center spar i.e. zeroing out,at every time step.
% m - no. of divisions in the half length of the domain.
% cr - variable used to pick the columns to the right of the spar



% build a grid of the correct size.
    bb = ones(2*n+1);


% the basic variables required to build the I - beam shaped grid
    m = n+1;
    s = log2(n);
    cc = (s-1)*(log2(n));

% determine the row numbers to be zeroed out.
   rows= m-(cc-1):m+(cc-1);


% columns to the left of the center spar, pcoll includes the boundary points
% coll does not.
   coll = 1:cc+1;
   pcoll = 1:cc+2;


% columns to the right of the center spar, pcolr includes the boundary points
% colr does not.

   cr = 2*n+1 - (cc);
   colr = cr:2*n+1;
   pcolr = cr-1:2*n+1;

% construct the block of zeros used to fill in the spaces on either side
% of the center spar at every time step.
   e = size(rows); e = e(2);
   d = size(coll); d = d(2);
   fill = zeros(e,cc+1);
```

```
bb(rows,coll) = fill;
bb(rows,colr) = fill;

b = bb;
```

```
%  function bndfnx - boundary facing in the negative x direction
%  i.e unit normal = (-1,0)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%  bndfnx applies the traction free boundary and the elastic
%  wave equation along the boundary of the I-beam with unit normal
%  (-1,0) the technique used is that as developed by Ilan and Lowenthal
%  and is not discussed here.
%  the columns containg the nodes on the surface in question and
%  the required neighbours (those one column in from the surface)
%  are picked off from the current and old values of u and v
%  the shifted and weighted with constants from the vectors
%  cunx (Constants for U-values for the Negative X boundary)
%  and cvnx (Constants for V-values for the Negative X boundary)
%  and inserted in the correct position of the updated u and v.



function [un ,vn] = bndfnx(uc,vc,uo,vo,un,vn,rows,pcoll,c,d);


% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll carries out the same function as rows for
% the columns, and the contains the column location of the
% of the boundary facing the negative x direction

 %  c = cunx;
 %  d = cvnx;


  [i3,j3] = size(rows);
  mrows = [rows(i3)-1 rows rows(j3)+1];
  [i4,j4] = size(mrows);
  [i5,j5] = size(pcoll);

%  cu1 cu2 cv1 cv2 co cov contain the necessary u and v values
%  for our calculations

 cu2 = uc(mrows,pcoll(j5)+1);
 cu1 = uc(mrows,pcoll(j5));

 cv2 = vc(mrows,pcoll(j5)+1);
 cv1 = vc(mrows,pcoll(j5));

 co = uo(mrows,pcoll(j5));
 cov = vo(mrows,pcoll(j5));


% the updated values are calculated

 uc1 = c(1)*cu1(2:j4-1) - co(2:j4-1) + c(2)*cu2(2:j4-1)...
       +c(3)*(cu1(3:j4) + cu1(1:j4-2))...
        +c(4)*(cv2(1:j4-2) - cv2(3:j4))...
```

```
         +c(5)*(cv1(3:j4) - cv1(1:j4-2));


 vcl = d(1)*cv1(2:j4-1) - cov(2:j4-1) + d(2)*cv2(2:j4-1)...
       +d(3)*(cv1(3:j4) + cv1(1:j4-2))...
        +d(4)*(cu2(1:j4-2) - cu2(3:j4))...
         +d(5)*(cu1(3:j4) - cu1(1:j4-2));


%  and put in their proper place in un and vn

   pb= size(mrows);
   pbc = mrows(2:pb(2)-1);


  un(pbc,pcoll(j5)) = ucl;
   vn(pbc,pcoll(j5)) = vcl;
```

```
%  function bndfny - boundary facing in the negative y direction
%  i.e unit normal = (0,-1)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%  bndfny applies the traction free boundary and the elastic
%  wave equation along the boundary of the I-beam with unit normal
%  (0,-1) the technique used is that as developed by Ilan and Lowenthal
%  and is not discussed here.
%  the rows containg the nodes on the surface in question and
%  the required neighbours (those one row in from the surface)
%  are picked off from the current and old values of u and v
%  the shifted and weighted with constants from the vectors
%  cuny (Constants for U-values for the Negative Y boundary)
%  and cvny (Constants for V-values for the Negative Y boundary)
%  and inserted in the correct position of the updated u and v.


function [un ,vn] = bndfny(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,c,d);

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll and pclor are both required as there are two regions,one on
% either side of the center spar of the I -beam which require
% our attention and they contain the location of the nodes in question



 % c = cuny;
 % d = cvny;


  [i1,j1] = size(pcoll);
  [i2,j2] = size(rows);
  [sr sc] = size(uc);

% rl** ru* rv* and ro* pick off the rows on either side of the
% center spar of the necessary u and v values.

 rlu1 = uc(rows(j2)+1,pcoll);
 rlu2 = uc(rows(j2)+2,pcoll);
 ru2 = uc(rows(j2)+2,pcolr);
 ru1 = uc(rows(j2)+1,pcolr);

 rlv1 = vc(rows(j2)+1,pcoll);
 rlv2 = vc(rows(j2)+2,pcoll);
 rv2 = vc(rows(j2)+2,pcolr);
 rv1 = vc(rows(j2)+1,pcolr);

 rol = uo(rows(j2)+1,pcoll);
 ro = uo(rows(j2)+1,pcolr);
 rolv = vo(rows(j2)+1,pcoll);
 rov = vo(rows(j2)+1,pcolr);
```

```
% the updated values are calculated

 ul = c(1)*rlu1(2:j1-1) - rol(2:j1-1) + c(2)*rlu2(2:j1-1)...
      +c(3)*(rlu1(3:j1) + rlu1(1:j1-2))...
       +c(4)*(rlv2(1:j1-2) - rlv2(3:j1))...
        +c(5)*(rlv1(3:j1) - rlv1(1:j1-2));

 ur = c(1)*ru1(2:j1-1) - ro(2:j1-1) + c(2)*ru2(2:j1-1)...
      +c(3)*(ru1(3:j1) + ru1(1:j1-2))...
       +c(4)*(rv2(1:j1-2) - rv2(3:j1))...
        +c(5)*(rv1(3:j1) - rv1(1:j1-2));


 vl = d(1)*rlv1(2:j1-1) - rolv(2:j1-1) + d(2)*rlv2(2:j1-1)...
      +d(3)*(rlv1(3:j1) + rlv1(1:j1-2))...
       +d(4)*(rlu2(1:j1-2) - rlu2(3:j1))...
        +d(5)*(rlu1(3:j1) - rlu1(1:j1-2));

 vr = d(1)*rv1(2:j1-1) - rov(2:j1-1) + d(2)*rv2(2:j1-1)...
      +d(3)*(rv1(3:j1) + rv1(1:j1-2))...
       +d(4)*(ru2(1:j1-2) - ru2(3:j1))...
        +d(5)*(ru1(3:j1) - ru1(1:j1-2));

%  and put in their proper place in un and vn

   pb= size(pcoll);
   pbl = pcoll(2:pb(2)-1);
   pbr = pcolr(2:pb(2)-1);


  un(rows(j2)+1,pbl) = ul;
   un(rows(j2)+1,pbr) = ur;


  vn(rows(j2)+1,pbl) = vl;
   vn(rows(j2)+1,pbr) = vr;

% the same procedure is reppeated for the 'bottom' of the I-beam

 un(1,2:sc-1) = c(1)*uc(1,2:sc-1) - uo(1,2:sc-1)...
    + c(2)*uc(2,2:sc-1)...
      +c(3)*(uc(1,3:sc) + uc(1,1:sc-2))...
       +c(4)*(vc(2,1:sc-2) - vc(2,3:sc))...
        +c(5)*(vc(1,3:sc) - vc(1,1:sc-2));


 vn(1,2:sc-1) = d(1)*vc(1,2:sc-1) - vo(1,2:sc-1)...
    + d(2)*vc(2,2:sc-1)...
      +d(3)*(vc(1,3:sc) + vc(1,1:sc-2))...
       +d(4)*(uc(2,1:sc-2) - uc(2,3:sc))...
        +d(5)*(uc(1,3:sc) - uc(1,1:sc-2));
```

```
%  function bndfpx - boundary facing in the positive x direction
%  i.e unit normal = (1,0)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%  bndfnx applies the traction free boundary and the elastic
%  wave equation along the boundary of the I-beam with unit normal
%  (0,1) the technique used is that as developed by Ilan and Lowenthal
%  and is not discussed here.
%  the columns containg the nodes on the surface in question and
%  the required neighbours (those one column in from the surface)
%  are picked off from the current and old values of u and v
%  the shifted and weighted with constants from the vectors
%  cupx (Constants for U-values for the Positive X boundary)
%  and cvpx (Constants for V-values for the Positive X boundary)
%  and inserted in the correct position of the updated u and v.

function [un ,vn] = bndfpx(uc,vc,uo,vo,un,vn,rows,pcolr,c,d)

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcolr carries out the same function as rows for
% the columns, and the contains the column location of the
% of the boundary facing the positive x direction


 %   c = cupx;
 %   d = cvpx;


  [i3,j3] = size(rows);
  mrows = [rows(i3)-1 rows rows(j3)+1];
  [i4,j4] = size(mrows);


%  cu1 cu2 cv1 cv2 co cov contain the necessary u and v values
%  for our calculations


 cu2 = uc(mrows,pcolr(1)-1);
 cu1 = uc(mrows,pcolr(1));

 cv2 = vc(mrows,pcolr(1)-1);
 cv1 = vc(mrows,pcolr(1));

 co = uo(mrows,pcolr(1));
 cov = vo(mrows,pcolr(1));

% the updated values are calculated


 ucr = c(1)*cu1(2:j4-1) - co(2:j4-1) + c(2)*cu2(2:j4-1)...
       +c(3)*(cu1(3:j4) + cu1(1:j4-2))...
        +c(4)*(cv2(1:j4-2) - cv2(3:j4))...
         +c(5)*(cv1(3:j4) - cv1(1:j4-2));
```

```
vcr = d(1)*cv1(2:j4-1) - cov(2:j4-1) + d(2)*cv2(2:j4-1)...
      +d(3)*(cv1(3:j4) + cv1(1:j4-2))...
       +d(4)*(cu2(1:j4-2) - cu2(3:j4))...
        +d(5)*(cu1(3:j4) - cu1(1:j4-2));


%  and put in their proper place in un and vn

   pb= size(mrows);
   pbc = mrows(2:pb(2)-1);


  un(pbc,pcolr(1)) = ucr;
   vn(pbc,pcolr(1)) = vcr;
```

```
%  function bndfny - boundary facing in the positive y direction
%  i.e unit normal = (0,1)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%  bndfpy applies the traction free boundary and the elastic
%  wave equation along the boundary of the I-beam with unit normal
%  (0,1) the technique used is that as developed by Ilan and Lowenthal
%  and is not discussed here.
%  the rows containg the nodes on the surface in question and
%  the required neighbours (those one row in from the surface)
%  are picked off from the current and old values of u and v
%  the shifted and weighted with constants from the vectors
%  cupy (Constants for U-values for the Positive Y boundary)
%  and cvpy (Constants for V-values for the Positive Y boundary)
%  and inserted in the correct position of the updated u and v.

function [un ,vn] = bndfpy(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,c,d);

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll and pclor are both required as there are two regions,one on
% either side of the center spar of the I -beam which require
% our attention and they contain the location of the nodes in question


 % c = cupy;
 %  d = cvpy;


  [i1,j1] = size(pcoll);
  [i2,j2] = size(rows);

% rl** ru* rv* and ro* pick off the rows on either side of the
% center spar of the necessary u and v values.


 rlu1 = uc(rows(1)-1,pcoll);
 rlu2 = uc(rows(1)-2,pcoll);
 ru2 = uc(rows(1)-2,pcolr);
 ru1 = uc(rows(1)-1,pcolr);


 rlv1 = vc(rows(1)-1,pcoll);
 rlv2 = vc(rows(1)-2,pcoll);
 rv2 = vc(rows(1)-2,pcolr);
 rv1 = vc(rows(1)-1,pcolr);

  ro1 = uo(rows(1)-1,pcoll);
  ro = uo(rows(1)-1,pcolr);
  rolv = vo(rows(1)-1,pcoll);
  rov  = vo(rows(1)-1,pcolr);
```

```
% the updated values are calculated


 ul = c(1)*rlu1(2:j1-1) - rol(2:j1-1) + c(2)*rlu2(2:j1-1)...
      +c(3)*(rlu1(3:j1) + rlu1(1:j1-2))...
       +c(4)*(rlv2(1:j1-2) - rlv2(3:j1))...
        +c(5)*(rlv1(3:j1) - rlv1(1:j1-2));

 ur = c(1)*ru1(2:j1-1) - ro(2:j1-1) + c(2)*ru2(2:j1-1)...
      +c(3)*(ru1(3:j1) + ru1(1:j1-2))...
       +c(4)*(rv2(1:j1-2) - rv2(3:j1))...
        +c(5)*(rv1(3:j1) - rv1(1:j1-2));


 vl = d(1)*rlv1(2:j1-1) - rolv(2:j1-1) + d(2)*rlv2(2:j1-1)...
      +d(3)*(rlv1(3:j1) + rlv1(1:j1-2))...
       +d(4)*(rlu2(1:j1-2) - rlu2(3:j1))...
        +d(5)*(rlu1(3:j1)- rlu1(1:j1-2));

 vr = d(1)*rv1(2:j1-1) - rov(2:j1-1) + d(2)*rv2(2:j1-1)...
      +d(3)*(rv1(3:j1) + rv1(1:j1-2))...
       +d(4)*(ru2(1:j1-2) - ru2(3:j1))...
        +d(5)*(ru1(3:j1) - ru1(1:j1-2));

%  and put in their proper place in un and vn

   pb= size(pcoll);
   pbl = pcoll(2:pb(2)-1);
   pbr = pcolr(2:pb(2)-1);


  un(rows(1)-1,pbl) = ul;
   un(rows(1)-1,pbr) = ur;

  vn(rows(1)-1,pbl) = vl;
   vn(rows(1)-1,pbr) = vr;
```

```
%  function cmodck - checks the amplitude of the propagating mode
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
% cmodck is the driver program for the problem.All the vaue of the constants
% are defined here and all quantities are scaled before they are fed into the

% decks are cleared before calculation
 record('erase')
   clear
     clg
      axis('auto')

dx = input('step size = ');
a =  input(' length scaling factor = ');
omg = input(' time scaling factor (omega) = ');

% constants
% ct: tranverse velocity of the solid (steel)
% cl: longitudinal velocity of the solid (steel)
% dnss: density of the solid (steel)
% cf : speed of sound in fluid (seawater)
% dnsf : density of fluid (seawater)
% epss: ratio ot fluid to solid densities

ct = 3200; cl = 5900; cf = 1500; dnsf = 1026; dnss = 7700;
epss = dnsf/dnss;

% the determination of the scaled variables
% dxs scaled distance
% dts scaled time
dxs = dx/a;
kf = omg/cf;kt = (omg*a)/ct; kl = (omg*a)/cl;

% dts1 and dts2 are the stability criteria for the solid and fluid
% always choose the minimum
dts1 =  (kl*dxs)/(sqrt(1 + (ct^2/cl^2)));
dts2 =  .5*((kf*a*dxs)/sqrt(2));

[dts1 dts2]

 dts = input(' desired time step = ');
 nn = input(' number of timesteps = ' );

% st - when to stop building the radiation boundary condition
% for in the construction of the matrix A each loop thru the construction
% cylce allows another mode to propagate
 st = stop(kf);

  i = sqrt(-1);
% x a vector the length of the domain used in several places
% i.e. when integrating etc.
% mm - the mode being checked 0-fundamental etc.
% nt - no of intervals in domain
% tt weighting factor for the trapezoidal rule
  x = -1:dxs:1 ;
  m = size(x) ; m = m(2) ; m1 = zeros(m) ;
      mm = 0
      nt = (1/dxs);
      tt = (m-1)/2;
```

```
[un,vn,mn,z0]  = cwv4(m1,k1,kt,kf,dts,dxs,x,nn,st,epss,tt,nt,mm);

% un displacment in the x dirn
% vn displacment in the y dirn
% mn displacment of the fluid
% z0 vector containing the amplitude of the propagating mode
%  function coupl - couples the two media at the fluid solid interface
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
% coupl applies the traction boundary conditions at the fluid solid
% interface where the shear component is zero but the normal component is
% given by  tau(yy) = -p(total) ,which causes the I beam to deform
% and causing the propagation of scattered pressure waves to propagate in
% fluid. To prevent cavitation at the interface we apply the inviscid
% form of the Navier-Stokes equation ( or Euler's equation ) at the
% the boundary


function [un ,vn,mnpd] = coupl(uc,vc,uo,vo,un,vn,c,d,k,dts,dxs,a11,b11,epss);
% variables
% un : updated values of u     vn : updated values of v
% uc : current values of u     vc : current values of v
% uo : old values of u         vo : old values of v



   % c = cupy;
   % d = cvpy;


  i = sqrt(-1);
 [sr sc] = size(uc);

% the forcing function
  time = exp(-i*k*dts)*ones(1,sc);


% the u component requires no modification and is treated in the
% usual fashion.
 un(sr,2:sc-1) = c(1)*uc(sr,2:sc-1) - uo(sr,2:sc-1)...
    + c(2)*uc(sr-1,2:sc-1)...
      + c(3)*(uc(sr,3:sc) + uc(sr,1:sc-2))...
       + c(4)*(vc(sr-1,1:sc-2) - vc(sr-1,3:sc))...
        + c(5)*(vc(sr,3:sc) - vc(sr,1:sc-2));

% the periodic boundary condition for u
 un(sr,1) = c(1)*uc(sr,1) - uo(sr,1)...
     + c(2)*uc(sr-1,1)...
      + c(3)*(uc(sr,2) + uc(sr,sc-1))...
       + c(4)*(vc(sr-1,sc-1) - vc(sr-1,2))...
        + c(5)*(vc(sr,2) - vc(sr,sc-1));

    un(sr,sc) = un(sr,1);


% the normal component of stress plus the incident,reflected
% and scattered pressures
 vn(sr,2:sc-1) = d(1)*vc(sr,2:sc-1) - vo(sr,2:sc-1)...
    + d(2)*vc(sr-1,2:sc-1)...
      +d(3)*(vc(sr,3:sc) + vc(sr,1:sc-2))...
       +d(4)*(uc(sr-1,1:sc-2) - uc(sr-1,3:sc))...
        +d(5)*(uc(sr,3:sc) - uc(sr,1:sc-2))...
          + ((2*dts^2)/dxs)*(2*epss*time(2:sc-1) + epss*a11(2:sc-1));

% the periodic boundary condition for v
```

```
 vn(sr,1) = d(1)*vc(sr,1) - vo(sr,1)...
    + d(2)*vc(sr-1,1)...
      +d(3)*(vc(sr,2) + vc(sr,sc-1))...
       +d(4)*(uc(sr-1,sc-1) - uc(sr-1,2))...
        +d(5)*(uc(sr,2) - uc(sr,sc-1))...
         + ((2*dts^2)/dxs)*(2*epss*time(1) + epss*a11(1,1));


    vn(sr,sc) = vn(sr,1);

% the compatability condition.
  mnpd = (-(2*dxs)/(dts^2))*(vn(sr,:)- 2*vc(sr,:) +vo(sr,:)) + b11(1,:);
```

```
% function cwv4 - coupled waves version 4
%  written by: Lt. Hugh Mc Bride USNR
%  date       : Apr 92
%
% cwv4 is the main program which couples the behaviour of the 'I-beam' shaped
% solid medium with the fluid medium. The elastic wave equation and the boundary
% conditions ,periodic and traction free are are satisfied for the solid, while
% the scalar wave equation and the periodic and radiating boundary conditions ar
% applied to the fluid.
%
%  The general steps of the program are as follows
%
%  1. The basic parameters are determined ,that is the size of the domains
%  as passed to it by the driver program cmodck.m
%
%  2. All the global variables are calculated for both the fluid and the solid
%     including the matrices required for the radiation boundary condition.
%
%  3. v1 and v3 are vectors used to find the amplitude, v1 is the
%     weighting vector 1/2 1 1 1 .... 1 1/2 for the trapezcidal rule
%     and v3 =  exp(i*n*pi*x) , the othhogonal vector.
%
%  4. The I beam is built by base1
%
%  5. The initial values of the u and v for the solid (un,vn,uc,vc,uo,vo)
%      and m (mn mc and mold) for the fluid are set to zero.
%
%  6. The elastic wave equation and the periodic boundary conditions
%     are applied to u and v
%
%  7. The boundary conditions for the solid are applied
%
%  8. The fluid and solid are coupled
%
%  9. The freespace portions of the I beam are zeroed out
%
% 10. Tne scalar wave equation and the periodic boundary conditions for
%     the fluid  are applied.
%
% 11. The values of the amplitude are calculated and accumulated
%
% 12.  u, v and m are updated ,that is the new values become
%      the current values and the current values become the old values.
%
%
%

 function  [v,un,vn,mn,z2] = cwv4(m,kl,kt,kf,dts,dxs,x,nn,st,epss,tt,nt,mm);

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll and pclor are both required as there are two regions,one on
% either side of the center spar of the I -beam which require
% our attention and they contain the location of the nodes in question
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step
% x - vector of length of the interval
% nn - no of timesteps
% st - stopping criteria for radiation boundary condition
% epss - ratio of solid density to fluid density
% nt - no of intervals in domain
% mm - prpoagating mode under investigation ,mm = 0 fundamental
% mm = 1 first and so forth
% z? - amplitude of propagating mode



% Step 1 : r,c the dimensions of the domain, r-1 ,c-1 and n-1
%          the dimensions of the interior

      [r c] = size(m);
      n = r-1; r = r-1;c = c-1;
      axis('xy')

% Step 2: The global variables


% For the solid
      rho = rhov(kl,kt,dxs,dts);     ccs = (2 - 2*rho(1)-2*rho(2));
      cunx = ucnx(kl,kt,dxs,dts);  cvnx = vcnx(kl,kt,dxs,dts);
      cuny = ucny(kl,kt,dxs,dts);  cvny = vcny(kl,kt,dxs,dts);
      cupx = ucpx(kl,kt,dxs,dts);  cvpx = vcpx(kl,kt,dxs,dts);
      cupy = ucpy(kl,kt,dxs,dts);  cvpy = vcpy(kl,kt,dxs,dts);

% For the fluid
      rhof =(dts^2)/(kf^2*dxs^2);    ccf = (2 - 4*rhof);
      rcl = (2*dts^2)/(kf^2*dxs^2) ;

% The matrices for the radiation boundary condition allowing the propagating
% modes to pass through the artificial boundary.


      anew = zeros(c+1);

         for  pm = 0:st

          acurr = rbc(r+1,c+1,pm,dxs,kf);

        anew = anew+acurr;

     end

     m1 = (eye(c+1) + (dts/(2*kf^2))*anew); m1 = inv(m1);

     m2 = (eye(c+1) - (dts/(2*kf^2))*anew);

     t = trid(dxs,dts,c+1,kf);
% Step 3: v1 and v3 calculated so as to be able to determine the amplitude
  v1 = del2(2*nt);
  v3 = exp(i*mm*pi*x');


% Step 4: basel builds the I -beam pclor,pcoll ,coll,colr and fill
% allow us to identify the boundaries of the solid domain and the
% corner nodes.

  [b,rows, pcoll,pcolr,fill,coll,colr] = basel(n/2);


% Step 5: All values for the fluid and the solid are initially
% set to zero.
       un = zeros(size(m));  uc = un; uo = un;

       vn = zeros(size(m)); vc = vn;  vo = vn;

       mn = zeros(size(m)); mc = mn ;mold = mn;



         for k = 1:nn

% Step 6: The elastic wave equation is applied to the interior
% of the solid

   un(2:n,2:n) = ccs*uc(2:n,2:n) - uo(2:n,2:n)...
    + rho(2)*(uc(1:n-1,2:n) + uc(3:n+1,2:n))...
     + rho(1)*(uc(2:n,1:n-1) + uc(2:n,3:n+1))...
       + rho(3)*(vc(1:n-1,1:n-1) + vc(3:n+1,3:n+1))...
         - rho(3)*(vc(1:n-1,3:n+1) + vc(3:n+1,1:n-1)) ;



      vn(2:n,2:n) = ccs*vc(2:n,2:n) - vo(2:n,2:n)...
      + rho(1)*(vc(1:n-1,2:n) + vc(3:n+1,2:n))...
       + rho(2)*(vc(2:n,1:n-1) + vc(2:n,3:n+1))...
        + rho(3)*(uc(1:n-1,1:n-1) + uc(3:n+1,3:n+1))...
         - rho(3)*(uc(1:n-1,3:n+1) + uc(3:n+1,1:n-1)) ;


% The periodic boundary conditions for the solid

      un(2:n,1) = ccs*uc(2:n,1) - uo(2:n,1)...
      + rho(2)*(uc(1:n-1,1) + uc(3:n+1,1))...
       + rho(1)*(uc(2:n,2) + uc(2:n,n))...
        + rho(3)*(vc(1:n-1,n) + vc(3:n+1,2))...
         - rho(3)*(vc(1:n-1,2) + vc(3:n+1,n)) ;

        un(2:n,n+1) = un(2:n,1);


      vn(2:n,1) = ccs*vc(2:n,1) - vo(2:n,1)...
       + rho(1)*(vc(1:n-1,1) + vc(3:n+1,1))...
        + rho(2)*(vc(2:n,2) + vc(2:n,n))...
         + rho(3)*(uc(1:n-1,n) + uc(3:n+1,2))...
          - rho(3)*(uc(1:n-1,2) + uc(3:n+1,n)) ;

        vn(2:n,n+1) =  vn(2:n,1);

% Step 7: The boundaries of the I beam and including the corner nodes
% are treated.
```

```
   [un ,vn] = bndfnx(uc,vc,uo,vo,un,vn,rows,pcoll,cunx,cvnx);
   [un ,vn] = bndfny(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,cuny,cvny);
   [un ,vn] = lcorny(uc,vc,uo,vo,un,vn,rows,pcoll,cuny,cvny);
   [un ,vn] = bndfpx(uc,vc,uo,vo,un,vn,rows,pcolr,cupx,cvpx);
   [un ,vn] = bndfpy(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,cupy,cvpy);
   [un ,vn] = lcorpy(uc,vc,uo,vo,un,vn,rows,pcoll,cupy,cvpy);
   [un,vn] = tcor13(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,rho);
   [un,vn] = tcor24(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,rho);


% Step 8 : The solid and the fluid are coupled (Note the is where the forcing
% function of the problem is contained )

   a11 =  mc(1,:); b11 = mc(2,:);
   [un ,vn,mnpd] = coupl(uc,vc,uo,vo,un,vn,cupy,cvpy,k,dts,dxs,a11,b11,epss);


% Spep 9: Both sides of the center spar for all values of u and v are zeroed
% out so as to prevent pollution

 un(rows,coll) = fill; vn(rows,coll) = fill;
 uc(rows,coll) = fill; vc(rows,coll) = fill;
 un(rows,colr) = fill; vn(rows,colr) = fill;
 uc(rows,colr) = fill; vc(rows,colr) = fill;
 uo(rows,coll) = fill; vo(rows,coll) = fill;
 uo(rows,colr) = fill; vo(rows,colr) = fill;



% Step 10: The scalar wave equation for the fluid


% First the fluid/solid interface
    mn(1,2:c) =  rhof*(mnpd(2:c) + mc(2,2:c)...
      +mc(1,1:c-1) + mc(1,3:c+1)) +ccf*mc(1,2:c)...
      -mold(1,2:c);

% and it's periodic boundary condition
      mn(1,1) =  rhof*(mnpd(1,1) + mc(2,1)...
      +mc(1,c) + mc(1,3)) +ccf*mc(1,1)...
      -mold(1,1);

      mn(1,c+1) = mn(1,1);



% The interior points of the fluid
    mn(2:r,2:c) =  rhof*(mc(1:r-1,2:c) + mc(3:r+1,2:c)...
      +mc(2:r,1:c-1) + mc(2:r,3:c+1)) +ccf*mc(2:r,2:c)...
      -mold(2:r,2:c);


% The radiation boundary condition
% Note: We are required to multiply a matrix by a  row vector,but to do this
% it must be transposed to a column . Matlab takes the Hermitian transpose
% by default , so to ensure the correct signs we must void this effect by
% taking the conjugate of the transpose before we do our calculations.
% The process is then reversed so as the updated value has the correct dimension
%
```

```
      s1 = conj(mc(r,:)');
       s2 = conj(mc(r+1,:)');
        s3 = conj(mold(r+1,:)');
         int = m1*(rc1*s1 + t*s2 - m2*s3);
          mn(r+1,:) = conj(int');


% the periodic boundary condition for the artificial boundary

     mn(2:r,1) = rhof*(mc(1:r-1,1) + mc(2:r,2) + ...
     mc(3:r+1,1)+ mc(2:r,c)) + ccf*mc(2:r,1) - mold(2:r,1);

     mn(2:r,c+1) = mn(2:r,1);


% Step 11: Calulates amd accumulates the values of the amplitude
     aa = abs((mn(r,:).*v1)*v3);
     z2 = [z2 aa];

% Step 12: The values for u,v and m are updated.

      uo = uc;   uc = un;
      vo = vc;   vc = vn;
      mold = mc;   mc = mn;


end
```

```
% function del2 - del funtion used for thr trapezoidal rule
%  written by: Lt. Hugh Mc Bride USNR
%  date       : Apr 92
% bulids a vector of appropiate length  used to weight the elements of the
% quantity being integrated

function  d = del2(n)
 a = ones(1,n+1);
  a(1) = .5; a(n+1) = .5;
   a2 = ones(1,n+1)/n;

  d = a.*a2;
```

```
%  function lcorny - left corner facing in the negative y direction
%  i.e unit normal = (0,-1)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%
%
% lcorny performs the same function as bndfny except it only operates on
% 4 points , those which lie  at the extremities of the domain, the points
% as if fig( ) Periodicity is used for pseudonodes which lie outside the domain.
% This requires us to pick  out each nodes individually (9 for each 4 points ,
% making a total of 36 )  they are weighted in the same fashion
% as in bndfny and the updated values for u and v are calculated for both sides

function [un ,vn] = lcorny(uc,vc,uo,vo,un,vn,rows,pcoll,c,d);

% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll allows us to pick out the column location of those nodes
% to the left of centerline and using the periodicity we substitute
% this value into the corresponding point on the right of
% centerline.


   % c = cuny;
   % d = cvny;


  [i1,j1] = size(pcoll);
  [i2,j2] = size(rows);
  [i3,j3] = size(uc);



% the updated values of u to the left  of centerline are
% calculated

  ucc = uc(rows(j2)+1,pcoll(1));
  ucc1 = uc(rows(j2)+2,pcoll(1));
  ucr = uc(rows(j2)+1,pcoll(2));
  ucl = uc(rows(j2)+1,j3-1);

  vcr = vc(rows(j2)+1,pcoll(2));
  vru = vc(rows(j2)+2,pcoll(2));
  vlu = vc(rows(j2)+2,j3-1);
  vcl = vc(rows(j2)+1,j3-1);

  uol = uo(rows(j2)+1,pcoll(1));

 ul = c(1)*ucc - uol + c(2)*ucc1 +c(3)*(ucr + ucl)...
      +c(4)*(vlu - vru) + c(5)*(vcr- vcl);


  ucc1 = uc(1,pcoll(1));
   ucu1 = uc(2,pcoll(1));
```

```
    ucr1 = uc(1,pcoll(2));
      ucl1 = uc(1,j3-1);

  vcr1 = vc(1,pcoll(2));
   vru1 = vc(2,pcoll(2));
    vlu1 = vc(2,j3-1);
     vcl1 = vc(1,j3-1);

       uol1 = uo(1,pcoll(1));


 ul1 = c(1)*uccl - uol1 + c(2)*ucu1 +c(3)*(ucr1 + ucl1)...
       +c(4)*(vlu1 - vru1) + c(5)*(vcr1- vcl1);



% the updated values of v to the left  of centerline are
% calculated



  vucc = vc(rows(j2)+1,pcoll(1));
   vuccl = vc(rows(j2)+2,pcoll(1));
    vucr = vc(rows(j2)+1,pcoll(2));
      vucl = vc(rows(j2)+1,j3-1);

  vvcr = uc(rows(j2)+1,pcoll(2));
   vvru = uc(rows(j2)+2,pcoll(2));
    vvlu = uc(rows(j2)+2,j3-1);
     vvcl = uc(rows(j2)+1,j3-1);

  vuol = vo(rows(j2)+1,pcoll(1));

 vul = d(1)*vucc - vuol + d(2)*vuccl + d(3)*(vucr + vucl)...
       +d(4)*(vvlu - vvru) + d(5)*(vvcr- vvcl);


  vuccl = vc(1,pcoll(1));
   vucu1 = vc(2,pcoll(1));
    vucr1 = vc(1,pcoll(2));
      vucl1 = vc(1,j3-1);

  vvcr1 = uc(1,pcoll(2));
   vvru1 = uc(2,pcoll(2));
    vvlu1 = uc(2,j3-1);
     vvcl1 = uc(1,j3-1);

       vuol1 = vo(1,pcoll(1));


 vul1 = d(1)*vuccl - vuol1 + d(2)*vucu1 +d(3)*(vucr1 + vucl1)...
       +d(4)*(vvlu1 - vvru1) + d(5)*(vvcr1- vvcl1);

% the values put in the correct positions
% and since we have periodic boundary conditions, we insert the left
% hand value into the correspondding right hand position

    un(rows(j2)+1,pcoll(1)) = ul;
    un(rows(j2)+1,j3) = ul;
```

```
vn(rows(j2)+1,pcoll(1)) = vul;
vn(rows(j2)+1,j3) = vul;


un(1,pcoll(1)) = ul1;
un(1,j3) = ul1;

vn(1,pcoll(1)) = vul1;
vn(1,j3) = vul1;
```

```
%  function lcorpy - left corner facing in the positive y direction
%  i.e unit normal = (0,1)
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%
%
% lcorpy performs the same function as bndfpy except it only operates on
% 2 points , those which lie  at the extremities of the domain, the points
% as in fig( ) Periodicity is used for pseudonodes which lie outside the domain.
% This requires us to pick  out each nodes individually (9 for each 2 points ,
% making a total of 18 )  they are weighted in the same fashion
% as in bndfpy and the updated values for u and v are calculated for both sides


function [un ,vn] = lcorpy(uc,vc,uo,vo,un,vn,rows,pcoll,c,d);

% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the nodes on the boundary.
% pcoll allows us to pick out the column location of those nodes
% to the left of centerline and using the periodicity we substitute
% this value into the corresponding point on the right of
% centerline.



  % c = cupy;
  %  d = cvpy;


% there are now only two values which need to be calculated
% as the other boundary facing in the positive
% y direction is at the fluid/solid interface and requires
% a special treatment

  [i1,j1] = size(pcoll);
  [i2,j2] = size(rows);
  [i3,j3] = size(uc);


% the updated u value

  ucc = uc(rows(1)-1,pcoll(1));
   ucu = uc(rows(1)-2,pcoll(1));
    ucr = uc(rows(1)-1,pcoll(2));
      ucl = uc(rows(1)-1,j3-1);

  vcr = vc(rows(1)-1,pcoll(2));
   vru = vc(rows(1)-2,pcoll(2));
    vlu = vc(rows(1)-2,j3-1);
     vcl = vc(rows(1)-1,j3-1);

       uol = uo(rows(1)-1,pcoll(1));

  ul = c(1)*ucc - uol + c(2)*ucu +c(3)*(ucr + ucl)...
       +c(4)*(vlu - vru) + c(5)*(vcr- vcl);
```

```
% the updated v value

  vucc = vc(rows(1)-1,pcoll(1));
   vucu = vc(rows(1)-2,pcoll(1));
     vucr = vc(rows(1)-1,pcoll(2));
       vucl = vc(rows(1)-1,j3-1);

  vvcr = uc(rows(1)-1,pcoll(2));
   vvru = uc(rows(1)-2,pcoll(2));
     vvlu = uc(rows(1)-2,j3-1);
      vvcl = uc(rows(1)-1,j3-1);

        vuol = vo(rows(1)-1,pcoll(1));

 vul = d(1)*vucc - vuol + d(2)*vucu + d(3)*(vucr + vucl)...
        +d(4)*(vvlu - vvru) + d(5)*(vvcr- vvcl);


% using periodicity we update the values to the left and right of
% the center spar

  un(rows(1)-1,pcoll(1)) = ul;
  un(rows(1)-1,j3) = ul;
  vn(rows(1)-1,pcoll(1)) = vul;
  vn(rows(1)-1,j3) = vul;
```

```
%  function rbc - radiating bondary condition
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Mar 92
%
% rbc builds the matrix A(i,k) which is required by the radiation
% boundary condition

function v = rbc(j,l,pm,dxs,kf)

% variables
% j,l -dimensions of the matrix
% pm - propagating modes
  a = zeros(j);

    for i = 1:j,

       for k = 1:l,

            a(i,k) = exp(sqrt(-1)*pm*pi*(i-k)*dxs);

           end

   end

        a(:,1) = .5*a(:,1) ; a(:,k) = .5*a(:,k);

     v = sqrt(kf^2-(pm*pi)^2)*a;
```

```
% function rho- del funtion used for thr trapezoidal rule
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% generates a vector containing constants used repeatedly throughout
% the program.

function  rho  = rhov(kl,kt,dxs,dts);


     rho1 =(dts^2)/(kl^2*dxs^2);

     rho2 =(dts^2)/(kt^2*dxs^2);

     rho3 = (dts^2/(4*dxs^2))*((1/kl^2)-(1/kt^2));

     rho = [rho1 rho2 rho3];
```

```
% function stop - defines the stopping criteria for constructing the radiation
% boundary condition i.e if the result of stop is 0 then only the fundamental mo
% is allowed to propagate, 1 only the fundamental and the first mode are allowed
% to prpogate and so on.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Mar 92
%
 function v = stop(k)


% variables
% k -
% n - nth propagating mode
 n = 0 ; m = -1;
 while (k^2 - n^2*pi^2) > 0.
      m = m+1; n = n+1;
end
v = m;
```

```
%  function tcor13 - treatment of corners 1 and 3
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%
%  tcor13 applies the elastic wave equation at
%  corners 1 and 3 as per fig( ) since they use the
%  same difference formula.
%  the corner nodes are located (1 first then 3)
%  identified as p and q (p1 ,q1 in the case of corner 3)
%  the necessary neighbours are picked off from the u and v
%  matrices and weighted accordingly and the new updated values
%  for u and v are  computed.


   function  [un,vn] = tcor13(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,rho);

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rho : vector containing global constants
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the corner nodes.
% pclor and pcoll carries out the same function as rows for
% the columns, and the contain the column location of the
% corner nodes.

     [i1,j1] = size(rows);
     [i2,j2] = size(pcolr);
     [i3,j3] = size(pcoll);

%  we pick off the elments of rows and pcolr which
%  identify the location of corner 1.

     p = rows(i1)-1;
     q = pcoll(j3);

%  generate any required  local constants

     cc = (2 - 2*rho(1)-2*rho(2));

%  the updated values of the corner nodes for u and v are
%  calculated


   un(p,q) = cc*uc(p,q) - uo(p,q)...
     + rho(1)*(uc(p,q-1) + uc(p,q+1))...
      + rho(2)*(uc(p-1,q) + uc(p+1,q))...
       - 2*rho(3)*(vc(p-1,q-1) + vc(p+1,q+1) + 2*vc(p,q))...
        + 2*rho(3)*(vc(p,q+1) + vc(p,q-1) + vc(p+1,q) + vc(p-1,q));


   vn(p,q) = cc*vc(p,q) - vo(p,q)...
   + rho(2)*(vc(p,q-1) + vc(p,q+1))...
     + rho(1)*(vc(p-1,q) + vc(p+1,q))...
```

```
      - 2*rho(3)*(uc(p-1,q-1) + uc(p+1,q+1) + 2*uc(p,q))...
       + 2*rho(3)*(uc(p,q+1) + uc(p,q-1) + uc(p+1,q) + uc(p-1,q));

%  the same process is repeated for corner 4 below

     p1 = rows(j1)+1;
     q1 = pcolr(i2);


    un(p1,q1) =cc*uc(p1,q1) - uo(p1,q1)...
    + rho(1)*(uc(p1,q1-1) + uc(p1,q1+1))...
     + rho(2)*(uc(p1-1,q1) + uc(p1+1,q1))...
      - 2*rho(3)*(vc(p1-1,q1-1) + vc(p1+1,q1+1) + 2*vc(p1,q1))...
       + 2*rho(3)*(vc(p1,q1+1) + vc(p1,q1-1) + vc(p1+1,q1) + vc(p1-1,q1));

   vn(p1,q1) =cc*vc(p1,q1) - vo(p1,q1)...
    + rho(2)*(vc(p1,q1-1) + vc(p1,q1+1))...
     + rho(1)*(vc(p1-1,q1) + vc(p1+1,q1))...
      - 2*rho(3)*(uc(p1-1,q1-1)+uc(p1+1,q1+1)+2*uc(p1,q1))...
       + 2*rho(3)*(uc(p1,q1+1) + uc(p1,q1-1) + uc(p1+1,q1) + uc(p1-1,q1));
```

```
%  function tcor24 - treatment of corners 2 and 4
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
%
%
%  tcor24 applies the elastic wave equation at
%  corners 2 and 4 as per fig( ) since they use the
%  same difference formula.
%  the corner nodes are located (2 first then 4)
%  identified as p and q (p1 ,q1 in the case of corner 4)
%  the necessary neighbours are picked off from the u and v
%  matrices and weighted accordingly and the new updated values
%  for u and v are  computed.


   function  [un,vn] = tcor24(uc,vc,uo,vo,un,vn,rows,pcoll,pcolr,rho);

% variables
% un : updated values of u    vn : updated values of v
% uc : current values of u    vc : current values of v
% uo : old values of u        vo : old values of v
% rho : vector containing global constants
% rows : used to identify the elements of the matrix which
% are zero for all times they also contain the row location of
% the corner nodes.
% pclor and pcoll carries out the same function as rows for
% the columns, and the contain the column location of the
% corner nodes.


       [i1,j1] = size(rows);
       [i2,j2] = size(pcolr);
       [i3,j3] = size(pcoll);

%  we pick off the elements of rows and pcolr which
%  identify the location of corner 2.

       p = rows(i1)-1;
       q = pcolr(i2);

%  generate any required  local constants

       cc = (2 - 2*rho(1)-2*rho(2));


%  the updated values of the corner nodes for u and v are
%  calculated

    un(p,q) = cc*uc(p,q) - uo(p,q)...
     + rho(1)*(uc(p,q-1) + uc(p,q+1))...
      + rho(2)*(uc(p-1,q) + uc(p+1,q))...
       + 2*rho(3)*(vc(p+1,q-1) + vc(p-1,q+1) + 2*vc(p,q))...
        - 2*rho(3)*(vc(p,q+1) + vc(p,q-1) + vc(p+1,q) + vc(p-1,q));


   vn(p,q) = cc*vc(p,q) - vo(p,q)...
   + rho(2)*(vc(p,q-1) + vc(p,q+1))...
     + rho(1)*(vc(p-1,q) + vc(p+1,q))...
      + 2*rho(3)*(uc(p+1,q-1) + uc(p-1,q+1) + 2*uc(p,q))...
       - 2*rho(3)*(uc(p,q+1) + uc(p,q-1) + uc(p+1,q) + uc(p-1,q));
```

```
% the same process is repeated for corner 4 below

      p1 = rows(j1)+1;
      q1= pcoll(j3);


    un(p1,q1) = cc*uc(p1,q1) - uo(p1,q1)...
     + rho(1)*(uc(p1,q1-1) + uc(p1,q1+1))...
      + rho(2)*(uc(p1-1,q1) + uc(p1+1,q1))...
       + 2*rho(3)*(vc(p1+1,q1-1) + vc(p1-1,q1+1) + 2*vc(p1,q1))...
        - 2*rho(3)*(vc(p1,q1+1) + vc(p1,q1-1) + vc(p1+1,q1) + vc(p1-1,q1));


   vn(p1,q1) = cc*vc(p1,q1) - vo(p1,q1)...
   + rho(2)*(vc(p1,q1-1) + vc(p1,q1+1))...
     + rho(1)*(vc(p1-1,q1) + vc(p1+1,q1))...
      + 2*rho(3)*(uc(p1+1,q1-1) + uc(p1-1,q1+1) + 2*uc(p1,q1))...
       - 2*rho(3)*(uc(p1,q1+1) + uc(p1,q1-1) + uc(p1+1,q1) + uc(p1-1,q1));
```

```
%  function trid - tridiagonal
%  written by:  Lt. Hugh Mc Bride USNR
%  date      : Mar 92
%
%  trid generates a tridiagonal matrix for the
%  radiation boundary condition of the fluid .
%  the elements of the sub and super diagonal are (dts/kf*dxs)^2
%  the main diagonal component is 2 - 4(dts/kf*dxs)^2
%  the (n,2) and (1,n-1) contain (dts/kf*dxs)^2 to satisfy the
%  periodic boundary conditions.

  function  m = trid(dxs,dts,n,kf)

% variables
%  dxs : scaled spacing
%  dts : scaled time step
%  n   : dimension of matrix
%  kf  : scaled constant



%  an identity matrix for the elements of the main diagonal

           d1 = eye(n);

%  the sub and super diagonals

           d2 = diag(ones(n-1,1),1) + diag(ones(n-1,1),-1);

%  the required co-efficients

          rho  = dts/(kf*dxs); rho = (rho^2);

%  generates the required matrix

          d =2*d1 -4*rho*d1 +rho*d2;

          d(1,n-1)= rho ; d(n,2) = rho;


 m = d;
```

```
% function ucnx - u coefficients/constants for the boundary facing the
% negative x direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(-1,0) for the corresponding u
% and v values.
%

function cunx = ucnx(kl,kt,dxs,dts)

% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cunx
% for use in bndfnx


 c1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

   c2  = 2*dts^2/(dxs^2*kl^2);

     c3 = (dts^2)/(dxs^2*kt^2);

       c4 = -(dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

         c5 = (dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cunx = [c1 c2 c3 c4 c5];
```

```
% function ucny - u coefficients/constants for the boundary facing the
% negative y direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(0,-1) for the corresponding u
% and v values.
%

function cuny = ucny(kl,kt,dxs,dts)


% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cuny
% for use in bndfny

 c1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

  c2  = 2*(dts^2/(dxs^2*kt^2));

   c3 = (dts^2)/(dxs^2*kl^2);

     c4 = -(dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

       c5 = (dts^2/(2*dxs^2))*((3/kt^2)-(1/kl^2));

  cuny = [c1 c2 c3 c4 c5];
```

```
% function ucpx - u coefficients/constants for the boundary facing the
% positive x direction.
% written by: Lt. Hugh Mc Bride USNR
% date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(1,0) for the corresponding u
% and v values.
%
function cupx = ucpx(kl,kt,dxs,dts)

% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cupx
% for use in bndfpx


 c1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

  c2  = 2*(dts^2/(dxs^2*kl^2));

   c3 = (dts^2)/(dxs^2*kt^2);

    c4 = (dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

      c5 = -(dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cupx = [c1 c2 c3 c4 c5];
```

```
% function ucny - u coefficients/constants for the boundary facing the
% positive y direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(0,1) for the corresponding u
% and v values.
%

function cupy = ucpy(kl,kt,dxs,dts)

% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cupy
% for use in bndfpy


 c1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

  c2  = 2*(dts^2/(dxs^2*kt^2));

   c3 = (dts^2)/(dxs^2*kl^2);

     c4 = (dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

       c5 = (dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cupy = [c1 c2 c3 c4 c5];
```

```
% function vcnx - v coefficients/constants for the boundary facing the
% negative x direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(-1,0) for the corresponding u
% and v values.
%

function cvnx = vcnx(kl,kt,dxs,dts)


% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cvnx
% for use in bndfnx


 d1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

   d2  = 2*(dts^2/(dxs^2*kt^2));

     d3 = (dts^2)/(dxs^2*kl^2);

       d4 = -(dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

         d5 = (dts^2/(2*dxs^2))*((3/kt^2)-(1/kl^2));

  cvnx = [d1 d2 d3 d4 d5];
```

```
% function vcny - v coefficients/constants for the boundary facing the
% negative y direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(0,-1) for the corresponding u
% and v values.
%


function cvny = vcny(kl,kt,dxs,dts)

% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cvny
% for use in bndfny


 d1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

   d2  = 2*(dts^2/(dxs^2*kl^2));

     d3 = (dts^2)/(dxs^2*kt^2);

       d4 = -(dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

         d5 = (dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cvny = [d1 d2 d3 d4 d5];
```

```
% function vcpx - v coefficients/constants for the boundary facing the
% positive x direction.
% written by: Lt. Hugh Mc Bride USNR
% date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(1,0) for the corresponding u
% and v values.
%
function cvpx = vcpx(kl,kt,dxs,dts)

% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cvpx
% for use in bndfpx


  d1 = 2 -2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

   d2  = 2*(dts^2/(dxs^2*kt^2));

     d3 = (dts^2)/(dxs^2*kl^2);

       d4 = (dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

         d5 = (dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cvpx = [d1 d2 d3 d4 d5];
```

```
% function vcny - v coefficients/constants for the boundary facing the
% negative y direction.
%  written by: Lt. Hugh Mc Bride USNR
%  date      : Apr 92
% provides the coefficients developed by applying the Ilan and Lowenthal
% technique to the boundary with unit normal(0,1) for the corresponding u
% and v values.
%


function cvpy = vcpy(kl,kt,dxs,dts)
% variables
% kl - scaled longitudinal speed
% kt - scaled transverse speed
% dxs - scaled spacing
% dts - scaled time step

% the coefficients are calculated and stored in the vector cvpy
% for use in bndfpy


 d1 = 2 - 2*(dts^2/dxs^2)*((1/kl^2)+(1/kt^2));

   d2  = 2*(dts^2/(dxs^2*kl^2));

    d3 = (dts^2)/(dxs^2*kt^2);

     d4 = (dts^2/(2*dxs^2))*((1/kl^2)-(1/kt^2));

       d5 = -(dts^2/(2*dxs^2))*((1/kl^2)-(3/kt^2));

  cvpy = [d1 d2 d3 d4 d5];
```

REFERENCES

Fuyuki, M., and Matsumoto, Y., "Finite Difference Analysis of Rayleigh Wave Scattering at a Trench," *Bulletin of the Seismological Society of America*, v. 70, December 1980.

Ilan A., and Lowenthal, D., "Instability of Finite Difference Schemes due to Boundary Conditions in Elastic Media," *Geophysical Prospecting*, v. 24, 1976.

Interview between Prof. C. L. Scandrett, Department of Mathematics, Naval Postgraduate School, and the author, 30 May, 1992.

Kinsler, L. E., Frey, A. R., Coppens, A. B., Sanders, J. V., *Fundamentals of Acoustics, Third Edition*, John Wiley & Sons, 1982.

Lalanne M., Berther P., and Der Hagopian, J., *Mechanical Vibration for Engineers, First Edition,*John Wiley & Sons, 1982.

Scandrett, C. L., and Kreigsmann, G. A., "Decoupling Approximation Applied to an Infinite Array of Fluid Loaded Baffled Membranes," to be submitted for publication.

INITIAL DISTRIBUTION LIST

1. Library (Code 52) 2
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Defense Technical Information Center 2
   Cameron Station
   Alexandria, VA 22304-6145

3. Prof. Clyde Scandrett 1
   Code MA/Sd
   Department of Mathematics
   Naval Postgraduate School,
   Monterey, CA 93943-5000

4. Prof. Van Emden Henson 1
   Code MA/Hv
   Department of Mathematics
   Naval Postgraduate School,
   Monterey, CA 93943-5000

5. Prof. Young W. Kwon 1
   Code ME/Kw
   Department of Mechanical Engineering
   Naval Postgraduate School,
   Monterey, CA 93943-5000

6. LT Hugh McBride 2
   201 Glenwood Circle, Apt 26D
   Monterey, CA 93940

7. Chairman 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, CA 93943