INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Ultrasound technique for tissue differentiation

Kachroo, Pushkin, M.S.

Rice University, 1990
RICE UNIVERSITY

ULTRASOUND TECHNIQUE FOR TISSUE DIFFERENTIATION

by

PUSHKIN KACHROO

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

APPROVED THESIS COMMITTEE

John B. Cheatham, Jr., Professor of Mechanical Engineering, Director

Thomas A. Krouskop, REC, TIRR, Baylor College of Medicine, Co-Director

John E. Akin, Professor of Mechanical Engineering

Ian Walker, Assistant Professor of Electrical Engineering

Houston, Texas

April, 1990
Abstract

"Ultrasound Technique for Tissue Differentiation"

Pushkin Kachroo

Pressure sores are a serious medical problem, which cause a great deal of economic and physical damage. The old method of skin color identification is not quantitative and is usually unreliable. Among various methods which are being tried for early identification of pressure sores, ultrasonic technique is one. A method has been developed which helps to calculate the stiffness of the tissue under the skin at various depths. This method consists of a perturbator which vibrates on the surface of the skin with a known amplitude and frequency. The wave generated on the surface behaves as an exponentially decaying sinusoidal wave with respect to time and downward distance for a constant stiffness of the medium. With the help of an ultrasonic transducer embedded on the perturbator head, displacement can be calculated at each depth for various times. Using computer software, these displacements are noted for various times to find out the amplitude for each depth which is then plotted on the computer screen. According to the changes in slope of the curve, different tissue regions are identified and their moduli are compared. A computer simulation to show the expected amplitude-depth plot for a possible pressure sore region is done.
Acknowledgements

The author wishes to express sincere appreciation for the advice and assistance of Dr. John B. Cheatham, Jr. and Dr. Thomas A. Krouskop throughout this research. The author is extremely grateful to Kumar and Vijay Krishen, Sunil and Lovely Fotedar, P.L, Sadhna and Dhananjaya Kachroo, and Anjala Krishen for her help and support in organizing this thesis.

The author would also like to express gratitude to the Mobility Foundation for sponsoring this research project.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>x</td>
</tr>
<tr>
<td>1. Introduction</td>
<td></td>
</tr>
<tr>
<td>1.1 Background</td>
<td></td>
</tr>
<tr>
<td>1.1.1 Pressure sore</td>
<td></td>
</tr>
<tr>
<td>1.1.2 Importance</td>
<td></td>
</tr>
<tr>
<td>1.1.3 Current research thrusts</td>
<td></td>
</tr>
<tr>
<td>1.2 Description of Present Work</td>
<td></td>
</tr>
<tr>
<td>1.2.1 Statement of the research problem</td>
<td></td>
</tr>
<tr>
<td>1.2.2 Objectives</td>
<td></td>
</tr>
<tr>
<td>1.2.3 Apparatus</td>
<td></td>
</tr>
<tr>
<td>1.2.4 Procedure</td>
<td></td>
</tr>
<tr>
<td>1.3 Outline of the Present Work</td>
<td></td>
</tr>
<tr>
<td>2. Methodology</td>
<td></td>
</tr>
<tr>
<td>2.1 Equipment</td>
<td></td>
</tr>
<tr>
<td>2.2 Procedure</td>
<td></td>
</tr>
<tr>
<td>2.2.1 Equipment set up</td>
<td></td>
</tr>
<tr>
<td>2.2.2 Software set up</td>
<td></td>
</tr>
<tr>
<td>2.2.3 Subject set up</td>
<td></td>
</tr>
<tr>
<td>2.2.4 Data collection</td>
<td></td>
</tr>
</tbody>
</table>
2.2.5 Data analysis

3. Tissue Differentiation

3.1 Visco-elastic material

3.1.1 Waves travelling in a visco-elastic material

3.1.2 Theoretical calculation of the elastic modulus

3.2 Condition near the boundary

3.2.1 Wave travelling in a layered elastic material near a boundary

3.2.2 Theoretical calculation of the elastic modulus near the boundary

3.2.3 Multiple reflections

3.3 Thin medium

3.3.1 Wave propagation in a multiple layered medium through a thin material

4. Instrumentation

4.1 Doppler Displacement Instrument

4.1.1 Description of operation

4.1.2 Specifications

4.1.3 Front panel controls

4.1.4 Rear panel controls

4.2 Ultrasound/Perturbator head

4.3 Perturbator box

4.4 Mechanical support

4.5 Computer

5. Software Development

5.1 Data collection

5.1.1 PERT.DATA/PERT.BIN
## List of Tables

<table>
<thead>
<tr>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Doppler Displacement Instrument Specifications</td>
<td>28</td>
</tr>
<tr>
<td>6.1 Modulus</td>
<td>42</td>
</tr>
<tr>
<td>7.1 Skin Table</td>
<td>45</td>
</tr>
<tr>
<td>7.2 Fat Table</td>
<td>46</td>
</tr>
<tr>
<td>7.3 Blood Table</td>
<td>47</td>
</tr>
<tr>
<td>7.4 Muscle Table 1</td>
<td>48</td>
</tr>
<tr>
<td>7.5 Muscle Table 2</td>
<td>49</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Displacement of the wave with distance and time</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Amplitude of the wave with distance and time</td>
<td>11</td>
</tr>
<tr>
<td>3.3</td>
<td>Fixed time plot</td>
<td>12</td>
</tr>
<tr>
<td>3.4</td>
<td>Near boundary condition</td>
<td>14</td>
</tr>
<tr>
<td>3.5</td>
<td>Boundary amplitude crest</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>Multi-boundary condition</td>
<td>19</td>
</tr>
<tr>
<td>3.7</td>
<td>Multi-crests</td>
<td>20</td>
</tr>
<tr>
<td>3.8</td>
<td>Waves on a thin boundary</td>
<td>21</td>
</tr>
<tr>
<td>3.9</td>
<td>Small crest due to thin boundary</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Perturbator equipment assembly</td>
<td>25</td>
</tr>
<tr>
<td>4.2</td>
<td>Side elevation of the perturbator head</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Plan of the perturbator head</td>
<td>32</td>
</tr>
<tr>
<td>4.4</td>
<td>Vibration of the perturbator head versus time</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>Software flowchart</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Normal curve and idealized curve</td>
<td>40</td>
</tr>
<tr>
<td>6.1</td>
<td>Modulus versus percentage increase in amplitude</td>
<td>43</td>
</tr>
<tr>
<td>7.1</td>
<td>Fat chart</td>
<td>50</td>
</tr>
<tr>
<td>7.2</td>
<td>Muscle chart</td>
<td>51</td>
</tr>
<tr>
<td>7.3</td>
<td>Blood chart</td>
<td>52</td>
</tr>
<tr>
<td>A.1</td>
<td>Stress-Force relationship</td>
<td>60</td>
</tr>
<tr>
<td>A.2</td>
<td>Stress distribution</td>
<td>60</td>
</tr>
<tr>
<td>A.3</td>
<td>Strain development</td>
<td>61</td>
</tr>
<tr>
<td>A.4</td>
<td>Shear relationship</td>
<td>62</td>
</tr>
</tbody>
</table>
A.5 Force equilibrium 64
B.1 Amplitude depth curve 73
B.2 Log amplitude depth curve 73
D.1 Plots for 20% amplitude increase 111
D.2 Plots for 40% amplitude increase 112
D.3 Plots for 60% amplitude increase 113
D.4 Plots for 80% amplitude increase 114
### Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1, x_2, x_3 )</td>
<td>Rectangular coordinates</td>
</tr>
<tr>
<td>( X_1, X_2, X_3 )</td>
<td>Displacement in the direction of ( x_1, x_2, x_3 ) axes</td>
</tr>
<tr>
<td>( P_{ij} )</td>
<td>Stress perpendicular to ( X_i ) parallel to ( X_j )</td>
</tr>
<tr>
<td>( s_{ij} )</td>
<td>Strain component</td>
</tr>
<tr>
<td>( v )</td>
<td>Poisson's ratio</td>
</tr>
<tr>
<td>( E )</td>
<td>Modulus of elasticity in tension</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Shearing strain (also used for curve fitting as a variable)</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Shearing stress</td>
</tr>
<tr>
<td>( G )</td>
<td>Modulus of rigidity or modulus of elasticity in shear</td>
</tr>
<tr>
<td>( P )</td>
<td>Compressional stress</td>
</tr>
<tr>
<td>( P_{ij} )</td>
<td>Deviotoric stress</td>
</tr>
<tr>
<td>( s )</td>
<td>Compressional strain</td>
</tr>
<tr>
<td>( s_{ij} )</td>
<td>Deviatory strain</td>
</tr>
<tr>
<td>( f_i )</td>
<td>Force/unit volume in the ( X_i ) direction</td>
</tr>
<tr>
<td>( m )</td>
<td>Mass</td>
</tr>
<tr>
<td>( F )</td>
<td>Force</td>
</tr>
<tr>
<td>( c, u )</td>
<td>Velocity</td>
</tr>
<tr>
<td>( t )</td>
<td>Time</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Density</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>Perturbation parameter</td>
</tr>
<tr>
<td>( \omega )</td>
<td>Angular frequency</td>
</tr>
<tr>
<td>( A )</td>
<td>Amplitude</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Variable for curve fitting</td>
</tr>
<tr>
<td>( R_1, R_2 )</td>
<td>Specific acoustic impedance</td>
</tr>
</tbody>
</table>
$A_r$ ratio between reflected and incident amplitudes

$R, R_\phi, R$ wave energy
1 INTRODUCTION

Research in the area pertaining to the interaction of ultrasound with biological systems involving ultrasound waves generated by man-made devices began in the 1920's (Fry).

1.1 BACKGROUND

In this section we provide the motivation for this research and present developments pertinent to it.

1.1.1 Pressure Sore: Pressure sores have been and remain a serious problem in the management and rehabilitation of the chronically ill patient (Nola). These sores are a common and serious problem facing all bedridden and wheelchair confined patients. The primary cause of pressure sores is unrelieved pressure, shearing force, or both, leading to ischemia of localized areas of skin and subcutaneous tissue, usually overlying weight-bearing bony prominences (sacrum, ischium, greater trochanter) (Reichel) (Nola). Experimental studies have not only shown that pressure is the main factor, but also that there exists an inverse relationship between the amount of pressure applied and the length of time of application in the production of pressure sores (Kosiak) (Husain) (Lindan). The incidence of pressure sores among paraplegic and quadriplegic patients ranges from 25 to 85%, depending on the medical and nursing care received (Dinsdale) (Hackler). An estimated 7 to 8% of deaths in this group can be attributed directly to pressure sores (Breithaupt, et al) (Freed, et al). Prolongation of hospitalization, amount of suffering and delay of rehabilitation are definitely mentionable on
account of the sores. Not only hospitalized patients are at risk, even persons who are simply inactive and have other medical problems such as poor blood circulation or incontinence are at risk of developing ulcers. If the degree of pressure applied to a tissue is high enough to interfere with tissue circulation and if the duration is also long enough, it causes skin breakdown (Lindan). Thus people who already have bad circulation are at higher risk of getting pressure sores.

Obviously pressure sores have bad effects economically. In a study done for the year 1966, it was discovered that each ulcer increased the cost of medical care by $5000 (Schell, et al). In a study done for the year 1969, it was noted that insurance companies allotted 25% of the anticipated expenses of a spinal cord injury for the treatment of pressure sores (Griffith). A study done for the year 1979 estimated the expense created by a single sore close to $14,000 (Vistines). Contemporary prices for medical care would obviously make the dollar figures considerably higher.

1.1.2 Importance: Early identification of tissue damage is of considerable importance. In many cases, early clinical intervention can prevent breakdown (Satsue Hagisawa, et al). Early detection of tissue distress is routinely practiced by conducting regular inspection of skin color at sites over bony prominences, but these observations have limited significance.
1.1.3 Current Research Thrusts

The observation of skin color, for the early detection of the formation of pressure sores, is non-quantitative and non-specific. Research has been done using different approaches in an attempt to detect the muscle damage early.

Hagisawa, et al, conducted a study which was undertaken to seek biochemical indicators in blood which would indicate the onset of a pressure sore. Their work showed the potential of serum creative phosphokinase (CPK) as a systematic indicator of the sore development. Another project (Bennett, et al) was carried out in order to study the skin blood flow in seated geriatric patients.

Dowd, et al worked on skin viability measurement using the transcutaneous oxygen monitor.

Thermographic study by Lewis, et al used temperature of the skin as a criterion.

Ultrasound has been used in some studies also. Bhagat, et al did work which dealt with attenuation and back scattering in freshly excised animal tissues. One in vivo ultrasound experiment done by Chu, et al also used attenuation and the back scattering property of the tissue.

1.2 DESCRIPTION OF PRESENT WORK

1.2.1 Statement of the Research Problem

Thus far, there have been no studies to determine the relationship between the health and the mechanical properties of human tissue. Although tissue breakdown can occur for many reasons, pressure bearing areas are most likely to develop ulcers. The most common reason for pressure sores is a lack of blood supply to the tissue, causing
the tissue to die and stiffen (high elastic modulus). Ultrasound
technology is used in the present work to differentiate between tissue
types and then highlight the region of a possible development of a
pressure sore.

1.2.2 Objectives
(1) To modify the existing ultrasound technology in order to measure the
stiffness of the various tissue types.
(2) To develop a theoretical basis for pressure sore identification.
(3) To pave the way for future work in the same domain.

1.2.3 Apparatus
The ultrasound system developed for this study has the following
main components.
(1) tissue vibrator (perturbation) having three ultrasonic transducers
embedded in it
(2) perturbator box to control vibrations of a specific frequency for the
tissue vibrator
(3) computer, so as to analyze and collect a set of data
(4) Doppler box, in order to communicate between the perturbator and
the computer

1.2.4 Procedure
The perturbator creates vibrations on the surface of the skin. The
transducer finds the amplitude of the vibrations at different depths, which
is then plotted on the computer screen. The software then calculates the
elastic moduli, differentiates between different tissue types, and finds out
if there is a development towards a pressure sore formation and if so, at which depth.

1.3 OUTLINE OF THE PRESENT WORK

The thesis is divided into chapters. The topics with which the chapters deal are given below:

Chapter 1: INTRODUCTION
- explains the importance and significance of the research

Chapter 2: METHODOLOGY
- explains how the experiment is conducted

Chapter 3: TISSUE DIFFERENTIATION
- explains the best-fit technique used for amplitude versus depth data and the theoretical development for tissue differentiation

Chapter 4: INSTRUMENTATION
- describes the various parts of the instrumentation

Chapter 5: SOFTWARE DEVELOPMENT
- describes the various data collection and data analysis programs used

Chapter 6: PRESSURE SORE SIMULATION
- computer simulation of a pressure sore based on high stiffness assumption
Chapter 7: STATISTICAL ANALYSIS
- statistical analysis done on data taken on various subjects to determine
ranges of elasticity for different tissues

Chapter 8: SUMMARY AND CONCLUSION
- discusses the conclusion of this research and possibilities for further
research
2 METHODOLOGY

The set up and procedure of the experiment are described in this chapter.

2.1 EQUIPMENT

The following equipment is needed for the experimental set up:

Ultrasonic displacement box, ultrasound head (perturbator), perturbator box
(and board fitted in the computer), computer with the computer programs, data
disk, ultrasound gel, table, connecting wires, etc.

2.2 PROCEDURE

The procedure is divided into the following steps.

2.2.1 Equipment set up

The various connections to be made are as follows:

(1) Perturbator box to the ultrasound head
(2) Ultrasound displacement box to the ultrasound head
(3) Ultrasound displacement box to the computer via a ribbon cable and
the perturbator board
(4) Perturbator board in slot 4
(5) Clock in slot 5

The ultrasound displacement box controls are set as follows:

(1) Volume - as desired
(2) Depth - external
(3) Signal level - almost full scale
(4) Polarity - away
(5) Output calibration - operate
(6) Full scale - 10 mm
(7) Output limit - 8 bits
(8) Reset point - 5 volts

2.2.2 Software set up
Insert the perturbator disk in the disk drive 1 and the date disk in drive 2 before switching the computer on. Run the program PERT.DATA.BAS. Input "yes" to "Collect data?" and for "full scale setting." Next, user can decide the range in cm for the depth measurements. The computer is ready at this point for data collection.

2.2.3 Subject set up
The subject sits on a stool and his/her arm is kept on the support table over a soft cushion. The perturbator head is put on the skin over the area of interest. A hydraulic jack is used to adjust the head of the perturbator such that it is just touching the subject's skin. The perturbator box is switched on which then causes the perturbator head to vibrate with a frequency of 10Hz ± 0.5 Hz and an amplitude of 2.54 mm.

2.2.4 Data Collection
By returning to the prompt of the PERT.DATA.BAS, the data collection is done until the end of the specified depth is obtained or a return is pressed in the middle. Data is collected in two columns. The first column is that of depth starting from 0 and being incremented by 0.015 cm. successively. The second column represents the corresponding amplitude of the vibration of the tissue for that depth. During the prompt, "Save Data?" data can be saved if wished, by giving a name to the data file on the next prompt. More data can be collected by inputting "yes" to
"Collect more data?". Ten sets of data can be collected at one time. The
PERT.COMB program is used to combine the sets of data in the files
which are used later for analysis.

2.2.5 Data analysis
Computer software is available to do analysis of the data in order to
calculate the elasticity of different regions under the skin, perform tissue
differentiation, and identify possible pressure sore formation region.
3 TISSUE DIFFERENTIATION

3.1 VISCO-ELASTIC MATERIAL

Equation (A.15) is the basic equation used here to develop a method by which to calculate the elastic modulus for a medium.

3.1.1 Waves travelling in a visco-elastic material

The solution to (A.15) for a visco-elastic fluid can be assumed to be of the form

\[ X_1 = X_F \cos(\omega t) + Y_F \sin(\omega t) \]  --(3.1)

where \( X_F \) and \( Y_F \) are time independent. Using (A.1), we can rewrite (A.15) as

\[ -\rho w^2 X_1 = (K + \frac{4}{3}G) \frac{\partial^2 X_1}{\partial x_1^2} \]  --(3.2)

By ignoring \( Y_F \), we get

\[ (K + \frac{4}{3}G) = \frac{-\rho w^2 A}{\frac{\partial^2 A}{\partial x_1^2}} \]  --(3.3)

where \( A \) is the amplitude of the wave.

Using equation (A.6), equation (3.3) can be further approximated to

\[ E = \frac{-3\rho w^2 A}{2 \frac{\partial^2 A}{\partial x_1^2}} \]  --(3.4)

Due to absorption in the viscous media, the amplitude of the wave decays exponentially with \( x_1 \), which is a solution of (3.4). So \( A \) can be written as
\[ A = \alpha e^{-\gamma x} \] \hfill (3.5)

where \( \alpha \) and \( \gamma \) are constants and \( x \) denotes the direction \( x_1 \).

Thus, for a medium with constant \( E \), the wave solution, if plotted in three dimensions with \( x \) and \( y \) axes as \( x_1 \) and \( t \) (time), will look like Figure 3.1.

**Figure 3.1** Displacement of the Wave with Time and Distance

Figure 3.2 shows the envelope of the first plot so that the exponential decay of the amplitude with respect to the \( x_1 \) direction is more obvious.

**Figure 3.2** Amplitude of the Wave with Distance and Time
If we fix time and draw the amplitude versus depth, we will obtain a plot as shown in Figure 3.3.

![fixed-time plot]

Figure 3.3 Fixed Time Plot

3.1.2 Theoretical Calculation of the Elastic Modulus

Best squares fit techniques as described in Appendix A can be used to solve equation (3.5) to calculate \( \alpha \) and \( \gamma \).

Using equation (3.5) in equation (3.4) we get

\[
E = \frac{-3\rho w^2}{2\gamma^2}
\]

3.2 CONDITION NEAR THE BOUNDARY

3.2.1 Wave travelling in a layered elastic material near a boundary

When a wave travelling through one material impinges on a boundary between it and a second medium, part of the wave is refracted to the next medium (if the incident wave is at an angle to the normal of the surface) while part is reflected back into the first medium, usually with a phase change. Boundary conditions should be taken care of for the wave (Brekhovskikh).
The reflected wave travels back to the first medium and the amplitude of the reflected wave is related to the incident wave as
\[ A_r = \frac{R_1 - R_2}{R_1 + R_2} \]  --(3.6)

where
\[ R_1 = \rho_1 c_1 \]
\[ R_2 = \rho_2 c_2 \]
\[ \rho = \text{density of each material} \]
\[ c = \text{velocity} \]
\[ A_r = \text{ratio between reflected and incident amplitudes} \]
\[ R \text{ is known as the "specific acoustic impedance".} \]

The subscripts 1 and 2 indicate the media.

As long as the wave travels in the same medium, the energy is proportional to the amplitude squared. Therefore,
\[ R = R_0 \left( \frac{\rho_1 c_1 - \rho_2 c_2}{\rho_1 c_1 + \rho_2 c_2} \right)^2 \]  --(3.7)

where
\[ R = \text{reflected energy} \]
\[ R_0 = \text{incident energy} \]

In a reflection system, the energy passes through an interface twice, once in each direction of travel (Carlin). Thus, the energy passed through the boundary is
\[ R' = (1-R)^2 \]  --(3.8)

where \( R' \) is the energy passed through.

3.2.2 Theoretical calculation of the elastic modulus near the boundary
Consider the case when the reflected wave dies out passing through a single constant elastic modulus material, as shown in the following figure.

![Graph showing incident and reflected waves](image)

**Figure 3.4 Near Boundary Condition**

We see from this figure that the reflected wave dies within a depth of 'd'. The form of the incident wave is

\[ A_1 = \alpha_1 e^{-\gamma x} \]  

---(3.9)

The reflected wave dies out exponentially. The form of the reflected wave is

\[ A_2 = a_0 + \alpha_2 e^{\gamma x} \]  

---(3.10)

Notice that \( \gamma \) is same for the incident as well as the reflected wave. This is due to the fact that \( \gamma \) is a measure of the elastic properties of the medium and since both waves are travelling in the same medium, it is natural to use the same \( \gamma \).
Using the boundary conditions on (3.10) at $x_1$ and $x_2$ we determine that

\[ 0 = a_0 + \alpha_2 e^{\gamma x_1} \]
\[ a_0 = -\alpha_2 e^{\gamma x_1} \quad \text{---(3.11)} \]

and

\[ A_r = a_0 + \alpha_2 e^{\gamma x_2} \]
\[ A_r = \alpha_2 \left( e^{\gamma x_2} - e^{\gamma x_1} \right) \]
\[ \alpha_2 = \frac{A_r}{\left( e^{\gamma x_2} - e^{\gamma x_1} \right)} \quad \text{---(3.12)} \]

Thus the overall equation for the region between $x_1$ and $x_2$ becomes

\[ A = A_1 + A_2 \]
\[ A = \alpha_1 e^{-\gamma x} - \alpha_2 e^{\gamma x_1} + \alpha_2 e^{\gamma x} \]

\[ A = \alpha_1 e^{-\gamma x} = \frac{A_r \left( e^{\gamma x} - e^{\gamma x_1} \right)}{\left( e^{\gamma x_2} - e^{\gamma x_1} \right)} \quad \text{---(3.13)} \]

There are three unknowns: $\alpha_1$, $\gamma$, and $A_r$ in this equation. Figure 3.5 shows how the plot of $A$ versus $x$ should look.
Figure 3.5 Boundary Amplitude Crest

The best way to solve equation (3.13) for $\alpha_1$, $\gamma$, and $A_r$ would be to employ one of the various techniques of best-square-fit but as is obvious from the nature of equation (3.13), the process would be cumbersome.

Another method, which, albeit not as accurate as the best-square-fit, is to take the values of $A_1$ at $x_1$, $A_2$ at $x_2$, and the point of maxima (i.e. $A_3$ at $x_3$) between $x_1$ and $x_2$ and put the values in equation (3.13) to extrapolate two equations and differentiate equation (3.13) to get the third equation.

By differentiating equation (3.13), we have

$$\frac{dA}{dx} = -\alpha_1 \gamma e^{-\gamma x} + \frac{A_r \gamma e^{\gamma x}}{\left(e^{\gamma x_2} - e^{\gamma x_1}\right)}$$

--(3.14)

Now

$$\frac{dA}{dx} = 0 \quad \text{at } x_3$$

We can solve equation (3.14) by taking $e^{\gamma x}$ as a variable $\nu$ which makes
\[ e^{-\gamma x} = \frac{1}{\nu} \]

Take

\[ k_1 = \frac{A_r \gamma}{\left( e^{\gamma x} - e^{\gamma x_1} \right)} \]

and

\[ k_2 = \alpha_1 \gamma \]

so by solving equation (3.14), we derive

\[ \nu = \frac{\alpha_1 \left( e^{\gamma x} - e^{\gamma x_1} \right)^{1/2}}{A_r} \]

or

\[ x_3 = \frac{1}{2\gamma} \ln \left[ \frac{\alpha_1 \left( e^{\gamma x} - e^{\gamma x_1} \right)}{A_r} \right] \]

These equations can be solved numerically by using the

SQA/MQA method. SQA/MQA gradient method is explained in Appendix

A for two variables and can be extended for the multivariable case (3, to

be specific) as well.

We can use the value of \( A_r \) to get elastic moduli of the media,

depending on the known values.

The velocity of longitudinal waves can be derived by comparing

equation (A.15) with the following longitudinal wave equation

\[ \frac{\partial^2 X_1}{\partial t^2} = c^2 \frac{\partial^2 X_1}{\partial x_1^2} \]

where \( c \) is the velocity of the longitudinal wave, thus
\[ c = \sqrt{\frac{k + \frac{4G}{3}}{\rho}} \]

writing \( k \) and \( G \) in terms of \( E \), we get

\[ c = \sqrt{\frac{E(1-v)}{d(1+v)(1-2v)}} \]  

\[ \text{---(3.16)} \]

where  
\( E \) = elastic modulus  
\( c \) = velocity  
\( v \) = poisson's ratio  
\( d \) = density

Using (3.6) and (3.16) with the value as obtained, we can calculate \( E_1 \) and \( E_2 \) for known densities and poisson's ratio.

3.2.3 Multiple Reflections

Consider the case when the reflected wave goes through more than one layer before its amplitude dies down. Figure 3.6 shows the reflected wave going through two layers.
The amplitude curve as seen is the sum of the incident and reflected amplitudes (being conservative about the phase change on reflection) and it resembles what is shown in the Figure 3.7.
Mathematical deduction can be used as was done in section 3.2.1 to calculate the elastic moduli for the media.

3.3 **THIN MEDIUM**

3.3.1 *Wave propagation in multiple layered medium through a thin material*

The nature of the obstacle and its size decides what kind of reflection will occur when a wave is impinged on it (Carlin). If the object is large in comparison to the wavelength, there is a strong reflected wave coming from it and a definite shadow is cast behind the obstacle. If the obstacle, however, is small (a small fraction of the wavelength), there will be no real reflection and no shadow. Similarly, the dimension of the cross-section of the obstacle determine if the reflection will be large or small.
In other words, there are two important facts that influence reflection. One is the dimension in the direction along the propagation path; the other is the cross-sectional area. It is the dimension along the propagation paths that interrupts the wave, but it is the cross-sectional area that determines the amount of reflection.

Now consider a thin material (with any elastic relationship i.e. linear or nonlinear) in the path of a wave travelling through another medium as shown in the Figure 3.8.

![Figure 3.8 Waves on a Thin Boundary](image)

*Figure 3.8 Waves on a Thin Boundary*
Figure 3.9  Small Crest due to Thin Boundary

The thin material will cause a small reflection (small shadow behind) and thus in the plot of amplitude versus depth, we will see a small crest before the thin material.
4 INSTRUMENTATION

The instrumentation for this project can be broadly consists of:

(1) Doppler displacement instrument
(2) Ultrasound head
(3) Perturbator box
(4) Mechanical supports
(5) Computer

Figure 4.1 shows the assembly for this project.

The crucial part of the project is the Doppler displacement instrument on which the whole experiment is based.

4.1 DOPPLER DISPLACEMENT INSTRUMENT

Doppler displacement instrument is a general term used here which consists of the Doppler box and the transducer embedded in the perturbator head.

4.1.1 Description of operation

Incident acoustic pulse production: 10 MHz crystal oscillator controls the timing for the pulsed doppler displacement instrument. 0.8 μsec pulses are generated by dividing the 10 MHz signal by 512 using a frequency divider. The (PRF) pulse repetition frequency of this signal is 19.53 KHz.

The original 10 MHz signal is mixed with this signal to produce a train of tone bursts, each tone made up of 8 cycles of the 10 MHz signal. These tone bursts are amplified to 20 volts peak-to-peak and applied to the ultrasonic transducer where they are converted to sound pulses.

Reflected acoustic pulse transformation: The incident acoustic pulses, when encountered by an obstacle in its path, are reflected back to the
transducer, where they are converted back to the voltage signal and amplified.

**Phase comparison**: The reflected signal is compared in phase to two signals from the quadrature phase detector. The two signals have a phase difference of 90° representing the phase difference between the echo and 10 MHz signal.

**Motion detection**: The outputs of the phase detector are periodic at 19.53 kHz producing identical echoes with identical phases if there is no motion in the medium. On the other hand, if there is some motion in the medium, echoes from the reflectors will change from pulse-to-pulse and a portion of the phase outputs will vary with each successive burst.

**Depth control**: To detect motion at a specific depth, we use a receiver pulse delayed by the proportional time with respect to the transmit pulse in order to sample the phase signals following each transmission. In doing so, we assume quite safely a linear relationship between the delay time given to the receiver pulse and the depth which we are probing. The delay in this instrument has the range from 1 to 52 μsec (1 to 40 mm) and the receiver pulse width is 0.3 μsec.

**Velocity and direction**: The sample-hold circuit has two signals as its output. They have Doppler shift in frequency between them and 90° phase difference. The amount of shift in the frequency is proportional to the average velocity of the reflectors in the sampling volume and the sign of the phase difference determines the direction of motion, i.e. whether it is towards or away from the transducer.

**Frequency filtering**: The quadrature audio signals are filtered to remove the frequencies lower than 10 Hz and those which are higher than the PRF.
**Figure 4.1** Perturbator Equipment Assembly

**Signal processing:** The filtered signal can be processed by an up/down counter (UDC) or listened to by a speaker. The signal first goes through a polarity reversing switch which controls the type of output (i.e. UDC or
audio). There is a volume potentiometer and a (signal level) potentiometer to control the input levels to the audio amplifier or the UDC.

**Up/Down counter:** The operation of the UDC is described below. First of all, a dual comparator is used to convert the two quadrature audio inputs into square waves with small hysteresis. A dead zone is created around zero due to the hysteresis; hence the signal control is set to keep the background noise within the dead zone and to keep the rest of the signal above it. A zero crossing detector is used which triggers a pulse every time it encounters each positive or negative signal going to zero. The clock input of an 8 bit UDC is connected to these pulses. A flip flop senses the sign of the phase detector to control the direction of the counter. Motion away from the transducer brings an increment to the count and motion towards it does the opposite. For the audio output, a digital to analog converter produces a voltage output.

**Velocity to displacement conversion:** Since there is no constant of integration here from changing velocity to displacement, the output signal has to be kept from crossing some limits. The counter is designed in such a way that it stops counting at zero (for decrementing) and by controlling the panel limits switch, we can have the UDC stop counting up at 64 (1/4), 128 (1/2) or 255 (max).

**Resolution:** The maximum output on the UDC corresponds to 4.8 mm for a 5 mm position which gives a sensitivity of 2 volts/mm and a resolution of 0.02 mm. At 10 mm position only the counts of one channel are added which increases the maximum displacement but reduces the resolution. At 20 mm position, the maximum calculable displacement increases further consequently degrading the resolution more. The output
sensitivity is changed to 1.0 volt/mm for 10 mm position and 0.5 volts/mm for 20 mm position.

**Miscellaneous:** if desired, the velocity signals can also be tape recorded or amplified. The same jacks can be used for playing the recorded signals back so that they can be processed by the UDC. A voltage proportional to the range control is also provided. A calibrator switch helps in calibration of a recording device.

### 4.1.2 Specifications

The specifications of the doppler displacement instrument are described in the following tabular form.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultrasonic frequency</td>
<td>10 MHz, crystal controlled</td>
</tr>
<tr>
<td>Pulse Repetition Frequency</td>
<td>19.53 kHz (10 MHz/512)</td>
</tr>
<tr>
<td>Transmitter Pulse Width</td>
<td>0.4 μsec (Xmit output)</td>
</tr>
<tr>
<td>Receiver Pulse Width</td>
<td>0.2 μsec (Gate output)</td>
</tr>
<tr>
<td>Transmitter Output Burst</td>
<td>15 volts p-p into 50 ohms</td>
</tr>
<tr>
<td>Variable Depth Gate</td>
<td>1 - 40 mm (1 - 50 μsec)</td>
</tr>
<tr>
<td>Audio Bandwidth</td>
<td>1 Hz to 10 kHz</td>
</tr>
<tr>
<td>Velocity Range</td>
<td>0.008 to 80 cm/sec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum Displacement* (mm)</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Resolution in wavelengths</th>
<th>w1/8</th>
<th>w1/4</th>
<th>w1/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution in mm</td>
<td>0.02</td>
<td>0.04</td>
<td>0.08</td>
</tr>
<tr>
<td>Calibration steps (2 VDC)</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Displacement limits</td>
<td>1/4 (6 bits)</td>
<td>1.25</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>2.50</td>
<td>5.00</td>
<td>10.0</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>1/2 (7 bits)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS (8 bits)</td>
<td>5.00</td>
<td>10.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Reset Point</td>
<td>0.62</td>
<td>1.25</td>
<td>2.50</td>
</tr>
<tr>
<td>Output Sensitivity (V/mm)</td>
<td>2.00</td>
<td>1.00</td>
<td>0.50</td>
</tr>
</tbody>
</table>

### Probe connection
Floating and differential

### Controls
- ON/OFF switch and pilot light
- Output (oper-0-2-4-6-8-10 VDC)
- Depth adjustment (0-40 mm)
- Volume control
- Signal level to processor
- Polarity (tward-off-away)
- Full scale disp. (5-10-20 mm)
- Limits (2.5, 5, 10 volts)*
- Reset button*
- Reset point (1.25, 2.5, 5.0 volts)

### Rear Panel Recorder Outputs
- Audio (2 in quadrature) + speaker
- Displacement (0=10 VFS)
- Analog range (0.1 V/cm)

### Oscilloscope Monitor Outputs
- Transmitter pulse (sync)
- Range gate pulse (gate)
- Phase detected echoes (echo)

### Other connectors
- Digital I/O, Reset

* Values depend on setting of Full Scale and Limit switches

---

**Table 4.1** Doppler displacement instrument specifications
4.1.3 Front panel controls

**Power on** - This switch turns on the main power (pilot light glows).

**Probe** - A stereo jack connects the displacement probe to the instrument. The input is differential and isolated from chassis and power line ground.

**Volume** - This potentiometer is for adjustment of audio sound to the speaker.

**Depth** - This is for the determination of the distance from the probe face at which displacement is being sensed. It is variable from approximately 1 to 40 mm with a depth window of about 0.5 mm.

**Signal Level** - This dual potentiometer sets the audio levels to the displacement processor and is used to provide the best sensitivity for the processor while rejecting background noise.

**Polarity** - A three position switch determines whether motion toward the probe or away from the probe is displayed as positive. The off position disconnects the audio signals and stops the display.

**Output Calibrator** - A voltage divider is provided for calibration of an external recording device connected to the displacement output. Voltages of 0, 2, 4, 6, 8, and 10 can be selected. The resolution or distance between steps in the counter output is 0.02, 0.04, or 0.08 mm.

**Full Scale** - A three position switch sets the maximum recordable displacement to 5, 10, or 20 mm corresponding to 10 volts of output. The resolution or distance between steps in the counter output is 0.02, 0.04, or 0.08 mm, respectively.

**Output Limits** - This three position switch limits the counter to 6 bits (2.5 volts), 7 bits (5 volts), or 8 bits (10 volts) with 10 volt allowing a total displacement as set by the 10 volts full scale switch.
Push to Reset - This push button can be used to set the counter to the value determined by the reset point switch.

Reset Point - This switch controls the counter value (1.25, 2.5, or 5 volts) when the reset button is pushed or an external reset command (TTL High) is input via the rear panel Reset or Digital I/O connectors.

4.1.4 Rear panel controls

Audio - These two BNC jacks can be used to record the two audio signals on tape or to play previously taped signals back through the displacement processor. A stereo recorder is needed to record both channels to maintain directional information. With the polarity switch in off position, tapes can be played into the instrument and decoded. While taping, only the polarity, and depth controls affect the audio signals being taped; the other controls do not affect the audio signals. While playing tapes back, the polarity switch should be in off position and all controls except depth will operate on the incoming signals. Signal level may have to be readjusted to insure proper decoding of tape recorded signals.

Depth - A depth output is provided for recording range gate position as set by the front panel depth control. This is useful when depth is adjusted during an experiment or when displacement is to be evaluated as a function of depth. The calibration is 0.1 volts/cm of range.

Displacement - This is the main recorder output from the instrument and is controlled by the front panel output control. The sensitivity is 10 volts full scale with the full scale determined by the front panel full scale switch.
**Speaker** - A built-in speaker is provided for monitoring one channel of the audio signal with the loudness determined by the volume control on the front panel. Changing the polarity from away to toward reverses the audio channel being monitored, and no difference in signal should be heard when this is done. In the off position, no sound will be heard.

**Xmit** - A signal from the transmitter gate is available at this jack and can be used to verify the transmit pulse width or for synchronizing an oscilloscope or other device to the pulse repetition frequency of the pulsed Doppler.

**Gate** - The range gate sampling pulse is available at this jack.

**Echo** - A buffered output from one of the quadrature phase detectors is available at this jack.

**Reset** - A high level TTL logic signal at this jack will reseat the displacement counter to 31, 63, or 127.

**Fuse** - The AC line fuse is 1/2 Amp slow blow.

**AC Line** - 110 VAC power is applied through the removable line cord.

**Digital I/O** - This DB-25 connector allows an 8-bit input to control the depth. Each increment is approximately 0.15 mm in depth (0.2μsec), with the maximum depth being approximately 3.9 mm (51.2 μsec). This connector also provides a digital output from the displacement counter. Each increment is about 0.02, 0.04, or 0.08 mm as determined by the front panel full scale switch.

### 4.2 ULTRASOUND/PERTURBATOR HEAD

The side elevation and plan of the perturbator head are shown in Figure 4.2. The perturbator has the transducer embedded in it. Thus, it has a dual purpose, one as a vibrator head and the other as a transducer.
4.3 **PERTURBATOR BOX**

The perturbator box controls the frequency of vibrations of the perturbator head. For a set of data acquisition, the frequency is held constant (8 Hz) and the signal makes the head have a sinusoidal vibration so as to satisfy the $x = 0$ (surface) boundary condition of the wave, so that the equation of the form $A e^{-8x} \cos(ct - dx + \phi)$ at $x = 0$ for $A = 1$ and $C = 8$ reduces to $\cos(ct + \phi)$. 
4.4 MECHANICAL SUPPORT

The mechanical supports used for data acquisition are the table, a cushion which is kept on the table to support the limb, and the hydraulic jack which has a pedal, with the help of which the height of the perturbator head can be adjusted according to need.

4.5 COMPUTER

An Apple Ile computer has been used for data acquisition to support the computer software with the perturbator board in slot 4 and clock in slot 5. For analysis, an Apple Ile as well as a Sun Workstation have been used.
5 SOFTWARE DEVELOPMENT

The software package is broadly divided into two parts: data collection and data analysis. Figure 5.1 shows broadly how all the different parts of the software are connected to each other.

**Figure 5.1** Overall software flow chart
5.1 DATA COLLECTION

Data collection is done with the help of the instrumentation described in Chapter 5 in combination with the software which constitutes the following programs.

5.1.1 PERT.DATA/PERT.BIN

PERT.DATA performs most of the dialogue with the user. It uses the assembly language program PERT.BIN to collect data from the perturbator, displays the data on the screen, and saves the data to the disk, if required. PERT.BIN is a set of assembly language routines that can be called from PERT.DATA. These routines interface with the perturbator via slot 4. There are three functions: initializer, de-initializer, and interrupt.

Initializer is called at $4000 (16384) by PERT.DATA at the beginning of data acquisition. Init sets the clock to cause an interrupt each .01 second during which the interrupt routine is run. It initializes constants and counters for the interrupt routine.

De-initializer is called at $4002 (16386) by PERT.DATA at the end of data acquisition. De-init disables the interrupts and stops the clock.

Interrupt is called by the interrupt clock every .01 seconds. Interrupt reads the input channel to get the displacement. It calculates an average peak to peak amplitude of motion and stores this in a specified location for PERT.DATA. Interrupt also gets this depth setting from PERT.DATA, via a memory location, and sends it to the output channel for the perturbator.

The assembly routines are stored in memory block $4000 through $4400. Assembly language program variables are kept with the routines.
Data is transferred to the basic program at memory locations $0300$ (768) through $03CF$ (975).

5.1.2 **PERT.COMB**

PERT.COMB combines data sets collected during a session into a large file that is easy to manipulate. The user selects the data sets to be combined and then gives the subject initials. The initials plus the date are used for the name of the file. The user may create one or more files from the data. Care is taken not to overwrite files.

PERT.COMB is called automatically from PERT.DATA. PERT.COMB will exit to the data analysis program if desired.

5.1.3 **PERT.CALIB**

PERT.CALIB guides the user through the calibration of the ultrasound Doppler machine. It may be helpful to have a copy of the schematics of the machine when using this program. Recalibration is done only if a part is replaced or if the machine appears to need it.

PERT.CALIB is called from PERT.DATA and calls PERT.DATA upon completion of the calibration.

5.2 **DATA ANALYSIS**

The algorithms for tissue differentiation are based on the broad theory developed in Chapter 4. The raw data on which analysis of vibration is done is in the form of amplitude of vibration of tissue for various depths starting from 0 cm. with an incremental step of 0.015 cm. The various patterns seen on the amplitude versus depth plot are recognized by the analysis program to do tissue differentiation.
For analysis, there are three C language programs which are run on Sun Workstations using MATLAB for plotting. The three programs are: analysis.c, muscle.c, and graph.c. analysis.c helps in eliminating various tissue regions, leaving behind the muscle region for muscle.c to work on.

5.2.1 analysis.c

This program inputs the depth and amplitude data from files and then calls function fat() to identify the fat and skin region. If this region is substantial, function fit() is called to do the best curve fitting. Function modulus() calculates the elastic modulus of this region. Then the function bone() is called to eliminate the region affected by the bone, if present. Next, function blood() is called recursively to eliminate the blood region present anywhere. The next section explains the functions used in the program analysis.c.

5.2.2 FUNCTIONS IN analysis.c

(1) fat(depth, disp, count)

This function takes in the depth and amplitude arrays with the count of entries and returns the end of the fat and skin region. It calls another function, derivative(), to calculate slopes at each depth. The derivative() function calculates slopes using the three point slope method for the first and last points, the two point (central - difference) slope method for the second and the second to last points, and the four point slope method for the rest of the points. Then the flat slope algorithm is used to identify the end of the fat and skin region. In this algorithm, if the slope is consistently less than -0.6 for four points, then the end of the region is identified.
(2) \textit{fit(depth, disp, start, fend, alpha, gamma)}

This function inputs the arrays for depth and amplitude, the start and end of the region for which fit parameters alpha and gamma have to be calculated. Alpha and gamma are calculated using equations (B.17) and (B.18).

(3) \textit{modulus(gamma, \textit{ym})}

This function calculates the elastic modulus value for a given gamma value.

(4) \textit{bone(disp, fend, count)}

This function uses the array disp, the integer fend (to show the end of the fat/skin region), and the total count as inputs. It outputs the start of the bone affected area and if one does not exist, outputs a zero.

To find the bone affected region, the program first searches for a steady minimum starting from the end depth. After finding a steady minimum, it searches for a steady maximum on its right. If the difference in the amplitudes at these two extremes is more than 0.4 mm, the region is identified. The function also exits if the maximum or minimum values are not found.

(5) \textit{blood(depth, disp, fend, bnstart, count, bstart, bend)}

For this function, arrays depth and disp, integer fend (to show the end of the fat/skin region), and integer bnstart (to show the start of bone affected region) are used as inputs. This function outputs bstart and bend, which show the start and the end of the blood vessel region.

This function finds the blood vessel region by searching between the fend and bnstart region. It searches for a steady minimum starting from the bnstart and moving towards zero depth and then searches for the first steady maximum on its left. If any or both are not found, it outputs
zeroes for bstart and bend indicating that the region was not found. It both are found, it calls fit() to calculate alpha and gamma and calculates the effective modulus using modulus(). If the value of the modulus is less than 0.2, the blood vessel region has been identified.

5.2.3 muscle.c

This program first asks for the number of times the program is to be used and then asks for the start and end depths from the user. It then outputs the fit parameters alpha and gamma, elastic modulus of the region, error in fit, and the error rate.

The fit parameters and the elastic modulus are calculated using fit() and modulus(). Error is calculated by comparing the ideal curve (calculated by using the fit parameters), and the actual curves. The error rate is the error per one data value. The err() function is called to calculate the error.

5.2.4 graph.c

This program asks the user for the start depth, end depth, and the fit parameters alpha and gamma. It then prints out the ideal exponential curve for that region.

5.2.5 Computer Outputs

An interaction with the three analysis programs is given in Appendix E. Figure 5.2 shows the normal amplitude-depth curve and the ideal exponential curve for one region.
Figure 5.2 Normal curve and idealized curve
6 PRESSURE SORE SIMULATION

6.1 BACKGROUND

6.1.1 Assumption

The assumption for pressure sore simulation is based on high stiffness of the ulcer region. The elastic property of the tissue is hypothesized to be the factor in the occurrence of pressure sores. In a healthy tissue, the force applied on the skin is dissipated within subdermal tissue layers without any necrotic changes. During necrosis, the collagen is broken down, which results in higher load distribution on the fluid vessels hampering blood supply and lymphatic fluid flow and thus causing the rupture of cell membranes and stiffness.

6.1.2 Stiff tissue

As has been discussed in Appendix A, stiffer tissue will have a less steep amplitude to depth curve. This was also supported experimentally by taking data on patients before and after they performed some exercise. After doing exercise for some time, the muscle region in the amplitude versus depth curve showed an increase in slope (relative to the previously negative slope).

6.2 SIMULATION

6.2.1 Computer Interaction

To create a pressure sore simulation graphically, the sim.c program increases the amplitude values in the region specified by the user. The increase in the displacement is in terms of percentage
increase with respect to difference in the amplitude of the starting point (given by the user) and the point in consideration.

In our simulation here, three ranges are identified; one from 0.5 cm to 1.0 cm, a second one from 0.75 cm to 1.25 cm, and the last one from 1.0 cm to 1.5 cm. To observe the effect of percentage change to the elastic modulus of the region, four different percentage increase values were taken, namely 20%, 40%, 60%, and 80%.

For each of these percentage values, program muscle.c was run in order to calculate the elastic modulus of the three chosen regions. The interaction with the computer program is given in the Appendix D. Figures D.1 to D.4 show the plots for various percentage increases. In each figure, plot #1 shows the normal curve, plot #2 shows the plot for change in the 0.5 to 1.0 cm region, plot #3 shows the plot for change in the 0.75 to 1.25 cm region, and plot #4 shows the plot for change in the 1.0 to 1.5 cm region.

The results of this work can be put into a tabular form as follows.

<table>
<thead>
<tr>
<th>Percentage increase (%)</th>
<th>0%</th>
<th>20%</th>
<th>40%</th>
<th>60%</th>
<th>80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 - 1.0</td>
<td>.986114</td>
<td>1.484137</td>
<td>2.390774</td>
<td>4.281199</td>
<td>9.21570</td>
</tr>
<tr>
<td>.75 - 1.25</td>
<td>.3917708</td>
<td>.616391</td>
<td>1.025794</td>
<td>1.870003</td>
<td>3.998277</td>
</tr>
<tr>
<td>1.0 - 1.5</td>
<td>.253292</td>
<td>.430065</td>
<td>.772437</td>
<td>1.537471</td>
<td>3.724491</td>
</tr>
</tbody>
</table>

Table 6.1 Modulus
Thus Table 6.1 shows the modulus values calculated by the program muscle.c for the various percentage increases and for different depth regions.

![Graph](image)

**Figure 6.1** Modulus versus percentage increase in amplitude

Figure 6.1 shows the plot of elastic modulus changing with different percentage increases in the amplitudes. The three curves show values for three different regions. The top curve represents the 0.5 to 1.0 cm region, the middle one represents the 0.75 to 1.25 cm region, and the lower one represents the 1.0 to 1.5 cm region.

The data is yet to be taken on patients with varying degrees of pressure sores. The simulation presented here gives an impression of our expectation of amplitude-depth curves for varying degrees of pressure sores.
7 STATISTICAL ANALYSIS

Data was collected on different regions of the arm on six subjects on various days. The data was in the form of files containing depth values and the corresponding amplitude of vibrations. The amplitude values having major oscillations which indicate bone affected regions were eliminated. Each data set was divided into parts for different depth ranges. For each range, best fitting was done to get the fit parameters and then elastic modulus of each region was calculated. On the basis of tissue differentiation performed manually, the modulus values were kept in separate sets for skin, fat, and muscle. The effective modulus for blood vessel area was also calculated. Statistical analysis was done on each set of data. This statistical analysis of the data on skin, fat, muscle, and blood is shown in the tables which follow. Although these values are not very consistent as can be seen on the plots as well as the statistical outputs, and also by noting the changes in the output, by removing three data values for muscle, the tissue types seem to fall in the following general ranges.

<table>
<thead>
<tr>
<th>Modulus Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>modulus for muscle</td>
<td>0.1 to 5 psi</td>
</tr>
<tr>
<td>modulus for fat</td>
<td>10 to 200 psi</td>
</tr>
<tr>
<td>modulus for skin</td>
<td>3000 and above</td>
</tr>
<tr>
<td>effective modulus for blood vessel region</td>
<td>0.2 psi and less</td>
</tr>
</tbody>
</table>

These values combined with the mathematical analysis developed to understand the amplitude versus depth patterns were used to develop the algorithms used in the analysis programs of Chapter 5.
Data File: STATS  
Variable: SKIN  
Observations: 13  

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Maximum</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.431</td>
<td>100000000000.000</td>
<td>9999999993.569</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Median</th>
<th>Mean: 823077016.009</th>
<th>Standard Error: 764865812.491</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000000000.000</td>
<td>7605256244528270324.000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standard Deviation:</th>
<th>Coefficient of Variation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2757762905.786</td>
<td>335.055</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.815</td>
<td>6.438</td>
</tr>
</tbody>
</table>

Table 7.1  Skin Table
Data File: STATS
Variable: FAT
Observations: 237

Minimum: 0.218
Maximum: 10000000000.000
Range: 9999999999.782
Median: 59.258

Mean: 770042315.197
Standard Error: 172260966.712

Variance: 7032700234640708983.000
Standard Deviation: 2651923874.217
Coefficient of Variation: 344.387

Skewness: 3.180
Kurtosis: 8.151

| Table 7.2 | Fat Table |
Data File: STATS
Variable: BLOOD  Observations: 98

<table>
<thead>
<tr>
<th>Minimum: 0.000</th>
<th>Maximum: 2.845</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range: 2.845</td>
<td>Median: 0.162</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean: 0.313</th>
<th>Standard Error: 0.043</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance: 0.178</td>
<td>Standard Deviation: 0.422</td>
</tr>
<tr>
<td>Coefficient of Variation: 134.902</td>
<td></td>
</tr>
</tbody>
</table>

| Skewness: 3.101 | Kurtosis: 12.867 |

**Table 7.3** Blood Table
Data File: STATS
Variable: MUSCLE Observations: 514

| Minimum: 0.000 | Maximum: 10000000000.000 |
| Range: 9999999999.999 | Median: 1.480 |

Mean: 19844407.893 Standard Error: 19456436.027

| Variance: | 194576192067132258.600 |
| Standard Deviation: | 441107914.310 |
| Coefficient of Variation: | 2222.832 |

| Skewness: | 22.533 |
| Kurtosis: | 506.808 |

Table 7.4 Muscle Table 1
Data File: STATS
Variable: MUSCLE

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations:</td>
<td>511</td>
</tr>
<tr>
<td>Minimum:</td>
<td>0.000</td>
</tr>
<tr>
<td>Maximum:</td>
<td>8314.600</td>
</tr>
<tr>
<td>Range:</td>
<td>8314.600</td>
</tr>
<tr>
<td>Median:</td>
<td>1.479</td>
</tr>
<tr>
<td>Mean:</td>
<td>50.209</td>
</tr>
<tr>
<td>Standard Error:</td>
<td>19.071</td>
</tr>
<tr>
<td>Variance:</td>
<td>185852.292</td>
</tr>
<tr>
<td>Standard Deviation:</td>
<td>431.106</td>
</tr>
<tr>
<td>Coefficient of Variation:</td>
<td>858.615</td>
</tr>
<tr>
<td>Skewness:</td>
<td>15.192</td>
</tr>
<tr>
<td>Kurtosis:</td>
<td>269.130</td>
</tr>
</tbody>
</table>

Table 7.5 Muscle Table 2
Figure 7.1  Fat Chart
Figure 7.2  Muscle Chart
Figure 7.3 Blood Chart
8 SUMMARY AND CONCLUSIONS

8.1 SUMMARY

In this project, the accomplishments can be summarized as follows. Equipment was set up to take data on various subjects on different regions of their arm on various days. The data was analyzed statistically as described in the previous chapter to come up with the ranges of elasticity. These values and the observations made during the analysis of this data and the mathematical steps described in Chapter 3 and Appendix A were used in building up algorithms used for tissue differentiation. Mathematical steps described in Chapter 3 were evolved to match the data patterns in terms of amplitude versus depth curves for various tissue regions with a mathematical theory. Algorithms were set up in such a way that skin, fat, bone, and blood region would be eliminated so that the muscle region could be checked for the possible development of a pressure sore. Later, computer programs were written based on these algorithms in order to be able to analyze new data for tissue differentiation. Finally, since real data on pressure sore patients was not available, a pressure sore graph simulation was done in C language. In this simulation, amplitude data for various depth regions was taken for one set of amplitude-depth data and the change of the elastic modulus with the change in the amplitude values was studied to give a basis for the curves expected to be made by pressure sores of various degrees.

8.2 CONCLUSIONS

The software which can differentiate tissues, needed ultimately for the development of a complete software package which can identify pressure sores
in their early stages, is complete. The expected patterns created on the amplitude versus depth curve by the varying degrees of pressure sores, were studied graphically. This can prove to be useful in the development of the final software for pressure sore identification.

8.3 FURTHER WORK

The aim of this project was to be able to detect pressure sores at early stages. To achieve that, data on patients with varied degrees of pressure sores needs to be taken. Next, an exact algorithm in the form of a computer program needs to be evolved from the study done on that data, which will detect the region where the pressure sore is building up. This program should also determine the intensity of the formation of the pressure sore.

Bhagat, P. K., V. C. Wu  
1980 "Attenuation and backscattering in freshly excised animal tissues."  

1961 "Late causes of death and life expectancy in paraplegia."  
*Can Med Assoc J* 85: 73.

Brekhovskikh, Leonid M.  

Carlin, Benson  

Cheney, Ward and David Kincaid  

Chu, W. K., J. C. Anderson, T. J. Imray, N. Batros, and J. Y. Cheung  
1985 "A Microcomputer-based in-vivo ultrasound tissue differentiation system."  
*Journal of Clinical Engineering,* 10(4).

Cox, R. H.  
1975 "Anisotropic Properties of the Canine Artery in vivo."  
*Journal of Biomechanics,* 8: 293-300.

Dinsdale, S. M.  
1974 "Decubitus ulcers: Role of pressure and friction in causation."  
Dowd, G. S., K. Linge, and G. Bentley  

Freed, M. M., H. J. Bakst, and D. C. Barrie  

Fry, Francis J.  

Gaeton, T. Noia and Lavs M. Vistnes  

Griffith, B. H.  

Hackler, R. H.  

Hagisawa, Satsue, M. W. Ferguson-Pell, V. R. Palmieri, and G. Cochran  

Herzfeld, Karl F. and Theodore A. Litovitz  

Husain, T.  
Hutchinson, K. J., E. Karpinski, and C. E. Capjack.

Kaufman, J., L. Breeding and N. Rosenberg

Kosiak, M.

Krouskop, T. A., D. R. Dougherty, and F. S. Vinson

Lewis, D. W., and R. E. McLaughlin

Lindan, O.

Loeffler L. and K. Sagawa

Nayfeh, Ali Hasan

Patel D. J., J. S. Janicki, R. N. Vaishnav, and J. T. Young
Reichel, S. M.
1948 "Shearing force as a factor in decubitus ulcers in paraplegics."
JAMA 166: 762.

Schnell, V. C. and L. E. Wolctoo
1966 "The etiology, prevention, and management of decubitus ulcers."

Timoshenko, S.P. and J. N. Goodier

Troung, X. T.


Visser, H. J., A. M. Jacobs, L. Oloff, and J. J. Drago

Vistines, L. M.

Wagner, F. W.
A THEORY OF ELASTICITY

Pertaining to the elastic properties of materials, the biological material beneath the outermost layer of the skin can be broadly divided into the following types:

(1) linear visco-elastic material
(2) linear elastic solid
(3) non-linear elastic solid

Here we are basically concerned with propagation of waves through these different types of materials. In this chapter, the basic equation is derived from the theory of elasticity. The notations used in the development of the basic equation are introduced side by side.

A.1 ELASTICITY

An understanding of the following concepts is necessary to appreciate the theory of elasticity.

A.1.1 Stress: Let $\delta P$ be the resultant force acting on an infinitesimal area $\delta A$, then the magnitude of the stress is defined as the limiting value of the ratio $\delta P/\delta A$ (Timoshenko and Goodier). The direction of the stress is given by the limiting direction of the direction of the resultant force $\delta P$. Generally, the resultant is inclined to the area $\delta A$ and so the stress can be resolved into three components acting in the three orthogonal directions; the one acting normal to the plane called "normal stress", the other two acting on the plane called "shearing stress."
Figure A.1 Stress-Force Relationship

The notation for the stress is $P_{ij}$, where $i$, $j$, can be 1, 2, or 3. $P_{ij}$ is the stress on an area perpendicular to the axis $x_i$ and acting in the direction of the axis $x_j$. $x_1$, $x_2$, and $x_3$ are the three orthogonal axes.

Figure A.2 Stress Distribution
Taking moments about a line through the midpoint parallel to $x_1$, we get

$$P_{32} dx_1 dx_2 dx_3 = P_{23} dx_1 dx_2 dx_3 \quad \text{so} \quad P_{32} = P_{23}, \quad \text{similarly} \quad P_{12} = P_{21},$$

$$P_{31} = P_{13}.$$ So now we have only six unknown components of stress.

A.1.2 Strain: Assume that there are enough constraints not to let the body move as a rigid body. If $X_1$, $X_2$, and $X_3$ are the components of displacement, then the strain $S_{ij}$ is defined as

$$S_{ij} = \frac{1}{2} \left( \frac{\partial X_i}{\partial x_j} + \frac{\partial X_j}{\partial x_i} \right) \quad \text{--(A.1)}$$

(Timoshenko and Goodier)

We can see that

$$S_{11} = \frac{\partial X_1}{\partial x_1}, \quad S_{22} = \frac{\partial X_2}{\partial x_2}, \quad \text{and} \quad S_{33} = \frac{\partial X_3}{\partial x_3}.$$

These are "unit elongations" in the directions of the three axes. $S_{ij}, \ i \neq j$ are the shearing strain components which are actually the angular deformations between the planes $ik$ and $jk; \ i \neq j \neq k.$

\[ \text{Figure A.3} \ \text{Strain Development} \]
A.1.3 Hooke's Law

Hooke's law states the linear relationship between the stress and strain components for the materials which behave in that fashion (Timoshenko and Goodier). We have

\[ S_{ij} = \frac{1}{E} [P_{ij} - \nu(P_{ji} + P_{kk})] \quad \text{where} \quad i \neq j \neq k; \quad -(A.2) \]

\( \nu \) is the Poisson's ratio which accounts for the lateral strain components accompanied by the strain in the axis in consideration; and \( E \) is the modulus of elasticity in tension. To use the superposition principle we have assumed deformations so small that they do not affect the external force action.

To obtain the relationship between shearing strain and shearing stress we consider a rectangular parallelepiped as shown in Figure A.4 taking \( P_{33} = -P_{22} = \sigma \) and \( P_{11} = 0 \).

\[ \begin{align*}
\begin{array}{c}
\text{Figure A.4 Shear Relationships}
\end{array}
\end{align*} \]
Equating the forces on the triangle OBC, we get $\tau = \sigma$. Assume size AB and BC remain constant and that elongation of OB is same as the OC contraction.

\[
\frac{OC}{OB} = \tan\left(\frac{\pi}{4} - \gamma\right) = \frac{1 + s_{22}}{1 + s_{11}}
\]

Now $\tau$ is shearing stress of the form $S_{ij}$ $i \neq j$ and $\gamma$ shearing strain of the form $P_{ij}$ $i \neq j$. Using equation (A.2)

\[
\gamma = \frac{(1 + v)\tau}{E} \quad \text{so} \quad \gamma = \frac{\tau}{2G}, \quad \text{where} \quad G = \frac{E}{2(1 + v)}
\]

$G$ is the modulus of elasticity in shear or the modulus of rigidity. Now, we have the same relationships.

\[
P_{ij} = 2GS_{ij} \quad i \neq j \quad \text{---(A.3)}
\]

We can separate stress into two parts, a compressional part

\[
P = \frac{1}{3}(P_{11} + P_{22} + P_{33}) \quad \text{---(A.4)}
\]

and a deviatory part

\[
P_{ij} = P_{ij} - P
\]

\[
P_{ij} = P_{ij} \quad i \neq j
\]

Strain, similarly, can also be divided into a compressional part

\[
-\dot{S} = S_{11} + S_{22} + S_{33}
\]

and a deviatory strain

\[
\ddot{S}_{ij} = S_{ij} + \frac{1}{3}S
\]

\[
\ddot{S}_{ij} = S_{ij} \quad i \neq j \quad \text{---(A.5)}
\]

From equations (A.2) and (A.3), we get for the compressional part:

\[
P = -kS
\]

where
\[
k = \frac{E}{3(1 - 2\nu)} \tag{A.6}
\]

and for the deviatory part
\[
P_{ij} = 2GS_{ij} \tag{A.7}
\]

### A.2 NEWTON'S LAWS

#### A.2.1 Forces acting on a continuum mass:

![Diagram of forces acting on a continuum mass](image)

**Figure A.5** Force Equilibrium

(Herzfeld and Litovitz)

Equating the forces in the direction \(i\) acting on a volume element in terms of stresses and noting \(P_{ij} = P_{ji}\), we get force per unit volume in the \(X_i\) direction

\[
f_1 = \frac{\partial P_{11}}{\partial x_1} + \frac{\partial P_{12}}{\partial x_2} + \frac{\partial P_{13}}{\partial x_3} = (\text{Div}P)_1
\]

Similarly, we get

\[
f_2 = \frac{\partial P_{21}}{\partial x_1} + \frac{\partial P_{22}}{\partial x_2} + \frac{\partial P_{23}}{\partial x_3} = (\text{Div}P)_2
\]
\[ f_3 = \frac{\partial P_{31}}{\partial x_1} + \frac{\partial P_{32}}{\partial x_2} + \frac{\partial P_{33}}{\partial x_3} = (\text{Div} P)_3 \]

where \( f_3 \) is the component of force/unit volume in the \( x_3 \) direction.

\[ f = \text{Div} P \quad \text{---(A.8)} \]

A.2.2 Newton's Equation for the continuum mass

If \( n \) is mass, \( F \) force, and \( u \) velocity, the familiar form of Newton's second law is:

\[ F = \frac{Dm}{Dt} \]

Note that we are ignoring the body force \( mg \).

\[ F = \frac{mDu}{Dt} + \frac{uDM}{Dt} \]

Using conservation of mass,

\[ \frac{Dm}{Dt} = 0 \quad \text{so} \quad F = \frac{mDu}{Dt} \]

Dividing both sides by unit volume

\[ f = \rho \frac{DU}{Dt} = \rho \left[ \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x_1} \cdot \frac{\partial x_1}{\partial t} + \frac{\partial u}{\partial x_2} \cdot \frac{\partial x_2}{\partial t} + \frac{\partial u}{\partial x_3} \cdot \frac{\partial x_3}{\partial t} \right] \]

\[ f = \rho \left[ \frac{\partial u}{\partial t} + (u \cdot \nabla) u \right] \quad \text{---(A.9)} \]

(Krouskop, Dougherty, et al, p. 2)

From (A.6) and (A.9), we get

\[ \text{Div} P = \rho \left[ \frac{\partial u}{\partial t} + (u \cdot \nabla) u \right] \quad \text{---(A.10)} \]

A.3 DEVELOPMENT OF THE BASIC EQUATION

A.3.1 Perturbation Method to solve differential equations

For convenience in working with nonlinear equations, for which exact solutions are difficult to find, the parameters in the equations are
separated into a linear combination of terms of different orders (Nayfeh).

To solve equation (A.7), we can write for small $\varepsilon$

$$f = f_0 + \varepsilon f_1 + \varepsilon^2 f_2 + \ldots$$

$$u = \varepsilon u_1 + \varepsilon^2 u_2 + \ldots$$

$$\rho = \rho_0 + \varepsilon \rho_1 + \varepsilon^2 \rho_2 + \ldots$$

---(A.11)

The zero-order contribution to $u$ does not appear because we observe that the material is stagnant in the absence of any disturbance. $f$ can also be written as $\varepsilon f_1 + \varepsilon^2 f_2 + \ldots$ because we are neglecting the zeroeth order term, i.e. the body force per unit volume $\rho g$. Using these parameters in the equation (A.9) and equating the first order terms, that is, the $\varepsilon$ terms, we get

$$f_1 = \rho \frac{\partial u}{\partial t}$$

---(A.12)

Now taking displacements as $X^j$ we can write equation (A.10) as

$$\rho \frac{\partial^2 X_i}{\partial t^2} = \sum_{j=1}^{3} \frac{P_j}{\partial x_j}$$

Dividing the stress into a compressional part and a deviatoric part as before, we get

$$\rho \frac{\partial^2 X_i}{\partial t^2} = \frac{\partial \rho}{\partial x_i} + \sum_{j=1}^{3} \frac{P_{ij}}{\partial x_j}$$

---(A.13)

Using (A.1), (A.4), and (A.5) in (A.13), we get
\[ \rho \frac{\partial^2 X_i}{\partial t^2} = (k - \frac{2}{3}G) \sum_{j=1}^{i=3} \frac{X_j}{\partial x_i \partial x_j} + G \sum_{j=1}^{i=3} \frac{\partial^2 X_i}{\partial x_j^2} \frac{\partial^2 X_j}{\partial x_j^2} \]  

---(A.14)

For an acoustic (irrotational) wave propagated along the \( x_1 \) axis, equation (A.14) becomes

\[ \rho \frac{\partial^2 X_1}{\partial t^2} = \frac{4}{3}G \frac{\partial^2 X_1}{\partial x_1^2} \]  

---(A.15)
B LEAST SQUARE FIT

B.1 LEAST SQUARE FIT

From equations (2.19) and (2.20) we see that

\[ E = \frac{-3pw^2}{2y^2} \]

---(B.1)

To obtain \( E \) in a constant \( E \) region, we need to find the value of \( y \) from the plot of \( A \) versus \( x \). To do that we must use the best-squares fit. That leads to the problem of minimizing the function (Cheney and Kincaid).

\[ \varphi(\alpha, y) = \sum_{k=1}^{m} (\alpha e^{-\gamma x_k} - A_k)^2 \]

---(B.2)

The minimum occurs for value of \( \alpha \) and \( y \) such that

\[ 0 = \frac{\partial \varphi}{\partial \alpha} = \sum_{k=1}^{m} 2(\alpha e^{-\gamma x_k} - A_k) e^{-\gamma x_k} \]

---(B.3)

and

\[ 0 = \frac{\partial \varphi}{\partial y} = \sum_{k=1}^{m} -2(\alpha e^{-\gamma x_k} - A_k) \alpha x_k e^{-\gamma x_k} \]

---(B.4)

These equations are nonlinear in \( \alpha \) and \( y \) and can be solved by using some numerical methods.

B.2 NONLINEAR SOLUTION

There are two ways of solving this problem using the numerical methods:

(1) Solving (4.3) and (4.4) by the Standard Quasilinear Algorithm (SQA) or the Modified Quasilinear Algorithm (MQA); or (2) Minimizing (4.2) directly using the Ordinary Gradient Algorithm (OGA).
B.2.1 MQA/SQA

MQA and SQA can be used to solve two simultaneous equations in two variables as (4.3) and (4.4). Let us solve

\[ \varphi = 0 \quad --(B.5) \]
\[ \psi = 0 \quad --(B.6) \]

Let the solution point be \((\alpha_0, \gamma_0)\) and the starting point be \((\alpha, \gamma)\); so now \(\varphi(\alpha, \gamma) \neq 0\) and \(\psi(\alpha, \gamma) \neq 0\).

If the error is \(P\),

\[ P = \varphi^2 + \psi^2 \quad --(B.7) \]

Let the incremented \(\varphi\) and \(\psi\) be \(\Phi\) and \(\Psi\) and taking the Taylor’s expansion to the first order, we get

\[ \tilde{\varphi} = \varphi + \Delta \varphi = \varphi + \delta \varphi \quad --(B.8) \]
\[ \tilde{\psi} = \psi + \Delta \psi = \psi + \delta \psi \quad --(B.9) \]
\[ \tilde{P} = P + \Delta P = P + \delta P \quad --(B.10) \]

So, to get a closer point to the solution, we need \(\delta P < 0\).

From (4.7) we get

\[ \delta P = 2 \varphi \delta \varphi + 2 \psi \delta \psi \quad --(B.11) \]

To get \(\delta P < 0\), we can choose \(\delta \varphi = -\alpha \varphi\) and \(\delta \psi = -\alpha \psi\) where \(\alpha\) is some positive real number. That would give us

\[ \delta P = -2\alpha(\varphi^2 + \psi^2) < 0 \]
\[ \varphi_x \Delta x + \varphi_y \Delta y + \alpha \varphi = 0 \quad --(B.12) \]
\[ \psi_x \Delta x + \psi_y \Delta y + \alpha \psi = 0 \quad --(B.13) \]

By choosing \(\alpha\) we can solve these equations and find \(\Delta x\) and \(\Delta y\). We have a unique solution if
\[
det \begin{bmatrix} \phi_x & \phi_y \\ \psi_x & \psi_y \end{bmatrix} \neq 0
\]

If there are infinite solutions, we pick up the solution for which 
\((\Delta x)^2 + (\Delta y)^2\) is a minimum. SQA is a special case of MQA for which \(\alpha = 1\).

MQA can be summarized in the following steps:

Step 1: Choose \(x\) and \(y\).

Step 2: Calculate \(\phi\) and \(\psi\).

Step 3: Check for
\[
P = \phi^2 + \psi^2 \quad \text{if} \quad P \leq \varepsilon \quad \text{stop}
\]
\[
\text{if} \quad P > \varepsilon \quad \text{continue}
\]
where \(\varepsilon\) is the tolerance error.

Step 4: Calculate \(\phi_{x'}\), \(\phi_{y'}\), \(\psi_{x'}\), \(\psi_{y'}\) and calculate \(\phi\) and \(\psi\) from the linear set of equations (B.12) and (B.13), choosing \(\alpha\) starting from 1 using the bisection method such that \(P(\alpha) < P(0)\). For SQA, take \(\alpha = 1\).

B.2.2 OGA

OGA minimizes a function such as \(f = f(x,y)\) as in the case in equation (B.2). For minimization, we want \(f_x = 0\) and \(f_y = 0\). If we begin with some \(x\) and \(y\), we get error
\[
Q = f_x^2 + f_y^2
\]
and we keep our stopping criterion as \(Q \leq \varepsilon\). Let \((\bar{x}, \bar{y})\) be the varied point and \(\bar{f}\) be the value of the function at the varied point. Then,
\[
\bar{x} = x + \Delta x
\]
\[
\bar{y} = y + \Delta y
\]
\[
\bar{f} = f + \Delta f
\]
we want $\tilde{f} < f$ or $\Delta f < 0$. Using Taylor's expansion to the first order, $\delta f < 0$

$$\delta f = f_x \Delta x + f_y \Delta y < 0$$

to ensure this take,

$$\Delta x = -\alpha f_x \quad \text{and}$$
$$\Delta y = -\alpha f_y$$

Since $\alpha$ and $Q$ are positive,

$$\delta f = -\alpha(f_x^2 + f_y^2) = -\alpha Q$$

Thus we have a guaranteed negative $\delta f$.

OGA can be summarized in the following steps:

Step 1: Select $x$ and $y$

Step 2: Calculate $f$, $f_x$, $f_y$, $C$ and

- If $Q \leq \varepsilon$ stop
- If $Q > \varepsilon$ continue

Step 4: Select $\alpha$ such that $\tilde{f}(\alpha) < \tilde{f}(0)$ employing bisection from $\alpha = \alpha_0$.

Step 5:

$$\tilde{x} = x - \alpha f_x$$
$$\tilde{y} = y - \alpha f_y$$

B.2.3 Scanning and cubic interpolation

Take two values of $\alpha$, namely, $\alpha_1$ and $\alpha_2$ such that $\tilde{f}_\alpha(\alpha_1) < 0$ and $\tilde{f}_\alpha(\alpha_2) < 0$.

Now

$$\tilde{f}(\alpha_1), \tilde{f}_\alpha(\alpha_1), \tilde{f}(\alpha_2), \text{ and } \tilde{f}_\alpha(\alpha_2)$$

are known and we can assume

$$\tilde{f}(\alpha) = k_0 + k_1 \alpha + k_2 \alpha^2 + k_3 \alpha^3$$
We thus have four equations for $\alpha_1$ and $\alpha_2$ and solving them we obtain $k_0$, $k_1$, $k_2$, and $k_3$.

For minimum $f$,

$$\tilde{f}_0(\alpha) = 0 \text{ and } \tilde{f}_\alpha(\alpha) > 0,$$

that is,

$$3k_3\alpha^2 + 2k_2\alpha + k_1 = 0 \text{ and } k_2 + 3k_3\alpha > 0.$$

Solving these yields $\alpha_0$.

### B.3 Linear Solution

In this method, the exponential problem is reduced to a linear one by taking the natural log of the equation which is being fitted. The method of solving the problem is as follows.

$$\ln A_k = \ln \alpha - \gamma x$$

which leads to the minimization of the function

$$\Phi(\alpha, \gamma) = \sum_{k=1}^{m} (\ln \alpha - \gamma x_k - \ln A_k)^2,$$

i.e.

$$0 = \frac{\partial \Phi}{\partial \alpha} = \sum_{k=1}^{m} 2(\ln \alpha - \gamma x_k - \ln A_k) \frac{1}{\alpha}$$

and

$$0 = \frac{\partial \Phi}{\partial \gamma} = \sum_{k=1}^{m} -2(\ln \alpha - \gamma x_k - \ln A_k)x_k$$

Solving these two equations, we have

$$\gamma = \frac{\left( \sum_{k=1}^{m} x_k \right) \left( \sum_{k=1}^{m} \ln A_k \right) - m \sum_{k=1}^{m} (\ln A_k)(x_k)}{m \left( \sum_{k=1}^{m} x_k^2 \right) - \left( \sum_{k=1}^{m} x_k \right)^2}$$

and $\alpha$ can be calculated by

$$\alpha = \frac{1}{\gamma} \left( \sum_{k=1}^{m} \ln A_k - \gamma \sum_{k=1}^{m} x_k \right)$$

**--(B.14)**

**--(B.15)**

**--(B.16)**

**--(B.17)**
\[
\alpha = \exp\left[\frac{\sum_{k=1}^{m} \gamma_k^2 + \sum_{k=1}^{m} (\ln A_k)(x_k)}{\sum_{k=1}^{m} x_k}\right]
\]  

\text{--(B.18)}

The linear method can be appreciated more when we transform the y-axis of the amplitude-depth plot of Figure B.1 into a natural log y and notice the straightening of the exponentiality. So, instead of trying to fit an exponential curve in the first case, we are trying to fit a straight line with \(-\gamma\) as the slope of this line.

**Figure B.1** Amplitude-Depth Curve

**Figure B.2** Log Amplitude-Depth Curve
C. PROGRAM LISTING

C.1 Data-collection programs

C.1.1 Pert.data

*****************************************************************************
| Program | Pert.data |
| Language | Applesoft |
| Equipment | Apple IIE, Perturbator interface board (Slot 4) |
| and Clock (Slot 5) |
*****************************************************************************

Run parameters
- FQ=Frequency of perturbation (Hz)
- ID=Start depth (cm)
- LD=End depth (cm)
- SD=Peak to peak amplitude of perturbator vibration at skin surface (mm)
- A0%=A9%=Memory locations for passing data to and from the interrupt routine

Raw Readings
- DEPTH=Depth (.015cm/step)
- DISP=Vpp (Displacement) (.04mm/step)
- NS%=Number of sample points

INITIALIZATION
10 GOSUB 100
| Clear screen to bottom, Clear keyboard |
20 VTAB 11:CALL -958:POKE -16368,0: |
| INPUT "Collect data? Y/N ":ANS: |
| IF ANS="N" OR ANS="n" THEN GOTO 50 |
40 GOSUB 300
| DATA COLLECTION LOOP |
50 IF INDEX%<10 THEN |
60 VTAB 13:CALL -958: POKE -16368,0: |
| INPUT "Collect more data? Y/N ":ANS: |
| IF ANS="Y" OR ANS="y" THEN GOTO 40 |
70 IF INDEX%>0 THEN GOSUB 700 |
| Clear screen to bottom, Clear keyboard |
80 VTAB 13:CALL -958:POKE -16368,0: |
| INPUT "Process data? Y/N ":ANS: |
| IF ANS="Y" OR ANS="y" THEN |
| PROCESS DATA |
| PRINT D$:"RUN PERT.ANLS.BAS" |

*** SETUP A/D AND D/A ***
| Clear screen to bottom, Clear keyboard |
90 VTAB 11:CALL -958:POKE -16368,0: |
| INPUT "Calibrate? Y/N ":ANS: |
| IF ANS="Y" OR ANS="y" THEN |
| PRINT D$:"RUN PERT.CALIB.BAS": |
| END |
| Clear screen to bottom, Clear keyboard |
100 VTAB 11:CALL -958:POKE -16368,0: |
| INPUT "Set date or time? Y/N ":ANS: |
| IF ANS="Y" OR ANS="y" THEN |
GOSUB 800

Clear screen to bottom, Clear keyboard
90 VTAB 13:CALL -95S:POKE -16368,0;
INPUT "End now? Y/N ":ANS;
IF ANS="Y" OR ANS="y" THEN
HOLE:CLEAR:END

95 GOTO 20

INITIALIZATION

Initialize variables and software and
Request run parameters from user

*** Initialize variables ***
100 DIM DEPTH(255),DISP(255),COMS(10):
    DS=CHR$(4):
    GS=CHR$(7):
    CS="":".

110 ONERR GOTO 1010

120 FF$="PERT.BIN":
    LOCS="54000":
    PRINT DS:"LOAD ":FF$","A":LOCS:
    FF$="PERT.SHAPE":
    LOCS="54800":
    PRINT DS:"LOAD ":FF$","A":LOCS

    Resume normal error handling
130 POKE 216,0:
    POKE in location of shape table
    POKE 232,0:POKE 233,72

160 PRINT DS:"PR5":PRINT:
    4 3,6,DAY:
    6 3,7,MTH:
    MTHS="JANFEBMARAPRMAJUNJULAGSEQPOCTNOVDEC":
    DT$=""+STR$(DAY)+MIDS(MTHS,MTH*3-2,3):
    D1T$=MIDS(MTHS,MTH*3-2,3="/"+STR$(DAY="/" 1989

170 HOME: VTAB 6:
    PRINT TAB(6):"PERTURBATOR":
    PRINT TAB(4):"DATA COLLECTION":
    PRINT TAB(8):"PROGRAM":
    PRINT TAB(6):D1T$:
    RETURN

INIT PERT.BIN

Activate Interrupt Software for Displacement monitoring

Constants:
    FO=frequency of perturbation (hz)
    RAA=sampling rate is pre-set in interrupt routine to 10ms
    PM=percentage of half cycle to use for window

Data transfer locations
    5300 NVA  A0 Log base two of the number of peaks of which to maintain
an average amplitude.

A1 Number of peaks which to maintain an average amplitude.
A2 Beginning of window in which to scan for max and min.
A3 End of window in which to scan for max or min
A4 Upper limit of displacement before reset
A5 Lower limit of displacement before reset
A6 Average amplitude peak to peak cm = n/1000
A7 0=bad data, 255=good
A8 POKE value to output 0=0cm to 255=3.825cm
A9 Flag 1=new data to read, 0=no new data.

Set data transfer locations

200  A0%=768:  A1%=769:
       A2%=770:  A3%=771:
       A4%=772:  A5%=773:
       A6%=774:  A7%=775:
       A8%=776:  A9%=777:

Set constants

210  RA%=10:  FO=8:
       PW%=80:  UL%=250:
       LL%=4:  SD=2.54:
       INDEX%=0:  CMPSTP=.015

220  CY%=(1000/RA%)/FO+.5:
       HY%=(CY%+1)/2:
       HW%=.5*CY%*PW%/200

Transfer constants

230  POKE A2%,BY%=HW%:
       POKE A3%,HY%=HW%:
       POKE A4%,UL%:
       POKE A5%,LL%

240  RETURN

******************************************************************************

DATA COLLECTION
******************************************************************************

ID=Start depth (cm)
LD=End depth (cm)
DEPTH=Average depth (1 step=.015cm)
DISP=Average amplitude (1 step=.04cm)

Clear screen to bottom, Clear keyboard

300  VTAB 12:CALL -958:POKE -16368,0:
       INPUT " Start depth (cm)? ":ID:
       IF ID<0 OR ID>3.8 THEN
       PRINT GS:
       GOTO 300

Clear screen to bottom, Clear keyboard

305  VTAB 13:CALL -958:POKE -16368,0:
       INPUT " End depth (cm)? ":LD:
       IF LD>3.8 THEN
       PRINT GS:
       GOTO 305

310  IF LD<ID THEN
       PRINT GS:
       GOTO 300
Set constants
312 IDEPTH%=ID/CMPSTP:
314 LDEPTH%=LD/CMPSTP:

315 PRINT "Full scale setting (1-3)?":
316 PRINT "1 5mm":
317 PRINT "2 10mm":
318 PRINT "3 20mm":
319 INPUT ANS%

320 IF ANS%<1 OR ANS%>3 THEN
321 VTAB 14:CALL -958:GOTO 315
322 MHPSTP=.01*(2^ANS%):
323 MF=.5*(2^ANS%):
324 INPUT "Press return to begin data collection":ANX:
325 CALL GRAPH
326 GOSUB 400

Clear screen to bottom
355 VTAB 21:CALL -958:
356 PRINT "Press return for emergency stop":
357 PRINT "Depth":TAB(10):"Amp":TAB(20):"Ave Amp":TAB(30):"Data good":
358 POKE -16368,0:
359 ONERR GOTO 1020
360 FERT.BIN initializer
361 CALL 16384

CALL SCAN LOOP
365 GOSUB 500

*** Stop Interrupts ***
373 CALL 16386:
Resume normal error handling
374 POKE 216,0:
TEXT

Clear screen to bottom
375 VTAB 11:CALL -958:
376 PRINT "Finished with Data Collection ":GS:
377 PRINT NSF:" samples taken":
( CLEAR KEYBOARD )
378 POKE -16368,0:

380 INPUT "Save data? Y/N ":ANS:
381 IF ANS="Y" OR ANS="y" THEN
382 SAVE DATA
383 GOSUB 600

390 RETURN

******************************************************************************
GRAPH ROUTINE
******************************************************************************

Draw vertical lines at 0, 1, 2, 3 and 4cm
400 I=FRE(0):
HGR:HCOLOR=3:
FOR I=29 TO 179 STEP 50:
HPLOT I,132 TO I,5

Draw vertical marks at .5, 1.5, 2.5, and 3.5 cm
and at .1,.2,.3,.4,...,1.1,1.2,1.3,...,3.3 cm
FOR J=32 TO 132 STEP 50:
HPLOT I+25,J+2 TO I+25,J-2:
FOR K=5 TO 45 STEP 5:
HPLOT I+K,J+1 TO I+K,J-1:
NEXT K:
NEXT J:
NEXT I:

Draw horizontal lines at 0, 1 and 2 mm
FOR I=32 TO 132 STEP 50:
HPLOT 29,I TO 229,I:
NEXT I:

Draw Y axis name and units
SCALE=1:
ROT=-48:
DRAW 1 AT 17,122:
DRAW 3 AT 17,56:

Draw X axis name and units and all numbers
ROT=-1:
DRAW 2 AT 39,152:DRAW 4 AT 83,152:
DRAW 5 AT 27,142:DRAW 6 AT 77,142:
DRAW 7 AT 127,142:DRAW 5 AT 21,134:
DRAW 6 AT 21,84:DRAW 7 AT 21,34:

RETURN

**************************************************************
SCAN LOOP
**************************************************************
DEPTH% 1 step=.015 cm depth
DISP% 1 step=.04 mm peak to peak amplitude of motion

500 NS%=-0:
DEPTH%=IDEPTH%

510 NP%=-DEPTH%/33.3:
IF NP%>8 THEN NP%=-8

520 NAVE%=-2*NP%:
IF NAVE%>255 THEN NAVE%=255

Give interrupt program the number of points to average, log
base 2 of the number of points to average, and the new depth.
525 POKE A8%,DEPTH%:
POKE A9%,NP%:
POKE A10%,NAVE%:
J=0

If a key has been pressed, end data collection
530 IF PEEK (-16384)>127 THEN RETURN

Repeat until BIN program signals new data
540 IF PEEK(A9%)=0 THEN GOTO 530
If the displacement has changed or an error signal is received then use new displacement, reset counter (J) and repeat loop.

550 DSPTMP%=PEEK(A6%):
| reset handshake flag
| POKE A9%,0:
| IF DSPTMP%<>DISP% OR PEEK(A7%)<>0 THEN
| DISP%=DSPTMP%:
| J=0:
| GOTO 530

Reread until have same amplitude 8 times with no error flag.

560 J=J+1:
| IF J<8 THEN GOTO 530

If data is good keep it

Data collected in perturbator head's frame of reference which is vibrating with an amplitude of SD mm. Also, the data is in units of MDSTP mm per step. We subtract DISP% from SD/MDSP to get data in a stationary frame of reference and in units of .04mm per step.

570 NS%=NS%+1:
| DEPTH%(NS%)=DEPTH%:
| DISP%(NS%)=SD/MDSP-DISP%

580 IF DISP%(NS%)<130/MF AND DEPTH%<266 THEN
| HPlot DEPTH%=-75+29.132*DISP%(NS%)*MF

590 DEPTH%=DEPTH%+1:
| IF DEPTH%<5DEPTH% THEN GOTO 510

595 RETURN

*****************************************************************************

SAVE DATA
*****************************************************************************

Save data to disc

600 INDEX%=INDEX%+1
| Clear screen to bottom
610 VTAB 13:CALL -958:
| ( CLEAR KEYBOARD )
| POKE -16368,0

620 FFS="F\^*STR$(INDEX%):
| ONERR GOTO 1030

660 PRINT DS:"OPEN";FFS:
| PRINT DS:"WRITE";FFS:
| PRINT NS%;CS:FQ:CS:SD

670 FOR I=1 TO NS%:
| DEPTH=DEPTH%(I)*CMSTP:
| DISP=DISP%(I)*MDSTP:
| PRINT DEPTH;CS:DISP:
| NEXT I

680 PRINT DS:"CLOSE";FFS
Resume normal error handling
690 POKE 216,0:
| INPUT "Data set name ":COM$(INDEX$):
| RETURN
|
|*******************************************************************************
| COMBINE DATA ROUTINE
|*******************************************************************************
|
| 700 FFS="PERT.TEMP":
| ONERR GOTO 720
|
| 710 PRINT DS;"VERIFY":FFS:
| PRINT DS;"DELETE":FFS:
| GOTO 730
|
| ON ERR correction
| 720 CALL -3288
|
| 730 ONERR GOTO 1030
|
| 735 PRINT DS;"OPEN":FFS:
| PRINT DS;"WRITE":FFS:
| PRINT INDEX$;DS;DTS;CS;DITS
|
| 740 FOR I=1 TO INDEX$:
| PRINT COM$(I):
| NEXT I:
| PRINT DS;"CLOSE":FFS
|
| 750 POKE 216,0:
| PRINT DS;"RUN PERT.COMB.BAS":
| END
|
|*******************************************************************************
| SET CLOCK
|*******************************************************************************
|
| Read Clock
| Clear screen to bottom
| 800 VTAB 11;CALL -958:PRINT " Set Time/Date":
| & 3,3,X:MIN=X:
| & 3,4,X:HR=X:
| & 3,5,X:DDT=X:
| & 3,6,X:DDD=X:
| & 3,7,X:HH=X
|
| 810 GOSUB 930:
| GOSUB 960:
| PRINT "The Current Settings are":
| PRINT TMS;TAB(15);DTS
|
| 820 INPUT "Change setting? Y/N ";ANS:
| IF ANS="N" OR ANS="n" THEN RETURN
|
| *** Update date including day of week
| 830 VTAB 14:PRINT "Enter new values or use arrow key to re-enter old values":
| VTAB 16:PRINT "Minutes ";MI$:
| VTAB 16:VTAB 8:INPUT " ";MI$:
| IF MI$<59 OR MI$<0 THEN GOTO 830
|
| 840 VTAB 17:PRINT "Hour ";HH$:
VTAB 17:HTAB 5:INPUT " *";HH%:
   IF HH%<9 OR HH%>24 GOTO 840
   845 IF HH%>0 AND HH%<13 THEN Gosub 900
   
   850 VTAB 18:PRINT "Day of week (1-7) " ;DW%:
   VTAB 18:HTAB 18:INPUT " *" ;DW%:
   IF DW%<1 OR DW%>7 THEN GOTO 850
   
   860 VTAB 19: PRINT "Day of month ";DD%:
   VTAB 19:HTAB 13: INPUT " *" ;DD%:
   IF DD%<1 OR DD%>31 THEN GOTO 860
   
   870 VTAB 20:PRINT "Month (1-12) ":MM%:
   VTAB 20:HTAB 13:INPUT " *" ;MM%:
   IF MM%<1 OR MM%>12 THEN GOTO 870
   
   880 & 0,3,MIA:
   & 0,4,HH%:
   & 0,5,DW%:
   & 0,6,DD%:
   & 0,7,MM%:
   GOTO 800
   
   [************************************************************************]
   
   900 VTAB 18:
   INPUT "AM or PM? ":PMS:
   PMS=LEFT$(PMS,1):
   IF PMS<>"P" AND PMS<>"p" AND PMS<>"A" AND PMS<>"a" THEN GOTO 900
   
   910 IF (PMS="P" OR PMS="p") AND HH%<12 THEN
   HH%=HH%+12
   
   915 IF (PMS="A" OR PMS="a") AND HH%<12 THEN
   HH%=0
   
   920 RETURN
   
   [************************************************************************]
   
   930 IF Mi%<0 OR Mi%>59 OR HH%<0 OR HH%>23 THEN
   TMS=" <no time> ":
   RETURN
   
   935 IF HH%=0 THEN HHS="12": PMS=" am"
   940 IF HH%>0 AND HH%<12 THEN PMS=" am":
   HHS=STR$(HH%)
   945 IF HH%=12 THEN PMS=" pm": HHS="12"
   950 IF HH%>12 THEN PMS=" pm":
   HHS=STR$(HH%)
   
   955 TMS=HHS+":"+RIGHT$("0"+STR$(Mi%),2)+PMS:
   RETURN
   
   [************************************************************************]
   
   960 IF DD%<1 OR DD%>31 OR MM%<1 OR MM%>12 OR DW%<1 OR DW%>7 THEN
   D2TS=" <no Date>":
   RETURN
   
   970 WKS="MonTueWedThrFriSatSun"
   
   980 D2TS=MIDS(WKS,DW%*3-2,3)+":",
   D2TS=D2TS+MIDS(MTHS,MM%*3-2,3)+":",
   D2TS=D2TS+STR$(DD%):
RETURN

******************************************************************************
ABORT ROUTINE
******************************************************************************

1010 AS="Missing file "+FF$:
   GOTO 1050

1020 TEXT:
   ( PERT.BIN DE-INITIALIZER )
   CALL 16386:
   AS="Error collecting data."
   GOTO 1050

1030 PRINT DS:"CLOSE":
   AS="Error writing to file "+FF$:
   GOTO 1050

   ONERR correction
1050 CALL -3288:
   Resume normal error handling
   POKE 216,0:
   Clear screen to bottom
   VTAB 16:CALL -958:
   PRINT G$:A$:
   PRINT "Program aborts!");
   END

B.1.2 Pert.bin

******************************************************************************
Program Pert.bin
Language 6502 Assembly
Purpose Interrupt routine to analyze real time perturbator displacement
Equipment XpClock, 1989 Perturbator/Apple Interface
            Enhanced Apple IIe (Interrupt RTI without restore $45)
******************************************************************************
Program Executes from $4000 initialize, $4002 deinitialize
XpClock in slot 5
Interface card in slot 4

   Call to $4000 (CALL 16384) initializes routine
   and sets interrupts causing depth to be sent out,
   and an average peak to peak amplitude to be maintained.

   Call to $4002 (CALL 16386) de-initializes routine (stops interrupts)

   The constants at $300-304 are preset.
   Data is transferred at $306-309
   $4800-$4FF are previous Vpp for averaging (accessed by irpt program)
******************************************************************************

   Interrupt handler:
   * K is the count of samples in the cycle, beginning at the last maximum or
     minimum and ending at the end of the next window. The beginning of the
     next window is calculated as the half cycle points from the max/min of
     the previous window minus a half window.
   * BWIND=HCYCLE-WIND/2
   * EWIND=HCYCLE+WIND/2
BWIND and EWIND are input as number of samples from PERT.DATA

wind:
| + | + |
|++|++|
++| + |
| wind |
| half cycle |

On each interrupt entry:
- a) set up next interrupt 100ms later
- b) output depth if new
- c) fetch/store data sample
- d) reset on overflow or underflow
- e) check for new max or min for window
- If K=EWIND after increment (last sample in window)...
- f) calculate the Amplitude from previous max/min,
- g) compute the running SUM and average amplitude
- h) K=K×XVAL

Input Constants:
- NPAVE = log base 2 of the number of amplitudes to average
  - 2=1, 4=2, 8=3, 16=4, 32=5, 64=6, 128=7, 256=8
- NPAVE2X = the number of amplitudes to average
- BWIND = beginning point of window (for reset only)
- EWIND = end point of window
- Integer arithmetic in 8 or 16 bits as appropriate
- Reset Doppler box and routine on underflow or overflow.

I/O AND VARIABLE LOCATIONS

Program locations
APROG .EQ $4000
PKPK .EQ APROG+$400
PVPV .EQ $3FE

Device Input/Output locations
DISP .EQ $C0C0
DEPTH .EQ $C0C1
RESET .EQ $C0C2
CLOCK .EQ $C0DD
CLOCK .EQ $C5F0
SCREEN .EQ $0750

User Input/Output
NPAVE .EQ $300
Number of amplitudes to average
BWIND .EQ $302
Beginning of window within which to scan for max/min
EWIND .EQ $303
End of window within which to scan for max/min
LIMH .EQ $304
Upper limit for any range value before reset
LIML .EQ $305
Lower limit for any range value before reset
AVEPK .EQ $306
8-bin average displacement
VALOK .EQ $307
FF=good data, 0=bad data, don’t use
DPTH .EQ $308
Depth at which to find AVEPK
READX .EQ $309
New value signal for PERT.DATA

.OP $5C02
.Use enhanced code
.OR APROG
.TA $4000

INIT BRA DOINIT
Jump to initialize routine
DENIT BRA DODENIT
Jump to de-initialize routine
Data Table

Safe for monitor, AppleSoft, and DOS.

**init'd zero **=constants **=returned values

ATABL .EQ *

AMP .DA #0 * Data table starts here
DIG1 .DA #0 Amplitude (peak to peak) of current cycle
DIG2 .DA #0 Hundreds place digit to be printed to screen
DIG3 .DA #0 Tens place digit to be printed to screen
FAC1 .DA #0 Ones place digit to be printed to screen
FAC2 .DA #0 Factor one in division routine
KERR .DA #0 * Factor two in division routine
KNT .DA #0 * Counts no. of consecutive errors
LDPTH .DA #0 * Count of samples of this cycle
LMNVAL .DA #0 Previous depth
LMXVAL .DA #0 Previous window min value
MNVAL .DA #0 Previous window max value
MAXMIN .DA #0 Pointer to MIN or MAX subroutine (flip-flops)
MNVAL .DA #0 Current window min value
MXVAL .DA #0 Current window max value
NERR .DA #8 * Number of consecutive errors allowed before reset
SDM .DA #0 * 16-bit sum of previous NPAVE Vpp
TEMP .DA #0 8-bit work area for shifting SDM
XMNVALL .DA #0 Current min location in cycle
XXVALL .DA #0 Current max location in cycle
XPREK .DA #0 * Pointer to current Fk to Fk value

DE-INITIALIZATION

DODINIT STZ CLOCK+9
LDA #$ECC
STA CLOCKC
SEI
RTS
Return

INITIALIZATION

*** Initialize: Setup clock and start interrupts ***

DOINIT SEI
LDA #$AIRP
STA IRPV
LDA /AIRP
(Address AIRP)

STA IRPV+1

LDA #1
STA CLOCK+9
STA CLOCKC
Turn on clock interrupt-accd-clock-RAM-regns.

STZ DEPTH
STZ LDPTH
LDA #MAX
STA MAXMIN
LDA /MAX
STA MAXMIN+1

JSR ZERO

CLI
Enable interrupts
RTS
 Exit

ZERO STZ KERR
STZ VALOK
LDA #$DFF
LDA #30
STA SCREEN,X

Error count.
Tell user next values are no good
Get ASCII for "--"
and print that on screen loc 30
LDA #1
STA READGO

STZ SUM
STZ SUM+1
STZ XPFPK

LDX #0
STZ PKPK,X
DEX
BNE .4

STZ RESET

LDA BWIND
DEC
STA KNT

STA XMVAL
STA XNVAL
LDA DISP
STA XMVAL
STA XMVAL
STA XMVAL
STA XMVAL

RTS

------------------------------------------------------------------------

Enhanced interrupt handling saves all registers for us

IRFT
JSR SERVE
Jump to interrupt service routine
JSR WIND
Jump to window processing

LDA KNT
IF this is the last sample
SEC
(Set carry flag for comparison)
CMP EWIND
in the cycle, then
BCC EIRFT
JSR ECYCLE
Jump to the end of cycle processing

EXIRFT
RTI
Exit from the interrupt

SERVE
LDA CLOCK+8
Service clock interrupt

LDA DPTH
Get a new depth from PERT.DATA
CMP LDPHT
If this is a new depth
BEQ INKNT
then

STA LDPHT
STA DEPTH
output the new depth
LDX #0
and print the new depth
JSR PValue
to the screen at location 0
JSR ZERO
Zero out data for averaging

INKNT
INC KNT
Increment the cycle counter
RTS
Return to IRFT

------------------------------------------------------------------------

WIND
LDA DISP
Get the current displacement from the Doppler
SEC
(Set carry for comparison)
CMP LIMH
If the displacement is too big
BCC ERROR
then do error processing
SEC
(Set carry for comparison)
CMP LIML
If the displacement is too small
BCC ERROR
then do error processing
STZ KERR
If no error, reset error count
INWIND  LDX KNT
  SEC  Compare DISP with MXVAL
  CMP MXVAL  Have we got a new maximum?
  BCC CHMIN  No then go to check for a new minimum
  STA MXVAL  Save then new maximum
  STX XMXVAL  and its pointer
CHMIN SEC  Compare DISP with MNVAL
  CMP MNVAL  Have we got a new minimum?
  BCS EXWIND  No then go to exit window
  STA MNVAL  Save new minimum
  STX XMNVAL  and its pointer
EXWIND RTS

ERROR PHA  Save the old displacement
  INK KERR  Count this error
  LDA KERR  If the error count is
  SEC (Set carry flag for comparison)
  CMP NERR  high enough
  BCC EXERR  then exit
  PLA  Retrieve the old displacement
  JSR ZERO  reset this program
  LDA DISP  Get the new reset displacement
  JMP WIND
EXERR PLA DISP  Retrieve the old displacement
  JMP INWIND  Return

ECYCLE NOP  Jump to the appropriate routine
  JMP (MAXMIN)  (Address stored in MAXMIN)
  ECYC2 LDA AMP  Store the current amplitude
  INC XPKPK  in the next location
  LDX XPKPK  of the PKPK buffer
  STA PKPK,k  LDX #10  and print it to the
  JSR PVALUE  screen location 10
  ALL THIS IS TO GET THE RESULT
  SUM=SUM+PKPK(XPKPK)-PKPK(XPKPK-NPAV2X)
  WHERE SUM IS A 16 BIT NUMBER (THE RUNNING SUM)
  CLC  ADC SUM+1  Add Amp to the lower 8 bits of SUM
  PHA  SAVE THAT ON STACK
  BCC SUBTRA  IF CARRY FLAG WAS SET THEN
  INC SUM  INCREMENT UPPER 8 BITS OF SUM
SUBTRA SEC  SET THE CARRY
  LDA XPKPK  MOVE XPKPK TO ACCUMULATOR
  SBC NPAV2X  ACC=XPKPK-NPAV2X THE WRAP IS INTENTIONAL.
  TAX  MOVE INDEX TO X BUFFER
  SEC  SET THE CARRY FOR SUBTRACTION
  PLA  FETCH LOWER 8 BITS OF SUM
  SBC PKPK,x  SUBTRACT FROM THAT THE SAMPLE NPAV2X
  STA SUM+1  KEEP THIS
BCS DIV  If a borrow was needed then
DEC SUM  decrement the upper 8 bits of SUM

Now we can divide SUM by NPAVE-2 by shifting SUM to the right
NPAVE times (equivalent to dividing by two NPAVE times)

DIV  Move the upper 8 bits of SUM
LDA SUM   into TEMP for shifting
STA TEMP
LDA SUM+1  Keep the lower 8 bits of SUM in Accumulator
LDX NPAVE  Set number of right shifts for SUM
'
1  Shifting bit 9 into carry
LSR TEMP  Moving carry into lower 8 bits and drop bit 1
ROR
DEX  Decount shift count
BNE .1  Stop when shift count is 0

STA AVEPK  Send average to the user
LDX #20  and print it to
JSR PVALUE  screen location 20

LDA XPKPK  If we have collected
SEC  (Set carry flag for comparison)
CMP NPAV2X  NPAV2X or more samples
BCC EXECYC  then

LDA #$FF  set VALOK flag to
STA VALOK  good data
LDA #$AB  and print "="
LDX #$30  at screen location 30
STA SCREEN,X

EXECYC  Tell the user there
LDA #1  STA READOK  is new data to read
RTS

MAX  AMPL-MXVAL-MNVAL
SEC
LDA MXVAL
SBC MNVAL
STA AMP

BCS GOODMX  If carry set (AMP is positive then continue)

JSR ZERO  Else reset routine
RTS  and return

GOODMX  Save maximum in LMXVAL for next
LDA MXVAL  peak to peak calculation
STA LMXVAL

LDA KNT  Reset counter to:
SBC XMIVAL  KNT=KNT-XMIVAL
STA KNT

LDA XMIVAL  If the minimum came after the maximum
SEC  (Set carry flag for subtraction)
SBC XMIVAL  then
BCC RSTMX

STA XMIVAL  keep the minimum for comparison (XMIVAL=XMIVAL-XMXVAL
JMP ENDMX  to adjust for counter reset)

RSTMX  Otherwise use the current displacement
LDA KNT  for the minimum comparison in
STA XMIVAL  the next window
STA MNVAL
ENDMX  LDA KNT
STA XMVAL
LDA DISP
STA MXVAL

* LDA #MIN
STA MAXMIN
LDA /MIN
STA MAXMIN+1

* JMP ECYC2

*******************************************************
MIN  SEC  AMP=LMXVAL-MXVAL
LDA LMXVAL
SEC MNVAL
STA AMP

* BCS GOODMN
If carry flag set (AMP positive), continue

* JSR ZERO
Else reset routine
and return

GOODMN  LDA MXVAL
STA LMXVAL
LDA KNT
SBC XMVAL
STA KNT

* LDA XMVAL
SEC
SBC XMVAL
If the maximum came after the minimum
BCC RSTMN
then

* STA XMVAL
JMP ENDMN

* RSTMN  LDA KNT
STA XMVAL
LDA DISP
STA MXVAL

* ENDMN  LDA KNT
STA XMVAL
LDA DISP
STA MNVAL

* LDA #MAX
STA MAXMIN
LDA /MAX
STA MAXMIN+1

* JMP ECYC2

*******************************************************
PVALUE  STA TEMP
LDA #5A0
STA DIG1
in hundreds and
STA DIG2
tens place.

* STZ FAC1
Zero out FAC1 for divide routine
LDA #100
STA FAC2
LDA TEMP
Get number and
JSR DIVIDE
Divide by FAC2

* PRA Temporarily save remainder.
LDA FAC1
Hundreds digit comes back in FAC1
If zero, leave hundreds place blank, goto 10's

ORA $80
STA DIG1
LDA $80
STA DIG2

Now divide remainder by 10
LDA #10
STA FAC2
STZ FAC1
PLA
JSR DIVIDE

Divide by FAC2
ORA $80
STA DIG3

Ones digit returned in A, get ASCII character
of ones digit
LDA FAC1
BCH PRNT
ORA $80
STA DIG2

If zero, leave DIG2 as is and print number
Tens returned in FAC1
Otherwise get ASCII character
for tens digit

PRNT
LDA DIG1
STA SCREEN,X
INX
LDA DIG2
STA SCREEN,X
INX
LDA DIG3
STA SCREEN,X

Store first digit
Store second digit
Store third digit

in hundreds place
in tens place
in ones place

Retrieve original number
and return

******************************************************************************
DIVIDE SEC
CMP FAC2
BLT EXDIV
INC FAC1
SEC
SBC FAC2
JMP DIVIDE

If remainder less than factor,
then return
Else add one to other factor,
set carry for subtraction
and subtract FAC2 from the remainder
Repeat until done

******************************************************************************
AIRP .EQ IRPT
.EN

B.1.3 Pert.comb

******************************************************************************
Program Pert.comb
Language Applesoft
Equipment Apple IIe
******************************************************************************

Data
FQ=Frequency of perturbation (Hz)
SD=Peak to peak amplitude of perturbator vibration at skin surface (mm)
DEPTH=Depth (cm)
DISP=Vpp (Displacement) (mm)
NS=Number of sample points

INITIALIZATION
10 GOSUB 100
COMBINE DATA
GOSUB 200

Clear screen to bottom, Clear keyboard
VTAB 13:CALL -958:POKE -16368,0:
INPUT "Process data? Y/N ":ANS:
IF ANS="Y" OR ANS="y" THEN
PROCESS DATA
PRINT DS:"RUN PERT. ANLS.BAS"

Clear screen to bottom, Clear keyboard
VTAB 13:CALL -958:POKE -16368,0:
INPUT "End now? Y/N ":ANS:
IF ANS="Y" OR ANS="y" THEN
HOME:CLEAR:END

PRINT DS:"RUN PERT. DATA.BAS"
END

INITIALIZATION

Initialize variables
DIM DEPTH(255),DISP(255):
DIM COM5(10),SFL5(10),F%(10):
DS=CHR$(4):
GS=CHR$(7):
CS=".";
FF$="PERT.TEMP":
ONERR GOTO 500
PRINT DS:"VERIFY":FF$:
ONERR GOTO 520
PRINT DS:"OPEN":FF$:
PRINT DS:"READ":FF$:
INPUT INDEX%,DS%,DITS
FOR I=1 TO INDEX%:
INPUT COM5(I):
NEXT I
PRINT DS:"CLOSE":FF$:
RESUME normal error handling
POKE 216,0
RETURN

COMBINE DATA ROUTINE

FOR I=1 TO 10:
SFL5(I)=" ":
NEXT I:
IX%=0
HOME:VTAB 3:
PRINT "You have collected this data:";
FOR I=1 TO INDEX%:
PRINT I;SFLS(I);COMS(I):
NEXT I

220 PRINT "Enter number (then return) of files to":
PRINT "combine for report. Return at end."

230 INPUT ANS:
IF ANS<="" THEN
IX%=IX%+1:
F%(IX%)=VAL(ANS):
SFLS(F%(IX%))=".;*":
IF IX%<10 THEN GOTO 230

COPY FILE
250 IF IX%>0 THEN GOSUB 300

260 HOME;VTAB 3:
PRINT "Data not used will be deleted."="used":
FOR I=1 TO INDEX%
PRINT I;SFLS(I):COMS(I):
NEXT I

270 INPUT "Another report? Y/N ";ANS:
IF ANS<="N" AND ANS<="n" THEN
IX%=0:
VTAB (INDEX%+4);RTAB 1:
Clear screen to bottom
CALL -958:
GOTO 220

280 FOR I=1 TO INDEX%
FFS="F";STRS(I);
PRINT DS;"DELETE";FFS:
NEXT I:
FFS="PERT.TEMP":
PRINT DS;"DELETE";FFS:
RETURN

COPY FILE ROUTINE

*******

300 HOME;VTAB 6:
PRINT TAB(6);"PERTURBATOR":
PRINT TAB(4);"DATA COLLECTION":
PRINT TAB(8);"PROGRAM":
PRINT TAB(6);"DS":

310 PRINT:INPUT "Patient initials ";PX$:
FFS="/PERT.DAT/";PX$+DS:
PRINT "Combining ":IX%:" files into":
INVERSE;PRINT FFS;NORMAL

320 FOR I=1 TO IX%
PRINT COMS(F%(I)):
NEXT I:
ONERR GOTO 345

330 PRINT DS;"VERIFY";FFS:
PRINT "File";FFS;" exists.";
INPUT "Delete? Y/N ";ANS:
IF ANS="N" OR ANS="n" THEN GOTO 300

340 PRINT DS;"DELETE";FFS:
GOTO 350
345 CALL -3288
350 ONERR GOTO 510
360 PRINT DS:"OPEN":FFS:
    PRINT DS:"WRITE":FFS:
    PRINT PX$;CS;DT$;CS;IX$:
    FOR I=1 TO IX$:
    FILES$= F$+STRS(F$(I)):
    PRINT DS:"OPEN":FILES:
    PRINT DS:"READ":FILES
370 INPUT NS$, FO, SD$:
    FOR J=1 TO NS$:
    INPUT DEPTH(J),DISP(J):
    NEXT J:
    PRINT DS:"CLOSE":FILES
380 PRINT DS:"WRITE":FFS:
    PRINT CM$(F$(I)):
    PRINT NS$; CS; FO; CS; SD$:
    FOR J=1 TO NS$:
    PRINT DEPTH(J),CS,DISP(J):
    NEXT J
390 NEXT I:
    PRINT DS:"CLOSE":FFS:
    Resume normal error handling
    POKE 216,0:
    RETURN

********* ABORT ROUTINE

500 A$="Missing file "+FFS$:
    GOTO 550
510 PRINT DS:"CLOSE":
    A$="Error reading file "+FILES$:
    A$=A$+CHR$(13)+"or writing file "+FFS$:
    GOTO 550
520 PRINT DS:"CLOSE":FFS$:
    A$="Error reading file "+FFS$:
    GOTO 550

ONERR correction
550 CALL -3288:
    Resume normal error handling
    POKE 216,0:
    Clear screen to bottom
    VTAB 16:CALL -958:
    PRINT GS:A$:
560 INPUT "Abort (y/n) ? ":A$:N$:
    IF A$="y" OR A$="y" THEN END
570 PRINT DS:"RUN PERT.COMB.BAS":
END

B.1.4 Pert.calib
Program: Pert.calib
Language: Applesoft
Equipment: Apple IIE, Perturbator Interface card (Slot 4) and
Clock card (Slot 5)

 Initialize
10 GOSUB 100
 Calibrate
20 GOSUB 200

30 PRINT D$:"RUN PERT.DAT.A.DAS"

40 END

*****************************************************************************

* INITIALIZE *
*****************************************************************************

Activate Interrupt Software for Displacement monitoring

Constants:
FQ=frequency of perturbation (Hz)
RAN=sampling rate is pre-set in interrupt routine to 10ms
PWA=percentage of half cycle to use for window

Data transfer locations
S300 NVA A0 Log base two of the number of peaks of which to maintain
an average amplitude.
S301 NAVG A1 Number of peaks of which to maintain an average amplitude.
S302 BWIND A2 Beginning of window in which to scan for max and min.
S303 EWIND A3 End of window in which to scan for max or min.
S304 ULA A4 Upper limit of displacement before reset.
S305 LLA A5 Lower limit of displacement before reset.
S306 DISP A6 Average amplitude peak to peak.
S307 ERR A7 Error flag. 0 = bad data. 255 = good data.
S308 DEPTH A8 POKE value to output 0=0cm to 255=3.825cm.
S309 DOKX A9 Flag. 1 = new data to read. 0 = no new data.

100 D$=CHR$(4):
110 DIM A1(9):
111 FOR I=0 TO 9:
112 A1(I)=768+I:
113 NEXT I

114 POKE A1(0),5: POKE A1(1),32:
115 POKE A1(2),2: POKE A1(3),12:

120 HOME: VTAB 2: PRINT " Ultrasonic Displacement":
121 PRINT " Calibration":
122 Set screen to lines 6 through 20
123 POKE 34,6: POKE 35,20:

130 HOME:
131 PRINT "You will need:":
132 PRINT " Oscilloscope (scope) and connectors,":
133 PRINT " Screwdriver,":
134 PRINT " Ultrasonic Displacement (UD) box."
135 PRINT "Remove box top and hole covers from":
136 PRINT "the UD box bottom. Connect UD box":
137 PRINT "to the computer and turn it on":
138 INPUT "Press return to continue";ANS
150 RETURN

CALIBRATION

200 HOME:
  PRINT "Connect UD audio outputs to X and Y":
  PRINT "inputs of scope."
  PRINT "Set scope X and Y gains equal."
210 PRINT "Connect probe to UD box."
  PRINT "Perturbate skin."
  PRINT "Adjust PHASE and BALANCE until picture":
  PRINT "is as round as possible."
  INPUT ANS

220 HOME: VTAB 19:
  PRINT "DEPTH";TAB(10);"AMPLITUDE";TAB(20);"AVE AMP";TAB(30);"DATA GOOD":
  VTAB 20:
  PRINT ".15mm";TAB(10);".04mm";TAB(20);".04mm";TAB(30);"= or ="
  Start interrupts
230 CALL 16384

240 VTAB 7:PRINT "Set DEPTH on UD box to ext."
  PRINT "Connect XBAT to trigger input of scope."
  PRINT "Connect GATE to vert. input of scope."
245 PRINT "Set scope timerbase for 5 usec/div."
  PRINT "Set scope trigger for a pos signal."
  PRINT "Adjust DELAY pot for a 32 usec delay."

250 POKE A8(8),200:
  Repeat until key pressed
  KEY=PEEK(49152):
  IF KEY<128 GOTO 250

  Clear keyboard
260 POKE 49168,0
  Stop interrupts
270 CALL 16386

280 HOME: PRINT "During normal operation."
  PRINT "front panel should be set as follows":
  PRINT "POWER ON";TAB(15);"on"
  PRINT "PROBE";TAB(15);"Connected to sensor"
290 PRINT "DEPTH";TAB(15);"ext"
  PRINT "SIGNAL LEVEL";TAB(15);"Almost full scale"
  PRINT "POLARITY";TAB(15);"away"
300 PRINT "FULL SCALE";TAB(15);"10 mm"
  PRINT "OUTPUT LIM";TAB(15);"8 bits"
  PRINT "RESET POINT";TAB(15);"5 volts"
310 PRINT "AC LINE";TAB(15);"To power supply"
  PRINT "DIGITAL I/O";TAB(15);"To computer (slot 4)"
  INPUT ANS
  POKE 34,0:POKE 35,23:HOME
320 RETURN
C.2 Data analysis programs

C.2.1 analysis.c

#include <stdio.h>
#include <math.h>
main()
{  
    FILE *fopen(),*f1,*f2;
    int i,count,fend,start;
    int bstart,bstart,bend,flag;
    float depth[300],disp[300];
    float alpha,gamma,ymfat;

    /*---------------------------------------------------------------------*/
    /* OPEN FILES FOR DEPTH AND AMPLITUDE DATA */
    /*---------------------------------------------------------------------*/

    f1 = fopen("disp","r");
    f2 = fopen("depth","r");

    i = 0;

    /*---------------------------------------------------------------------*/
    /* INPUT DATA */
    /*---------------------------------------------------------------------*/

    while(fscanf(f1,"%f",&disp[i])!=EOF)
        i++;
    count = i;
    for(i=0;i<count;i++)
        fscanf(f2,"%f",&depth[i]);

    /*---------------------------------------------------------------------*/
    /* CALL THE FUNCTION TO IDENTIFY FAT REGION */
    /*---------------------------------------------------------------------*/

    fend = fat(depth,disp,count);

    /*---------------------------------------------------------------------*/
    /* PRINT FAT INFORMATION */
    /*---------------------------------------------------------------------*/

    printf("FAT REGION\n");
    printf("fat ends at %f cm\n",depth[fend]);

    /*---------------------------------------------------------------------*/
    /* CALL THE FUNCTION TO CALCULATE THE CURVE FITTING PARAMETERS */
    /*---------------------------------------------------------------------*/

    if(fend>5){
        start = 0;
        fit(depth,disp,start,fend,&alpha,&gamma);
        printf("fit parameters: alpha = %f & gamma = %f\n",alpha, gamma);

        /*---------------------------------------------------------------------*/
        /* CALL THE FUNCTION TO CALCULATE THE ELASTIC MODULUS OF THE FAT REGION */
        /*---------------------------------------------------------------------*/

        modulus(&gamma,&ymfat);
        printf("ymfat = %f\n",ymfat);)
}
/* CALL THE FUNCTION TO IDENTIFY */
  /* BONE REGION */
/*------------------------------------*/

bnstart = bone(disp,fend,count);

/* PRINT BONE INFORMATION */

printf("BONE REGION\n");
if(bnstart==0) {
  printf("bone region not identified\n");
  bnstart = count - 1;
} else
  printf("bone starts at \%f cm. \n",depth[bnstart]);

/*------------------------------------*/
/* CALL THE FUNCTION TO IDENTIFY */
/* BLOOD REGION */
/*------------------------------------*/

printf("BLOOD REGION\n");
flag=blood(depth,disp,fend,bnstart,count,&bstart,&bend);

if(flag==2) {

/*------------------------------------*/
/* PRINT BLOOD INFORMATION */
/*------------------------------------*/

printf("bstart = \%f cm. \t bend = \%f cm.\n",depth[bstart],depth[bend]);
  bnstart = bstart;

/*------------------------------------*/
/* RECURSIVE CALL TO BLOOD FUNCTION */
/* TO CHECK FOR BLOOD EVERYWHERE */
/*------------------------------------*/

flag=blood(depth,disp,fend,bnstart,count,&bstart,&bend);
} else
  printf("blood not identified\n");


/*------------------------------------*/
/* FUNCTION TO CALCULATE THE END OF FAT REGION */
/* ON THE AMPLITUDE-DISPLACEMENT DATA USING THE */
/* THE FLAT SLOPE ALGORITHM */
/*------------------------------------*/

fat(depth,disp,count)
int count;
float depth[],disp[];
{
  int i,numb;
  float fend,slope[300];
/**
 * CALL THE FUNCTION TO CALCULATE
 * THE SLOPE AT EACH DEPTH
 */

derivative(disp, slope, count):

i = 0;
numb = 0;
fend = 0;

/**
 * FLAT SLOPE ALGORITHM
 */

while (numb < 4) {
  if (slope[i] < -0.6)
    { numb++; }
  else { numb = 0; }
  i++;
  fend = i-1;
return(fend);
}

/**
 * FUNCTION TO CALCULATE THE SLOPE OF THE
 * AMPLITUDE-DEPTH CURVE AT EACH DEPTH USING
 * MULTISTEP ALGORITHM
 */

derivative(disp, slope, count)

int count;
float disp[], slope[];

{
  int i, count2, count3, count4;
  float step;

  count2 = count - 1;
  count3 = count - 2;
  count4 = count - 3;

  /**
   * STEP SIZE
   */

  step = 0.015;

  /**
   * THREE POINT SLOPE FOR FIRST POINT
   */

  slope[0] = (-3*disp[0] + 4*disp[1] - disp[2])/(2.0*step);

  /**
   * TWO POINT SLOPE FOR SECOND POINT
   */
slope[1]=(disp[2]-disp[0])/(2.0*step);

rief FOUR POINT SLOPE FOR ALL **
** MIDDLE POINTS ***/

for(i=2;i<count3;i++)
slope[i]=(disp[i-2]-8*disp[i-1]+8*disp[i+1]-disp[i+2])/(12*step);

rief TWO POINT SLOPE FOR THE **
** SECOND LAST POINT ***/

slope[count3]=(disp[count2]-disp[count4])/2.0*step;

rief THREE POINT SLOPE FOR THE LAST POINT ***/

slope[count2]=(3*disp[count2]-4*disp[count3]+disp[count4])/2*step;

return:

重要作用：FUNCTION TO CALCULATE FIT PARAMETERS **
** ALPHA AND GAMMA FOR A CURVE ***/

fit(depth,disp,start,fend,alpha, gamma)

int fend;
float depth[], disp[], *alpha, *gamma:
{
    int i, total;
    float sumx, sumindisp, sumindisp2;
    float sumx2, sumg;

    /* INITIALIZING THE SUMMATION VARIABLES */
    /****
    sumx = 0;
    sumindisp = 0;
    sumindisp2 = 0;
    sumx2 = 0;
    sumg = 0;
    total = fend+1-start;
    
    /* CALCULATING THE SUMMATIONS */
    /****
    for(i=start;i<=fend;i++)
    {
        sumx += depth[i];
        sumindisp += log(disp[i]);
        sumindisp2 += log(disp[i])*depth[i];
        sumx2 += depth[i]*depth[i];
    

/*-----------------------------*/
/* CALCULATING GAMMA */
/*-----------------------------*/

*gamma = (sumx*sumlnDisp-total*sumlnDisp)/(total*sumxSq-sumx*sumx);

/*-----------------------------*/
/* CALCULATING THE EXTRA SUMMATION */
/* TERM FOR ALPHA */
/*-----------------------------*/

for(i=start;i<fend;i++)
    sumx += (*gamma)*depth[i]*depth[i];

/*-----------------------------*/
/* CALCULATING ALPHA */
/*-----------------------------*/

*alpha = exp((sumx+sumlnDisp)/sumx);
    return;
}

/*******************************/
/* FUNCTION TO CALCULATE THE ELASTIC MODULUS */
/* GIVEN THE VALUE OF GAMMA */
/*******************************/

modulus(gamma, ym)
float *gamma, *ym:
{
    *ym = 0.4835941/(*gamma)*(*gamma);
}

/*******************************/
/* FUNCTION TO IDENTIFY THE BONE REGION */
/* INDICATING ITS START */
/*******************************/

bone(disp, fend, count)
int fend, count;
float disp[];
{
    int end, i, min, compare, max;
    int flag, bnstart:

    /******************************/
    /* INITIALIZE VARIABLES */
    /******************************/

    bnstart = 0;
    flag = 0;
end = count - 1;
i = end-2;
min = end;
compare = end;

while((compare>end) & (flag==0))
  if(min==compare) & (i>end))
  {
    
    /* SEARCH FOR A STEADY MINIMUM STARTING */
    /* FROM THE END DEPTH */
    
    if((disp[i-1]>disp[i]) & (disp[i-2]>disp[i-1]))
      if((disp[i+1]>disp[i]) & (disp[i+2]>disp[i+1])) min = i; 
      i--; 
    
    /* IF MIN NOT FOUND EXIT */
    
    if(min==compare) {
      flag = 1;
      break;
    }
    
    /* BOUNDARY DEPTH CONDITIONS FOR EXIT */
    
    if((i<=end) & (i>(end-10))) {
      flag = 1;
      break;
    }
    
    /* SEARCH FOR THE FIRST STEADY MAXIMUM */
    /* ON THE RIGHT OF THE MINIMUM */
    
    if(min<(end-10))
    {
      max = min;
      i = min;
      while((max==min) & (i<(end-5)))
        if((disp[i-1]<disp[i]) & (disp[i-2]<disp[i-1]))
          if((disp[i+1]<disp[i]) & (disp[i+2]<disp[i+1])) max = i; 
          i++; 
        
        /* IF MAX NOT FOUND EXIT */
        
        if(max==min) {
          flag = 1;
          break;
        }
    
    /* END DEPTH CONDITION FOR EXIT */
    
    if(i==(end-5))
    {
      flag = 1;
      break;
    }

}

/*/-------------------------------------------------------*/
/*/ USE MAX-MIN DIFFERENCE FOR */
/*/ BONE IDENTIFICATION */
/*/-------------------------------------------------------*/
{if((disp[max]-disp[min])>.4) {
    bstart = min;
    flag = 1;
    break;
}
else {
    compare = min;
    i = min;
};
}

return(bnstart):


/**-------------------------------------------------------*/
/** FUNCTION TO CALCULATE THE BLOOD VESSEL */
/** REGIONS ACCORDING TO THE PSEUDO */
/** MODULUS CRITERION */
/**-------------------------------------------------------*/

blood(depth, disp, fend, bnstart, count, bstart, bend)

int bnstart, fend, count, *bstart, *bend:
float depth[], disp[];
{
    int end, i, min, compare, max:
    int flag:
    float ymblood, alpha, gamma:

    /**-------------------------------------------------------*/
    /** INITIALIZE VARIABLES */
    /**-------------------------------------------------------*/

    flag = 0;
    bstart = 0;
    bend = 0:
    end = bnstart;
    if(bnstart == 0) end = count - 1;
    i = end - 2;
    min = end;
    compare = end;

    while((compare>fend)&(flag==0))
    {
        while((min==compare)&(i>fend))
        {
            /*-------------------------------------------------------*/
            /* SEARCH FOR A STEADY MINIMUM STARTING */
            /* FROM THE END DEPTH */
            /*-------------------------------------------------------*/

            if((disp[i-1]>disp[i])&disp[i-2]>disp[i-1])
            {


```
if((disp[i+1]>disp[i])&(disp[i+2]>disp[i+1])) min = i; 

if(compare) 
  flag = 1;
  break;

if((i<end)&&(min>(end-10))) 
  flag = 1;
  break;

max = min; 
i = min; 
while((max==min)&&(i>end)) 
  if((disp[i-1]<disp[i])&(disp[i-2]<disp[i-1])) 
    if((disp[i+1]<disp[i])&(disp[i+2]<disp[i+1])) max = i; 
    i--;

if(max==min) 
  flag = 1;
  break;

if(i==(end-5)) 
  flag = 1;
  break;

if(flag=1) break;
fit(depth,disp,max,min,&alpha,&gamma);
ymblood = modulus(gamma);

if(ymblood < .2) 
  bstart = max; 
bend = min;
flag = 2;
break;
else {compare = min;
i=min;}
}
return(flag);

C.2.2 muscle.c

#include <stdio.h>
#include <math.h>

main()
{
FILE *fopen(), *f1, *f2;
int i, j, count, number;
int flag1, flag2, i1, i2;
float depth[300], disp[300];
float alpha, gamma, ym, error, errate;
float d1, d2;

/**************************/
/* OPEN FILES FOR DEPTH AND AMPLITUDE DATA */
/**************************/

f1 = fopen("disp", "r");
f2 = fopen("depth", "r");
i = 0;

/**************************/
/* INPUT DATA */
/**************************/

while(fscanf(f1, "%f", &disp[i]) != EOF)
    i++;
    count = --i;
for(i=0; i<=count; i++)
    fscanf(f2, "%f", &depth[i]);

/**************************/
/* INPUT # OF TIMES ANALYSIS NEEDED */
/**************************/

printf("# of analysis ? \t ");
scanf("%d", &number);
printf("\n");

/**************************/
/* START ANALYSIS */
/**************************/

j = 1;
while(j<=number)
{

/**************************/
/* INPUT START AND END DEPTH TO BE ANALYZED */
/* AS MUSCLE */
/**************************/

printf("analysis # %d\n", j);
/**FUNCTION TO CALCULATE FIT PARAMETERS**/
/* ALPHA AND GAMMA FOR A CURVE*/

fit(deepth, disp, start, fend, alpha, gamma)

int fend;
float deepth[], disp[], *alpha, *gamma;

{
    int i, total;
    float sumx, sumlndisp, sumlndispx;
    float sumsq, sumx;

    /* INITIALIZING THE SUMMATION VARIABLES*/
    /*-------------------------------*/

    sumx = 0;
    sumlndisp = 0;
    sumlndispx = 0;
    sumsq = 0;
    sumx = 0;

    total = fend + 1 - start;

    /* CALCULATING THE SUMMATIONS*/
    /*-------------------------------*/

    for (i = start; i < fend; i++)
    {
        sumx += deepth[i];
        sumlndisp += log(disp[i]);
        sumlndispx += log(disp[i]) * deepth[i];
        sumsq += deepth[i] * deepth[i];
    }

    /* CALCULATING GAMMA*/
    /*---------------------*/

    *gamma = (sumx * sumlndisp - total * sumlndispx) / (total * sumsq - sumx * sumx);

    /* CALCULATING THE EXTRA SUMMATION*/
    /* TERM FOR ALPHA*/
    /*---------------------*/

    for (i = start; i < fend; i++)
    {
        sumx += (*gamma) * deepth[i] * deepth[i];
    }

    /* CALCULATING ALPHA*/
    /*---------------------*/

    *alpha = exp((sumx * sumlndisp) / sumx);

    return;
}
```c
/* FUNCTION TO CALCULATE THE ELASTIC MODULUS */
/* GIVEN THE VALUE OF GAMMA */
修士(gamma, ym)
float *gamma, *ym;
{
    *ym = 0.4835941/(*gamma)*(*gamma);
}

/* FUNCTION TO CALCULATE ERROR IN A BEST FIT */
err(depth, disp, il, i2, alpha, gamma, error)
int il, i2;
float depth[], disp[];
float *alpha, *gamma, *error;
{
    int i;
    float temp, er;
    er = 0.0;
    i = il;
    while (i <= i2)
    {
        temp = (disp[i] - (*alpha)*exp(-(*gamma)*depth[i]));
    }
}

C. 2. 3 graph.c
#include <stdio.h>
#include <math.h>
main()
{
    FILE *fopen(), *f1, *f2, *f3, *f4;
    int i, count, il, i2, flag1, flag2;
    float depth[300], disp[300], ndisp[300];
    float alpha, gamma, d1, d2;
}
printf("input starting depth = ");
scanf("%f", &d1); printf("\n");
printf("input ending depth = ");
scanf("%f", &d2); printf("\n");
flag1 = 0;
flag2 = 0;

/!* FIND START AND END DEPTH #S *!/  
/!*---------------------------------------------------------------*/

for(i=0;i<count;i++)
{
    if((d1<depth[i])&&(flag1 == 0)){
        i1 = i;
        flag1 = 1;
    }
    if((d2<depth[i])&&(flag2 == 0)){
        i2 = i;
        flag2 = 1;
    }
}

/!* CHECK TO SEE IF TOO LESS DATA *!/  
/!*---------------------------------------------------------------*/

if((i2-i1)<5)
{  
    printf("increase depth range\n");
    break;
}

/!* CALL THE FUNCTION TO CALCULATE *!/  
/!* THE CURVE FITTING PARAMETERS *!/  
/!*---------------------------------------------------------------*/

fit(depth, disp, i1, i2, &alpha, &gamma);
printf("fit parameters: alpha = %f & gamma = %f\n", alpha, gamma);

/!* CALL THE FUNCTION TO CALCULATE *!/  
/!* THE ELASTIC MODULUS OF THE REGION *!/  
/!*---------------------------------------------------------------*/

modulus(&gamma, &ym);
printf("ym = %f\n", ym);

/!* CALL THE FUNCTION TO CALCULATE *!/  
/!* THE ERROR IN THE FIT *!/  
/!*---------------------------------------------------------------*/

err(depth, disp, i1, i2, &alpha, &gamma, &error);
printf("error in fit = %f\n", error);

/!* CALCULATE AND PRINT THE ERROR RATE *!/  
/!*---------------------------------------------------------------*/

erate = error/(i2-i1);
printf("error rate = %f\n", rate);

j++;
}  
}
D COMPUTER INTERACTION FOR SIMULATION

-------------------------
for normal curve
-------------------------

# of analysis ? 3

analysis # 1
input starting depth = 0.5
input ending depth = 1.0

fit parameters: alpha = 3.360128 & gamma = 0.700289
ym = 0.986114
error in fit = 1.477677
error rate = 0.047778

-------------------------

analysis # 2
input starting depth = 0.75
input ending depth = 1.25

fit parameters: alpha = 4.846448 & gamma = 1.111016
ym = 0.391778
error in fit = 2.621740
error rate = 0.079447

-------------------------

analysis # 3
input starting depth = 1.0
input ending depth = 1.5

fit parameters: alpha = 6.541478 & gamma = 1.381751
ym = 0.253292
error in fit = 2.258990
error rate = 0.066441

-------------------------

20% increase of amplitude
-------------------------

# of analysis ? 1

analysis # 1
input starting depth = 0.5
input ending depth = 1.0

fit parameters: alpha = 3.148402 & gamma = 0.570826
ym = 1.484137
error in fit = 1.348711
error rate = 0.040754

-------------------------

# of analysis ? 1

analysis # 1
input starting depth = 0.75
input ending depth = 1.25

fit parameters: alpha = 4.061493 & gamma = 0.885752
ym = 0.616391
error in fit = 2.279800
error rate = 0.069085

---------------------
# of analysis?  1

analysis # 1
input starting depth = 1.0
input ending depth = 1.5
fit parameters: alpha = 4.799197 & gamma = 1.060409
ym = 0.430065	error in fit = 1.801297	error rate = 0.052979
---------------------
40% increase
---------------------
# of analysis?  1

analysis # 1
input starting depth = 0.5
input ending depth = 1.0
fit parameters: alpha = 2.964222 & gamma = 0.449750
ym = 2.390774	error in fit = 1.253163	error rate = 0.037975
---------------------
# of analysis?  1

analysis # 1
input starting depth = 0.75
input ending depth = 1.25
fit parameters: alpha = 3.481140 & gamma = 0.686610
ym = 1.025794	error in fit = 2.054263	error rate = 0.062250
---------------------
# of analysis?  1

analysis # 1
input starting depth = 1.0
input ending depth = 1.5
fit parameters: alpha = 3.718506 & gamma = 0.791241
ym = 0.772437	error in fit = 1.590854	error rate = 0.046790
---------------------
60% increase
---------------------
# of analysis ? 1

analysis # 1
input starting depth = 0.5
input ending depth = 1.0

fit parameters: alpha = 2.802624 & gamma = 0.336092
ym = 4.281199
error in fit = 1.237456
error rate = 0.037499

---------------------

# of analysis ? 1

analysis # 1
input starting depth = 0.75
input ending depth = 1.25

fit parameters: alpha = 3.038050 & gamma = 0.508533
ym = 1.870003
error in fit = 2.045779
error rate = 0.061993

---------------------

# of analysis ? 1

analysis # 1
input starting depth = 1.0
input ending depth = 1.5

fit parameters: alpha = 2.999394 & gamma = 0.560838
ym = 1.537471
error in fit = 1.437880
error rate = 0.042291

---------------------

80% increase
---------------------

# of analysis ? 1

analysis # 1
input starting depth = 0.5
input ending depth = 1.0

fit parameters: alpha = 2.659833 & gamma = 0.229073
ym = 9.215790
error in fit = 1.478147
error rate = 0.044792

---------------------

# of analysis ? 1

analysis # 1
input starting depth = 0.75
input ending depth = 1.25

fit parameters: alpha = 2.690770 & gamma = 0.347780
ym = 3.998277
error in fit = 2.246068
error rate = 0.068063

# of analysis ?  1

---

analysis # 1
input starting depth = 1.0
input ending depth = 1.5

fit parameters: alpha = 2.494963 & gamma = 0.360335
ym = 3.724491
error in fit = 1.658859
error rate = 0.048790
Figure D.1  Plots for 20% Amplitude Increase
Figure D.2 Plots for 40% Amplitude Increase
Figure D.2 Plots for 60% Amplitude Increase
Figure D.2  Plots for 80% Amplitude Increase
E COMPUTER INTERACTION FOR ANALYSIS

---------------
INTERACTION WITH analysis.c
---------------

FAT REGION
fat ends at 0.825000 cm.
fit parameters: alpha = 2.622190 & gamma = 0.309912
ymfat = 5.035063

BONE REGION
bone region not identified

BLOOD REGION
blood not identified

---------------
INTERACTION WITH muscle.c
---------------

# of analysis ? 3

analysis # 1
input starting depth = 0.5
input ending depth = 1.0

fit parameters: alpha = 3.360128 & gamma = 0.700289
ym = 0.986114
error in fit = 1.477677
error rate = 0.044778

----------

analysis # 2
input starting depth = 1.0
input ending depth = 1.5

fit parameters: alpha = 6.541478 & gamma = 1.381751
ym = 0.253292
error in fit = 2.258990
error rate = 0.066441

----------

analysis # 3
input starting depth = 0.5
input ending depth = 1.5

fit parameters: alpha = 4.452850 & gamma = 1.076018
ym = 0.417679
error in fit = 6.120311
error rate = 0.091348

---------------
INTERACTION WITH graph.c
---------------

input starting depth = 0.5
input ending depth = 1.5

alpha = 4.452850
gamma = 1.076018
F SIMULATION PROGRAM

#include <stdio.h>

main()

FILE *fopen(), *f1, *f2, *f3, *f4, *f5;

int i, count, flag1, flag2, i1, i2;
float depth[300], disp[300], d1, d2;
float odispl[300], odisp2[300], odisp3[300];

f1 = fopen("depth","r");
f2 = fopen("disp","r");
f3 = fopen("odispl","w");
f4 = fopen("odisp2","w");
f5 = fopen("odisp3","w");

i = 0;
while(fscanf(f1,"%f",&depth[i]) != EOF) i++;
i = 0;
while(fscanf(f2,"%f",&disp[i]) != EOF) i++;
count = 1;
printf("PRESSURE SIMULATION\n");
/*--------FIRST SIMULATION---------------------*/
printf("input starting depth = ");
scanf("%f",&d1);printf("\n");
printf("input ending depth = ");
scanf("%f",&d2);printf("\n");
flag1 = 0;
flag2 = 0;
for(i=0;i<count;i++)

if((d1<depth[i])&&(flag1 == 0))

i1 = i;
flag1 = 1;

if((d2<depth[i])&&(flag2 == 0))

i2 = i;
flag2 = 1;
for(i=0;i<i1;i++) odispl[i] = disp[i];
for(i=i1;i<i2;i++) odispl[i] = disp[i]+0.6*(disp[i1]-disp[i]);
for(i=0;i<count;i++)

printf(f3,"%f\n",odispl[i]);
/*--------SECOND SIMULATION---------------------*/
printf("input starting depth = ");
scanf("%f",&d1);printf("\n");
printf("input ending depth = ");
scanf("%f",&d2);printf("\n");
flag1 = 0;
flag2 = 0;
for(i=0;i<count;i++)

if((d1<depth[i])&&(flag1 == 0))

i1 = i;
flag1 = 1;

if((d2<depth[i])&&(flag2 == 0))

i2 = i;
flag2 = 1;
for(i=0;i<i1;i++) odisp2[i] = disp[i];
for(i=i1;i<i2;i++) odisp2[i] = disp[i]+0.6*(disp[i1]-disp[i]);
for(i=0;i<count;i++)

fprintf(f4,"%f\n",odisp2[i]);
/*--------THIRD SIMULATION---------------------*/
printf("input starting depth = ");
scanf("%f", &dl); printf("\n");
printf("input ending depth = ");
scanf("%f", &d2); printf("\n");
flag1 = 0;
flag2 = 0;
for (i=0; i<count; i++) {
    if ((dl<depth[i]) && (flag1 == 0)) {
        il = i;
        flag1 = 1;
    }
    if ((d2<depth[i]) && (flag2 == 0)) {
        i2 = i;
        flag2 = 1;
    }
}
for (i=0; i<=il; i++) odisp3[i] = disp[i];
for (i=il+1; i<count; i++) odisp3[i] = disp[i];
for (i=il+1; i<i2; i++)
    odisp3[i] = disp[i] + 0.6 * (disp[il] - disp[i]);
for (i=il; i<count; i++)
    fprintf(f5, "%f\n", odisp3[i]);