PULSE-ECHO ULTRASONIC IMAGING METHOD FOR ELIMINATING SAMPLE THICKNESS VARIATION EFFECTS

Inventor: Don J. Roth, Lakewood, Ohio

Assignee: The United States of America as represented by the Administrator of the National Aeronautics and Space Administration, Washington, D.C.

Abstract

A pulse-echo, immersion method for ultrasonic evaluation of a material which accounts for and eliminates nonlevelness in the equipment set-up and sample thickness variation effects employs a single transducer and automatic scanning and digital imaging to obtain an image of a property of the material, such as pore fraction. The nonlevelness and thickness variation effects are accounted for by pre-scan adjustments of the time window to insure that the echoes received at each scan point are gated in the center of the window. This information is input into the scan file so that, during the automatic scanning for the material evaluation, each received echo is centered in its time window. A cross-correlation function calculates the velocity at each scan point, which is then proportionalized to a color or grey scale and displayed on a video screen.

15 Claims, 5 Drawing Sheets

(1 of 5 Drawings in Color)
PULSE-ECHO ULTRASONIC IMAGING METHOD FOR ELIMINATING SAMPLE THICKNESS VARIATION EFFECTS

ORIGIN OF THE INVENTION

The invention described herein was made by an employee of the United States Government and may be manufactured and used by or for the Government for governmental purposes without payment of any royalties thereon or therefor.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to ultrasonic evaluation of material properties. More particularly, the invention relates to non-destructive ultrasonic evaluation of materials by measuring velocity using a single transducer pulse-echo immersion system, automatic scanning and digital imaging, which provides a video image of the sample in color or grey scale which is a map of a material property such as porosity fraction.

2. Background Of the Disclosure

Nondestructive evaluation applicable to evaluating properties of materials such as ceramics, metals, plastics and various composites are known to those skilled in the art and include x-radiography, ultrasound or ultrasonic evaluation, and thermal methods. These methods provide an efficient, quasi-quantitative measure of material homogeneity, but often lack the precision necessary for microstructure evaluation of high-performance materials, such as high temperature oxidation resistant ceramics and the like. The development and use of materials for high-performance applications requires detailed, quantitative knowledge of microstructural and compositional variability for defining acceptable levels of variability and for rejecting those materials and processes that yield sample-to-sample and within-sample variations likely to result in unacceptable property (e.g., strength, thickness variation from one edge to another) likely to result in unacceptable property (e.g., strength, thickness variation from one edge to another). Microstructure-property relations have been previously established.

Repeated, uniformly spaced ultrasonic contact measurements have been successful for quantifying and mapping porosity and inhomogeneity in various ceramics (e.g., SiC, Al₂O₃, YBa₂Cu₃O₇ and Si₃N₄) and metals in terms of ultrasonic material properties such as reflection coefficient, velocity and attenuation coefficient as mentioned, for example, by Roth, et. al. in Quantitative Mapping of Pore Fraction Variations in Silicon Nitride Using an Ultrasonic Contact Scan Technique, NASA TP 3377 (1993). This publication describes quantitatively characterizing material (e.g., Si₃N₄) microstructure in terms of actual ultrasonic wave parameters. The wave parameters include reflection coefficient, attenuation coefficient and velocity. A post-scan interactive data display system is used for comparing ultrasonic properties at different locations within samples and viewing the resultant ultrasonic images. Further refinement of this process is disclosed by Roth, et. al. in PSIDD: A Post-Scan to Interactive Data Display System for Ultrasonic Scans, NASA TM-4545 (1993). This process relates to contact scans and does not disclose how to account for thickness variations in the sample being measured. Piche discloses a single transducer immersion method for evaluating plastic using a technique in which 16 scan points are pulsed for the sample and the results evaluated using regression analysis [L. Piche, Ultrasonic Velocity Measurement for the Determination of Density in Polyethylene, Polymer Eng. & Sci. v. 24, n.17, p. 1358–58 (Dec. 1984)]. This method does not relate to forming an image of the sample property, nor does it provide an experimental technique that automatically accounts for nonlinearity and thickness variation during a scan procedure required to form an image. Consequently, a need still exists for a method which will permit ultrasonic material evaluation that will account for nonlinearity and thickness variations in the material, require only a single transducer, eliminate problems associated with physical contact between the transducer and sample or buffer rod, and display, on a video screen in grey scale or color, an image of the scanned material which is a map of an internal structural property of the material, such as porosity fraction.

SUMMARY OF THE INVENTION

The invention relates to a method for nondestructive ultrasonic evaluation of materials by measuring velocity using a pulse-echo immersion system with automatic scanning, echo cross-correlation and digital imaging to obtain a grey scale or color image of the sample. The velocity values obtained for each scan point are scaled on a grey or color scale and displayed on a video screen which shows a material property such as porosity fraction. After the automatic scanning, nonlinearity in the set-up and sample thickness variation effects are accounted for and eliminated by insuring that the echoes at each scan point are first gated and input into a scan parameter file in a computer, so that during the subsequent automatic scanning each received echo is centered in the time window set for it. While it is possible, but not practical to do a manual prescan at each and every scan point needed for a two dimensional video image of the material property being evaluated, many sample thickness variations are in the form of a uniform thickness variation from one edge to another. In this case of uniform thickness variations from one edge to another, preliminary scans are performed along a single line in both the x- and y-directions of the sample to provide slant correction factors. The slant correction factors are input into the scan parameter file so that any wedge shape variations are taken into account during the automatic scanning for the material evaluation, to insure that each echo received during the automatic scanning is centered in the time window. A single transducer is used in a preferred embodiment of the invention.

In the immersion method of the invention, the material to be evaluated is surrounded by a liquid and positioned over an acoustic reflector which is also immersed in the liquid. An ultrasonic wave of a known frequency is transmitted through the liquid and four separate echoes are recorded and evaluated at each scan point. Each echo is received as an analog waveform which is digitized and stored in a computer. The echoes received, digitized and stored during the sample evaluation scans are the first two successive echoes reflected off the back surface of the sample, the first echo reflected off the front surface of the acoustic reflector on which the received wave has passed through the sample, and the forth is the first echo reflected off the front surface of the reflector to with the sample not present, so that the received wave does not pass through the sample. This means that at least two separate scans must be made, with and without the sample present between the transducer and reflector. However, as a practical matter it is difficult from both a hardware and software perspective to accomplish this in just two scans and obtain maximum time resolution and thus maximum accuracy. Consequently three or four separate scans are
performed, with three being faster and four being more accurate. The choice is left to the discretion of the practitioner. In the embodiment in which four separate scans are performed during the sample evaluation, the ultrasonic wave goes through both the liquid and the material during the first three scans. For the forth scan the material sample is removed so that the transmitted wave is reflected off the front surface of the reflector without going through the material. Although the order is not important, it is convenient to receive the first echo reflected off the back surface of the material during the first scan and the second successive echo reflected off the back surface of the sample during the second scan. During the third scan in which the transmitted wave goes through both the immersion liquid and the material sample, the first echo reflected off the front surface of the reflector is received. The first echo reflected off the front reflector surface is received during the fourth scan during which the sample is evaluated, nonlevelness and thickness variations are taken into account during the automatic scanning for the material evaluation.

BRIEF DESCRIPTION OF THE DRAWINGS

The file of this patent contains at least one drawing executed in color. Copies of this patent with color drawings (s) will be provided by the Patent and Trademark Office upon request and payment of the necessary fee.

FIG. 1 schematically illustrates the spatial relationship between the transducer, liquid, material sample, reflector plate and the transmitted and reflected ultrasonic echo waves in the practice of the invention.

FIGS. 2(a) and 2(b) graphically illustrate the amplitude and time delay of the received analog ultrasonic echoes reflected off the material sample and reflector.

FIGS. 3(a) and 3(b) graphically illustrate respective first and second sample back surface echoes in which the second is inverted with respect to the first.

FIG. 4 is a block diagram schematically illustrating the instrumentation used in the invention.

FIGS. 5(a) and 5(b) are photographs of video grey scale displays of a thickness based ultrasonic velocity image of a ceramic according to the prior art method and the method of the invention, respectively.

DETAILED DESCRIPTION

Referring to FIG. 1, transducer 10 is schematically shown as partially immersed in a liquid 12 which is the immersion fluid. Material sample 14 is shown positioned in the fluid 12 between the transducer 10 and reflector plate 16 in container or tank 18 located on top of x- and y-direction motorized stages 19 and 20. Supports 15 and 15' maintain sample 14 above reflector 16. The transducer 10 and motorized x- and y-direction stages 19 and 20 are electrically connected by means not shown to a pulser-receiver (not shown) and to means (not shown) for moving the x- and y- stages in their respective directions. Similarly, liquid 12 is connected to means not shown for maintaining the temperature of the liquid preferably within at least about ± 1° F. of the temperature at which the scans are to be run. It is possible to maintain the liquid temperature within ±0.1° F. The better the temperature control, the more accurate the results will be. For example, if the temperature of the immersion liquid is ±1° F. and the liquid is water, a 1.5% error in velocity is possible. If the porosity fraction or other property of the material at a particular point is such as to result in a velocity difference in the sample of 2%, only a 0.5% micro-structural velocity difference might be detected if a 1° F. temperature variation is present during the scan. At each scan point an ultrasonic wave 22 of a known frequency is transmitted from transducer 10 through liquid 12 and into material sample 14. Entering material 14 causes part of wave 22 to be reflected (not shown) off the top surface of the material, with the rest of the wave passing through the material as 23. Part of wave 23 continues through the material and to the top surface of acoustic reflector 16 as 24, is reflected back off the top surface of reflector 16 as 25, and continues back through the sample as 26 and 27. Echoes 28 and 29 are recorded by the receiver. The velocity of the transmitted wave in the sample is determined by measuring the time delay between the first two successive sample back surface echoes in which the second is inverted with respect to the first.
passes back through the sample and returns to the transducer as wave 26. A portion of wave 23 is reflected back to the top surface of the sample and returns to the transducer as wave 27. Part of the wave 23 reflected off the back surface of the material is reflected off the top surface, returns to the back surface, is again reflected back to the top surface and exits as wave 28. Waves 27 and 28 are the first two successive back surface reflected waves used in the method of the invention at each scan point. Not shown is the wave transmitted through the liquid and reflected back to the transducer without going through the material. This wave which is not shown and wave 26 are the two reflector front surface echoes used in the method of the invention. Motorized stages 19 and 20 form part of an automated scanning system which incrementally moves in both the x- and y-directions to obtain an ordered array of points across the entire surface of the material sample. A 20 MHz broadband transducer was used in the practice of the invention. Broadband transducers emit a broadband frequency content dominated by a center frequency. That is, they are made to emit at a nominal frequency proximate that of the design frequency (e.g., 20 MHz), with a Gaussian fall-off on either side of the nominal center frequency. Thus, a 20 MHz broadband transducer will also emit frequencies slightly above and below the nominal center frequency of 20 MHz. In the Pichi article referred to above, although the two different reflector front surface echoes are captured and recorded, the first front surface echo and the first back surface echo are captured and recorded. This is different from the method of invention which captures and records the first two successive sample back surface echoes and not the first front surface echo. Further, Pichi does not use automatic sample scanning or digital imaging. FIGS. 2(a) and 2(b) graphically illustrate the reflected waveforms received and displayed on the CRT of an oscilloscope as time domain analog waveforms. Turning to FIGS. 2(a) and 2(b), the intensity or strength of the received waveform is displayed as voltage amplitude, which is the ordinate of the graph, and the received time delay as the abscissa. In this representation, B1, B2 and M' refer to waves 27, 28 and 26 of FIG. 1, respectively, with M' representing the wave transmitted through the liquid and reflected off the front surface of the reflector without passing through the material sample. The time delay between the first two successive echoes reflected from the back or bottom surface of the material back to the transducer, B1 and B2, is readily obtained, as is the time delay between the two reflections received from the front surface of the reflector, M' and M. Since the velocity of the ultrasonic wave is faster in denser media than in less dense media, voids, delaminations, porosity and other density variables within the material are obtained as a function of the speed of the wave, which is determined by the time delay between the first two successive echoes received which have been reflected off the back of the material, and the time delay between the two different reflections from the front surface of the reflector. As set forth above under SUMMARY, the speed or velocity of the transmitted wave traveling through the material sample is determined according to the simple equation:

\[ v = c \left( \frac{\Delta t}{2\ell} + 1 \right) \]

wherein \( \Delta t \) is the received time delay between the two successive material sample back surface echoes, wherein \( \ell \) is the time delay between the two different echoes received from the reflector with and without going through said sample, and wherein \( c \) is the speed of the transmitted wave in the liquid. This equation is accurate for a single point measurement. Prior art velocimetry techniques such as that of Roth et al. in the quantitative mapping publication referred to above, assume that the material sample is of uniform thickness and do not take into account nonlevelness and material thickness variations as does the method of the invention.

In the practice of the method of the invention, tank 18 may be made of any suitable material. Clear plastic such as polymethylmethacrylate (e.g., Lucite or Plexiglass) has been found useful. The sample tank contains a suitable elastic liquid, such as water, as the immersion fluid to provide an acoustic coupling between the transducer, material and reflector plate. Since the x-, y-direction scans made across the sample surface in the method of the invention can take a significant amount of time compared to that for a single point measurement and since the speed of sound in a liquid is also a function of temperature, the water is maintained at a constant temperature during the scanning. This is readily accomplished simply by using a constant temperature regulating means, such as a constant temperature water circulator, for maintaining the desired temperature constant during the ultrasonic scanning. It is convenient to keep the temperature of the water at about ambient or 68° F. ±1° F. during the scan, although other temperatures may be used if desired, as long as the temperature is maintained within no more than ±1° F. In the case of distilled, deionized water, the wave velocity may be obtained from published tables. However, tap water may be used as long as the velocity in the water is actually measured. The reflector is placed on the bottom of the tank. Other immersion liquids may be used, if desired, such as Dow Corning 704 vacuum pump oil.

The reflector is a solid plate of material having an acoustic impedance significantly different from that of the liquid or water. A flat plate of tungsten (e.g., 1/8"-1/4" thick) is preferably used, because tungsten has an acoustic impedance almost two orders of magnitude higher than water in units of g/cm²-sec. The use of a tungsten plate results in the highest possible reflection amplitude of any solid material for the echoes reflecting off the front surface of the reflector plate. This large difference in acoustic impedance is important when attempting to obtain ultrasonic echoes that have to travel into and through immersion liquid and the sample, bounce off the reflector plate, and travel back through the liquid and sample to the transducer for reception. High frequency ultrasound provides greater time resolution than lower frequencies and is therefore more desirable for greater accuracy of the velocity of the ultrasound through the sample and corresponding velocity image. The higher the frequency, the greater the velocity accuracy. By having the highest reflection amplitude possible, it is possible to use the method of the invention (a) at higher frequencies where attenuation through the sample is greater than if using lower frequencies and (b) with materials that significantly attenuate ultrasound, such as composite materials. By high frequency ultrasound is meant from 1-100 MHz, typically 3-50 MHz and more typically 10-30 MHz.

The material sample is easily positioned over the reflector plate by using spacers on top of the plate and placing the material on top of the spacers. It is important that the spacers have the same height or thickness so that the material is as level as possible. Lucite is available as sheets which are very uniformly thick and it is convenient to use 0.5" cubes of this plastic as spacers. The material sample, such as a plate of silicon nitride ceramic, is placed on the plastic spacers over the tungsten reflector plate prior to scanning.

FIG. 4 schematically illustrates, in block diagram fashion, the basic system and instrumentation used for the scanning.
and ultrasonic imaging according to an embodiment used in the practice of the invention.

Thus, referring to FIG. 4, the basic instrumentation includes a transducer 59, a pulser-receiver 52, and a programmable waveform digitizer 54 having associated with it a vertical voltage amplifier 56, programmable time base 58, and analog and digital monitors 60 and 62, respectively. Also included are a time delay or synthesizer 64, an image processor 66, an X, Y, Z controller 68, computer or central processing unit (CPU) 70, and video display 72. In the embodiment shown, the computer 70 with terminal 74 and associated video display 76 also forming part of the system. Monitors 60 and 62, along with digitizer 54, voltage amplifier 56 and time base 58 also serve as respective analog and digital oscilloscopes. The time synthesizer, time base, voltage amplifier and waveform digitizer are all general purpose interfaces (IEEE-488/us [hereinafter "GPIOB") programmable and interconnected via GPIOB cables. The computer 70 is programmed in Fortran and contains an image processor system which is connected to the video color and gray scale display 72. The computer controls the GPIOB instrumentation and acquiring of the desired waveforms via the GPIOB. The process includes data acquisition, analysis/calculation, image processing and display. The Fortran software used in the practice of the invention includes the scanning program, the analysis and cross-correlation program, the gray scale image maker program, and the display program are contained in the attached Appendix. The waveform digitizer is a Tektronix 7912 AD Programmable Digitizer along with a Tektronix 7A16 P Programmable Vertical Voltage Amplifier (voltage base) and a Tektronix 7B90 P Programmable Time Base. The time delay for the received echoes is determined a priori and input to the waveform digitizer. The slant correction factors are input into the computer, and a Mitsubishi 20LP is the video display and pixel sweep. The X and Y positional and Z intensity outputs from the waveform digitizer are connected to the analog and digital monitors 60 and 62. The analog monitor is used for the prescans and the digitizer for the automatic scanning.

As set forth under the SUMMARY, prior to the two, three or four scans during which an operator notes the time delay of each echo received at each scan point so that it is no longer centered within the oscilloscope time window, if a received echo is not centered within the time window on the scope, this is noted and the time window changed for each such echo received until the received echo time domain waveform is complete within the new time window set for it to insure that the complete time domain waveform is captured or gated completely within the new window. This time delay information at each scan point is inputted into the scan parameter file and recalled during the actual scanning during the material evaluation, to automatically adjust the time delay for the received echoes at each scan point so that each echo received during the scanning is centered within the time window set for it. This is very time consuming to do for each scan point. However, having a thickness variation in the form of a uniform thickness variation from one edge to another, preliminary scans are performed along a single line in both the x- and y-directions of the sample to provide slant correction factors. The slant correction factors are input into the computer scan parameter file so these variations are taken into account during the automatic scanning for the material evaluation. It is important that the echo at each scan point is centered in its time window, because the whole pulse or echo time domain waveform is needed to give the precise time delay between echoes for the cross-correlation which provides the velocity value. In doing this for a wedge shaped sample, prior to the automatic scan, the transducer scans along two straight lines over the sample, once in the x-direction and once in the y-direction, in which the sample is evaluated, nonlevelessness and sample thickness variations are accounted for and eliminated by pre-scans to insure that the received reflections or echoes are within their set time windows to provide a complete waveform for evaluation and cross-correlation to accurately obtain the time delay data used in calculating the velocity values. That is, during the nonleveness and material thickness variation scans, the operator notes if the time delay of each echo received at each scan point is such that it is no longer centered within the oscilloscope time window, if a received echo is not centered within the time window on the scope, this is noted and the time window changed.
oscilloscope time window set for it so that the received waveform is displayed with the maximum possible amplitude on the CRT and still have the complete waveform. This permits the maximum time resolution of individual echoes without losing any part of the time domain waveform which appears on the CRT screen as a function of voltage (amplitude) and time, wherein time is the x-axis and voltage or amplitude is the y-axis. It is important and forms an aspect of the invention that the complete waveform or pulse echo be captured or "gated" on the CRT screen in order to perform an accurate cross-correlation later on in the procedure of the process of the invention. The cross-correlation of echoes provides the precise time delay between received echoes or pulses which is required to calculate the velocity or speed of the ultrasound in the material evaluated which, in turn, provides the information to gray or color scale the velocity data into a digitized map of the material density. This slant correction procedure also allows an accurate evaluation to be made without the need for specialized leveling equipment. These x- and y-direction time window corrections are called slant correction factors and they are inputted into the scan parameter file in units of "nsec/μm" where (a) the number of nsec is the time extent from sample end-to-end that is required to keep the specific echo centered and is determined using the (a) time synthesizer to reposition echoes in time and (b) the number of μm is the distance traveled by the transducer for which this slant factor is determined. By way of an illustrative, but nonlimiting example, the first scan point (0.0) along the x-direction may have a B1 echo centered at a time=6.77 μsec, while the last scan point (40.0), may have B1 centered at time=7.14 μsec. If the x-direction scan line length is 40 mm, the x-direction slant correction factor is obtained from (7.14–6.77)/40 μsec/ mm. It should be noted that slant correction factors can be negative as well as positive numbers. The location of the time window during scanning for the material evaluation is automatically adjusted via computer control by using the formula:

\[ W_{DT}=T_P\left[(X_{SC}(X_{SN})(X_{SN})(Y_{SN})(Y_{SN})]\right] \]

wherein \( W_{DT} \) is the correct delay time window at a particular scan location, \( T_P \) is the time delay at the initial scan location, \( X_{SC} \) and \( Y_{SC} \) are the x- and y-direction slant correction factors, \( X_{SN} \) and \( Y_{SN} \) are the scan point numbers in the x- and y-directions, and \( X_{SN} \) and \( Y_{SN} \) are the x- and y-direction scan increments. With many samples it has been found that the slant correction factors turn out to be the same for the B1 B2 echoes and the slant correction factors for the M' M* echoes are the same. However, for some samples (e.g., thick samples), they may not be the same. In such cases a first x- and y-direction scan is made for the B1 echoes and a second x- and y-direction scan made for the B2 echoes. The same holds for the M' and M* echoes for which two separate scans are made in the x- and y-directions.

A scan parameter file is input into a computer which includes all of the information necessary to automatically scan the material sample being evaluated. This information includes a predefined and ordered array of scan points over which to run the scan. By way of an illustrative, but nonlimiting example, in an example of the method of the invention in which the material being evaluated was a monolithic ceramic wedge, the scan consisted of a 41 (X-direction) by 81 (Y-direction) grid of measurements for a total of 3,200 scan points, with each measurement or scan point separated by 1 mm (x-) and y- scan increment). Information input into the scan parameter file (NOTHICK_ALLSHAPE1.DAT) includes the following:

---

C **TITLE** NOTHICK ALLSHAPE1.DAT
C **SCAN INCREMENT (μm) IN X-DIRECTION IS: 1000.
C **SCAN INCREMENT (μm) IN Y-DIRECTION IS:
40000.
C **SCAN LENGTH (μm) IN X-DIRECTION IS:
C **SCAN LENGTH (μm) IN Y-DIRECTION IS:
C **X-DIRECTION SLANT CORRECTION FACTOR (nsec/μm) FOR B1 & B2 ECHOES IS:
-0.0055
C **Y-DIRECTION SLANT CORRECTION FACTOR (nsec/μm) FOR REFLECTOR ECHOES IS:
-0.0055
C **Y-DIRECTION SLANT CORRECTION FACTOR (nsec/μm) FOR B1 & B2 ECHOES IS:
-0.00075
C **Y-DIRECTION SLANT CORRECTION FACTOR (nsec/μm) FOR REFLECTOR ECHOES IS
0.0
C **TIME LOCATION (μSEC) OF B1 ECHO AT SCAN ORIGIN IS:
52.83
C **TIME LOCATION (μSEC) OF B2 ECHO AT SCAN ORIGIN IS:
52.31
C **TIME LOCATION (μSEC) OF REFLECTOR ECHO W/SAMPLE PRESENT AT SCAN LOCATION IS:
69.46
C **TIME LOCATION (μSEC) OF REFLECTOR ECHO W/O SAMPLE PRESENT AT SCAN LOCATION IS:
72.48
C **IMMERSION FLUID VELOCITY (cm/μSEC) IS:
0.348
C **B2 PHASE-INVERTED WRT B1 (Y/N):N
C **M' PHASE INVERTED WRT M (Y/N):N
---

As set forth above, in the method of the invention, the transducer is activated so that the first front surface echo off the sample, the first two successive back surface echoes, and the first echo off the reflector plate are all seen in the oscilloscope display at the same time by adjusting the time base to the appropriate time per division setting and
adjusting the time synthesizer delay time. Viewing the first front surface echo off the sample enables the operator to know if the back surface echoes are also on the CRT screen. The unfocused transducer is positioned above the sample at a distance determined initially by the natural focal distance. When using an unfocused transducer a good initial starting height is approximately one to two inches above the sample. The reflector plate front surface echo may be low in amplitude compared to the sample back surface echoes, so that the pulser-receiver gain/attenuation or vertical amplifier gain settings may have to be increased to see this echo. It is important not to confuse the echoes off the front surface of the reflector plate with the second set of echoes originating from the front and back surfaces of the sample. The second set of echoes originating from the front and back surfaces of the sample will always occur at twice the delay time where the first set of these echoes appears. For example, if the first set of echoes begins at 50 msec on the digital oscilloscope, the second set will begin at 100 msec. If using, for example, a three millimeter thick sample placed on 0.5" thick plastic supports on the reflector plate, the first reflector echo will occur at about 20 msec after the time where the first set of echoes originates and thus the reflector echo will be seen at about 70 msec in this illustration. Another way to note the reflector plate echo is to raise and lower the sample while noting the location of the stationary echo corresponding to the stationary reflector plate. It is essential to have reflector plate echoes that will not interfere with the second set of echoes originating from the front and back sample surfaces. Attention is next focused on the first back surface echo from the sample, B1. The echo is centered in the oscilloscope time window to obtaining maximum time resolution by adjusting the time base time per division and the time synthesizer delay. The synthesizer time is recorded and inputted into the scan parameter computer file. This procedure is repeated for the second back surface sample echo, B2, the first front surface echo off the reflector plate with the sample present, M' and the first echo off the reflector plate with the sample removed, M". The next step is to account for and eliminate the nonlevhess elimination procedure is two, but the time per division setting for only two scans cannot be obtained in this case as the time per division setting would be fixed for all three echoes obtained in the first of the scans using this scan procedure. The following is an algorithm of a scanning program which accounts for and eliminates the nonlevhess of the set-up and uniform thickness variation effects of the sample in the resulting ultrasonic image displayed on the video, the code for which is included in the Appendix.

1) Determine the scan lengths and scan increments in the x- and y-directions, time positions of echoes at scan origin, slant correction factors, and immersion fluid velocity.

2) Edit NOTHICK_ALLSHAPEDAT FILE, which is the scan parameter file, and input information from 1) above.

3) Start scanner fortran program on computer which automatically does the following:

A) Initialize all GPIB instrumentation, which includes the time synthesizer, digitizer, time base, voltage amplifier, Klinger X, Y stages.

B) Perform scan to digitize B1 echoes and store in file

1) Digitize B1 at scan origin

2) Digitize voltage base for echo with maximum amplitude in video/oscilloscope window

3) Move Klinger tables under transducer in x-direction specified x-direction increment

4) Time synthesizer moves to delay time position

5) This results in echo in video being centered in the Tektronics analog video/oscilloscope display and subsequently digitized and stored.

\[ T_0 = \text{correct delay time window at a particular scan location} \]

\[ \text{Sxy} = \text{x-direction slant correction factor} \]

\[ \text{Syz} = \text{y-direction slant correction factor} \]

\[ \text{Nxy} = \text{scan point number in x-direction} \]

\[ \text{Nyz} = \text{scan point number in y-direction} \]

\[ \text{Lxy} = \text{y-direction scan increment} \]

\[ \text{Lyz} = \text{y-direction scan increment} \]

II) Repeat 1) until one scan line in x-direction is completed.

III) Increment transducer in y-direction specified y-direction increment and repeat I)-II).

IV) Repeat I)-III) until y-scan length is traversed and scan is completed.

V) Return Klinger tables to scan origin.

C) Perform scan to digitize B2 echoes and store in file by repeating steps B(I-V)

D) Perform scan to digitize reflector echoes with sample present and store in file by repeating steps B(I-V), but using reflector echo slant correction factor

E) Remove sample. Perform scan to digitize reflector echoes without sample present and store in file by repeating steps B(I-V), but using reflector echo slant correction factor

4) Start velocity calculation Fortran program on computer to produce a file of velocities at each scan location by performing the cross-correlation algorithm.

Time position = \( T_0 + (S_{xy}(N_{xy} - 1) + S_{yz}(N_{yz} - 1)) \)
5 Start image formation Fortran program on computer which results in a file of values between 0 and 255 which scale directly with the velocity values.
6 Start image display program which brings grey scale level image up on video.

Before initiating the scanning procedure, the temperature of the water or other immersion fluid is measured. If the fluid is water, published tables or graphs of temperature and velocity can be used to determine the velocity of the ultrasound in the constant temperature water bath. If the immersion liquid is a liquid other than water, or if a more precise temperature than that available in published graphs and tables is desired, the velocity of the ultrasound in the liquid is determined by recording the times (Tf) where ultrasonic peaks occur for two different vertical positions (Z1 and Z2) of the reflector plate above the reflector plate. The velocity, V, is then determined from

\[ V = \frac{Z1 - Z2}{Tf1 - Tf2} \]

The phase relationships of (a) B1 compared to B2 and also (2) the reflector front surface echo with the sample present (M') compared to that without the sample present (M") are examined. These phase relationships are important for the computation of the velocity image of the scanned sample. The quantity 2t is obtained by cross-correlating echoes B1 and B2 which is defined as the precise time delay between the B1 and B2 echoes. If B1 and B2 are phase inverted with respect to each other, the time occurrence of the minimum in the cross-correlation function is used to obtain 2t. If M' and M" are phase inverted with respect to each other, the time occurrence of the minimum in the cross-correlation function is used to obtain At. Otherwise, at the time occurrence of the maximum in the cross-correlation function is used.

FIGS. 3(a) and 3(b) graphically illustrate the case in which B2 is phase inverted with respect to B1. The same holds for M' and M". Phase relationships generally remain the same throughout the scan, unless significant discrete microstructural defects are encountered by the ultrasonic wave. After the scan has been completed and all the received echoes have been digitized and stored in the scan data file in the computer, they are recalled from the data file to perform the velocity image calculation for each scan location. In performing this cross-correlation, an overlap method is used by the computer based on a cross-correlation program using Fast Fourier transforms published in pages 415 and 416 (Correlation and Autocorrelation Using the FFT) in the book Numerical Recipes-The Art of Scientific Computing, by Press, et. al., 1986 Edition, Cambridge Univ. Press.. The Fortran program used is in the Appendix. Echoes M' and M" are also cross-correlated to obtain At where where M' is the echo reflected off the reflector plate front surface with the sample present, M" is the echo reflected off the reflector plate front surface without the sample present, and At is the time delay between them. If M' and M" are phase inverted with respect to each other, the time occurrence of the minimum in the cross-correlation function is used to obtain At. Otherwise the time occurrence of the maximum in the cross-correlation function is used. The velocity, V, at each scan location is then calculated from the equation referred to above. The velocity value for each scan location is sequentially stored in the computer. After the scan is completed the velocity values are scaled on a gray or color scale with a value directly proportional to the velocity values, with the highest and lowest scale values corresponding to the highest and lowest velocity values.

The invention will be further understood with reference to the example below.

EXAMPLE

In this example, a sample of silicon nitride ceramic was evaluated using the thickness based velocity image method disclosed in the NASA Technical Memorandum TP 3377 referred to under Background. This method is based on a velocity, cross-correlation ultrasonic imaging method without the pre-scan to account for and eliminate nonlevelness in the set-up and sample thickness variations. In this method, only the first two sample back surface echoes are captured and evaluated. The silicon nitride ceramic was 3.5 mm thick with a uniform 300 micron thickness gradient. Very coarse time scaling was used so that the B1 and B2 echoes stayed in the time window while the sample thickness changed as the scan proceeded.

The same silicon nitride ceramic sample was also scanned and velocity imaged on a grey scale according to the method of the invention which included the prescans to eliminate set-up and sample thickness variations and which also captured and cross-correlated both the first two successive sample back surface echoes and the two different reflector front surface echoes.

In both cases, the 20 MHz broad band transducer was used, the immersion liquid was water and the back plate was tungsten as set forth above.

FIGS. 5(a) and 5(b) are photographs of video grey scale displays of the thickness based ultrasonic velocity image of a ceramic according to the prior art method, and an image according to the method of the invention which included the prescans, respectively. Referring to FIG. 5(a), it is seen that the top defect is masked due to that part of the sample being thicker than the bottom part. Also, the defect near the bottom is not too discernable and the lower portion is very light due to it being thinner. In marked contrast and as shown in FIG. 5(b), the method of the invention clearly and correctly illustrates the defect areas, including resolution of the upper defect and an overall porosity gradient in the sample. It is believed that this demonstrates the efficacy and improvement to the art of the invention.

It is understood that various other embodiments and modifications in the practice of the invention will be apparent to, and can be readily made by, those skilled in the art without departing from the scope and spirit of the invention described above. Accordingly, it is not intended that the scope of the claims appended hereto be limited to the exact description set forth above, but rather that the claims be construed as encompassing all of the features of patentable novelty which reside in the present invention, including all the features and embodiments which would be treated as equivalents thereof by those skilled in the art to which the invention pertains.
notthick_SCAN.FOR --- 4 SCAN METHOD

Scan and data acquisition for the modified ultrasonic Immersion scanner

by Don J.Roth

PLEASE USE CLNEW.COM TO COMPILE & LINK

Latest Update 20-sep-94

**integer** SCANNSTP,INDXSTEP
**integer** SCANAXIS, INDXAXIS, ZSTEP
**INTEGER** 117
character VOLANS*1,TSCHME*1,ULOLANS*1
CHARACTER 1,LOOPANS*1,SHAPEANS*1,ZIGANS*1,FLAG*1
character 1,BUF*S,SCHEME*1,FILNAME*34,SCANMODE*1
real DELAY(0:4),VOLTSET(0:4),MAXVALF,MAXVALT,VOL(4)
real slant1,slant2,slant3,slant4
**integer** NSCAN, NINDX,PLACE,ZIGCOUNT,XXX,SCANKNT
**integer** A(512),NAVE,CHECKP,WWFLAG
**INTEGER** 115,116,126,127,MAXFSIP,MAXFSIPT
**INTEGER** 136,137,146,147
**INTEGER** 2, ZIGPATHU_11, ZIGPATHU_21, ZIGPATHU_31, ZIGPATHU_41
BYTE WKRK(80)
common /BLK_/ WKRK, BUFFER
common /BLK_/ A, DELAY, VOLTSET, MAI
data CHECKP/0/ZIGCOUNT/1/
data 16/1/11/16/1/11/1/PLACE/0,
data 126/1/127/136/1/137/1/146/1/147/1/

**format** (A )
**format** (1 )
**format** (1 )
OPEN(unit=8, file="TXAl", status="NEW")
**CALL** ENTERPARAM( SCANNSTP, INDXSTEP, NSCAN, NINDX, NAVE, VOLTAGE1,
VOLAGEU, VOLTAGE_MAX, SHAPEANS, LOOPANS, ZIGPATHU_11, ZIGPATHU_21,
ZIGPATHU_31, ZIGPATHU_41, SLANT1, SLANT2,
SLANT3, SLANT4)

4235 SCANAXIS = 1
INDXAXIS = 2
ZSTEP = 10

**CALL** STRTPGPIB
**DO** (1800 IADDR= 5,1,-1)
1800 **CALL** INITINSTR( IADDR )

C Setup of 7912: begin with V/D = -5, then automatically find the best intensity and Digitize Defects
C
C NOTE: MAY NEED TO CHANGE THIS V/D TO START
C
2000 **CALL** TEKGLT
C*******************************************************************************
INOTHICK - STARTING 1ST BACK SURFACE ECHO WINDOW TIME
INOTHICK - STARTING 2ND BACK SURFACE ECHO WINDOW TIME
INOTHICK - STARTING (W/SAMPLE) REFLECTOR ECHO WINDOW TIME
INOTHICK - STARTING (W/O SAMPLE) REFLECTOR ECHO WINDOW TIME

READ (10,65403) DELAY1
READ (10,65403) DELAY2
READ (10,65403) DELAY3
READ (10,65403) DELAY4

READ (10,65402) VOLANS
READ (10,65403) VOL(1)
READ (10,65403) VOL(2)
READ (10,65403) VOL(3)
READ (10,65402) VOLANS
READ (10,65403) VOL(4)

65402 FORMAT (A)
65403 FORMAT (F)
51 FORMAT (A32)
52 FORMAT (A2)
53 FORMAT (A4)
54 FORMAT (A2)

CALL INITINSTR(0)
CALL SETVOLTDIV(0.5)
CALL TEKGTL
CALL GETMAI(MA1)
CALL TEKGTL
CALL PUTTIME(DELAY(1))
CALL TEKGTL
CALL GETBESTMAI(MA1)
CALL DIGDEF

CALL INITINSTR(0)
CALL SETVOLTDIV(0.5)
CALL TEKGTL
CALL GETMAI(MA1)
CALL TEKGTL
CALL PUTTIME(DELAY(1))
CALL TEKGTL
CALL GETBESTMAI(MA1)
CALL DIGDEF

!Preliminary Digitizations to get Watch level and Timeset (Time/div)
CALL TEKGTL

CALL AUTOSETVOLTS(MA1, VOLTSET(I), NUMINT)

TYPE *=AUTOSET VOLTS*

IF(VOLTSET(I)EQ.999.)THEN
  TYPE *=DIGITIZER IS SCREWED UP
  GOTO 2000
ENDIF

C
CALL GETGRID( TIMESET, X )
CALL GETSA( NAVE, MAI, A )
CALL TEKCTL

TYPE *, 'Timeset=', TIMESET,' Delay=', DELAY(I),
+ ' Voltset=', VOLTSET(I)

**CALL GETGRID**

**CALL GETSA**

**CALL TEKCTL**

**WRITE(16,rec=116,fmt=53)TIMESET**
16=416-1

**C**

Get WATCH - standard ground level

**TYPE *, '**
**TYPE *, '**
**TYPE *, '**
**TYPE *, '**

W = A(1)/NAVE
WATCH = W - 100.

**WRITE(8,12030)WATCH**

**TYPE *, '**
**TYPE *, '**
**TYPE *, '**

12030 format( 'Minimum acceptable ground level=',F )

**C**

SCANSTEP = 1000*SCANSTEP

**INDXSTEP = 1000*INDXSTEP**

**CALL SETXYZ( SCANSTEP, INDXSTEP, ZSTEP )**

**WRITE(5,12140,format=f**

**IDIR=1**

WWFLAG=0

do 2600 noth=1,4 ! thickness mod
if(noth.LE.2)then
CALL MOVORXY( ZIGANS,SCANAXIS,NSCAN,INDXAXIS,NINDX)
REWIND(3) ! DATHS FILE
if(ZIGANS.EQ.'Y')ZIGCOUNT=1
endif

if (noth.LE.4) then
**type *, '*** Finished w/ Sample Scan **********'**
**type *, *** Remove Sample from tray & Hit <RET> to perform water scan'**
**type *, '***'**
**type *, '***'**

Read (5,10020) IO
ENDIF
DO 2400 I=1,NNDX ! Outer loop over index
DO 2200 J=I,NSCAN ! Inner loop over scan
IF (NOTH_I.ge.2) THEN
  READ(12)JI,J,FLAG,SCANKNT 'READ WHETHER 'H' OR 'S'
  TYPE *, '1ST TIME AROUND FOR JJ,II'
  TYPE *, FLAG = FLAG
  ENDIF
I = I-((SCANMODE.EQ.'P'.OR.SCANMODE.EQ.'L').AND.I.GT.1)GOTO 2400 ! MOD FOR LINE / POINT SCAN
IF (NOTH_I.EQ.I) TYPE *, 'I= ',I,' AND J= ',J
PLACE = PLACE + 1
WRITE( 8,13000 )J,I,PLACE
ENDIF
C************************************************************
3429 CONTINUE
IF (NOTH_I.EQ.1 .AND.ILT.4) THEN
  TYPE *, 'I16=',I16,' I17=',I17
ELSEIF (NOTH_I.EQ.1 .AND.I.EQ.4) THEN
  TYPE *, 'I46=',I46,' I47=',I47
ENDIF
IF (I.EQ.NSCAN ) GOTO 2200 !
IF (LOOPANS.EQ.'N') GOTO 2200 ! DON'T PICK UP XDUCER EACH TIME
IF (NOTH_I.EQ.1 ) TYPE *, 'IT=',IT
IF (I.EQ.NSCAN) GOTO 2200 !
IF (NOTH_I.EQ.1) TYPE *, *
IF (NOTH_I.EQ.1) TYPE *, 'VOLANS (WRT "ON SAMPLE")=', VOLANS
IF (NOTH_I.EQ.1) TYPE *, 'VOLANS (WRT "NOT ON SAMPLE")=', VOLANS
IF (NOTH_I.EQ.1) TYPE *, *

C************** MOD FOR ZIGZAG SCAN ********************

22001 IF (ZIGANS.EQ.'Y' AND ZIGCOUNT.EQ.0) THEN ! ZIGZAG SCAN
IF (SCANMODE.EQ.'P') GO TO 12306 ! FOR POINT MEASURE, DON'T MOVE X-AXIS
CALL MOVXYZ(SCANAXIS,-1) ! MOVE X-AXIS BACKWARD
ELSEIF (ZIGANS.EQ.'Y' AND ZIGCOUNT.EQ.1) THEN
IF (SCANMODE.EQ.'P') GO TO 12306 ! FOR POINT MEASURE, DON'T MOVE X-AXIS
CALL MOVXYZ(SCANAXIS,1) ! MOVE X-AXIS FORWARD
ENDIF

12306 CONTINUE
CALL WAIT2(1000) ! ADD DELAY SO THAT DATA IS NOT TAKEN BEFORE
C KLINER STAGES STOP MOVING
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
IF (ZIGANS.EQ.'Y') GO TO 12305

C***************

2402 CONTINUE
IF (LOOPANS.EQ.'N') GO TO 2200

12305 CONTINUE ! NOTHICK
2200 CONTINUE

! -
IF (I.EQ.NINDX) GO TO 2400 !

C FOR LINE SCAN OR POINT MEASURE, DON'T MOVE Y-AXIS

IF (SCANMODE.EQ.'P' OR SCANMODE.EQ.'I') GO TO 2401
CALL MOVXYZ(INDXAXIS, 1) ! Move
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)
CALL WAIT2(1000)

2401 NTGO = NINDX-1

IF (ZIGANS.EQ.'Y') THEN
  NORMAL SCAN
  GO TO 15021
ELSEIF (ZIGCOUNT.EQ.1) THEN
  ZIGCOUNT = 0
  GOTO 15021
ELSEIF (ZIGCOUNT.EQ.0) THEN
  ZIGCOUNT = 1
ENDIF
ENDIF

C************ CONTROL MOD FOR ZIGZAG SCAN ************
ELSEIF (ZIGANS.EQ.'Y') THEN
  ZIGCOUNT = 0
  GOTO 15021
ENDIF

C************ CONTROL MOD FOR ZIGZAG SCAN ************
15021 CONTINUE ! NOTHICK

2400 CONTINUE

2660 continue

CALL TIME(TBUF)
WRITE(*,1300)TBUF
CLOSE(8)
CLOSE(6)
CLOSE(14)
JXXX = JXXX + 2
MEASURE
WRITE(15,REC=115,FMT=52)JXXX
I15 = I15 + 1
C
STOP

SUBROUTINE TAKEDATA_O(FILENAME,VOLANS,UOOLANS,VOL,J16,J17,J26,J27,
J36,J37,J46,J47,NAVE,WATCH,J,mod,SLANTJ)
integer*2 A(512), WWFLAG,J16,J17,J18,J19,J26,J27,J36,J37,
J46,J47,
integer*4 I17SUB,J17,J27,SCANSTEP,INDEXSTEP,ZIGANS,NSCAN)
integer*2 A(512), WWFLAG,J16,J17,J18,J19,J26,J27,J36,J37,
J46,J47,
integer*4 I17SUB,J17,J27,SCANSTEP,INDEXSTEP,ZIGANS,NSCAN)
INTEGER*4 I17SUB,J17,J27,SCANSTEP,INDEXSTEP,ZIGANS,NSCAN)
real  DELAY(0:4), VOLTSET(0:4), VOL.TAGE(4), VOL(4), SLANT1, SLANT2, SLANT3, SLANT4
character DAY*9, TIM*9, VOLANS*, VOI..AGE(4), VOL(4), SLANT1, SLANT2, SLANT3, SLANT4
character DAY*9, TIM*9, VOLANS*, VOI..AGE(4), VOL(4), SLAN

common /SBLK/ A, DELAY, VOLTSET, MA1
C DATA 1/4/0
C if (noth.l.eq.1) then
   OPEN (unit=14, file='koth.menumonscans_interpret.log',
     status='NEW', access='SEQUENTIAI',
     form='UNFORMATTED')
endif
C
C 51 FORMAT(A42)
C 52 FORMAT(A42)
C 53 FORMAT(A4)
C 54 FORMAT(A2)
WFLAG = 0
10600 format('**+**
C
********** MODIFICATION FOR ZIGZAG SCAN *******************C
C
IF (ZIGANS.EQ.'Y') THEN
   IF (L.EQ.2.OR.L.EQ.4.OR.L.EQ.6.OR.L.EQ.8.OR.L.EQ.10
      OR.I.EQ.40.OR.L.EQ.42.OR.L.EQ.44.OR.I.EQ.46.OR.I.EQ.
      48.OR.L.IEQ.50.OR.L.LEQ.52.OR.L.EQ.54.OR.L.EQ.56.OR.
      I.EQ.58.OR.L.EQ.60.OR.L.EQ.62.OR.L.EQ.64.OR.L.EQ.66
      OR.L.EQ.68.OR.L.EQ.70.OR.L.EQ.72.OR.L.EQ.74.OR.I.
      EQ.76.OR.L.EQ.78.OR.L.EQ.80.OR.L.EQ.82.OR.L.EQ.84.OR
   ENDIF
I.EQ.86.OR.I.EQ.88.OR.I.EQ.90.OR.I.EQ.92.OR.I.
I.EQ.94.OR.I.EQ.96.OR.I.EQ.98.OR.I.EQ.100)THEN
ILOCX=NSCAN-J IF FOR ZIGZAG SCAN, REVERSE BACK FOR DISPLAY
ELSE
ILOCX=J-1
ENDIF
elseif(zigans.eq.'N')then !George Wood addition
ILOCX=J-1 !George Wood addition
ENDIF
ILOCY=1-1
IF ((IT.EQ.1.OR.IT.EQ.2)THEN !CORRECT DELAY WINDOW FOR NONLEVELNESS
DELAY_CORR=DELAY(IT)=(SLANTX*ILOCX*SCANSTEP)+(SLANTY*ILOCY*INDXSTEP)
ELSEIF (IT.EQ.3.OR.IT.EQ.4)THEN
DELAY_CORR=DELAY(IT)+(SLANTX*ILOCX*SCANSTEP)+(SLANTYR*ILOCY*INDXSTEP)
ENDIF
TYPE *,
TYPE *,
TYPE *,
TYPE *,!IT,J,; DELAY(IT)= (DELAY(IT)
TYPE *,!DELAY_CORR,; DELAY_CORR
TYPE *,
TYPE *,
TYPE *
CALL PUTTIME( DELAY_CORR ) ! Set delay
CONTINUE
IF (((IT,NE.4.AND.VOLANS.EQ.'U'))OR((IT.EQ.4.AND.VOLANS.EQ.'U')))THEN
VOI1S=VOI1S(VOL)
CALL SETVOLTDIV(VO1TS)
GOTO 1132
ENDIF
CALL AUTOSETVI0TS( MA1,VO1TS,NUMINT ) ! USER-DEFINED VOLTAGE SETTINGS
!
GOTO 1132
IF((VOI1TS.GT.1.0).OR.(VOI1TS.LT.0.01))THEN
WRITE(5,1133)VOI1TS
1133 FORMAT(*, 'BAD VOLTAGE SETTING',E10.5)
CALL TEKRESET(MAI)
GOTO 100
ENDIF
!
CALL GETSA(NAVE,MA1,A) 
!
GET waveform

WATCHI = A(I)' NAVE 
! Make sure its acceptable
TYPE *,
TYPE *
TYPE *
WRITE(5,1133)WATCHI,WATCH
1133 FORMAT(*,'CURRENT WATCH LEVEL IS:',F10.5,' MIN =',F10.5)
TYPE *
TYPE *
TYPE *, *

IF ((IT.NE.4.AND.VOLANS.EQ.'U') OR (IT.EQ.4.AND.UOLANS.EQ.'U')) GOTO 56732

!!! USER-DEFINED VOLTAGE SETTINGS

C IF (VOLANS.EQ.'U') GOTO 56732 !! SKIP WATCH

[Code snippet]

C*******************************~***~**~**~~~~~********~*****
56732 CONTINUE

C*******************************************************************************

10201  format('A', 'A', 'X=', 'J3', 'MAY LEAD TO BAD PROPERTY VALUE')
10200  format( 'WAVEFORM BELOW "WATCH", MAY LEAD TO BAD PROPERTY VALUE')
I27=I27+1

elseif (noth_l.eq.3) then

5800 WRITE(36,REC=I36,FMT=53) DELAY_CORR
I36=I36+1
WRITE(36,REC=I36,FMT=53) VOLTS
I36=I36+1
C
DO 58329 II=1,512
I37SUB=((I37-1)*512)+II
58329 WRITE(37,REC=I37SUB,FMT=54) A(III)
I37=I37+1

elseif (noth_l.eq.4) then

5900 WRITE(46,REC=I46,FMT=53) DELAY_CORR
I46=I46+1
WRITE(46,REC=I46,FMT=53) VOLTS
I46=I46+1
C
DO 59329 II=1,512
I47SUB=((I47-1)*512)+II
59329 WRITE(47,REC=I47SUB,FMT=54) A(III)
I47=I47+1
endif

C******************************************************************************C
IF (IT.EQ.1) VOLTAGEx1=VOLTS
IF (IT.EQ.2) VOLTAGEx2=VOLTS
IF (IT.EQ.3) VOLTAGEx3=VOLTS
IF (IT.EQ.4) VOLTAGEx4=VOLTS
C******************************************************************************C
600 CONTINUE
C******************************************************************************C
IF (IEQ.0.AND.IEQ.0) GOTO 20000 ! SKIP NEXT STEP (NOISE MEASURE)
C
CHECK FOR ERROR IN FS2 VOLTS
C
IF (VOLTAGEx1.EQ.0) THEN
WRITE (14) I,J,' VOLT FOR FS2= ',VOLTAGEx1
'C + ' #INT.PO. = ',NUMINT
GOTO 100
C
ENDIF
C
CHECK FOR ERROR IN B1 VOLTS
C
IF (VOLTAGEx1.LT.VOLTAGEx2) THEN
WRITE (14) I,J,' VOLT FOR B1= ',VOLTAGEx1
'C + ' #INT.PO. = ',NUMINT
GOTO 100
C
ENDIF
C
7 FORMAT (A4)
C******************************************************************************C
20000 CONTINUE

!0
RETURN

C

5,629,865
SUBROUTINE TAKEDATA(PLACE, SCHEME, VOLANS, VOL, LZEROO, FILENAME,  
UOLANS, I36, I27, NOTH_1, I35, I37, I46, I47,  
SLANT1, SLANT2, SLANT3, SLANT4, SCANSTEP, INDXSTEP, ZIGANS, NSCAN,  
RCMAX, MAXFS1PT,.MaxValue, RF1, RF2, MAXFS1PT, MAXVALT, RT1, RT2, J16,  
I17, NAVE, WATCH, J115, IT)  

integer*2 A16(512), I26, ILOCX, NSCAN, WFLAG, I16, MAXFS1PT, PLACE  
integer*2 WATCH_COUNT, JJ17, ILOCY, I36, I46  
integer*2 MAXFS1PT, NAVE, A(512), I17, I15, SCANKNT  
integer*4 I17SIJB, I17T, STAT, I17,FIONXSTEP, I37, I47  
integer*4 I17SUB, I17SUB, I17SUB  
REAL*4 ASPEC1(1024), PHASE1(1024), MAXVALF, MAXVALT, VOL(3), SLANT3, SLANT4  
real*4 DELAY(0:4), VOLTSET(0:4), VOLTAGE(4), SLANT, SLANT1, SLANT2, F(512)  
COMPLEX CF1(1024), CSPEC12(1024)  
character SCHEME*, FFLAG*, FILENAME*, UOLANS*, ZIGANS*  
character VOLANS*, TSCHEME*  
common /SBLW/ A, DELAY, VOLTSET, MAI  

C DATA I14/0/1137/0/  
C IF (NOTH_1.EQ.1) THEN  
OPEN (unit=14, file='FROTH.MEN/SCAN_INTERP.LOG',  
+ status='NEW', ACCESS='SEQUENTIAL',  
+ FORM='UNFORMATTED')  
ENDIF  
C  
51 FORMAT(A32)  
52 FORMAT(A2)  
53 FORMAT(A4)  
54 FORMAT(A2)  
WFLAG = 0  
C WATCH_COUNT=0  
10600 format (' ',!15 )  
IF (noth_1.I.EQ.1.AND.PLACE.EQ.1) THEN  
OPEN (unit=12, file='FILENAME/TH', status='NEW',  
+ form='UNFORMATTED', ORGANIZATION='SEQUENTIAL')  
SCANKNT=0  
ENDIF  
if (noth_1.I.EQ.1) then  
IX=1  
IY=1  
elseif (noth_1.I.EQ.2) THEN  
IX=2  
IY=2  
elseif (noth_1.I.EQ.3) THEN  
IX=3  
IY=3  
elseif (noth_1.I.EQ.4) THEN  
IX=4  
IY=4  
ENDIF
C MODIFICATION FOR ZIGZAG SCAN

IF (ZIGZANS.EQ.'Y') THEN
  IF (I.EQ.3 OR I.EQ.4 OR I.EQ.6 OR I.EQ.8 OR I.EQ.10)
    ELSE
      ILOCX=NSCAN-J
      ! FOR ZIGZAG SCAN, REVERSE BACK FOR DISPLAY
  ELSE
    ILOCX=J-1
  ENDIF
ELSEIF (ZI_EANS.EQ.'N') THEN
  ILOCX=J-I
  ! George Wood addition
ENDIF

ILOCY=I-I
IT:
  IF (IT.EQ.1 OR IT.EQ.2) THEN
    SLANTX=SLANTI
    SLANTY=SLANT3
  ELSEIF (IT.EQ.3 OR IT.EQ.4) THEN
    SLANTXR=SLANT2
    SLANTYR=SLANT4
  ENDIF

IF (NOTH.I.EQ.I)'I'YPE:
  IF (NOTH.I.EQ.1) TYPE *, 'IT=', IT
  IF (NOTH.I.EQ.N) TYPE *, 'PLACE=', PLACE
  IF (NOTH.I.EQ.I) TYPE *, 'SCHEME=', SCHEME
  IF (NOTH.I.EQ.I) TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  TYPE *, ' ', TYPE
  END

DELAY_CORR=DELAY(IT)+(SLANTX*ILOCX*SCANSTEP)+(SLANTY*ILOCY*INDEXSTEP)
ELSEIF (IT.EQ.1 OR IT.EQ.2) THEN
  DELAY_CORR=DELAY(IT)+(SLANTXR*ILOCX*SCANSTEP)+(SLANTYR*ILOCY*INDEXSTEP)
ENDIF
IF (PLACE.GE.1.AND.SCHEME.EQ.'B'.AND.NOTH-I.EQ.1) THEN
  HIT$=2  ! CORRECT DELAY WINDOW FOR NONLEVELNESS
  DELAY_CORR=DELAY(2)+(SLANTX*ILOCX*SCANSTEP)+(SLANTY*ILOCY*INDXSTEP)
  CALL PUTTIME(DELAY_CORR)  ! Set delay FOR B2

!! USER-DEFINE VOLTAGE SETTINGS?

IF (VOLANS.EQ.'U') THEN
  VOLTS=VOL(I)
  CALL SETVOLTDIV(VOLTS)
ENDIF
IF (VOLANS.EQ.'A') CALL AUTOSETVOLTS( MA1,VOLTS,NUMINT ) Set V/D
GOTO 1132
ENDIF

C KEEP TRACK OF POSITION WHERE DATA IS ACTUALLY TAKEN WITH A COUNTER
IF (IT.EQ.1.AND.PLACE.GE.1.AND.SCHEME.EQ.'X') SCANKNT=SCANKNT+1
IF (NOTH-I.EQ.1) TYPE*,' '
IF (NOTH-I.EQ.1) TYPE*,' X=',J,' Y=',I,' SCANKNT=',SCANKNT
IF (NOTH-I.EQ.1) TYPE*,' '
CALL PUTTIME(DELAY_CORR)  ! Set delay AS USUAL

CONTINUE
IF ((VOLANS.EQ.'U'.AND.(IT.EQ.1.OR.IT.EQ.2.OR.IT.EQ.3)).OR.
1 (VOLANS.EQ.'U'.AND.IT.EQ.4)) THEN  !!! USER-DEFINE VOLTAGE SETTINGS
  VOLTS=VOL(IT)
  CALL SETVOLTDIV(VOLTS)
  GOTO 1132
ENDIF

CALL AUTOSETVOLTS( MA1,VOLTS,NUMINT )  ! Set V/D

1132 IF(VOLTS.GT.1.0).OR.(VOLTS.LT.0.0))THEN
  WRITE(5,1131) VOLTS
1131   FORMAT(10.5) BAD VOLTAGE SETTING
  CALL TERESET(MA1)
  GOTO 100
ENDIF

CALL GETSAVAVE,MA1,A1)  ! Get waveform
  TYPE*,' '
  TYPE*,' Make sure its acceptable
  WRITE(5,1133) WATCHI,WATCH
1133   FORMAT('+','CURRENT WATCH LEVEL IS',F10.5,'MIN =',F10.5)
  TYPE*,' '
  TYPE*,' '

IF (VOLANS.EQ.'U'.OR.VOLANS.EQ.'U') GOTO 56732  !! SKIP WATCH
IF (WATCH.LT.WATCH) THEN
  WFLAG = 1
  WRITE(5,10200)
  FLAG='H' !H = SAMPLE HOLDER
  TYPE='WAVELAG'=FLAG
END IF

C COUNTER FOR BAD WATCH LEVEL, IE. ON WHICH WAVEFORM (B1 OR B2) DOES IT OCCUR
C
C IF (SCHEME.EQ.'X'.AND.I'I.EQ.1) WATCH_COUNT=1
C IF (SCHEME.EQ.'X'.AND.I'I.EQ.2) WATCH_COUNT=2
C TYPE='WATCH_COUNT'=WATCH_COUNT
GOTO 20000
10201 format('@A',A; ',A: X=',X; 'Y=',Y )
10200 format( 'WAVEFORM BELOW "WATCH", GOING TO NEXT SCAN POINT')
C GOTO 100
ENDIF

C**********************************************************************

C**********************************************************************

C**********************************************************************

C**********************************************************************

C**********************************************************************

C **********************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************

C*************************************************************************
ELSEIF (TSCHEME.EQ.'M'.AND.PLACE.EQ.1) THEN
IF (NOTH_I.EQ.1) TYPE *,
IF (NOTH_I.EQ.1) TYPE *
MANUALLY-SET THRESHOLD VOLTAGE NOISE LEVEL = ZERO
ENDIF

C ##########################################################################
IF (ZEROO.LT.ZERO0) THEN !!!< EX.002 VOLTS - ON SAMPLE HOLDER
IF (NOTH_I.EQ.1) TYPE *,FLAG=!,FLAG
GOTO 20000
ELSEIF (ZEROO.GE.ZERO0) THEN !!!> EX.002 VOLTS - ON SAMPLE
SHEME=*
GOTO 10386 |START TAKEDATA LOOP AGAIN
ENDIF
ENDIF

IF (NOTH_I.EQ.1) THEN
WRITE(16,REC=I16,FMT=53) DELAY_CORR
I16=I16+1
WRITE(16,REC=I16,FMT=53) VOLTS
I16=I16+1
C
DO 54321 IJI=1,512
I17SUB=((17-1)*5)+1J1
54321 WRITE(17,REC=I17SUB,FMT=54) A(IJI)
I17=I17+1
elseif (notI.EQ.2) then
WRITE(26,REC=I26,FMT=53) DELAY_CORR
I26=I26+1
WRITE(26,REC=I26,FMT=53) VOLTS
I26=I26+1
C
DO 54329 IJI=1,512
I27SUB=((27-1)*5)+1I1
54329 WRITE(27,REC=I27SUB,FMT=54) A(IJI)
I27=I27+1
elseif (notI.EQ.3) then
WRITE(36,REC=I36,FMT=53) DELAY_CORR
I36=I36+1
WRITE(36,REC=I36,FMT=53) VOLTS
I36=I36+1
C
DO 58329 IJI=1,512
I37SUB=((37-1)*5)+1I1
58329 WRITE(37,REC=I37SUB,FMT=54) A(IJI)
I37=I37+1
elseif (notI.EQ.4) then

16
5900 WRITE(46,REC=146,FMT=53)DELAY_CORR
146=146+1
WRITE(46,REC=146,FMT=53)VOLTS
146=146+1
C
DO 5929 JJ=1,512
147SUB=(147-1)*512+1
59329 WRITE(47,REC=147SUB,FMT=54) A(JJ)
147=147+1
.endsf

C
C WRITE(8,10600) A(200)
C*******************************************************************************C
IF (IT.EQ.1)VOLTAGE(1)=VOLTS
IF (IT.EQ.2)VOLTAGE(2)=VOLTS
IF (IT.EQ.3)VOLTAGE(3)=VOLTS
IF (IT.EQ.4)VOLTAGE(4)=VOLTS
C*******************************************************************************C
600 CONTINUE
C*******************************************************************************C
7 FORMAT (A4)
C*******************************************************************************C
20000 IF (WFLAG.EQ.1)THEN
CLOSE(8)
OPEN(unit=8, file="1'XA I
C Status='OLD")
ENDIF
IF (SCHEME.EQ.'X')SCHEME='B'
!RESET BACK '1'0
'H' FOR NEXT SCAN POINT
IF (1.NE.0.AND.J.NE.0)THEN
IF (notl-1.eq.1)WRITE(12)J,I,FI,AS,SCANKNT
!NOTE LOCATION WIT11 FLAG
& LEAVE SUBR.
IF (NOTH-1.EQ.l)TYPE *,'X=',J,' Y=',I,' FLAG=',FLAG
IF (NOTH-1.EQ.1)TYPE *,'
I WHERE BAD WATCH LEVEL WAS FOUND
C WHERE BAD WATCH LEVEL WAS FOUND
IF (NOTH .LEQ. 1) TYPE *, *
IF (NOTH .LEQ. 1) TYPE *, *
IF (NOTH .LEQ. 1) TYPE *, *
C ENDIF
C WATCH_COUNT=0
RETURN
end

C
SUBROUTINE PRESSURE(VOLTAGEL, VOLTAGEU)

C

C **** PRESSURE=20*VOLTAGE **** (LOWER.LE.VOLTAGE.GE.UPPER)
C
C Adjust Z axis to get good pressure (PSI)
C
C <<< 11 = DOWN >>>
C
C PUPPER = .85 ! Upper Pressure = 17
C PLOWER = .75 ! Lower Pressure = 15
C PUPPER = .3 ! Upper Pressure = 6
C PLOWER = .2 ! Lower Pressure = 4
C PUPPER = .5 ! Upper Pressure = 10
C PLOWER = .4 ! Lower Pressure = 8
C PUPPER = .6 ! Upper Pressure = 12
C PLOWER = .7 ! Lower Pressure = 14
C PUPPER = .8 ! Upper Pressure = 16
C PLOWER = .6 ! Lower Pressure = 12
C PUPPER = .61 ! Upper Pressure = 12.2
C PLOWER = VOLTAGE
C PUPPER = VOLTAGEU

200 CALL GETFLUKE(P)
IF( P.GE.PLOWER .AND. P.LE.PUPPER) GOTO 900
IF( P.LT.PLOWER) CALL MOVXYZ(3, -I)
IF( P.GT.PUPPER) CALL MOVXYZ(3, 1)
GOTO 200

900 RETURN

end
SUBROUTINE PRESSURE_MAX(VOLTAGE_MAX,LLL,FILENAME,J1,I15,CHECKP)
C
 CHARACTER FILENAME*34
 INTEGER*2 CHECKP,LLL,LINCR,J,JXXX,J15,ITTT
52 FORMT(A2)
P_MAX=VOLTAGE_MAX
200 CALL GETFLUKE( P )
 IF( P.GT.P_MAX) THEN
  CHECKP=CHECKP+1
  IF( CHECKP.EQ.1)LINCR=LLL/3
  IF( CHECKP.EQ.2)LINCR=LLL/2
  IF( CHECKP.EQ.3)LINCR=1000
  DO 2900 ITTT=1,LINCR
  CALL MOVXYZ( 3,-1 ) ! 3 IS Z-AXIS, MOVE Z-AXIS UP
  IF( CHECKP.EQ.3) THEN
  JXXX=+1 ! IS Y-LOC; GO BACK TO PREVIOUS ROW
  WRITE(15,REC=-15,FMT=52)JXXX,
  RETURN
  ENDIF
  ENDFP
2900 RETURN
C
end
SUBROUTINE MOVORG( SCANAXIS, NSTEPS, VOLTAGEL, VOLTAGEU,  
    1 LOOPTHRU_H)
C   Run the thing back to the SCANAXIS origin
C
   integer*4 SCANAXIS
   integer*2 NSTEPS,LOOPTHRU_H
C
   DO 1000 K=I,LOOPTHRU_H
C 1000 CALL MOVXYZ(3,-I) ! Move up
   DO 1200 K=I,NSTEPS-I
C 1200 CALL MOVXYZ( SCANAXIS, -1 ) ! Move back
C   DO 1400 K=I,LOOPTHRU_H
C 1400 CALL MOVXYZ(3,1) ! Move down
C   CALL PRESSlRE(VOLTAGEL, VOLTAGEU) !nothick
C   CALL MOVXYZ( SCANAXIS, 1 )
   RETURN
end

SUBROUTINE MOVORGXY( ZIGANS, SCANAXIS, NSCAN, INDXAXIS, NINDX)
C   Run the thing back to the SCAN origin
C
   CHARACTER ZIGANS*1
   integer*4 SCANAXIS,INDXAXIS
   integer*2 NSCAN,NINDX,1
C
   IF (ZIGANS.EQ.'Y')THEN
      l=Nindx
      IF (l.EQ.1 .OR. l.EQ.3 .OR. l.EQ.5 .OR. l.EQ.7 .OR. l.EQ.9  
          1 .OR. l.EQ.11 .OR. l.EQ.13 .OR. l.EQ.15 .OR. l.EQ.17 .OR. l.  
          1 .OR. l.EQ.19 .OR. l.EQ.21 .OR. l.EQ.23 .OR. l.EQ.25 .OR. l.EQ.27 .OR.  
          1 .OR. l.EQ.29 .OR. l.EQ.31 .OR. l.EQ.33 .OR. l.EQ.35 .OR. l.EQ.37 .OR.  
          1 .OR. l.EQ.39 .OR. l.EQ.41 .OR. l.EQ.43 .OR. l.EQ.45 .OR. l.EQ.47 .OR.  
          1 .OR. l.EQ.49 .OR. l.EQ.51 .OR. l.EQ.53 .OR. l.EQ.55 .OR.  
          1 .OR. l.EQ.57 .OR. l.EQ.59 .OR. l.EQ.61 .OR. l.EQ.63 .OR. l.EQ.65 .OR.  
          1 .OR. l.EQ.67 .OR. l.EQ.69 .OR. l.EQ.71 .OR. l.EQ.73 .OR. l.  
          1 .OR. l.EQ.75 .OR. l.EQ.77 .OR. l.EQ.79 .OR. l.EQ.81 .OR. l.EQ.83 .OR.  
          1 .OR. l.EQ.85 .OR. l.EQ.87 .OR. l.EQ.89 .OR. l.EQ.91 .OR.  
          1 .OR. l.EQ.93 .OR. l.EQ.95 .OR. l.EQ.97 .OR. l.EQ.99)THEN
         DO 1200 K=1,NSCAN-1
C 1200 CALL MOVXYZ( SCANAXIS, -1 ) ! Move back X
ENDIF
ENDIF
C
   DO 1400 K=1,NINDX-1
C 1400 CALL MOVXYZ(INDXAXIS, -1) ! Move back Y
RETURN

end

C
SUBROUTINE ENTERPARAM( SCANSTEP,INDXSTEP,NSCAN,NINDX,NAVE,VOLTAGE1

1, VOLTAGET,VOLTAGEMAX,SHAPEANS,LOOPANS,LOOPTHRU_H,LOOPTHRU_S,
1 ZIGANS,SCHEME,TSCHMEM,IZERGO,FILNANE1,I5,SCANMODE,SLANT1,SLANT2,
1 SLANT3,SLANT4)

integer*4 SCANSTEP,INDXSTEP
real SCANDIST,INDXDIST,PU,PL,SLANT,SLANT1,SLANT2,
REAL SLANT3,SLANT4,RSLANT3,RSLANT4
INTEGER*2 LOOPTHRU_H,LOOPTHRU_S,ISLANT1,ISLANT2
integer*2 NSCAN,NINDX,IDIST1,IDIST2,16,115
integer*2 FREQ,NAVE,ISLANT3,ISLANT4
character FILENAME*34, FILEEXT*23, CHEADER*32
CHARACTER SHAPEANS*I,ZIGANS*I,
CHARACTER TSCHMEM*I,SCSCHEME*I
DATA 16/11
115=1
10010 format( )
10012 format(A)
10020 format(I)
10030 format(F)
C C Initialization and entry of parameters
C C
TYPE *,
TYPE *, *****> DONSCANNFOR......AUTO SCAN
TYPE *,
OPEN(UNIT=10,FILE='NOTHICK_ALLSHAPE.1.DAT',STATUS='OLD',
FORM='FORMATTED')
READ (10,93764) SCANMODE
READ (10,93765) XSCANSTEP !(X IND)
IF (SCANMODE.EQ.'P')XSCANSTEP=2000.
READ (10,93765) YINDXSTEP !(Y IND)
IF (SCANMODE.EQ.'P')YINDXSTEP=2000.
SCANSTEP=JNINT(XSCANSTEP)
INDXSTEP=JNINT(YINDXSTEP)
READ (10,93765) SCANDIST !(X DIST)
IF (SCANMODE.EQ.'P')SCANDIST=2000.
READ (10,93765) INDXDIST !(Y DIST)
IF (SCANMODE.EQ.'P')INDXDIST=2000.
IF (SCANMODE.EQ.'L')THEN
YINDXSTEP=2000.
INDXSTEP=JNINT(YINDXSTEP)
INDXDIST=2000.
ENDIF
READ (10,93764) ZIGANS
READ (10,93765) PL

!CONVERT SLANT TO INTEGER FOR STORAGE IN .DAT2 FILE
ISLANT1=INT(RSLANT1)

!CONVERT SLANT TO INTEGER FOR STORAGE IN .DAT2 FILE
ISLANT2=INT(RSLANT2)

!CONVERT SLANT TO INTEGER FOR STORAGE IN .DAT2 FILE
ISLANT3=INT(RSLANT3)

!CONVERT SLANT TO INTEGER FOR STORAGE IN .DAT2 FILE
ISLANT4=INT(RSLANT4)

READ (10,93765) SLANT1
SLANT1=SLANT1/10.**9.
RSLANT1=SLANT1*10.**13.

READ (10,93765) SLANT2
SLANT2=SLANT2/10.**9.
RSLANT2=SLANT2*10.**13.

READ (10,93765) SLANT3
SLANT3=SLANT3/10.**9.
RSLANT3=SLANT3*10.**13.

READ (10,93765) SLANT4
SLANT4=SLANT4/10.**9.
RSLANT4=SLANT4*10.**13.

READ (10,93765) FREQUENCY
READ (10,93764) FILEEXT
K=0
K=K+1
IF FILEEXT(K:K).NE."')GOTO 126 ! FILEEXT(K+1:K+4)=.DAT ! UNIT 6 = raw data

READ (10,93764) CHEADER
93764 FORMAT (A)
93765 FORMAT (F)
93766 FORMAT (I)
C VOLTAGE=PL/20.
C VOLTAGE=PL/20.
C TYPE="RUN TEMPDIRSCAN.COM - DIRSCAN SMALLTRANS_DATA.DAT"
TYPE="DATA FOR ULTRASONIC SCAN (ALL FLOATING PT. EXCEPT INDEXES)"
TYPE="SCAN DIMENSIONS/INDEXING DATA (MICRONS)"
TYPE="X INDEX - INTEGER EXPRESSION"
TYPE="SCANSTEP"
TYPE="Y INDEX - INTEGER EXPRESSION"
TYPE="INDEXSTEP"
TYPE="X SCAN DISTANCE (MICRONS) - FLOATING POINT"
TYPE="X SCAN DISTANCE (MICRONS) - FLOATING POINT"
TYPE="INDEXDIST"
C TYPE="LOWER & UPPER Pressures FOR CONTACT SCAN"
C TYPE="PL,PU"
C TYPE="THICKNESS (MM)"
C TYPE="THICKH"
TYPE="SLANT1 (X-LEVELNESS TIME CORRECTION FACTOR for b1,b2) (nsec / um)"
TYPE="SLANT1"
TYPE="SLANT2 (X-LEVELNESS TIME CORRECTION FACTOR for reflector echoes) (nsec / um)"
TYPE="SLANT2"
TYPE="SLANT3 (Y-LEVELNESS TIME CORRECTION FACTOR for b1,b2) (nsec / um)"
TYPE="SLANT3"
TYPE="SLANT4 (Y-LEVELNESS TIME CORRECTION FACTOR for reflector echoes) (nsec / um)"
TYPE="SLANT4"
TYPE="TRANSDUCER CENTER FREQUENCY (FLOATING POINT)"
TYPE="FREQUENCY"
TYPE="FILENAME FOR RAW DATA"
CFILEEXT
C FILEEXT
TYPE="CHEADER INFO (UP TO 32 CHARACTERS)"
TYPE="CHEADER"
TYPE="*
1000 NSCAN = (SCANSTEP/SCANSTEP) + 1
NINDX = (INDEXDIST/INDEXSTEP) + 1
SCANSTEP = NSCANSTEP(NSCAN-1)
INDEXDIST = INDEXSTEP(NINDEX-1)
IF (SCANMODE.EQ.'S') THEN
WRITE(5,11100) SCANDIST, NSCAN, SCANSTEP, INDEXDIST, INDEXSTEP
ELSEIF (SCANMODE.EQ.'L') THEN
WRITE(5,11200) NSCAN, SCANSTEP, INDEXDIST, INDEXSTEP
ELSEIF (SCANMODE.EQ.'P') THEN
WRITE(5,11300)
ENDIF
11100 format (*
+ "X scan to .F7.0,.'; steps of .J5,
+ "Y index to .F7.0,.'; steps of .J5)
5,629,865

11200 format(' ',
     - // Line scan to 'F7.0'; 'J3'; steps of 'J5')
11300 format(' ', // Point Measure repeated 2 times)

C
THICK= INT(THICKN*100.)
FREQ= FREQUENCY
NAV=64
FILENAME = [ROTH.DATA]//FILEEXT
WRITE( 5,11690 )
11690 format( /'NUMBER OF AVERAGES SET AT 64' )

C
C DIRECT ACCESS STORAGE
C
OPEN( unit=6, file=FILENAME//CII, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=32,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=15, file=FILENAME//DII, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=16, file=FILENAME//FII, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=4,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=17, file=FILENAME//WAV, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=26, file=FILENAME//RIV, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=4,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=27, file=FILENAME//RWAV, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=36, file=FILENAME//SR4, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=4,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=37, file=FILENAME//SWAV, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=46, file=FILENAME//TR4, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=4,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )
OPEN( unit=47, file=FILENAME//TWAV, status='NEW,
     + ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
     + form='FORMATTED',ORGANIZATION='SEQUENTIAL' )

51 FORMATA(32)
52 FORMATA(2)
53 FORMATA(4)
54 FORMATA(2)
C
C Write header : X, Y dist, X. Y scan points, dummies
C
WRITE(6,REC=46,FMT=51) CHEADER
16=(a+1)
WRITE(6,REC=46,FMT=51) SHAPESANS
69

WRITE(6,REC=16,FMT=51)ZIGANS

IDIST1=-HFIX(SCANDIST/1000.)
IDIST2=IFIX(INDXDIST/1000.)
IF (IDIST2.LT.1.)IDIST2=1. !TRICK TO ALLOW DISPLAY OF IMAGE ON PSIDD

WRITE(15,REC=115,FMT=52)IDIST1

115=115+1
WRITE(15,REC=115,FMT=52)IDIST2

115=115+1
WRITE(15,REC=115,FMT=52)NSCAN

115=115+1
IF (SCANMODE.EQ.‘L’.O.R.SCANMODE.EQ.‘P’)NINDX=1 !MOD FOR LINE/POINT SCAN

WRITE(15,REC=115,FMT=52)NINDX

115=115+1
WRITE(15,REC=115,FMT=52)ISLANT1

115=115+1
WRITE(15,REC=115,FMT=52)ISLANT2

115=115+1
WRITE(15,REC=115,FMT=52)ISLANT3

115=115+1
WRITE(15,REC=115,FMT=52)ISLANT4

115=115+1
WRITE(15,REC=115,FMT=52)DENS

115=115+1
WRITE(15,REC=115,FMT=52)FREQ

115=115+1
WRITE(15,REC=115,FMT=52)NAVE

115=115+1
RETURN
end

SUBROUTINE XPARAM( SCANSTEP,INDXSTEP,NSCAN,NINDX )

integer*4 SCANSTEP,INDXSTEP

integer*2 NSCAN,NINDX

SCANSTEP = 50

INDXSTEP = 50

NSCAN = 50

NINDX = 50

THICK = 1

OPEN(unit=6, file='XXX.DAT', status='OLD',

+ form='UNFORMATTED')

SCANSTEP = 50

RETURN

end

SUBROUTINE MOVKLING (AXIS,DIST)

MOVE KLINGER A SPECIFIED DISTANCE IN SPECIFIED DIRECTION

C

byte IXYZ

character*1 XYZ

equivalence( XYZ, IXYZ )

common WRK0,BUFFER
CALL STRTGPIB
CALL INITINSTR(2)
CALL INITINSTR(3)
CALL INITINSTR(4)
10010 format(A)
10020 format(I)
10100 format("Distance ?")
100 WRITE(5,10000)
10000 format("X, Y, or Z ?")
READ(5,10010)XYZ
IF(IXYZ.GT.90. OR .IXYZ.LT.88)GOTO 900
IPORN=1
WRITE(5,10100)
READ(5,10020)IREQ
IF(IREQ.LT.0)THEN
   IPORN=-1
   IREQ=ABS(IREQ)
ENDIF
CALL SETXYZ(IREQ,IREQ,IPORN)
CALL MOVXYZ(IAXIS,IPORN)
GOTO 100
900 STOP
RETURN
end

INCLUDE [ROTH.MENU]BASEO_2SEC_M.FOR
C northick_crunh.f90 --- 4 SCAN METHOD
C Crunch program for Noththickness velocity scan
C Don Roth 20-asp-1994

integer*2 SCANDIST,INDXDIST,NSCAN,INDX,J,SCANKNT
integer*2 DENS,TRFREQ,AVES,NPOINT,16,115,116,118,326
integer*2 ARRAYSZ,EFFREQ,ENDFREQ,ISJ36,446
INTEGER*2 islant1,islant2,ISLANT3,ISLANT4,ACTPOINT,J XXX
integer*2 rawwav1(512),rawwav2(512),rawwav3(512),rawwav4(512)
INTEGER*4 117,17,REC,117SUB,337,447,117SUB,447SUB,117SUB
real*4 LI,VEL_U,VEL_AV,VEL
real*4 ASPCB1(1024), ASPCB2(1024)
real*4 BSPEC(256), BSPEC2(256)
real*4 ASPCR(1024), ASPCRNS(1024)
real*4 RSPEC(256), RNSPEC(256)
REAL*4 R1PHASE(256), PHASEB1(1024)
REAL*4 R2PHASE(256), PHASEB2(1024)
REAL*4 RS1PHASE(256), PHASERS(1024)
REAL*4 BB1(512),BB2(512),RSR(512),RNSR(512)
REAL*4 VEL(5000),TRDIAM,BUFFLENGTH
character FILENAME*52, FILEEXT*26, HEADER*32, NOISANS*1
character FSIEXT*I6, CALNAME*32, FLAG*1
CHARACTER DIFFANS*I2, PHASE*I1, PHASEI*I1
CHARACTER SELFREQANS*I1
CHARACTER SHAPEANS*I1, ZIGANS*I1
CHARACTER DIFKANS*1, DIFK*I
CHARACTER CRUNCH_CODE*2, CRUNCH_CODEI*1
character actual_selfreqans*I1
byte \ FILEFS1N
equivalence ( BSPEC, ASPECB1 ) !ASPEC 256
equivalence ( BSPEC2, ASPECB2 ) !ASPEC 256
equivalence ( RSPEC, ASPECB1 ) !RSPEC 256
equivalence ( RSPEC2, ASPECB2 ) !RSPEC 256
equivalence ( RNSPEC, ASPECB1 ) !RSPEC 256
equivalence ( RNSPEC2, ASPECB2 ) !RSPEC 256
equivalence ( B1PHASE, PHASEB1 )
equivalence ( B2PHASE, PHASEB2 )
equivalence ( R1PHASE, PHASEB1 )!
equivalence ( R2PHASE, PHASEB2 )!
equivalence ( RNS1PHASE, PHASERS )!
equivalence ( RNS2PHASE, PHASERS )!
equivalence ( FSIEXT, FILEFS1N )

DATA 16/115/116/117/117/118/1/117/117/117/117/1/118/1/
DATA 126/1/127/1/185/1/136/1/137/1/146/1/147/1/
NP=0

10010 format(A)
10014 format(’A ’)
10020 format(I)
10030 format(F)
C
OPEN (UNIT=14, FILE='NOT_HICK_SELFREQ.DAT', STATUS='OLD', FORM='FORMATTED')
READ (14,98652) FILEEXT
READ (14,98652) DRIVE
READ (14,98652) PHASE
READ (14,98652) PHASE1
READ (14,98652) DIFFANS
READ (14,10030) BUFLENGTH
READ (14,10030) BUFVEL
READ (14,10030) TRDIAM
CLOSE (14)

OPEN (UNIT=24, FILE='NOT_HICK_LOWER.LIMITS.DAT', STATUS='OLD', FORM='FORMATTED')
READ (24,98652) CFILTER
READ (24,10030) L_VEL
READ (24,10030) U_VEL
U_VEL=L_VEL+U_VEL
AV_VEL=(L_VEL+U_VEL)/2.

K=0
K=K+1
IF (FILEEXT(K:K).NE. 'DAT') GOTO 126 ! FILEEXT(K+1:K+4)='.DAT' ! UNIT 6 = raw data
IF (DRIVE.EQ.'A') DIR='DUC2:'
IF (DRIVE.EQ.'C') DIR='DUC6:'
FILENAME = 'ROTH.DAT'/FILEEXT

OPEN (UNIT=6, FILE=FILENAME//'CH', ACCESS='DIRECT', RECORDTYPE='FIXED', RECL=4, FORM='FORMATTED', ORGANIZATION='SEQUENTIAL')
OPEN (UNIT=66, FILE=FILENAME//'CH', ACCESS='DIRECT', RECORDTYPE='FIXED', RECL=32, FORM='FORMATTED', ORGANIZATION='SEQUENTIAL')
OPEN (UNIT=13, FILE=FILENAME//'FE', ACCESS='DIRECT', RECORDTYPE='FIXED', RECL=2, FORM='FORMATTED', ORGANIZATION='SEQUENTIAL')
OPEN (UNIT=16, FILE=FILENAME//'PI', ACCESS='DIRECT', RECORDTYPE='FIXED', RECL=4, FORM='FORMATTED', ORGANIZATION='SEQUENTIAL')
OPEN (UNIT=17, FILE=FILENAME//'AV', ACCESS='DIRECT', RECORDTYPE='FIXED', RECL=2, FORM='FORMATTED', ORGANIZATION='SEQUENTIAL')
C
51 FORMAT(A32)
52 FORMAT(A2)
53 FORMAT(A4)
153 FORMAT(1X,14,1X,14,1X,1X,E12.5,E12.5,E12.5,E12.5,E12.5,E12.5)
54 FORMAT(A2)
K=0
120 IF( FILEEXT(K-K-1),NE.'.' )GOTO 120 ! UNIT 6 - raw data
FILEEXT(K)=FILEEXT(K+1:K+3)='CAL'
CALNAME = '/ROTH.DATA'//FILEEXT
OPEN( unit=8, file=CALNAME, status='NEW', ACCESS='DIRECT,' + 
    RECORDTYPE='FIXED',RECL=4,FORM='FORMATTED', + 
    ORGANIZATION='SEQUENTIAL')
WRITE( 6,10200 )FILENAME
10200 FORMAT( *Filename of analyzed data is ',A 1 )
FILEEXT(K+1:K+3)='CAL'
CALNAME = '/ROTH.DATA'//FILEEXT
OPEN( unit=8, file=CALNAME, status='NEW', ACCESS='DIRECT,' + 
    RECORDTYPE='FIXED',RECL=4,FORM='FORMATTED', + 
    ORGANIZATION='SEQUENTIAL')
89651 DO 76065 IOPI=1,1024
    PHASEB1(IOPI)=0.
    PHASEB2(IOPI)=0.
    PHASERB(IOPI)=0.
    PHASERSN(IOPI)=0.
    ASPECB1(IOPI)=0.
    ASPECB2(IOPI)=0.
    ASPECERSN(IOPI)=0.
76065 ASPECERSN(IOPI)=0.

KCOUNT=1

C # # # Read preliminary scan info # # #
C
79193 READ(66,REC=16,FMT=51)CHHEADER
  16=16+1
READ(66,REC=16,FMT=51)SHAPEANS
  16=16+1
READ(66,REC=16,FMT=51)ZIGANS
  16=16+1

IF (SHAPEANS.NE.'Y')GOTO 89714
OPEN(UNIT=12,FILE='FOTH.DATA',FILEEXT(1,K)=DATTH,
  STATUS='OLD',FORM='UNFORMATTED') INFO ON WHETHER WE ARE ON HOLDER/SAMPLE
1

89714 READ(15,REC=115,FMT=52)SCANDIST
  115=115+1
READ(15,REC=115,FMT=52)INDEXDIST
  115=115+1
READ(15,REC=115,FMT=52)NSCAN
  115=115+1
READ(15,REC=115,FMT=52)NINDEX
  115=115+1
READ(15,REC=115,FMT=52)ISLANT1
  115=115+1
READ(15,REC=115,FMT=52)ISLANT2
  115=115+1
READ(15,REC=115,FMT=52)ISLANT3
  115=115+1
READ(15,REC=115,FMT=52)ISLANT4
  115=115+1
READ(15,REC=115,FMT=52)DENS
  115=115+1
READ(15,REC=115,FMT=52)JXXX
  115=115+1
ARRAYSIZE=(NSCAN)*(JXXX) IF SCAN FINISHED PROPERLY, JXXX=NINDEX
READ(16,REC=16,FMT=53)TIMESET
  116=116+1

ENDFREQ=TRFREQ*2.5
CALL POINT_FROM_FREQ(ENDFREQ,TIMESET,ACTPOINT)

DELTAF = 1./TIMESET**20.
NPOINT=NSCAN*JXXX IF SCAN FINISHED PROPERLY, JXXX=NINDEX
NWAVES = 3*NPOINT

C !!! Loop thru scan data

ITEM_COUNT=0
DO 1890 NIP=1,NPOINT !NPOINT IS TOTAL OF ALL POINTS
SCANKNT=NIPI
ITEM_COUNT=ITEM_COUNT+1
IF (SHAPEANS.EQ.'N' AND NIP.GE.2)ITEM_COUNT=2
TYPE *, *
ITEM_COUNT=ITEM_COUNT+1
TYPE *, CRUNCH_CODE=CRUNCH_CODE
TYPE *, *
CCVACCUM=0.
CONTINUE
NIPI=((NIPI-1)/SCAN)+1 !Y LOCATION
NIPI=NIPI+1 !X LOCATION
IF (NIPI.EQ.(SCAN+1))NIPI=1 !RESET X LOCATION
C ***** MODIFICATION FOR ZIGZAG SCAN *****************************************
IF (ZIGANS.EQ.'Y')THEN
IF (NIPI.EQ.2 OR NIPI.EQ.4 OR NIPI.EQ.6 OR NIPI.EQ.8 OR NIPI.EQ.10)
1 OR NIPI.EQ.12 OR NIPI.EQ.14 OR NIPI.EQ.16 OR NIPI.EQ.18 OR NIPI.
1 EQ.20 OR NIPI.EQ.22 OR NIPI.EQ.24 OR NIPI.EQ.26 OR NIPI.EQ.28 OR
1 NIPI.EQ.30 OR NIPI.EQ.32 OR NIPI.EQ.34 OR NIPI.EQ.36 OR NIPI.EQ.38.
1 OR NIPI.EQ.40 OR NIPI.EQ.42 OR NIPI.EQ.44 OR NIPI.EQ.46 OR NIPI.EQ.
1 48 OR NIPI.EQ.50 OR NIPI.EQ.52 OR NIPI.EQ.54 OR NIPI.EQ.56 OR.
1 NIPI.EQ.58 OR NIPI.EQ.60 OR NIPI.EQ.62 OR NIPI.EQ.64 OR NIPI.EQ.66.
1 OR NIPI.EQ.68 OR NIPI.EQ.70 OR NIPI.EQ.72 OR NIPI.EQ.74 OR NIPI.
1 EQ.76 OR NIPI.EQ.78 OR NIPI.EQ.80 OR NIPI.EQ.82 OR NIPI.EQ.84 OR
1 NIPI.EQ.86 OR NIPI.EQ.88 OR NIPI.EQ.90 OR NIPI.EQ.92 OR NIPI.
1 EQ.94 OR NIPI.EQ.96 OR NIPI.EQ.98 OR NIPI.EQ.100)THEN
NIPI=SCAN-NIPI+1 !FOR ZIGZAG SCAN, REVERSE BACK FOR DISPLAY
ELSE
NIPI=NIPI
ENDIF
elseif(zigans.eq.'N')then !George Wood addition
nipl=nipl !George Wood addition
ENDIF
TYPE *, *
TYPE *, *
IF (SHAPEANS.EQ.'N')GOTO 89715
READ(12,II,FLAG,SCANKNT !READ WHETHER 'H' OR 'S'
TYPE *, X=NIPI, Y=NIPI, FLAG=FLAG
ENDIF
IF (FLAG.EQ.'S')THEN
READ(16,REC=116,FMT=53)DELAYB1
116=116+1
READ(16,REC=116,FMT=53)VOLTSETB1
116=116+1
READ(26,REC=126,FMT=53)DELAYB2
126=126+1
READ(26,REC=126,FMT=53)VOLTSETB2
126=126+1
READ(36, REC=136, FMT=53) DELAY RS
READ(36, REC=136, FMT=53) VOLTSETRS
READ(46, REC=146, FMT=53) DELAYRNS
READ(46, REC=146, FMT=53) VOLTSETRNS
READ(57, REC=157, FMT=53) VOLTSETRNS

DO 54322 J1=1,512
54322 READ(17, REC=((117-1)*512)+J1, FMT=54) RAWWAVEB1(J1)
117=117+1
DO 54323 J1=1,512
54323 READ(27, REC=((127-1)*512)+J1, FMT=54) RAWWAVEB2(J1)
127=127+1
DO 54324 J1=1,512
54324 READ(37, REC=((137-1)*512)+J1, FMT=54) RAWWAVEB3(J1)
137=137+1
DO 54325 J1=1,512
54325 READ(47, REC=((147-1)*512)+J1, FMT=54) RAWWAVERN(J1)
147=147+1

ELSE IF (FLAG.EQ.'H') THEN
GOTO 1890
ENDIF

II=II+1
A = 0.
DO 1400 J1=1,512
1400 RAWWAVEB1(J1)=RAWWAVEB1(J1)/AVES
A=A+RAWWAVEB1(J1)
ZERO=AVES
DO 1560 J1=1,512
1560 BB1(J1)= (REAL(RAWWAVEB1(J1))-ZERO)*(VOLTSETB1)*10./512.
II=II+1
A = 0.
DO 1401 J1=1,512
1401 RAWWAVEB2(J1)=RAWWAVEB2(J1)/AVES
A=A+RAWWAVEB2(J1)
ZERO=AVES
DO 1570 J1=1,512
1570 BB2(J1)= (REAL(RAWWAVEB2(J1))-ZERO)*(VOLTSETB2)*10./512.
II=II+1
A = 0.
DO 1402 J1=1,512
1402 RAWWAVEB3(J1)=RAWWAVEB3(J1)/AVES
A=A+RAWWAVEB3(J1)
ZERO=AVES
DO 1580 J1=1,512
1580 BB3(J1)= (REAL(RAWWAVEB3(J1))-ZERO)*(VOLTSETB3)*10./512.
II=II+1
A = 0.
DO 1403 I = 1, 512
RAWWAVERNS(I) = RAWWAVERNS(I) / AVES
A = A + RAWWAVERNS(I)
ZERO = A / 512.
DO 1590 I = 1, 512
RRNS(I) = (REAL(RAWWAVERNS(I)) - ZERO) * VOLTSERNS * 10. / 512.

IF (PHASE.EQ. 'N') THEN
CALL CORR(BB1, BB2, TWOTAU, DELAY) ! for time delay in sample
ELSEIF (PHASE.EQ. 'Y') THEN  ! PHASE INVERSION FOR BETWEEN B1 & B2
CALL MCORR(BB1, BB2, TWOTAU, DELAY) ! for time delay in sample
ENDIF

IF (PHASE.EQ. 'N') THEN
CALL CORR(RRS, RRNS, DELTAT, DELAY) ! for delay between reflector peaks
ELSEIF (PHASE.EQ. 'Y') THEN ! PHASE INVERSION FOR BETWEEN RR & RNS
CALL MCORR(RRS, RRNS, DELTAT, DELAY) ! for delay between reflector peaks
ENDIF

CALL OBTAIN_MAGNITUDE_SPECTRA(BB1, ASPECB1)
CALL OBTAIN_MAGNITUDE_SPECTRA(BB2, ASPECB2)
CALL OBTAIN_MAGNITUDE_SPECTRA(RRS, ASPECRS)
CALL OBTAIN_MAGNITUDE_SPECTRA(RRNS, ASPECRNS)

ADELAYS = DELAYB2 - DELAYB1
ADELAYR = DELAYRNS - DELAYRS

TWOTAU = ADELAYS + (TIMESET / 51.2) * TWOTAU, Delay ! for time delay in sample
DELTAT = ADELAYR + (TIMESET / 51.2) * DELTAT, DELAY ! for delay between reflector peaks (sample vs. no sample)

CALL NOTHICK_VEL_CALC(DELTAT, TWOTAU, VEL, NOTHICK)
type * vel_nothick = vel_nothick

CALL FILTER(VEL, NOTHICK, VELPREV, FILTER, SHAPEANS, L_VEL, U_VEL, 1, AV_VEL, NING, NIP, SCAN_CNT, 118)
VEL(SCAN_CNT) = VEL, NOTHICK
VELPREV = VEL, NOTHICK

CALL STORE_SPECTRA(ASPECB1, ACTPOINT, 17, 17SUB) ! STORE SPECTRAS AT EACH SCAN POINT
CALL STORE_SPECTRA(ASPECB2, ACTPOINT, 17, 17SUB)
CALL STORE_SPECTRA(ASPECRS, ACTPOINT, 17, 17SUB)
CALL STORE_SPECTRA(ASPECRNS, ACTPOINT, 17, 17SUB)
IF (SHAPEANS.EQ.'N') SCANKNT=NPOINT
CALL STORE_VEL(I7SUB,185,SCANKNT,ACTPOINT,VEL)  ! STORE VELOCITIES AT END OF .SPC FILE
END

SUBROUTINE STORE_VEL(I7SUB,185,NPOINT,ACTPOINT,VEL)
!!! STORE VELOCITIES AND FIND MAX,MIN & STORE IN CAL FILE
INTEGER*2 NPOINT,185,ACTPOINT
INTEGER*4 I7SUB
REAL*4 VEL(5000)

VUP=0.0
VLO=10.**8.
VACCUM=0.

I7SUB=I7SUB+1
DO 94326 IJI=1,NPOINT
VMAX=AMAX1(VUP,VEL(IJI))
VUP=VMAX
VMIN=AMIN1(VLO,VEL(IJI))
VLO=VMIN
VACCUM=VACCUM+VEL(IJI)
WRITE(7,REC=I7SUB,FMT=53)VEL(IJI)
94326 I7SUB=I7SUB+1
53 FORMAT(A4)

VAVE=VACCUM/NPOINT
WRITE(8,REC=185,FMT=53)VUP  
185=185+1
WRITE(8,REC=185,FMT=53)VLO  
185=185+1
WRITE(8,REC=185,FMT=53)VAVE

RETURN
END

SUBROUTINE POINT_FROM_FREQ(FREQ,TIMEPERDIV,ACTPOINT)
!!! DETERMINE SPECTRA POINT FROM FROM SPECIFIC FREQUENCY IN SPECTRA
SEE SUBROUTINE CENTERFREQ FOR SIMILAR PROCESSING EXPLANATION
INTEGER*4 DUMMYFREQ
INTEGER*2 FREQ, ACTPOINT
REAL*4 TIMEPERDIV, DEL_FREQ, TEMPOINT

C
C
DUMMY_FREQ=FREQ*1E+06
DEL_FREQ=(1./2.*10.*TIMEPERDIV(A))
TEMPOINT=DUMMY_FREQ/DEL_FREQ
ACTPOINT=INT(TEMPOINT)

C
RETURN
END

SUBROUTINE OBTAIN_MAGNITUDE_SPECTRA(TWAVE, DUMMY_ASPEC)
COMPLEX C1(1024), CSPEC(1024)
INTEGER*4 ISTAT
REAL*4 DUMMY_ASPEC(1024), PHASE(1024)

DO 500 I=1, 512
C(I)=CMPLX(TWAVE(I))
CALL LSPSFFT_COMPLEX(C1, CSPEC, 1024, ISTAT)
CALL LSPSPS_PHASE_ANGLE(CSPEC, PHASE, DUMMY_ASPEC, 1024)

RETURN
END

SUBROUTINE STORE_SPECTRA(DUMMY_ASPEC, ACTPOINT, I7, I7SUB)
INTEGER*2 ACTPOINT
INTEGER*4 REC, I7SUB
REAL*4 DUMMY_ASPEC(1024)

DO 94326 IJI=1, ACTPOINT
I7SUB=((I7-1)*ACTPOINT)+IJI
WRITE(7, REC-I7SUB, FMT=53) DUMMY_ASPEC(IJI)
I7=I7+1

FORMAT(A4)

RETURN
END

SUBROUTINE FILTER(VEL_NOTHICK, VELPREV, CFILTER, SHAPEANS, L_VEL, U_VEL, AV_VEL, NIPG, NIPJ, SCANKNT, I18)
CHARACTER CFILTER*, CRUNCH_CODE*, SHAPEANS*
INTEGER NIPG, NIPJ, I18, SCANKNT
REAL*4 L_VEL, U_VEL, AV_VEL

C ********** TEST FOR VELOCITY OUTSIDE FILTER LIMITS ***** C
IF (SCANKNT.EQ.1)*VELPREV=AV_VEL
IF (FILTER.EQ.'Y.AND.(VEL_NOTHICK.LT.L_VEL.OR.VEL_NOTHICK.GT.U_VEL)) THEN
VEL_NOTHICK=VELPREV
CRUNCH_CODE='BC'

END
ICRUNCH_CODE_COUNT=1
TYPE *,'BAD POINT AT .NIPG,NIPJ'
ENDIF

IF (ICRUNCH_CODE_COUNT.EQ.0)CRUNCH_CODE='G'
WRITE(76,REC=I18,FMT=53)CRUNCH_CODE
I18=I18+1
WRITE(76,REC=I18,FMT=53)NIPG
I18=I18+1
WRITE(76,REC=I18,FMT=53)NIPJ
I18=I18+1

53 FORMAT(A4)
RETURN
END

SUBROUTINE NOTHICK_VEL_CALC(DELTAT,TWOTAU,VEL_NOTHICK)

VELWATER=0.149 !CM/USEC

VEL_NOTHICK=VELWATER*((DELTAT/TWOTAU)+1)
RETURN
END

SUBROUTINE CORR(B1,B2,CXX)

COMPLEX*8 NC11(512),NC12(512),NC13(512)
COMPLEX*8 NC14(512),NC15(512),NC22(512)
REAL*4 NC16(512),B1(512),B2(512),CXX
INTEGER*4 STATUS

DO 333 I=1,512
NC11(I)=CMPLX(B1(I))
NC22(I)=CMPLX(B2(I))
333 CONTINUE

CALL LSPPSFFT.COMPLEX(NC11,NC12,512,0,STATUS)
CALL LSPPSFFT.COMPLEX(NC14,NC15,512,0,STATUS)

DO 777 IO=1,512
NC14(IO)=NC12(IO)*CONJG(NC13(IO))
777 CONTINUE

CALL LSPPSFFT.COMPLEX(NC14,NC15,512,0,STATUS)
IF (.NOT. STATUS) CALL LIB$STGNAI (%VAL(STATUS))

AMAX=0.0
CXX=0.0

DO 765 I=1,256
NC16(I+256)=REAL(NC15(I))
NC16(I)=REAL(NC15(I+256))
765 CONTINUE
SUBROUTINE MCORR(B1, B2, CXX)

!! modified by using absolute value
!! of minimum of correlation function
!! to take into account phase inversion
!! of B2 w/ respect to B1 such as what happens w/ PMCs

COMPLEX*8 NC11(512), NC12(512), NC13(512)
COMPLEX*8 NC14(512), NC15(512), NC22(512)
REAL*4 NC16(512), B1(512), B2(512), CXX
INTEGER*4 STATUS

DO 333 I=1,512
NC11(I)=CMPLX(B1(I))
NC22(I)=CMPLX(B2(I))
CONTINUE

CALL LSP$FFT-COMPLEX(NC11, NC12, 512, 0, STATUS)
CALL LSP$FFT-COMPLEX(NC22, NC13, 512, 0, STATUS)

DO 777 IO=1,512
NC14(IO)=NC12(IO)*CONJC2(NC13(IO))
CALL LSP$FFT-COMPLEX(NC14, NC15, 512, 0, STATUS)

IF (.NOT STATUS) CALL LIB$SIGNAL(IVAL(STATUS))

AMAX=0.0
AMIN=1.0E+6
CXX=0.0
DXX=0.0

DO 765 I=1,256
NC16(I+256)=REAL(NC15(I))
NC16(I)=REAL(NC15(I+256))
CONTINUE

DO 881 I=1,512

IF(NC16(I).GT.AMAX) THEN
AMAX=NC16(I)
ENDIF
CXX=CXX-257.
RETURN
END
to take into account phase inversion of $H_2$ w/ respect to $B_1$ such as what happens w/ PMCs

```plaintext
IF(NC16(I).LT.AMIN) THEN
  AMIN=NC16(I)
  DXX=I
ENDIF

CONTINUE

CXX=DXX

CXX=CXX-257.
RETURN
END
```
PROGRAM: NOTHICK_IMAGEMAKER.FOR

Read Data file written by CRUNCH and place data into
"Grinnell-ready" files

Header: X3, Y3 Dimension of scan (SCAN DIRECTION, INDEX DIRECTION)
NI, N2 Number of points (SCAN, INDEX)

INTEGER ACTPOINT
BYTE BFLEEXT(23)
CHARACTER FILENAME*44,FILEEXT*23,DRIVE*1
CHARACTER CALNAME*44,SELFREQANS*1,SHAPEANS*1,DISKT*5
CHARACTER HEAD*23,DIR*16
EQUIVALENCE (FILEEXT, BFLEEXT)

JMARK=0
RS=1

OPEN (UNIT=10,FILE='NOTHICK_DATA1.DAT',STATUS='OLD',
FORM='FORMATTED')
READ (10,34572) FILEEXT
READ (10,34572) DRIVE
34572 FORMAT (A)
4321 CONTINUE

HEAD=FILEEXT

IF (DRIVE.EQ.'A') DIR='DUC2:\ROTH.IMAG'
IF (DRIVE.EQ.'A') DISKT='DUC2:
IF (DRIVE.EQ.'C') DIR='DUC0:\ROTH.IMAG'
IF (DRIVE.EQ.'C') DISKT='DUC0:

ICH=0
DO I=1,LEN(FILEEXT)
IF (ICH.EQ.0) GOTO 98564
IF (BFLEEXT(I).EQ.52) THEN
ICH=I
FILEEXT(ICH:ICH+4)=.'SPC'
ENDIF
98564 END DO

FILENAME = 'ROTH.DAT1'//FILEEXT

K=0
20 K=K+1
IF (FILEEXT(K:K).NE."") GOTO 20
FILEEXT(K+1,K+13)=.'CAL'

CALNAME = 'ROTH.DATA1'//FILEEXT

CALL DONDISPLS (ICH,FILENAME,CALNAME,HEAD,DIR,DISKT,
SHAPEANS,SELFREQANS,ACTPOINT)
SUBROUTINE DONDISP1S (ICHA,FILENAME,CALNAME,HEAD,DIR,DISK,T,
SHAPEANS,SELFREQANS,ACTPOINT)

CLOSE(10)
END

Read Data file written by CRUNCH and write values to Grinnell
- UNCORRECTED & CORRECTED DATA IS DISPLAYED---

Header:  X3, Y3 Dimension of scan (SCAN, INDEX)
N1, N2  Number of points (SCAN, INDEX)

INTEGER*4 REC,SCANPOS,XXX
INTEGER*2 IXLENGTH_A,IXLENGTH_B,ACTPOINT
INTEGER*2 X3,Y3,Y87,N1,N2,NNNNN2,NNNNNN1
INTEGER*2 DENS,NFREQ,AVES,X4,Y4,J87
INTEGER*2 COUNT,NCOUNT,NSCAN,NINDEX
INTEGER*2 C1(1024),D(460)
INTEGER*2 SCANDIST,INDEXDIST,TRFREQ,ENDFREQ,16
INTEGER*2 D Shape(460),189,16,115
INTEGER*2 ISLANT1,ISLANT2,ISLANT3,ISLANT4,JXXX,JABC,JABC,SCANNT
INTEGER*2 PSEUDOGRNL(460,1024),SELFREQ
REAL  FREQ, MEAN
REAL  RNUP(7),RNLO(7),CBUFFER(81)
BYTE  BSCALEMARK(4),CHFREQ(3)
BYTE  BSUFFREQ
BYTE  BFILENAME(44),BNAMELAB(20),BDATA-STATUS(40)
CHARACTER FILENAMEC44,FILEEXT*23,SELFREQANS*
CHARACTER CALNAME*44,CHHEADER*32,SHAPEANS*1,ZIGANS*
CHARACTER NAMELAB*20,DATA_STATUS*40,NEWHEADER*35,ACTION_SELFREQANS*1
CHARACTER CORNAME*15,CHFREQ_C*,PHASE*1,PHASE1*
CHARACTER SCALEMARK*4,SUFFREQ*4,HEAD*23,DIR*16,DISK*5

EQUIVALENCE ( CHFREQ, CHFREQ, C )
EQUIVALENCE ( BNAMELAB, NAMELAB )
EQUIVALENCE ( BDATA_STATUS, DATA STATUS )
EQUIVALENCE ( BSCALEMARK, SCALEMARK )
EQUIVALENCE ( BSUFFREQ, SUFFREQ )

SCALEMARK="1 MM"
SUFFREQ=" MHZ"

TAKE CHARACTER "FILENAME" AND DECODE TO BYTE "BFILENAME"

DECODE (44,533,FILENAME) BFILENAME

533 FORMATT(44A1)
DETERMINE SHORTENED (-.SIT) FILENAME POSITION

CONIINUE

OPEN( unit=15, file=FILENAME(1:IEND-1)/:dat2", status='OLD',
+ ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=2,
+ form='FORMATTED',ORGANIZATION='SEQUENTIAL' )

OPEN( unit=16, file=FILENAME(1:IEND-1)/:dat4", status='OLD',
+ ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=4,
+ form='FORMATTED',ORGANIZATION='SEQUENTIAL' )

OPEN( unit=66, file=FILENAME(1:IEND-1)/:dat3", status='OLD',
+ ACCESS='DIRECT',RECORDTYPE='FIXED',RECL=32,
+ form='FORMATTED',ORGANIZATION='SEQUENTIAL' )

***********UNIT=7, SPC FILE FOR SPECTRA, vel, FOR DIRECT ACCESS********C

C OPEN( unit=7, file=FILENAME, status='OLD',ACCESS='DIRECT',
C RECORDTYPE='FIXED',RECL=4,FORM='FORMATTED',
C ORGANIZATION='SEQUENTIAL' )

OPEN (UNIT=14,FILE='NOTHICK_SELFREQ1.DAT',
1 STATUS='OLD,FORM='FORMATTED')

READ(66,REC=16,FMT=51)CHEDER
16=16+1
READ(66,REC=16,FMT=51)SHAPEANS
16=16+1
READ(66,REC=I6,FMT=51)ZIGANS
I6=I6+1

READ (14,98652) FILEEXT
READ (14,98652) DRIVE
READ (14,98652) PHASE
READ (14,98652) PHASE1
READ (14,98652) DIFFANS
READ (14,10030) BUFSIZE
READ (14,10030) TRDIAM
CLOSE (14)

98652 FORMAT (A)
10020 FORMAT (I)
10030 FORMAT (F)

89714 READ(15,REC=I15,FMT=32)SCANDIST
I15=I15+1
READ(15,REC=I15,FMT=32)INDEXDIST
I15=I15+1
READ(15,REC=I15,FMT=32)NSCAN
I15=I15+1
READ(15,REC=I15,FMT=32)NINDEX
I15=I15+1
READ(15,REC=I15,FMT=32)SLANT1
I15=I15+1
READ(15,REC=I15,FMT=32)SLANT2
I15=I15+1
READ(15,REC=I15,FMT=32)SLANT3
I15=I15+1
READ(15,REC=I15,FMT=32)SLANT4
I15=I15+1
READ(15,REC=I15,FMT=32)TRFREQ
I15=I15+1
READ(15,REC=I15,FMT=32)AVES
I15=I15+1
READ(15,REC=I15,FMT=32)XXX
I15=I15+1

C ARRAYSIZE=(NSCAN)*(JXXX) IF SCAN FINISHED PROPERLY,JXXX=NINDEX

ENCODE(3,19878,CHFREQ) TRFREQ (ANALYSIS FREQUENCY
19878 FORMAT (I3)

READ(16,REC=I16,FMT=73)TIMESET
I16=I16+1

ENDFREQ=2.5*TRFREQ
CALL POINT_FROM_FREQ(ENDFREQ,TIMESET,ACTPOINT)
DO 9999 ICHOICE=1,1
187=1
ACCUM=0.
NBAD=0
IF (ICHICE.EQ.1) THEN
NAMELAB="VELOCITY CM/S"
CORNAME="_VEL_C_"/CHFREQ_C
NEWHEADER="CORRECTED DATA FOR VELOCITY"
NEWU=25
NEWU1=45
NEWU2=99
ENDIF
OPEN (UNIT=NEWU,FILE=DISK7//FILENAME(1:END-1)//CORNAME//
'\CORHEADER',STATUS='NEW',FORM='UNFORMATTED')

IF (JMARK.EQ.1) GOTO 90 !FILES ARE ALREADY OPEN
30 CONTINUE
90 CONTINUE
IF (JMARK.EQ.1) GOTO 89306
10149 CONTINUE

IF (SHAPEANS.NE..'Y')GOTO 89317
OPEN(UNIT=52,FILE=FILENAME(1:CHA+10)/'\DATHS',
STATUS='OLD',FORM='UNFORMATTED') INFO ON WHETHER WE ARE ON HOLDER/SAMPLE

89317 SCANDIST=ABS(SCANDIST)
AD=NSCAN
XREAL = REAL(ABS(SCANDIST))
YREAL = REAL(INDXDIST)
NPTS = NSCAN*JXXX IF SCAN COMPLETED PROPERLY, JXXX=INDEX
IF (SHAPEANS.EQ..'Y')THEN
DO 5697 I=1,NPTS
5697 READ(UNIT=52, END=89318)ABC,JABC,FFLAG,SCANKNT
ENDIF
89318 PTS = REAL(NSCAN)*REAL(JXXX)
X3=ABS(SCANDIST)
Y3=INDEXDIST
N1=NSCAN
N2=JXXX
IF (L.EQ.1) THEN
N1NNNN1=N1
N1NNNN2=N2
ENDIF
XREAL = REAL(ABS(X3))
YREAL = REAL(Y3)
FREQ = REAL(NFREQ)
PTS = REAL(N1)*REAL(N2)

C Proportion GRINNELL window similar to scan dimensions
C

200 YS7 = ININT(REAL(Y3)*8./7.)
IF (YS7.GE.X3) THEN
   R1 = FLOAT(X3)/FLOAT(Y3)
   X4 = 400
   Y4 = ININT(400./R1)
   IF (Y4.GE.480) THEN
      X4=300
      Y4 = ININT(300./R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=200
      Y4 = ININT(200./R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=100
      Y4 = ININT(100./R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=50
      Y4 = ININT(50./R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=25
      Y4 = ININT(25./R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=10
      Y4 = ININT(10./R1)
      ENDIF
   ELSEIF (Y87.LT.X3) THEN
      R1 = FLOAT(Y3)/FLOAT(X3)
      Y4 = IFIX( 400.*R1 )
   X4 = 400
   IF (Y4.GE.480) THEN
      X4=300
      Y4 = ININT(300.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=200
      Y4 = ININT(200.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=100
      Y4 = ININT(100.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=50
      Y4 = ININT(50.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=25
      Y4 = ININT(25.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=10
      Y4 = ININT(10.*R1)
      ENDIF
   ELSEIF (Y87.LT.X3) THEN
      R1 = FLOAT(Y3)/FLOAT(X3)
      Y4 = IFIX( 400.*R1 )
   X4 = 400
   IF (Y4.GE.480) THEN
      X4=300
      Y4 = ININT(300.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=200
      Y4 = ININT(200.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=100
      Y4 = ININT(100.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=50
      Y4 = ININT(50.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=25
      Y4 = ININT(25.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=10
      Y4 = ININT(10.*R1)
      ENDIF
   ELSEIF (Y87.LT.X3) THEN
      R1 = FLOAT(Y3)/FLOAT(X3)
      Y4 = IFIX( 400.*R1 )
   X4 = 400
   IF (Y4.GE.480) THEN
      X4=300
      Y4 = ININT(300.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=200
      Y4 = ININT(200.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=100
      Y4 = ININT(100.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=50
      Y4 = ININT(50.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=25
      Y4 = ININT(25.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=10
      Y4 = ININT(10.*R1)
      ENDIF
   ELSEIF (Y87.LT.X3) THEN
      R1 = FLOAT(Y3)/FLOAT(X3)
      Y4 = IFIX( 400.*R1 )
   X4 = 400
   IF (Y4.GE.480) THEN
      X4=300
      Y4 = ININT(300.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=200
      Y4 = ININT(200.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=100
      Y4 = ININT(100.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=50
      Y4 = ININT(50.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=25
      Y4 = ININT(25.*R1)
      ENDIF
   IF (Y4.GE.480) THEN
      X4=10
      Y4 = ININT(10.*R1)
      ENDIF
X4=50  
Y4 = INT(N0.*R1)  
ENDIF  
IF (Y4.GE.480) THEN  
X4=25  
Y4 = INT(N0.*R1)  
ENDIF  
IF (Y4.GE.480) THEN  
X4=10  
Y4 = INT(N0.*R1)  
ENDIF  
CC  
XD2G = REAL(N1)/REAL(X4)  
YD2G = REAL(N2)/REAL(Y4)  
XD2X = REAL(X4)/REAL(N1)  
YD2Y = REAL(Y4)/REAL(N2)  
I_INT=JINT(XD2X)+1  
J_INT=JINT(YD2Y)+1  
CONTINUE  
54322 FORMAT (A)  
4319 FORMAT (*)(VALUE=,F)  
TYPE *, '  
TYPE *, 'Working...';  
43191 FORMAT (11,1,BAD PTS/TOT. PTS<:15,7,J4, +  ' CURRENT VALUE=",F11.2)  
7493 FORMAT (F11.2)  
7494 FORMAT (I)  
C  
C  
RETURN MAX/MIN OF DATA FOR GRINNELL DISPLAY & PRODUCE I, N.  
C OF GRINNELL DATA (CHUFFER)  
CALL RETMAXMIN_FILTER (FILENAME,CORNAME,CALNAME,NAMELAB,DISKIT,END,JXXX,  
NSCAN,SCANLN,SHAPEANS,ZIGANS,CHOICE,TIMESET,189,COUNT,ACTPOINT,NEWU,GRBOT,  
1, GRTOP,MEAN)  
7487 COUNT=NCOUNT  
RMIN=GRBOT  
RMAX=GRTOP  
187=1
RNLO(CHOICE)=RMIN
RNUP(CHOICE)=RMAX

DO 500 KI=1,NSCAN
DO 4972 JI=1,NSCAN
NIPI=JI
NIPJ=KI
SCANPOS=((NIPI-1)*NSCAN)+NIPI
I87=SCANPOS
READ (NEWU,REC=I87,FMT=S) CBUFFER(JI)
4972 I 187=187+1
462 FORMAT(F11.2)
CALL SCGREAL (CBUFFER, C, NI, GRBOT, GRTOP, MEAN) !GREY SCALED
DATA-STATUS='CORRECTED DATA
NAMELAB
INTERPOLATE DATA GRID (EX.81X81) FOR X4 x Y4 GRINNELL DISPLAY
CALL INTR (C,D,NSCAN,X4 )
C
C IF (KI.EQ.20)TYPE *,D=,'D
C
C ********** MOD TO GET RID OF FALSE BORDER ********** C
IF (SHAPEANS,EQ,'Y')THEN
IIXCOUNT=0
DO 555 IOP=1,X4 !FORWARD X-DIRECTION
IF (D(IOP).EQ.)GOTO 555
IIXCOUNT=IIXCOUNT+1
IF (IIXCOUNT.EQ.1)THEN
DO 655 IQ=1,1_INT
655 D(IOP+IQ-1)=I
ENDIF
555 CONTINUE

IIXCOUNT=0
DO 755 IOP=1,X4,-1 !BACKWARD X-DIRECTION
IF (D(IOP).EQ.)GOTO 755
IIXCOUNT=IIXCOUNT+1
IF (IIXCOUNT.EQ.1)THEN
DO 855 IQ=1,1_INT
855 D(IOP-IQ+1)=I
ENDIF
755 CONTINUE
C
IF (KI.EQ.20)TYPE *,D=,'D
ENDIF
C
C ********** MOD TO GET RID OF FALSE BORDER ********** C
DO 480 J=1,X4
480 PSEUGOKNL(J,NPJ ) = D(J )
500 CONTINUE

48
OPEN (UNIT=NEWU2, FILE=DIR/HEAD/CORNAME/.PDS, STATUS="NEW", FORM="UNFORMATTED")
WRITE(NEWU2) X4, Y4, N1, N2, ICH, X4, Y4, N1, N2, ICH
WRITE(NEWU2) SCANDIST, TRFREQ
WRITE(NEWU2) RMIN, RMAX, MEAN
TYPE *, X4, Y4, N1, N2, ICH
TYPE *, SCANDIST, TRFREQ
TYPE *, RMIN, RMAX, MEAN

ILENGTH=0
605 DO 800 I=1, X4
   IX = I-1
   DO 540 J=1, N2
540   C(J) = PSEUDORNOL(I, J)
   CALL INTR(C, D, JXXX, Y4)
C IF (I.EQ.200) TYPE *, D=;

C ********** MOD TO GET RID OF FALSE BORDER ********** C
IF (SHAPEANS.EQ.Y) THEN
   IIXCOUNT=0
   DO 1555 IOP=1, Y4, 1; FORWARD Y-DIRECTION
      IF (D(IOP).EQ.1) GOTO 1555
      IIXCOUNT=IIXCOUNT+1
      IF (IIXCOUNT.EQ.1) THEN
         DO 1655 I=IOP-IJQ+1, IJQ-1, -1
      1655     D(IOP+IJQ-1)=1
         ENDIF
955   CONTINUE
   IIXCOUNT=0
   DO 1755 IOP=Y4, 1, -1; BACKWARD Y-DIRECTION
      IF (D(IOP).EQ.1) GOTO 1755
      IIXCOUNT=IIXCOUNT+1
      IF (IIXCOUNT.EQ.1) THEN
         DO 1855 I=IOP-JQ+1, JQ-1, 1
      1855     D(IOP-JQ+1)=1
         ENDIF
955   CONTINUE
C IF (I.EQ.200) TYPE *, D=;
ENDIF
C ********** MOD TO GET RID OF FALSE BORDER ********** C
C***************MODIFICATION FOR COMPLEX SHAPE **************C

IF (SHAPEANS.NE.'Y')GOTO 675 ! NOT A COMPLEX SHAPE
IF (SHAPEANS.EQ.'Y')THEN

XXX=1
IXSTART=U
IYLENGTH=0
NKEEP=0
DO 8970 J=1,Y4
IF (D(J).LT.2)GOTO 8970 !1 IS CODED FOR LUCITE (IN SCGREAL)
IF (D(J).GE.2)THEN !SAMPLE DATA
IF (NKEEP.EQ.0)IXSTART=J
NKEEP=1
C**** FURTHER MOD FOR GEORGE'S DISCONTINUITY ON YBCO HEX SAMPLE *****
IF (IYLENGTH.GT.1.AND.D(J).GE.2.AND.D(J-1).LT.2)GOTO 8875
ENDIF
C**** FURTHER MOD FOR GEORGE'S DISCONTINUITY ON YBCO HEX SAMPLE *****
8875 XXX=XXX+1
IYLENGTH=IYLENGTH+1
D SHAPE(IYLENGTH)=1
IF (D(J-XXX).LT.?)GOTO 8875
C**** FURTHER MOD FOR GEORGE'S DISCONTINUITY ON YBCO HEX SAMPLE *****
IYLENGTH=IYLENGTH+1
D SHAPE(IYLENGTH)=D(J)
ENDIF
C******* FURTHER MOD FOR GEORGE'S DISCONTINUITY ON YBCO HEX SAMPLE *****
CONTINUE
C TYPE *,'IXSTART='IXSTART,'JSTART='JSTART,'IYLENGTH='IYLENGTH,
C 1 IYLENGTH
IF (IYLENGTH.EQ.0)GOTO 800 ! ALL LUCITE IN THIS COLUMN
IXLENGTH=IXLENGTH+1
WRITE(NEWU2)IXSTART,JSTART,IYLENGTH,
DO 8979 UK=1,IYLENGTH
WRITE(NEWU2)D SHAPE(UK)
GOTO 800
ENDIF
C******************************************************************************C

675 WRITE(NEWU2) D

800 CONTINUE

TYPE *:
TYPE **: 'HIGHEST',NAMELAB,'IS ','RMAX
TYPE **: 'LOWEST',NAMELAB,'IS ','RMIN
TYPE **:
WRITE (NEWU1)NEWHEADER
WRITE (NEWU1)RMIN,RMAX
WRITE (NEWU1)MEAN
WRITE (NEWU1)ISCALEF
SUBROUTINE SCGREAL( CC, DD, N, BOT, TOP, MEAN )

C Convert values in CC to 1 thru 255 (same as SCGR, with CC REAL)

INTEGER*2 N
INTEGER*2 DD(N)
REAL CC(N), MEAN

RAN = TOP - BOT
IF ( RAN.EQ.0. ) RAN = 1.
DO 2000 J = 1, N
CO = CC(J)
Q = ( CO - BOT ) / RAN
IF ( Q.LE.0. ) Q = 0.
IF ( Q.GT.1. ) Q = 1.

CGRNLBASE.FOR
C
C - A set of subroutines useful for applications
C with the GRINNELL image processor.
C GRINIT must be called in main program before calling these
C subroutines.
C
David B. Stang latest update 19-APR-88
Q = Q^251.

C DD(J) = NINT( Q ) + 1

C FOR MY MODIFICATION ************

IF (CO.EQ.0.) THEN
  DD(J) = 1
GOTO 2000
ELSEIF (CO.NE.0.) THEN
  DD(J) = NINT(Q) + 2
ENDIF

C CONTINUE

RETURN

END

SUBROUTINE INTR

INTERPOLATE VALUES IN C(N2) TO "FIT" INTO D(NN)

INTEGER*2 N2, NN
INTEGER*2 CC(N2), DD(NN)

IR = IFIX(K)
IR = INT(R)
IRP1 = IR + 1
IF (IRP1.GT.W) THEN
  CO = FLOAT(CC(1R))
  DO = (1. - RR) * CD + RR * CI
D 3000 DD(K) = INT(D0)
C 3000 DD(K) = INT(D0)
C RETURN
C END

SUBROUTINE RETMAXMIN_FILTER(FILENAME, CORNAME, CALNAME, NAMELAB, DISK2, EEND, JXX, NSCAN, SCANKNT, SHAPEANS, ZIGANS, ICHOICE, TIMESET, 189, COUNT, ACTPOINT, NEWU, ORBOT, GKTOP, MEAN)

INTEGER*4 REC, SCANPOS, ID7
INTEGER*2 NSCAN, ACTPOINT
INTEGER*2 IS5,IR7,SCANKNT
INTEGER*2 IABC,JABC,JXXX,IB9
INTEGER*2 NCOUNT
INTEGER*2 ZIGCOUNT
INTEGER*2 NBAD,COUNT,LCOUNTER
REAL DATATHING(1024,9)
REAL MEAN,VEL(5000)
REAL CBUFFER(81)
CHARACTER FILENAME*44,CORNAME*15,NAMELAB*20,FFLAG*1
CHARACTER SHAPEANS*1,ZIGANS*1,DISKT*5,CALNAME*4
INTEGER*4 SCANPOSI

TYPE *,NAMELAB
LCOUNTER=1
ZIGCOUNT=1
ACCUM=0.
SCANPOS=0
SCANPOS1=0
NBAD-COUNT

XMIN=0.
XMAX=0.

C CAL. FILE WITH MAX/MIN/MED MEAN
C
OPEN(unit=8, file=CALNAME, status='OLD', ACCESS='DIRECT',
+ RECORD=TYPE='FIXED', RECL=4, FORM='FORMATTED',
+ ORG=ANIZATION='SEQUENTIAL')

read(8,REC=185,FMT=53)VUP
185=185+1
read(8,REC=185,FMT=53)VLO
185=185+1
read(8,REC=185,FMT=53)VAVE

IF (SHAPEANS.EQ.'Y') THEN
NPTS=SCANKNT
REWIND(52)
ENDIF

17=NPTS*4*ACTPOINT-1 !START OF VELOCITY READS IN SPC FILE
C
C SPC FILE (UNIT=7) CONTAINS ALL VELOCITY & SPECTRA DATA
C
OPEN (UNIT=7, FILE=DISKT//FILENAME, STATUS='OLD', ACCESS='DIRECT',
+ RECORD=TYPE='FIXED', RECL=4, FORM='FORMATTED',
+ ORGANIZATION='SEQUENTIAL')

OPEN (UNIT=NEWU, FILE=DISKT//FILENAME(1:END-1)/CORNAME//COR'
DO 500 NIPJ=1,JXXX
DO 450 NIPI=I,NSCAN
C***** GET INFO FROM HOLDER.DAT (FFLAG='S' -> SAMPLE) (FFLAG='H' -> HOLDER)
KCON=KCON+I
J=NIPI
SCANPOS=((NIPJ-I)*NSCAN)+NIPI ! TRUE SCAN POSITION
IF (SHAPEANS.NE."Y")GOTO 87123
READ(32)IABC,JABC,FFLAG,SCANKNT
C TYPE *,XH=IABC, YH=JABC, FFLAG=FFLAG
IF (FFLAG.NE."S")GOTO 8702
! TRANSDUCER IS ON HOLDER GOTO NEXT PT.
87123 SCANPOS=SCANPOS+1
CC ! SCANPOS FOR FILE RETRIEVAL FOR CIRCLE DISK SCANS
READ(7,REC=I7,FMT=53)VEL(SCANPOS)
17=17+1
DATATHING(J,5)=VEL(SCANPOS)

NPTS=(NIPJ-I)*NSCAN+NIPI
IF (NPTS.NE.NSCAN*JXXX)NPTS=NSCAN*JXXX

C IF (ICHOICE.EQ.1) THEN
IF (SCANPOS.EQ.1) DATATHINGPREVIOUS=VAVE
IF (DATATHING(J,3).LT.VLO+VLO*0.01) OR.
1 DATATHING(J,5).GT.(VUP-VUP*0.01)/DATATHING(J,5)=DATATHINGPREVIOUS
ENDIF
IF (SHAPEANS.NE."Y")GOTO 8765
8702 IF (FFLAG.NE."S") THEN SET TO 0 // TRANSDUCER IS ON HOLDER GOTO NEXT PT.
CBUFFER(J)=0.0
GOTO 450
ENDIF
87651 CBUFFER(J)=DATATHING(J,5)
390 CONTINUE
C****** KEEP TRACK OF MAX & MIN LIMITS OF CORRECTED DATA****** C
C **** FOR FFLAG.NE."S" -- WE SKIP THIS STEP SO THAT '0' IS NOT CONSIDERED
RMAX=AMAX((CBUFFER(J),XMAX)
RMIN=AMIN((CBUFFER(J),XMIN)
XMAX=RMAX
XMIN=RMIN
450 CONTINUE

IF (ZIGANS.EQ.'N') THEN ! NOT A ZIGZAG SCAN - A NORMAL SCAN
   DO 49720 JI=1,NSCAN
   WRITE (NEWU,REC='I87',FMT='(I5','CBUFFER(JI)
   ACCUM=ACCUM + CBUFFER(JI)
49720   I87='I87'+1
ELSEIF (ZIGANS.EQ.'Y',AND.ZIGCOUNT.EQ.1) THEN ! ZIGZAG LEFT->RIGHT
   DO 49721 JI=1,NSCAN
   WRITE (NEWU,REC='I87',FMT='(I5','CBUFFER(JI)
   ACCUM=ACCUM + CBUFFER(JI)
C TYPE 'CBUFFER(JI)~CBUFFER(JI)
49721   I87='I87'+1
ELSEIF (ZIGANS.EQ.'Y'.AND.ZIGCOUNT.EQ.0) THEN ! ZIGZAG RIGHT->LEFT
   DO 49722 JI=NSCAN,1,-1
   WRITE (NEWU,REC='I87',FMT='(I5','CBUFFER(JI)
   ACCUM=ACCUM + CBUFFER(JI)
C TYPE 'CBUFFER(JI)=CBUFFER(JI)
49722   I87='I87'+1
ENDIF

C***** FOR ZIGZAG SCANS - TO ALTER ROW STORAGE *************C
IF (ZIGANS.EQ.'Y'.AND.ZIGCOUNT.EQ.0) THEN
   ZIGCOUNT=1
   GOTO 500
ELSEIF (ZIGANS.EQ.'Y'.AND.ZIGCOUNT.EQ.1) THEN
   ZIGCOUNT=0
   ENDIF
C500 TYPE 'Y=JABC','CBUFFER=CBUFFER
500 CONTINUE

501 GBOT = RMIN
     GRTOP = RMAX
     TYPE *,RMIN=GBOT
     TYPE *,RMAX=GRTOP
     MEAN = ACCUM/NSCANPOS
     TYPE *,MEAN=MEAN
     IF (SHAPEANS.NE.'Y') GOTO 88888
     REWIND(52)
88888 CONTINUE

53 FORMAT (A4)
43191 FORMAT ('+',BAD PTS =') AT LOCATION (SCANPOS) =')
RETURN

END
DETERMINE SPECTRA POINT FROM FROM SPECIFIC FREQUENCY IN SPECTRA
SEE SUBROUTINE CENTERFREQ FOR SIMILAR PROCESSING EXPLANATION

INTEGER*4 DUMMYFREQ
INTEGER*2 SELREQ,ACTPOINT
REAL*4 TIMEPERDIV,DELREQ,TEMPPOINT

DUMMYFREQ=SELREQ*E+06
DELREQ=(1/(2*(10.*TIMEPERDIV)))
TEMPPOINT=DUMMYFREQ/DELREQ
ACTPOINT=JNINT(TEMPPOINT)

RETURN
END

SUBROUTINE CENTERFREQ(DUMMYP,TIMEPERDIV,ACTFREQ)
DETERMINE FREQUENCY FROM POINT IN B1(F) SPECTRA

INTEGER*2 DUMMYP,ACTFREQ
REAL*4 TIMEPERDIV,DELREQ,TEMPFREQ

WE KNOW THAT DELFREQ=(1/(2N*DELTIME))
=(1/(2N*(10.*TIMEPERDIV/N)))
=(1/(2*(10.*TIMEPERDIV)))

WHERE N = 512 POINTS/ACQUISITION (SCREEN)
TIMEPERDIV = TIME SETTING PER DIVISION (EX. 50 NSEC)
2 = WE EXTENDED 512 ARRAY TO 1024 ARRAY FOR PROCESSING
10 = THE NUMBER OF DIVISIONS/SCREEN

FOR EXAMPLE:
WE CAN DETERMINE THE FREQUENCY FOR WHICH THE POINT "MAXBIP"
CORRESPONDS TO BY JUST MULTIPLYING (DELFREQ) * (MAXBIP)

DELFREQ=(1/(2*(10.*TIMEPERDIV)))
TEMPFREQ=DELFREQ*DUMMYP
TEMPFREQ=TEMPFREQ/1E+06
ACTFREQ=JNINT(TEMPFREQ)

RETURN
END

SUBROUTINE INTR (F1, F2, N, M)
IMPLICIT NONE

INTEGER*2 I, M, N
INTEGER*2 FI(N), F2(M)
REAL*4 X, XA(512), Y, YA(512)

DO I = 1, N
   XA(I) = FLOAT(I)
END DO

DO I = 1, N
   YA(I) = FLOAT(FI(I))
END DO

DO I = 1, M
   X = FLOAT(I-1) * FLOAT(N-I) / FLOAT(M-I) + 1.0
   CALL LINEAR_INTERPOLATION (XA, YA, N, X, Y)
   F2(I) = INT(Y)
END DO

RETURN
END

LINEAR_INTERPOLATION computes the value of the piecewise linear spline interpolant of the points in the arrays XA and YA at the point X. This routine is a modified version of the FORTRAN routine SPLINT found in the book "Numerical Recipes: The Art of Scientific Computing" by Press, Flannery, Teukolsky, and Vetterling published by the Cambridge University Press, 1986.

Variables:

NMAX : A constant which determines the array sizes for U, X, and Y.

K : The midpoint between KHI and KLO, used to find the two indices into the XA table such that XA(KLO) < X < XA(KHI).

KHI : The greater of the two indices into the XA array such that XA(KLO) < X < XA(KHI).

KLO : The lesser of the two indices into the XA array such that XA(KLO) < X < XA(KHI).

N : The index of the arrays U and V that contain interpolation points.

A : Weight given to YA(KLO).

B : Weight given to YA(KHI).

H : The difference of XA(KHI) and XA(KLO).

X : The value of X at which to compute the value of the interpolation function.

XA : The table of X coordinates of the interpolation points.

YA : The table of Y coordinates of the interpolation points.

SUBROUTINE LINEAR_INTERPOLATION (XA, YA, N, X, Y)
IMPLICIT NONE

INTEGER NMAX
PARAMETER (NMAX=512)

INTEGER*2 K, KHI, KLO, N
REAL*4 A, B, H
REAL*4 X, XA(NMAX), Y, YA(NMAX)

KLO = 1
KHI = N
DO WHILE (KHI-KLO.GT.1)
  K = (KHI + KLO)/2
  IF (XA(K).GT.X) THEN
    KHI = K
  ELSE
    KLO = K
  END IF
END DO

H = XA(KHI) - XA(KLO)
IF (H.LT.0.0) THEN
  A = (XA(KHI) - X) / H
  B = (X - XA(KLO)) / H
  Y = A * YA(KLO) + B * YA(KHI)
END IF

RETURN

END
C ******** NOTHICK_DRAW.FOR  **********************

INTEGER*2 TRFREQ,JXXX,IXLENGTH
INTEGER*2 SCANDIST,INDXDIST,NSCAN,NIDX,DENS
INTEGER*2 SELFREQ,AVES,Il5,1SLANT1,1SLANT2,1SLANT3,1SLANT4
CHARACTER RESPI*1,SELFREQANS*1,RESP2*2
CHARACTER NEWOLD*3,SHAPEANS*1,ZIGANS*1,CHFREQ C*3,CHDR*1
BYTE  B, CHFREQ(6), BSUFFREQ(4)
BYTE  BSCALEMARK(4)
BYTE  BDATA_STATUS(40), BLABEL(30), FILEEXT(23)
CHARACTER LABEL*30., FILEEXT*23, CHEADER*32, TEBAR*1
CHARACTER NEWOLD='OLD'

16=1
79193 READ(66,REC=16,FMT=51)CHEADER
16=16+1
READ(66,REC=16,FMT=51)SHAPEANS
16=16+1
READ(66,REC=16,FMT=51)ZIGANS
16=16+1

READ (14,98652) FILEEXT
READ (14,98652) DRIVE
READ (14,98652) PHASE
READ (14,98652) PHASE1
READ (14,98652) DIFFANS
READ (14,10030) BUFLENGTH
READ (14,10030) BUFSIZE
CLOSE (14)
98652 FORMAT (A)
10000 FORMAT (F)

911652 FORMAT (A)
I0020 FORMAT (I)
10030 FORMAT (F)

115=1
89714 READ(15,REC=115,FMT=52)SCANDIST
115=115+1
READ(15,REC=115,FMT=52)INDXDIST
115=115+1
READ(15,REC=115,FMT=52)INSCAN
115=115+1
READ(15,REC=115,FMT=52)NINDEX
115=115+1
READ(15,REC=115,FMT=52)ISPAN1
115=115+1
READ(15,REC=115,FMT=52)ISPAN2
115=115+1
READ(15,REC=115,FMT=52)ISPAN3
115=115+1
READ(15,REC=115,FMT=52)ISPAN4
115=115+1
READ(15,REC=115,FMT=52)JXX
115=115+1
READ(15,REC=115,FMT=52)JXXLEN
115=115+1
READ(15,REC=115,FMT=52)TRFREQ
115=115+1
READ(15,REC=115,FMT=52)AVES
115=115+1
READ(15,REC=115,FMT=52)JXX
115=115+1
READ(15,REC=115,FMT=52)JXXLEN

C ARRAYSIZE=((NSCAN)*(JXXX)) IF SCAN FINISHED PROPERLY,JXXX=NINDEX

ENCEDE (3,19878,FIRQ) TRFREQ  ANALYSIS FREQUENCY
19878 FORMAT (F)

C DETERMINE SHORTENED (.DAT) FILEEXT POSITION
C KMARK=0
C DO 4321 IPR=1,34
C IF (KMARK.EQ.1) GOTO 4321
C IF (FILEEXT(1:IPR).EQ.46) THEN !PERIOD/DOT IS ASCII 46
C IEND=IPR
C KMARK=1
C ENDIF
C4321 CONTINUE
C
FILEEXT=FILEEXT(1:IEND-1)

TYPE *, 'CURRENT IMAGE FILE IS ',FILEEXT
TYPE *, 'IF YOU WANT A DIFFERENT IMAGE, YOU MUST '
TYPE *, 'RECALL AN OLDER "NOTHICK_DADQ1.DAT" FILE'
TYPE *, '
IF (DRIVE.EQ.'A')DIR='DUC2:[ROTH.IMAG]
IF (DRIVE.EQ.'C')DIR='DUCO:[ROTH.IMAG]

5321 TYPE *, 'CURRENT DRIVE TO READ IMAGES IS ',DIR(1:5)
IF (DIR(1:5).EQ.DUCO:)THEN
TYPE *, 'DO YOU WANT TO CHANGE TO DUC2: (Y/N ) ?'
READ (5,51)CHDR1
IF (CHDR1.NE.'Y' .AND. CHDR1.NE.'N')GOTO 5321
ELSEIF (DIR(1:5).EQ.DUC2:)THEN
TYPE *, 'DO YOU WANT TO CHANGE TO DUCO: (Y/N ) ?'
READ (5,51)CHDR1
IF (CHDR1.NE.'Y' .AND. CHDR1.NE.'N')GOTO 5321
ENDIF

211 TYPE *, 'DO YOU WANT TO SEE TEXT AND COLORBAR (Y/N)?'
READ (5,51)TEBAR
IF (TEBAR.NE.'Y' .AND. TEBAR.NE.'N')GOTO 211
CONTINUE
FILENAME='[ROTH.DATA]'IIFILEEXT
CALL GRINIT
CALL GRSRST

TYPE *, 'IMAGE FREQUENCY (MHz) = ',TRFREQ
ENCODE (3,9878,CHFREQ) TRFREQ
9878 FORMAT (I3)

234 TYPE *, '****** NOTHICKNESS VELOCITY IMAGE ******
TYPE *, '
WRITE (5,394)
394 FORMAT ('Y'; COLOR (C) OR B&W (B) IMAGE? ')
READ (5,2222) RESP1
IF (RESP1.NE.'C' .AND. RESP1.NE.'B') GOTO 234

IF (RESP1.EQ.'C') THEN
  TYPE * 'COLORRED (I<) OR NORMAL COLOR (C) SCHEME?'
  READ (5,2222) RESP2
ENDIF

CONTINUE
STOP
END

SUBROUTINE IMAGE(NEWOLD,FILEEXT,CHFREQ_C,RESP1,RESP2,SHAPEANS,IXLENGTH,
  FILENAME,DIR,TEBAR)

INTEGER*2 D(460),CHANNEL,INPUT,IXLENGTH
INTEGER*2 X4,Y4,N1,N2,TRFREQ,SCANDIST
CHARACTER*1 RESP1*1,SHAPEANS*1,RESP2*1,TEBAR
CHARACTER CHFREQ_C*5
REAL RMIN,RMAX,MEAN
BYTE B, CHFREQ(3), BSUFFREQ(4), BZERO(2)
BYTE BFFILENAME(34), BSCALEMARK(4)
BYTE BDATATAG(34), BLABEL(30)
CHARACTER FILNAM*48, LABEL*30, FILEEXT*23
CHARACTER*,*4, FILENAME*34, SCALEMARK*4, DATA_STATUS*40

EQUIVALENCE (L_B, BFILENAME)
EQUIVALENCE (BSSCALEMARK, SCALEMARK)
EQUIVALENCE (BSUFFREQ, SUFFREQ)
EQUIVALENCE (BDATATAG, DATA_STATUS)
EQUIVALENCE (BLABEL, LABEL)
EQUIVALENCE (BZERO, BZERO)

CZERO='0'
LLLL='L'
IIIMAGE='I'

CONTINUE

IF (IIIMAGE.EQ.1) FILNAM=' .VEL_C/CHFREQ_C/FDK'

CONTINUE
CHANNEL=0
IF (CHANNEL.NE.0 .AND. CHANNEL.NE.2 .AND. CHANNEL.NE.3) GOTO 501
IF (CHANNEL.EQ.0) THEN
  INPUT=0
MASK=1
ELSEIF (CHANNEL.EQ.2) THEN
INPUT=1
MASK=4
ELSEIF (CHANNEL.EQ.3) THEN
INPUT=2
MASK=8
ENDIF

CALL ERASE

19 OPEN (UNIT=6,FILE=DIR//FILEEXT//FILNAM,STATUS='OLD',
FORM='UNFORMATTED')

READ(6) X4,Y4,N1,N2,IVALUE
READ(6) SCANDIST,TRFREQ
READ(6) RMIN,RMAX,MEAN

IF (IVALUE.EQ.1 ) LABEL="VELOCITY CM/US"

ICOLORBAR=0
SUFFIX="MIZ"
SCALEMARK="1 MM"
DATA_STATUS=LABEL

CALL GRMN(1,INPUT,INPUT,INPUT)
CALL GRNBY(1,1,1,1)
CALL GRSBFD

C **********************MODIFICATION FOR COMPLEX SHAPES ******C
IF (SHAPEANS.NE.'Y')IX49=X4

IF (SHAPEANS.EQ.'Y')THEN
IX49=IXLENGTH
ENDIF

DO 100 I=I,IX49

ELSEIF (SHAPEANS.EQ.'Y')THEN
IF (SHAPEANS.NE.'Y')THEN
IXSTART=I-1
IYSTART=1
IYLENGTH=Y4
READ(6)D
GOTO 789
ELSEIF (SHAPEANS.EQ.'Y')THEN
IF (NE.I,N1,AND.IYLENGTH.EQ.1))GOTO 100
READ(6,END=100)IXSTART,IYSTART,IYLENGTH
D0 7897 IX=1,IYLENGTH

7897 READ(6)(D,IX)
789 CONTINUE

ELSEIF (SHAPEANS.EQ.'Y')THEN
IF (NE.I,N1,AND.IYLENGTH.EQ.1))GOTO 100
READ(6,END=100)IXSTART,IYSTART,IYLENGTH
D0 7897 IX=1,IYLENGTH

C **********************************************************C
CALL GRWAW(D,IXSTART,IYSTART,1,1,YLENGTH,0,0,0,0,1,1)
C **********************************************************C
CALL GRSBFD
100 CONTINUE
IF (RESP1.EQ.'B') THEN
IF (TEBAR.EQ.'N') GOTO 213
CALL PLACEBAR(0)
GOTO 8999
ELSEIF (RESP1.NE.'B') THEN
IF (RESP2.EQ.'C') THEN
CALL COLORS (IVALUE, ICOLORBAR)
ELSEIF (RESP2.EQ.'R') THEN
CALL COLORRED (IVALUE, ICOLORBAR)
ENDIF
ENDIF
IF (TEBAR.EQ.'W') GOTO 213
DECODE (34,8989,FILENAME) BFILENAME
FORMAT (34A1)
ENCODE (3,909,CHFREQ) TRFREQ
FORMAT (13)
CALL GRFCD (1,255,0,0,FILENAME,1,441,6,0,0,0,0,0)
CALL GRFCD (1,255,0,0,CHFREQ,200,441,6,0,0,0,0,0)
CALL GRFCD (1,255,0,0,VSUFFREQ,215,441,6,0,0,0,0,0)
CC PRINT LABEL TO GRINNELL
CC NCHAR=LEN(DATA-STATUS)
DO 765 IKY=1,NCHAR
IKX=400-(IKY-1)*13
CALL GRFCD (1,255,0,0,BDATA STATW(IKY),505,IKX,0,0,0,0,0,0)
CALL GRFCD (1,255,0,0,DATA STATUS(IKY),505,IKX,0,0,0,0,0,0)
CC CALCULATE MAX, MIN, AND MEAN
CC IF (IVALUE.EQ.1.OR.IVALUE.EQ.2.OR.IVALUE.EQ.6.OR.IVALUE.EQ.7) THEN
IF (IVALUE.EQ.1) THEN
IF (RMAX.EQ.0) RMAX=9999
ENDIF
RRMAX=RMAX**4.
RMIN=RMIN*10**4.
RMEAN=MEAN*10**4.
ELSEIF (IVALUE.EQ.3.OR.IVALUE.EQ.4) THEN
RRMAX=RMAX/100.
RRMIN=RMIN/100.
RMEAN=MEAN/100.
ELSEIF (IVALUE.EQ.5) THEN
  IF (RMAX.GE.100.) RMAX=99.9
  IF (RMIN.GE.100.) RMIN=99.9
  IF (MEAN.GE.100.) MEAN=99.9
  IF (RMAX.LT.10.) THEN
    RRMAX=RMAX*10.**4.
    RRMIN=RMIN*10.**4.
    RMEAN=MEAN*10.**4.
  ELSE
    RRMAX=RMAX*10.**3.
    RRMIN=RMIN*10.**3.
    RMEAN=MEAN*10.**3.
  ENDIF
ENDIF
ENDIF
ELSEIF (IVALUE.EQ.2.OR.IVALUE.EQ.6.OR.IVALUE.EQ.7) THEN
  JKMN=465
ELSEIF (IVALUE.EQ.1.OR.IVALUE.EQ.3.OR.IVALUE.EQ.4) THEN
  JKMN=470
ENDIF

CC DRAW DECIMAL POINT FOR MIN AND MAX TO GRINNELL
DO 9879 IUY=0,350,350
  CALL GRFAR (1,255,0,0,JKMN,IUY,1.1)
  CALL GRSBFD
9879 CONTINUE

IMEAN=JNINT(RMEAN)
IRMAX=JNINT(RRMAX)
IRMIN=JNINT(RRMIN)

TYPE *:' CORRECTED:
TYPE *,IRMAX=,IRMAX
TYPE *,IRMIN=,IRMIN
TYPE *,IMEAN=,IMEAN
TYPE **:

IF (IMEAN.GE.1.0*10.**5 ) GOTO 73920
ENCODR (5,9878,CHMEAN) IMEAN
73920 IF (IRMAX.GE.1.0*10.**5 ) GOTO 73921
ENCODR (5,9878,CHRMAX) IRMAX
73921 IF (IRMIN.GE.1.0*10.**5 ) GOTO 73922
ENCODR (5,9878,CHRMIN) IRMIN
73922 CONTINUE
9878 FORMAT (15)
IF (IMAGEN GE 1.0*10**6.) GOTO 93920 

ZZZZZ = RRMAX - RRMIN

IF (ZZZZZ EQ 0.) ZZZZZ = 1. 

SCALEF = ( (RRMAX - RRMIN) * ZZZZZ ) * 350. ! SC. MEAN TO CLR. BAR

SCALEF = ININT (SCALEF)

93920 CONTINUE

IF (IMAGEN GE 1.0*10**6.) GOTO 83920

CALL GRFDCS (1,255,CHMEAN,458,ISCALEF,6,0,4)

83920 IF (IMAXN GE 1.0*10**6.) GOTO 83921

CALL GRFDCS (1,255,CHRMAX,458,350,6,0,4)

83921 IF (IMINN GE 1.0*10**6.) GOTO 83922

CALL GRFDCS (1,255,CHRMIN,458,0,6,0,4)

CALL GRSBFD

83922 CONTINUE

9877 CONTINUE

IF (IVALUE EQ 2) THEN

JKMN = 465

ELSEIF (IVALUE EQ 1 OR IVALUE EQ 3 OR IVALUE EQ 4 OR IVALUE EQ 6

JKMN = 463

ELSEIF (IVALUE EQ 5) THEN

JKMN = 470

ENDIF

CALL GRFAR (1,255,0.0,JKMN,ISCALEF,1,1)

CALL GRSBFD

95631 CONTINUE

CC

CC DRAW SCALE ONTO IMAGE

CC

RSCANDIST = FLOATI(SCANDIST)

PIXELS = (X4*1) / (RSCANDIST)

PIXELS = ININT (PIXELS)

CALL GRFVCS (1,255,340,420,340+1PIXELS,420)

CALL GRFDCS (1,255,BSCALEMARK,345,410,6,0,4)

CALL GRSBFD

C! FOR DRAWING INITIAL ZEROS FOR ATTENUATION COEFF. < 0.1

IF (IVALUE EQ 2 AND IMINN LT 1000.) THEN

CALL GRFDCS (1,255,BZERO,JKMN,0,6,0,2)

CALL GRSBFD

ENDIF

213 CONTINUE

201 CLOSE (6)

1 FORMAT (1, WHICH IMAGE TO VIEW ['*.PDK IN ROTH.IMAG']?)

2 FORMAT (A)

3 FORMAT (3, 'Channel (0,2,3) ?')
SUBROUTINE COLORSUB (IVALUE, COLORBAR)
INTEGER NR(256), NB(256), NG(256), IARRAY(18)
DATA IARRAY / 0, 2, 1, 0, 1, 2, 1, 0, 2, 1, 2, 0, 2, 0, 1, 2, 1, 0 /!
IF (IVALUE.GT.10) GOTO 9876
CALL GRINIT_OLD
CALL GRSKST
CALL GRZFC(0,0)
CALL GRZCL(0,255,255)
CALL GRZON(0,1)
CALL GRSBFD
R=4
A=2
L=256
IL=L/B
JL=A*IL
DO 100 I=1, IL
   NB(I)=255
   NG(I)=B*I
   IF (NG(I).GT.255) THEN
      NG(I)=255
   ENDIF
100 CONTINUE
DO 200 I=IL+1, JL
   BB=((B*(1-A))*I+((A/A(1-A))*L)
   NG(I)=255
   NR(I)=0
   NB(I)=BB
   IF (NB(I).GT.255) THEN
      NR(I)=255
   ENDIF
200 CONTINUE
DO 300 I=I1+1, I1+1
RR=(B*I)-(A*L)
NG(I)=255
NR(I)=RR
IF (NR(I).GT.255) THEN
  NR(I)=255
ENDIF
CONTINUE

DO 400 I=I1+1, I1
GG=((-B)*I)+(B*L)
NR(I)=ZS5
NB(I)=O
NG(I)=GG
IF (NG(I).GT.255) THEN
  NG(I)=255
ENDIF
CONTINUE

KC=O
NR(I)=0
NG(I)=O
NB(I)=0
NR(256)=255
NG(256)=255
NB(256)=255

CALL GRNWWR(I,NB,IARRAY(1+KC),0,256,0)
CALL GRNWWR(I,NG,IARRAY(2+KC),0,256,0)
CALL GRNWWR(I,NR,IARRAY(3+KC),0,256,0)
CALL GRSBFD
CALL GRNIN(O,O,O,O)
IF (ICOLORBAR.EQ.1) THEN
  CALL GRNBK(O,O,O,O)
ELSEIF (IVALUE.EQ.1 .OR. IVALUE.EQ.3 .OR. IVALUE.EQ.5 .OR. IVALUE.EQ.7) THEN
  DO 111 I=1,300
  DO 111 J=1,900
  CONTINUE
  GOTO 950
ENDIF
950 KC=O
RETURN
END

SUBROUTINE PLACEBAR (ICOLORBAR)
CC ------Draw colorbar at right side of screen------
INTEGER*2 NA(45)
IF (ICOLORBAR.EQ.1) GOTO 1010
CONTINUE
DO 1000 I=0,350
DO 2000 J=1,19
G=I*(254.0/350.0)
NA(I)=G
2000 CONTINUE
CALL GRWLW(NA,410,1,19,0,0,1,1)
CALL GRSBFD
1000 CONTINUE
1010 CONTINUE
RETURN
END

SUBROUTINE SAVEBAR
INTEGER*2 BAR(6)

TYPE *, "Creating and saving the color bar...
TYPE *, "
OPEN(UNIT=34,NAME="BARFILE",STATUS="NEW", I
FORM=UNFORMATTED,BLOCKSIZE=1024)

CALL GRNIN(1,0,0,0)
CALL GRNBY(1,1,1,1)
CALL GRSBFD
DO 2060 IBBEL=0,459,3
CALL GRRLW (410,IBBEL,6,1,BAR)
WRITE(34) BAR
2060 CONTINUE
CALL GRSBFD
CLOSE(34)
RETURN
END

SUBROUTINE ERASE
M=4095
N=4095
CALL GRFER (M,N,0)
CALL GRSBFD
END

SUBROUTINE COLORS (VALUE,ICOLORBAR)
INTEGER*2 NR(256),NB(256),NG(256),IARRAY(18)
DATA IARRAY/0,2,1,0,1,2,1,0,2,1,0,2,0,1,2,1,0/
INTEGER*2 NA(45)

IF (IVALUE.GT.10) GOTO 9876
CALL GRINIT_OLD
CALL GRSRST
CALL GRZFC(0,0)
CALL GRZCL(0.255,255)
CALL GRZON(0,1)
CALL GRSBFD

B=4
A=2

L=256
IL=L/B
JL=A*IL

DO 100 I=1,IL
NB(I)=255
NR(I)=0
NG(I)=B*I
IF(NG(I).GT.255)THEN
NG(I)=255
ENDIF

100 CONTINUE

DO 200 I=IL+1,JL
BB=((B/(I-A))*L)+((A/(A-I))*L)
NG(I)=255
NR(I)=0
NB(I)=BB
IF (NR(I).GT.255)THEN
NR(I)=255
ENDIF

200 CONTINUE

DO 300 I=JL+IL+1,JL+II
RR=(B*I)-(A*L)
NG(I)=255
NB(I)=0
NR(I)=RR
IF (NR(I).GT.255)THEN
NR(I)=255
ENDIF

300 CONTINUE

DO 400 I=JL+II+1,JL+II+II
GG=((B-1)*L)+(B*L)
NR(I)=255

400 CONTINUE
NB(I)=0
NG(I)=GG
IF(NG(I).GT.255) THEN
  NG(I)=255
ENDIF
400 CONTINUE

800 KC=0

NR(I)=0
NG(I)=0
NB(I)=0
NR(256)=255
NG(256)=255
NB(256)=255

889 CALL GRNWR(1,NB,IARRAY(1+KC),0,256,0)
CALL GRNWR(1,NG,IARRAY(2+KC),0,256,0)
CALL GRNWR(1,NR,IARRAY(3+KC),0,256,0)
CALL GRSBFD
CALL GRNIN(0,0,0,0)

IF (COLORBAR.EQ.1) THEN
  CALL GRNBY (0,0,0,0)
ENDIF
CALL GRSBFD
DO 11 1=1,300
  DO 111 J=1,900
    GOTO 950
1111 CONTINUE
GOTO 950
950 KC=0

CC ------------------------Draw colorbar at right side of screen------------------------

9876 IF (COLORBAR.EQ.1) GOTO 1010
CONTINUE
DO 1000 I=0,350
  DO 2000 J=1,19
    G=I*(254.0/350.0)
    NA(J)=G
2000 CONTINUE
CALL GRWLW(NA,410,1,19,0,0,1,1)
CALL GRSBFD
1000 CONTINUE
1010 CONTINUE
RETURN
END

Subroutine colorred(VALUE,COLORBAR)

INTEGER*2 NR(256),NB(256),NG(256),IARRAY(18)
DATA IARRAY/0,1,2,0,2,1,1,0,2,1,2,0,1,2,0,1,0/
INTEGER*2 NA(45)
10010  FORMAT(A)
10020  FORMAT(I)

NM256 = 0

IF (IVALUE.GT.10) GOTO 9876

CALL GRINIT_OLD
C CALLGRSRST
CALL GRSBFD
L=256
IL=I/3
JL=J*IL
NM256=255

DO 100 I=1,IL
NB(I)=0
NR(I)=I*3
NG(I)=0
IF (NR(I).GT.253) NR(I)=253
100 CONTINUE

DO 200 I=IL+1,JL
NR(I)=253
NB(I)=0
IF (NG(I).GT.253) NG(I)=253
200 CONTINUE

DO 300 I=JLI,1,255
NR(I)=253
NB(I)=I*3*(I-JL)
NG(I)=253
IF (NR(I).GT.253) NR(I)=253
300 CONTINUE

800 KC=6
NR(I)=0
NG(I)=0
NB(I)=0
NR(256)=NM256
NG(256)=NM256
NB(256)=NM256

840 CALL GRNWR(1,NB,IARRAY(l+KC),0.256,0)
CALL GRNWR(1,NG,IARRAY(2+KC),0.256,0)
CALL GRNWR(1,NR,IARRAY(3+KC),0.256,0)
CALL GRSBFD
CALL GRNIN(0,0,0,0)
CALL GRSBFD
CALL GRNBY(0,0,0,0)
CALL GRSBFD
C WRITE(5,10860)
C10860  FORMAT(’$<CR>$ for next color scheme, Q to quit: ’)
C READ(5,10010)ANS
C IF(ANS.EQ.'Q' .OR. ANS.EQ.'q') GOTO 9876
GOTO 9876
KC = KC+3
IF(KC.GE.18) KC=0
GOTO 840

CC -------- Draw colorbar at right side of screen --------

9876 IF (ICOLORBAR.EQ.1) GOTO 1010
CONTINUE
DO 1000 I=0,350
DO 2000 J=1,19
G=1*(254.0/350.0)
NA(J)=G
2000 CONTINUE
CALL GRWLW(NA,410,1,19,0,0,1,1)
CALL GRSBFD
1000 CONTINUE
1010 CONTINUE

C 900 CALL GRSEND
return
END
What is claimed is:

1. A pulse-echo, immersion method for ultrasonic evaluation of a material, employing automatic scanning and digital imaging to obtain an image of a property of said material, wherein said material is held in a holding apparatus which is positioned in an immersion liquid over an acoustic reflector, said reflector having an acoustic impedance which is greater than that of said liquid, and wherein nonlevelness in said holding apparatus and material thickness are accounted for and eliminated, said method comprising:

(i) ultrasonically scanning said material at a plurality of scan points and receiving the first and second echoes, each of which is a complete waveform, reflected off the back surface of said material and the first echo reflected off the front surface of said reflector both with and without the presence of said material;

(ii) adjusting the time delay for each said received echo from each said scan point during said scanning in (i) above, gating each said received echo so that it is centered within its respective time window;

(iii) automatically scanning said material at said scan points to receive said first and second back surface echoes and said two front surface echoes using the information obtained in (ii) above, so that each echo received from each scan point during said automatic scanning is centered within its time window;

(iv) digitizing each echo received during said automatic scanning and determining the time delay between said first two successive sample back surface echoes, $t_1$, and the time delay, $t_2$, between the two different reflector front surface echoes received at each scan point during said automatic scanning and calculating the wave velocity, using a cross correlation function, at each said scan point from

$$
\nu = c \left( \frac{t_1}{2} + 1 \right)
$$

where $c$ is the speed of the ultrasonic wave transmitted in said liquid, and

(v) scaling the velocity values obtained in (iv) to corresponding proportional color or grey scale values and displaying the resulting image.

2. A method according to claim 1 wherein a single transducer is used.

3. A method according to claim 2 wherein a single transducer is a high frequency transducer which emits a frequency between 1–100 MHz.

4. A pulse-echo, immersion method for ultrasonic evaluation of a material employing a single transducer, automatic scanning and digital imaging to obtain an image of a property of said material, wherein said material has a uniform thickness variation and is positioned in an immersion liquid between said transducer and an acoustic reflector, said method comprising:

(i) accounting for and eliminating nonlevelness in the set-up and said material thickness variation by;

(a) performing a preliminary scan along both the x- and y-directions of the material to provide slant correction factors which are input into a computer to account for said nonlevelness and thickness variation during the subsequent automatic scanning for said material evaluation in (ii) below;

(b) adjusting the time delay during said preliminary scan for any received echoes which are not centered in the scan time window, so that each received echo is centered in its time window;

(ii) automatically scanning said material at a plurality of scan points in both the x- and y-directions to receive the first and second back surface echoes and front surface echoes with and without the presence of said material between said transducer and reflector using the information obtained in (i) above, so that each echo received from each scan point during said automatic scanning is centered within its time window;

(iii) automatically scanning said material and determining the time delay between said first two successive sample back surface echoes, $t_1$, and the time delay, $t_2$, between the two different reflector front surface echoes received at each scan point during said automatic scanning and calculating the wave velocity at each said scan point from

$$
\nu = c \left( \frac{t_1}{2} + 1 \right)
$$

where $c$ is the speed of the ultrasonic wave transmitted in said liquid, and

(iv) scaling the velocity values obtained in (iii) to corresponding proportional color or grey scale values and displaying the resulting image.

5. A method according to claim 4 wherein said back and front surface echoes received from the first and last scan points in both the x- and y-directions during said preliminary scan are made in said third scan, and wherein said reflector echo without said material present are made in said first scan.

6. A method according to claim 5 wherein the location of the time window during said automatic scanning for said material evaluation is automatically adjusted via computer control by using the formula:

$$
W_{DP} = T_f \left( X_{SC}(X_{SC})X_{SC}(Y_{SC})(X_{SC}(Y_{SC})) \right)
$$

wherein $DP$ is the correct delay time window at a particular scan location, $T_f$ is the time delay at the initial scan location, $X_{SC}$ and $Y_{SC}$ are the x- and y-direction slant correction factors, $X_{SP}$ and $Y_{SP}$ are the scan point numbers in the x- and y-directions, and $X_{SP}$ and $Y_{SP}$ are the x- and y-direction scan increments.

7. A method according to claim 6 wherein two scans are automatically made to obtain said first two back surface echoes and said two different reflector front surface echoes.

8. A method according to claim 7 wherein said first two back surface echoes and said reflector echo with said material present are made in said first scan.

9. A method according to claim 6 wherein three scans are automatically made to obtain said first two back surface echoes and said two different reflector front surface echoes.

10. A method according to claim 9 wherein said first two back surface echoes are received in one scan wherein said reflector echo with said material present is made in another scan, and wherein and said reflector echo without said material present is received in yet another scan.

11. A method according to claim 9 wherein said first back surface echo is received in said first scan, wherein said second back surface echo is received in said second scan, wherein said reflector echo with said material present is made in said third scan, and wherein said reflector echo without said material present is received in said fourth scan.

12. A method according to claim 6 wherein four scans are automatically made to obtain said first two back surface echoes and said two different reflector front surface echoes.

13. An ultrasonic, pulse-echo, immersion method employing automatic scanning and digital imaging to obtain an
image of a microstructural property of a material positioned in an immersion liquid between a transducer and an acoustic reflector, said method comprising:

(i) automatically scanning said material at least three times at a plurality of scan points in both the x- and y-directions to receive the first and second back surface echoes and front surface echoes with and without the presence of said material between said transducer and reflector, each of said echoes received being a complete waveform and gated within a time window;

(ii) digitizing each echo received during said automatic scanning and determining the time delay between said first two successive sample back surface echoes, 2τ and the time delay, Δt, between the two different reflector front surface echoes received at each scan point during said automatic scanning and calculating the wave velocity at each said scan point from

\[ v = c \left( \frac{\Delta t}{2\tau} + 1 \right) \]

where c is the speed of the ultrasonic wave transmitted in said liquid, and

(iii) scaling the velocity values obtained in (ii) to corresponding proportional color or grey scale values and displaying the resulting image.

14. A method according to claim 13 wherein four scans are automatically made to obtain said first two back surface echoes and said two different reflector front surface echoes.

15. A method according to claim 14 wherein said first back surface echo is received in said first scan, wherein said second back surface echo is received in said second scan, wherein said reflector echo with said material present is made in said third scan, and wherein said reflector echo without said material present is received in said fourth scan.

* * * * *