NOAA Technical Memorandum ERL AOML-66

JOINT CGS-AOML ACOUSTICAL BOTTOM ECHO-FORMATION RESEARCH I: LITERATURE SEARCH AND INITIAL MODELLING RESULTS

Thomas L. Clarke
John R. Proni
Dennis A. Seem
John J. Tsai

Atlantic Oceanographic and Meteorological Laboratory
Miami, Florida
March 1988

Stimulating America's Progress
1913-1988

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL AOML-66

JOINT CGS-AOML ACOUSTICAL BOTTOM ECHO-FORMATION RESEARCH I: LITERATURE SEARCH AND INITIAL MODELLING RESULTS

Thomas L. Clarke
John R. Proni
Dennis A. Seem
John J. Tsai

Atlantic Oceanographic and Meteorological Laboratory
Miami, Florida
March 1988

UNITED STATES DEPARTMENT OF COMMERCE

C.William Verity
Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

Environmental Research Laboratories

Vernon E. Derr,
Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA Environmental Research Laboratories. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

For sale by the National Technical Information Service, 5285 Port Royal Road
Springfield, VA 22161

CONTENTS

LIST OF FIGURES

JOINT CGS-AOML ACOUSTICAL BOTTOM ECHO-FORMATION RESEARCH I: LITERATURE SEARCH AND INITIAL MODELLING RESULTS

Thomas L. Clarke, John R. Proni, Dennis A. Seem and John J. Tsai

National Oceanic and Atmospheric Administration
Atlantic Oceanographic and Meteorological Laboratory
4301 Ricenbacker Causeway
Miami, Florida 33149

ABSTRACT. A literature survey has found work dealing with the problem of echo-formation for rough sedimentary bottoms. The main attempts at practical application have been at frequencies lower than those used for echo-sounding work. A model has been formulated that includes the effect of surface scattering and volume scattering in a unified manner. This model has been implemented on the AOML computer and exercised for a variety of bottom types and echo-sounding frequencies.

1. BACKGROUND

In the summer of 1984 a joint program for investigating the mechanism of formation of acoustic bottom echoes was initiated by the Charting and Geodetic Servics (CGS) branch of the National Ocean Survey (NOS) and by the Atlantic Oceanographic and Meteorological Laboratory (AOML) of the Environmental Research Laboratories (ERL) of NOAA. The objective of this research was to better understand the complex interaction of sound, including its transmission and reflection, with the diverse types of bottoms encountered in U.S. coastal waters.

A major tool of the CGS is the classical acoustic echo sounder. CGS depends on the echo sounder for accurate bottom depth information for the construction of nautical charts. More recent technological innovations include the use of two-frequency, two-beamwidth echo sounders and multiple-beam bathymetric swath sounding systems (BS^3 and Sea-Beam) as well as enhanced digital recording systems. In order to understand the limitations and possibilities of these instruments, a good empirically tested model, or perhaps models, of the formation of echoes arising from bottom sound scattering or reflection is needed.

An additional benefit that could result from a sound theoretical and empirical understanding of the acoustic echo formation process is the ability to remotely classify sediment on the basis of the echo waveform. This ability would permit nautical charts to contain greatly expanded information about the bottom type. Exploration of the U.S. Exclusive Economic Zone (EEZ) would also be enhanced by a remote bottom classification ability.

This report describes the first phase of model development.

2. PREVIOUS STUDIES

Before embarking upon the construction of a model(s) for bottom echo formation, a literature review was conducted. The standard journals dealing with underwater acoustics and its applications, Acta Acoustica, J. Acoustical Society of America (JASA), J. Geophysical Research, Geophysics, and J. Sedimentary Petrology, were scanned for relevant articles. In addition, symposium volumes, texts, technical reports and other books were examined. Dr. Er-Chang Shang from the Institute for Acoustical Research in China, at AOML on an NRC Post-Doctoral Fellowship, provided some valuable references to unpublished Chinese work that might not have been available otherwise.

The most important of these references found are listed together with a brief summary in the annotated bibliography.

Particularly relevant is the work by Dodds (1984), Breslau (1967), McLeroy (1972), and Meng and Guan (1984). All of these authors deal with the problem of using acoustics to remotely classify bottom types. Dodds' company, Huntec Ltd. (Canada), has developed the Deep Tow system which uses a broadband acoustic source (0-10 kHz) to remotely classify the bottom with some success. This system is thus not directly applicable to the problem of bottom classification using echo-sounding frequencies which are 20 kHz and higher. Some of Dodds' data analysis techniques may prove useful, however, in analyzing data taken at higher frequencies.

The other authors describe measurements made at higher frequencies, and their work is more directly relevant to the echo-sounder problem. Breslau is concerned with inferences based on the reflection coefficient of the bottom, whereas McLeroy and Meng and Dinghua are concerned with extracting information from the echo waveform. The work of these authors establishes the viability of using acoustic echoes to remotely sense bottom type.

In support of the project, OAD has conducted a regular seminar or discussion session series in which topics related to the bottom echo problem were discussed. In addition to presentations by OAD personnel, talks were given by Dr. Frederick Tappert and by Dr. Tokuo Yamamoto, both of the Rosenthiel School of Marine and Atmospheric Science (RSMAS) of the University of Miami.

Dr. Yamamoto's talk dealt with the application of Biot theory to the propagation of sound within marine sediments. This is particularly important to calculating the penetration depth of sound within the sediment. Dr. Tappert's talk concerned the scattering of sound from the statistically rough seawater/sediment interface. Result's from Dr. Tappert's theory will be presented in the next section, and they will be the subject of a joint paper with OAD personnel.

3. MODEL FORMULATION

The basic picture is that of an acoustic beam from an echo sounder impinging on the sea bottom. Sound reflected and scattered from the bottom is received by a hydrophone which is usually, but not necessarily, the same transducer used to emit the sound. The basic quantities, beam angle θ, range R, bottom roughness η, and length scale L, are defined in Figure 1. Additional parameters are the acoustic wavelenth λ, wavenumber $k = 2\pi/\lambda$, and frequency $f = c/\lambda$, where c is the speed of sound in water.

For this first version of the model, the bottom will be assumed to be composed of a uniform sediment with grains of radius a. A sound beam will then be subject to three types of reflection or scattering:

1) Coherent or mirror reflection from the bottom,
2) Incoherent or statistically variable scattering from the surface irregularities of the bottom, and
3) Incoherent scattering from within the volume of the sediment due to the acoustically irregular matrix of the sediment.

Physical reasoning suggests that 3) is independent of 1) and 2) except for a reduction in acoustic amplitude due to reflection at the surface. Once the sound has penetrated the seawater/sediment interface, the existence of the interface should have no effect on the way in which the granular matrix of the sediment scatters the sound. Consequently, the volume scattering 3) will be treated separately from the surface scattering 1) and 2). The equations describing the reflection and scattering of sound will now be presented. Those interested in the overall result can skip ahead to equation (7).

A rigorous calculation of expressions for the surface scattering was presented by Dr. Tappert during his contribution to the OAD discussion

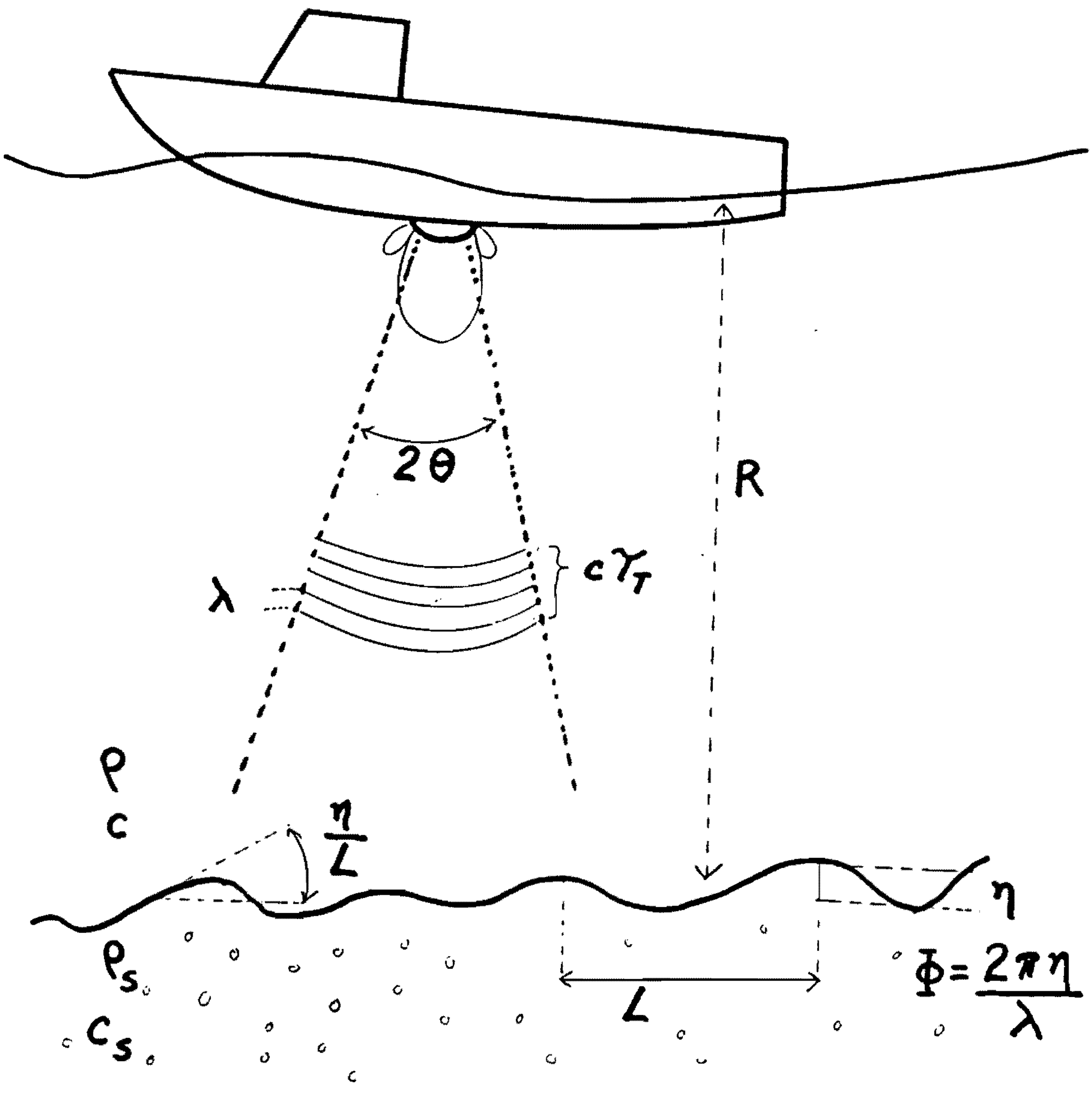

Figure 1. Echo-sounding geometry assumed in bottom acoustic echo formation model.

session. This calculation will be presented in some detail in a future paper, but the main result is

$$I_{coh} = \left[\frac{R^2 \ e^{-\Phi^2}}{R^2} \ \delta(t-2R/c) \right] * I_{trans} \tag{1}$$

and

$$\langle I_{sur} \rangle = \left[\frac{R^2 \ (1-e^{-\Phi^2})}{R^2} \ H(t-2R/2) \ e^{(2R/c-t)/\tau_s} \right] * I_{trans} \tag{2}$$

where $R = (\rho_s c_s - \rho_c)/(\rho_s c_s + \rho_c)$ is the plane wave reflection coefficient of the sediment, δ is the delta function, * denotes convolution, H is the unit step function, and

$$\tau_s = (\eta/L)^2 \ R/2c \tag{3}$$

measures the pulse broadening due to the surface irregularities. Note that the plane wave reflection coefficient is dependent primarily on the density of the sediment which is highly correlated to the sediment porosity. A measurement of the total surface scattering (1) + (2) thus provides an estimate of sediment porosity.

Equation (1) shows that the coherent part of the signal is a replica of the transmitted intensity I_{trans} delayed by the travel time 2R/c, and reduced in amplitude by $R^2 \ e^{-\Phi^2}/R^2$. The R^2 dependence is simply one-way spherical spreading for a mirror reflection, whereas the $\exp[-\Phi^2]$ term reflects the conversion of coherent energy into incoherent scattering. For rippled sandy bottoms, η is a few cm so that for echo sounding wavelengths of 6 cm and less $\Phi = k\eta$ will be large and the coherent echo will be exponentially small.

Equation (2) shows that the incoherent scattering takes the form an exponentially decaying pulse with width given by equation (3). The one-way spherical spreading in equation (2) includes the assumption that the bottom slope η/L is the limiting factor the areal extent of the bottom scattering. In case $\theta < \eta/L$ is limiting or if a very short transmitted pulse is limiting, then other modified, but similar expressions obtain, e.g., replace η/L by θ, but will not be presented in detail in this report.

Equation (3) for the amount of pulse broadening due to surface scatter can be approximated in an intuitive fashion from Figure 1. Following an off-axis acoustic ray at angle η/L, a typical angle for a bottom roughness element, suggests that a typical value for the additional travel time due to off-axis scatter is

$$2/c\sqrt{R^2 + R^2\eta^2/L^2} - R \approx (\eta/L)^2\ R/c \quad (4)$$

This expression differs from the exact equation (3) by only a factor of 2. A more careful argument involving root mean square bottom slopes could repair this deficit.

Once the sound has penetrated the sediment, it will propagate with exponentially decaying amplitude. The coefficient of this decay is described by Biot theory for granular sediments when the grains are small compared to a wavelength and is shown in Figure 2. The equations on which this figure is based can be found in Yamamoto (1983) and will not be presented here. The rate of decay increases strongly with frequency but undergoes a change in slope at a frequency dependent on the grain size. This is due to a match between the pore size of the sediment and the thickness of the boundary layer produced by the sound-induced relative pore water/grain motion.

Biot theory treats the sediment water system as a continuum and thus predicts no backscatter. In order to calculate the amount of scattering from the volume of the sediment, the effective field approximation of multiple

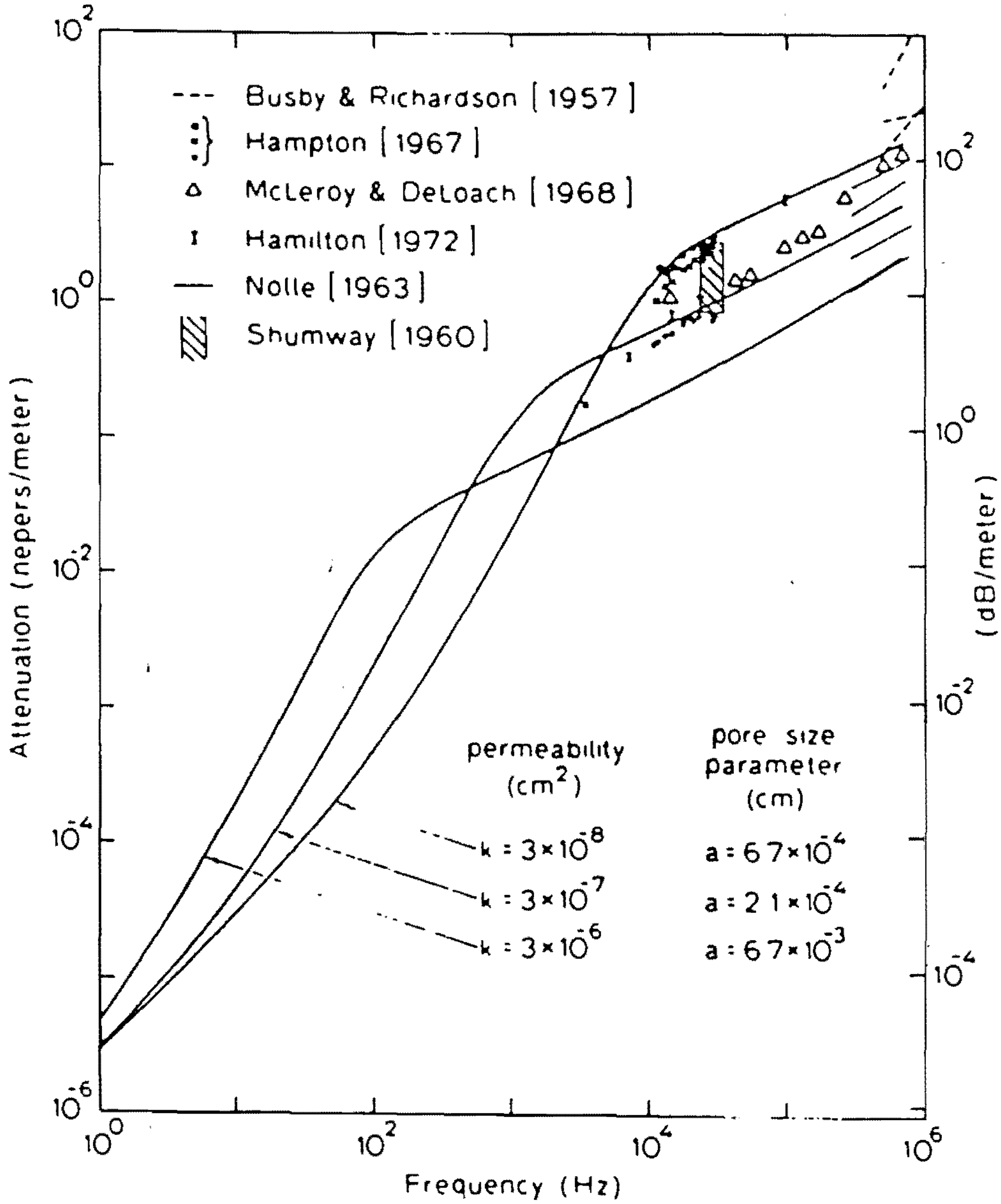

FIGURE 4

ATTENUATION VERSUS FREQUENCY FOR SANDS

Figure 2. Curves of attenuation coefficient calculated from Biot theory versus frequency for three grain sizes of sediment. Reproduced from Stoll (1974).

scattering theory will be used (Waterman, 1961). This is a good approximation as long as the grains are small compared to a wavelength (Rayleigh regime). This approximation says that the scattering from a single particle can be calculated on the basis of the field that would be present at the particle location in the absence of the particle. In essence this approximation says that the particles are acoustically small and individually do not perturb the incident field greatly.

Making use of this approximation, the incident field will be assumed to be that given by Biot theory; the reduction in acoustic field strength caused by the scattering will be ignored. This will be a good approximation provided the particles are small compared to a wavelength. In this case the scattering from an individual particle will be given by the Rayleigh formula:

$$\sigma = R_p \ (ka)^4 \ \pi a^2 \tag{5}$$

where R_p is the grain/water reflection coefficient. Note that the acoustic cross section σ is much smaller than the geometric cross section πa^2, justifying the effective field approximation. Consequently, the volume scattering from the mass of particles making up the sediment is described by

$$\langle I_{vol} \rangle = \left\{ \left[\frac{3 \ \theta_p \ (ka)^4}{4 \ (1-B) \ a\alpha} \right] 1/R^2 \ H(t-2R/2-2_z/c_s) e \ (2R/c+2_z/c_s-t)\alpha \right\} * I_{trans} \tag{6}$$

where α is the Biot absorption coefficient, z is depth within the sediment, and B is the porosity.

The volume scattering has the same form as the surface scattering where the depth z is converted to time by use of the sound speed c_s within the sediment. The surface scattering, however, depends on frequency only through the amplitude term $\exp(-\Phi^2)$ which has the effect of shifting energy from a coherent reflection to incoherent scattering at a frequency where $\Phi \approx 1$; the surface pulse broadening is frequency independent. In contrast, the amplitude of the volume scattering increases continuously as the fourth power of frequency, whereas the amount of pulse broadening decreases with frequency. The volume scattering amplitude does not increase without bound, of course, eventually the particles become comparable to the wavelength of the sound so that the effective field approximation breaks down. This does not happen until the frequency reaches the megahertz range for sand size grains.

The mathematical form of the model is thus

$$\langle I_{rec} \rangle = [A_c \delta/t) + A_s e^{-t/\tau_s} + A_v e^{-t/\tau_v}] * I_{trans} \tag{7}$$

where the amplitude terms A_c, A_s, and A_v, and the pulse spreads, τ_s and τ_v, can be extracted from equations (1) through (6) and the time origin has been shifted by the travel time 2R/c. Equation (7) represents the impulse response of the bottom. To predict the echo produced by a given incident waveform, the pulse waveform must be convolved with the impulse response. The delta function in equation (7) then gives a reduced amplitude replica of the transmitted pulse, and the two exponentials give rise to stretched pulse returns.

To carry out the convolution needed to evaluate the bottom echo, a discrete time version of equation (7) was used along with a discrete or digitized value of the incident waveform. If Δt is the discrete time interval and N is the length of the time series, the operations to be carried out on a digital computer are given by

$$\langle I_{rec}(j) \rangle = \sum_{\substack{i=1 \\ j \geq 1}}^{N} I_{trans}(i) \left[A_s e^{-(j-i)\Delta t/\tau_s} + A_v e^{-(j-1)\Delta t/\tau_v} \right] + A_c I_{trans}(j) \tag{8}$$

To carry out the N sums needed for the discrete convolution the Fast Fourier Transform (FFT) was used. Since the FFT of a convolution of two series is the product of the FFTs of the two original series, using the FFT converts the N^2 operations needed for a discrete convolution into two forward FFTs and an inverse FFT. These FFTs require $N\log_2 N$ operations so that for the value N = 1024 used, over a million operations required by a brute force technique is reduced to about 30,000. This increase in computational speed of a factor of 30 can be very important when evaluating the bottom echo for many different frequencies.

4. MODEL RESULTS

The model described above can be applied to any combination of acoustic frequency, bottom roughness (both η and L), and grain size a. In addition, to display realistic bottom echoes a transmitted waveform (rectangular?, long?, short?) must be chosen. The full range of possibilities cannot be displayed in this memorandum, and a representative sample has been chosen. The model can be evaluated for any desired combination of parameters using the AOML VAX computer system and the computer programs shown in the appendices.

Twenty-nine "standard" cases of bottom types have been developed and run on the VAX. The input parameters for each of these cases are listed on the following page. When parameters are not listed for a particular case, they are the same as those of the previous case. Cases 1 through 3 are for a mud bottom. Case 1 is a very smooth surfaced bottom. Case 2 is a smooth bottom. Case 3 is also a smooth bottom, but with different height characteristics. Cases 4 through 6 are for a fine sand bottom. Case 4 is a smooth bottom. Case 5 is a rough bottom. Case 6 is a very rough bottom. Cases 7 through 9 are for a medium sand bottom. They have the same surfaces as cases 4 through

6. Cases 10 through 12 are for a coarse sand bottom. They have the same surfaces as cases 4 through 6. Cases 13 through 15 are for a gravel bottom. They have the same surfaces as cases 4 through 6.

Several cases representing a two layered bottom have been developed. Case 16 is smooth, very porous mud, a half meter thick, over a smooth surfaced medium sand. Case 17 is the same as 16 except that the sand has a rough surface. Case 18 is the same as case 16 except that the mud is now one meter thick. Case 19 is the same as case 17 except that the mud is now one meter thick. Cases 20 through 23 are the same as cases 16 through 19 except that the bottom layer is now composed of less porous mud. Cases 24 through 26 are for a medium sand bottom with a rough surface. The difference between the cases is that the depth ranges from 15 meters to 30 meters to 60 meters. Cases 27 through 29 are for a porous mud bottom, 1 meter thick, over a rough surfaced medium sand bottom for depth ranges of 15, 30, and 60 meters.

The bottom echo waveforms calculated from the model are displayed in the form of contour plots of expected acoustic intensity versus time and frequency. These contour plots are thus similar to Dobbs' sonograms. A horizontal slice from one of these plots gives the expected echo waveform at a given frequency as a function of time. A vertical slice gives the echo at a given time (or depth using c_s) as a function of frequency.

Figure 3 shows three such contour plots for three combinations of sediment size and bottom roughness, and for times between 0 and 2 milliseconds (msec) and for frequencies between 10 and 200 kHz. The assumed pulse length is 100 microseconds (usec), and the mean bottom slope η/L is fixed at 0.1.

The contour plot labelled Smooth Mud, has a roughness η of only .001 m so that $\Phi \approx 1$ only for frequencies in the 100 kHz range. The grain size is very small (20 microns), so that ka is always very small and there is relatively

Table of Model Cases

Case	Roughness Length	Roughness Height	Depth	Beam Width	Beta	Radius	SEDSPD	Rho	Thick	Roughness Length	Roughness Height	Beta	Radius	SEDSPD	Rho
1	1.0	0.001	50.	12.	0.75	50.e-6	1520.	1.42	---	---	---	---	---	---	---
2	0.1	0.01	---	---	---	---	---	---	---	---	---	---	---	---	---
3	0.1	0.001	---	---	---	---	---	---	---	---	---	---	---	---	---
4	1.0	0.001	---	---	0.50	100.e-6	1710.	1.91	---	---	---	---	---	---	---
5	0.1	0.01	---	---	---	---	---	---	---	---	---	---	---	---	---
6	0.3	0.03	---	---	---	---	---	---	---	---	---	---	---	---	---
7	1.0	0.001	---	---	0.40	200.e-6	1740.	1.98	---	---	---	---	---	---	---
8	0.1	0.01	---	---	---	---	---	---	---	---	---	---	---	---	---
9	0.3	0.03	---	---	---	---	---	---	---	---	---	---	---	---	---
10	1.0	0.001	---	---	---	400.e-6	1840.	2.03	---	---	---	---	---	---	---
11	0.1	0.01	---	---	---	---	---	---	---	---	---	---	---	---	---
12	0.3	0.03	---	---	---	---	---	---	---	---	---	---	---	---	---
13	1.0	0.001	---	---	---	800.e-6	1860.	2.10	---	---	---	---	---	---	---
14	0.1	0.01	---	---	---	---	---	---	---	---	---	---	---	---	---
15	0.3	0.03	---	---	---	---	---	---	---	---	---	---	---	---	---
16	0.1	0.001	---	---	0.95	25.e-6	1515.	1.30	0.5	1.0	0.001	0.40	200.e-6	1740.	1.98
17	---	---	---	---	---	---	---	---	---	0.1	0.01	---	---	---	---
18	---	---	---	---	---	---	---	---	1.0	1.0	0.001	---	---	---	---
19	---	---	---	---	---	---	---	---	---	0.1	0.01	---	---	---	---
20	---	---	---	---	---	---	---	---	0.5	1.0	0.001	0.75	50.e-6	1520.	1.42
21	---	---	---	---	---	---	---	---	---	0.1	0.01	---	---	---	---
22	---	---	---	---	---	---	---	---	1.0	1.0	0.001	---	---	---	---
23	---	---	---	---	---	---	---	---	---	0.1	0.01	---	---	---	---
24	---	0.01	15.	---	0.40	200.e-6	1740.	1.98	X	X	X	X	X	X	X
25	---	---	30.	---	---	---	---	---	---	---	---	---	---	---	---
26	---	---	60.	---	---	---	---	---	---	---	---	---	---	---	---
27	1.0	0.001	15.	---	0.95	25.e-6	1520.	1.42	1.0	0.1	0.01	0.40	200.e-6	1740.	1.98
28	---	---	30.	---	---	---	---	---	---	---	---	---	---	---	---
29	---	---	60.	---	---	---	---	---	---	---	---	---	---	---	---

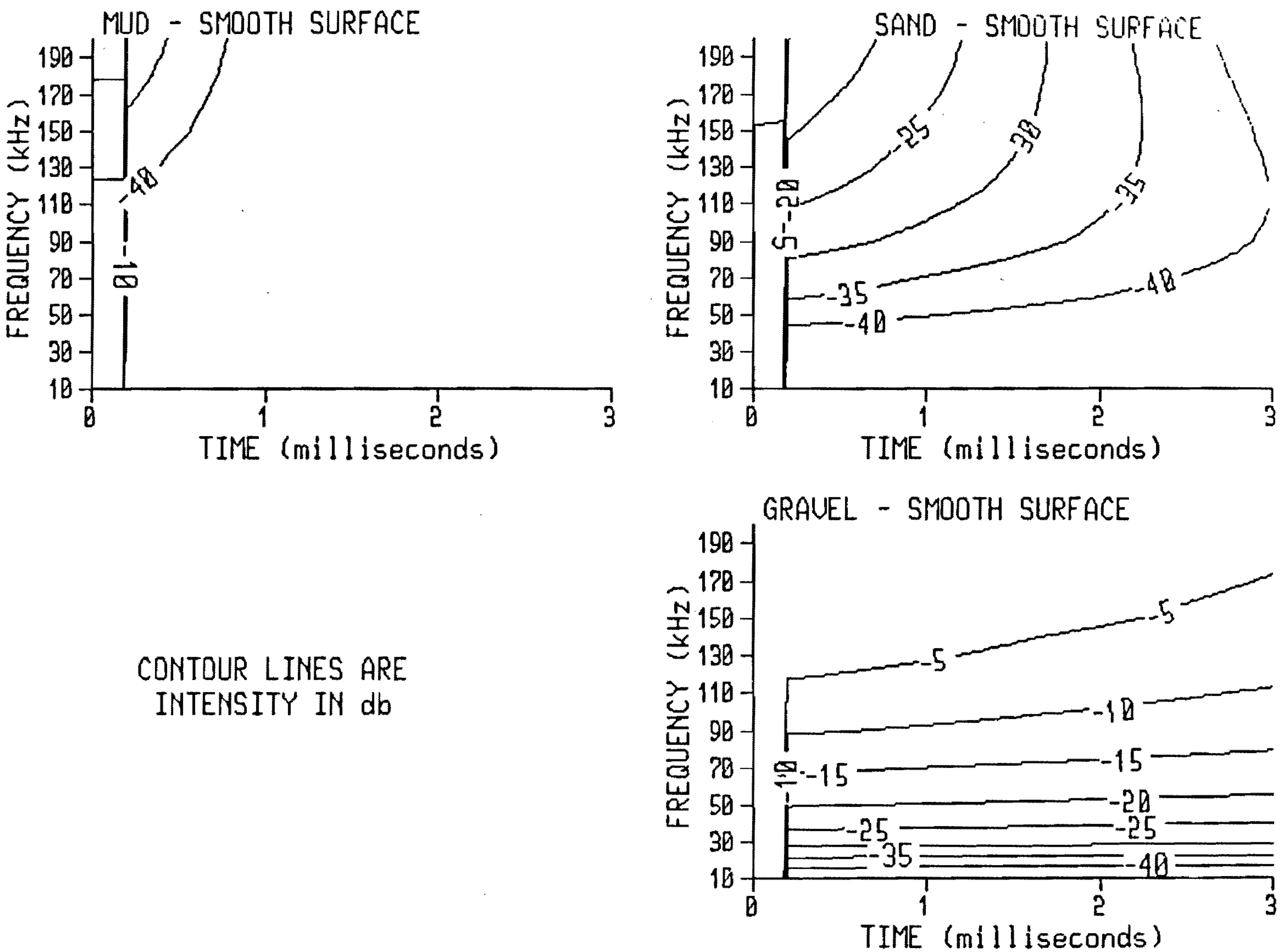

Figure 3. Contour plots of expected echo intensity versus time and frequency for Smooth Mud, Fine Sand and Coarse Sand.

little volume scatter from within the sediment. Consequently, the echo waveform is a very precise replica of the transmitted waveform at all frequencies up to the 100 kHz range, and even at these high frequencies the spread due to surface scattering is small.

The two plots labeled Fine Sand and Coarse Sand have in common a rougher bottom, η = .01 m such as might characterize a rippled sandy bed. The surface scattering has a similar effect in these two plots; beginning at about 20 kHz, the narrow incident waveform is transformed into a much broader incoherent return. Because of the small grain size (200 microns), the Fine Sand has little volume scattering at frequencies below 200 kHz so that the purely geometric, frequency independent surface effects dominate for Fine Sand. This is reflected in the verticality of the intensity contours. Only near 200 kHz is any effect of volume scattering discernible in the first msec of the echo.

The Coarse Sand plot is more complex and shows a much stronger influence of volume scattering due the larger grain size (420 microns). Instead of remaining vertical beyond the relection/surface scattering transition, the contours bulge outward and then inward with increasing frequency. This bulge is due to the competition between Biot absorption and Rayleigh scattering within the sediment volume. The tail of the reflected pulse depends strongly on the amount of sound scattered at the corresponding depth of penetration in the sediment. This scattering depends on the product of Biot absorption and Rayleigh scattering, which has a relative maximum versus frequency at any given depth. The waveform at 200 kHz is dominated by the effects of volume scattering so that the sediment characteristics should be deducible from the waveform. This supports the work of McLeroy and of Meng and Dinghue.

To make more clear the effect of surface scattering as opposed to volume scattering, Figure 4 shows three contour plots for the same grain size as the

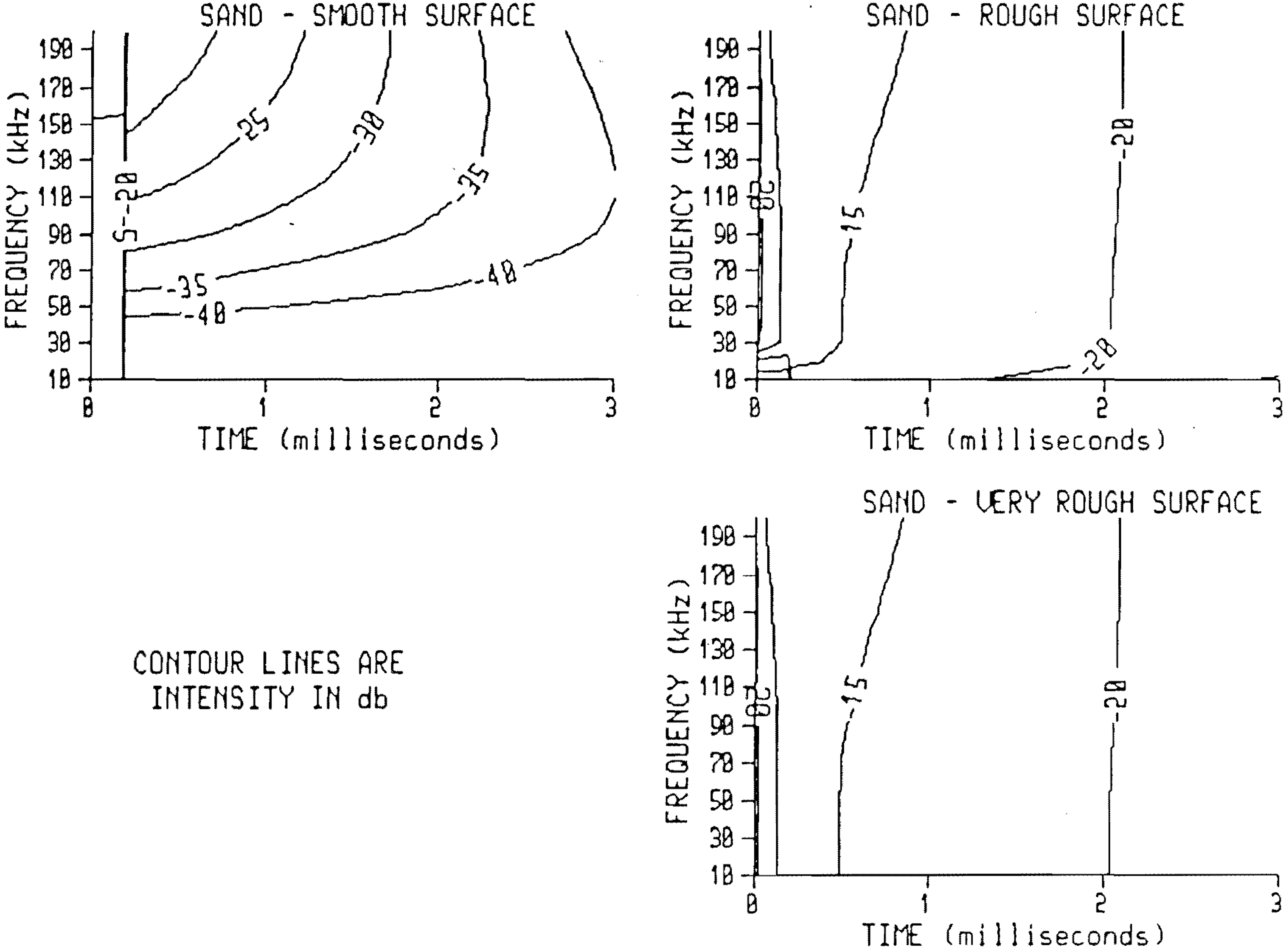

Figure 4. Contour plots of expected echo intensity versus time and freuqency for Smooth Surface, Rough Surface (same as Coarse Sand in Figure 3), and Very Rough Surface.

Coarse Sand plot but for varying degrees of surface roughness η. The Smooth Surface plot uses η = .001 m, which probably corresponds to the irreducible roughness of a sandy bed. This plot shows a coherent reflection maintained up to a fairly high frequency, but with a strong volume scattering component superimposed. This is to be compared with the Rough Sand plot using η = .01 m and is the same as the Coarse Sand of Figure 3. For the Rough Surface, the coherent echo is all but gone by 20 kHz, leaving an echo which is entirely determined by the competing effects of surface scattering and volume scattering.

For the Very Rough Surface plot, η = .1 m, so that $\Phi > 1$ for all frequencies plotted. All echoes predicted on this plot are thus a function solely of surface and volume scattering. For high frequencies, this plot differs little from that for η = .01 m because once $\Phi > 1$ the effect of surface scattering saturates and does not depend strongly on η.

5. SUMMARY

The model constructed to date draws on ideas from several sources: Biot theory, multiple scattering theory, and rough surface scattering theory. While some simplifying assumptions are made, e.g., that volume scattering does not add significantly to attenuation of sound within the sediment, they are relatively mild. The results of this model should thus predict fairly accurately averaged pulse returns from bottoms that fit the homogeneity assumption of the model.

The model will be extended to include effects of layered sediment by adding additional surface interfaces, with seperate Biot coefficients and grain sizes for each layer. This enhanced model should provide good results for bottoms characterized by layers of different sediment types. Possible coherent interference effects between the layers will be allowed for by using a parabolic equation model to propagate sound into the sediment.

While the model has been specialized to near normal incidence, it can be expanded to the case of grazing incidence by modifying the surface roughness parameters by an angular factor, and by the deletion of the coherent term. At grazing incidence, there is no coherent reflection back in the direction of the beam, the entire echo is due to scattering. A grazing model would be very useful in analyzing performance of muliple beam bathymetric systems such as the BS^3 and the Sea-Beam.

This model has not been empirically tested, but a field program designed to test the model should begin within the year. Present AOML equipment can be used to measure echo waveforms at 20 and 200 kHz, and the AOML 3 MHz acoustic system can be used to produce profiles of bottom height and hence to determine n. Some ground truth will be needed to actually determine the porosity and grain size of the bottom material.

The ideal site for an experiment would include bottoms of several types within easy traveling distance so that the model can be compared to measurements for a variety of bottom parameters. It would also be desirable to obtain additional acoustic equipment so that echo waveforms at additional frequencies could be measured.

The ability of the model to quickly generate echo waveforms for a large range of botom types should prove useful in developing software for recognizing bottom types from the echo. The similarity of the sonogram contours to speech spectrograms suggest the use of the same type of software used in speech recognition. These algorithms commonly are based on pattern recognition or artificial intelligence techniques wherein the recognizer "learns" to correctly identify spoken words by being presented with many examples. On the basis of the correctness/incorrectness of the recognizer's identification of the word, the recognition parameters are updated so that subsequent identifications are more accurate. These algorithms have been incorporated into special speech recognition integrated circuits.

An "intelligent" echo sounder capable of recognizing the bottom type from the echo could be formed by mating speech recognition algorithms using "chips" or general purpose computing hardware with an echo sounder and "training" the resulting system using the model developed here.

6. ANNOTATED BIBLIOGRAPHY

Berman, A., 1972. "On the medium from the point of view of underwater acoustics," JASA, 51(3), 994-1009.

A brief survey of the state of knowledge in ocean acoustics circa 1948 and review of the advances made during the past two decades.

Berkson, J. M., 1980. "Measurements of coherence of sound reflected from ocean sediments," JASA, 68(5), 1436-1441.

Bezdek, H. F., 1973. "Reflection of high-frequency sound at normal incidence from the ocean bottom," J. Geophys. Res., 78(17), 3390-3394.

Measures peak echo level of 75 kHz pulse as a function of range. Results show that specular reflection dominates, not scattering processes in the San Diego trough.

Breslau, L. R., 1967. "The normally-incident reflectivity of the seafloor at 12 kc and its correlation with physical and geological properties of naturally-occurring sediments," WHOI Reference No. 67-16, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts.

Bottom loss was measured in terms of peak pressure and of the time integral of the square of pressure (total energy). Bottom loss (Rayleigh reflection coefficient) measurement was related to physical and geological properties of sediment via porosity.

Bucker, H. P., "Sound propagation calculations using bottom reflection functions," In: Physics of Sound in Marine Sediments, Lloyd Hampton (ed.), Plenum Press, New York.

Carter, C. H., Williams, S. J., Fuller J. A. and Meisburger, E. P., 1980. "Regional geology of the southern Lake Erie (Ohio) bottom: a seismic reflection and vibracore study," MR 82-15, U.S. Army, Corps of Engineers, Coastal Engineering Research Center, Fort Belvoir, Virginia.

Classify seven sediment surfaces by echo character types (number of multiples).

Chapman, R. P., and Harris, J. H., 1962. "Surface backscattering strengths measured with explosive sound sources," JASA, 34(10), 1592-1597.

Measure backscatter strength as functions of wind speed, frequency (400 to 6400 cps), and grazing angle (below 40 degrees) from bubble and shock pulses.

Clay, C. S., 1966. "Coherent reflection of sound from the ocean bottom," J. Geophys. Res., 71(8), 2037.

Measure coherent component of reflected signals for several Fresnel zones as a function of grazing angle. The coherent reflection coefficient is given by Eckart, exponentially decaying as square of RMS roughness.

Clay C. S., and Leong, W. K., "Acoustic estimates of the topography and roughness spectrum of the seafloor southwest of Iberlian Peninsula," In: Physics of Sound in Marine Sediments, Lloyd Hampton (ed.), Plenum Press, New York.

Describes method to measure remotely the properties of the seafloor by defining (1) coherent reflection coefficient which is obtained by measuring the average of an ensemble of reflected signals, and (2) scattering coefficient which is the mean square received signal.

Danbom, S. H., 1976. "Sediment classification by seismic reflectivity," Oceans '76, MTS-IEEE, 16D-1.

Use a semi-automatic system to measure seismic reflectivity, and express the reflectivity amplitude as a function of mean grain size.

Dodds, D. J., 1984. "Surface and volume backscattering of broadband acoustic pulses normally incident on the seafloor: observations and models," JASA, 75, Suppl. 1, S29.

Use sonogram (a contour diagram of signal power as a function of time and frequency) to disply scattering strength, and to estimate sediment parameters. The approach is to calculate a synthetic sonogram from parameters of volume scattering and surface roughness, and fit to an actual sonogram, yielding estimates of the parameters.

Dunsiger, A. D., and MacIsaac, R. R., 1978. "Broadband seismic data used for seafloor sediment classification," Oceans '78, MTS-IEEE, 521-526.

Separates four different sediment types based on two metrics: (1) the maximum value of the normalized cross-correlation function, and (2) the normalized water-sediment interface energy.

Fortuin, L., 1970. "Survey of literature on reflection and scattering of sound waves at the sea surface," JASA, 47(5), 1209-1228.

Review of wave diffraction (reflection and sering) t rough surface up to 1969. Most literatures only deal with one of the three basic quantities (time, frequency of incident wave, and geometry), but not all the three simultaneously.

Frisk, G. V., and Lynch, J. F., 1984. "Shallow water waveguide characterization using the Hankel transformation," JASA, 76(1), 205-216.

Present a method to measure the magnitude and phase versus range of the pressure field due to a cw point source, and Hankel transforming these data to obtain the depth-dependent Green's function versus horizontal wavenumber. The Green's function can then be used to extract modal properties and characteristics of the bottom (via calculation of reflection coefficient).

Hamilton, E. L., 1970. "Reflection coefficients and botttom losses at normal incidence computed from Pacific sediment properties," Geophysics, 35(6), 995-1004.

Compute Rayleigh reflection coefficients and bottom losses at normal incidence with values of density and velocity measured in the seafloor sediment samples of (1) continental shelf, (2) abyssal plain, and (3) abyssal hill. Resare presented as impendance, reflection coefficient and bottom loss as functions of ty and ity respectively.

Hamilton, E. L., 1980. "Geoacoustic modeling of the seafloor," JASA, 68(5), 1313-40.

Define geoacoustical models, and review recent results and methods of sediment parameter measurements. The geoacoustical model should include the following: (1) identification of sediment and rock types; (2) true thickness and shapes of underlying layers; (3) compressional wave ity; (4) shear wave velocity; (5) attenuation of compressional waves; (6) attenuation of shear waves; (7) density; (8) additional elastic properties (e.g., dynamic rigidity and Lame's constant).

Hoven, J. M., and Holt, R. M., 1983. "Acoustical classification of the seafloor," In: Issues in Acoustic Signal-Image Processing and Recognition, C. H. Chen (ed.), Springer, New York, pp. 77-93.

An exposition of the wide angle reflection method and of a near boundary wave technique for determining sheer strength.

Huang, J. C., 1971. "Analysis of acoustic wave scattering by a composite Rough surface," JASA, 49(5), 1600-1608.

Derive theoretical scattered acoustic intensity from composite rough surface in terms of surface slope, incident angle, and wavelength. Wavelength is the most important factor among the three. The specular reflection intensity is relatively larger than the diffusion scattering even at high frequency (100 kHz).

Jackson, D. R., 1984. "A semi-empirical model for high-frequency bottom backscattering," JASA, 75, Suppl. 1, S.

Relate backscattering strength to four bottom parameters: (1) ratios of sediment to water mass density, (2) sediment to water sound speed, (3) volume backscatter cross section to absorption coefficient, and (4) roughness spectrum. The model parameters are an empirical expression of sediment grain size. Somewhat similar to approach taken in this memo but emphasis is on predicting reflection coefficient versus angle.

Jackson, D. R., D. P. Winebrenner, and D. Ishimoru, 1986. "Application of the composite roughness model to high-frequency bottom backscattering," JASA, 79(5), 1410-1422.

An extension of Jackson's earlier work to include a more sophisticated surface scattering model.

Krobles, D. P., and P. J. Vidmar, 1986. "Simulation of bottom interacting waveforms," JASA, 79(6), 1760-1766.

Synthesize echo waveforms from bottom characteristics in the same spirit as the CGS-AOML model.

Lo, En-cen, and Zhou, Ji-xun, 1983. "Normal mode filtering in shallow water," JASA, 74(6), 1833-1848.

Two parameters, P and Q, were calculated from the normal mode eigenfunctions, and used to describe the characteristics of bottom reflection at small grazing angle.

Lonsdale, P. A., Tyce, R. C. and Spiess, F. N., 1974. "Near-bottom acoustic observations of abyssal topography and reflectivity," In: Physics of Sound in Marine Sediments, Lloyd Hampton (ed.), Plenum Press, New York, pp. 293-317.

Ma, Y., Voradan, V. V., and Voradan, V. V., 1984. "Application of Tuersky's multiple scattering formalism to a dense suspension of elastic particles in water," JASA, 75(2), 335-339.

Analytical expressions for phase velocity and attenuation are obtained. An alternative to the Waterman approach but asymptotic results are identical.

MacIsaac, R. R., and Dunsiger, A. D., 1977. "Ocean sediment properties using acoustic sensing," POAC 77, Proceedings, Vol. II, 1074-86.

Measure coherence on a ping to ping basis over various sediment types via (1) maximum value of the normalized cross-correlation function between two successive echoes, and (2) normalized cumulative energy function.

Mackenzie, K. V., 1961. "Bottom reverberation for 530- and 1030-cps sound in deep water," JASA, 33(11), 1498-1504.

Measure reverberation levels of specular and nonspecular reflection by using different pulse length at fixed frequency. The nonspecular reflection was assumed to obey the Lambert's law for deriving the bottom scattering strength.

McCave, I. N., "Suspended sediment," In: Estuarine Hydrography and Sedimentation, K. R. Dyer (ed.), Cambridge University Press, London, England.

McKinney, C. M., and Anderson, C. D., 1964. "Measurements of backscattering of sound from the ocean bottom," JASA, 36(1), 158-163.

Measure backscattering strength as functions of grazing angle, varying as (sin) or (sine square), and frequency, increasing as 1.6 power.

McLeroy, E. G., 1972. "Measurement and correlation of acoustic reflection and sediment properties off Panama City, Florida," Informal Report NCSL 112-72, Naval Coastal Systems Laboratory, Panama City, Florida.

Measure reflectivity amplitude and echo length of 12 kHz frequency pulse (1 msec) and predict them as functions of sediment parameters (water content, porosity, void ratio, bulk density, average specigravity of particles, percent fines, percent sand, and percent gravel) by regression analysis.

Meng, J., and Guan, D., 1984. "Acoustical method for remote sensing of seafloor sediment types," unpublished manuscripts.

Define and use two quantities, F1 and F2, to classify sediment types. F1 is extracted from the relative form of the echo envelope, and is closely related to the attenuation of sound in sediment. F2 is the correlation coefficient of echo with the source pulse, and depends on porosity and mean grain size of the sediment. Field experiment used frequency of 120 kHz for F1, and 1.8 to 8 kHz for F2.

Mikeska, E. E., and McKinney, C. M., 1978. "Range dependence of underwater echoes from randomly rough surfaces," JASA, 63(5), 1375-1380.

A tank experiment to measure peak pressure at normal incidence at 100 kHz as functions of range, relative roughness, and grazing angle.

Milligan, S. D., LeBlanc, L. R., and Middleton, F. H., 1978. "Statistical grouping of acoustic reflection profiles," JASA, 64(3), 795-807.

Apply multivariate statistics technique of principal component to analyze the variability in the shape and amplitude of the entire reflected waveform. Results can produce maps of sediemnt distribution when used in conjunction with cluster analysis.

Nafe, J. E., and Drake, C. L., 1963. "Physical properties of marine sediments," In: The Sea, Vol. 3, M.H. Hill (ed.), Interscience Publishers, New York.

List all physical properties and the equations that relate them. Describe methods of measurement and their results, including density, compressional velocity, Poisson's ratio, and thermal conductivity as a function of porosity.

Parker, W. R., and Kirby, R., 1977. "Fine sediment studies relevant to dredging practice and control," Second International Symposium on Dredging Technology, November 1977, Texas A&M University.

Plumley, W. J., and Davis, D. H., 1956. "Estimation of recent sediment size parameters from a triangular diagram," J. Sed. Petrol, 26(2), 140-155.

Classify sediment types by a three-component plot of sand, silt and clay percentages. There is a one-to-one correspondence between points on the triangular diagram and pairs of median and deviation measurement values, excluding the periphery of triangle.

Schmidt, P. B., 1971. "Monostatic and bistatic backscattering measurements from the deep ocean bottom," JASA, 50(1), 326-331.

Measure backscattering strength for both monostatic and bistatic cases as functions of frequency (0.5 to 6.3 kHz) and grazing angle. Results for grazing angle below 50 degrees can be approximated by Lambert's law of diffusion reflection.

Shepard, F. P., 1954. "Nomenclature based on sand-silt-clay ratios," J. Sed. Petrol, 24(3), 151-158.

Classify sediment types relative to sand, silt clay content by a triangle diagram. It applies only to sediment grade size, and not to sediments containing large percentages of gravel.

Stoll, R. D., 1974. "Acoustic waves in saturated sediments," In: Physics of Sound in Marine Sediments, Lloyd Hampton (ed.), Plenum Press, New York, 19-39.

Thorne, P. D., and Pace, N. G., 1984. "Acoustical studies of broadband scattering from a model rough surface," JASA, 75(1), 133-144.

Derives results equivalent to some of Tappert's calculations for rough surfaces. Also presents experimental data for scattering from prepared model surfaces.

Tindle, C. T., Guthrie, K. M., Bold, E. J., Johns, M. D., Jones, D., Dixon, K. O., and Birdsall, T. G., 1978. "Measurements of the frequency dependence of normal modes," JASA, 64(4), 1178-85.

Use excited normal mode amplitudes as a function of range to determine rate of attenuation of the medium.

Tsang, L., Kong, J. A., and Habashy, T., 1982. "Multiple scattering of acoustic waves by random distribution of discrete spherical scatterers with the Quasicrystalline and Percus-Yevick approximation," JASA, 71(3), 552-558.

A refinement of the basic Waterman multiple-scattering theory to give improved results for very dense suspensions.

Volovov, V. I., and Lysanov, Yu. P., 1973. "Correlation of sound signals reflected from the ocean bottom with variation of the transmission frequency," Soviet Physics-Acoustics, 19(3), 216-220.

Volovov, V. I., and Lysanov, Yu. P., 1973. "Spatial correlation of sound signals reflected from the ocean bottom," Soviet Physics-Acoustics, 19(1), 11-13.

Volovov, V. I., and Lysanov, Yu. P., 1969. "Correlation of the fluctuations of sound signals reflected from the ocean bottom," Soviet Physics-Acoustics, 15(2), 179-83.

Waterman, P. C., and Truell, R., 1961. "Multiple scattering of waves," J. Math. Phys., 2, 512-537.

This is the classic paper in the field of multiple scattering theory.

Williams, S. J., Carter, C. H., Meisburger, E. P., and Fuller, J. A., 1980. "Sand resources of southern Lake Erie, Conneant to Toledo, Ohio - a seismic reflection and vibracore study", MR 80-10, U.S. Army, Corps of Engineers, Coastal Engineering Research Center, Fort Belvoir, Virginia.

Classify sediment types by grain size (unified soils classification and Wentworth classification) through core sediments.

Yamamoto, T., 1983. "Acoustic propagation in the ocean with a poro-elastic bottom," JASA, 73(5), 1587-1596.

An application of Biot theory to sound propagation in the ocean.

Zhou, Ji-xun, Guan, Ding-hua, Shang, Er-chang, and Luo, En-shen, 1982. "Long-range reverberation and bottom scattering strength in shallow water," Acta Acoustica (in Chinese with English abstract).

Use the method of angular power spectrum to compute bottom scattering strength at long range in terms of grazing angle, frequency, and range (by introducing an additional multipath factor). Results for frequency of 0.8 to 4.0 kHz and grazing angle range of 10 degrees in the Yellow Sea, the Bohai Sea and the East China Sea show that the bottom scattering constant and angle index for high speed sediemnet of continental shelf depend not only on sea area and frequency, but also on angle range.

Appendix I

Instructions for running computer programs.

These instructions assume that you are using the AOML VAX system with UNIRAS software. Listings of the programs are included in Appendix II to facilitate conversion to other systems.

A. Instructions for Running the One-Layer Bottom Echo Program

Before running the one-layer bottom reflection model, set one of your own directories as default. It is necessary to run the program out of your own directory because the program creates output files on Fortran units 10 and 20, and you will have privileges to create files in the program directories.

The run command along with typical user inputs and program screen output is shown below:

```
$ run D5:[OAD.SEEM.BS.ONELAYER]ONELAYER

LAYER              1             *** LAYER              1
 Enter bottom roughness length parameter (M)
  0.1
 Enter the rms height of the bottom roughness (M)
  0.01
 Enter the depth of the sounding (M)
  60.
 Enter the beamwidth of the echo sounder (DEG)
  12.
 Enter the porosity of the sediment ( Range(0.,1.) )
  0.40
 Enter the radius of the sediment (M)
  2.e-4
 Enter the sound speed of the sediment (M/S)
  1740.
 Enter the density of the sediment (g/cm^-3)
```

```
 1.98
FORTRAN STOP
$
```

Alternatively you may create a command file in your own directory to run the program. The command file should be modeled after the following example:

```
$ run D5:[OAD.SEEM.ONELAYER]ONELAYER
0.1                ! Roughness length
0.01               ! Roughness height
60.                ! Depth
12.                ! Beamwidth
0.40               ! Porosity
2.e-4              ! Grain radius
1740.              ! Sound speed
1.98               ! Density
```

These inputs correspond to the comments on the right which are ignored by the program. The comments should be left in your command file, as they are a useful guide for changing the inputs for different runs with a minimum of confusion. To run your command file (if it is named COMFILE.COM), type @COMFILE. The computer screen response will be as follows:

```
LAYER             1          *** LAYER            1
 Enter bottom roughness length parameter (M)
 Enter the rms height of the bottom roughness (M)
 Enter the depth of the sounding (M)
 Enter the beamwidth of the echo sounder (DEG)
 Enter the porosity of the sediment( Range(0.,1.) )
 Enter the radius of the sediment (M)
 Enter the sound speed of the sediment (M/S)
 Enter the density of the sediment (g/cm-3)
FORTRAN STOP
```

The one layer model takes approximately 16 seconds of CPU time to run.

B. Data Files

You now have two files in your directory named FOR010.DAT and FOR020.DAT. FOR010.DAT is 66 blocks long. It contains unformatted output data. The unformatted output data is written to the file as 20 sets of two records. The first record is two words long. The first word contains the frequency (real variable) in Hz of upcoming intensity values. The second word contains the length of the second record (integer variable). This second word is presently set to 400, corresponding to 4 milliseconds of intensity values. The second record contains intensity values (real variables).

FOR020.DAT is 3 blocks long. It contains a single record of unformatted output data, 10 words in length described as follows:

WORD	TYPE	NAME	MEANING
1	INT	LAYER	For the one layer model this will always be one. This is here for compatability with the two layer model output.
2	REAL	ROUGH	The length parameter of the surface roughness in meters.
3	"	ETA	The height parameter of the surface roughness in meters.
4	"	RANGE	The depth of the water column in meters.
5	"	BEAM	The beamwidth in degrees.
6	"	BETA	The fractional porosity of the sediment. RANGE:(0.,1.)
7	"	RADIUS	The radius of the average grain in meters.
8	"	SEDSPD	The sound speed in the sediment in meters per second.
9	"	RHO	The density of the sediment in grams per cubic centimeter.
10	"	THICK	The thickness of the layer. Set to 0. here. It only has meaning for the two-layer model and is output for compatability.

C. Instructions for Running the Two-Layer Bottom Echo Program

Before running the two-layer bottom reflection model, set one of your own directories as default. It is necessary to run the program out of your own directory because the program creates output files on Fortran units 10 and 20, and you will have privileges to create files in the program directories. Commands, along with typical user inputs and program screen output, are shown below:

```
$ run D5:[OAD.SEEM.BS.TWOLAYER]TWOLAYER

LAYER                 1              *** LAYER              1
 Enter bottom roughness length parameter (M)
  0.1
 Enter the rms height of the bottom roughness (M)
  0.001
 Enter the depth of the sounding (M)
  50.
 Enter the beamwidth of the echo sounder (DEG)
  12.
 Enter the porosity of the sediment( Range(0.,1.) )
  0.95
 Enter the radius of the sediment (M)
  2.5e-5
 Enter the sound speed of the sediment (M/S)
  1515.
 Enter the density of the sediment (g/cm**3)
  1.30
 Enter the thickness of the layer of sediment (M)
  1.0

LAYER                 2              *** LAYER              2
 Enter bottom roughness length parameter (M)
  1.0
 Enter the rms height of the bottom roughness (M)
```

```
 0.001
Enter the porosity of the sediment( Range(0.,1.) )
 0.40
Enter the radius of the sediment (M)
 2.e-4
Enter the sound speed of the sediment (M/S)
 1740.
Enter the density of the sediment (g/cm-3)
 1.98
FORTRAN STOP
$
```

Alternatively, you may create a command file in your own directory to run the program. The command file should be modeled after the following example:

```
$ run D5:[OAD.SEEM.BS.TWOLAYER]TWOLAYER
 0.1            ! Roughness length
 0.001          ! Roughness height          FLUID MUD OVER MEDIUM SAND
 50.            ! Depth
 12.            ! Beamwidth
 0.95           ! Porosity
 2.5e-5         ! Grain radius
 1515.          ! Sound speed
 1.30           ! Density
 1.0            ! Thickness of top layer
 1.0            ! Roughness length
 0.001          ! Roughness height
 0.40           ! Porosity
 2.e-4          ! Grain radius
 1740.          ! Sound speed
 1.98           ! Density
```

These inputs crrepsond to the comments on the right which are ignored by the program. The comments should be left in your command file so that you can command file (if it is named COMFILE.COM), type @COMFILE. The computer screen response will be as follows:

LAYER	1	*** LAYER	1

```
Enter bottom roughness length parameter (M)
Enter the rms height of the bottom roughness (M)
Enter the depth of the sounding (M)
Enter the beamwidth of the echo sounder (DEG)
Enter the porosity of the sediment( Range(0.,1.) )
Enter the radius of the sediment (M)
Enter the sound speed of the sediment (M/S)
Enter the density of the sediment (g/cm**3)
Enter the thickness of the layer of sediment (M)
```

LAYER	2	*** LAYER	2

```
Enter bottom roughness length parameter (M)
Enter the rms height of the bottom roughness (M)
Enter the porosity of the sediment( Range(0.,1.) )
Enter the radius of the sediment (M)
Enter the sound speed of the sediment (M/S)
Enter the density of the sediment (g/cm**3)
FORTRAN STOP
$
```

The two layer model takes approximately 18 seconds of CPU time to run.

You now have two files in your directory named FOR010.DAT, and FOR020.DAT. FOR010.DAT is identical to the data file produced by the one-layer model. FOR020.DAT is still 3 blocks long (minimum file size), but it now has 20 words of unformatted output data. The first 10 words are for the top layer (layer 1), and the second 10 words are for layer two. See the one layer model for a description of the data. Since the beamwidth and the range remain the same for both layers, these values are set to 0. in the second ten words of output. Since the thickness refers to the top layer, this value is set to zero in the second ten words of output.

D. Instructions for Running the ECHOPLOT Program

The ECHOPLOT program gives a line plot on a TEKTRONIX screen of either intensity vs. frequency or intensity vs. time. The package used to create this plot is IGL. You must set one of your directories as default before running this program because it creates a file during execution called SEGFIL.DAT. This file is unimportant and may be deleted after you execute the program. The program reads from Fortran unit 10 which is the file that you created with either the one layer model or the two-layer model. If you have a file in your directory by the name of FOR010.DAT you may go on to execute the program. If your data file has another name, you can do the following to define Fortran unit 10.

```
$ DEF FOR010 [YOUR.DIRECTORY]YOUR.DATA
```

Before executing the program, there is a terminal characteristic that you must change. Most TEKTRONIX terminals reset to ANSI code when they are turned on. This prohibits the terminal from translating the plotting commands. To correct this problem, hit the SETUP key on the terminal and enter the following:

```
*CODE TEK
```

You are now ready to run the program. The run command, terminal responses and typical inputs are shown below:

```
$ RUN D5:[OAD.SEEM.BS]ECHOPLOT

Input "T" or "F" to hold Time or Frequency constant.

F
```

Input the array element to hold.

15

Give a plot heading.

FLUID MUD OVER MEDIUM SAND

At this point the screen goes blank and the plot is drawn.

If "T" or "t" is not input for the first response, frequency is assumed to be held constant. If it is time that you are holding constant, you may choose up to 400 array elements, corresponding to the first 4 milliseconds which are written by the one and two layer models. The time resolution is 10 microseconds, so array element 4 would be 40 microseconds. If it is frequency that you are holding constant, you may choose up to 20 array elements. The frequencies start at approximately 10 kHz and run up to 200 kHz in 10 kHz steps. The frequencies are not exactly at the 10 kHz intervals in order to prevent harmonics from occurring in other programs. The plot heading should be less than or equal to 30 characters.

After the plot has been drawn you are left with a dialog area which is 4 lines wide. You will also notice that if you hit the "G Eras" button on the terminal that the graph is erased but immediately comes back. This is because the plot has been drawn into hardware segment number 1. Some useful setup commands are listed below. You must first hit the "Setup" button to enter these commands. To get out of the "Setup" mode, hit the "Setup" button again.

```
*<ESC>SK 1          ! Kill segment number 1.  This erases the plot.
*SGV  1,N           ! This erases the segment but saves it.
*SGV  1,Y           ! This makes segment 1 visible again.
```

```
*SGR  1,2          ! Rename segment 1 to segment 2.  This is good to use if you
                     want to draw another plot into segment 1 but save the seg-
                     ment that is already there.
*DAB 300           ! Increase the dialog area buffer to 300 lines.
*DAL 32            ! Increase the dialog area to 32 lines.
*CODE  ANSI        ! Reset the command character interpretations so that you
                     may use the editor.
```

E. General Instructions for Running Contour Programs

There are four contour programs that may be run using the output of either the one or two-layer bottom reflection models. These programs are all written using the UNIRAS graphics package and must be run on a TEKTRONIX terminal or other graphics terminal for which a UNIRAS driver is available. In order to execute any of these programs, you must set your terminal up to run UNIRAS and you must have your data available.

To set your terminal up to run UNIRAS, input the following command:

```
$ UNIRAS
```

When this is done, several logical units and names are assigned, as shown below:

```
%DCL-I-SUPERSEDE, previous value of UNI_WORK has been superseded
  D5:[OAD.SEEM.UNIRAS]
%DCL-I-SUPERSEDE, previous value of UNIRAS has been superseded
%DCL-I-SUPERSEDE, previous value of HP7221 has been superseded
%DCL-I-SUPERSEDE, previous value of REGIS has been superseded
%DCL-I-SUPERSEDE, previous value of TEK41XX has been superseded
%DCL-I-SUPERSEDE, previous value of TEK4014 has been superseded
%DCL-I-SUPERSEDE, previous value of T4662 has been superseded
%DCL-I-SUPERSEDE, previous value of MAPDAT has been superseded
%DCL-I-SUPERSEDE, previous value of UNITEXT has been superseded
%DCL-I-SUPERSEDE, previous value of UNIWORK has been superseded
```

```
%DCL-I-SUPERSEDE, previous value of UNIINFO has been superseded
%DCL-I-SUPERSEDE, previous value of UNILIB has been superseded
%DCL-I-SUPERSEDE, previous value of LAYW has been superseded
%DCL-I-SUPERSEDE, previous value of LAYF has been superseded
```

If you do not have a UNIRAS default directory set up, the computer will ask you for a UNIRAS working directory. This directory must be one of yours that has already been created. UNIRAS creates several files as it draws plots so you must be operating in a directory in which you have privilege to create files.

UNIRAS writes a great deal of information to fortran unit 6 as it draws the plots. This information is generally of no importance, but if you do not redefine unit 6 to be a file, the information will go to the screen at the same time as the plot is being drawn, causing the terminal to become extremely confused. To redefine unit 6, input:

```
$ DEF FOR006 UNI.LIS
```

The four contour programs consist of two line-contour programs and two color-contour programs. One of each type draws a contour plot of a single output file from either the one or two layer bottom reflection model. The other of each type draws a set of three contour plots on a single screen for a more compact display.

F. Single Contour Plots

To draw a single contour plot, you must have the data files from a single run. These are output from the bottom reflection models as Fortran units 10 and 20 and are also input to the contour programs as the same Fortran units. If you do not have a file named for010.dat or one named for020.dat in your UNIRAS working directory, you must do the following:

```
$ DEF FOR010 [YOUR.DIRECTORY]YOUR.DATA
$ DEF FOR020 [YOUR.DIRECTORY]YOUR_OTHER.DATA
```

See the documentation on the one layer bottom reflection model for the data file specifications. The run statements and the terminal response is shown below:

```
$ RUN D5:[OAD.SEEM.UNIRAS]ONECON          ! Color contour program :: or
$ RUN D5:[OAD.SEEM.UNIRAS]ONELINE         ! Line contour program.
Enter plot title- TERMINATE $
FLUID MUD OVER MEDIUM SAND$
```

Note that you must terminate the plot title with a "$". The plot will now be drawn to the screen. If you hit the "G Eras" button on the terminal now, the plot will be erased and it will not reappear. If the plot appears not to be quite right, there is probably something wrong in the terminal setup parameters. Go into Setup mode and enter the following:

*FLAG IN/OUT

If you run a number of plots, you will gather quite a collection of junk files named UNI.LIS and UNISEG.DAT in your directory. Don't forget to PURGE them once in a while.

G. Three Contour Plots Per Screen

To run the programs that draw three contour plots, three data files are needed. The data files produced by the bottom reflection programs called FOR020.DAT are not used and may be deleted if desired. The bottom reflection program does need to be run three times to produce the data files for the contour programs. Each of these files when produced is called FOR010.DAT (or your defined equivalent). One of these must remain FOR010.DAT for the operation of the three contour program. The other two files should be assigned to Fortran units 11 and 12. One way to do this could be:

```
$ DEF FOR010 [YOUR.DIRECTORY]FOR010.DAT
$ DEF FOR011 [YOUR.DIRECTORY]FOR010.DAT.-1
$ DEF FOR012 [YOUR.DIRECTORY]FOR010.DAT.-2
```

The run statements and the terminal responses are shown below:

```
$ RUN D5:[OAD.SEEM.UNIRAS]THREECON       ! Color contours  :: or
$ RUN D5:[OAD.SEEM.UNIRAS]THREELINE      ! Line contours
Enter plot title- TERMINATE $
15 M DEPTH$
Enter plot title- TERMINATE $
30 M DEPTH$
Enter plot title- TERMINATE $
60 M DEPTH$
```

The plot will now be drawn on the screen. The same comments about terminal setup apply as for the single contour case.

Appendix II

COMPUTER PROGRAM LISTINGS

These Fortran program listings are presented in the same order as they are discused in Appendix I.

A. ONE LAYER BOTTOM ECHO PROGRAM

```
C  ONE LAYER *** Main Program *** ONE LAYER *** Main Program *** ONE LAYER


      Parameter        points = 1024, pi = 3.14159
      Dimension        IWK(11), AFREQ(20), OUTRAY(points)
      Complex          AFFT(points), A(points), B(points), C(points)

      Data AFREQ/ 11.e3, 19.e3, 31.e3, 40.15e3, 50.333e3, 61.2e3,
     #       69.e3, 79.555e3, 91.333e3, 100.5e3, 110.11e3,
     #       121.501e3, 131.e3, 139.e3, 151.e3, 159.e3, 169.5e3,
     #       181.e3, 190.266e3, 199.999e3 /

      Data NUMPTS / points /
      Data NUMOUT / 400    /
      Data MPOWER / 10     /
      Data DELTAT / 1.E-5  /          ! sampling rate

      Call GETINP( 1,      ROUGH,  ETA,    RANGE,  BEAM,
     #             BETA,   RADIUS, SEDSPD, RHO,    DUMMY )

C           Determine the reflection coefficients.

      Call REFLCT( RHO,    SEDSPD, BETA,
     #             1.027,  1500.,  0.0,   REFL01 )

      Call REFLCT( RHO,    SEDSPD, 0.0,
     #             1.027,  1500.,  0.0,   REFL11 )

C           Create a transducer signal of finite duration.  10 counts
C           corresponds to a .1 millisecond pulse.

      NSAMP = 10
      Call GETDAT( NSAMP, NUMPTS, AFFT )

C           Perform an FFT on the transmitted signal.

      Call CMPFFT( AFFT, MPOWER, IWK )
```

```
C           Iterate through the frequencies of interest

      Do 10 JFREQ = 1, 20
          FREQ = AFREQ(JFREQ)

          Call GETSRF( FREQ,    ROUGH,   ETA,   RANGE, BEAM,
     #                 REFL01,  DELTAT, NUMPTS,     B   )

C           The above is the surface scattering only.
C           Get the volume scattering.

          Call GETVOL( FREQ,   BETA,   RADIUS, SEDSPD,
     #                 REFL11, DELTAT, NUMPTS, C )

C           Add the surface and volume scattering functions.

          Do 40 I = 1, NUMPTS
              B(I) = B(I) + C(I)
40        Continue

C           We now have the response of the bottom to an impulse function.
C           We want to convolve this response with the transmitted signal.
C           Get the FFT of the original pulse.

          Do 50 I = 1, NUMPTS
             A(I) = AFFT(I)
50        Continue

C           Perform an FFT on the scattering response function.

          Call CMPFFT( B, MPOWER, IWK )

C           Multiply the transformed arrays of the transmitted signal
C           and the response function.

          Do 30 I = 1, NUMPTS
              B(I) = B(I) * A(I)
30        Continue

C           Do the inverse transform.

          Call FFT2C( B, MPOWER, IWK )
          Do 20 I = 1, NUMPTS
              B(I) = B(I) * NUMPTS
20        Continue
```

```
C           Get the magnitude and phase angle for UNIRAS output.

          Write(10) FREQ, NUMOUT
          Do 80 I = 1, NUMOUT
              S = Real( B(I) )
              T = Aimag( B(I) )
              OUTRAY(i) = Sqrt( S*S + T*T )
80        Continue
          Write(10) ( OUTRAY(i), i = 1, NUMOUT )

10      Continue

C           Finish

        Stop
        End
```

```
C         GETINP  ---ONE LAYER---  GETINP  ---ONE LAYER---  GETINP

      Subroutine GETINP( LAYER, ROUGH,  ETA,    RANGE, BEAM,
     #                   BETA,  RADIUS, SEDSPD, RHO,   THICK )

C          Get the parameters for the bottom scattering function.
C          There are two exponentially decreasing curves.  One is the
C          reflection decay curve which is relatively high in amplitude
C          and narrow in width.  It is dependent on the travel time(depth),
C          the frequency of the signal, and the mean slope and roughness of
C          the bottom.  The second curve is due to the volume scattering
C          properties of the bottom.  This is a much lower amplitude curve
C          with a much longer tail.  The volume scattering properties of the
C          bottom depend on the sediment type, the rigidity, and the porosity.
C          These are both impulse response functions.  To get the expected
C          pulse shape from a return due to a pulse of finite duration, they
C          must be convolved with the original pulse.

      Parameter  pi = 3.14159

      Print*,' LAYER ',LAYER,'          *** LAYER', LAYER
      Print*,' ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~',

C           SURFACE SCATTERING PARAMETERS
C           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      Print*,' Enter bottom roughness length parameter (M)'
      Read(5,*) ROUGH

      Print*,' Enter the rms height of the bottom roughness (M)'
      Read(5,*) ETA

      If( LAYER .eq. 1 ) Then
         Print*,' Enter the depth of the sounding (M)'
         Read(5,*) RANGE

         Print*,' Enter the beamwidth of the echo sounder (DEG)'
         Read(5,*) BEAM
         BEAM = BEAM * pi / 180.

      Endif
```

```
C            VOLUME SCATTERING PARAMETERS
C            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Print*,' Enter the porosity of the sediment( Range(0.,1.) )'
        Read(5,*) BETA

        Print*,' Enter the radius of the sediment (M)'
        Read(5,*) RADIUS

        Print*,' Enter the sound speed of the sediment (M/S)'
        Read(5,*) SEDSPD

        Print*,' Enter the density of the sediment (g/cm**3)'
        Read(5,*) RHO

c        If( LAYER .eq. 1 ) Then
c           Print*,' Enter the thickness of the layer of sediment (M)'
c           Read(5,*) THICK
c        Endif

c        If( LAYER .eq. 2 ) Then
c           RANGE = 0.
c           BEAM  = 0.
c           THICK = 0.
c        Endif
        THICK = 0.

        Write(20) LAYER,  ROUGH,  ETA,    RANGE,   BEAM,
     #           BETA,   RADIUS, SEDSPD, RHO,     THICK

        Return
        End
```

B. COMMON SUBROUTINES

These routines are used by both the one and two layer programs.

```
C           Set up a function for computation of the FFT.
C
      Subroutine CMPFFT( A, MPOWER, IWK )
C
      Parameter points = 1024,  power = 10      ! points = 2**power
C
      Dimension IWK(power + 1)
      Complex A(1)
C
      ANORM = points
C
      Do 10 i = 1, points
         A(i) = Conjg( A(i) )
10    Continue
C
      Call FFT2C( A, MPOWER, IWK )
C
      Do 20 i = 1, points
         A(i) = Conjg( A(i) ) / ANORM
20    Continue

      Return
      End
C                 This is input routine for the FFT program.
C
      Subroutine GETDAT( NSAMP, NUMPTS, A )
C
C                 Set up basic parameters for waveform.
C
      Complex      A(NUMPTS)
C
      Do 10 I = 1, NSAMP
           A(I) = (1., 0.)
10    Continue
C
      Do 20 I = NSAMP + 1, NUMPTS
          A(I) = ( 0.0, 0.0 )
20    Continue
C
      Return
      End
```

```
      Subroutine REFLCT( RHO2, C2, BETA2,
     #                   RHO1, C1, BETA1, REFL12 )

      parameter   rhowat = 1.027

      RHOTOP = ( 1.0 - BETA1 ) * RHO1 + BETA1 * rhowat
      RHOBOT = ( 1.0 - BETA2 ) * RHO2 + BETA2 * rhowat

      REFL12 = ( RHOBOT*C2 - RHOTOP*C1 ) / ( RHOBOT*C2 + RHOTOP*C1 )

      Return
      End

C               This is input routine for the impulse response
C               function for surface scattering.
C
      Subroutine GETSRF( FREQ,   ROUGH,   ETA,    RANGE, BEAM,
     #                   RFLCT,  DELTAT,  NUMPTS, A      )
C
C               Set up basic parameters for waveform.
C
      Parameter    cspeed = 1500.
      Parameter    pi = 3.1415927
      Double precision   DSEED
      Complex      A(NUMPTS)
      Data  DSEED/ 1.D0 /
C
C           Determine the wave number.
C
      WAVNUM = 2. * pi * FREQ / cspeed
C
C           Determine the Rayleigh parameter PHI.
C
      PHI = 2. * WAVNUM * ETA
C
C            Model the small scale geometric scattering.
C               Do the incoherent part first.
C
      PHITRM = 1. - Exp( - PHI**2 )
      RTAU = 2. * RANGE * ( ETA / ROUGH )**2 / cspeed
C
      Do 10 I = 1, NUMPTS
          A(I) =  RFLCT * (   DELTAT / RTAU * PHITRM *
     #              Exp( - I * DELTAT / RTAU )                 )
10    Continue
C
C             Determine the coherent part of the impulse response.
C
       A(1) = A(1) + RFLCT * Exp( - PHI**2 ) / ( 2. * BEAM )**2
C
      Return
      End
```

```
      Do 10 JFREQ = 1, 20
          FREQ = AFREQ(JFREQ)

C               LAYER 1 *** LAYER 1 *** LAYER 1 *** LAYER 1 *** LAYER 1
C               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          Call GETSRF( FREQ,    ROUGH1,   ETA1,   RANGE1, BEAM,
     #                 REFL01,  DELTAT,   NUMPTS, B      )

C           The above is the surface scattering only.
C           Get the volume scattering.

          Call GETVOL( FREQ,   BETA1,  RADIS1, SEDSP1,
     #                 REFL11, DELTAT, NUMPTS, C      )

C           Add the surface and volume scattering functions.

          Do 40 I = 1, NUMPTS
              B(I) = B(I) + C(I)
40        Continue

C               LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2
C               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          Call GETSRF( FREQ,    ROUGH2,   ETA2,   RANGE2, BEAM,
     #                 REFL12,  DELTAT,   NUMPTS, C      )

C           The above is the surface scattering only.
C           Get the volume scattering.

          Call GETVOL( FREQ,   BETA2,  RADIS2, SEDSP2,
     #                 REFL22, DELTAT, NUMPTS, D      )

C           Add the surface and volume scattering functions.

          Do 60 I = 1, NUMPTS
              C(I) = C(I) + D(I)
60        Continue
C               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C           We now have the response of the layers to an impulse function.
C           Perform FFT's on the scattering response functions and then
C           apply a time delay to layer two.

          Call CMPFFT( B, MPOWER, IWK )
          Call CMPFFT( C, MPOWER, IWK )

          DELAY = 2.0 * THICK / SEDSP1
          DELAY = DELTAT * AINT( DELAY / DELTAT )
          TOTALT = NUMPTS * DELTAT
          CONST = - 2.0 * pi * DELAY / TOTALT
```

C. TWO LAYER BOTTOM ECHO PROGRAM

```
C  TWO LAYER *** Main Program *** TWO LAYER *** Main Program *** TWO LAYER


      Parameter        points = 1024, pi = 3.14159, ptsdim = 1025
      Dimension        IWK(11), AFREQ(20), OUTRAY(ptsdim)
      Complex          AFFT(ptsdim), A(ptsdim), B(ptsdim)
      Complex             C(ptsdim), D(ptsdim)

      Data AFREQ/ 10.13e3, 19.7e3, 30.11e3, 40.15e3, 50.333e3, 61.2e3,
     #       69.e3, 79.555e3, 91.333e3, 100.5e3, 110.11e3,
     #       121.501e3, 131.e3, 139.e3, 151.e3, 159.e3, 169.5e3,
     #       181.e3, 190.266e3, 199.999e3 /

C          DELTAT = SAMPLING RATE, 2**MPOWER = points

      Data NUMPTS / points /
      Data NUMOUT / 400    /
      Data MPOWER / 10     /
      Data DELTAT / 1.E-5  /
      Data NSAMP  / 10     /

      Call GETINP( 1,      ROUGH1, ETA1,   RANGE1, BEAM,
     #             BETA1, RADIS1, SEDSP1, RHO1,   THICK )

      Call GETINP( 2,      ROUGH2, ETA2,   DUM1,  DUM2,
     #             BETA2, RADIS2, SEDSP2, RHO2,  DUM3   )
      RANGE2 = RANGE1 + THICK

C           Determine the reflection coefficients.

      Call REFLCT( RHO1, SEDSP1, BETA1,
     #            1.027, 1500.,  0.0,    REFL01 )
      Call REFLCT( RHO2, SEDSP2, BETA2,
     #             RHO1, SEDSP1, BETA1, REFL12 )
      Call REFLCT( RHO1, SEDSP1, 0.0,
     #            1.027, 1500.,  0.0,    REFL11 )
      Call REFLCT( RHO2, SEDSP2, 0.0,
     #            1.027, 1500.,  0.0,    REFL22 )

C           Create a transducer signal of finite duration.  10 counts
C           corresponds to a .1 millisecond pulse ( NSAMP )

      Call GETDAT( NSAMP, NUMPTS, AFFT )

C           Perform an FFT on the transmitted signal.

      Call CMPFFT( AFFT, MPOWER, IWK )

C           Iterate through the frequencies of interest
```

```
 Do 10 JFREQ = 1, 20
     FREQ = AFREQ(JFREQ)

C           LAYER 1 *** LAYER 1 *** LAYER 1 *** LAYER 1 *** LAYER 1
C           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     Call GETSRF( FREQ,    ROUGH1,   ETA1,   RANGE1, BEAM,
    #             REFL01,  DELTAT,   NUMPTS, B      )

C         The above is the surface scattering only.
C         Get the volume scattering.

     Call GETVOL( FREQ,   BETA1,  RADIS1, SEDSP1,
    #             REFL11, DELTAT, NUMPTS, C      )

C         Add the surface and volume scattering functions.

     Do 40 I = 1, NUMPTS
         B(I) = B(I) + C(I)
40   Continue

C           LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2
C           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     Call GETSRF( FREQ,    ROUGH2,   ETA2,   RANGE2, BEAM,
    #             REFL12,  DELTAT,   NUMPTS, C      )

C         The above is the surface scattering only.
C         Get the volume scattering.

     Call GETVOL( FREQ,   BETA2,  RADIS2, SEDSP2,
    #             REFL22, DELTAT, NUMPTS, D      )

C         Add the surface and volume scattering functions.

     Do 60 I = 1, NUMPTS
         C(I) = C(I) + D(I)
60   Continue
C           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C         We now have the response of the layers to an impulse function.
C         Perform FFT's on the scattering response functions and then
C         apply a time delay to layer two.

     Call CMPFFT( B, MPOWER, IWK )
     Call CMPFFT( C, MPOWER, IWK )

     DELAY = 2.0 * THICK / SEDSP1
     DELAY = DELTAT * AINT( DELAY / DELTAT )
     TOTALT = NUMPTS * DELTAT
     CONST = - 2.0 * pi * DELAY / TOTALT
```

```
      Call GETVOL( FREQ,   BETA1,  RADIS1, SEDSP1,
     #             REFL11, DELTAT, NUMPTS, C      )

C       Add the surface and volume scattering functions.

      Do 40 I = 1, NUMPTS
         B(I) = B(I) + C(I)
40    Continue

C        LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2 *** LAYER 2
C        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      Call GETSRF( FREQ,    ROUGH2,   ETA2,   RANGE2, BEAM,
     #             REFL12,  DELTAT,   NUMPTS, C       )

C       The above is the surface scattering only.
C       Get the volume scattering.

      Call GETVOL( FREQ,   BETA2,  RADIS2, SEDSP2,
     #             REFL22, DELTAT, NUMPTS, D      )

C       Add the surface and volume scattering functions.

      Do 60 I = 1, NUMPTS
         C(I) = C(I) + D(I)
60    Continue
C        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C       We now have the response of the layers to an impulse function.
C       Perform FFT's on the scattering response functions and then
C       apply a time delay to layer two.

      Call CMPFFT( B, MPOWER, IWK )
      Call CMPFFT( C, MPOWER, IWK )

      DELAY = 2.0 * THICK / SEDSP1
      DELAY = DELTAT * AINT( DELAY / DELTAT )
      TOTALT = NUMPTS * DELTAT
      CONST = - 2.0 * pi * DELAY / TOTALT
      Do 70 i = 1, NUMPTS/2
        C(i) = Cexp( Cmplx(0., (i-1) * CONST) ) * C(i)
        B(i) = B(i) + C(i)
70    Continue

      Do 90 i = NUMPTS, NUMPTS/2+1, -1
        C(i) = Cexp( Cmplx( 0., -(NUMPTS-i) * CONST ) ) * C(i)
        B(i) = B(i) + C(i)
90    Continue

C       We want to convolve this response with the transmitted signal.
C       Get the FFT of the original pulse.

      Do 50 I = 1, NUMPTS
        A(I) = AFFT(I)
50    Continue

C       Multiply the transformed arrays of the transmitted signal
C       and the response function.

      Do 30 I = 1, NUMPTS
         B(I) = B(I) * A(I)
```

```
        Do 70 i = 1, NUMPTS/2
           C(i) = Cexp( Cmplx(0., (i-1) * CONST) ) * C(i)
           B(i) = B(i) + C(i)
70      Continue

        Do 90 i = NUMPTS, NUMPTS/2+1, -1
           C(i) = Cexp( Cmplx( 0., -(NUMPTS-i) * CONST ) ) * C(i)
           B(i) = B(i) + C(i)
90      Continue

C         We want to convolve this response with the transmitted signal.
C         Get the FFT of the original pulse.

        Do 50 I = 1, NUMPTS
           A(I) = AFFT(I)
50      Continue

C         Multiply the transformed arrays of the transmitted signal
C         and the response function.

        Do 30 I = 1, NUMPTS
            B(I) = B(I) * A(I)
30      Continue

C         Do the inverse transform.

        Call FFT2C( B, MPOWER, IWK )
        Do 20 I = 1, NUMPTS
            B(I) = B(I) * NUMPTS
20      Continue

C         Get the magnitude and phase angle for UNIRAS output.

        Write(10) FREQ, NUMOUT
        Do 80 I = 1, NUMOUT
            S = Real( B(I) )
            T = Aimag( B(I) )
            OUTRAY(i) = Sqrt( S*S + T*T )
80      Continue
        Write(10)  ( OUTRAY(i), i = 1, NUMOUT )

10    Continue

C         Finish

      Stop
      End
```

```
C           GETINP  ---TWO LAYER---  GETINP  ---TWO LAYER---  GETINP

      Subroutine GETINP( LAYER, ROUGH,  ETA,    RANGE, BEAM,
     #                  BETA,  RADIUS, SEDSPD, RHO,   THICK )

C          Get the parameters for the bottom scattering function.
C          There are two exponentially decreasing curves.  One is the
C          reflection decay curve which is relatively high in amplitude
C          and narrow in width.  It is dependent on the travel time(depth),
C          the frequency of the signal, and the mean slope and roughness of
C          the bottom.  The second curve is due to the volume scattering
C          properties of the bottom.  This is a much lower amplitude curve
C          with a much longer tail.  The volume scattering properties of the
C          bottom depend on the sediment type, the rigidity, and the porosity.
C          These are both impulse response functions.  To get the expected
C          pulse shape from a return due to a pulse of finite duration, they
C          must be convolved with the original pulse.

      Parameter  pi = 3.14159

      Print*,' LAYER ',LAYER,'          *** LAYER', LAYER
      Print*,' ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~'

C            SURFACE SCATTERING PARAMETERS
C            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      Print*,' Enter bottom roughness length parameter (M)'
      Read(5,*) ROUGH

      Print*,' Enter the rms height of the bottom roughness (M)'
      Read(5,*) ETA

      If( LAYER .eq. 1 ) Then
         Print*,' Enter the depth of the sounding (M)'
         Read(5,*) RANGE

         Print*,' Enter the beamwidth of the echo sounder (DEG)'
         Read(5,*) BEAM
         BEAM = BEAM * pi / 180.

      Endif
```

```
C          VOLUME SCATTERING PARAMETERS
C          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      Print*,' Enter the porosity of the sediment( Range(0.,1.) )'
      Read(5,*) BETA

      Print*,' Enter the radius of the sediment (M)'
      Read(5,*) RADIUS

      Print*,' Enter the sound speed of the sediment (M/S)'
      Read(5,*) SEDSPD

      Print*,' Enter the density of the sediment (g/cm**3)'
      Read(5,*) RHO

      If( LAYER .eq. 1 ) Then
         Print*,' Enter the thickness of the layer of sediment (M)'
         Read(5,*) THICK
      Endif

      If( LAYER .eq. 2 ) Then
         RANGE = 0.
         BEAM  = 0.
         THICK = 0.
      Endif

      Write(20) LAYER,  ROUGH,  ETA,    RANGE,   BEAM,
     #          BETA,   RADIUS, SEDSPD, RHO,     THICK

      Return
      End
```

D. ECHOPLOT program

```
C   ***  ECHOPLOT  ***  ECHOPLOT  ***  ECHOPLOT  ***  ECHOPLOT  ***

      Character*1  HOLD
      Character*25 HVAR
      Character*30 HEAD

      Dimension FREQ(20), TIME(512), AMPTUD(512,20)
      Dimension ZVAL(512)

      NUMPTS = 400

C            Determine whether to hold time or frequency constant.

      Print*, ' Input "T" or "F" to hold Time or Frequency constant.'
      Read(5,400) HOLD
400   Format(A1)
      Print*, ' Input the array element to hold.'
      Read(5,*) IHOLD

C             Get a plot heading.

      Print*,' Give a plot heading.'
      Read(5,500) HEAD
500   Format( A30 )

C             Read the amplitude arrays arranged by frequency
C             in time successive order.

      DO 10 I = 1, 20
        Read(10) FREQ(I),NUM1
        FREQ(I) = FREQ(I) / 1000.
        Read(10) ( AMPTUD(J,I), J = 1, NUM1 )
10    Continue

      DO 11 I = 1, 20
          Do 12 J = 1, NUMPTS
              If( AMPTUD(J,I) .eq. 0.0 ) AMPTUD(J,I) = 1.e-6
12        Continue
11    Continue
      Do 15 I = 1, NUMPTS
          TIME(I) = I / 100.
15    Continue
```

```
C           Set up the intensity array for either time or freq.

      ZMAX = -99999.
      ZMIN =  99999.
      If( HOLD .eq. 'T' .or. HOLD .eq. 't' ) Then
          Do 50 I = 1, 20
              ZVAL(I) = 10. * ALOG10( AMPTUD( IHOLD, I ) )
              If( ZVAL(I) .gt. ZMAX ) ZMAX = ZVAL(I)
              If( ZVAL(I) .lt. ZMIN ) ZMIN = ZVAL(I)
              HVAR(1:7) = 'Time = '
              HVAR(12:25) = ' milliseconds'
              Write( HVAR(8:11), 300 ) TIME(IHOLD)
50        Continue
          XPMIN = -40.
          XMIN  =  10.
          XMAX  =  200.
          XPMAX =  220.
          XSTEP =  20.
300       Format( F4.2 )
      Else
          Do 55 I = 1, 200
              ZVAL(I) = 10. * ALOG10( AMPTUD( I, IHOLD ) )
              If( ZVAL(I) .gt. ZMAX ) ZMAX = ZVAL(I)
              If( ZVAL(I) .lt. ZMIN ) ZMIN = ZVAL(I)
              HVAR(1:12) = 'Frequency = '
              HVAR(16:25) = ' kHz      '
              IFREQ = FREQ(IHOLD)
              Write( HVAR(13:15), 600 ) IFREQ
55        Continue
          XPMIN = -0.4
          XMIN  =  0.0
          XMAX  =  2.0
          XPMAX =  2.2
          XSTEP =  0.2
600       Format( I3 )
      Endif
```

```
C              Clean up the limits for the intensity values.

      If( ZMAX .gt. 0. ) Then
          ZMAX = INT( ZMAX ) + 1.
      Else
          ZMAX = INT( ZMAX )
      Endif

      If( ZMIN .gt. 0. ) Then
          ZMIN = INT( ZMIN )
      Else
          ZMIN = INT( ZMIN ) - 1.
      Endif

      ZPMIN = ZMIN - ( ZMAX - ZMIN ) / 8.
      ZPMAX = ZMAX + ( ZMAX - ZMIN ) / 8.

C                Start Plot

      CALL GRSTRT( 4113, 1 )

C                 Text Fundamentals

      CALL TXWORL
      CALL TXPROP

      Call Window( XPMIN, XPMAX, ZPMIN, ZPMAX )

      CALL OPNSEG(1)

C                Do the X-axis

      Call Move( XMIN, ZMIN )
      Call Draw( XMAX, ZMIN )

      XFACT  = 100.
      ZFACT  = 60.
      TIKSIZ = ( ZPMAX - ZPMIN ) / 80.
      XTXSIZ = ( XPMAX - XPMIN ) / XFACT
      ZTXSIZ = ( ZPMAX - ZPMIN ) / ZFACT
      CALL TXSIZE( 0, XTXSIZ, ZTXSIZ )
      DO 80 XTICK = XMIN, XMAX, XSTEP
          CALL Move( XTICK, ZMIN )
          CALL Draw( XTICK, ZMIN - TIKSIZ )
          Call Move( XTICK-2.2*XTXSIZ, ZMIN-1.5*TIKSIZ-ZTXSIZ)
          IF( HOLD .eq. 'T' .or. HOLD .eq. 't' ) Then
              Call Inumbr( INT(XTICK), 3 )
          Else
              Call Rnumbr( XTICK, 1, 3 )
          Endif
80    END DO
```

```
      CALL TXSIZE( 0, 1.5*XTXSIZ, 1.5*ZTXSIZ )
      NUMCHR = 25
      XPOSIT = XMIN + (XMAX-XMIN)/2. - 0.6*(NUMCHR * XTXSIZ)
      CALL MOVE( XPOSIT, ZMAX )
      CALL TEXT( NUMCHR, HVAR )
      NUMCHR = 15
      XPOSIT = XMIN + (XMAX-XMIN)/2. - 0.6*(NUMCHR * XTXSIZ)
      CALL MOVE( XPOSIT, ZPMIN + (ZPMAX-ZPMIN)/100.)
      If( HOLD .eq. 'T' .or. HOLD .eq. 't' ) Then
          Call Text( 15, 'FREQUENCY (kHz)' )
      Else
          Call Text( 19, 'TIME (milliseconds)' )
      Endif
      Call Move( XPOSIT, ZPMAX - 1.5 * ZTXSIZ )
      Call Text( 30, HEAD )

C                Do the Z-Axis

      CALL MOVE( XMIN, ZMIN )
      CALL DRAW( XMIN, ZMAX )
      TIKSIZ = XSTEP / 10.
      CALL TXSIZE( 0, XTXSIZ, ZTXSIZ )
      DO 82 ZTICK = ZMIN, ZMAX, (ZMAX - ZMIN) / 8.
            CALL MOVE( XMIN, ZTICK )
            CALL DRAW( XMIN - TIKSIZ, ZTICK )
            CALL MOVE( XMIN-6.5*XTXSIZ-TIKSIZ, ZTICK-0.5*ZTXSIZ )
            CALL RNUMBR( ZTICK, 1, 5 )
82    CONTINUE
      NUMCHR = 14
      ZPOSIT = ZMIN + (ZMAX-ZMIN)/2. - 0.6*(NUMCHR * ZTXSIZ)
      CALL MOVE( XMIN-8.*XTXSIZ-2.*TIKSIZ, ZPOSIT)
      CALL TXANGL(90.)
      CALL TXSIZE( 0, 1.5*(ZPMAX-ZPMIN)/ZFACT, 1.5*XTXSIZ )
      CALL TEXT( 14, 'INTENSITY (dB)' )
      CALL TXANGL(0.)
C                Plot the curve.
      If( HOLD .eq. 'T' .or. HOLD .eq. 't' ) Then
          Call Move( FREQ(1), ZVAL(1) )
          Do 60 I = 1, 20
              Call Draw( FREQ(I), ZVAL(I) )
60        Continue
      Else
          Call Move( TIME(1), ZVAL(1) )
          Do 70 I = 1, 200
              Call Draw( TIME(I), ZVAL(I) )
70        Continue
      Endif
      CALL CLOSEG
      CALL GRSTOP
      STOP
      END
```

E. SINGLE LINE CONTOUR PLOT - ONELINE

```
C       ***  ONELINE  ***  ONELINE  ***  ONELINE  ***  ONELINE  ***


C          This program does a contour plot of the intensity for an
C          array of Frequency and Time values.

      Dimension     ZVALIN(400,20),   ZVALOT(200,20), WORK(4000)
      Dimension     FREQ(20),         ZCL(9)
      Character*40 LABEL

      Data ZCL/ -40., -35., -30., -25., -20., -15., -10., -5., 0. /

      Data XSTART, YSTART, PSIZE/ 22., 20., 100./
      Data XTRANS, YTRANS, YLABEL, X2TRAN / 105., 135., 1.22, 1.8 /
      Data HTCHAR, YCHST,  XCHAR,  XLEFT  / 4.5, 90., 5., 10. /

      Print*,' Enter plot title- TERMINATE $'
      Read(5,100) LABEL
100   Format( a40 )

C             Do the contour plot.

      Call Lt41xx
      Call Gopen
      Call GRPSIZ( XSIZMM, YSIZMM )

C             The following values are set if the terminal is a 4107.

      If( XSIZMM .eq. 240.00 ) Then
         XSTART = 27.
         YSTART = 15.
         YLABEL = 1.25
         PSIZE  = 65.
         XTRANS = 50.
         X2TRAN = 2.0
         YTRANS = 90.
         XLEFT  = 0.005
         XCHAR = 5.
         YCHST = 65.
         HTCHAR = 3.5
      Endif

      Call Glimit( 0., 4., 10., 200., 0., 0. )
      Call Gvport( XSTART, YSTART, 3.0*PSIZE, PSIZE )
      Call Gwbox( 3.0, 1., 0. )
      Call Gzcl( ZCL, 8, 0 )
      Call Gcona( 4., 0, 40., 3 )

      Call Gsegcr( 10 )
```

```
C          Read the values for the scattering.

      Do 10 I = 1, 20

C                Get the total scattering.

         Read(10) FREQ(i), NUMPTS
         Read(10)( ZVALIN(J,I), J = 1, NUMPTS )

         K = 1
         Do 40 J = 1, NUMPTS, 2
            ZVALOT(K,I) = 10.* ALOG10( ZVALIN(J,I) )
            K = K + 1
40       Continue
10    Continue

      Call Gcnr2v( ZVALOT, 200, 20, WORK, 4000 )
      Call Gchar( LABEL, 0.5*PSIZE, YLABEL*PSIZE, .08*PSIZE )

      If( XSIZMM .eq. 240.00 )
     #             Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )

      Call Gaxis( 1, 0.0, 1.0, 4.0, 'TIME (milliseconds)$' )
      If( XSIZMM .eq. 240.00 )
     #             Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )
      Call Gaxis( 2, 10., 20., 200., 'FREQUENCY (kHz)$' )
c      Call Gaxis2( 0, 'TIME (milliseconds)$', 'FREQUENCY (kHz)$' )

      Call Gsegcl( 10 )

85    Continue
      Read(20,END=86,ERR=86) LAYER,  ROUGH,  ETA,    RANGE,  BEAM,
     #               BETA,   RADIUS, SEDSPD, RHO,    THICK
      Call Gsegcr( LAYER )

      YCHAR = YCHST
      If( (THICK .gt. 0.001 .and. LAYER.eq.1) .or. LAYER.eq.2 )  Then
         RLAYER = LAYER
         Call Gchar( 'LAYER $', XCHAR, YCHAR, HTCHAR )
         Call Gnumb( RLAYER, 9999., YCHAR, HTCHAR, -1 )

         YCHAR = YCHAR - 1.5 * HTCHAR
         If( THICK .gt. 0.001 ) Then
            Call Gchar( 'WIDTH:  $', XCHAR, YCHAR, HTCHAR )
            Call Gnumb( THICK, 9999., YCHAR, HTCHAR, 1 )
            Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
         Endif
      Endif
```

```
YCHAR = YCHAR - 2.0 * HTCHAR
Call Gchar( ' SURFACE PROPERTIES$', XCHAR, YCHAR, HTCHAR )
YCHAR = YCHAR - 0.65 * HTCHAR
Call Gchar( ' -----------------$', XCHAR, YCHAR, HTCHAR )
Call Gchar( ' -----------------$', XCHAR-1., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'ROUGHNESS LENGTH:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( ROUGH, 9999., YCHAR, HTCHAR, 2 )
Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'ROUGHNESS HEIGHT:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( ETA, 9999., YCHAR, HTCHAR, 3 )
Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 2.0 * HTCHAR
Call Gchar( ' VOLUME PROPERTIES$', XCHAR, YCHAR, HTCHAR )
YCHAR = YCHAR - 0.65 * HTCHAR
Call Gchar( ' ----------------$', XCHAR, YCHAR, HTCHAR )
Call Gchar( ' ----------------$', XCHAR-1., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
BETA = BETA * 100.
Call Gchar( 'POROSITY:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( BETA, 9999., YCHAR, HTCHAR, 1 )
Call Gchar( ' %$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
RADIUS = 1000. * RADIUS
Call Gchar( 'SEDIMENT RADIUS:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( RADIUS, 9999., YCHAR, HTCHAR, 3 )
Call Gchar( ' mM$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'SEDIMENT SOUND SPEED:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( SEDSPD, 9999., YCHAR, HTCHAR, 0 )
Call Gchar( ' M/S$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'SEDIMENT DENSITY:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( RHO, 9999., YCHAR, HTCHAR, 2 )
Call Gchar( ' G/cM**3$', 9999., YCHAR, HTCHAR )
```

```
        IF( LAYER .eq. 1 ) Then
         YCHAR = YCHST + 5. *  HTCHAR
         XCHAR = XCHAR + 10. * HTCHAR
         Call Gchar( ' SYSTEM PARAMETERS$', XCHAR, YCHAR, HTCHAR )
         YCHAR = YCHAR - 0.65 * HTCHAR
         Call Gchar( ' -----------------$', XCHAR, YCHAR, HTCHAR )
         Call Gchar( ' -----------------$', XCHAR-1., YCHAR, HTCHAR )
         YCHAR = YCHAR - 1.5 * HTCHAR
         Call Gchar( 'SOUNDING DEPTH:  $', XCHAR, YCHAR, HTCHAR )
         Call Gnumb( RANGE, 9999., YCHAR, HTCHAR, 1 )
         Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
         YCHAR = YCHAR - 1.5 * HTCHAR
         Call Gchar( 'BEAMWIDTH:  $', XCHAR, YCHAR, HTCHAR )
         BEAM = BEAM * 180. / 3.14159
         Call Gnumb( BEAM, 9999., YCHAR, HTCHAR, 1 )
         Call Gchar( ' DEG$', 9999., YCHAR, HTCHAR )

      Endif

      Call Gsegcl( LAYER )
      Go To 85
86    Continue

      Call Gclear
      Call Gsegtr( 10, XLEFT, YTRANS, 1., 1., 0. )
      Call Gsegtr(  1, XTRANS, 0.005, 1., 1., 0. )
      If( LAYER .eq. 2 )
     #   Call Gsegtr( 2, X2TRAN * XTRANS, 0.005, 1., 1., 0. )

      Call Gsegwk(  2 )
      Call Gsegwk(  1 )
      Call Gsegwk( 10 )

C           Close up the plot.

      Call Gclose

      Stop
      End
```

F. SINGLE COLOR CONTOUR PLOT - ONECON

```
C       ***  ONECON  ***  ONECON  ***  ONECON  ***  ONECON  ***

C           This program does a contour plot of the intensity for an
C           array of Frequency and Time values.

      Dimension     ZVALIN(400,20),   ZVALOT(200,20)
      Dimension     FREQ(20),         ZCL(9)
      Dimension     KODE1(12),        KODE2(12),     KODE3(12)
      Character*40 LABEL

      Data ZCL/ -40., -35., -30., -25., -20., -15., -10., -5., 0. /

C         KODEx contains color definitions for special color scheme.

      Data KODE1/  220,  240,  260,  280,  300,    0,
     #              20,   40,   75,  100,  120,  140/

      Data KODE2/   50,   50,   50,   50,   50,   40,
     #              42,   44,   46,   48,   50,   50/

      Data KODE3/  100,  100,  100,  100,  100,  100,
     #             100,  100,  100,  100,  100,  100/

      Data LABEL/ '0.5 M FLUID MUD$   '/

      Data XSTART, YSTART, PSIZE,  XMTR   / 22., 20., 100., -10. /
      Data XTRANS, YTRANS, YLABEL, X2TRAN / 105., 135., 1.22, 1.8 /
      Data HTCHAR, YCHST,  XCHAR,  XLEFT/ 4.5, 90., 5., 10./

      Print*,' Enter plot title- TERMINATE $'
      Read(5,100) LABEL
100   Format( a40 )

C              Do the contour plot.

      Call Lt41xx
      Call Gopen
      Call GRPSIZ( XSIZMM, YSIZMM )
```

```
C             Following values are set if the terminal is a 4107.

      If( XSIZMM .eq. 240.00 ) Then
         XSTART = 27.
         YSTART = 15.
         YLABEL = 1.25
         PSIZE  = 65.
         XTRANS = 50.
         X2TRAN = 2.0
         YTRANS = 90.
         XLEFT  = 0.005
         XMTR   = -20.
         XCHAR = 5.
         YCHST = 65.
         HTCHAR = 3.5
      Endif

      Call Glimit( 0., 4., 10., 200., 0., 0. )
      Call Gvport( XSTART, YSTART, 3.0*PSIZE, PSIZE )
      Call Gwbox( 3.0, 1., 0. )
      Call Gzcl( ZCL, 8, 0 )

      Call Gsegcr( 10 )

C          Read the values for the scattering.

      Do 10 I = 1, 20

C                Get the total scattering.

         Read(10) FREQ(i), NUMPTS
         Read(10)( ZVALIN(J,I), J = 1, NUMPTS )

         K = 1
         Do 40 J = 1, NUMPTS, 2
C         do 40 j = 1, numpts/2
            ZVALOT(K,I) = 10.* ALOG10( ZVALIN(J,I) )
            K = K + 1
40       Continue
10    Continue
```

```
      Call Gcnr2s( ZVALOT, 200, 20 )
      Call Gchar( LABEL, 0.5*PSIZE, YLABEL*PSIZE, .08*PSIZE )

      If( XSIZMM .eq. 240.00 )
     #             Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )

      Call Gaxis( 1, 0.0, 1.0, 4.0, 'TIME (milliseconds)$' )
      If( XSIZMM .eq. 240.00 )
     #             Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )
      Call Gaxis( 2, 10., 20., 200., 'FREQUENCY (kHz)$' )
c      Call Gaxis2( 0, 'TIME (milliseconds)$', 'FREQUENCY (kHz)$' )

      Call Gsegcl( 10 )

      Call Gsegcr( 3 )
      Call Gscsco( 'Below$Above$$', 0.08*PSIZE, 1, 0, 1 )
      Call Gcoscl( XSTART, YSTART )
      Call Gchar( 'INTENSITY (db)$', XCHAR+15., 1.1*PSIZE, HTCHAR )
      Call Gsegcl( 3 )

85    Continue
      Read(20,END=86,ERR=86) LAYER,  ROUGH,  ETA,    RANGE,  BEAM,
     #               BETA,   RADIUS, SEDSPD, RHO,    THICK
      Call Gsegcr( LAYER )

      YCHAR = YCHST
      If( (THICK .gt. 0.001 .and. LAYER.eq.1) .or. LAYER.eq.2 )  Then
         RLAYER = LAYER
         Call Gchar( 'LAYER $', XCHAR, YCHAR, HTCHAR )
         Call Gnumb( RLAYER, 9999., YCHAR, HTCHAR, -1 )

         YCHAR = YCHAR - 1.5 * HTCHAR
         If( THICK .gt. 0.001 ) Then
            Call Gchar( 'WIDTH:  $', XCHAR, YCHAR, HTCHAR )
            Call Gnumb( THICK, 9999., YCHAR, HTCHAR, 1 )
            Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
         Endif
      Endif
```

```
YCHAR = YCHAR - 2.0 * HTCHAR
Call Gchar( ' SURFACE PROPERTIES$', XCHAR, YCHAR, HTCHAR )
YCHAR = YCHAR - 0.65 * HTCHAR
Call Gchar( ' -----------------$', XCHAR, YCHAR, HTCHAR )
Call Gchar( ' -----------------$', XCHAR-1., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'ROUGHNESS LENGTH:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( ROUGH, 9999., YCHAR, HTCHAR, 2 )
Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'ROUGHNESS HEIGHT:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( ETA, 9999., YCHAR, HTCHAR, 3 )
Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 2.0 * HTCHAR
Call Gchar( ' VOLUME PROPERTIES$', XCHAR, YCHAR, HTCHAR )
YCHAR = YCHAR - 0.65 * HTCHAR
Call Gchar( ' -----------------$', XCHAR, YCHAR, HTCHAR )
Call Gchar( ' -----------------$', XCHAR-1., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
BETA = BETA * 100.
Call Gchar( 'POROSITY:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( BETA, 9999., YCHAR, HTCHAR, 1 )
Call Gchar( ' %$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
RADIUS = 1000. * RADIUS
Call Gchar( 'SEDIMENT RADIUS:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( RADIUS, 9999., YCHAR, HTCHAR, 3 )
Call Gchar( ' mM$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'SEDIMENT SOUND SPEED:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( SEDSPD, 9999., YCHAR, HTCHAR, 0 )
Call Gchar( ' M/S$', 9999., YCHAR, HTCHAR )
YCHAR = YCHAR - 1.5 * HTCHAR
Call Gchar( 'SEDIMENT DENSITY:  $', XCHAR, YCHAR, HTCHAR )
Call Gnumb( RHO, 9999., YCHAR, HTCHAR, 2 )
Call Gchar( ' G/cM**3$', 9999., YCHAR, HTCHAR )
```

```
      If( LAYER .eq. 1 ) Then
         YCHAR = YCHST + 5. *  HTCHAR
         XCHAR = XCHAR + 10. * HTCHAR
         Call Gchar( ' SYSTEM PARAMETERS$', XCHAR, YCHAR, HTCHAR )
         YCHAR = YCHAR - 0.65 * HTCHAR
         Call Gchar( ' -----------------$', XCHAR, YCHAR, HTCHAR )
         Call Gchar( ' -----------------$', XCHAR-1., YCHAR, HTCHAR )
         YCHAR = YCHAR - 1.5 * HTCHAR
         Call Gchar( 'SOUNDING DEPTH:  $', XCHAR, YCHAR, HTCHAR )
         Call Gnumb( RANGE, 9999., YCHAR, HTCHAR, 1 )
         Call Gchar( ' M$', 9999., YCHAR, HTCHAR )
         YCHAR = YCHAR - 1.5 * HTCHAR
         Call Gchar( 'BEAMWIDTH:  $', XCHAR, YCHAR, HTCHAR )
         BEAM = BEAM * 180. / 3.14159
         Call Gnumb( BEAM, 9999., YCHAR, HTCHAR, 1 )
         Call Gchar( ' DEG$', 9999., YCHAR, HTCHAR )

      Endif

      Call Gsegcl( LAYER )
      Go To 85
86    Continue

      Call Gclear
      Call Gsegtr(  3, XMTR,    0.005, 1., 1., 0. )
      Call Gsegtr( 10, XLEFT, YTRANS, 1., 1., 0. )
      Call Gsegtr(  1, XTRANS, 0.005, 1., 1., 0. )
      If( LAYER .eq. 2 )
     #   Call Gsegtr( 2, X2TRAN * XTRANS, 0.005, 1., 1., 0. )

      Call Gsegwk(  3 )
      Call Gsegwk(  2 )
      Call Gsegwk(  1 )
      Call Gsegwk( 10 )

      Call Gcmode( 4, IAMODE )
      Call Gcolor( 2, KODE1, KODE2, KODE3, 12 )

C           Close up the plot.

      Call Gclose

      Stop
      End
```

G. THREE LINE CONTOUR PLOT _ THREELINE

```
C  ***  THREELINE  ***  THREELINE  ***  THREELINE  ***  THREELINE  ***


C           This program does a contour plot of the intensity for an
C           array of Frequency and Time values.

      Dimension     ZVALIN(400,20),    ZVALOT(150,20), WORK(4000)
      Dimension     FREQ(20),          ZCL(9)
      Character*40 LABEL(10:12)

      DATA ZCL/  -40., -35., -30., -25., -20., -15., -10., -5., 0. /

      Data XSTART, YSTART, PSIZE,  XFACT/  22., 20., 100.,  0.90/
      Data XLEFT,  XRITE,  YTRANS, YLABEL/ 10., 177., 135., 1.22/

      Print*,' Enter the label for the first plot. TERMINATE--$'
      Read(5,100) LABEL(10)
      Print*,' Enter the label for the second plot. TERMINATE--$'
      Read(5,100) LABEL(11)
      Print*,' Enter the label for the third plot. TERMINATE--$'
      Read(5,100) LABEL(12)
100   Format(a40)

C              Do the contour plot.

      Call Lt41xx
      Call Gopen
      Call GRPSIZ( XSIZMM, YSIZMM )
      If( XSIZMM .eq. 240.00 ) Then
         XSTART = 17.
         YSTART = 15.
         YLABEL = 1.25
         PSIZE  = 65.
         XLEFT  = 0.005
         XRITE  = 124.
         YTRANS = 90.
         XFACT  = 1.0
      Endif
```

```
      Call Glimit( 0., 3., 10., 200., 0., 0. )
      Call Gvport( XSTART, YSTART, 1.5*PSIZE, PSIZE )
      Call Gwbox( 1.5, 1., 0. )
      Call Gzcl( ZCL, 8, 0 )
      CALL GCONA( 4.0, 0, 40., 3 )

      Do 30 KREAD = 10, 12
         Call Gsegcr( KREAD)

C           Read the values for the scattering.

         Do 10 I = 1, 20

C                 Get the total scattering.

            Read(KREAD) FREQ(i), NUMPTS
            Read(KREAD)( ZVALIN(J,I), J = 1, NUMPTS )

            K = 1
            Do 40 J = 1, 300, 2
               ZVALOT(K,I) = 10.* ALOG10( ZVALIN(J,I) )
               K = K + 1
40          Continue
10       Continue

         Call Gcnr2v( ZVALOT, 150, 20, WORK, 4000 )
         Call Gchar( LABEL(KREAD), 0.25*PSIZE, YLABEL*PSIZE, .06*PSIZE )

         If( XSIZMM .eq. 240.00 )
     #            Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )

         Call Gaxis2( 0, 'TIME (milliseconds)$', 'FREQUENCY (kHz)$' )

         Call Gsegcl( KREAD)

30    Continue
```

```
      Call Gclear
      Call Gchar( 'CONTOUR LINES ARE$', 0.4*PSIZE,
     #              0.8*PSIZE, 0.06*PSIZE )
      Call Gchar( ' INTENSITY IN db$', 0.4*PSIZE,
     #              0.7*PSIZE, 0.06*PSIZE )
      Call Gsegtr( 11, XRITE, YTRANS, XFACT, 1., 0. )
      Call Gsegtr( 12, XRITE,   .005, XFACT, 1., 0. )
      Call Gsegtr( 10, XLEFT, YTRANS, XFACT, 1., 0. )
      Call Gsegwk( 10 )
      Call Gsegwk( 11 )
      Call Gsegwk( 12 )

C           Close up the plot.

      Call Gclose

      Stop
      End
```

H. THREE COLOR CONTOUR PLOT _ THREECON

```
C     ***  THREECON  ***  THREECON  ***  THREECON  ***  THREECON  ***


C          This program does a contour plot of the intensity for an
C          array of Frequency and Time values.

      Dimension    ZVALIN(400,20),   ZVALOT(150,20)
      Dimension    FREQ(20),         ZCL(9)
      Dimension    KODE1(12),        KODE2(12),     KODE3(12)
      Character*40 LABEL(10:12)

      DATA ZCL/  -40., -35., -30., -25., -20., -15., -10., -5., 0. /

C           KODEx values are for color values for special color scheme.

      Data KODE1/ 220,  240,  260,  280,  300,    0,
     #             20,   40,   75,  100,  120,  140/

      Data KODE2/  50,   50,   50,   50,   50,   40,
     #             42,   44,   46,   48,   50,   50/

      Data KODE3/ 100,  100,  100,  100,  100,  100,
     #            100,  100,  100,  100,  100,  100/

      Data XSTART, YSTART, PSIZE,  XFACT/  22., 20., 100.,  0.90/
      Data XLEFT,  XRITE,  YTRANS, YLABEL/ 10., 177., 135., 1.22/

      Print*,' Enter the label for the first plot. TERMINATE--$'
      Read(5,100) LABEL(10)
      Print*,' Enter the label for the second plot. TERMINATE--$'
      Read(5,100) LABEL(11)
      Print*,' Enter the label for the third plot. TERMINATE--$'
      Read(5,100) LABEL(12)
100   Format(a40)
```

```
C             Do the contour plot.

      Call Lt41xx
      Call Gopen
      Call GRPSIZ( XSIZMM, YSIZMM )

C             Following values are set if the terminal is a 4107.

      If( XSIZMM .eq. 240.00 ) Then
         XSTART = 17.
         YSTART = 15.
         YLABEL = 1.25
         PSIZE  = 65.
         XLEFT  = 0.005
         XRITE  = 124.
         YTRANS = 90.
         XFACT  = 1.0
      Endif

      Call Glimit( 0., 3., 10., 200., 0., 0. )
      Call Gvport( XSTART, YSTART, 1.5*PSIZE, PSIZE )
      Call Gwbox( 1.5, 1., 0. )
      Call Gzcl( ZCL, 8, 0 )

      Do 30 KREAD = 10, 12
         Call Gsegcr( KREAD)

C          Read the values for the scattering.

         Do 10 I = 1, 20

C                Get the total scattering.

            Read(KREAD) FREQ(i), NUMPTS
            Read(KREAD)( ZVALIN(J,I), J = 1, NUMPTS )

            K = 1
            Do 40 J = 1, 300, 2
               ZVALOT(K,I) = 10.* ALOG10( ZVALIN(J,I) )
               K = K + 1
40          Continue
10       Continue
```

```
         Call Gcnr2s( ZVALOT, 150, 20 )
         Call Gchar( LABEL(KREAD), 0.25*PSIZE, YLABEL*PSIZE, .06*PSIZE )

         If( XSIZMM .eq. 240.00 )
     #              Call Gaxlab( .06*PSIZE, .05*PSIZE, 0, 0 )

         Call Gaxis2( 0, 'TIME (milliseconds)$', 'FREQUENCY (kHz)$' )

         Call Gsegcl( KREAD)

30    Continue

      Call Gsegcr( 3 )
      Call Gchar( 'INTENSITY (db)$', 20., PSIZE+10., .05*PSIZE )
      Call Gscsco( 'Below$Above$$', 0.08*PSIZE, 1, 0, 1 )
      Call Gcoscl( XSTART, YSTART )
      Call Gsegcl( 3 )
      Call Gclear
      Call Gsegtr(  3, XLEFT,  0.005, XFACT, 1., 0. )
      Call Gsegtr( 11, XRITE, YTRANS, XFACT, 1., 0. )
      Call Gsegtr( 12, XRITE,   .005, XFACT, 1., 0. )
      Call Gsegtr( 10, XLEFT, YTRANS, XFACT, 1., 0. )
      Call Gsegwk(  3 )
      Call Gsegwk( 10 )
      Call Gsegwk( 11 )
      Call Gsegwk( 12 )

      Call Gcmode( 4, IAMODE )
      Call Gcolor( 2, KODE1, KODE2, KODE3, 12 )

C           Close up the plot.

      Call Gclose

      Stop
      End
```