SAND 75-0650
Unlimited Distribution

EVALUATION OF CONSTITUTIVE PROPERTIES FROM VELOCITY INTERFEROMETER DATA

D. E. Grady and E. G. Young

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115 and Livermore, California 94550 for the United States Energy Research and Development Administration under Contract AT (29-1)-789

Printed August 1976

Sandia Laboratories

SF 2900 Q(7-73)

When printing a copy of any digitized SAND Report, you are required to update the markings to current standards.

Issued by Sandia Laboratories, operated for the United States Energy Research & Development Administration by Sandia Corporation.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SF 1004-DF (3-75)

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $ 5.50 ; Microfiche $2.25

SAND 75-0650
Unlimited Distribution
Printed August 1976

EVALUATION OF CONSTITUTIVE PROPERTIES FROM VELOCITY INTERFEROMETER DATA

D. E. Grady and E. G. Young
Geomechanics Research Division 5163
Sandia Laboratories
Albuquerque, New Mexico 87115

ABSTRACT

Accurate constitutive relations are required to predict the transient response of rock and soil media in a number of engineering and geophysical applications. Diffuse velocity interferometry has been specialized to this task. In this work, we report a method of analysis which accounts for the laser window-sample material impedance mismatch in determining material constitutive properties and describe two computer programs which have been developed to provide for rapid evaluation of these properties.

I. INTRODUCTION

There is currently a need for constitutive models which are capable of predicting the transient dynamic response of rock and soil media in a number of engineering and geophysical applications. Controlled plate impact experiments on samples of geological media provide one technique for determining the experimental data base necessary to develop adequate models. Attractive features of this approach are; (1) a controlled laboratory situation which allows measurement of the input loading pulse and subsequent evolution of the deformational stress wave and; (2) the fact that the experiment is dynamic, consistent with most of the application requirements.

Instrumentation capable of providing accurate data during a plate impact test has evolved over the past decade. At present, transducer techniques which measure stress-time or particle velocity-time histories have been the most successful. These include piezoresistant,[1] piezoelectric,[2] magnetic induction,[3] and laser interferometry techniques.[4] A notable advance has been the recent development of diffuse surface laser interferometry[5] which does not have the severe optical reflectivity requirements of earlier laser systems. Development of this very accurate instrumentation capability has made feasible the study of transient wave propagation in rock and soil media, and several recent studies on rocks have been conducted in this laboratory.[6,7] A particularly attractive feature has been the capability of making accurate relief wave studies which has been a limiting problem with other transducer techniques.

A critical problem in controlled impact studies has been the difficulty in relating the large amplitude transient wave propagation data to the stress-

strain constitutive properties and to the physical effects underlying the observed propagation characteristics. The measured stress or velocity profiles are only indirectly related to these properties and methods must be developed to effectively analyze the data. This problem has been considered by a number of authors.[8,9,10,11]

A further difficulty occurs with velocity interferometer techniques where velocity is measured at an interface between the sample under test and a laser window material. Since the window material is never an exact impedance match for the test material, a region of wave interaction is created in the test material near the sample-window interface and the measured interface velocity is a distorted version of the profile originally propagating in the undisturbed test medium. This sample-window impedance mismatch also complicates the problem of relating the wave propagation data to the rock material properties.

The present effort was undertaken to develop a systematic method for reducing velocity interferometer data to determine constitutive rock properties. A computer program, VISAR, has been developed by Barker[12] to determine particle velocity profiles from the raw interferometer fringe data. In the present work two additional programs, MODIFY and IMP have been developed to obtain loading and unloading stress-strain, stress-particle velocity, and material moduli data for the test material from the measured particle velocity profiles. These programs were developed for the PDP-10 computer system and make full use of the interactive graphics feature of this system.

In this report we first outline the experimental conditions upon which the method of analysis is based. We then provide the mathematical background and assumptions necessary to treat the problem of wave interaction

due to the sample-window impedance mismatch. In many rock materials of interest, the dispersive and dissipative character of the medium makes it difficult to design experiments in which the loading wave and the overtaking relief wave do not interact. Wave attenuation, therefore, results and introduces special problems in the analysis. These problems are discussed in the succeeding section. In the remaining sections an operational description of the analysis programs is provided.

II. EXPERIMENTAL CONDITIONS

The experimental methods which have been employed in this laboratory to investigate the dynamic properties of rock and soil media under conditions of plate impact are illustrated in Fig. 1. A 10 cm diameter light gas gun is used to impel flat-nosed aluminum projectiles at the target media. Projectiles are faced with thin plates of mechanically well known impact material which in turn are backed with low impedance solid foam. Fused quartz has been the standard impact material in most studies. Impact on the target material provides a step input stress wave in the sample material followed by an unloading wave originating at the impact plate-solid foam interface. Projectile velocities, ranging from 0.01 to 1 km/s, can be controlled within about 2 percent and impact planarity can be maintained within approximately one milliradian. The projectile velocity is measured by offset pins to within 0.2 percent and impact time at center of impact is measured to within about 10 ns using an offset impact fiducial pin and a planarity measurement with coplanar flush pins.

A laser window material is mounted on the back surface of the sample specimen. Although fuzed quartz, plexiglas, and sapphire have been calibrated for laser window materials,[13] only fused quartz has been used in the current study of rock properties to date. The window material surface which is to be placed in intimate contact with the sample surface is first vapor deposited with silver or aluminum to provide a diffusely reflecting surface. Upon emergence of the impact-produced stress wave at the sample-window interface, the particle velocity is continuously and accurately recorded through the complete history of loading and unloading with the diffuse surface velocity interferometer.

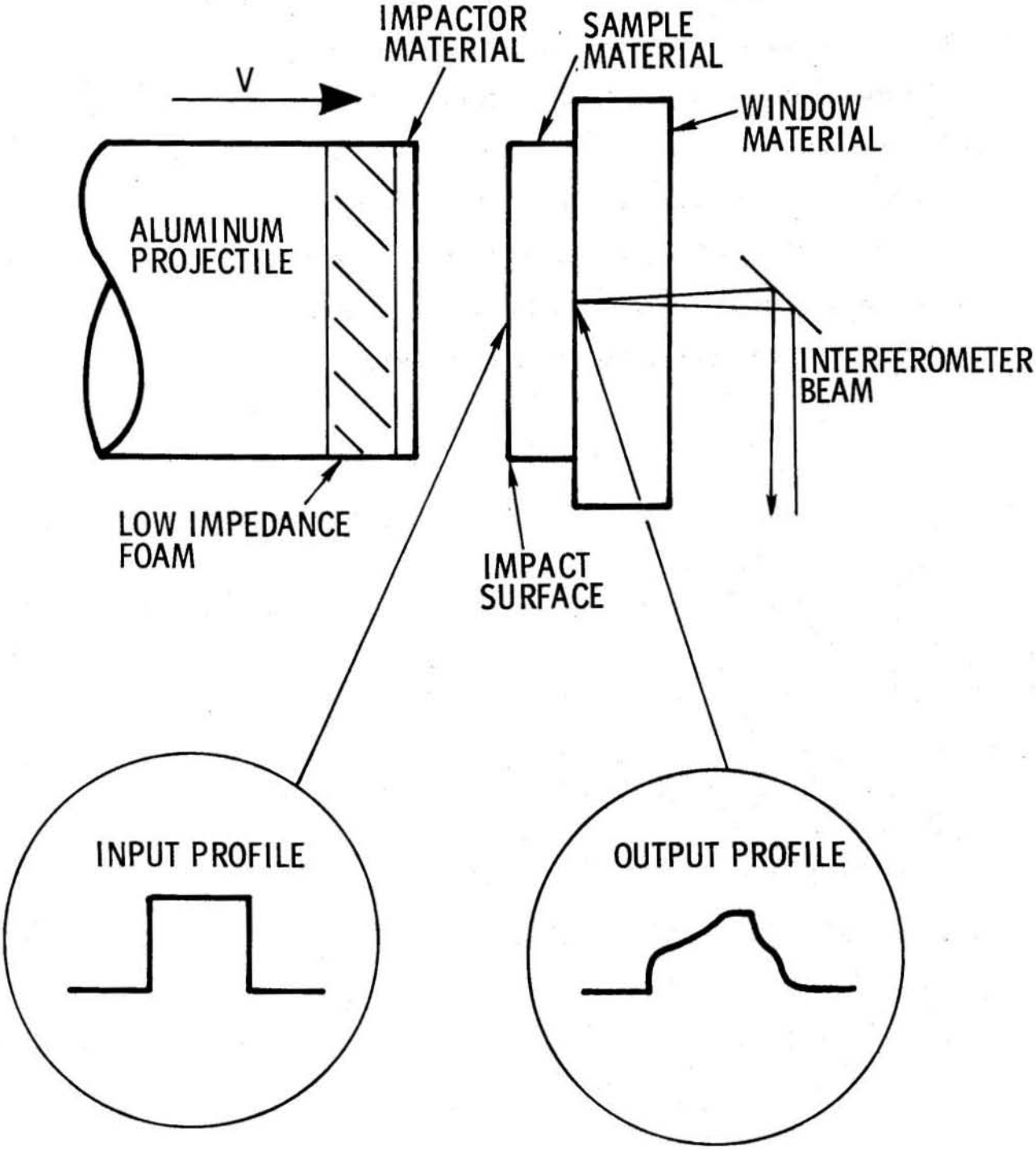

Fig. 1: EXPERIMENTAL CONFIGURATION FOR STUDY OF CONSTITUTIVE PROPERTIES OF ROCKS.

A series of tests on one rock material consisted of selecting a number of impact amplitudes within the range of interest and conducting several experiments at each impact amplitude. Sample thicknesses vary, usually ranging between 5 and 25 mm. By this procedure we can determine the evolution of the wave propagation which will be determined by the particular dispersive and dissipative properties of the rock under test. A set of particle velocity profiles obtained in such a series of tests on Solenhofen limestone are illustrated in Fig. 9 of this text.

III. ANALYTIC BASIS

In. Fig. 2 an x-t diagram for a typical plate slap experiment using a window material is shown. The test material is impacted from the left producing a simple loading stress wave propagating to the right in the sample material and to the left in the impactor material. The wave in the impactor material reflects off of the foam-impactor interface producing a right-going unloading wave which enters the sample material at the impact interface. The impactor plate, therefore, produces a complete loading and unloading stress pulse which propagates to the right in the test material and emerges at the sample-window interface where the resulting velocity profile is measured with laser interferometry. As was mentioned in the previous section, experiments using several sample thicknesses are conducted so the propagation velocities and dispersion characteristics of the wave pulse can be determined. Since the window material is never an exact impedance match for the test material, a region of wave interaction is created in the sample to the left of the sample-window interface and the measured velocity profile is a distorted version of the input profile, depending on the coupled mechanical properties of the window material and the unknown sample material. The objective is to subtract out properties of the window material so that the constitutive properties of the test material can be determined from the experimental wave profiles.

The method we have used to account for the interface mismatch has been to correct the distorted profiles and obtain the profiles which would have been measured at the sample-window interface if no impedance mismatch occurred. These profiles are then used to determine the stress-strain and stress-particle velocity properties of the test material. This

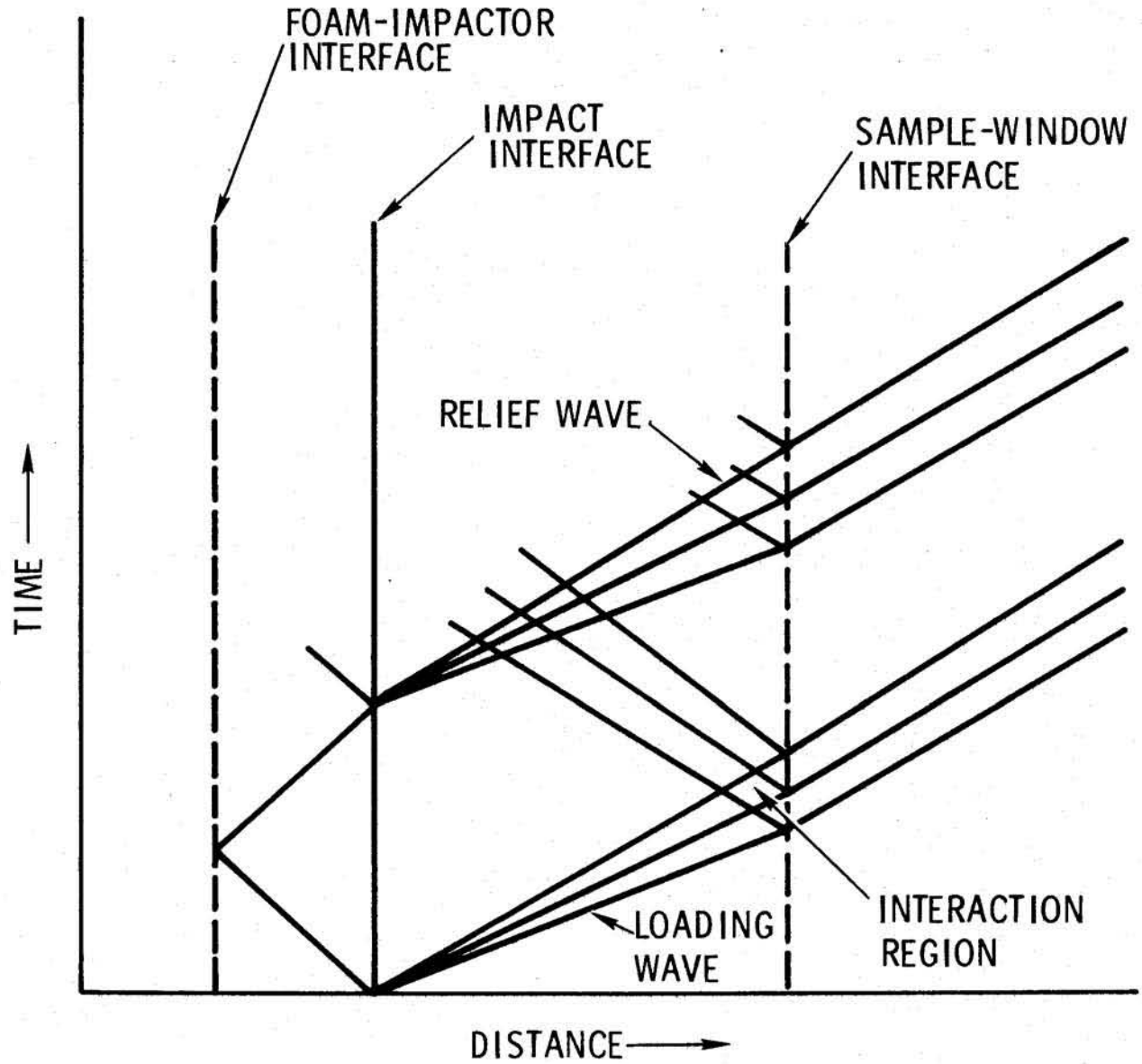

Fig. 2: DISTANCE-TIME PLOT

Illustrates the initial step input loading of the sample on impact and interaction regions due to impedance differences between sample and window material.

approach is based on invariance of the Riemann integrals in regions of wave interaction and, hence, precludes the analysis of experimental profiles which exhibit highly rate dependent material behavior. This is not a critical limitation, however, to a large class of geological and engineering materials where working constitutive relations are required. When significant rate dependence occurs, it is recognized as curvature in the distance-time characteristics of constant particle velocity amplitude.[8]

Wave propagation in the unknown sample material is governed by the equations of continuity and momentum. In Lagrangian coordinates they are:

$$\left(\frac{\partial \epsilon}{\partial t}\right)_h + \left(\frac{\partial u}{\partial h}\right)_t = 0 \tag{1}$$

$$\left(\frac{\partial u}{\partial t}\right)_h + \frac{1}{\rho_o}\left(\frac{\partial \sigma}{\partial h}\right)_t = 0 \tag{2}$$

where

$$\epsilon = 1 - \rho_o/\rho \ . \tag{3}$$

The time coordinate and Lagrangian space coordinate are t and h, respectively, and ϵ, u, and σ are the strain, particle velocity, and stress referred to the direction of wave propagation. ρ and ρ_o are the present and initial densities. Equations 1 and 2 can be rewritten as

$$\left(\frac{\partial \epsilon}{\partial t}\right)_h + \left(\frac{\partial u}{\partial h}\right)_t = 0 \tag{4}$$

$$\left(\frac{\partial u}{\partial t}\right)_h + c^2\left(\frac{\partial \epsilon}{\partial h}\right)_t = 0 \tag{5}$$

where

$$c^2 = \frac{1}{\rho_o}\left(\frac{\partial \sigma}{\partial \epsilon}\right)_h \ . \tag{6}$$

By adding and subtracting, we can put Eqs. 4 and 5 into the following form.

$$\frac{\partial u}{\partial \alpha} + c\,\frac{\partial \epsilon}{\partial \alpha} = 0 \tag{7}$$

$$\frac{\partial u}{\partial \beta} - c\,\frac{\partial \epsilon}{\partial \beta} = 0 \tag{8}$$

$$\frac{\partial x}{\partial \alpha} - c\,\frac{\partial t}{\partial \alpha} = 0 \tag{9}$$

$$\frac{\partial x}{\partial \beta} + c\,\frac{\partial t}{\partial \beta} = 0 \tag{10}$$

Equations 9 and 10 define families of right- and left-facing characteristics, C_+ and C_- and α and β are new independent variables referred to these characteristic coordinates. Equations 7 and 8 can be integrated to obtain the Riemann invariants

$$J_+(\beta) = u + f(\sigma) \tag{11}$$

and

$$J_-(\alpha) = u - f(\sigma) \tag{12}$$

where

$$f(\sigma) = \int^{\sigma} \frac{d\sigma}{\rho_o c} \tag{13}$$

is a function, to be determined, which described the stress-strain response of the unknown sample material. Continuity of stress and particle velocity require that, at the sample-window interface,

$$\frac{1}{2}\,(J_+ + J_-) = U(t) \tag{14}$$

and

$$\frac{1}{2}\,(J_+ - J_-) = f(P(t)) \tag{15}$$

where U(t) is the measured particle velocity profile and P(t) is the stress profile determined from the known properties of the window material (for instance, a nonlinear elastic equation of state for fused silica). Solving Eqs. 14 and 15 for the Riemann invariants, results in

$$J_+ = U(t) + f(P(t)) \ , \tag{16}$$

and

$$J_- = U(t) - f(P(t)) \ . \tag{17}$$

The x-t diagram in Fig. 2 shows that the propagating stress wave is originally a simple wave (all wave information is carried on C_+ characteristics) which enters an interaction region caused by reflection at the sample-window interface. Due to invariance, the same value of J_+ in the simple wave region is carried into the interaction wave region. Therefore, if <u>no interface were present the Riemann invariants would be</u>

$$J_+ = U(t) + f(P(t)) \tag{18}$$

$$J_- = 0 \ . \tag{19}$$

Equations 18 and 19, with the definition of the Riemann invariants in Eq. 11 and 12 provide the in-material particle velocity profile if no mismatched interface were present,

$$u(t) = \frac{1}{2}\Big[U(t) + f(P(t))\Big] \ . \tag{20}$$

Neglected in this analysis is the slight refraction of J_+ characteristics when they enter the interaction region which causes slight arrival time errors upon reaching the sample-window interface. These time errors will be small if propagation time in the region of interaction is maintained small compared to propagation time in the noninteraction region, and if

the impedance mismatch is kept small so that the bending of characteristics is minimized. Also, since the refraction of characteristics will be in the same direction for experiments using different sample thicknesses, the errors in time of arrival used to calculate wave velocities will tend to cancel.

A geometric interpretation of equation (20) is helpful in visualizing the impedance mismatch correction. In Fig. 3 we show a representative loading response curve for the window material and the sample material in the pressure-particle velocity plane. Also shown is the mirror image of the sample material loading curve about a point u(t) on the particle velocity axis. Referring to Eq. (20) the quantities U(t), P(t), f(P(t)), and u(t) are shown in the figure. A similar geometric representation can be drawn for the unloading wave.

It is necessary to clarify one further point concerning Fig. 3. We see that it conforms with a familiar interpretation of a wave of amplitude u(t) "unloading" to an amplitude U(t) upon encountering the lower impedance window material. The concept of unloading, however, is misleading in the case of a continuous loading wave arriving at the sample-window interface. In the case of a hysteretic material (loading and unloading response paths are different), it is important to note that it is the <u>loading and not unloading</u> curve that is used as the reflected curve in Fig. 3. This geometric procedure is then consistent, as it must be, with Eq. (20).

Eq. (20) provides the analytic expression for correcting the experimental profiles for impedance mismatch of the window material. The function f(P(t)) in Eq. (20) or, equivalently, the response curve for the sample material in Fig. 2 is not known before the fact. Since

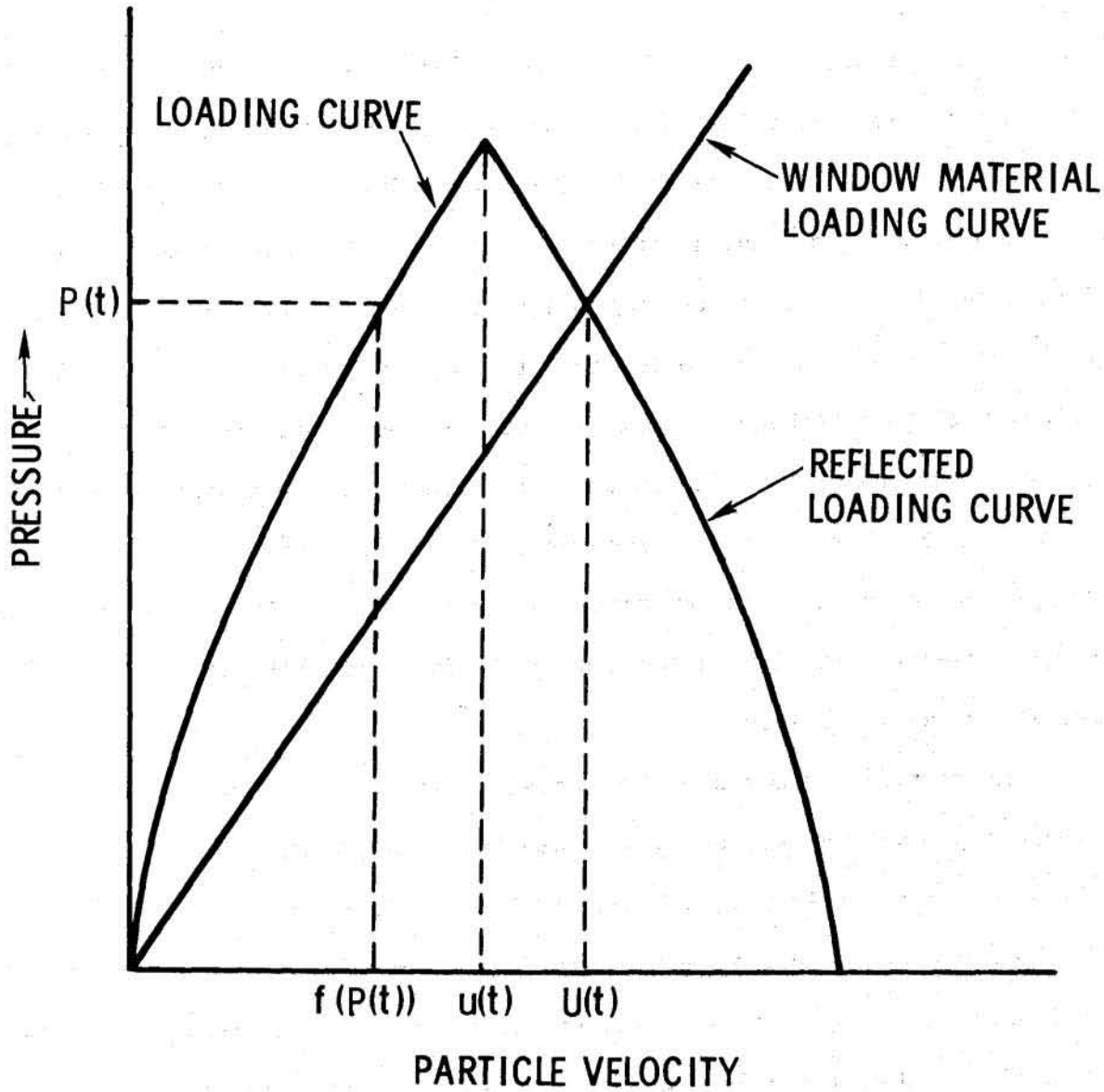

Fig. 3: GEOMETRIC REPRESENTATION OF THE SAMPLE-WINDOW IMPEDANCE MISMATCH CORRECTION.

$$f(\sigma) = \int \frac{d\sigma}{\rho_o c} \tag{21}$$

we see that knowledge of the functional dependence of the wave velocity within the integral in Eq. 21 is equivalent to knowing $f(\sigma)$. The procedure followed is to use the multiple uncorrected particle velocity profiles obtained at different sample thicknesses to provide an estimate of the functional dependence of the Lagrangian wave velocity on particle velocity. Eq. (20) is then used to provide an estimate for the corrected particle velocity profiles (those occurring if no interface were present) which in turn provides a better estimate of the functional dependence of the wave velocity. By this iterative procedure, the value of $f(\sigma)$ or equivalently the σ-u curve for the unknown sample material is determined. Having this, the stress-strain paths and longitudinal modulus for material loading and unloading readily follow.

In practice, it has been found that no iterating was necessary to obtain satisfactory results in the materials studied to this date. Further iteration provides improvement only if the mesh size is sufficiently fine. To check the procedure, a test problem for a sample material with a variable impedance differing from fused silica window material by about 25 percent on loading and by about 100 percent on unloading was developed. A total of 28 mesh points were used to describe the total loading and unloading profile. The analysis scheme reproduced the correct in-material particle velocity profiles within about 0.5 percent on loading and about 2.5 percent on unloading after one iteration.

IV. ATTENUATING WAVE

When a one-dimensional loading stress wave propagates into a region of uniform state, as in the situation of a typical plate impact experiment, the wave velocity at any amplitude on the loading wave relates directly to the loading stress-strain modulus through the relation*

$$\frac{d\sigma}{d\epsilon} = \rho_o c^2(\sigma) \ . \tag{22}$$

Similarly, if an overtaking relief wave propagates into the uniform state remaining after passage of the loading wave then the relief wave velocity at any amplitude is related to the unloading stress-strain modulus through the same relation (Eq. 22). If, however, the unloading wave overtakes the loading wave and attenuation of the wave peak occurs then wave velocities determined from the unloading wave do not provide the unloading stress-strain modulus as in Eq. 22. Care must be exercised in extracting unloading stress-strain data from wave velocities determined from attenuating wave propagation data.

We have found attenuating wave data difficult to circumvent in the study of some rock types. A good example is Solenhofen limestone[7] (See Fig. 9) which has an extremely slow loading wave velocity above the axial failure stress allowing rapid overtaking of the unloading wave and subsequent attenuation of the peak.

The approach we have taken to extract the unloading stress-strain response from attenuating data has been to examine the equations governing attenuating wave propagation and determine, quantitatively, how significantly the measured unloading wave velocities differ from Eq. 22. We have found a

---

*This observation is strictly true only if rate dependent effects are not present. The experimental situations in which rate dependence will adversely effect the results are limited, however, and will not be considered here.

relation for this difference which depends on the rate of attenuation of the peak and the slope of the wave in front of and in back of the peak. We have found from this analysis that in a majority of experimental situations the unloading wave velocity does not differ significantly from Eq. 22, and therefore, analysis methods valid for nonattenutating wave propagation can be used with little error on attenuating data. The results of that calculation follow.

In Fig. 4 we depict a set of particle velocity histories which might be obtained in a typical experiment where wave attentuation has occurred. Let the Lagrangian distance and time coordinates be h and t, respectively, and u the particle velocity. The velocity of the wave peak will be denoted by D and quantities immediately in front of and immediately in back of the peak will be denoted by a subscript 1 or a subscript 2, respectively.

We ultimately wish to relate the unloading wave velocity at a given amplitude to the unloading stress-strain modulus

$$\frac{d\sigma}{d\varepsilon} = \frac{\sigma_t}{\epsilon_t} = \rho_o c^2 \ . \tag{23}$$

From the equation of continuity

$$\epsilon_t = - u_h = \frac{1}{D} (u_t - \dot{u}) \ . \tag{24}$$

The respective partial derivatives are expressed by subscript t or h and the dot refers to a directional time derivative along the path of the attenuating peak. The equation of motion gives

$$\sigma_h = - \rho_o u_t \tag{25}$$

and since

$$\sigma_t = \dot{\sigma} - D\sigma_h \tag{26}$$

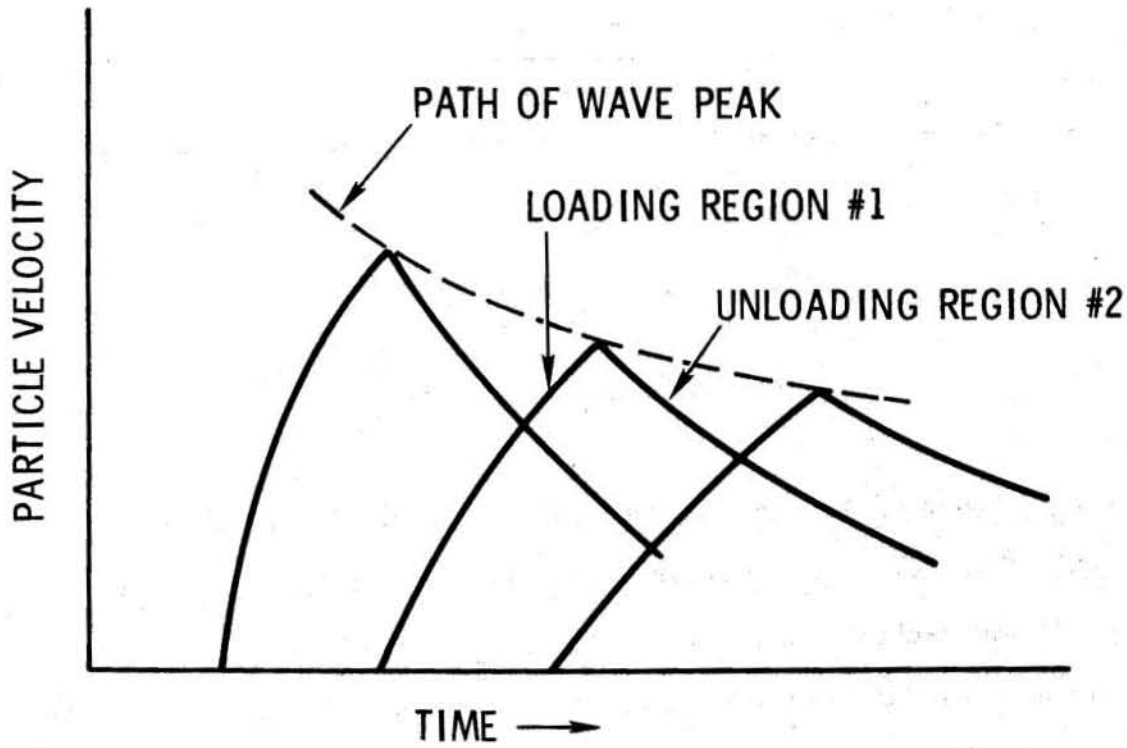

Fig. 4: ATTENUATING WAVE PROPAGATION.

we obtain

$$\sigma_t = \dot{\sigma} + \rho_o D u_t \tag{27}$$

Equations 23, 24 and 27 provide

$$\rho_o c^2 = \frac{D(\dot{\sigma} + \rho_o D u_t)}{(u_t - \dot{u})} \tag{28}$$

Since Eq. 28 applies either in front of or behind the peak, we obtain the relations

$$\rho_o c_1^2 (u_{t1} - \dot{u}) - \rho_o D^2 u_{t1} = D\dot{\sigma} \tag{29}$$

$$\rho_o c_2^2 (u_{t2} - \dot{u}) - \rho_o D^2 u_{t2} = D\dot{\sigma} \ , \tag{30}$$

where $\rho_o c_1^2$ and $\rho_o c_2^2$ are the slope of the stress-strain curve immediately in front of and immediately behind the peak. Similarly, u_{t1} and u_{t2} are slopes of the particle velocity profile on opposite sides of the peak.

To eliminate σ it is necessary to restrict the form of the loading wave. We will make a simple wave assumption

$$\dot{\sigma} = \rho_o c_1(u)\dot{u} \tag{31}$$

which includes as limiting cases shock loading, $\dot{\sigma} = \rho_o D\dot{u}$ or a steady wave, $\dot{\sigma} = \rho_o c\dot{u}$, where c = constant.

Substitute Eq. 31 into Eq. 29 and 30 and define $\alpha = \dot{u}/u_{t1}$ and $\beta = \dot{u}/u_{t2}$ which are ratios of the rate of profile decay in front of and behind the peak. The resulting equations are

$$c_1^2(1 - \alpha) - D^2 = Dc_1\alpha \tag{32}$$

$$c_2^2(1 - \beta) - D^2 = Dc_1\beta \ . \tag{33}$$

In region two we will define an unloading wave velocity

$$c_u = \left(\frac{\partial h}{\partial t}\right)_u \tag{34}$$

which is the experimentally measured wave velocity at particle velocity amplitude u. The purpose of this calculation is to determine how c_u differs from c_2 which in turn is identified with the unloading stress-strain modulus through Eq. 22. In region 2 (at the peak) we obtain the relation

$$c_u = D/(1 - \beta) \tag{35}$$

which is used to eliminate D from Eqs. 32 and 33. Let $x = c_2/c_u$ and eliminating c_1 from the resulting expression provides, after some algebra, the relation

$$x^2 = (1 - \beta)\left[1 + \frac{\beta}{1 - \alpha}\right] \tag{36}$$

when there is no attenuation $x = 1$ ($C_u = C_2$) and the measured wave velocity identically determines the stress-strain modulus. When attenuation occurs α and β can be determined from the rate of attenuation and the local slopes of the profiles in front of and behind the peak and Eq. 36 can be used to determine the error in assuming that the measured wave velocity determines the stress-strain modulus. In Fig. 5, Eq. 36 is displayed graphically. The curves shown correspond to 1 percent, 2 percent, and 3 percent errors, respectively. Also shown is the point corresponding to the Solenhofen limestone data presented in Fig. 9 which places the order of error expected in perspective.

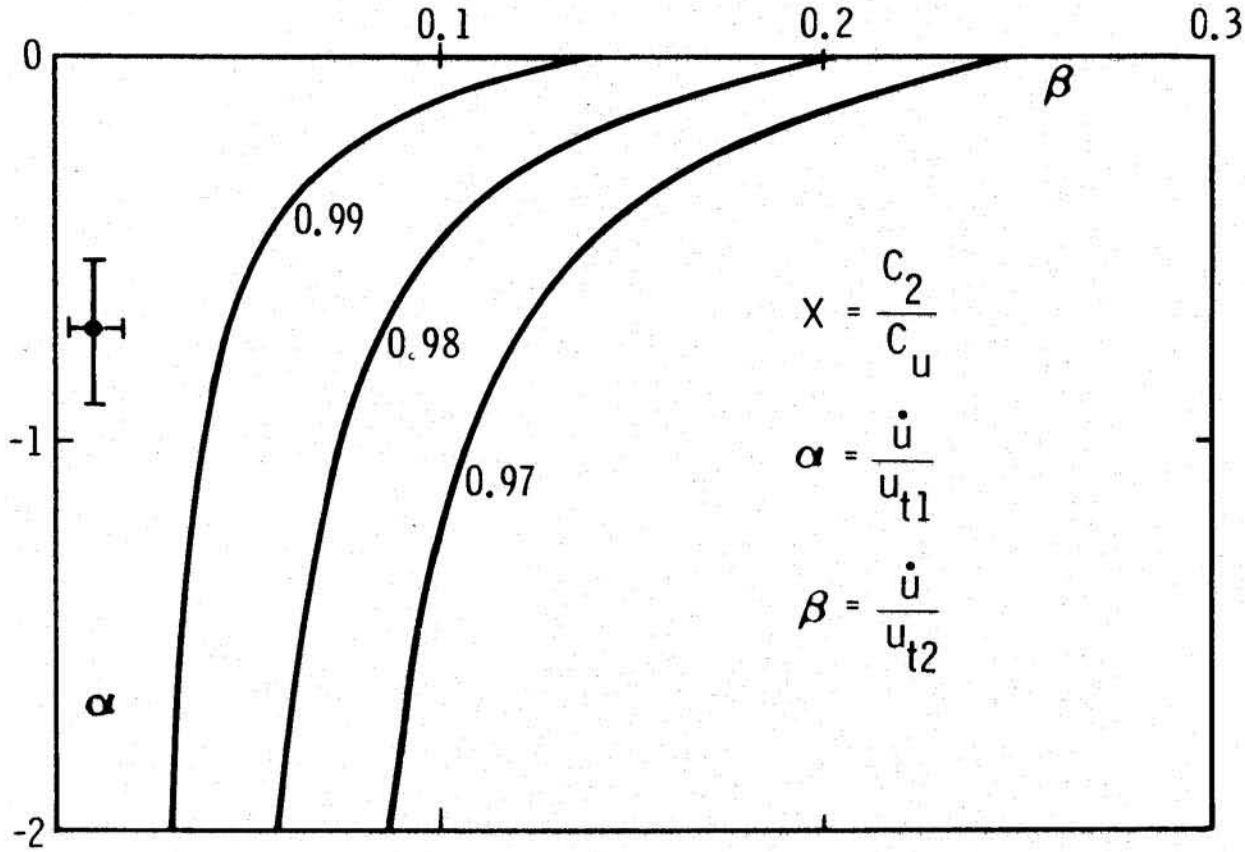

Fig. 5: PLOT OF EQUATION 36 AT CONSTANT VALUES OF X.

The ordinate and abcissa are the relative slopes of the loading and unloading waves to the slope of the peak attenuation. The point shown corresponds to the Solenhofen limestone data plotted in Fig. 9.

V. OPERATIONAL DESCRIPTION

For each individual experiment, program "VISAR"[12] will provide a particle velocity-time history from the raw interference fringe records. The profile shape is determined by the impact velocity, impactor thickness, constitutive properties of the test medium, and impedance differences between the test sample and the window material. The profiles from a series of experiments are used to determine the constitutive behavior of the test material. Programs "MODIFY" and "IMP" have been developed to assist in this analysis.

Program "MODIFY" Analysis begins by taking the velocity profiles from the same series (similar impact velocities and impactor thicknesses but different sample thicknesses) and plotting them on the same particle velocity-time graph. This can be done using program "PLOTPL" which is available on the PDP10, and provides a qualitative picture of the propagation properties of the wave profiles. Since profiles were obtained from individual experiments, there are obvious inconsistencies due to slight differences in impact velocities and impactor thicknesses. Adjustment of the amplitudes and unloading wave times to account for these slight differences is required in order to make the profiles from a given series compatible. Also, individual profiles may have irregularities due to superpose time marks, triggering noise or other sources which would be desirable to eliminate. Program "MODIFY" was constructed to account for these problems.

When "MODIFY" is executed (EX@MOD) the user is asked the input file name and then the desired output or modified file name. If the user is sufficiently confident of the adjustments required, the new file may be read into the old file name. This will irreversibly kill the old file,

however. The input profile is then automatically plotted and provides the user with a working plot of the velocity-time history.

The user is then asked to distinguish the point in time separating the loading wave from the unloading wave ("DO YOU WISH TO MARK THE PEAK?"). This has two purposes. First, if adjustment of the unloading wave due to different impactor thicknesses is desired this separation time is necessary. Second, this time is required in program IMP. When the user responds with yes ("Y") he or she is given the option to replot. This can be done on a magnified scale so that the time of the peak can be more closely identified. The user is then requested to identify the time ("ENTER TIME OF PEAK").

The next operation allows adjustment of the data in time and amplitude. The user is asked "DO YOU WISH TO SCALE OR TRANSLATE THE DATA." An affirmative response is immediately followed by "DO YOU WISH TO TRANSLATE IN TIME?" If yes, then the user is asked whether they wish to shift the entire profile or the unloading wave only. The former provides a rigid translation of the profile while the latter shifts only the unloading wave. The user then has an opportunity to scale the profile in time by some constant scale factor. This operation is used primarily to compare profiles for self similar flow where the profile is some function of the distance divided by the time. Finally, the user is asked "DO YOU WISH TO SCALE IN VELOCITY?" This operation allows the user to adjust the profile amplitude for differences in impact velocity.

In the next operation the user is given a chance to alter the profile ("ALTER ANY PORTION OF THE PROFILE?"). If yes, the user is allowed to replot and focus on any portion of the profile. The user may then select any time interval of the data file to be eliminated and then specify the

points to replace the eliminated data. The process is repeated until the user has no further alterations to perform.

Lastly, the user is given an opportunity to plot both the old and the new data file on the same velocity-time plot. The old data appears as a solid line while the new data is plotted as a dashed line. This provides a check on the adjustments and alterations made on the data file. This comparison cannot be made if the modified file has been given the same as the input file.

Program "IMP" After a set of particle velocity profiles from any given series have been satisfactorily correlated using "MODIFY," program "IMP" is executed (EX@IMP). This program is intended to evaluate the constitutive properties of the test material (stress-strain, stress-particle velocity, and wave velocity-stress behavior) and account for the impedance difference between the sample material and the laser window material as was discussed in Section III. At execution, the user is asked the number of input files and their names. Two to five files can presently be analyzed with the program. Following this, the size of the velocity increment must be input. We have found that about 20 to 30 increments from wave foot to peak convenient and sufficient and the increment is selected accordingly. The program then asks whether the user desires a plot of the incremented velocity profiles. An example of such a plot is provided in Fig. 6. If this is the first program run such a plot is necessary as a working plot for subsequent user interaction operations. The user is then asked to supply sample thicknesses in millimeters corresponding to the input profiles in order (increasing sample thickness). Finally, the user is given the opportunity to adjust timing of the profiles such that initial breakaway of the wave (foot of wave) corresponds to a prescribed velocity. This is convenient if a break-

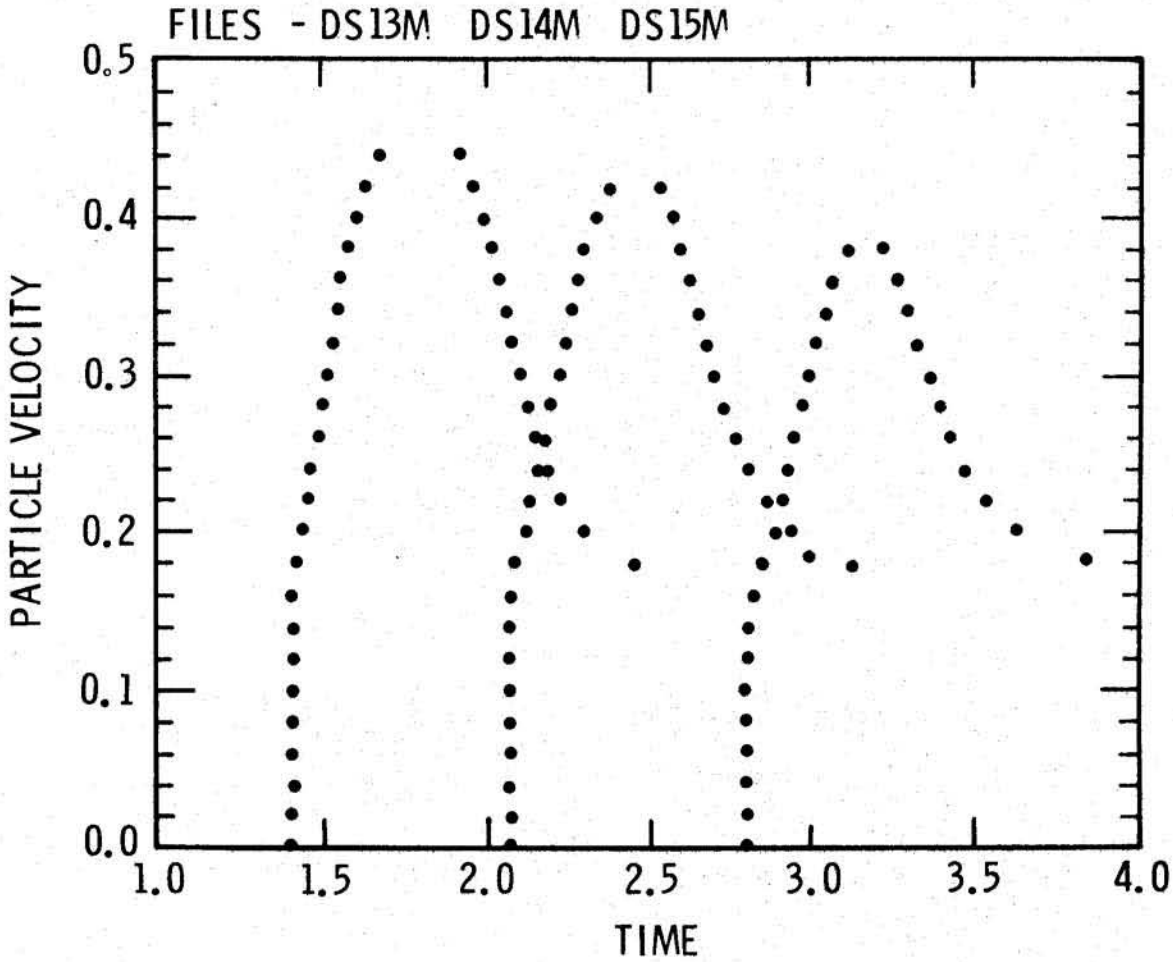

Fig. 6: INCREMENTED PARTICLE VELOCITY PROFILES.

Selected increment size is 0.02 km/s. Data was obtained on Blair dolomite.[6]

away velocity has been determined from a number of experiments and it is desired to have all profiles consistent with this velocity. The program then proceeds with the incremental velocity plot shown in Fig. 6.

The next program section provides distance-time plots for data evaluation and possible adjustment. The user is asked which characteristics he wishes to plot. Response is governed by a systematic numbering of the particle velocity levels. Referring to Fig. 6 the zero velocity foot of the wave is numbered one. Each consecutive velocity level through the full profile (loading and unloading) is numbered with consecutive integers. A response of '5', for example, would plot the first, sixth, eleventh, and so forth characteristics. An irregularity occurs if the wave attenuates with propagation distance as occurs in the example in Fig. 6. In such a case the level count up the loading wave proceeds as long as any two profiles can contribute to that characteristic. Failing this, the count switches to the unloading wave and continues.

The user is then asked to provide the minimum and maximum time desired in the plot. This can be estimated from the velocity plot in Fig. 6. The distance axis is determined automatically from the sample dimensions entered. A distance-time plot corresponding to the velocity data in Fig. 6 is shown in Fig. 7. The small circles indicate the exact distance-time point for a given profile and particle velocity level. The line is a best linear fit to equivalent particle velocity level points on different contributing profiles. The user is then provided an opportunity to add or delete characteristics, adjust timing in terms of a rigid translation in time of any profile, and replotting. This process is iterated until the user is satisfied with the distance-time properties of the series.

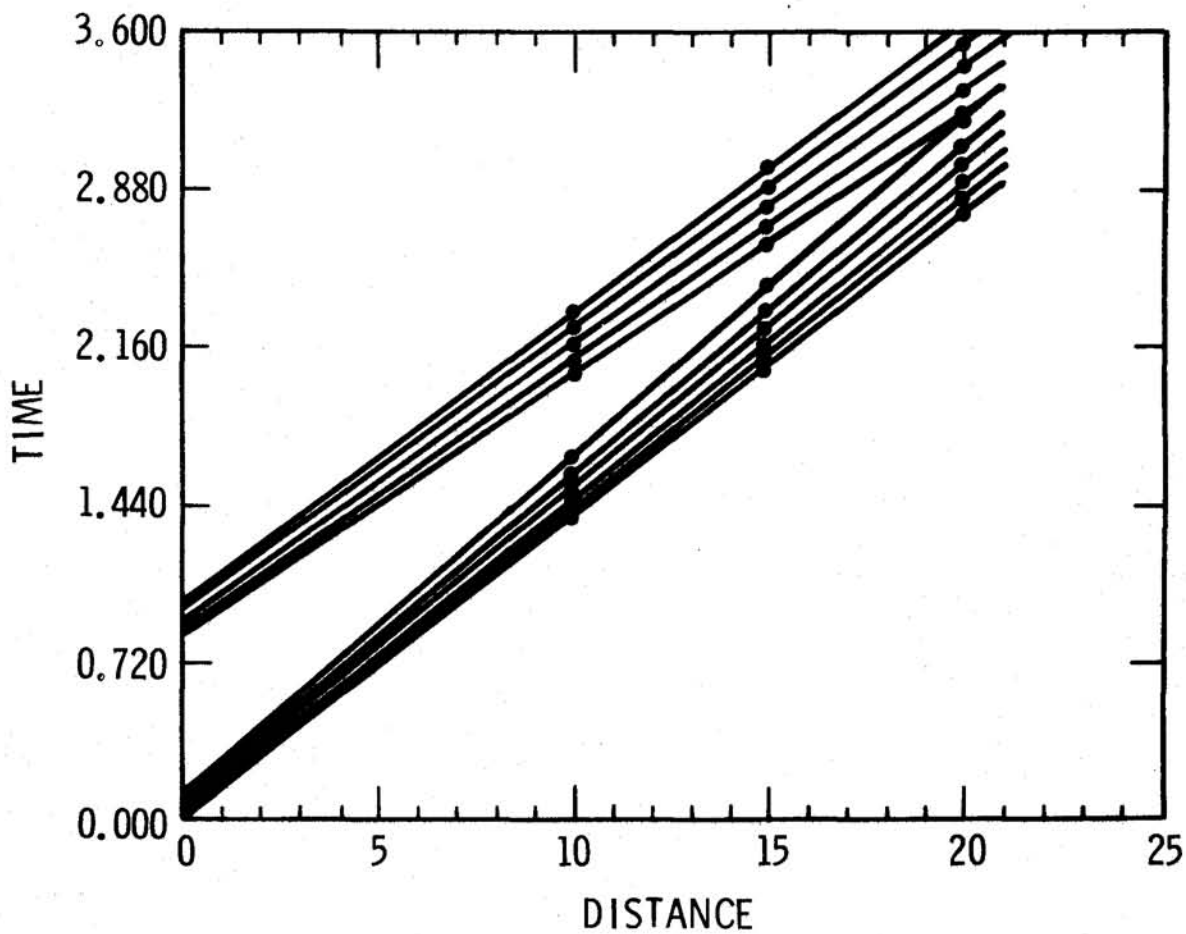

Fig. 7: DISTANCE-TIME PLOT CORRESPONDING TO INCREMENTED PROFILES DISPLAYED IN FIG. 6.

The user may then plot the wave velocity data determined from the linear fit to the incremented profile data. This plot is shown in Fig. 8. Note that there is a one-to-one correspondence between the points in the wave velocity-particle velocity plot in Fig. 8 and the particle velocity-time plot in Fig. 6. Adjustment of the wave velocity may be made at this point, a feature which is particularly valuable in assessing the effect of velocity variations on the ultimate constitutive properties of the material. Adjustment may be made in two ways: (1) The user may adjust the velocity corresponding to particular particle velocity level points on loading or unloading. The numbering scheme is the same as that incorporated for the characteristics in the distance-time plot. Or, (2) the user may adjust the total velocity spectrum in order to bring a given level point into correspondence with a desired wave velocity. This acts as a rigid translation of the wave profiles to bring about the desired correspondence.

At this point the user must enter the initial sample density (g/cc) and then the program uses the final wave velocity-particle velocity data to determine, and account for, the impedance difference between the sample material and the window material according to the methods discussed in Section III and to provide the final constitutive properties for the sample material. In Fig. 9 measured particle velocity profiles for Solenhofen limestone are shown and compared with profiles corrected with program IMP for the limestone-fused silica interface mismatch. The user may, at this point, print out in tabular form the incremental stress, strain, particle velocity and wave velocity data for both loading and unloading. The user may also choose to write a data file for the stress-strain, stress-particle velocity,

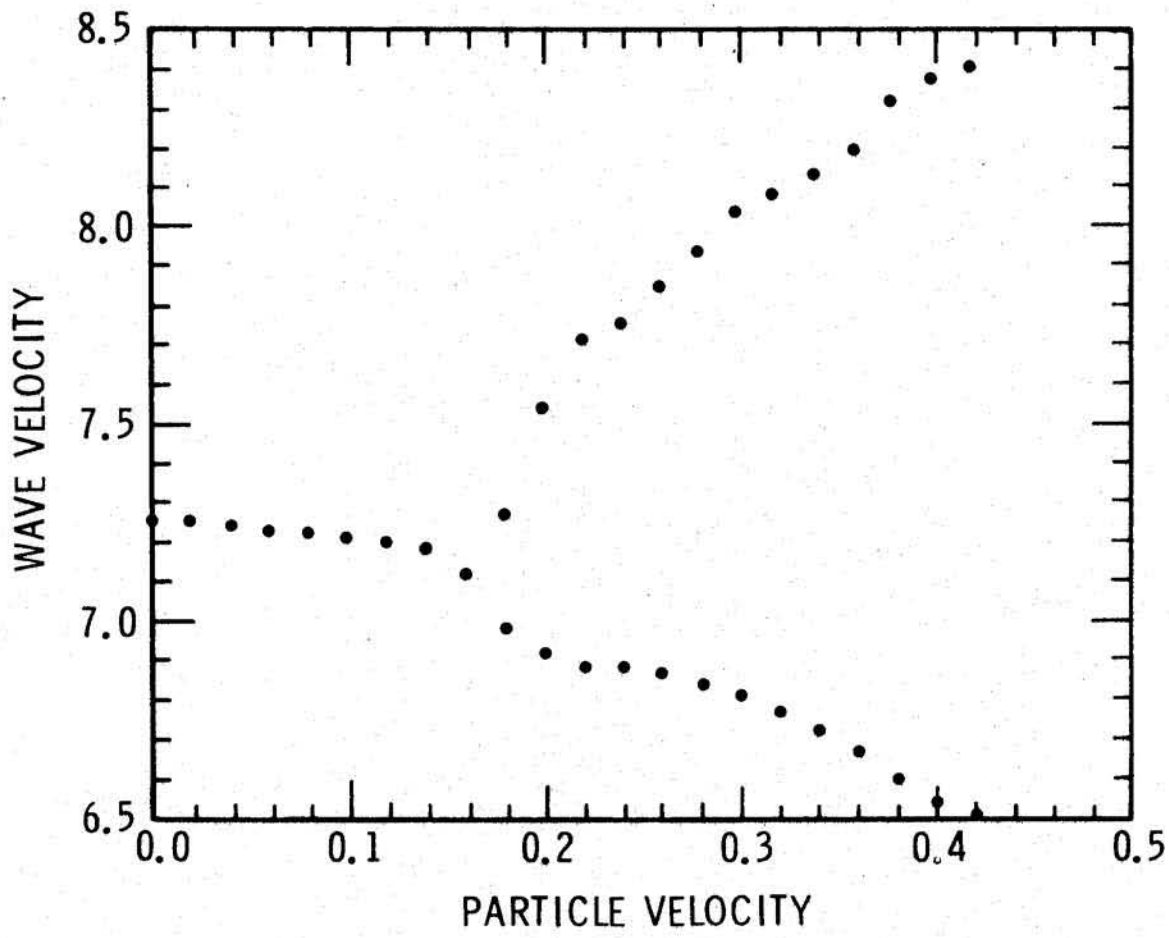

Fig. 8: WAVE VELOCITY VALUES CORRESPONDING TO THE INCREMENTED PROFILES IN FIG. 6.

Lower values are for the loading wave; upper values for the unloading wave. Note the correspondence between points in Fig. 6 and Fig. 8.

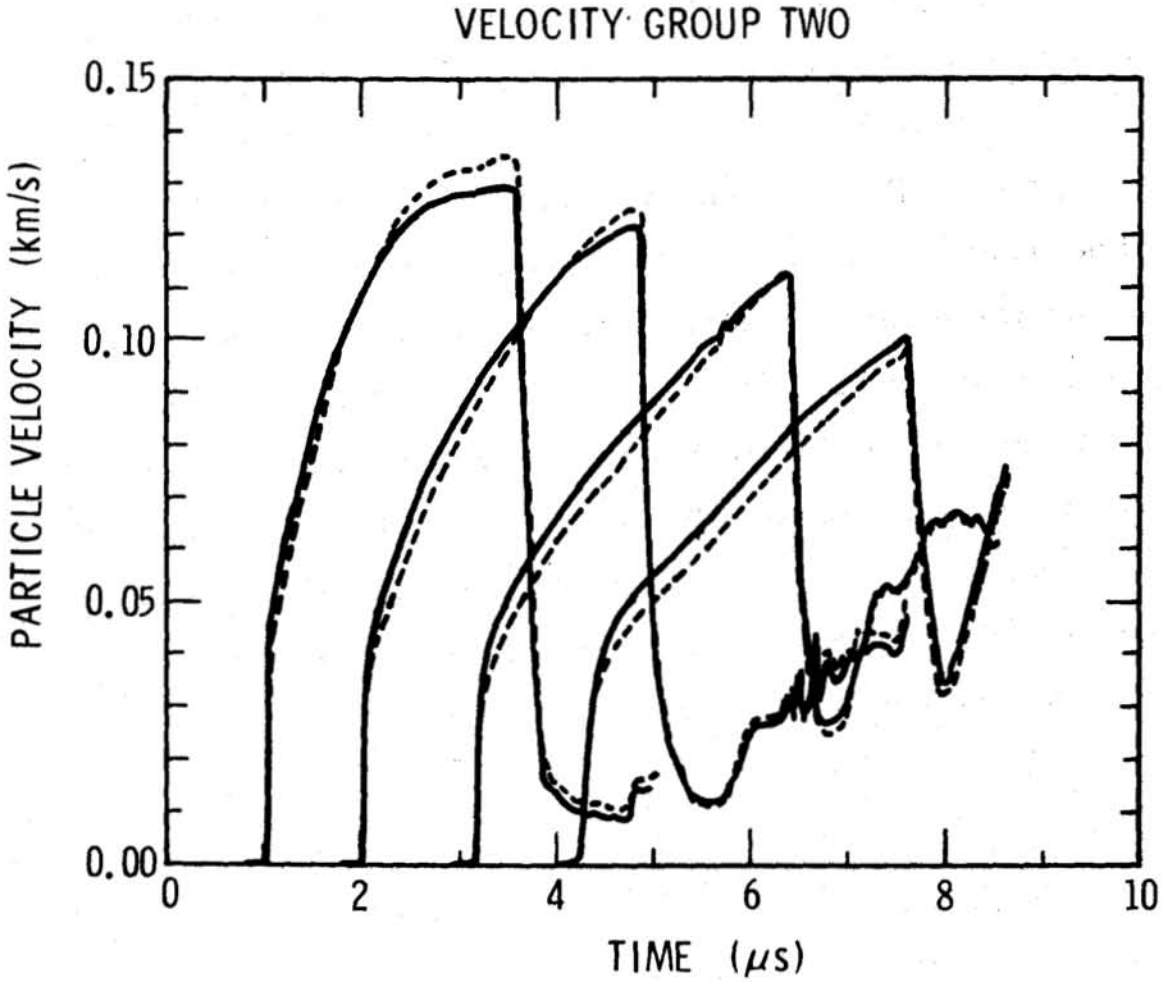

Fig. 9: PARTICLE VELOCITY PROFILES IN SOLENHOFEN LIMESTONE.[7]

Solid lines are the uncorrected profiles. Dashed lines represent the in-material profiles corrected for the limestone-fused silica interface impedance mismatch.

or wave velocity-stress data. This is convenient if, for example, a combined plot of the stress-strain data from several series is desired. As mentioned earlier, program "PLOTPL" may be used for this purpose. Finally, the user is given an opportunity to make immediate plots of the constitutive data.

Program "IMP" has been found convenient for rapid evaluation of the constitutive properties of a large number of experiments. It has also proved useful where questions of how errors in timing and amplitude in the original profiles effect the final constitutive properties since such variations may be introduced in the initial profiles or, at some other point and the resulting variation observed in the final output. The program was developed with velocity interferometer data from a sample-window interface configuration. It could be extended for use in other experimental configurations without undue difficulty.

APPENDIX I

Listing of Program MODIFY

```
      PROGRAM MODIFY
      DIMENSION  X(500), Y(500),LABX(25),LABY(25),LTITLE(25)
     1,UCOR(100),MPROF(2),XC(20),YC(20)
C     INITIALIZATION OF TEKTRONICS SOFTWARE SWITCHES
      CALL SET4TK
      CALL INITT(30)
      CALL TERM(1,1024)
1002  CALL BINITT
      CALL ANMODE
C
C     PLOT FILE --CARTESIAN COORDINATES ONLY
C
      TYPE 14
14    FORMAT(' INPUT FILENAME  '$)
      ACCEPT 100, IFIL
100   FORMAT( A5)
      TYPE 25, IFIL
25    FORMAT(' OUTPUT FILE FOR MODIFIED OR CORRECTED  ',A5,'  '$)
      ACCEPT 100, MOUT
      CALL NEWPAG
C     TEK SUBPROGRAMS FOR PLOTTING EITHER REQUIRE MUCH STORAGE OR AN
C     EXTRA READ OF THE FILES TO PREDETERMINE PLOTTING PARAMETERS.
C     THE LATTER WAS CHOSEN IN THIS PROGRAM.
C
C
C     PRESET GLOBAL MAXIMA AND MINIMA FOR ALL X AND Y ARRAYS
C     FROM ALL  FILES.
      CALL PRESET (XMIN,XMAX,YMIN,YMAX)
      CALL READ(IFIL,X,Y)
      CALL MINMAX(X,XMIN,XMAX)
      CALL MINMAX(Y,YMIN,YMAX)
C
C     PLOT UNALTERED DATA
18    CALL PLSCAL(X,Y,IFIL,1,XMIN,XMAX,YMIN,YMAX,'Y',1)
      CALL TINPUT(M)
      CALL NEWPAG
      CALL ANMODE
C
1008  TYPE 150
150   FORMAT(' DO YOU WISH TO MARK THE PROFILE PEAK? '$)
C     DETERMINE PROFILE PEAK
      ACCEPT 21, LYES
      MARPEK=0
      IF(LYES.EQ.'N') GO TO 199
      MARPEK=1
151   TYPE 152
152   FORMAT(' REPLOT? '$)
      ACCEPT 21, LYES
      IF(LYES.EQ.'N') GO TO 156
      TYPE 153
153   FORMAT(' ENTER UMIN,UMAX,TMIN,TMAX ')
      ACCEPT 154, YMIN,YMAX,XMIN,XMAX
154   FORMAT(4G)
      CALL BINITT
      CALL NEWPAG
155   CALL PLSCAL(X,Y,IFIL,1,XMIN,XMAX,YMIN,YMAX,'Y',1)
      CALL TINPUT(M)
      CALL NEWPAG
      CALL ANMODE
156   TYPE 157
157   FORMAT(' ENTER TIME OF PEAK '$)
      ACCEPT 158,TPEAK
```

```
158     FORMAT(G)
        CALL READ(IFIL,X,Y)
        LREAD='Y'
        N=X(1)
        DO 160 I=2,N
        DIF=X(I)-TPEAK
        IF(DIF.GE.0.) GO TO 162
160     CONTINUE
162     MUNLOD=N-I+1
        NNL=I
        DO 164 J=1,MUNLOD
        Y(N+2-J)=Y(N+1-J)
        X(N+2-J)=X(N+1-J)
164     CONTINUE
        N=N+1
        X(1)=N
        Y(1)=N
C
C
199     TYPE 40
40      FORMAT(' DO YOU WISH TO SCALE OR TRANSLATE THE PROFILE? '$)
        ACCEPT 21,NYES
        IF(NYES.EQ.'N') GO TO 169
        N=X(1)
        TYPE 26
26      FORMAT(' DO YOU WISH TO TRANSLATE IN TIME? '$)
        ACCEPT 21,NYES
        IF(NYES.EQ.'N') GO TO 220
        TYPE 202
202     FORMAT(' ENTIRE PROFILE OR UNLOADING ONLY? '$)
        TYPE 206
206     FORMAT('  (ANSWER "ENT" OR "UNL") '$)
        ACCEPT 203,PSHIFT
203     FORMAT(A3)
        TYPE 204
204     FORMAT(' ENTER TIME SHIFT '$)
        ACCEPT 28,XTS
28      FORMAT(G)
        IF(PSHIFT.EQ.'UNL') GO TO 207
        DO 205 I=2,N
205     X(I) = X(I)+XTS
        GO TO 220
207     IF(MARPEK.EQ.1) GO TO 240
        TYPE 241
241     FORMAT(' YOU FORGOT TO MARK THE PEAK! ')
        GO TO 1008
240     NNL1=NNL+1
        DO 208 I=NNL1,N
208     X(I)=X(I)+XTS
        DO 209 I=2,NNL
        IF(X(I).GE.X(NNL1)) GO TO 210
209     CONTINUE
        X(NNL)=X(NNL1)
        GO TO 220
210     IL=I
        DO 215 I=IL,NNL
        DO 214 J=NNL1,N
        IF(X(J).LT.X(I)) GO TO 214
        IF(Y(J).LT.Y(I)) GO TO 216
        GO TO 215
214     CONTINUE
```

```
215     CONTINUE
216     X(I+1)=X(I)
        Y(I+1)=Y(I)
        I2=I+2
        N=N+I-J+2
        DO 217 I=I2,N
        X(I)=X(J+(I-I2))
        Y(I)=Y(J+(I-I2))
217     CONTINUE
        X(1)=N
        Y(1)=N
220     TYPE 221
221     FORMAT(' DO YOU WISH TO SCALE IN TIME? '$)
        ACCEPT 21,NYES
        IF(NYES.EQ.'N') GO TO 224
        TYPE 222
222     FORMAT(' ENTER TIME SCALE FACTOR '$)
        ACCEPT 25,XF
        DO 223 I=2,N
223     X(I)=X(I)*XF
224     TYPE 225
225     FORMAT(' DO YOU WISH TO SCALE IN VELOCITY? '$)
        ACCEPT 21,NYES
        IF(NYES.EQ.'N') GO TO 169
        TYPE 226
226     FORMAT(' ENTER VELOCITY SCALE FACTOR '$)
        ACCEPT 28,YF
        DO 227 I=2,N
        Y(I)=Y(I)*YF
227     CONTINUE
C
169     TYPE 175
175     FORMAT(' ALTER ANY PORTION OF PROFILE? '$)
C       ALTER PORTIONS OF PROFILE
        ACCEPT 21,LYES
        IF(LYES.EQ.'N') GO TO 42
        L=X(1)
176     TYPE 171
        ACCEPT 21,NYES
        IF(NYES.EQ.'N') GO TO 177
        CALL ALTPLT(MOUT,X,Y)
177     TYPE 178
178     FORMAT(' ENTER TMIN AND TMAX FOR DOMAIN OF ALTERATION '$)
        ACCEPT 179,TMIN,TMAX
179     FORMAT(2G)
        TYPE 180
180     FORMAT(' ENTER NUMBER OF POINTS TO INSERT '$)
        ACCEPT 182,NCORP
182     FORMAT(I)
        TYPE 183
183     FORMAT(' ENTER CORRECTED U-T POINTS IN ORDER '/)
        DO 185 I=1,NCORP
        TYPE 184
184     FORMAT(' ENTER U-T PAIR '$)
        ACCEPT 179,YC(I),XC(I)
185     CONTINUE
        DO 186 I=2,N
        DIF=X(I)-TMIN
        IF(DIF.GE.0.) GO TO 187
186     CONTINUE
187     IMIN=I-1
```

```
      DO 188 I=IMIN,N
      DIF=X(I)-TMAX
      IF(DIF.GE.0.) GO TO 189
188   CONTINUE
189   IMAX=I
      DO 190 I=1,NCORP
      X(IMIN+I)=XC(I)
      Y(IMIN+I)=YC(I)
190   CONTINUE
      LEND=N-IMAX
      DO 191 I=1,LEND
      X(IMIN+NCORP+I)=X(IMAX+I)
191   Y(IMIN+NCORP+I)=Y(IMAX+I)
      N=N+IMIN-IMAX+NCORP
      X(1)=N
      Y(1)=N
      TYPE 171
171   FORMAT(' REPLOT? '$)
      ACCEPT 21, LYES
      IF(LYES.EQ.'N') GO TO 174
      CALL ALTPLT(MOUT,X,Y)
174   TYPE 192
192   FORMAT(' FURTHER ALTERATIONS? '$)
      ACCEPT 21, LYES
      IF(LYES.EQ.'Y') GO TO 176
C     WRITE MODIFIED FILE
42    CALL WRITE(MOUT,X,Y)
24    CONTINUE
C
37    TYPE 39
39    FORMAT(' DO YOU WISH TO PLOT INPUT AND OUTPUT FILES? '$)
      ACCEPT 21,LYES
      IF(LYES.EQ.'N') GO TO 17
      TYPE 35
35    FORMAT('  DO YOU WISH TITLES AND LABELS?')
      ACCEPT 21,MT
21    FORMAT(A1)
      IF(MT.EQ.1HN) GO TO 1000
      CALL MYLABL(LTITLE,LABX,LABY)
      TYPE 33
33    FORMAT('  PLOT WITH GRID?')
      ACCEPT 21,MO
1000  CALL BINITT
      TYPE 15
15    FORMAT('  ENTER UMIN, UMAX, TMIN, TMAX  '$)
      ACCEPT 16, YMIN,YMAX,XMIN,XMAX
16    FORMAT(4G)
C
C     PLOT INPUT AND MODIFIED FILES.
36    CALL NEWPAG
      DO 23 K = 1,2
      MM=MOD(K,2)
      MPROF(K)=IFIL
      IF(MM.EQ.0) MPROF(K)=MOUT
22    CALL PLSCAL(X,Y,MPROF,K,XMIN,XMAX,YMIN,YMAX,MO,MM)
23    CONTINUE
      IF(MT.EQ.1HN) GO TO 38
C     OUTPUT TITLE AND LABELS
      CALL MOVABS(300,780)
      CALL HLABEL(25,LTITLE)
      CALL MOVABS(5,660)
```

```
      CALL VLABEL(25,LABY)
      CALL MOVABS(300,0)
      CALL HLABEL(25, LABX)
38    CALL TINPUT(M)
      CALL NEWPAG
      CALL ANMODE
C     REPLOT AGAIN?
      TYPE 34
34    FORMAT('  REPLOT FILES? '$)
      ACCEPT 21, MG
      IF (MG.EQ.'N')  GO TO 17
      GO TO 1000
17    TYPE 30
30    FORMAT('  MODIFY OTHER FILES?  '$)
      ACCEPT 21, MG
      IF (MG .EQ. 'Y')   GO TO 1002
1001  CONTINUE
      CALL NEWPAG
300   CALL FINITT(0,0)
      CALL CLR4TK
      END
C
C
C     PROFILE PLOT DURING INTERMEDIATE ALTER STEP.
C
      SUBROUTINE ALTPLT(MOUT,X,Y)
      DIMENSION X(1),Y(1)
      CALL WRITE(MOUT,X,Y)
      TYPE 10
10    FORMAT(' ENTER UMIN,UMAX,TMIN,TMAX ')
      ACCEPT 20, YMIN,YMAX,XMIN,XMAX
20    FORMAT(4G)
      CALL BINITT
      CALL NEWPAG
      CALL PLSCAL(X,Y,MOUT,1,XMIN,XMAX,YMIN,YMAX,'Y',1)
      CALL TINPUT(M)
      CALL NEWPAG
      CALL ANMODE
      RETURN
      END
C
C
C     READ ONE FILE FROM DISK
      SUBROUTINE READ(IFI,X,Y)
      DIMENSION X(1), Y(1)
      OPEN(UNIT =20, FILE=IFI, ACCESS='SEQIN')
      NO = 0
      DO 10 I = 2,500
      READ (20, 106,END = 20) X(I),Y(I)
106   FORMAT(2G)
10    NO = NO +1
      TYPE 22, IFI
22    FORMAT(' NOT ALL DATA READ FROM FILE ',A5, '  500 PTS. MAX.'/)
20    X(1) = NO-1
      Y(1) = NO-1
      CALL RELEAS(20)
      RETURN
      END
C
C
      SUBROUTINE PLSCAL(X,Y,KFIL,K,XMI,XMA,YMI,YMA,MO,MM)
```

```
C
C       PLOT ONE DISK FILE USING GLOBAL MAXIMA AND MINIMA FOR
C        X AND Y ARRAYS.
C       MM IS 0--PLOT A DOTTED LINE FOR FILES 2,4,6, ETC.
C       MM IS 1--PLOT A SOLID LINE FOR FILES 1, 3, 5, ETC.
        DIMENSION X(1),Y(1),KFIL(1)
        IFI = KFIL(K)
        CALL READ(IFI, X,Y)
        CALL DLIMX(XMI,XMA)
        CALL DLIMY(YMI,YMA)
        IF(MO.EQ.1HY) GO TO 10
C       DELETE GRID WITH NEXT TWO CALLS
        CALL YFRM(2)
        CALL XFRM(2)
10      CALL LINE (0)
C       IF MM IS 0, THEN DOT LINES--DOTTED LINES FOR 2,4,..ETC
        IF (K.EQ.1)   GO TO 12
        IF (MM.EQ.0) CALL LINE(1212)
C       PLOT FILES 2,3,....ETC.
        CALL CPLOT(X,Y)
        RETURN
C
C       PLOT FILE 1
12      CALL CHECK(X,Y)
        CALL DSPLAY(X,Y)
        RETURN
        END
C
C       THIS SUBROUTINE FINDS THE MAXIMUM AND MINIMUM OF AN ARRAY
        SUBROUTINE MINMAX(XX,XX1,XX2)
        DIMENSION XX(1)
        N = XX(1)
        DO 10 K = 2,N
        XX1 = AMIN1(XX(K),XX1)
        XX2 = AMAX1(XX(K),XX2)
10      CONTINUE
        RETURN
        END
C       THIS SUBROUTINE WRITES ONE FILE TO DISK
        SUBROUTINE WRITE(OFIL, X,Y)
        DIMENSION X(1), Y(1)
        OPEN(UNIT=21, FILE=OFIL,ACCESS='SEQOUT')
C       NO. OF POINTS RESIDES IN X(1), Y(1)
        NO = X(1)
        NOP1=NO+1
        DO 20 I = 2, NOP1
10      WRITE (21, 100) X(I), Y(I)
100     FORMAT(2F)
20      CONTINUE
        CALL RELEAS (21)
        RETURN
        END
C
C
C       PRESET MINIMA AND MAXIMA OF X AND Y ARRAYS
        SUBROUTINE PRESET(XMIN,XMAX,YMIN,YMAX)
        XMIN = 1.E8
        XMAX =-1.E8
        YMIN = XMIN
        YMAX = XMAX
        RETURN
```

```
      END
C     THIS SUBROUTINE ACCEPTS LABELS AND TITLES FOR GRAPH
C     FROM THE KEYBOARD,  25 CHARACTERS PER TITLE OR LABEL.
      SUBROUTINE MYLABL(LTITLE,LABX, LABY)
      DIMENSION LTITLE(1),LABX(1),LABY(1),KTITLE(5)
16    CALL ANMODE
      TYPE 8
8     FORMAT(' TITLE?  '/)
      READ (5,11) KTITLE
11    FORMAT(5A5)
      CALL KAM2AS(25,KTITLE,LTITLE)
      DO 15 J = 1,5
15    KTITLE(J) = '
      CALL ANMODE
      TYPE 10
10    FORMAT('  LABEL FOR X-AXIS?  '/)
      READ(5,11) KTITLE
      CALL KAM2AS(25,KTITLE,LABX)
      DO 20 J = 1,5
20    KTITLE(J) ='
      CALL ANMODE
      TYPE 12
12    FORMAT('  LABEL FOR Y-AXIS?  '/)
      READ(5,11) KTITLE
      CALL KAM2AS(25,KTITLE,LABY)
      TYPE 14
14    FORMAT('  ALL TITLES AND LABELS OK?  '$)
      ACCEPT 5, MO
5     FORMAT(A1)
      IF (MO.EQ.'N')   GO TO 16
      RETURN
      END
C
C
```

APPENDIX II

Listing of Program IMP

```
      PROGRAM IMP
C     DENNIS GRADY, FEBRUARY, 1975
C     THE PRESENT PROGRAM PROVIDES A REDUCTION
C     OF VELOCITY INTERFEROMETER DATA TO STRESS-STRAIN
C     PATHS AND MODULI.
      COMMON/BICF  /XSAM(5),A(100),B(100), C(100),TV(5,100),NOF
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BICOS /RHO
      COMMON/BICOV /DUX
      COMMON/BIV   /U(500),T(500),TTV(100),NL(5),NT(5)
      COMMON/BIS   /S(100),E(100)
      COMMON/BICCS /UCOR(100)
      DIMENSION IFIL (10), DELU(100),NDEL(100),X1(100)
      LOGICAL SW, ICOPY
C
C     PRESET FOR TEK PLOTTING
C
      CALL SET4TK
      CALL INITT(30)
      CALL TERM (1,1024)
      CALL BINITT
      CALL ANMODE
C
C     OPERATION 1,2,3.

C     INPUT AND IDENTIFY ALL PROFILES, NORMALIZE, AND INCREMENT.

1     TYPE 7
7     FORMAT(' ENTER NUMBER OF FILES TO READ  '$)
      ACCEPT 9, NOF
9     FORMAT(I)
      DO 8 I = 1, 100
        NDEL (I) = 0
8       CONTINUE
      DO 36 J=1,NOF
        TYPE 11
11      FORMAT(' INPUT FILE?  '$)
        ACCEPT 12, IFIL(J)
12      FORMAT(A5)
36      CONTINUE
      TYPE 21
21    FORMAT(' ENTER VELOCITY INCREMENT  '$)
      ACCEPT 22, DUX
22    FORMAT(F)
      TYPE 23
      ACCEPT 47, NYES
23    FORMAT('  PLOT VELOCITY VS, TIME?'  $)
      DO 41 I = 1,NOF
        TYPE 39, I
39      FORMAT(' ENTER SAMPLE',I4,' THICKNESS.  '$)
        ACCEPT 40, XSAM(I)
40      FORMAT(G)
41      CONTINUE
      CALL NEWPAG
      XMIN = 1.E5
      XMAX = -1.E5
      DO 35 J = 1, NOF
        CALL READ(IFIL(J),NF)
        CALL VINC(NNT,NNL,NF, IFIL(J))
        NL(J)=NNL
        NT(J)=NNT
          DO 24 K = 1, NNT
```

```
            TV(J,K) = TTV(K)
24          CONTINUE
35        CONTINUE
        TYPE 70
70      FORMAT(' FIT PROFILE FEET TO PRESCRIBED VELOCITY?  '$)
        ACCEPT 47, MYES
        IF(MYES.EQ.'N') GO TO 38
        TYPE 72
72      FORMAT(' ENTER FOOT VELOCITY?  '$)
        ACCEPT 4?, CFOOT
        CALL FFIT(CFOOT)
38      IF (NYES .EQ. 'N')   GO TO 77
        DO 76 J = 1,NOF
          NNT = NT(J)
          DO 75 I = 1,NNT
            XMIN = AMIN1(XMIN,TV(J,I))
            XMAX = AMAX1(XMAX,TV(J,I))
75          CONTINUE
76        CONTINUE
        YMIN = 0.
        YMAX = YMIN
        DO 37 K = 1, NOF
          YMA = DUX * FLOAT(NL(K))
          YMAX = AMAX1( YMAX, YMA)
37        CONTINUE
        SW = .FALSE.
        DO 26 J = 1, NOF
          J1 = J
          NNT = NT(J)
          DO 28 K = 1, NNT
            X1(K) = TV(J,K)
28          CONTINUE
          NNL = NL(J)
          DO 25 K = 1, NNL
            DELU (K) = DUX * FLOAT (K-1)
25          CONTINUE
          NNL1 = NNL + 1
          DO 27 K = NNL1, NNT
            DELU(K) = DELU (NNL) - DUX * (FLOAT(K-NNL1))
27          CONTINUE
          CALL PLOT(X1,DELU,XMIN,XMAX,YMIN,YMAX,IFIL,SW,J1,NNT)
26        CONTINUE
        CALL TINPUT(KK)
77      CALL NEWPAG
        CALL ANMODE
C
C       DETERMINE WAVE VELOCITIES AT EACH
C       PARTICAL VELOCITY LEVEL.

        CALL CFIT
C       SUBRTN CFIT PROVIDES A LINEAR FIT TO EACH VELOCITY
C       LEVEL AND RETURNS A(I) AND B(I) IN THE RELATION T=A+BX,
C       AND C(I), THE SOUND SPEED.
        TYPE 65
65      FORMAT('  WHAT CHARACTERISTICS TO PLOT?   '$)
        ACCEPT 9, ND
        DO 45 K = 1, NCT, ND
          NDEL (K) = 1
45        CONTINUE
        XMAX = -1.E5
        DO 63 K = 1, NOF
```

```
          XMAX = AMAX1(XMAX,XSAM(K))
          XMIN = AMIN1(XMIN,XSAM(K))
63        CONTINUE
61      TYPE 64
64      FORMAT('  ENTER TMIN,TMAX  '$)
        ACCEPT 49, TMIN, TMAX
49      FORMAT(2G)
        CALL NEWPAG
        CALL XTPLOT(NDEL,TMIN,TMAX)
        CALL HOME
        CALL TINPUT(KKK)
        CALL ANMODE
        TYPE 46
46      FORMAT('  SHIFT TIME ON ANY PROFILE?  '$)
        ACCEPT 47, MYES
47      FORMAT(A1)
        IF (MYES.EQ. 'N')   GO TO 60
        TYPE 48
48      FORMAT('  WHICH PROFILE ?  '$)
        ACCEPT 9, NS
        TYPE 50
50      FORMAT('  TIME SHIFT?  '$)
        ACCEPT 22, TS
        CALL TSHIFT(NS, TS)
        CALL CFIT
        TYPE 51
51      FORMAT( '  REPLOT ?  '$)
        ACCEPT 47, MYES
        IF (MYES. EQ. 'Y')   GO TO 61
60      CALL ANMODE
        TYPE 52
52      FORMAT('  ADD ANY CHARACTERISTICS?  '$)
        ACCEPT 47, MYES
        IF(MYES.EQ.'Y') CALL DELADD(NDEL,1)
53      TYPE 54
54      FORMAT('  DELETE ANY CHARACTERISTICS?  '$)
        ACCEPT 47, MYES
        IF (MYES. EQ. 'Y')       CALL DELADD(NDEL,0)
        TYPE 51
        ACCEPT 47, MYES
        IF (MYES. EQ. 'N')   GO TO 79
        GO TO 61
C
79      TYPE 80
80      FORMAT('  DO YOU WISH TO ADJUST THE WAVE VELOCITY DATA? '$)
        ACCEPT 47, LYES
        IF (LYES .EQ. 'N')  GO TO 66
        CALL CADJST(IFIL,DELU)
C
C       CORRECT FOR THE MATERIAL-WINDOW INTERFACE IMPEDANCE MISMATCH.
66      CALL NEWPAG
        CALL ANMODE
        TYPE 13
13      FORMAT(' ENTER SAMPLE DENSITY IN GRAMS/CC  '$)
        ACCEPT 14,RHO
14      FORMAT(G)
        RHO = RHO * 10.
        CALL NEWPAG
        CALL CORVEL
C       SUBROUTINE CORVEL RETURNS THE CORRECTED WAVE VELOCITIES.
C       DETERMINE THE STRESS-STRAIN DATA.
```

```
99      CALL STST
C       SUBRROUTINE STST RETURNS CORRESPONDING STRESS-STRAIN DATA.
C
        TYPE 81
81      FORMAT(' DO YOU WISH A PRINT OUT OF THE DATA? '$)
        ACCEPT 47, LYES
        IF(LYES.EQ.'N') GO TO 100
        CALL NEWPAG
        CALL ANMODE
        TYPE 82
82      FORMAT('                          LOADING DATA  '//)
        TYPE 83
83      FORMAT('            STRESS    STRAIN    PARTICLE   WAVE ')
        TYPE 84
84      FORMAT('                                VELOCITY   VELOCITY'/)
        LINE = 5
        DO 88 I=1,NCL
          LINE = LINE + 1
          ICOPY = LINE .GT. 30
          IF (ICOPY)  CALL HDCOPY
          IF (ICOPY)  TYPE 83
          IF (ICOPY)  LINE = 0
          TYPE 85,I,S(I),E(I),UCOR(I),C(I)
88        CONTINUE
85      FORMAT(1H ,I3,3X,F10.2,2F10.4,F8.2)
        CALL TINPUT(KKK)
        CALL NEWPAG
        CALL ANMODE
        TYPE 86
86      FORMAT('                        UNLOADING DATA  '//)
        TYPE 83
        TYPE 84
        LINE = 5
        ICOPY = .FALSE.
        NCLP1=NCL+1
        DO 89 I=NCLP1,NCT
          LINE = LINE + 1
          ICOPY = LINE .GT. 30
          IF (ICOPY)   CALL HDCOPY
          IF (ICOPY)   TYPE 83
          IF (ICOPY)   LINE = 0
          TYPE 85,I, S(I),E(I),UCOR(I),C(I)
89        CONTINUE
        CALL TINPUT(KKK)
100     CALL NEWPAG
        TYPE 110
110     FORMAT(' DO YOU WISH TO MAKE OUTPUT DATA FILES? '$)
        ACCEPT 47,LYES
        IF(LYES.EQ.'N') GO TO  115
        CALL WRITE(1,E,S,1,NCT)
        CALL WRITE(2,UCOR,S,1,NCT)
        CALL WRITE(3,S,C,1,NCL)
        NCL1=NCL+1
        CALL WRITE(4,S,C,NCL1,NCT)
115     CALL RESULT
        TYPE 67
67      FORMAT('  REDUCE ANOTHER SET OF DATA?  '$)
        ACCEPT 47, MYES
        IF (MYES .EQ. 'Y')    GO TO 1
        CALL FINITT(0,0)
```

```
      CALL CLR4TK
      END

      SUBROUTINE WRITE(N0,A,B,N1,N2)
      DIMENSION A(1),B(1)
      IF(N0.EQ.1) TYPE 111
      IF(N0.EQ.2) TYPE 112
      IF(N0.EQ.3) TYPE 113
      IF(N0.EQ.4) TYPE 114
111   FORMAT(' NAME STRESS-STRAIN FILE '$)
112   FORMAT(' NAME STRESS-PARTICLE VELOCITY FILE '$)
113   FORMAT(' NAME LOADING WAVE VELOCITY-STRESS FILE '$)
114   FORMAT(' NAME UNLOADING WAVE VELOCITY-STRESS FILE '$)
      ACCEPT 18, IDFIL
18    FORMAT(A5)
      OPEN(UNIT=20,FILE=IDFIL,ACCESS='SEQOUT')
      WRITE(20,120)(A(I),B(I),I=N1,N2)
120   FORMAT(2F)
      CALL RELEAS(20)
      RETURN
      END
C
C
      SUBROUTINE DELADD(NDEL ,ISW)
C     THIS ROUTINE SETS FLAGS TO DELETE OR ADD CHARACTERISTIC LINES
      DIMENSION NDEL(1), NDELTA(10)
      TYPE 10
10    FORMAT('  HOW MANY?  (<=10)'$)
      ACCEPT 12, NUMB
12    FORMAT(10I)
      TYPE 14
14    FORMAT('  ENTER ALL CHARACTERISTIC NOS,  ON ONE LINE '/)
      ACCEPT 12, (NDELTA(K), K = 1, NUMB)
      DO 16 K = 1, NUMB
        I =NDELTA(K)
        IF(ISW.EQ.1) NDEL(I) = 1
        IF(ISW.EQ.0) NDEL(I) = 0
16      CONTINUE
      RETURN
      END
C
C
      SUBROUTINE TSHIFT(NS,TS)
C     TIME SHIFT ARRAY TV BY TIME, TS
      COMMON /   BIV/U(500),T(500),TTV(100),NL(5),NT(5)
      COMMON /  BICF/ XSAM(5), A(100), B(100), C(100), TV(5,100),NOF
      COMMON/BICCCS /NCL,NCT,TORDER(5,100)
      NTT = NT(NS)
      DO 10 K = 1, NTT
        TV(NS,K) = TV(NS,K) + TS
10      CONTINUE
      RETURN
      END
C
      SUBROUTINE FFIT(CFOOT)
      COMMON/   BIV/U(500),T(500),TTV(100),NL(5),NT(5)
      COMMON/  BICF/XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      DO 10 I = 1,NOF
        TCOR = TV(I,1)-XSAM(I)/CFOOT
        NTT = NT(I)
        DO 8 J = 1,NTT
```

```
            TV(I,J) = TV(I,J)-TCOR
8           CONTINUE
10        CONTINUE
        RETURN
        END
C
C
C       PLOT X VS. TIME
        SUBROUTINE XTPLOT(NDEL,TMIN,TMAX)
        COMMON/BICF  /XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
        COMMON/BICCCS/NCL,NCT,TORDER(5,100)
        LOGICAL ITRIP
        DIMENSION TIME(100),X(100),APT(3),XX(3),NDEL(1),LAB1(6),LAB2(6)
        DATA LAB1/5H     ,5H     ,5HDISTA,5HNCE  ,5H     ,5H     /
        DATA LAB2/5H     ,5H     ,5HTIME ,5H     ,5H     ,5H     /
        CALL BINITT
        XMIN = 0.
        XMAX = XSAM(NOF) + 1.
        ITRIP = .TRUE.
        DO 10 J = 1, NCT
          IF (NDEL (J) .EQ. 0)   GO TO 10
          N = 1
          DO 12 K = 1, NOF
            IF (TORDER(K,J) .LT. 1.E-4)   GO TO 12
            N = N + 1
            X(N) = XSAM (K)
            TIME(N) = TORDER(K,J)
12          CONTINUE
          TIME (1) = FLOAT (N-1)
          X(1) = FLOAT (N-1)
          APT (1) = 2.
          APT (2) = A(J)
          APT (3) = A(J) + B(J) *( XSAM(NOF) +1.)
          XX (1) = 2.
          XX (2) = 0.
          XX (3) = XMAX
          CALL XFRM(2)
          CALL YFRM(2)
          CALL YNEAT(0)
          CALL DLIMX(XMIN,XMAX)
          CALL DLIMY(TMIN,TMAX)
          CALL LINE (-1)
          CALL SIZES(0.25)
          CALL SYMBL(2)
          CALL CHECK(X,TIME)
          IF (ITRIP)   GO TO 14
          CALL CPLOT (X, TIME)
          CALL LINE(0)
          CALL SYMBL(0)
          CALL CPLOT (XX, APT)
          GO TO 10
14        CALL DSPLAY( X, TIME)
          CALL LINE (0)
          CALL SYMBL (0)
          CALL CPLOT (XX, APT)
          ITRIP = .FALSE.
10        CONTINUE
        IF (TMIN .GE. 0 )  GO TO 18
        APT(2) = 0.
        APT(3) = 0.
        CALL LINE (5252)
```

```
      CALL CPLOT (XX, APT)
18    CONTINUE
      CALL PUTON (LAB1, LAB2)
      RETURN
      END
C
      SUBROUTINE VINC(NNT,NNL,NF, IFI)
      LOGICAL JHI, PEAK
      COMMON/BICOV/DUX
      COMMON/BIV    /U(500),T(500),TTV(100),NL(5),NT(5)
C
C     NOISE NOT USED IN THE RAW DATA TRACE
      DO 8 I=1,50
        DIF=ABS(U(I+1)-U(I))
        IF(DIF.GE..0001) GO TO 9
8       CONTINUE
      TYPE 10, IFI
10    FORMAT('  TTV(1) NOT FOUND IN FIRST 50 DATA POINTS
      1 FROM FILE ',A5,' IN S.R. VINC')
      CALL EXIT
C
C     FIRST POINT OF TTV ARRAY IS T(I)
9     TTV(1) = T(I)
      JHI = .FALSE.
      PEAK = .FALSE.
      UX = DUX
      NNL = 1
      J = I - 1
20    J = J + 1
C     J WITHIN RAW DATA?
      IF (J .GT. (NF-1))  JHI =.TRUE.
      IF (.NOT. JHI)   GO TO 21
      TYPE 23,IFI
23    FORMAT('  NO PEAK FOUND  ON LOADING CURVE
      1  IN FILE ',A5,'  IN S. R. VINC.')
      CALL EXIT
C     CALCULATE LOADING DATA
21    DIF = T(J+1) - T(J)
      IF (DIF.LT. (1.0E-5))   PEAK = .TRUE.
      IF (PEAK)   GO TO 26
      IF (U(J) .LT. UX)   GO TO 20
      NNL = NNL +1
      S = (U(J) - U(J-1))/(T(J) - T(J-1))
      TTV(NNL) = T(J) - (U(J) - UX)/S
      UX = UX + DUX
      J = J - 1
      GO TO 21
C
C     PEAK FOUND
C     ALL LOADING DATA CALCULATED
26    NNT = NNL
      UX = UX - DUX
C     J WITHIN RAW DATA?
30    IF (J. GT. (NF-1))  JHI = .TRUE.
C     ALL RAW DATA HAS BEEN USED
      IF (JHI)    RETURN
      IF (U(J) .LE. UX)    GO TO 32
      J = J + 1
      GO TO 30
C     CALCULATE UNLOADING DATA
32    NNT = NNT + 1
```

```
      S = (U(J) -  U(J-1))/(T(J) -T(J-1))
      TTV(NNT) = T(J) -(U(J) - UX)/S
      UX = UX - DUX
      J = J - 1
      GO TO 30
      END
C
C
      SUBROUTINE CFIT
      COMMON/BICF  / XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      COMMON/BIV   /U(500),T(500),TTV(100),NL(5),NT(5)
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      DIMENSION NSW(5),NX(5)
      NMAX = 0
      DO 3 I = 1,  NOF
        NMAX = MAX0(NMAX, NL(I))
3       CONTINUE
      DO 4 I = 1, NOF
        NX(I) = NL(I)
4       CONTINUE
C     ARRANGE ARRAY NX IN HI TO LO ORDER
      NOF1 = NOF -1
      DO 5 I = 1, NOF1
        IST = I + 1
        DO 7 J = IST, NOF
          IF (NX(J) - NX(I) .LT. 0)   GO TO 7
          SAVE = NX(J)
          NX(J) = NX(I)
          NX(I)  = SAVE
7       CONTINUE
5     CONTINUE
      NCL =   NX(2)
C
C     ORDER TIME ARRAY TO AN ARRAY FOR FITTING CHARACTERISTICS
      DO 10 J =  1, NOF
        DO 14 I = 1, 100
          TORDER (J, I) = 0.
14        CONTINUE
10      CONTINUE
      DO 50 J = 1, NOF
        IDIF = NCL - NL(J)
        LENGTH = NT(J) + 2 * IDIF
        DO 30 I = 1, LENGTH
          IF (I .LE. NCL)   TORDER(J,I) = TV(J,I)
          IF (I .GT. NCL)   TORDER(J,I) = TV(J,I-2*IDIF )
30        CONTINUE
        IF (IDIF.LE. 0)    GO TO 50
        LMIN = NCL - IDIF +1
        LMAX = NCL + IDIF
        DO 45  M = LMIN, LMAX
          TORDER (J,M) = 0.
45        CONTINUE
50      CONTINUE
      NCT = 1
12    DO 16 K = 1, NOF
        NSW(K) = 0
16      CONTINUE
      NTOT = 0
      DO 17 J = 1, NOF
        TTEST = TORDER(J,NCT)
        IF (TTEST .LT. 1.E-4)    GO TO 17
```

```
          NSW(J) = 1
          NTOT = NTOT + 1
17        CONTINUE
        IF (NTOT .LT. 2)   GO TO 55
C       DETERMINE LEAST SQUARE POINTS
        SUMX = 0.
        SUMT = 0.
        SUMXT = 0.
        SUMX2 = 0.
        NSUM = NTOT
        DO 18 IS = 1,NOF
          SUMX = SUMX + NSW(IS) * XSAM(IS)
          SUMT = SUMT + NSW(IS) * TORDER(IS,NCT)
          SUMXT = SUMXT + NSW(IS) * XSAM(IS) * TORDER(IS,NCT)
          SUMX2 = SUMX2 + NSW(IS) * XSAM(IS)**2
18        CONTINUE
        DEN = NSUM * SUMX2 - SUMX**2
        ANUM = SUMT * SUMX2 - SUMXT * SUMX
        BNUM = NSUM * SUMXT - SUMX * SUMT
        A(NCT) = ANUM / DEN
        B(NCT) = BNUM / DEN
        C(NCT) = 1./ B(NCT)
        NCT = NCT + 1
        GO TO 12
55      NCT = NCT - 1
        RETURN
        END

        SUBROUTINE CORVEL

C   << COMPUTES CORRECTED PARTICLE VELOCITIES >>
        COMMON/BICCCS/NCL,NCT,TORDER(5,100)
        COMMON/BICCS/UCOR(100)
        COMMON/BICOS/RHO
        COMMON/BICOV/DUx
        COMMON/BICF/XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
        DIMENSION FP(100)
C   << INITIAL SETTING OF F' >>
        DO 10 I=1,NCT
10      FP(I)=RHO*C(I)
        NS=0
C   << EVALUATION OF CORRECTED U >>
15      UCOR(1)=0.
        DO 20 I=2,NCL
        UCOR(I)=UCOR(I-1)+0.5*(1.0+GP(I-1)/FP(I-1))*DUx
20      CONTINUE
        UCOR(NCL+1)=UCOR(NCL)
        NCLP2=NCL+2
        DO 30 I=NCLP2,NCT
        UCOR(I)=UCOR(I-1)-0.5*(1.0+GP(I-1)/FP(I-1))*DUx
30      CONTINUE
        IF(NS.EQ.1) RETURN
        NS=NS+1
C   << REEVALUATE F' >>
        DO 40  I=2,NCT
        J=I
        FTEM=FPP(J)
40      FP(I)=RHO*FTEM
        GO TO 15
        END
C
```

```
C
      FUNCTION FPP(I)
C   << REEVALUATE F' FOR SUBRTN CORVEL >>
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BICF/XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      IF(I.GT.NCL) GO TO 10
C   << LOADING >>
      STEST=GG(I)
      CALL SINT(STEST,SF,CF,1)
      IF(SF.GT.0.) FPP=CF
      IF(SF.GT.0.) RETURN
      FPP=C(NCL)
      RETURN
C   << UNLOADING >>
10    STEST=GG(NCL)
      CALL SINT(STEST,SF,CF,1)
      IF(SF.GT.0.) GO TO 20
      IF(GG(I).GE.SF) FPP=C(NCL+1)
      IF(GG(I).GE.SF) RETURN
20    STEST=GG(NCL)
      CALL SINT(STEST,SF,CF,2)
      IF(SF.GT.0.) FPP=CF
      IF(SF.GT.0.) RETURN
      FPP=C(NCT)
      RETURN
      END
C
C
      SUBROUTINE SINT(STEST,SF,CF,IS)
C   << INTEGRATES CURRENT CORRECTED P-U PATH >>
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BICF/XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      COMMON/BICCS/UCOR(100)
      COMMON/BICCS/RHO

      IF (IS.EQ.2) GO TO 30
      S=0.
      DO 20 I=2,NCL
      Z=RHO*(C(I)+C(I-1))/2.
      R=S
      S=S+(UCOR(I)-UCOR(I-1))*Z
      IF(S.LT.STEST) GO TO 20
      CF=C(I-1)+(C(I)-C(I-1))*(STEST-R)/(S-R)
      SF=S
      RETURN
20    CONTINUE
      SF=-1.0
      RETURN
30    S=0.
      DO 40 I=2,NCT
      Z=RHO*(C(I)+C(I-1))/2.
      R=S
      S=S+(UCOR(I)-UCOR(I-1))*Z
      IF(I.LE.NCL) GO TO 40
      IF(S.GT.STEST) GO TO 40
      CF=C(I-1)+(C(I)-C(I-1))*(STEST-R)/(S-R)
      SF=S
      RETURN
40    CONTINUE
      SF=-1.0
      RETURN
```

```
      END
C
C
      FUNCTION GG(I)
C   << P-U EQUATION OF STATE FOR WINDOW MATERIAL >>
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BICOV/DUX
      IF(I.LE.NCL) U=(I-1)*DUX
      IF(I.GT.NCL) U=(2*NCL-I)*DUX
      GG=131.7*U-73.51*U**2+99.47*U**3-41.63*U**4
      RETURN
      END
C
      FUNCTION GP(I)
C   << P-U DERIVATIVE FOR WINDOW MATERIAL EOS >>
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BICOV/DUX
      IF(I.LE.NCL) U=(I-1)*DUX
      IF(I.GT.NCL) U=(2*NCL-1)*DUX
      GP=131.7-147.22*U+298.41*U**2-166.52*U**3
      RETURN
      END
C
C
      SUBROUTINE STST
C
C     COMPUTE STRESS STRAIN DATA
      COMMON/BICF  /XSAM(5), A(100),B(100),C(100),TV(5,100),NOF
      COMMON/BICOS/RHO
      COMMON/BICCCS/NCL,NCT,TORDER(5,100)
      COMMON/BIS/S(100),E(100)
      COMMON/BICOS/UCOR(100)
      LOGICAL EQUAL
      S(1)=0.
      E(1)=0.
      EQUAL = .FALSE.
      DO 20 I=2,NCT
        EQUAL = I .EQ. (NCL+1)
        IF(.NOT. EQUAL)  S(I)=S(I-1)+RHO*(UCOR(I)-UCOR(I-1))*
     1 0.5*(C(I)+C(I-1))
        IF(.NOT. EQUAL)  E(I)=E(I-1)+(UCOR(I)-UCOR(I-1))/
     1 (0.5*(C(I)+C(I-1)))
        IF (EQUAL)  S(I) = S(I-1)
        IF (EQUAL)  E(I) = E(I-1)
20      CONTINUE
      RETURN
      END
C                    READ ONE FILE FROM DISK
      SUBROUTINE READ(IFILE,NF)
      COMMON /BIV   /U(500),T(500),TTV(100),NL(5), NT(5)
      OPEN (UNIT= 20, FILE = IFILE, ACCESS = 'SEQIN')
      NF = 0
      DO 10 I = 1,500
        READ (20, 106,END = 20) T(I),U(I)
106     FORMAT(2G)
        NF = NF +1
10      CONTINUE
      TYPE 22, IFIL
22    FORMAT(' NOT ALL DATA READ FROM FILE ',A5, '  500 PTS. MAX,'/)
20    CONTINUE
      CALL RELEAS(20)
```

```
      RETURN
      END
C
C
      SUBROUTINE PLOT(X1,Y,XMI,XMA,YMI,YMA,IFIL,SW,K,NO)
      DIMENSION Y(1),X1(1),ISYM(12),IFIL(1)
      LOGICAL SW
      COMMON /BICF  /XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      DIMENSION LAB1(6), LAB2(6), LAB3(6), LAB4(6)
      DATA ISYM/1,1,1,1,1,1,1,1,1,1,1,1/
C
      DATA LAB1/5H      ,5H      ,5HTIME ,5H      ,5H      ,5H      /
      DATA LAB2/5H      ,5H      ,5HVELOC,5HITY  ,5H      ,5H      /
      DATA LAB3/5H      ,5HSOUND,5HVELOC,5HITY  ,5H      ,5H      /
      DATA LAB4/5H      ,5HPARTI,5HCLE V,5HELOCI,5HTY    ,5H      /
      CALL DLIMX(XMI,XMA)
      CALL DLIMY(YMI,YMA)
      CALL LINE (-1)
      CALL SIZES (0,15)
      CALL SYMBL (ISYM(K))
      IF (K.EQ.1)   GO TO 12
C     PLOT FILES 2, 3,....,ETC.
      CALL NPTS(NO)
      CALL CHECK( X1, Y)
      CALL CPLOT(X1,Y)
      IF (SW)  GO TO 27
      IF (K .LT. NOF) RETURN
      CALL PUTON (LAB1, LAB2)
      CALL MOVABS(40,760)
      CALL ANMODE
      TYPE 26,(IFIL(K1),K1=1,NOF)
26    FORMAT(' FILES--', 10A7)
      RETURN
27    CONTINUE
      CALL PUTON(LAB4,LAB3)
      RETURN
C
C     PLOT FILE 1
C     DRAW BOX
12    CALL FRAME
      CALL NPTS(NO)
      CALL CHECK(X1,Y)
      CALL DSPLAY(X1,Y)
      RETURN
      END
C
      SUBROUTINE RESULT
      COMMON /  BICF/ XSAM(5), A(100), B(100), C(100), TV(5,100), NOF
      COMMON /  BIV /U(500), T(500), TTV(100), NL(5), NT(5)
      COMMON /  BIS /S(100), E(100)
      COMMON /BICCS /UCOR(100)
      COMMON /BICCCS/NCL,NCT,TORDER(5,100)
      DIMENSION LAB1(6),LAB2(6), LAB3(6),LAB4(6)
      DATA LAB1/5H      ,5H      ,5HSTRES,5HS     ,5H      ,5H      /
      DATA LAB2/5H      ,5HSOUND,5H VELO,5HCITY ,5H      ,5H      /
      DATA LAB3/5H      ,5H      ,5HSTRAI,5HN     ,5H      ,5H      /
      DATA LAB4/5H      ,5HPARTI,5HCLE V,5HELOCI,5HTY    ,5H      /
C
C     KEY TO REQUESTS FOR PLOTTING
C
      CALL NEWPAG
```

```
      CALL ANMODE
      TYPE 4
4     FORMAT('  TO OBTAIN GRAPH STRESS-STRAIN, TYPE "SE"'/
      118X, 'STRESS-PARTICLE VELOCITY, TYPE "SV"'/
      218X, 'SOUND VELOCITY-STRESS,  TYPE "CS"'/
      318X, 'SOUND VELOCITY-PARTICLE VELOCITY, TYPE "CV"')
5     CALL ANMODE
      TYPE 12
12    FORMAT('  WHICH GRAPH?,,,,TYPE "NONE" TO EXIT ')
      ACCEPT 10, NEM
10    FORMAT(A3)
      CALL NEWPAG
      IF (NEM .EQ. 'NON')   RETURN
      CALL BINITT
      CALL NPTS(NCT)
C
C     STRESS-STRAIN PLOT
      IF (NEM .NE. 'SE')  GO TO 20
      CALL CHECK (E,S)
      CALL DSPLAY (E,S)
      CALL  PUTON (LAB3, LAB1)
      CALL TINPUT(KKK)
      GO TO 60
C
C     STRESS- PARTICLE VELOCITY (CORRECTED)  PLOT
20    IF( NEM .NE. 'SV')   GO TO 30
      CALL CHECK (UCOR, S)
      CALL DSPLAY ( UCOR,S)
      CALL PUTON (LAB4, LAB1)
      CALL TINPUT(KKK)
      GO TO 60
C
C     STRESS-SOUND VELOCITY PLOT
30    IF (NEM .NE. 'CS')  GO TO 50
      CALL CHECK (S,C)
      CALL DSPLAY(S,C)
      CALL PUTON (LAB1, LAB2)
      CALL TINPUT(KKK)
      GO TO 60
C
50    IF (NEM .NE. 'CV')  GO TO 5
      CALL CHECK ( UCOR,C)
      CALL DSPLAY( UCOR,C)
      CALL PUTON (LAB4, LAB2)
      CALL TINPUT(KKK)
60    CALL HOME
      GO TO 5
      END
C
C           THIS SUBROUTINE PUTS TITLE AND LABELS ON A GRAPH
      SUBROUTINE PUTON(LABX,LABY)
      DIMENSION LTITLE(1),LABX(1),LABY(1),LAB(30)

C     CALL MOVABS(300,780)
C     CALL HLABEL(30,LTITLE)
      CALL MOVABS(5,660)
      CALL KAM2AS(30,LABY,LAB)
      CALL VLABEL(30,LAB)
      CALL MOVABS(300,25)
      CALL KAM2AS(30,LABX,LAB)
      CALL HLABEL(30,LAB)
```

```
      CALL ANMODE
      RETURN
      END
      SUBROUTINE CADJST(IFIL,DELU)
C     ADJUST  THE WAVE VELOCITY DATA
C
      COMMON/ BICF /XSAM(5),A(100),B(100),C(100),TV(5,100),NOF
      DIMENSION DELU(1),IFIL(1)
      COMMON /BICOV /DUX
      COMMON/ BICCCS/NCL,NCT,TORDER(5,100)
C
C
      LOGICAL SW
      SW = .TRUE.
      DO 5 K = 1, NCL
        DELU (K) = DUX * FLOAT(K-1)
5       CONTINUE
      NL1 = NCL +1
      DO 6 K = NL1, NCT
        DELU(K) = DELU(NCL) - DUX * FLOAT(K-NL1)
6       CONTINUE
2     CALL BINITT
      UMIN = 0.
      UMAX = DUX * FLOAT(NCL)
      CMIN = 1.E5
      CMAX = -CMIN
      DO 8 K = 1, NCT
        CMIN = AMIN1( CMIN, C(K))
        CMAX = AMAX1( CMAX, C(K))
8       CONTINUE
      CALL NEWPAG
      CALL PLOT(DELU,C,UMIN,UMAX,CMIN,CMAX,IFIL,SW,1,NCT)
      CALL HOME
      CALL TINPUT(KKK)
      CALL ANMODE
      TYPE 13
13    FORMAT(' DO YOU WISH TO ADJUST PARTICULAR POINTS? '$)
      ACCEPT 14, IYES
14    FORMAT(A1)
      IF(IYES.EQ.'N') GO TO 40
      TYPE 15
15    FORMAT(' ENTER NUMBER OF POINTS TO CORRECT. '$)
      ACCEPT 20,NTS
20    FORMAT(I)
      DO 30 I = 1,NTS
        TYPE 23
23      FORMAT(' ENTER POINT NUMBER AND NEW VELOCITY. '$)
        ACCEPT 25,NPT,CNEW
25      FORMAT(2G)
        C(NPT) = CNEW
30      CONTINUE
40    TYPE 45
45    FORMAT(' DO YOU WISH PROFILE VELOCITIES SCALED AT A
     1  PARTICULAR LEVEL?  '$)
      ACCEPT 14, IYES
      IF (IYES. EQ. 'N')   GO TO 80
      TYPE 55
55    FORMAT('  ENTER LEVEL NUMBER AND NEW VELOCITY  '$)
      ACCEPT 60, NLEV, CNEW
60    FORMAT(2G)
      CNLEV = C(NLEV)
```

```
      DO 70 I = 1, NCT
        C(I) = 1. / (1./C(I) -1./ CNLEV + 1. /CNEW)
70      CONTINUE
80    TYPE 85
85    FORMAT('  REPLOT?'$)
      ACCEPT 14, IYES
      IF (IYES. EQ. 'Y')   GO TO 2
      RETURN
      END
```

REFERENCES

1. Bernstein, C., and D. D. Keough, "Piezoresistivity of Manganin," J. Appl. Phys., 41, 13 (1964).
2. Graham, R. A., F. W. Neilson, and W. B. Benedick, "Piezoelectric Current from Shock-Loaded Quartz - A Submicrosecond Stress Gauge," J. Appl. Phys. 36, 1775 (1965).
3. Dremin, A. N., and K. K. Shvedov, "The Determination of Chapman-Jougnet Pressure and of the Duration of Reaction in the Detonation Wave of High Explosives," Zh. Prikl. Mekh. Tekh. Fiz., 2, 154-159 (1964).
4. Barker, L. M., and R. E. Hollenbach, "Interferometer Technique for Measuring the Dynamic Mechanical Properties of Materials," 36, 1617 (1965).
5. Barker, L. M., and R. E. Hollenbach, "Laser Interferometry for Measuring High Velocities of Any Reflecting Surface," J. Appl. Phys., 43, 4669 (1972).
6. Grady, D. E., R. E. Hollenbach, K. W. Schuler and J. Callender, "Compression Wave Studies in Blair Dolomite," Sandia Laboratories Report SAND 76-0005 (1976).
7. Schuler, K. W., and D. E. Grady, "Compression Wave Studies in Solenhofen Limestone," in preparation.
8. Fowles, R. and R. F. Williams, "Plane Stress Wave Propagation in Solids," J. Appl. Phys., 41, 460 (1970).
9. Cowperthwaite, M. and R. F. Williams "Determination of Constitutive Relations with Multiple Gauges in Non-divergent Flow," J. Appl. Phys., 42, 456 (1971).

10. Grady, D. E., "Experimental Analysis of Spherical Wave Propagation," J. Geophys. Res., 78, 1299 (1973).

11. Seaman, L., "Lagrangian Analysis for Multiple Stress or Velocity Gauges in Attenuating Waves," J. Appl. Phys., 45, 4303 (1974).

12. Barker, L. M., "Visar Data Reduction," Sandia Laboratories Rept. SLA-73-1038 (1974).

13. Barker, L. M. and R. E. Hollenbach, "Shock-Wave Studies of PMMA, Fused Silica, and Sapphire," J. Appl. Phys., 41, 4208 (1970)

UNLIMITED DISTRIBUTION

DISTRIBUTION:

U.S. Energy Research and Development Administration
Bethesda Technical Library
Washington, D.C. 20545

U.S. Energy Research and Development Administration
ERDA Library
Washington, D.C. 20545

U.S. Energy Research and Development Administration
Albuquerque Operations Office
P.O. Box 5400
Albuquerque, NM 87115

D. K. Nowlin, Director
Special Programs Division
U.S. Energy Research and Development Administration
Albuquerque, NM 87115

T. J. Ahrens
California Institute of Technology
Pasadena, CA 91109

W. Brace
Massachusetts Inst. of Technology
Cambridge, MA 02139

J. F. Callender (5)
University of New Mexico
Albuquerque, NM 87106

W. J. Carter, M-6
Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, NM 87544

R. J. Clifton
Dept. of Engineering
Brown University
Providence, RI 02912

D. R. Curran, G178
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, CA 94025

G. Duvall
Dept. of Physics
Washington State University
Pullman, WA 99163

G. R. Fowles
Dept. of Physics
Washington State University
Pullman, WA 99163

S. I. Green
Terra Tek, Inc.
420 Wakara Way
Salt Lake City, UT 84108

D. Grine
Systems, Science & Software, Inc.
P.O. Box 1620
La Jolla, CA 92037

H. C. Heard, L-437
Lawrence Livermore Laboratory
P.O. Box 808
Livermore, CA 94550

G. Jackson
USAE Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

L. M. Barker
Terra Tek, Inc.
420 Wakara Way
Salt Lake City, UT 84108

J. Ramsey
U.S. ERDA Fossil Energy
2100 M. Street N. W.
Washington, DC 20037

R. N. Schock, L-437
Lawrence Livermore Laboratory
P.O. Box 808
Livermore, CA 94550

L. Seaman
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, CA 94025

DISTRIBUTION (CONT.)

T. Stong, SPSS
Director, Defense Nuclear Agency
Washington, DC 20305

J. B. Walsh
Massachusetts Inst. of Technology
Cambridge, MA 02139

L. Lee
KTech.
P.O. Box 5490
Kirtland AFB
Albuquerque, NM 87115

2513 D. B. Hayes
2513 D. E. Mitchell
4363 W. E. Alzheimer
5000 A. Narath
Attn: 5200 E. H. Beckner
5400 A. W. Snyder
5700 J. H. Scott
5800 R. S. Claassen
5100 J. K. Galt
Attn: 5110 F. L. Vook
5120 G. J. Simmons
5130 G. A. Samara
5150 J. E. Schirber
5131 J. E. Kennedy
5160 W. Herrmann
5162 L. D. Bertholf
5163 D. E. Munson
5163 D. E. Grady (25)
5163 R. E. Hollenbach
5163 K. W. Schuler
5163 R. A. Schmidt
5163 E. G. Young (5)
5163 W. R. Wawersik
5166 A. J. Chabai
5166 R. K. Byers
5167 B. M. Butcher
5167 J. R. Asay
5734 A. L. Stevens
8266 Tech. Library (2)
3141 Tech. Library (5)
3151 Tech. Writing (3)
For ERDA/TIC (Unlimited Release)
ERDA/TIC (25)
(3171-1 R. P. Campbell)