Methodology for Determining Time-Dependent Mechanical Properties of Tuff SubJECTED TO Near-Field Repository Conditions

J. D. Blacic
R. Andersen
METHODOLOGY FOR DETERMINING TIME-DEPENDENT MECHANICAL PROPERTIES OF TUFF SUBJECTED TO NEAR-FIELD REPOSITORY CONDITIONS

by

J. D. Blacic and R. Andersen

ABSTRACT

We have established a methodology to determine the time dependence of strength and transport properties of tuff under conditions appropriate to a nuclear waste repository. Exploratory tests to determine the approximate magnitudes of thermo-mechanical property changes are nearly complete. In this report we describe the capabilities of an apparatus designed to precisely measure the time-dependent deformation and permeability of tuff at simulated repository conditions. Preliminary tests with this new apparatus indicate that microclastic creep failure of tuff occurs over a narrow strain range with little precursory Tertiary creep behavior. In one test, deformation under conditions of slowly decreasing effective pressure resulted in failure, whereas some strain indicators showed a decreasing rate of strain.

I. INTRODUCTION

The purpose of this preliminary report is to introduce the topic of time-dependent mechanical properties of tuff and to describe an approach for measuring these properties.

The objectives are listed below.

(1) Measure deformation of tuff over an extended time at conditions of stress, temperature, confining pressure, and water pore pressure likely to be encountered in a nuclear waste repository. From these exploratory tests, we will determine if the magnitude of strength and mineralogic changes warrants a more extensive set of measurements proposed in (2) and (3) below.
(2) Measure deformation of tuff over a sufficient range of physical variables to enable the formulation of a constitutive description that can be used in numerical models for design and performance evaluation of a repository. Particular emphasis will be placed on evaluating the possibility of delayed failure in tuff and what controls it.

(3) A subsidiary goal is to measure the time variation of water permeability in tuff and how it is affected by slow deformation of the rock matrix. A detailed background discussion of time-dependent microclastic deformation of repository-candidate rocks is given by Blacic and will only be summarized here. The basic problem is that present design and performance calculations for a repository assume that thermomechanical and transport properties for a particular rock are constant over time. These properties are determined by standard, short-time engineering tests that do not examine a possible time dependence. Repository designs are unusual in that they must stand the test of very long term performance. Analysis that does not take into account time effects can lead to erroneous answers. For example, thermal conductivity might decrease over time as a result of slow, progressive microcracking of rock in the near field. Thermal calculations that do not incorporate this effect will underestimate near-field temperatures. Similarly, permeability of tuff is relatively low because of the apparently low connectivity of porosity. This could change substantially over time during slow deformation. Nuclide transport models that do not include this change would overestimate transit times.

A reliable evaluation of the magnitudes of these time phenomena is difficult, principally because of a lack of data. We describe below a two-part experimental approach to obtaining the required data. First we will make a preliminary evaluation of the existence of the time-dependent phenomena in tuff and a rough estimate of their magnitudes. Then, if, based on these results, analysis indicates that more precise measurements are required for design and performance calculations, we will proceed with these measurements.

The first part of our approach is almost complete. An exploratory test was performed in which samples of a variety of tuff types from the southwest part of the Nevada Test Site (NTS) were exposed to a range of temperatures, confining pressures, and water pore pressures for times of 2 to 6 months. A set of thermomechanical properties was measured at ambient conditions before and after this exposure and compared. Preliminary results indicate that
substantial and statistically significant changes occurred in tensile strength, compressive strength, and mineralogic content in some tuff types. Comparisons of data from before and after exposure of thermal conductivity and permeability are currently in progress. Details of this test are given in Ref. 2. Guided by these results, we are in the process of extending the measurements with long-term deformation (creep) tests that should indicate the magnitude of strength change more precisely. A prototype apparatus was designed and constructed for the creep measurements, and a few check-out tests have been performed. Details of the apparatus and the first few tests are given below.

II. DEVELOPMENT OF EXPERIMENTAL EQUIPMENT

Because laboratory data on time dependence of mechanical and transport properties of rock will have to be extrapolated orders of magnitude in time even to encompass the operational time period, rather precise measurements of these properties are required. Physical variables such as temperature and pressure must be controlled, and response measurements must be made over time periods of months. This requires specialized testing equipment not widely available. Consequently, we designed and built a prototype apparatus that incorporates what we believe to be current state-of-the-art capabilities in rock mechanics testing equipment. Considerable effort and time was spent developing the apparatus in order to reduce programmatic risk. If more detailed creep measurements will be required, as now appears to be the case based on the preliminary testing described above, then availability of suitable equipment will reduce time delays in the program. Even so, tests of such long duration over a range of physical conditions and rock characteristics would prevent us from obtaining this information on an accelerated or crash basis.

The apparatus is best described in terms of distinct subsystems. These are (1) differential stress, (2) confining pressure, (3) pore pressure and permeability, and (4) computer control and digital data acquisition systems.

A. Differential Stress Subsystem

The purpose of the differential stress subsystem is to rapidly attain the desired axial differential stress on the test sample and to precisely maintain that value for the duration of the test. The differential stress is applied only after steady values of confining pressure, pore pressure, and temperature
have been achieved. Because of the long duration of the planned tests, it is desirable to have an automated control system so that an operator need not always be present. The computer control and digital data acquisition subsystem is described below. We will describe only the mechanical portion of the subsystem here.

Figure 1 shows the basic components of the subsystem. The jacketed sample is contained in an externally heated pressure vessel and pressurized with silicone oil. Details of the pressure vessel design are discussed below with the description of the confining pressure subsystem. The pressure vessel is mounted within, and axial force is applied by a standard four-post hydraulic load frame. Details of the load frame are not significant; the one we use in the prototype system is capable of generating 500,000 lb of force. The important aspects of the system are how the hydraulic ram pressure is generated and automatically adjusted to maintain a constant differential stress on the test sample (shown in Fig. 2).

The ram is manually advanced or retracted for initial contact or unloading of the sample, respectively, by means of an air/oil pump (P1) actuated through regulator R1 by low-pressure house air. The direction of ram movement is determined by electric solenoid valves SV5 and SV6, and pressures are monitored on gauges G3 and G4. At the beginning of a test, once contact of the sample and the loading piston is made, the manual system is no longer used. High pressure is then applied suddenly to the ram by opening solenoid valve SV1. Before opening SV1, an accumulator (AC) is precharged from a high-pressure gas bottle (B1) through regulator R2 and gauge G2. The accumulator is a bladder type that separates the charging gas from the hydraulic oil. Because there is essentially no friction in this type of accumulator, a differential stress within about 5% of the desired value can be obtained by accurately precharging. This value is rapidly attained at sample strain rates of about $10^{-3} \text{ s}^{-1}$. About 2 seconds after SV1 is opened, automatic trimming of the accumulator gas pressure begins with feedback from an internal force transducer. The internal force gauge consists of a strain gauge with four active arms that is bridge-mounted on a steel end piece between the sample and the loading piston. Differential stress on the sample is calculated from the force gauge signal and the current cross sectional area of the sample that is determined from strain gauges on the sample and the
Fig. 1.
Photograph of the apparatus.
The differential stress subsystem. The following symbols are used: $V = \text{manual valve}$, $SV = \text{electric solenoid valve}$, $R = \text{regulator}$, $B = \text{bottle}$, $AC = \text{accumulator}$, $RV = \text{relief valve}$, $P = \text{pump}$, $CKV = \text{check valve}$, and $G = \text{gauge}$.

starting sample dimensions. The current stress is compared to the desired value and any difference controls the action of the control loop.

The actuator part of the control loop consists of solenoid valves $SV_2$, $SV_3$, $SV_4$; regulator $R_3$; manual throttling valve $V_3$; and bottle $B_2$. Valves $SV_2$, $SV_3$, and $SV_4$ are normally closed. If, for example, an increment of pressure is required to increase the stress towards the desired value, $SV_2$ is momentarily opened and then closed, trapping a small volume of gas in $B_2$ at a pressure (determined by $R_3$) higher than that in the accumulator. This increment of pressure is then injected by opening and closing $SV_3$. To lower the applied stress, the reverse process takes place, venting an increment from the accumulator through $SV_4$ and $V_3$. This trimming process is under complete computer control, the details of which are described in Sec. II.D. Typically, the system is capable of maintaining a differential stress on the sample with about 0.25% precision.

B. Confining Pressure Subsystem

A schematic of the confining pressure subsystem is shown in Fig. 3. Various pressure vessels may be used in this subsystem. The one we most commonly use accepts an NX size sample (5.4 cm in diameter) and is capable of 70
Fig. 3.
Schematic of the confining pressure subsystem. The following symbols are used: B = bottle, V = manual valve, R = regulator, G = gauge, and X = pressure transducer.

MPa (10,000 psi) pressure and 250°C using external band heaters. Electrical feed-throughs for up to 40 signals are provided for internal instrumentation of the sample. These normally consist of three pairs of axial and circumferential strain gauges mounted at 120° intervals around the sample, three axial LVDT displacement transducers mounted on steel buttons that are epoxied to the sample and rotated 60° from the strain gauges, three or four thermocouples at different heights along the sample, and the internal force transducer. The rock sample is isolated from the confining pressure medium (DOW 710 silicone oil) by an RTV silastic or other barrier painted on the sample or by a viton rubber jacket. In the latter case, strain gauges or LVDT buttons cannot be mounted directly on the sample surface. Sample pore pressure access is through the sample pedestal and a hollow upper end cap.

Primary confining pressure is generated by a 15:1 piston intensifier actuated by compressed gas through regulator R1. Once pressure is attained, this system is valved off, and fine control is obtained by means of an automated, screw-driven intensifier. Pressure is measured by one of several strain gauge pressure cells, depending on pressure range and Bourdon tube gauges. The motor-driven screw intensifier is computer controlled using a feedback signal from one of the pressure transducers. By adjusting motor speed and transducer sensitivity, pressure can be controlled to within a few
pounds per square inch over long time periods. At large sample strains, screw piston displacement can be measured to obtain sample volume strain using the method described by Wawersik.

C. Pore Pressure and Permeability Subsystem

A schematic of the pore pressure subsystem is shown in Fig. 4. Initial water pore pressure is obtained by means of an air/water pump after first evacuating the sample and pressure lines. Once pressure is attained, the pump is valved off, and fine control is obtained by actuating a bladder-type accumulator with compressed gas. This system is sufficient for long-term control of pore pressure within about 2%. For finer control required for permeability determinations, one or the other of two subsystems is used. At permeabilities above about 1 mdarcy, the sample is isolated with a Constametric-brand liquid chromatograph pump. This pump is capable of generating constant flow rates of 0.1 to 10 mL/min at line pressures up to 40 MPa. At a given flow rate, the pressure drop across the sample is measured by a differential pressure transducer (X2) to an accuracy of 0.25%. The permeability is then calculated from Darcy's equation for viscous flow through a porous medium as

$$k = \frac{\mu \nu L}{AP},$$

where $k$ is permeability, $\mu$ is the viscosity of water, $\nu$ is the volume flow rate per unit time per unit cross sectional area, and $AP$ is the pressure drop over a sample of length $L$.

At very low permeabilities, flow rates are too low for any reasonable pressure drop and so a relaxation method is used. The method is described by Brace et al. in which a pressure step is applied to the sample, and the exponential decay of pressure is recorded from the differential pressure gauge. Permeability is then related to the pressure decay constant, $\alpha$, by the equation

$$k = \frac{\alpha \delta \mu LV_2 V_1}{A(V_2 + V_1)},$$

where $k$ is permeability, $\alpha$ is the decay constant, $\delta$ is the isothermal compressibility of water, $\mu$ is the dynamic viscosity of water, $L$ is the sample length, $A$ is the sample cross sectional area, and $V_1$ and $V_2$ are the volumes of pore fluid reservoirs at the two ends of the sample. The pressure step can be applied in a number of ways. For automated measurements in which the mean
The pore pressure and permeability system. The following symbols are used: B = bottle, R = regulator, V = manual valve, SV = electric solenoid valve, VP = vacuum pump, P = pump, G = gauge, S = test sample, F = filter, M = motor, X = pressure transducer, and DX = differential pressure transducer.

pore pressure is constant, a motorized, double-acting piston pump may be used. The pressure difference is produced by advancing the piston in one direction or the other and then opening solenoid valves SV2 and SV4 to suddenly apply the pressure step to the sample. A simpler method of applying a pressure step is to suddenly turn the stem of valve V5.

D. Computer Control and Digital Data Acquisition Subsystem

A block diagram of the computer control and digital data acquisition system is shown in Fig. 5. The principal component of this system is a Digital Equipment Corporation LSI-11 microcomputer and flexible disk system. The computer performs the following functions.

1) Control Differential Stress. The computer gathers force, confining pressure, and circumferential strain values from the internal load cell, pressure transducer, and sample-mounted strain gauges, respectively. This information is used to calculate the current value of the true differential stress on the test sample. The computer will then actuate a valve sequence to adjust the force so as to maintain the differential stress at the desired level, as described above.
(2) Control Confining Pressure. Feedback from the confining pressure transducer is used in a process to actuate a motor-driven pressure intensifier and maintain a desired level of pressure. Priority of confining pressure control is lower than that of differential stress. In practice, interaction between the two systems is minimal enough to avoid instability in the overall control process.

(3) Acquire Data. The computer is continually acquiring raw data from the various transducers and storing this information on a floppy disk for postprocessing. There is a real-time display of selected test variables on a computer graphics terminal to help monitor the status of the test. The rate at which data is collected and the number of data channels used is set by the operator.

A 32-channel signal conditioner module can provide excitation, balance, and amplification for transducers, strain gauges, etc. The module, which incorporates the ability to shunt-calibrate strain gauges, has a digital voltmeter to aid setting gain and offset balancing.

A 16-channel valve-switch module and a 16-bit input/output card located in the LSI-11 computer provide the capability of opening or closing any of the
solenoid valves in the control system. The switch module also contains the power supply necessary to operate the solenoids.

The unit that controls the motor speed also controls the direction and speed of the confining pressure intensifier. The unit can be operated locally or automatically by the computer. Two channels of the valve-switch module are used by the computer to control the direction and time that the intensifier is actuated.

An analog temperature controller is used to control band heaters mounted on the pressure vessel. Feedback is obtained from platinum RTD temperature sensors, and their values are acquired by the computer for postprocessing.

Software for the system is stored on a separate floppy disk. A flow diagram of the control and data acquisition program is shown in Fig. 6, and a Fortran listing is provided in the Appendix.

E. Results of Preliminary Experiments

A few room-temperature, uniaxial creep tests were performed primarily to check out the operation of the differential stress system and the computer control and data acquisition programs. We selected samples of Grouse Canyon welded tuff from G-tunnel that were reputed to be from the same block tested by Olsson and Jones of Sandia National Laboratories (SNL). According to the SNL data, this tuff should have a uniaxial compressive strength of 180 MPa. Consequently, the first test was planned for a differential stress of 100 MPa and ambient confining pressure and temperature. Under these conditions, creep deformation lasting at least 6 weeks was anticipated. However, the sample failed catastrophically as soon as the stress was applied.

Before a second test on this tuff, some uniaxial, constant-displacement-rate tests were performed to verify the expected failure strength. At a strain rate of about $10^{-4}$ s$^{-1}$, samples with diameters of 1.25 and 2.5 cm failed at stresses ranging from 65 to 180 MPa (Fig. 7). The weakest sample contained a fragment of pumice that apparently also acted as a weak zone in the 5.4-cm-diam creep test sample. Consequently, the sample selected for the second creep test did not contain any pumice fragments visible at the surface.

In the second test, a differential stress of 75 MPa was applied and held for 5 days (Fig. 8). The load control system worked well. Some minor problems in the data acquisition systems were identified and subsequently corrected. As can be seen in Fig. 8, there was a very low rate of creep in the axial direction. In the last few days of the test, the average strain rate
Initialize all valves and turn off confining pressure motor. Read from disk:
Data acquisition parameters
Test parameters
Initial transducer offset values.
Allow operator to change data/test parameters.
Calculate and display required accumulator pressure.
Start data acquisition at a rate of 100 s/scan.

Is this a restart?
- No
- Yes

Collect initial transducer offset values.
Allow operator to:
Set accumulator pressure
Set confining pressure
Set pore pressure
Set temperature
Position ram.
Collect final transducer offset values.

Write to Disk:
Data acquisition parameters
Test parameters
Transducer offset values.

Is this a restart?
- No
- Yes

Set-up graphics screen.
Set data rate = 0.5 s/scan.
Collect data for 1 s.
Pop valve (load ram).
Collect data for 5 s.
Set data rate = 1 s/scan.

Collect data and store on disk.
Control load to maintain a constant differential stress.
Control confining pressure.

Any operator requests?
- Yes
- No

Allow operator to:
Graph or no graph of data
Change data rate
Clear graph screen
Change data disk
Continue data acquisition
Pop valve (load ram)
Display amount of data disk used

Stop test?
- Yes
- No

Stop

Fig. 6.
Flow diagram of test control and data acquisition program.
Unaxial stress-strain curves for Grouse Canyon welded tuff at 25°C. A and B are 0.5-in.-diam cylindrical samples and C and D are 1-in.-diam samples.

was $\sim 7 \times 10^{-10} \text{ s}^{-1}$. Most of the axial strain was recovered immediately upon unloading; the remaining amount was recovered within a few days. This observation leads us to believe that the creep strain was due to slow closing of pre-existing cracks preferentially oriented perpendicular to the principal stress direction (also, no new cracks were produced). Therefore, because we believed the sample was not damaged, we decided to reload it at a higher stress level. The sample was loaded to 100 MPa, which again should have been well below the short-time failure strength if no weak zones were present.

However, this sample again failed immediately as the higher stress was applied. Figure 9 shows the first few seconds of the test. Comparison of Fig. 8 and 9 illustrates the wide dynamic range of our digital data acquisition system and the short risetime of the stress application. The comparison also shows that at 75 MPa, the strains are the same as in the earlier loading to 75 MPa, and only a small increase in strain from this level
resulted in unstable conditions and failure. In the case of the circumferential strain, which is the most sensitive indicator of crack formation in a uniaxial test, the strain increased from a stable level of $0.8 \times 10^{-3}$ to an unstable level of $1.1 \times 10^{-3}$ within 1 second after the 75-MPa level was exceeded. The volume strain (axial minus twice the circumferential strain) at the end of the first loading was approximately $1.9 \times 10^{-3}$. The volume strain at the beginning of failure in the second loading was about $2.4 \times 10^{-3}$ and rapidly decreasing as a result of dilatant crack formation (Fig. 9). In retrospect, it appears that if the average strain rate in the first loading to 75 MPa had been maintained for an additional 120 h, the failure-onset strain would have been reached. Thus, onset of creep failure of this tuff under uniaxial conditions in a dry atmosphere at ambient temperature and pressure appears to occur over a narrow strain range with little indication of the nearness to the failure stress. Finally, examination of the sample did not reveal any pumice fragment or other weak zone in the interior that could have been responsible for the low failure stress ($\sim 90$ MPa compared to 180 MPa from the SNL data and the maximum values in Fig. 7).

Fig. 8. Axial and circumferential creep strain vs time for Grouse Canyon welded tuff. Uniaxial stress is 75 MPa and temperature is 25°C.
A third test was performed on a granite sample to test the confining pressure and temperature control systems. Granite was used instead of tuff because of the uncertainty in predicting the failure stress in tuff, as described above. We wanted to be certain that the sample would not fail during the check-out procedures. This test lasted approximately 3 weeks at 100°C, 10 MPa confining pressure, and 100 MPa differential stress. During this time, we tuned the temperature controller and attempted to control confining pressure. However, it was noted that small temperature variations (±1 to 2°C) would cause unacceptable pressure variations (±10%) because of thermal expansion of the silicon confining pressure oil. As a result, we added the active confining pressure control that is described in Sec. II.B. With this device we are able to hold confining pressure to within ±0.5% of the desired value regardless of temperature or other variations.

Following these preliminary experiments and the resultant tuning of the apparatus, we began testing potential target horizon rock under conditions expected in the near field of a repository. The first of these tests has been completed, and although the test was not completely successful because of a jacket leak, some interesting results were obtained.
The test specimen was Bullfrog tuff from the 2483-ft level of hole USW-G1. The sample was 11.1 cm long and 5.4 cm in diameter. The test conditions were 100°C, 50 MPa differential stress, 20 MPa confining pressure, and 5 MPa water pore pressure. The effective confining pressure of 15 MPa was attained initially, but because a slow jacket leak developed early in the test, the effective pressure slowly dropped over the duration of the test to about 50% of its initial value. Because of the slow decrease in effective pressure, axial strain actually decreased slowly from the initial loading value while circumferential strain increased slowly (Fig. 10). Over the last 20 h of the test, the average circumferential strain rate was $1.2 \times 10^{-9}$ s$^{-1}$, which increased very slightly in the last 2 h. In retrospect, this almost imperceptible increase in strain rate was the onset of a type of Tertiary creep that only developed strongly in the last 200 s of the test. During this time, strain accelerated rapidly and the sample failed at a total test time of about 69 h. Figure 10 shows the average of two axial strain gauges away from the ultimate zone of failure (which apparently was also near the jacket leak because the gauge that failed showed a much greater decrease in strain over time). The increase in circumferential and volume strain at about 23 h corresponds to the time when the pore pressure accumulator was valved off, resulting in a transient increase in the rate of effective pressure decrease.

This type of sudden Tertiary creep failure is similar to that which was observed in uniaxial, room-temperature creep of Grouse Canyon welded tuff. In each case there has been little indication of the nearness to failure, and so far there is no evidence of the classical, exponentially increasing Tertiary creep that has been observed in granite and other materials. This may be a reflection of the structural inhomogeneity of our tuff samples in which failure may start in a very local region with no general increase in microfracturing activity that might be noticeable at strain gauges outside of the ultimate failure zone.

Although this test alone does not establish a new issue of concern, it does suggest a type of potential failure that should be evaluated. In the near field of a repository, in a material of very low permeability and relatively high water content such as a zeolitized tuff, thermal expansion of pore water or mineral dehydration water could lead to a local increase in pore pressure if the water could not leak away rapidly enough. This could lead to local rock failure initiation because of the reduction in effective pressure.
Axial, circumferential, and volume strain vs time for Bullfrog tuff experiment. Compression is positive, but the sign of the circumferential strain is reversed in the plot. Tests were performed at 100°C, 20 MPa confining pressure, 5 to 12.5 MPa pore pressure, and 50 MPa differential stress.

This would occur while at least some of the strain monitors were indicating a decreasing rate of strain because of the overall decompression associated with a falling effective pressure (analagous to the axial strain record in Fig. 10).

III. CONCLUSIONS AND STATUS

We have designed and perfected an apparatus capable of performing precise, long-term creep deformation tests on rock samples. The apparatus is capable of performing tests lasting several months at maximum differential stress of 970 MPa, confining pressure of 1200 MPa, pore pressure of 40 MPa, and temperature of 250°C. A system for measuring sample permeability under the above conditions has been designed but remains to be completed and tested.

Grouse Canyon welded tuff has a uniaxial compressive strength inhomogeneity of at least a factor of three on a scale of a few centimeters. Note that we are considering only "intact" samples not containing larger scale inhomogeneities such as joints, fractures, or lithophysae. If target horizon tuffs show a similar inhomogeneity, then many more tests will have to be performed.
than originally anticipated to get good statistics on mechanical properties, or larger samples containing a representative number of inhomogeneities will have to be tested. Probably both approaches will be required.

The onset of microclastic creep failure in Grouse Canyon welded tuff at ambient conditions and Bullfrog tuff at 100°C, 15 to 7.5 MPa effective pressure, and 50 MPa differential stress appears to occur over a narrow strain range with little Tertiary creep warning. In one test of Bullfrog tuff under conditions of slowly decreasing effective pressure, a failure occurred; during the failure, some strain gauges actually indicated a decreasing rate of strain, which was caused by the overall decompression of the sample.

REFERENCES


APPENDIX

FORTRAN LISTING OF CONTROL AND DATA ACQUISITION PROGRAM

Los Alamos Identification No. LP-1465

C
C PAGE 1 OF 'MAIN'
C
C MAIN 3-18-82 1030 HRS
C MODULES NEEDED WITH THIS PROGRAM:
C MAIN.OBJ
C SUBATN.OBJ
C SUBPTZ.OBJ
C LSILIB.OBJ
C TIMDAT.OBJ
C TKDATA.OBJ
C TKLIB.OBJ
C
C WHEN COMPILING DATACO CHOOSE THE FORTRAN SWITCHES:
C /OPT:SPD /NOSWAP /REC:1000 OR MORE
C LET COMPILER KNOW 'STORE' IS A ROUTINE - NOT A VARIABLE
C
C COMMON/ACQDAT/FULSCL,FRATE,INTCNT,ISTCHN,JRATE,NCHAN,PRESET,
C * RTIME,STIME
C COMMON/ARRAYS/CONV(32),DSCRPT(15,32),DSFLAG(15),
C * FDSCR(15,2),IRTNT(6),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2)
C COMMON/CHANH/LOAD,KCONF,KFORE,KDIFF,KRAD1,KRAD2,KRAD3
C COMMON/PROCES/IBUF(1000),NBUN,ISIZ,NCOUNT,
C * IFAST,ICHAN1,ICHAN2,J,NORH,IFYO32,IX2,LEVENT,IRECD
C COMMON/SCREEN/KDIP,LOAD,IGTEMP,ONST1,ONST2,ODTMP
C COMMON/TEST/ACPSI,ARENW,AXLCONF,CIRREF,CONFSTR,CPERNT,DIFDSR,
C * GAUXL,GAURAD,ICORF,ICIRF,ICONF,ICONFF,ICIFFI,
C * ICFLG,IFLAG,LOADF,LOADI,IPORF,IPREF,PERCT,PI,PISARE,
C * RANCONF,LOAD,RSFLAG,SPDIA,SPDAR,
C * SNAIL,SNCONF,SNIFF,SNLOAD,SNFORE,SNRADL,STRSS1
C
C DATA IX,IY0/1295,1295,32*1560/
C DATA DSFLAG/HFLAG,CHN,NNEL,12*4H /
C DATA KEVENT,LEVENT/O+0/
C DATA IBUF/1000*0/
C DATA NOGRPH/1/
C DATA JUNK/19190/
C DATA JUNK/JUNKR/19190,19191,19192,19193,19194,19195/
C
C ICHAN1=3
C ICHAN2=4
C PI=3.141593
C PISARE=3.54656

CHAME
C ! WE DON'T KNOW WHY, BUT DO THIS FIRST
C CALL ASSIGN (ICHAN2,'DY1:EXPTD,ATA',13,'NEW',2)
C REWIND ICHAN2
C TYPE 211
C
211 FORMAT(/' DO YOU WISH TO CREATE A RESTART FILE (Y OR N) ?','*')
C ACCEPT 2300,ANSW
C IF(ANSW.EQ. 'N') GO TO 1
C CALL CHANGE
C CALL ASSIGN (ICHAN1,'DYO:RESTRT,DATA',14)
C CALL WRSTR(ICHAN1)
C CALL CLOSE(ICHAN1)
C
APPENDIX (cont)

1 CONTINUE
C INITIALIZE ALL VALVES AND TURN OFF CONF.PRESS MOTOR
CALL SYSINT(IFLAG)
C READ FROM DISC: DATA ACQUISITION AND TEST PARAMETERS
CALL ASSIGN (ICHAN1,'DYOJRESTRT.DAT',14)
REWIND ICHAN1
CALL RDRSTR(ICHAN1)
CALL CLOSE (ICHAN1)
C ALLOW OPERATOR TO CHANGE DATA ACQ./TEST PARAMETERS
CALL CHANGE
C CALCULATE AND DISPLAY REQUIRED ACCUMULATOR PRESSURE
CALL ACPRES
C INITIAL START POINT FOR DATA ACQUISITION SYSTEM
  TYPE 250
  ACCEPT 2300, IDUMMY
C OPEN DATA FILE, SET FIRST DATA FILE NAME = 'DY1:EXPTD.ATA'
C CALL ASSIGN (ICHAN2,'DY1:EXPTD.ATA',13,'NEW',',2)
C REWIND ICHAN2
C WRITE HEADER DATA ON DISC
CALL WTHEAIKICHAN2)
C INITIALIZE DATA ACQUISITION BUFFER POINTERS AND MISC.
CALL INTACQ
  TYPE 2400
  ACCEPT 2300, IDUMMY
C SET INITIAL START TIME FOR THIS TEST
  STIME=SECNDS(0.)
  CALL TIME(TIMSTR)
C WRITE RELATIVE TIME (RTIME) AND START TIME FOR THIS DISC
C THIS IS WRITTEN ONLY ONCE PER DATA FILE
WRITE (ICHAN2) RTIME,TIMSTR
C START THE DATA ACQ. SYSTEM
CALL TKDATA(IRBUF,ISIZ,ISTCHN,NCHAN,NBUF)
CALL SETRJRATE,1,PRESET,ICMF,INTCNT,STORE)
C SET DATA RATE = #3
  JUNK=JUNKR(3)
  WRITE (ICHAN2) (JUNKKI,DUMMY=1,NCHAN+1)
  NCOUNT=INTCNT(3)
C IS THIS A RESTART ?
10 TYPE 2000
  ACCEPT 2300,RSFLAG
  IF(RSFLAG.EQ.'N') GO TO 15
  IF(RSFLAG.EQ.'Y') GO TO 50
  GO TO 10
APPENDIX (cont)

C PAGE 3 OF 'MAIN'

C IF NOT A RESTART, THEN:
C COLLECT INITIAL XIUCER OFFSETS
15 CALL INTOFF
C LET OPERATOR SET ACCUMULATOR, CONFINING AND
C F finalists, TEMPERATURE, AND POSITION RAM
20 TYPE 2100
   ACCEPT 2300, ANSW
   IF(ANSW.EQ., 'GO') GO TO 20
C COLLECT FINAL OFFSETS
   CALL FINOFF
C COLLECT CIRCUMFERENCE REFERENCE VALUE
   CALL CIRRFF
C WRITE RESTART FILE
   CALL ASSIGN (ICHAN1, 'DYO:RESTRT.DAT', 14)
   CALL WRSTR(ICHAN1)
   CALL CLOSE (ICHAN1)
C SET UP GRAPHICS
   NOGRPH*0
   IX(2)*1295
C SET DATA RATE TO FASTEST
   IFAST=1
   WRITE (ICHAN2) (JUNK, IDUMMY=1, NCHAN)
C COLLECT DATA FOR 2 SECONDS
   T1=SECONDS(0)
   T2=SECONDS(T1)
   IF(T2.LT.2) GO TO 30
C POP VALVE CONTINUE FAST RATE FOR 5 SECONDS
   T2=SECONDS(T1)
   IF(T2.LT.7) GO TO 40
C SET DATA RATE  *  01
   IFAST=0
   JUNK=JUNKR(1)
   WRITE (ICHAN2) (JUNK, IDUMMY=1, NCHAN+1)
   NCOUNT=IRTCNT(1)
   GO TO 190
C IF THIS IS A RESTART, THEN:
50 CONTINUE
C WRITE RESTART FILE
   CALL ASSIGN (ICHAN1, 'DYO:RESTRT.DAT', 14)
   CALL WRSTR(ICHAN1)
   CALL CLOSE (ICHAN1)
   CALL ACPRES
60 TYPE 2100
   ACCEPT 2300, ANSW
   IF(ANSW.EQ., 'GO') GO TO 60
C SET UP GRAPHICS
   NOGRPH*0
   IX(2)*1295
C POP VALVE
   CALL POP
   '0 TO 190
APPENDIX (cont)

C THIS IS THE RESTART POINT WHEN THE CHANGE FLOPPY OPTION IS DONE
150 RTIME=SECND$$(STIME)
C WRITE RELATIVE TIME (RTIME) AND START TIME FOR THIS DISC
C THIS IS WRITTEN ONLY ONCE PER DATA FILE
WRITE (ICHAN2) RTIME,TIMSTR
C TAKE DATA AND PROCESS ANY COMMANDS FROM KEYBOARD
180 CALL TKDATA(IBUF.ISIZ,ISTCHAN,NCHAN,NBUF)
CALL SETR(JR,1,PRESET,ICMF,INTCNT,STORE)
C PROCESS OPERATOR REQUEST (ALLOW DATA ACQ. TO INTERRUPT)
190 CALL CURATE
MENFLG=0
C TYPE C<CR> TO CHANGE OPTIONS
100 TYPE 200
200 FORMAT (' TYPE C<CR> FOR OPTION. ','$)
ACCEPT 210,IDUMMY
210 FORMAT (A2)
C TEST IF ERROR FROM COMPLETION ROUTINE 'STORE'
IF (ICMF.LT.0) STOP ' DATA OVERRUN, USE SLOWER RATE!!'
IF (IDUMMY.NE.'C') GO TO 100
215 TYPE 220
220 FORMAT (' OPTIONS: ','$')
% 1) GRAPH$/;
% 2) NO GRAPH$/;
% 3) CHANGE DATA RATE$/;
% 4) WRITE EVENT FLAG$/;
% 5) CLEAR SCREEN$/;
% 6) NEW DATA FILE$/;
% 7) STOP TEST$/;
% 8) NO OPERATION$/;
% 9) POP VALUE$/;
% 10) BLOCK COUNT$/;
% 11) CHANGE TEST PARAMETERS$/)
TYPE 230
230 FORMAT (' OPTION? ','$
ACCEPT 240,IDUMMY
240 FORMAT (I2)
IF(IDUMMY.LT.1.OR.IDUMMY.GT.11) GO TO 215
GO TO (500,510,520,530,540,550,560,570,580,590,600) IDUMMY
C C GRAPH ON
500 CONTINUE
NOGRPH=0
IX(2)=1295
GO TO 999
C NO GRAPH
510 CONTINUE
NOGRPH=1
GO TO 999
APPENDIX (cont)

C
C             PAGE 5 OF 'MAIN'
C
C CHANGE DATA RATES
C
C CONTINUE
  C  TYPE 527
  C  TYPE 523, (SCANS(6))
  C  TYPE 525
  DO 521 IDUMMY=1,5
     C  TYPE 524, (IDUMMY, SCANS(IDUMMY))
  C  TYPE 525
  C CONTINUE
  C  TYPE 528
  ACCEPT 529, IRATE
  IF (IRATE LE 0) GO TO 522
  IF (IRATE GT 5) GO TO 520
  IFAST=0
  JUNKK=JUNKR(IRATE)
  WRITE (ICHAN2) (JUNKK, IDUMMY=1, NCHAN+1)
  NCOUNT=IRTCNT(IRATE)
  GO TO 999
C  CONTINUE
  IF (IFAST EQ 1) GO TO 999
  IFAST=1
  WRITE (ICHAN2) (JUNK, IDUMMY=1, NCHAN+1)
  C  FORMAT
  523 FORMAT(' RATE * 0 = FAST DATA RATE (SECS/SCAN) ', F10.2, *)
  524 FORMAT(' RATE * ', I2, ' = (SECS/SCAN) ', F10.2, $)
  525 FORMAT(' ')
  526 FORMAT(' ')
  527 FORMAT('/ / DATA ACQUISITION RATES !')
  528 FORMAT('/ DESIRED RATE *!*, *')
  529 FORMAT (I2)
  GO TO 999
C  WRITE AN EVENT FLAG TO CHANNEL NCHAN+1
  530 CONTINUE
  KEVENT=KEVENT+1
  LEVENT=KEVENT
  TYPE 531, KEVENT
  531 FORMAT (' EVENT FLAG NUMBER *I3, * WRITTEN')
  GO TO 999
C  CLEAR
  THE SCREEN
  540 CONTINUE
  CALL TKPACK('34)
  CALL TKPACK('33)
  CALL TKPACK('14)
  CALL TKPACK('30)
  GO TO 999
APPENDIX (cont)

C PAGE 6 OF 'MAIN'

C WRITE NEW DATA FILE

C

550 CONTINUE

C TURN DATA ACQ. OFF
CALL SETR(-1,0)

C DO DUMMY DELAY FOR COMPLETION OF PRESENT SCAN
DO 551 IDUMMY=1,500
DUMMY=SQRT(FLOAT(IDUMMY))*SQRT(FLOAT(IDUMMY))

551 CONTINUE

C SET RT-11 CLOCK = BATT CLOCK (GET CORRECT DATE ,ONLY)
CALL TIMDAT

C CLOSE EXISTING FILE
CALL CLOSE(ICHAN2)
TYPE 55?

55? FORMAT(//' ENTER NEW DATA FILE NAME:'),*)

C OPEN NEW DATA FILE
CALL ASSIGN(ICHAN2,'-',1)
REWORK ICHAN2

C WRITE HEADER DATA ON DISC
CALL WTHEAD(ICHAN2)

C RE-INITIALIZE SOME VARIABLES AND CLEAR SCREEN
IFAST=0
NCOUNT=IRTCDT(1)
ICMF=0
NBUF=1000/NCHAN
IRECR=0
J=1
IX(2)=1295
CALL TKPACK('34')
CALL TKPACK('33')
CALL TKPACK('30')

C GO TO RESTART THE DATA ACQUISITION
TYPE 553

553 FORMAT(//' TYPE <RET> TO BEGIN TAKING DATA ',*)
ACCEPT 554,DUMMY

554 FORMAT(A4)
MENFLG=1
GO TO 999

C NO OPERATION

570 CONTINUE
GO TO 999

C POP VALVE TO LOAD RAM IN CREEP TEST AND BEGIN LOAD CONTROL

580 CONTINUE
CALL POP
GO TO 999

C BLOCK COUNT

590 CONTINUE
IBLOCK=IRECRD*(NCHAN+2.25)/256+2
TYPE 591,IBLOCK

591 FORMAT(//' NUMBER OF BLOCKS USED SO FAR IS: ',I5)
GO TO 999
APPENDIX (cont)

C PAGE 7 OF 'MAIN'

C

C STOP

560 CONTINUE
C TURN OFF DATA ACQ.
CALL SETR(-2,1)
C CLOSE EXISTING FILE
CALL CLOSE(ICHAN2)
CALL SYSINT
TYPE 561
561 FORMAT ('//,'GOOD-BY FOLKS! ')
STOP 'PROGRAM FINISHED'
C CHANGE TEST PARAMETERS
C SET DATA RATE TO SLOWEST
600 IFAST=0
JUNKK=JUNKR(5)
WRITE (ICHAN2) (JUNKK,IDUMMY=1,NCHAN+1)
NCOUNT=IRTCNT(5)
C CLEAR SCREEN AND TURN OFF GRAPHICS
NOORPH=1
   CALL TKPACK('34)
   CALL TKPACK('33)
   CALL TKPACK('14)
   CALL TKPACK('30)
GO TO 605
603 TYPE 735
   TYPE 700,DIFDSR
   CALL SAME(DIFDSR)
   VAR=100,*PERCNT
   TYPE 705,VAR
   CALL SAME(VAR)
   PERCNT=VAR/100.
   TYPE 710,CONDSDR
   CALL SAME(CONDSR)
   VAR=100,*CPERNT
   TYPE 715,VAR
   CALL SAME(VAR)
   CPERNT=VAR/100.
C PRINT OUT TEST PARAMETERS
605 TYPE 735
   TYPE 700,DIFDSR
   TYPE 730
   VAR=100,*PERCNT
   TYPE 705,VAR
   TYPE 730
   TYPE 710,CONDSDR
   TYPE 730
   VAR=CPERNT*100.
   TYPE 715,VAR
   TYPE 730
APPENDIX (cont)

C
C PAGE 8 OF 'MAIN'

C NOW SEE IF ALL DATA CORRECT

TYPE 720
ACCEPT 725,ANS
IF(ANS.EQ.'N') GO TO 603

C REWRITE RESTART FILE
CALL ASSIGN (ICHAN1,'DYORRESTRT.DAT',14)
CALL WRSTR(ICHAN1)
CALL CLOSE (ICHAN1)
GO TO 999

700 FORMAT(' DESIRED DIFFERENTIAL STRESS (PSI) ',12X,F10.3,$)
705 FORMAT(' DESIRED STRESS CONTROL PERCENTAGE ',12X,F10.3,$)
710 FORMAT(' DESIRED CONF. PRESSURE (PSI) ',13X,F10.3,$)
715 FORMAT(' DESIRED CONF. PRESS. CONTROL PERCENTAGE ',6X,F10.3,$)
720 FORMAT(/' IS THIS CORRECT ? (Y OR N) ')
725 FORMAT(A4)
730 FORMAT(' ')
735 FORMAT(/'/ ')
C ********************************************
999 CONTINUE
IF(NESTAB.EQ.1) GO TO 150
GO TO 190

C
?000 FORMAT(/' IS THIS A RESTART (Y OR N) ? ')
2100 FORMAT(/' TYPE <GO> TO CONTINUE, AFTER:' ,'/
# ACUMULATOR,CONFING,FORE PRESSURE,TEMPERATURE SET','/
# RAM IS IN POSITION'
2300 FORMAT(A4)
2400 FORMAT(/' TYPE <RET> TO BEGIN TAKING DATA ')
250 FORMAT(/' PUT IN THE DATA DISC,' DATA WILL BE WRITTEN TO
$ FILE DYOR!EXPTDA.DAT',' TYPE <RET> WHEN READY ! ')
END

26
APPENDIX (cont)

C------------------------ START OF ALL SUBROUTINES ------------------------
C

SUBROUTINE ACPRES
C ACPRES CALC. AND PRINTS THE STARTING PRESSURE FOR ACCUHULATOR
COMMON/TEST/ACPSI, AREANU, XLCON, CIRREF, CONDSR, CPERNT, DIFDSR,
* GAUAXL, GAURAD, ICONF, ICIRF, ICIRF, ICONFF, IDIFFI,
* IFLAG, IFLAG, LOADF, LOADI, IPORE, IPOREF, PERCNT, PI, PISARE,
* RADCON, RLOAD, RSFALL, SPDIA, SPDIA,
* SNAXIL, SNCONF, SNDIFF, SNLOAD, SNPORE, SNRADL, STRSS1
C CALC. FORCE ON PISTON DUE TO CONFINING PRESSURE
FORCE1*CONDSR*PISARE
C CALC. NEEDED FORCE ON SPECIMEN
FORCE2*DIFDSR*SPDIA*SPDIA*PI/4.
C CALC. TOTAL FORCE
FORCE=FORCE1+FORCE2
C CALC. RAM PSI
ACPSI=FORCE/113.1
TYPE 10, ACPSI
10 FORMAT('CHARGE ACCUMULATOR TO ',F8.1,' PSI')
RETURN
END

SUBROUTINE ACQPAR
C ACQPAR COLLECTS DATA ACQUISITION PARAMETERS
COMMON/ACGDAT/FULSCL, FRATE, IRENT, ISTCHN, JRATE, NCHAN, PRESST,
* RTIME, STTIME
COMMON/ARRAYS/CONV<32>, DSCRPT<15.32>, HSFLAG<15>,
* FDSCRPT(15.2>, IRTCNT<6>, JUNKR<5>, RATES<6>, SCANS<6>, TIMSTR<2>
1 GO TO 40
2 TYPE 360
C COLLECT 1ST LINE OF FILE DESCRIPTION
TYPE 200
TYPE 320,(FDSCRPT(IDUMMY.1), IDUMMY.15)
CALL SAMASCFDSCRPT(I))
C COLLECT 2ND LINE
TYPE 210
TYPE 320,(FDSCRPT(IDUMMY.2), IDUMMY.15)
CALL SAMASC(FDSCRPT(I))
C COLLECT A/D FULLSCALE VALUE
FULSCL=FULSCL*2048.
TYPE 225,FULSCL
CALL SAME(FULSCL)
FULSCL=FULSCL/2048.
C COLLECT NUMBER OF CHANNELS
C (CHANNELS ARE NUMBERED 0-31 ON THE RTI-1250).
20 TYPE 220, NCHAN
CALL SAME(NCHAN)
IF (NCHAN.GT.32) GO TO 20
IF (NCHAN.LE.0) GO TO 20
C COLLECT STARTING CHANNEL NUMBER
30 TYPE 230, ISTCHN
CALL SAME(ISTCHN)
IF (ISTCHN.LT.0. OR. ISTCHN.GT.31) GO TO 30
C COLLECT CHANNEL DESCRIPTION AND CONVERSION FACTOR
TYPE 235
C ISTCHN MAY BE 0; ARRAYS BEGIN NUMBERING WITH 1
C SO STORE ISTCHN DESCRIPTION AT 1ST LOCATION IN ARRAY
DO 35 IDUMMY=ISTCHN+1,ISTCHN+NCHAN
Jdummy=IDUMMY-1
C REMOVE OFFSET TO GET CORRECT CHANN NUMBER
Jdummy=IDUMMY-1
TYPE 240, JDUUMY
C SET POINTER AT DESCRIPTION
Jdummy=IDUMMY-ISTCHN
TYPE 320,(DSCRPT(I),JDUUMY), I=1,15
CALL SAMASC(DSCRPT(I),JDUUMY))
TYPE 245,C( CONV(JDUUMY))
CALL SAME(CONV(JDUUMY))

27
APPENDIX (cont)

35      CONTINUE
C PRINT OUT DATA ACQUISITION PARAMETERS
40      TYPE 360
        TYPE 200
        TYPE 320,(FDSCR$(IDUMMY=1),IDUMMY=1,15)
        TYPE 300
        TYPE 210
        TYPE 320,(FDSCR$(IDUMMY=2),IDUMMY=1,15)
        TYPE 300
        FULSCL=FULSCL*2048.
        TYPE 225,FULSCL
        FULSCL=FULSCL/2048.
        TYPE 300
        TYPE 220,NCHAN
        TYPE 300
        TYPE 230,ISTCHN
        TYPE 350
        TYPE 235
        ACCEPT 330,ANSW
C PRINT CHANN DESCRIPTION AND CONV. VALUE
KCOUNT=0
DO 45 IDUMMY=ISTCHN+1,ISTCHN+NCHAN
      JDUMMY=IDUMMY-1
      TYPE 240,JDUMMY
      JDUMMY-IDUMMY-ISTCHN
      TYPE 320,<PSCRPT$(JDUMMY),I1.15>
      TYPE 245,(CONV(JDUMMY))
      TYPE 350
      KCOUNT=KCOUNT+1
      IF(KCOUNT.LT.5) GO TO 45
        TYPE 215
        ACCEPT 330,ANSW
        KCOUNT=0
45      CONTINUE
C SEE IF O.K.
46    TYPE 310
        ACCEPT 330,ANSW
        IF(ANSW.EQ.,'Y') RETURN
        IF(ANSW.EQ.,'N') GO TO 2
        GO TO 46
200    FORMAT(1ST LINE FILE DESCRIPTION')
210    FORMAT(2ND LINE FILE DESCRIPTION')
215    FORMAT('TYPE <RET> TO CONTINUE : ',$)
220    FORMAT('NUMBER OF CHANNELS',I2,')$
225    FORMAT('A/D FULLSCALE (+/- VOLTS ',F10.3,')$
230    FORMAT('STARTING CHANNEL NUMBER ',I2,')$
235    FORMAT('DESCRIPTION AND CONVERSION VALUE(UNITS/VOLT)'$
240    FORMAT('CHAN',I2,'DESCRIPTION',')$
245    FORMAT('CONV. VALUE=',F10.3,')$
300    FORMAT('')
310    FORMAT('IS THIS CORRECT (Y OR N ) ',$)
320    FORMAT(4X,15A4,'$
330    FORMAT(A4)
340    FORMAT(F10.3)
350    FORMAT('')$
360    FORMAT('/// DATA ACQUISITION DESCRIPTION')
RETURN
END
APPENDIX (cont)

SUBROUTINE CHAN
C CHAN COLLECTS XDUCER CHANNEL ASSIGNMENTS
COMMON/CHANN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
1 GO TO 40
2 TYPE 360
C COLLECT CHANNEL ASSIGNMENTS
TYPE 200,KLOAD
CALL SAMEI(KLOAD)
TYPE 300
TYPE 210,KCONF
CALL SAMEI(KCONF)
TYPE 300
TYPE 220,KPORE
CALL SAMEI(KPORE)
TYPE 300
TYPE 230,KDIFF
CALL SAMEI(KDIFF)
TYPE 300
TYPE 240,KRAD1
CALL SAMEI(KRAD1)
TYPE 300
TYPE 250,KRAD2
CALL SAMEI(KRAD2)
TYPE 300
TYPE 260,KRAD3
CALL SAMEI(KRAD3)
TYPE 300
C PRINT CHANNEL ASSIGNMENTS
40 TYPE 360
TYPE 200,KLOAD
TYPE 300
TYPE 210,KCONF
TYPE 300
TYPE 220,KPORE
TYPE 300
TYPE 230,KDIFF
TYPE 300
TYPE 240,KRAD1
TYPE 300
TYPE 250,KRAD2
TYPE 300
TYPE 260,KRAD3
TYPE 300
C SEE IF O.K.
45 TYPE 310
ACCEPT 330,ANSW
IF(ANSW.EQ.'Y') RETURN
IF(ANSW.EQ.'N') GO TO 2
GO TO 45
200 FORMAT('LOAD CELL '12X)
210 FORMAT('CONFINING PRESSURE '12X)
220 FORMAT('PORE PRESSURE '12X)
230 FORMAT('DIFFERENTIAL PORE '12X)
240 FORMAT('RADIAL GAUGE #1 '12X)
250 FORMAT('RADIAL GAUGE #2 '12X)
260 FORMAT('RADIAL GAUGE #3 '12X)
300 FORMAT(')
310 FORMAT(' IS THIS CORRECT (Y OR N) '4X)
330 FORMAT(44)
350 FORMAT(')
360 FORMAT('/// CONTROL XDUCER CHANNEL ASSIGNMENT :')
END
SUBROUTINE CHANGE
C CHANGE ALLOWS OPERATOR TO CHANGE DATA ACQ./TEST PARAMETERS
CALL CHAN
CALL PARAM
CALL ACQPAR
CALL DATRAT
RETURN
END

SUBROUTINE CIRRFF
C CIRRFF CALC. REFERANCE STRAIN A/D UNITS, SPEC.DIA/CIRCUM AFTER THE
C APPLICATION OF CONFINE/PORE PRESS AND TEMP
C ENTER SPDIA=INIT. SPECIMEN DIA. IN., ICIRI=INIT.STRAIN A/D UNITS
C EXIT REF. VALUES: SPDIA=DIAM.(IN.), CIRREF=CIRCUM.(IN.), ICIRRF=STRAIN(A/D)
COMMON/PROCES/IBUFOOO, NBUF, ISIZ, NCEOUNT,
* IFAST, ICHAN1, ICHAN2, I, NGRPH, IYO(16), IX(2), LEVENT, LRECUD
COMMON/TEST/A CPSI, AREAAM, AXLCON, CIRREF, CONDSR, CPNRT, BIFDSR,
* BAXAL, BAXAR, ICONF, ICI, ICRF, ICONF, ICONF, IDIFFI,
* ICFLAG, IFLAG, LOADF, LOAD1, IP0RE, IP0RE, PERCENT, FI, PISARE,
* RAD, RADL, RSFLAG, SPDIA, SPDIA,
* SNAXIL, SCONF, SDIFF, SNLOAD, SNPORE, SNRAD, STRSS1
C MEASURE ALL 3 RADIAL GAUGES, FIND AVG., A/D UNITS
IDATA1=IBUF((J-ISTICN)+KRAD1)
IDATA2=IBUF((J-ISTICN)+KRAD2)
IDATA3=IBUF((J-ISTICN)+KRAD3)
IAVG=(IDATA1+IDATA2+IDATA3)/3
C CALC. DELTA A/D UNITS USING NO TEMP/PRESS VALUE
DELTA=IAVG-ICIRI
C CALC. EPSLON CONVERT FROM MILLISTRAIN TO STRAIN
EPSLON=DELTA*RA0CON/1000.
C CALC. NEW DIAMETER
SPDIAR=SPDIA*(1+EPSLON)
CIRREF=SPDIAR*FI
RETURN
END
SUBROUTINE CONFMF(ICFLAG)
C CONFMF INCREASES/DECREASES CONFINE PRESS.
IF(ICFLAG.EQ.0) GO TO 100
GO TO (1,2,3,4) ICFLAG
TYPE 1000 ICFLAG
1000 FORMAT(/' ERROR IN 'PUMP', ICFLAG=',I8)
GO TO 100
C THIS DECREASES PRESSURE
C SELECT DIRECTION TURN MOTOR ON SAVE TIME
1 IVALVE=8
CALL VALON(IVALVE)
IVALVE=9
CALL VALON(IVALVE)
T1=SECONDS(0)
ICFLAG=2
GO TO 100
C IF 2 SECONDS HAVE PASSED - TURN OFF MOTOR
2 T2=SECONDS(T1)
IF(T2.LT.2) GO TO 100
GO TO 90
C THIS INCREASES PRESSURE
3 IVALVE=8
CALL VALOFF(IVALVE)
IVALVE=9
CALL VALOFF(IVALVE)
T1=SECONDS(0)
ICFLAG=4
GO TO 100
C IF 2 SECONDS HAVE PASSED - TURN OFF MOTOR
4 T2=SECONDS(T1)
IF(T2.LT.2) GO TO 100
GO TO 90
C TURN OFF MOTOR CLEAR FLAG
90 IVALVE=9
CALL VALOFF(IVALVE)
IVALVE=8
CALL VALOFF(IVALVE)
ICFLAG=0
GO TO 100
100 RETURN
END

SUBROUTINE CURATE
C CURATE CALCULATES AND DISPLAYS CURRENT DATA RATE
COMMON/ARRAYS/CONV(32),DSCRPT(15,32),DSFLAG(15),
% FDSCR(15,2),IRTCNT(6),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2)
COMMON/PROCES/IBUF(1000),NBUF,XSIZ,NCOUNT,
% IFAST,ICHAN1,ICHAN2,JLONGPH,IXO(32),IX(2),LEVENT,IRECRI
C FIND POSITION IN ARRAY OF CURRENT DATA RATE
IDUMMY=1
10 IF(IRTCNT(IDUMMY).EQ.NCOUNT) GO TO 20
IDUMMY=IDUMMY+1
IF(IDUMMY.LT.6) GO TO 10
TYPE 200
RETURN
20 TYPE 210,IDUMMY,SCANS(IDUMMY)
RETURN
200 FORMAT(/' ERROR !!! CURRENT DATA RATE CANNOT BE CALCULATED'/)
210 FORMAT( ' CURRENTLY RUNNING DATA RATE ',F10.2,' SECS/SCAN'/)
END
APPENDIX (cont)

SUBROUTINE DATRAT
C DATRAT COLLECTS DATA ACQUISITION RATE PARAMETERS
COMM/ARRAYS/CONV(32),DESCRPT(15,32),DBFLAG(15),
* SCANS15,2),JUNKR(5),RATES(4),TIMSTR(2)
COMM/ACBDAT/FULSCL,FRATE,INTCNT,ISTCHN,JRATENCHAN,PRESET,
* RTIME,STTIME
COMMON/PROCES/IDBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,NOGRPH,ITYO(32),IX(2),LEVENT,IRECRO
1 GO TO 25
2 TYPE 360
C COLLECT DATA RATES
15 TYPE 200,(SCANS(6))
CALL SAME(SCANS(6))
IF(SCANS(6).LT.01)GO TO 15
C INSURE CORRECTNESS OF PRESET
FRATE=1./SCANS(6)
IDUMMY=100./FRATE
PRESET=IDUMMY
FRATE=100./PRESET
SCANS(6)=1./FRATE
ITRCNT(6)=PRESET
C COLLECT SLOWER DATA RATES
10 20 IDUMMY=1,5
18 TYPE 210,(IDUMMY,SCANS(IDUMMY))
CALL SAME(SCANS(IDUMMY))
TYPE 300
IF(SCANS(IDUMMY)/32000.GT.SCANS(6))GO TO 18
C INSURE CORRECTNESS OF RATES AND SCAN VALUES
RATES(IDUMMY)=1./SCANS(IDUMMY)
ITRCNT(IDUMMY)=FRATE/RATES(IDUMMY)
RATES(IDUMMY)=FRATE/FLOAT(ITRCNT(IDUMMY))
SCANS(IDUMMY)=1./RATES(IDUMMY)
20 CONTINUE
C PRINT OUT DATA RATES
25 TYPE 360
TYPE 200,(SCANS(6))
TYPE 300
DO 30 IDUMMY=1,5
TYPE 210,(IDUMMY,SCANS(IDUMMY))
TYPE 300
30 CONTINUE
35 CONTINUE
C SEE IF O.K.
32 ACCEPT 330,ANSW
IF(ANSW.EQ.'Y') RETURN
IF(ANSW.EQ.'N') GO TO 2
GO TO 45
200 FORMAT(' FAST DATA RATE (SECS/SCAN) ',F10.2,*)
210 FORMAT(' * ',12,' DATA RATE (SECS/SCAN) ',F10.2,*)
300 FORMAT(1X,' IS THIS CORRECT (Y OR N) ',*)
320 FORMAT(4X,15A4,*)
330 FORMAT(A4)
340 FORMAT(F10.3,*)
350 FORMAT(//,' DATA ACQUISITION RATES :')
END
APPENDIX (cont)

SUBROUTINE FIMOFF
C FIMOFF COLLECTS XDECER OFFSET (A/D UNITS) AFTER TEMP/PRESS SET
COMMON/CHANN/KLOAD,KCONF,KPORE,KGRA,KRAD1,KRAD2,KRAD3
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,MOORPH,IXO(32),IX(2),LEVENT,IRECRD
COMMON/TEST/ACPSI,AREANW,AXCON,CIRREF,COND,CPERNT,DIFDSR,
* GAUAXL,GOURAD,ICONF,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF,
* ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PIARE,
* RACION,KLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAXI1,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1
LOADF=IBUF((J-ISTCHN)+KLOAD)
ICONFF=IBUF((J-ISTCHN)+KCONF)
IPOREF=IBUF((J-ISTCHN)+KPORE)
C MEASURE ALL 3 RADIAL GAUGES, FIND AVG.
IDATA1=IBUF((J-ISTCHN)+KRAD1)
IDATA2=IBUF((J-ISTCHN)+KRAD2)
IDATA3=IBUF((J-ISTCHN)+KRAD3)
ICIRF=(IDATA1+IDATA2+IDATA3)/3
RETURN
END

SUBROUTINE GRIDVR
C DRAW CRUDE GRID AND PRINT VARIABLE LABELS
LOGICAL*1 CANiSUB,ESC,FS,GS,US,DEL,LITTLA,FF
LOGICAL*1 ISTRS(15)
LOGICAL*1 OLDSTR(15)
LOGICAL*1 LABSTR(7)
C
DIMENSION KHARUO)
DIMENSION IXX(2)
C
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,MOORPH,IXO(32),IX(2),LEVENT,IRECRD
COMMON/SCREEN/KDISP,IKLOAD,IKTEMP,QONST1,QONST2,QTEMP
COMMON/TEST/ACPSI,AREANW,AXCON,CIRREF,COND,CPERNT,DIFDSR,
* GAUAXL,GOURAD,ICONF,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF,
* ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PIARE,
* RACION,KLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAXI1,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1
C
DATA IXX/1295.4095/
DATA CAN SUB,ESC,FS,GS,US,DEL,LITTLA/"30","32","33","34","35","37",
"177","141/"
DATA FF/"14/"
DATA KHAR/"60","61","62","63","64","65","66","67","70","71/"
DATA ICOUNT/1/
DATA LABSTR/1H$ ,1H$ ,1H$ ,1H$ ,1H$ ,1H$ ,1H$ /
C
C CLEAR SCREEN
C GO TO POINT MODE
CALL TKPACK(FS)
C GO TO 4010 ALPHA, CLR SCREEN, DATA=WHITE, HOME CURSOR
CALL TKPACK(ESC)
CALL TKPACK(FF)
C GO TO ADM-3 ALPH
CALL TKPACK(CAN)
C - FULL SCALE LINE
C GO TO VECTOR MODE
CALL TKPACK(DS)
APPENDIX (cont)

C SEND FIRST CO-ORDINATE
CALL TKVECT(IXX(1),1)
C SEND SECOND CO-ORDINATE \ DRAW LINE
CALL TKVECT(IXX(2),1)
C -HALF SCALE LINE
C PREPARE TO ACCEPT FIRST CO-ORD.
CALL TKPACK(GS)
CALL TKVECT(IXX(1),780)
CALL TKVECT(IXX(2),780)
C 0 LINE
CALL TKPACK(GS)
CALL TKVECT(IXX(1),1560)
CALL TKVECT(IXX(2),1560)
C +HALF SCALE LINE
CALL TKPACK(GS)
CALL TKVECT(IXX(1),2340)
CALL TKVECT(IXX(2),2340)
C +FULL SCALE LINE
CALL TKPACK(GS)
CALL TKVECT(IXX(1),3120)
CALL TKVECT(IXX(2),3120)

C PRINT 'STRESS' TO GRAPHIC SCREEN
C GO TO VECTOR MODE
CALL TKPACK(GS)
C POSITION CURSOR
CALL TKVECT(1295,3000)
C GO TO 4010 ALPHA MODE
CALL TKPACK(IS)
C WRITE LABEL TO GRAPHICS SCREEN
DO 145 IDUMMY=1,7
CALL TKPACK(LABSTR(IDUMMY))
145 CONTINUE
C GO TO ADM-3 ALPHA
CALL TKPACK(CAN)
RETURN
END

SUBROUTINE INTACQ
C INITIALIZE PARAMETERS FOR STARTING DATA ACQUISITION SYS.
C NBUF, ICMF ARE PARAMETERS FOR TKDATA And SETR
C J IS AN INDEX USED IN THE COMPLETION ROUTINE.

COMMON/ARRAYS/CONV(32),DSRPT(15,32),DSFLAG(15),
* GRIDT(15,2),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2)
COMMON/ACODAT/FULSCL,FRATE,INTCNT,ISTCHN,JRATE,NCHAN,PRECSET,
* RTIME,STTIME
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,NOGRAPH,IY(32),IX(2),LEVENT,IRECRDN
IDUMMY=1000
NBUF=IDUMMY/NCHAN
ISIZ=NCHAN*NBUF
IRECRD=0
INTCNT=1
ICMF=0
J=1
IFAST=0
NCOUNT=INTCNT(1)
JRATE=5
RETURN

C STTIME IS STARTING TIME IN SECONDS PAST MIDNIGHT OR BOOT
C RTIME IS TIME IN SECONDS RELATIVE TO STTIME
C STTIME IS FIXED WHEN THE FIRST DATA FLOPPY IS STARTED. RTIME IS
C CALCULATED TO GIVE THE TIME FROM THE START OF THE EXPERIMENT AND IS
C WRITTEN TO EACH OF THE FLOPPYS USED IN THE TEST.
C TIMSTR IS AN ASCII STRING giving SYSTEM TIME IN HR:MIN:SEC FORMAT.
END
APPENDIX (cont)

SUBROUTINE INTOFF
C INTOFF COLLECTS INITIAL OFFSET A/D UNITS FOR VARIOUS XDUCERS
COMMON/CHAN/kLOAD,kCONF,kPORE,kDIFF,kRAD1,kRAD2,kRAD3
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,NODRPH,LOYO(32),IX(2),LEVENT,IRECRD
COMMON/TEST/ACPSI,AREANW,AXLCON,CIRREF,CONDsr,CPERNT,DIFDSR,
* GAUAXL,GaURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF;
* ICONF,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PISARE;
* RADCON,RL0AD,RSFLAG,SPDIA,SPDIAR,
* SNAXL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1
C
C GET INIT. VALUES FOR LOAD CELL, CONFIN AND PORE XDUCERS, RADIAL GAUGES
C GET INIT. VALUES FOR LOAD CELL (LOADI)
1 LOADI=IBUF(J-ISTCHN)+kLOAD
IDUMMY=kLOAD
IF(IABS(LOADI).GT.2) GO TO 1020
C GET INIT. VALUES FOR CONFINING PRESSURE (ICONFI)
ICONFI=IBUF(J-ISTCHN)+kCONF
IDUMMY=kCONF
IF(IABS(ICONFI).GT.2) GO TO 1020
C GET INIT. VALUES FOR PORE PRESSURE (IPORE)
IPORE=IBUF(J-ISTCHN)+kPORE
IDUMMY=kPORE
IF(IABS(IPORE).GT.2) GO TO 1020
C GET INIT. VALUES FOR RADIAL GAUGES (CIRIN)
C MEASURE ALL 3 RADIAL GAUGES, FIND AVG., SAVE IN A/D UNITS
IDATA1=IBUF(J-ISTCHN)+kRAD1
IDUMMY=kRAD1
IF(IABS(IDATA1).GT.2) GO TO 1020
IDATA2=IBUF(J-ISTCHN)+kRAD2
IDUMMY=kRAD2
IF(IABS(IDATA2).GT.2) GO TO 1020
IDATA3=IBUF(J-ISTCHN)+kRAD3
IDUMMY=kRAD3
IF(IABS(IDATA3).GT.2) GO TO 1020
ICIRF=(IDATA1+IDATA2+IDATA3)/3
RETURN
1020 TYPE 1025,IDUMMY
1025 FORMAT(2I2) THEN HIT <RET> *, THEN HIT <RET> '*,
ACCEPT 1030, ANSW
1030 FORMAT(A4)
GO TO 1
END

SUBROUTINE MENUE(MENFLG)
C MENUE PROCESSES OPERATOR REQUESTS
COMMON/ACQDAT/FULSCL,FRATE,INTCNT,ISTCHN, J RATE, NCHAN, PRESRT,
* RTIME, STTIME
COMMON/ARRAYS/CONV(32),DSCRP(15,32),DSFLAG(15),
* FDS (15,2), JRTCTN(4), JUNKR(5), RATES(6), SCANS(6), TIMSTR(2)
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,NODRPH,LOYO(32),IX(2),LEVENT,IRECRD
COMMON/TEST/ACPSI,AREANW,AXLCON,CIRREF,CONDsr,CPERNT,DIFDSR,
* GAUAXL,GaURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF;
* ICONF,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PISARE;
* RADCON,RL0AD,RSFLAG,SPDIA,SPDIAR,
* SNAXL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1
C
MENFLG=0
C TYPE <CR> TO CHANGE OPTIONS
100 TYPE 200
200 FORMAT(’ TYPE <CR> FOR OPTION, ’$)
ACCEPT 210,IDUMMY
210 FORMAT(A4)

APPENDIX (cont)

C TEST IF ERROR FROM COMPLETION ROUTINE ‘STORE’
IF (ICHF.LT.O) STOP ‘DATA OVERRUN, USE SLOWER RATE!!’
IF (IDUMMY.NE.’C’) GO TO 100

215 TYPE 220
220 FORMAT (‘ OPTIONS!’,’/
  1) GRAPH’,’/
  2) NO GRAPH’,’/
  3) CHANGE DATA RATE’,’/
  4) WRITE EVENT FLAG’,’/
  5) CLEAR SCREEN’,’/
  6) NEW DATA FILE’,’/
  7) STOP TEST’,’/
  8) NO OPERATION’,’/
  9) POP VALVE’,’/
 10) BLOCK COUNT’,’/
 11) CHANGE TEST PARAMETERS’)
TYPE 230
230 FORMAT (‘ OPTION? ’,’$
ACCEPT 240,IDUMMY

240 FORMAT (12)
IX(IDUMMY).LT.1,OR.IDUMMY.GT.11) GO TO 215
GO TO (500,510,520,530,540,550,560,570,580,590,600) IDUMMY

C C GRAPH ON
500 CONTINUE
NGRAPH=0
IX(2)=1295
RETURN

C NO GRAPH
510 CONTINUE
NGRAPH=1
RETURN

C CHANGE DATA RATES
520 CONTINUE
TYPE 527
TYPE 523,(SCANS(6))
TYPE 525
DO 521 IDUMMY=1,5
  TYPE 524,(IDUMMY,SCANS(IDUMMY))
TYPE 525
521 CONTINUE

TYPE 528
ACCEPT 529,IRATE
IF (IRATE.LE.0) GO TO 522
IF (IRATE.GT.5) GO TO 520
IFAST=0
JUNK=JUNKR(IRATE)
WRITE (ICHAN2) (JUNK,IDUMMY*1,NCHAN+1)
NCOUNT=IRTCTN(IRATE)
RETURN

522 CONTINUE
IF (IFAST.EQ.1) RETURN
IFAST=1
WRITE (ICHAN2) (JUNK,IDUMMY*1,NCHAN+1)

523 FORMAT(’ RATE @ 0 = FAST DATA RATE (SECS/SCAN) ’,F10.2,’$)
524 FORMAT(’ RATE @ ’,I2,’ = (SECS/SCAN) ’,F10.2,’$)
525 FORMAT(’ ’)
526 FORMAT(’ ’)
527 FORMAT(’ /// DATA ACQUISITION RATES ’)
528 FORMAT(’ DESIRED RATE ’,I’,$)
529 FORMAT (I2)
RETURN

C WRITE AN EVENT FLAG TO CHANNEL NCHAN+1
530 CONTINUE
APPENDIX (cont)

KEVENT=KEVENT+1
KEVENT=KEVENT
TYPE 531,KEVENT

531 FORMAT (' Event Flag Number ',I3, ' Written')
RETURN
C CLEAR THE SCREEN
540 CONTINUE
CALL TKPACK("34")
CALL TKPACK("33")
CALL TKPACK("14")
CALL TKPACK("30")
RETURN
C WRITE NEW DATA FILE
550 CONTINUE
C TURN DATA ACQ. OFF
CALL SETR(-1,,)
C DO DUMMY DELAY FOR COMPLETION OF PRESENT SCAN
DO 551 IDUMMY=1,500
   DUMMY*SQRT(<FLOAT(IDUMMY))*SQRT(<FLOAT(IDUMMY)>)
551 CONTINUE
C SET RT-11 CLOCK = BATT CLOCK (GET CORRECT DATE ONLY)
CALL TIMDAT
C CLOSE EXISTING FILE
CALL CLOSE(ICHAN2)
TYPE 552
552 FORMAT (' Enter New Data File Name! ')*
C OPEN NEW DATA FILE
CALL ASSIGN(ICHAN2,"*,-1")
REWIND ICHAN2
C WRITE HEADER DATA ON DISC
CALL WTHEAD(ICHAN2)
C RE-INITIALIZE SOME VARIABLES AND CLEAR SCREEN
IFAST=0
NCOUNT=IRTcnt(1)
ICMF=0
NBUF=1000/NCHAN
IRECRD=0
J=1
IX(2)=1295
CALL TKPACK("34")
CALL TKPACK("33")
CALL TKPACK("14")
CALL TKPACK("30")
C GO TO RESTART THE DATA ACQUISITION
TYPE 553
553 FORMAT(// Type <RET> TO BEGIN TAKING DATA ')*
ACCEPT 554,DUMMY
554 FORMAT(44)
MENFLG=1
RETURN
C NO OPERATION
570 CONTINUE
RETURN
C POP VALVE TO LOAD RAM IN CREEP TEST AND BEGIN LOAD CONTROL
580 CONTINUE
CALL POP
RETURN
C BLOCK COUNT
590 CONTINUE
IBLOCK=IRECRD*(NCHAN+2.25)/256+2
TYPE 591,IBLOCK
591 FORMAT(//' Number of Blocks Used So Far Is: ',I5)
RETURN
APPENDIX (cont)

C STOP
560 CONTINUE
C TURN OFF DATA ACQ.
CALL SETR(-2,
C CLOSE EXISTING FILE
CALL CLOSE(ICHAN2)
CALL SYSINT
TYPE 561
561 FORMAT ('//'}GOOD-BY FOLKS! ')
STOP 'PROGRAM FINISHED'
C CHANGE TEST PARAMETERS
C CLEAR SCREEN
600 CALL TKPACK('34)
CALL TKPACK('33)
CALL TKPACK('14)
CALL TKPACK('30)
CALL PARAM
C REWRITE RESTART FILE
CALL ASSIGN (ICHAN1,'DYo!RESTRT.DAT',14)
CALL WTRSTR(ICHAN1)
CALL CLOSE (ICHAN1)
RETURN
END

SUBROUTINE NWAREA
C NWAREA MEASURES PRESENT RADIAL STRAIN AND CALCULATES THE NEW AREA
C OF THE SPECIMEN.
C ENTER CIRREF=REF.SPEC.CIRCUM.(IN.)*ICIRRF=REF.STRAIN (A/D)
C EXIT AREANW=NEW AREA (IN.)
C
COMMON/CHANN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
C
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
C IFAST,ICHAN1,ICHAN2,JNODGRPH,IYO(16),IX(2),LEVENT,IRECkD
C
COMMON/TEST/ACPS,NWAREA,AXLCON,CIRREF,CONDSR,CPERNT,DIFDSR,
C GAUAXL,GAURAD,KCONF,ICIRF,ICIRF,ICONFI,ICONFF,IDENT,
C CIRFLAG,FFLAG,LOADF,LOADI,IPORE,IPOLR,PERCNT,PI,PISARE,
C RADCON,RLOAD,RFLAG,SPDIA,SPDIAR,
C SNAZIX,NCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1
C MEASURE ALL 3 RADIAL GAUGES, FIND AVG., CONVERT TO VOLTS
IDATA1=IBUF((J-ISTCHN)+KRAD1)
IDATA2=IBUF((J-ISTCHN)+KRAD2)
IDATA3=IBUF((J-ISTCHN)+KRAD3)
IAVG=(IDATA1+IDATA2+IDATA3)/3
C CALC. DELTA A/D UNITS
DELTA=(IAVG-ICIRRF)
C CALC. NEW CIRCUM.
EPSON=DELTA*KRadCON/1000.
CIRNEW=CIRREF*(1+EPSON)
C CALCULATE NEW AREA IN INCHES
AREANW=(CIRNEW*CIRNEW)/(4.0*PI)
RETURN
END
SUBROUTINE PARAH
C PARAM COLLECTS ALL TEST PARAMETERS
COMMON/CHANK/LOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/TEST/ACPSI,AREMAW,AXLCON,CIRREF,CONDSDR,CPERNT,DIFDSR,
* GAUAXL,GUARAD,ICONFP,ICIRF,ICIRI,ICONF1,ICONFF,ICNFF,ICNFF1,
* ICFLAG,ICFLG,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PISARE,
* RADCON,KLOAD,KSPFLAG,SPDIA,SPDIAR,
* SNAXIL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRSS1

C
GO TO 80
1 TYPE BB
   TYPE 200,SPFIA
   CALL SAME(SPIDIA)
   TYPE 205,DIFDSR
   CALL SAME(DIFDSR)
   VAR=100.*PERCNT
   TYPE 208,VAR
   CALL SAME(VAR)
   PERCNT=VAR/100.
   TYPE 210,CONDSDR
   CALL SAME(CONDSDR)
   VAR=100.*CPERNT
   TYPE 212,VAR
   CALL SAME(VAR)
   CPERNT=VAR/100.
   TYPE 225,SNLOAD
   CALL SAME(SNLOAD)
   TYPE 230,SNCONF
   CALL SAME(SNCONF)
   TYPE 235,SNPORE
   CALL SAME(SNPORE)
   TYPE 233,SNDIFF
   CALL SAME(SNDIFF)
   TYPE 240,GAUAXL
   CALL SAME(GAUAXL)
   TYPE 245,GAURAD
   CALL SAME(GAURAD)
   TYPE 250,SNAXIL
   CALL SAME(SNAXIL)
   TYPE 255,SNRADL
   CALL SAME(SNRAPl)
C CALC. AXIAL CONVERSION FACTOR
AXLCON=GAUAXL*SNAXIL/4096.
C CALC. RADIAL CONVERSION FACTOR
RADCON=GAURAD*SNRADL/4096.

C
C PRINT OUT TEST PARAMETERS
80 TYPE BR
   TYPE 200,SPFIA
   TYPE 500
   TYPE 205,DIFDSR
   TYPE 500
   VAR=100.*PERCNT
   TYPE 208,VAR
   TYPE 210,CONDSDR
   TYPE 500
   VAR=CPERNT*100.
   TYPE 212,VAR
   TYPE 225,SNLOAD
   TYPE 500
   TYPE 235,SNPORE
   TYPE 500
   TYPE 233,SNDIFF
   TYPE 500

39
APPENDIX (cont)

TYPE 500
TYPE 235, SPORE
TYPE 500
TYPE 238, SNDIFF
TYPE 500
TYPE 240, GAUAXL
TYPE 500
TYPE 245, GAURAD
TYPE 500
TYPE 250, SNAXIL
TYPE 500
TYPE 255, SWRADL
TYPE 500

C NOW SEE IF ALL DATA CORRECT
100 TYPE 290
ACCEPT 295, ANS
IF (ANS .EQ. 'Y') RETURN
IF (ANS .EQ. 'N') GO TO 1
GO TO 100

200 FORMAT (' SPECIMEN DIAMETER (INCHES) ' 1X, F10.3, $)
205 FORMAT (' DESIRED DIFFERENTIAL STRESS (PSI) ' 12X, F10.3, $)
208 FORMAT (' DESIRED STRESS CONTROL PERCENTAGE ' 12X, F10.3, $)
210 FORMAT (' DESIRED CONFINING PRESSURE (PSI) ' 13X, F10.3, $)
212 FORMAT (' DESIRED CONF. PRESS. CONTROL PERCENTAGE ' 8X, F10.3, $)
220 FORMAT (' LOAD CELL RANGE (POUNDS FULL SCALE) ' 10X, F10.3, $)
225 FORMAT (' CONF. PRESSURE XDUCER RANGE (PSI FULL SCALE) ' F10.3, $)
230 FORMAT (' DIFF. PRESS. XDUCER RANGE (PSI FULL SCALE) ' F10.3, $)
235 FORMAT (' AXIAL STRAIN GAUGE FACTOR ' 20X, F10.3, $)
240 FORMAT (' RADIAL STRAIN GAUGE FACTOR ' 19X, F10.3, $)
245 FORMAT (' AXIAL STRAIN RANGE (MILLISTRAIN FULL SCALE) ' 2X, F10.3, $)
250 FORMAT (' RADIAL STRAIN RANGE (MILLISTRAIN FULL SCALE) ' F10.3, $)
255 FORMAT (' IS THIS CORRECT? (Y OR N) ' 4X)
290 FORMAT (' ')
APPENDIX (cont)

SUBROUTINE PUMP(IFLAG)
C PUMP EITHER CHARGES OR DISCHARGES ACCUM., DEPENDING ON IFLAG
IF(IFLAG.EQ.0) GO TO 100
GO TO(1,2,3,4,5,6,7,8,9,10,100),IFLAG
TYPE 1000,IFLAG
1000 FORMAT('/' ERROR IN "PUMP", IFLAG='18>
GO TO 100
C THIS IS THE CHARGE ROUTINE
C OPEN VAL. 1; SAVE TIME
I VALVE=1
CALL VALON(I ValVE)
T1=SECONDS(0)
IFLAG=2
GO TO 100
C IF 2 SECS. HAVE PASSED, CLOSE VAL. 1; SAVE TIME
T2=SECONDS(T1)
IF(T2.LT.2) GO TO 100
CALL VALOFF(I ValVE)
T1=SECONDS(0)
IFLAG=3
GO TO 100
C IF 1 SEC. HAS PASSED, OPEN VAL. 2; SAVE TIME
T2=SECONDS(T1)
IF(T2.LT.1) GO TO 100
IVALVE=2
CALL VALON(I ValVE)
T1=SECONDS(0)
IFLAG=4
GO TO 100
C IF 2 SECS. HAVE PASSED, CLOSE VAL. 2; SAVE TIME
T2=SECONDS(T1)
IF(T2.LT.2) GO TO 100
CALL VALOFF(I ValVE)
IFLAG=5
GO TO 100
C IF 1 SEC. HAS PASSED, CLEAR IFLAG, DONE!
T2=SECONDS(T1)
IF(T2.LT.1) GO TO 100
IFLAG=0
GO TO 100
C THIS IS THE DISCHARGE ROUTINE
C OPEN VAL. 2; SAVE TIME
IVALVE=2
CALL VALON(I ValVE)
T1=SECONDS(0)
IFLAG=7
GO TO 100
C IF 2 SECS. HAVE PASSED, CLOSE VALVE 2
T2=SECONDS(T1)
IF(T2.LT.2) GO TO 100
CALL VALOFF(I ValVE)
T1=SECONDS(0)
IFLAG=8
GO TO 100
C IF 1 SEC. HAS PASSED, OPEN VALVE 3
T2=SECONDS(T1)
IF(T2.LT.1) GO TO 100
IVALVE=3
CALL VALON(I ValVE)
T1=SECONDS(0)
IFLAG=9
GO TO 100
APPENDIX (cont)

C IF 2 SECS. HAVE PASSED, CLOSE VALVE 3
9 T2=SECnds(T1)
   IF(T2.LT.2) GO TO 100
   CALL VALOFF(IValve)
   T1=SECnds(0)
   IFlag=10
   GO TO 100
C IF 1 SEC. HAS PASSED, CLEAR IFLAG, DONE!
10 T2=SECnds(T1)
   IF(T2.LT.1) GO TO 100
   IFlag=0
   GO TO 100
100 RETURN
END

SUBROUTINE RDACQ(ILUN)
C RDACQ READS FULSCL, FRATE, INTCNT, ISTCHN, NCHAN, PRESET, RTIME
C STTIME FROM DISC
COMMON/ACGDAT/FULSCL, FRATE, INTCNT, ISTCHN, JRATE, NCHAN, PRESET,
  $ RTIME, STTIME
READ (ILUN) FULSCL, FRATE, INTCNT, ISTCHN, JRATE, NCHAN, PRESET,
  $ RTIME, STTIME
RETURN
END

SUBROUTINE RDARAY(ILUN)
C RDARAY READ ARRAYS CONV, IRTCNT, RATES, SCANS, TIMSTR FROM DISC
COMMON/ARRAYS/CONV(32), DSCRPT(15,32), DSFLAG(15),
  $ FDSRCP(15,2), IRTCNT(6), JUNKR(5), RATES(6), SCANS(6), TIMSTR(2)
C
C READ CONV
READ (ILUN) (CONV(IDUMMY), IDUMMY=1,6)
C READ IRTCNT
READ (ILUN) (IRTCNT(IDUMMY), IDUMMY=1,6)
C READ RATES
READ (ILUN) (RATES(IDUMMY), IDUMMY=1,6)
C READ SCANS
READ (ILUN) (SCANS(IDUMMY), IDUMMY=1,6)
C READ TIMSTR
READ (ILUN) TIMSTR
RETURN
END
SUBROUTINE RDHEAD(ILUN)
C RDHEAD READS HEADER FROM DISCS
COMMON/ARRAYS/CONM(32),DSCRP(15,32),DSFLAG(15),
  FDSCRP(15,2),IRTCNT(6),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2),
  COMMON/ACQDAT/FULSCL,FRATE,INTCNT,ISTRCHN,JRATE,NCHAN,PRESET,
  TIME,STTIME

C READ FIRST FILE DESCRIPTION LINE
READ (ILUN) (FDSCRP(IDUMMY,1,IDUMMY=1,15)
C READ SECOND FILE DESCRIPTION LINE
READ (ILUN) (FDSCRP(IDUMMY,2,IDUMMY=1,15)
C READ NO.CHANS+FLAG
READ (ILUN) NCHAN
C CORRECT NCHAN (IT'S STORED AS NCHAN + 1 EVENT FLAG
NCHAN=NCHAN-1
C READ '2' TO SIGNIFY TYPE 2 FORMAT FILE
IDUMMY=2
READ (ILUN) IDUMMY
C READ FASTEST RATE + 5 SLOWER RATES
READ (ILUN) FRATE,(RATES(IDUMMY),IDUMMY=1,5)
C READ STARTING CHANNEL NO., A/D FULLSCALE
READ (ILUN) ISTCHN,FULSCL
DO 385 IDUMMY=1,NCHAN
C READ CHANNEL DESCRIPTION
READ (ILUN) (DSCRPT(I,IDUMMY),I=1,15)
C READ CHANNEL CONVERSION FACTOR
READ (ILUN) CONV(IDUMMY)
385 CONTINUE
C READ EVENT FLAG DESCRIPTION
READ (ILUN) (DSFLAG(IDUMMY),IDUMMY=1,15)
C READ CONVERSION FACTOR FOR FLAG CHANNEL
READ (ILUN) FULSCL
READ (ILUN) DUMMY
RETURN
END

SUBROUTINE RDRSTR(ILUN)
C RDRSTR READS FROM DISCS: DATA ACQUISITION AND TEST PARAMETERS
COMMON/ACQDAT/FULSCL,FRATE,INTCNT,ISTRCHN,JRATE,NCHAN,PRESET,
  TIME,STTIME
COMMON/ARRAYS/CONV(32),DSCRP(15,32),DSFLAG(15),
  FDSCRP(15,2),IRTCNT(6),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2)
COMMON/CHANN/KLOAD,KCONF,KRAD1,KRAD2,KRAD3
COMMON/F0PFLG/IP0P
COMMON/PROCESS/IBUF(1000),NBUF,ISIZ,NCOUNT,
  IFAST,ICHAN1,ICHAN2,J,NOGRAPH,IYO(32),IX(2),LEVENT,IRECRD
COMMON/SCREEN/KDISP,1QLOAD,1QTEMP,Q0NST1,Q0NST2,QTEMP
COMMON/TEST/ACPSR,AREA0,AXLCON,CIRREF,CONDSR,CPRNT,DIFDSR
  GAUXL,GARAD,ICONF,ICIFR,ICIRF,ICONF,ICONFF,IC0DII,
  ICFLAG,IFLAG,LOADF,LOAD1,IP0RE,IP0REF,PERCNT,PI,PISARE,
  RADCON,RLOAD,RSLAG,SPDIA,SPDIAR,
  SNAXIL,SNC0NF,SNDIFF,SNLOAD,SNPORE,SNRAD1,STRSS1
CALL RDHEAD(ILUN)
CALL RDRARAY(ILUN)
CALL RDACR(ILUN)
CALL RDTPAR(ILUN)
RETURN
END
APPENDIX (cont)

SUBROUTINE RDTPAR(ILUN)
C RDTPAR READS TEST PARAMETERS FROM DISC
COMMON/CHANN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/TEST/ACPSI,AREANU,AXLCON,CIRREF,CONDSR,CPERN,T,DIFDSR,
$ GAUAXL,Gaurad,ICONFP,ICIRF,ICONFI,ICIRF,ICIRF,ICIRF,ICIRF,
$ ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPORE,PERCNT,PI,PISARE,
$ RADCON,KLOAD,RSFLAG,SPDIA,SPDIAR,
$ SNAXIL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRESS
C
READ (ILUN) ACPSI,AREANU,AXLCON,CIRREF,CONDSR,CPERN,T,DIFDSR,
$ GAUAXL,Gaurad,ICONFP,ICIRF,ICONFI,ICIRF,ICIRF,ICIRF,ICIRF,
$ ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPORE,PERCNT,PI,PISARE,
$ RADCON,KLOAD,RSFLAG,SPDIA,SPDIAR,
$ SNAXIL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRESS
READ (ILUN) KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
RETURN
END

SUBROUTINE RUN
C RUN CONTROLS DIFFERENTIAL STRESS AND CONFINING PRESS.
COMMON/CHANN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/PDOPFLG/IPOP
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
$ IFAST,ICHAN3,ICHAN2,J,NOGPH,IXO,IX2),LEVENT,IRECD
COMMON/TEST/ACPSI,AREANU,AXLCON,CIRREF,CONDSR,CPERN,T,DIFDSR,
$ GAUAXL,Gaurad,ICONFP,ICIRF,ICONFI,ICIRF,ICIRF,ICIRF,ICIRF,
$ ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPORE,PERCNT,PI,PISARE,
$ RADCON,KLOAD,RSFLAG,SPDIA,SPDIAR,
$ SNAXIL,SNCONF,SNDIFF,SNLOAD,SNPORE,SNRADL,STRESS
C
DATA IPPO/0/
DATA ICFLAG/0/
DATA IFLAG/0/
C
C IS VALVE POPPED YET?
IF (IPPO.EQ.0) RETURN
IVALVE=0
CALL VALOFF(IVALVE)
20 CONTINUE
C IF ICFLAG SET, PROCESS COMPRESS, OTHERWISE TEST CONF. PRESS.
IF (ICFLAG.GT.0) GO TO 1105
C IS CONFINING PRESSURE CORRECT?
ICONFP=IBUF(J-ISTCHN)+KCONF
CONPS=SNCONF*(ICONFP-ICONFI)/2048.
IF(CONPS.GT.(CONDGR*(1.+CPERN)) ICFLAG=1
IF(CONPS.LT.(CONDGR*(1.-CPERN)) ICFLAG=3
1105 CALL COMPRESS(ICFLAG)
C IF IFLAG SET, PROCESS PUMP, OTHERWISE TEST AXIAL STRESS
IF (IFLAG.GT.0) GO TO 1150
C CALC. PRESENT DIFFERENTIAL STRESS
CALL AXSTRS
C IS DIFFERENTIAL STRESS = DESIRED?
IF(STRESSG.T.(DIFDSR*(1.+PERCNT))) IFLAG=6
IF(STRESSG.LT.(DIFDSR*(1.-PERCNT))) IFLAG=1
1150 CALL PUMP(IFLAG)
RETURN
END
SUBROUTINE SAHASC(VAR)
C SAHASC INPUTS ASCII STRING.DEFAULT LEAVES VARIABLE AS IS
LOGICAL*! SCRAT(41)
DIMENSION VAR(15)
TYPE 10
CALL GETSTR(5,SCRAT,60,ERROR)
I=LEN(SCRAT)
IF(I.EQ.0) RETURN
DECODE(60,20,SCRAT)VAR
CONTINUE
10 FORMAT(60*E)
20 FORMAT(15A4)
RETURN
END

SUBROUTINE SAME(VAR)
C SAME INPUTS REAL VAR.DEFAULT LEAVES VAR. AS IS
TYPE 10
ACCEPT 100,DUMMY
IF(DUMMY.NE.0) VAR=DUMMY
10 FORMAT(60*E)
100 FORMAT(F10.3)
RETURN
END

SUBROUTINE SAHEKIVAR)
C SAMEI INPUTS INTEGER VAR.DEFAULT LEAVES AS IS
VAR=IVAR
CALL SAME(VAR)
IVAR=VAR+.5
RETURN
END

SUBROUTINE STORE
C STORE IS THE COMPLETION ROUTINE FOR DAT ACQ.
C THIS COMPLETION ROUTINE DOES THE FOLLOWING THINGS:
C 1: WRITES THE DATA TO
C THE DISK FILE
C 2: GRAPHS EACH CHANNEL AS A FUNCTION OF TIME ON THE
C ADM TERMINAL USING THE TK GRAPHICS SOFTWARE.
C 3: SLOW MODE WORKS AS FOLLOWS:
C A: MODE IS ACTIVE WHEN IFAST DOES NOT =1
C B: ONCE EVERY NCOUNT CALLS TO THE STORE ROUTINE
C ONE SET OF DATA IS SAVED AND STORED ON DISK
C AND PLOTTED ON THE ADM TERMINAL.
C 4: FAST MODE WORKS AS FOLLOWS:
C A: ACTIVATED BY IFAST=1
C B: DATA FROM EVERY CALL TO THE STORE ROUTINE IS
C STORED ON DISK.
C: ONCE EVERY NCOUNT CALLS A SET OF DATA IS PLOTTED
ON THE ADM TERMINAL IF THE GRAPHING IS TURNED ON.

COMMON/ACDDAT/FULSL,FRATE,INTCNT,ISTCHN,JRATE,NCHAN,PRESET,
$ RTIME,STIME
COMMON/ARRAYS/CONV(32),DESCRP(15,32),ISFLAG(15),
$ FDSCR(15,2),IRTCNT(6),JUNKR(5),RATES(6),SCANS(6),TIMSTR(2)
COMMON/CHAN/NLOAD,KCONF,KPARE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/PROCESS/IBUF(1000),NBUF,ISIZ,NCOUNT,
$ IFAST,ICHAN1,ICHAN2,J,NOGRAPH,IY0(32),IX(2),LEVENT,IRECD
COMMON/SCREEN/KDISP,ICOMP,ISIZ,ISIZ,IS1,IS2,GDISP,CONV,KDISP
COMMON/TEST/ACPS,AASX,AXLCON,CHIP,CONP,KCOM,KDISP
$ GAUXL,GAURAD,ICONFF,ICIRF,ICONF,ICONF,ICONF,ICONF,ICONF,ICONF
$ ICFlag,IFlag,LOAD,IPRED,IPRED,IPRED,IPRED,IPRED,IPRED
$ SLOAD,RLOAD,RSFLAG,SPDIA,SPDIAR,
$ SNAUL,SNCONF,SNDIFF,SNLOAD,SNPRED,SNRAIL,STRESS1

LOGICAL*! CAN,SUBJESC,FS,GS,US,DEL,LITTLA,FF
LOGICAL*! ISTRS(15)
LOGICAL*! LDIRS(15)
LOGICAL*! LABSTR(7)

DIMENSION KHA1(10)
DIMENSION IY(2),IXX(2)

DATA IXX/1295.4095/
DATA CAN,SUB,ESC,FS,GS,US,DEL,LITTLA/*30,32,33,34,35,37
$'177,'141/
DATA FF/'14/
DATA KHA1/*60,61,62,63,64,65,66,67,70,71/
DATA NCOUNT/1/
DATA LABSTR/1HS,1HT,1HR,1HE,1HS,1HS,1H=/

CALL STRESS AND CONFINING PRESS. CONTROL ROUTINE
CALL RUN

DO BOOKKEEPING FOR DIFFERENT DATA RATES
FAST OR SLOW MODE???
IF IFAST, WRITE ALL OF THE DATA TO DISK
IF IFAST, ONLY WRITE EVERY NCOUNT'TH DATA SET
to the disk
IF (IFAST.EQ.1) GO TO 99
IF (ICOUNT.GE.NCOUNT) GO TO 99
J=J+NCHAN
IF (J.GT.ISIZ) J=1
GO TO 125

99 CONTINUE

WRITE DATA TO DISC
J0=J
J=J+NCHAN
WRITE (ICHAN2) (IBUF(K),K=J0,J-1),LEVENT
IRECD=IRECD+1
LEVENT=0
IF (J.GT.ISIZ) J=1

GRAPH DATA IF NOGRAPH=0
IF (ICOUNT.GE.NCOUNT) GO TO 125
IF (NOGRAPH.EQ.1) GO TO 125
IX(1)=IX(2)
IX(2)=IX(2)+28
IF (IX(1).NE.1295) GO TO 110

DRAW GRID,WRITE LABEL
CALL GRIDVR
GO TO 110

CONTINUE

SCREEN DISPLAY OF INTERNAL LOAD CELL CORRECTED FOR TEMPERATURE
GO TO VECTOR
CALL TKPACK(GS)
APPENDIX (cont)

C POSITION CURSOR
CALL TKVECT (1690,3000)
C GO TO 4010 ALPHA MODE
CALL TKPACK(US)
C SET DATA LEVEL = WHITE
CALL TKPACK(ESC)
CALL TKPACK(LITTLE)
C WRITE VARIABLE TO GRAPHICS SCREEN
DO 114 IDUMMY=1,15
CALL TKPACK(ISTR(IDUMMY))
114 CONTINUE
CALL TKPACK(GS)
CALL TKVECT(1295,1560)
CALL TKPACK(CAN)
119 CONTINUE
DO 120 K=1,NCCHAN
IY(1)=IY0(K)
IY(2)=.76171875*FLOAT(IBUF(J0+K-1))+1560.
IF (IY(2).GT.3120) IY(2)=3120
IF (IY(2).LT.0) IY(2)=0
CALL TKPACK(GS)
CALL TKVECT(IY(1),IY(1))
CALL TKVECT(IY(2),IY(2))
IF (IX(1).NE.1295) GO TO 111
K2=K+ISTCHN
IF(K2.GT.10) K2=K2-10
IF (K2.GT.10) K2=K2-10
IF (K2.GT.10) K2=K2-10
CALL TKPACK(US)
KCHAR(K2)=KCHAR(K2).AND.\'177
CALL TKPACK(KCHAR(K2))
111 IY0(K)=IY(2)
CALL TKPACK(CAN)
120 CONTINUE
IF (IX(2).GE.4095) IX(2)=1295
C C EXIT THE SUBROUTINE
C 125 CONTINUE
ICOUNT=ICOUNT+1
IF (ICOUNT.GT.NCOUNT) ICOUNT=1
NBUF=NBUF+1
IF (NBUF.LE.0) TYPE 9999
9999 FORMAT (//' HELP!! DATA OVERRUN!!!'))
RETURN
C C 300 CONTINUE
APPENDIX (cont)

TYPE 301

FORMAT ('///,' BAD WRITE TO DISK!!!!!!!!!!!!!!')

IF (J.GT.ISIZ) J=1

GO TO 125

C C C
C 310 CONTINUE
C CALL CLOSEC(ICHAN2)
C STOP ' END OF FILE REACHED!!'
C END

SUBROUTINE SYSINT(IFLAG)
C SYSINT INITIALIZES SYSTEM, ALL VALVES, IFLAG
IFLAG=0
IDRCSR='167770
IDROUT='167772
C DISABLE I/O CARD INTERRUPT CAPABILITY
CALL IPOKE(IDRCSR,0)
C FIRST OPEN ALL VALVES EXCEPT VALVE 0
CALL IPOKE(IDROUT,'177776)
RETURN
END

SUBROUTINE VALOFF(IVALUE)
C VALOFF DE-ENERGIZES VALVE 'IVALUE', (0-15)
IDROUT='167772
IBIT=2**IVALUE
CALL IPOKE(IDROUT,IBIT.OR.IPEEK(IDROUT))
RETURN
END

SUBROUTINE VALON(IVALUE)
C VALON ENERGIZES VALVE 'IVALUE', (0-15)
IDROUT='167772
IBIT=2**IVALUE
IBIT=.NOT.IBIT
CALL IPOKE(IDROUT,IBIT.AND.IPEEK(IDROUT))
RETURN
END

C

SUBROUTINE WTACQ(ILUN)
C WTACQ WRITES FULSCL,FRATE,INTCNT,ISTCHN,NCHAN,PRESET,
C STTIME TO DISC
COMMON/ACQDAT/FULSCL,FRATE,INTCNT,ISTCHN,JRATE,NCHAN,PRESET,
* RTIME-STTIME
WRITE (ILUN) FULSCL,FRATE,INTCNT,ISTCHN,JRATE,NCHAN,PRESET,
* RTIME-STTIME
RETURN
END
SUBROUTINE UTARAY(ILUN)
C UTARAY WRITES ARRAYS CONV, IRTCNT, RATES, SCANS, TIMSTR TO DISC
COMMON/ARRAYS/CONV(32), DSCRPT(15,32), DSFLAG(15),
* FDSCR(15,2), IRTCNT(6), JUNKR(5), RATES(6), SCANS(6), TIMSTR(2)
C
C WRITE CONV
WRITE (ILUN) (CONV(IDUMMY), IDUMMY=1,6)
C WRITE IRTCNT
WRITE (ILUN) (IRTCNT(IDUMMY), IDUMMY=1,6)
C WRITE RATES
WRITE (ILUN) (RATES(IDUMMY), IDUMMY=1,6)
C WRITE SCANS
WRITE (ILUN) (SCANS(IDUMMY), IDUMMY=1,6)
C WRITE TIMSTR
WRITE (ILUN) TIMSTR
RETURN
END

SUBROUTINE WTHEAD(ILUN)
C WTHEAD WRITES HEADER TO DISC
COMMON/ARRAYS/CONV(32), DSCRPT(15,32), DSFLAG(15),
* FDSCR(15,2), IRTCNT(6), JUNKR(5), RATES(6), SCANS(6), TIMSTR(2)
COMMON/ACQDAT/FULSCL, FRATE, IRTCNT, ISTCHN, JRATE, NCHAN, PRESET,
* RTIME, STTIME
C
C WRITE FIRST FILE DESCRIPTION LINE
WRITE (ILUN) (FDSCR(IDUMMY,1), IDUMMY=1,15)
C WRITE SECOND FILE DESCRIPTION LINE
WRITE (ILUN) (FDSCR(IDUMMY,2), IDUMMY=1,15)
C WRITE NO. CHANS+FLAG
C  STORE NCHAN = NO. CHANNELS + 1 EVENT FLAG
NCHAN = NCHAN +1
WRITE (ILUN) NCHAN
C CORRECT NCHAN
NCHAN = NCHAN -1
C WRITE '2' TO SIGNIFY TYPE 2 FORMAT FILE
NCHAN = 2
WRITE (ILUN) IDUMMY
C WRITE FASTEST RATE + 5 SLOWER RATES
WRITE (ILUN) (FRATE, RATES(IDUMMY), IDUMMY=1,5)
C WRITE STARTING CHANNEL NO., A/D FULLSCALE
WRITE (ILUN) ISTCHN, FULSCL
DO 385 IDUMMY = 1, NCHAN
C WRITE CHANNEL DESCRIPTION
WRITE (ILUN) (DSCRPT(I, IDUMMY), I=1,15)
C WRITE CHANNEL CONVERSION FACTOR
WRITE (ILUN) CONV(IDUMMY)
385 CONTINUE
C WRITE EVENT FLAG DESCRIPTION
WRITE (ILUN) (DSFLAG(IDUMMY), IDUMMY=1,15)
C WRITE CONVERSION FACTOR FOR FLAG CHANNEL
IDUMMY = 1/FULSCL
WRITE (ILUN) DUMMY
RETURN
END
APPENDIX (cont)

SUBROUTINE AXSTRS
C AXSTRS CALC. THE PRESENT DIFFERENTIAL STRESS
C REMEMBER: THE LOAD CELL'S OUTPUT IS THE DIFFERENCE BETWEEN
C THE LOAD AND CONFINING PRESSURE
C ENTER AREANW-SPEC.AREA(IN.)EXIT STRSS1-AXIAL STRESS (PSI)
C
COMMON/PROCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,IN,IX(2),IX,LEVENT,IRECRD
C
COMMON/TEST/ACPSI,AREANW,AXLCON,CIRREF,CONDNSR,CPERNT,DIFDSR,
* GAUXL,GAURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF1,
* ICFLAG,IFLAG,LOADF,LOADI,IPORE,IPOREF,PERCNT,PI,PISARE,
* RADCON,RLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAAXL,SNCONF,SNDIFF,SNLOAD,SNPARE,SNRADL,STRSS1
C
LEN=LOAD/AREANW
RETURN
END

SUBROUTINE WTRSTR(ILUN)
C WTRSTR WRITES RESTART FILE
COMMON/ACQDAT/FULLSCL,FRATE,INTCNT,ISTCHN,ISTCHN,FRATE,NCHAN,PRES,
* RTIME,STTIME
COMMON/ARRAYS/CONV(32),DSR(15,32),DSFLAG(15),
* FDSR(15,2),INTCNT(4),JUNKR(3),RATES(6),SCANS(6),TIMSTR(2),
COMMON/CHAN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/POPFLG/IPOP
COMMON/PRCES/IBUF(1000),NBUF,ISIZ,NCOUNT,
* IFAST,ICHAN1,ICHAN2,J,IN,IX(2),IX,LEVENT,IRECRD
COMMON/SCREEN/KDISP,CLPI,LOADF,IDAT,INT,TIM,CONST1,CONST2,CONST3
COMMON/TEST/ACPSI,AREANW,AXLCON,CIRREF,CONDNSR,CPERNT,DIFDSR,
* GAUXL,GAURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF1,
* ICFLAG,IFLAG,LOADF,LOADI,IPHERE,IPOREF,PERCNT,PI,PISARE,
* RADCON,RLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAAXL,SNCONF,SNDIFF,SNLOAD,SNPARE,SNRADL,STRSS1
CALL WTHREAD(ILUN)
CALL WARRAY(ILUN)
CALL WTCO(ILUN)
CALL WTPAR(ILUN)
RETURN
END

SUBROUTINE WTPAR(ILUN)
C WTPAR WRITES TEST PARAMETERS TO DISC
COMMON/CHAN/KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
COMMON/TEST/ACPSI,AREANW,AXLCON,CIRREF,CONDNSR,CPERNT,DIFDSR,
* GAUXL,GAURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF1,
* ICFLAG,IFLAG,LOADF,LOADI,IPHERE,IPOREF,PERCNT,PI,PISARE,
* RADCON,RLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAAXL,SNCONF,SNDIFF,SNLOAD,SNPARE,SNRADL,STRSS1
C
WRITE (ILUN) ACPSI,AREANW,AXLCON,CIRREF,CONDNSR,PERNT,DIFDSR,
* GAUXL,GAURAD,ICONFP,ICIRI,ICIRF,ICONFI,ICONFF,IDIFF1,
* ICFLAG,IFLAG,LOADF,LOADI,IPHERE,IPOREF,PERCNT,PI,PISARE,
* RADCON,RLOAD,RSFLAG,SPDIA,SPDIAR,
* SNAAXL,SNCONF,SNDIFF,SNLOAD,SNPARE,SNRADL,STRSS1
WRITE (ILUN) KLOAD,KCONF,KPORE,KDIFF,KRAD1,KRAD2,KRAD3
RETURN
END