AN ABSTRACT OF THE THESIS OF

Richard Prendergast Beyer, Jr. for the degree of Master of Science in the Chemical Engineering Department presented on April 2, 1982

Title: Automation of a low-temperature calorimeter

Redacted for Privacy

Abstract approved: ______________________________

Robert V. Mrazek

An adiabatic calorimeter operating over the range 5 to 300 K has been automated. Instrumentation includes computer controlled data acquisition, a low-thermal emf multiplexer for switching low-level dc signals, and automatic analog shield controls. Temperature measurements are made with platinum and germanium resistance thermometers, and the thermometer resistances are determined with a programmable dc voltage source and a digital nanovoltmeter. Heat capacity measurements of benzoic acid agree to within ± 0.1 per cent at 300 K with accepted values. Accuracy is equal to the manual method from 5 to 50 K, and greater than the manual method from 50 to 300 K.

Automation of a Low-Temperature
Calorimeter

by

Richard P. Beyer, Jr.

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed April 2, 1982

Commencement June 1982

APPROVED:

Redacted for Privacy

Professor of Chemical Engineering in charge of major

Redacted for Privacy

Head of Department of Chemical Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 2, 1982

Typed by Richard P. Beyer, Jr.

Acknowledgement

I would like to give special thanks to my advisor, Dr. Robert V. Mrazek, for his kind help and encouragement throughout this project. Also, thanks to John Stuve for initiating me into the secrets of low-temperature calorimetry.

In addition, I thank Dr. Nev Gokcen for his encouragement and Howard Poppleton for his support. I am also grateful to the Bureau of Mines for providing the financial means necessary for carrying out this research.

Most important, I give my deepest thanks to Catharine H. Beyer, whose love and care and frequent prodding made it all possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PLATES

Automation of a Low-Temperature Calorimeter

I. Introduction

Thermodynamics is making an ever larger impact on science and engineering. As resources become scarcer and energy supplies dwindle, the need to make existing chemical processes more efficient and productive gains priority. Also, the feasibility of new processes, such as the recovery of minerals in very low-grade ore deposits, can be judged more confidently with accurate thermodynamic analysis. However, the effectiveness of thermodynamics relies on accurate data. The best way of generating thermodynamic data is experimental measurement of the property in question. One such approach is presented here.

The thermodynamic property studied in the present work was the heat capacity of condensed phases over the temperature range 5 to 300 K. Figure 1 shows an idealized calorimeter used for heat capacity measurements. The calorimeter consists of a sample holder, a thermometer, and an electrical heater. This in turn is surrounded by a shield, and the whole assembly is suspended in a cryostat that is maintained at high vacuum. For heat capacity measurements over the 5 to 300 K range, two methods are commonly employed.

In the adiabatic method, the shield temperature is regulated so it is as nearly identical to the calorimeter temperature as possible. The heat capacity is determined from (see Westrum et al. (1))

$$C_p = (iv\Delta t)/(m\Delta T) \qquad (1)$$

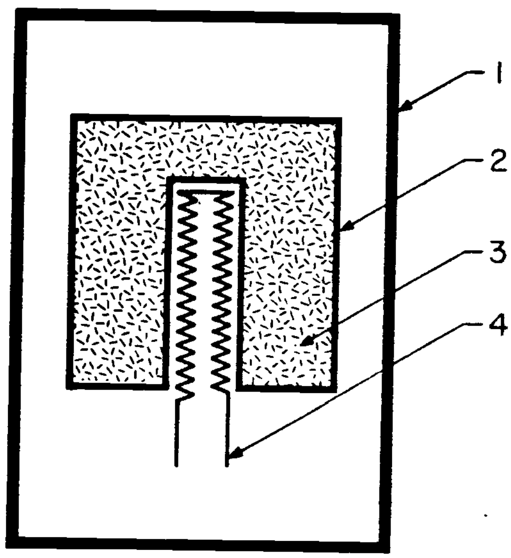

FIGURE 1. Idealized calorimeter: 1) adiabatic shield, 2) calorimeter (sample holder), 3) sample, 4) thermometer and heater.

where C_p = heat capacity, J K^{-1} mol^{-1}

i = current to heater, amperes

v = voltage across heater, volts

Δt = time heater is on, seconds

ΔT = temperature rise, K

m = mass of sample, mol

The temperature of the sample is measured, the heater is turned on for Δt seconds, and the heater current and voltage are measured. The heater is turned off, and the sample temperature is measured again. All the energy input remains in the sample, because the adiabatic shield temperature tracks the sample temperature, thereby eliminating any heat loss.

The second method for heat capacity measurements over the temperature range 5 to 300 K is the isothermal method. In an isothermal calorimeter, the shield is massive and is held at constant temperature throughout the heating cycle. Knowledge of the heat leak and the other variables in equation (1) allows the C_p to be calculated. In the adiabatic method the heat leak to the surroundings is made negligibily small; in the isothermal method, the heat leak is much larger but is a known quantity. Both methods give comparable accuracies.

The inaccuracies of low-temperature adiabatic calorimetry arise mainly from measuring the various voltages in the circuit. The heater current and voltage and the temperature of the sample need to be measured as accurately as possible. In the past, these voltage measurements have been done with a manually operated potentiometer. Using this type of potentiometer, a person measuring the heat capacity has to tediously log the data for up to two or three weeks. With the advent of programmable instruments with accuracy and stability comparable to the manually operated potentiometer, automation of the measurements becomes possible. In the present study, both the data logging and the experiment control were automated by using programmable instruments and a minicomputer.

Automated calorimeters using both ac bridges and dc potentiometers have been described previously (2-6). Even though the dc potentiometric method described here is less accurate than the ones using ac bridges, dc methods are less expensive and can be used in both the temperature measurement and the heater energy measurements.

Automation offers many advantages over manual measurements. The operator is freed from the repetitive task of voltage measurements and data logging , therefore possible human errors are eliminated. Repeated measurements become more consistent and, therefore, more reproducible. And finally, automatic calorimeters can be run around-the-clock and unattended, which increases the number of compounds that can be run.

II. Apparatus

A. Cryostat and calorimeter

The mechanical equipment of the low-temperature adiabatic calorimeter consists of a cryostat, calorimeter, diffusion pump and vacuum pumps. This mechanical apparatus has been described previously by Stuve et al. (7). Only the modifications will be described here.

The equipment suspended inside the cryostat is shown in Plate 1. This includes the liquid cryogen tanks, shielding, and calorimeter. The calorimeter, or sample holder, is shown in Plate 2. It is constructed of high-purity oxygen-free copper and is gold-plated inside and out. The outer gold plating serves to reduce radiant energy exchange with the surrounding shields, and the inner gold-plating protects the copper from chemical reaction with the sample. A central well holds the heater winding tube and the platinum resistance thermometer. The platinum resistance thermometer (25 ohms at 273.15 K) is a capsule type Leeds & Northrup Model 8164, calibrated by the National Bureau of Standards according to the International Practical Temperature Scale of 1968 (IPTS-68). Another smaller well holds the germanium resistance thermometer. The germanium resistance thermometer (1200 ohms at 4 K) is manufactured by Solitron, Inc. and was calibrated according to the National Bureau of Standards Provisional Scale 2 to 20 K (1965). The thermometer cover, held in place by three 0-80 sized nuts, secures the platinum resistance thermometer inside the well.

There are six copper fins mounted inside the calorimeter which

PLATE 1. Equipment inside cryostat.

PLATE 2. Close-up of calorimeter.

increase heat conduction from the centrally located heater to the sample. The sealing plug at the top of the calorimeter is made of stainless steel and has a knife edge machined into the bottom. This knife edge and a thin gold gasket form a vacuum tight seal.

The heater is 4500-ohm wire winding on a copper tube. The wire is 0.061 mm diameter, is made of Pt-8 atomic per cent W, and is insulated with enamel. The heater has a coating of General Electric type 7031 varnish on the copper tube for added electrical insulation and also a similar coating on top of the winding for mechanical protection of the wire. The heater and thermometers are also coated with a thin layer of Apiezon T grease which helps insure good thermal contact with the calorimeter.

As shown in Plate 2, the wires from the heater and thermometers, as well as those from the central-shield control thermocouple, are wound around the outside of the calorimeter and attached with nylon thread and G.E. type 7031 varnish. This wiring arrangement minimizes heat conduction along the wires during the heating period when the bottom of the calorimeter is hotter, due to the location of the heater, than the rest of the calorimeter. One leg of the differential thermocouple, which is connected between the calorimeter and the shield, is located near the bottom of the calorimeter. This position was experimentally determined to give the smallest heat loss due to conduction along the wire leads during heating periods. Another differential thermocouple is used to determine the temperature difference between the two ends of the wire bundle. One end is attached to the leads where they leave the calorimeter, and the other end is attached where the leads contact the main shield. This temperature difference measurement is then used to calculate any conduction that occurs along the leads during heating periods.

B. Instrumentation

1. Overview

An overall block diagram of the measurement circuitry is shown in Figure 2. The actual instrumentation is shown in Plate 3. The instrumentation is divided into four main groups:

1) the thermometer section, consisting of a platinum resistance thermometer, a germanium resistance thermometer, standard resistors, and a constant current source;
2) the heater section, consisting of a 4500 ohm resistance heater, standard resistor, and a programmable constant current source;
3) the voltage measurement section, consisting of a 3-1/2 digit nanovoltmeter, a 6-1/2 digit programmable voltmeter, and a programmable dc voltage source; and
4) the multiplexer section, consisting of an enclosed box containing single-pole latching relays.

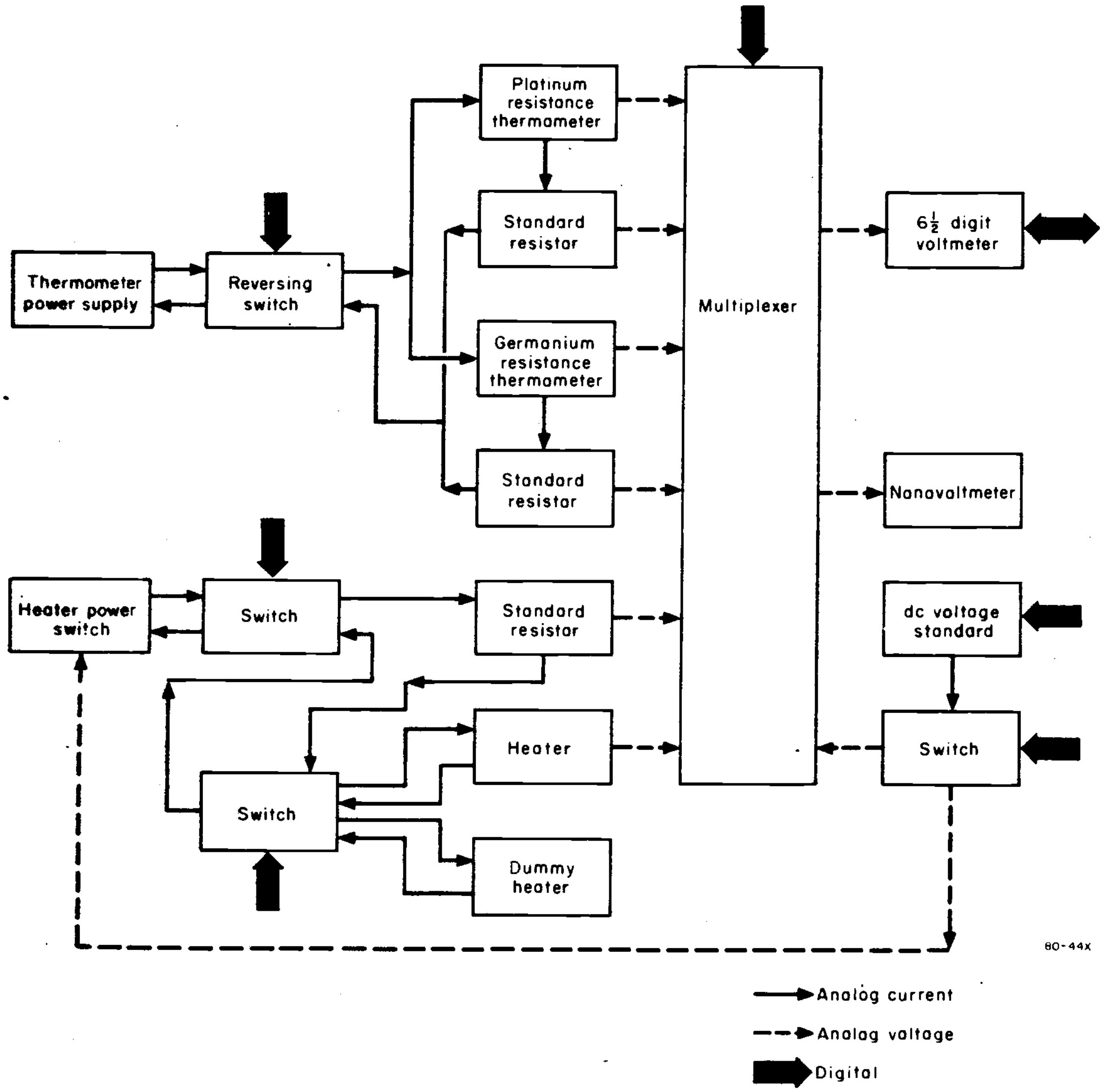

FIGURE 2. Block diagram of automated low-temperature calorimeter instrumentation. All switches, multiplexer, standard resistors, and thermometer power supply are in an insulated "box-within-a-box".

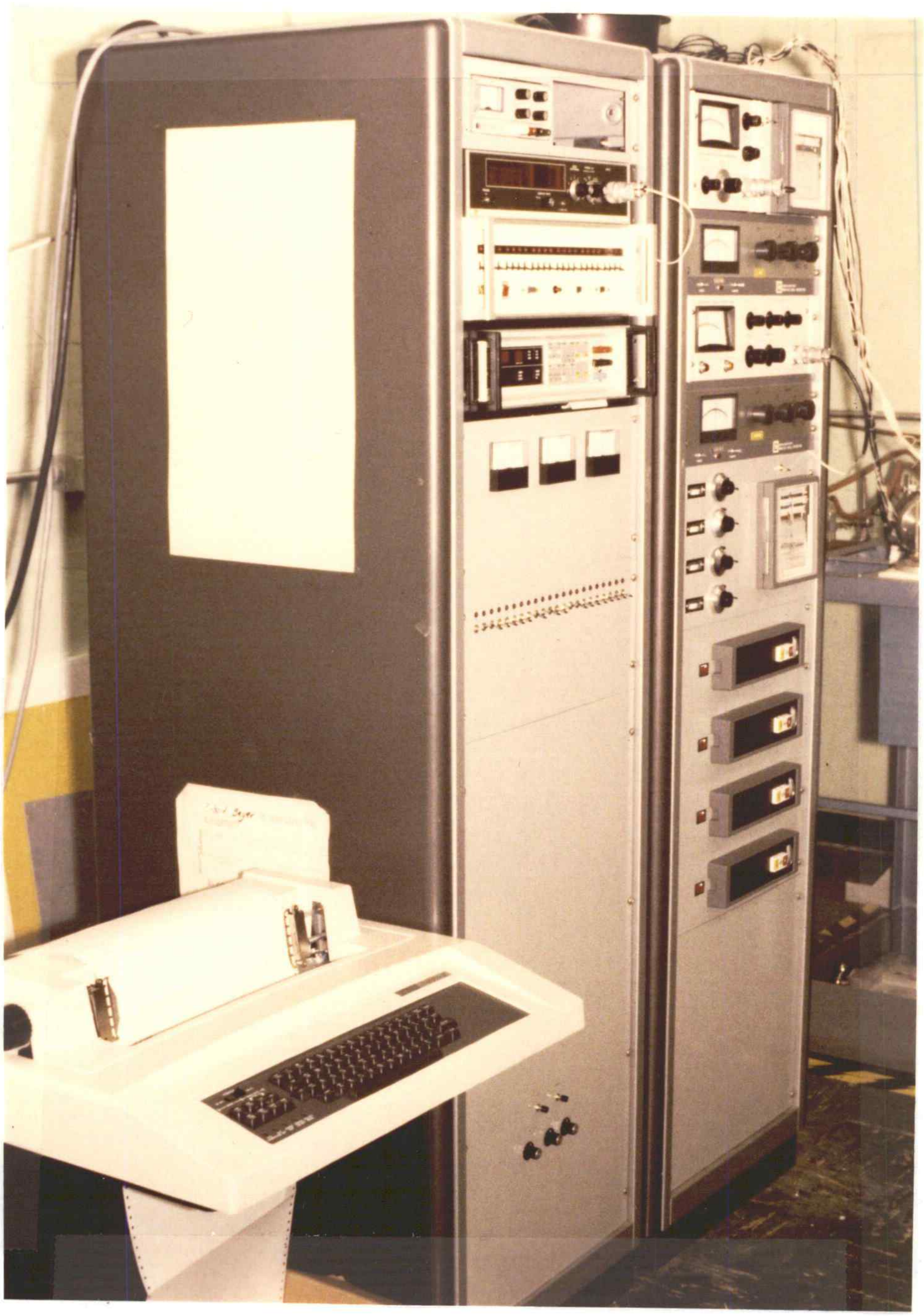

PLATE 3. Measurement instrumentation.

2. Shield Control

Also described previously by Stuve et al.(7) was the analog circuitry used to control the temperature of the adiabatic shields surrounding the calorimeter. The only modification was to use Leeds & Northrup CAT 80 proportional-integral-derivative controllers on all four shield sections: the top, middle, and bottom shields and also the tempering ring.

3. Thermometer Circuit

As shown in Figure 3, the thermometer circuit consists of a platinum resistance thermometer in series with a 100 ohm standard resistor, both of which are in parallel with a germanium resistance thermometer that is in series with a 10,000 ohm standard resistor. This thermometer ciruit is powered by a constant current source, shown in Figure 4, built according to the design given by Chang (8). This constant current source has a short term drift of 20 ppm/hour and a peak-to-peak noise of 5 ppm. The short term performance of the circiut is illustrated in Figure 5.

4. Heater Circuit

The heater circuit, shown in Figure 6, consists of a 4500-ohm resistance heater, 10 ohm standard resistor, a 4500-ohm dummy heater and a Hewlett-Packard model 6181B constant current source. This constant current source has an eight hour stability of less than (100

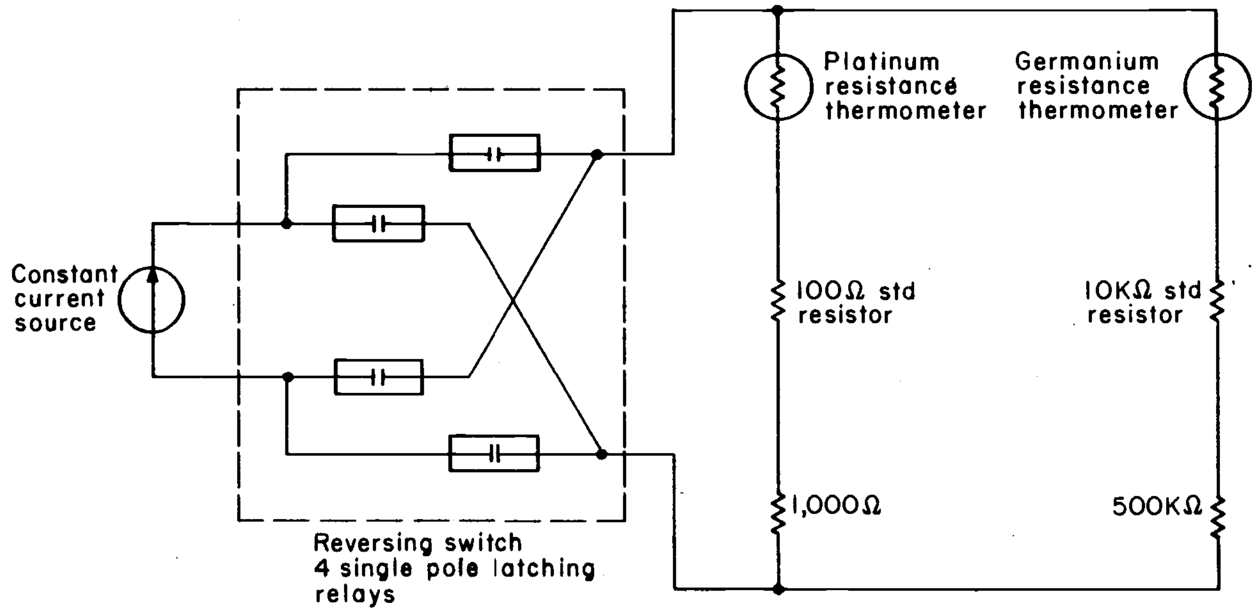

FIGURE 3. Thermometer circuit.

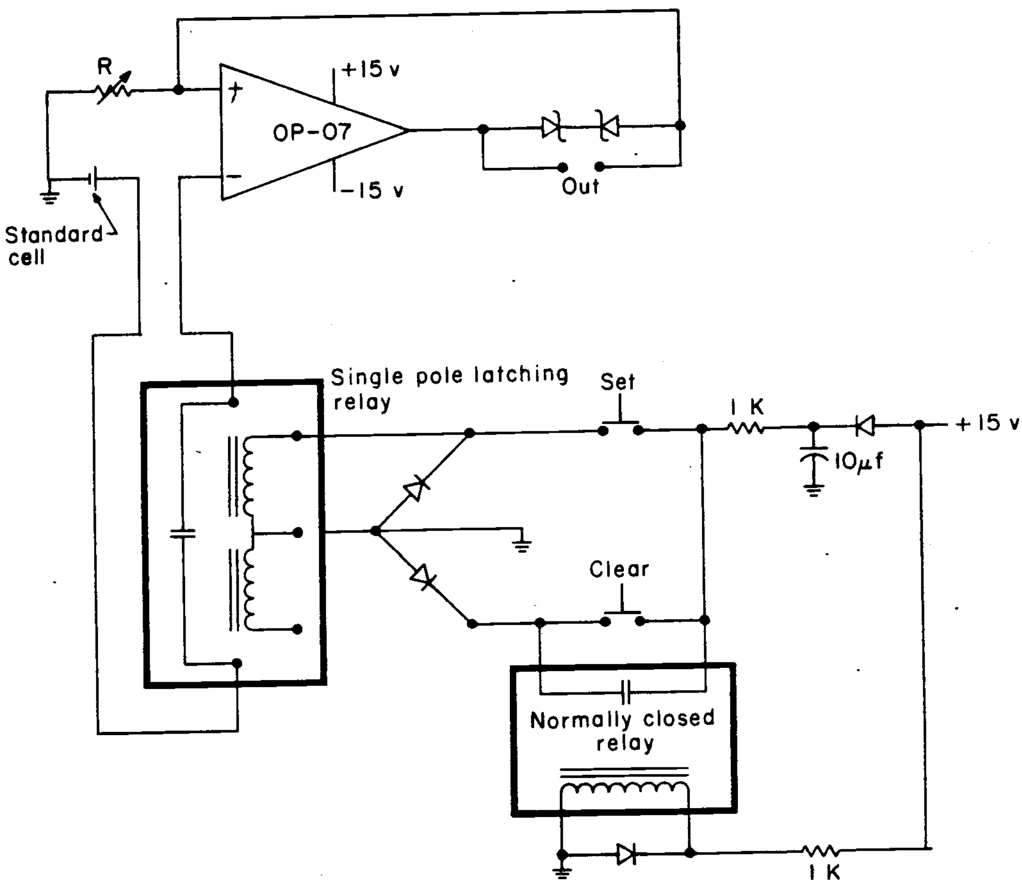

FIGURE 4. Thermometer constant-current source circuit.

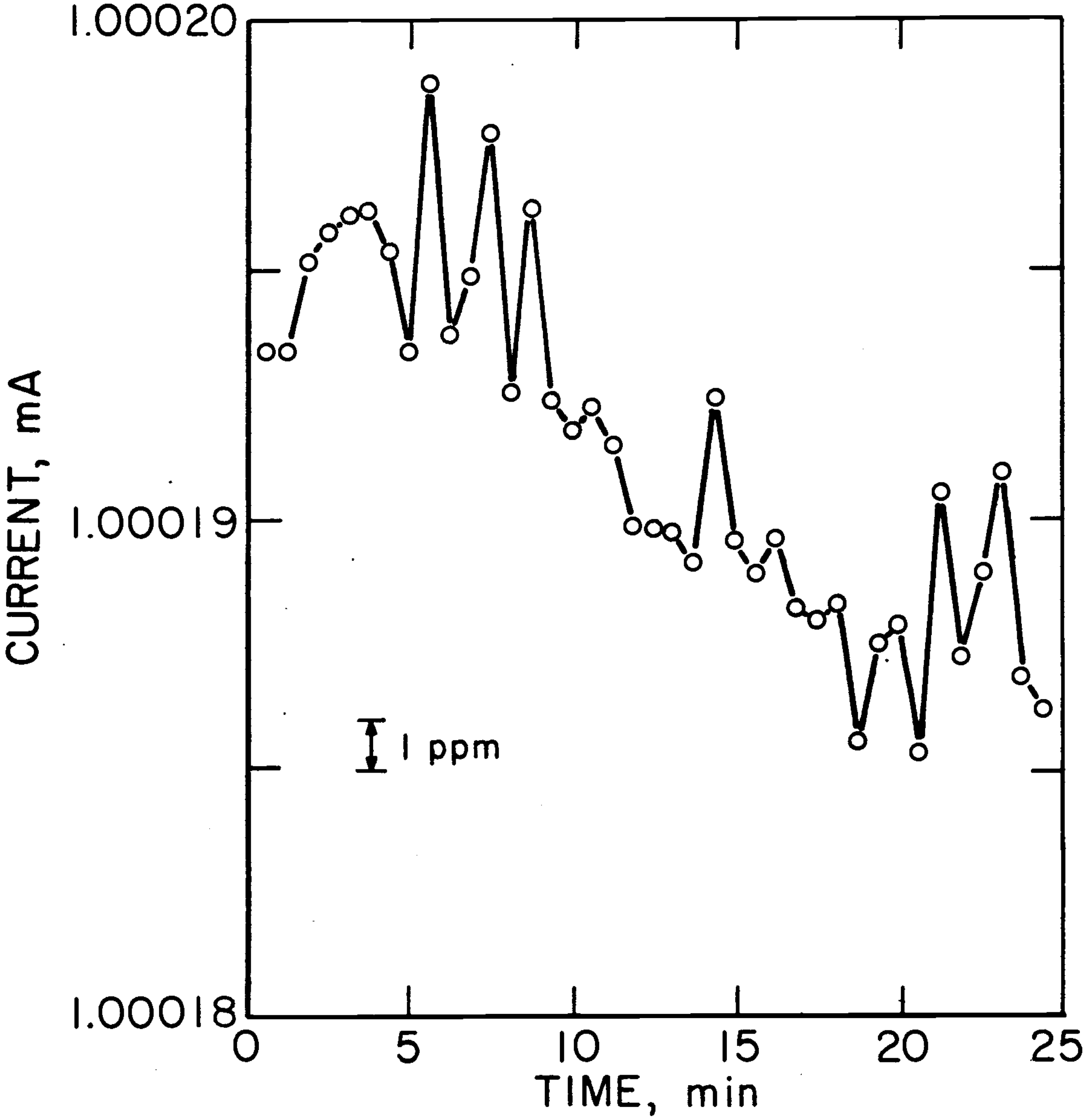

FIGURE 5. Thermometer constant-current source short-term performance.

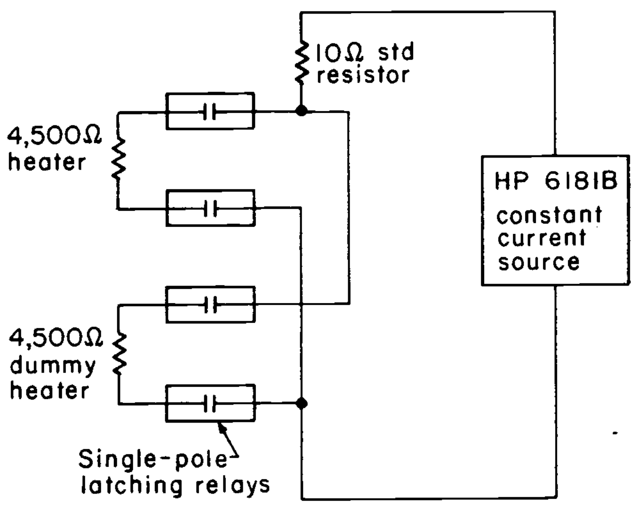

FIGURE 6. Heater circuit.

ppm output + 25 ppm of range), a temperature coefficient of less than (75 ppm of output + 5 ppm of range)/deg C, and a ripple and noise of 2 microamperes rms on the 25 mA range and 0.2 microamperes rms on the 2.5 mA range. The dummy heater is used to minimize the power supply's excursions away from its programmed value which occur during the switching of the heater. The power supply current is first routed through the dummy heater and allowed to settle to its programmed value. The dummy heater is then switched out of the circuit so the power supply current is open circuited. The power supply current is then switched to the heater for the energy input period. This break-before-make type switching reduces the power supply transients and has been explained by Chang (9).

5. Voltage Measurement

Voltage measurement is done with the following:

1) A Keithley Model 180 nanovoltmeter is used to measure the offset between the programmable dc voltage source and whichever resistor voltage is being measured. The manufacturer's specifications are:

a) 90 day accuracy of ±0.03 per cent of reading ±0.02 per cent of full scale;

b) temperature coefficient of less than ±0.01 per cent of reading per deg C;

c) zero drift of less than ±(30 nanovolts + 0.005

per cent of full scale) per deg C;

d) input noise of less than 30 nanovolts peak-to-peak with a source resistance of up to 1000 ohms;

e) an input resistance of greater than 30 megohms; an offset current of less than 100 picoamperes;

f) common-mode rejection ratio of greater than 160 dB at dc.

2) A Fluke Model 8500A digital voltmeter is used to measure the voltages in the heater circuit and measure the analog output of the Keithley model 180. The manufacturer's specifications are:

a) 90 day accuracy of ±(0.002 per cent + 8 digits);

b) temperature coefficient of ±(3ppm/reading+0.1 digit)/deg C;

c) input impedance of greater than 10^{10} ohms;

d) input bias current less than ±5 picoamperes;

e) common-mode rejection ratio of 100 dB.

3) An Electronic Development Corporation Model 501J Programmable DC Voltage Standard is the precision voltage source. This voltage source is used potentiometrically to offset the main portion of voltage across one of the thermometers or standard

resistors. This compares favorably to the Honeywell potentiometer used in the manual operation which has an accuracy of ±(0.005 per cent +0.025 microvolt). For an output of 100 mV the worst case for the EDC would be 100000.0±10.5 microvolts and for the Honeywell it would be 100000.0±10.1 microvolts. The manufacturer's specifications are:

a) worst case accuracy of ±(0.005 per cent of programmed value +0.0005 per cent of range + 5 microvolts);

b) 90 day stability of ±0.0025 per cent;

c) ripple and noise of 0.0005 per cent of range plus 5 microvolts rms over a band of 0.8 Hz to 100 kHz;

d) temperature coefficient of ±0.0005 per cent per deg C;

e) output impedance of 0.03 ohms on the 10.0 volt range and 3 ohms on the 100 mV range.

6. Multiplexer

All switching of the dc signals is done in a "box within a box" enclosure. The interior of the box containing the switches and the thermometer constant-current source is shown in Plate 4. The switching of the dc signals is done with single-pole latching relays (Coto-Coil type CR-3207-5-411) which are in turn switched by digitally controlled relay switches (Coto-Coil type CR-2042-5-1111).

The multiplexer consists of 32 of these single-pole latching relays, 16 four-pole relays, and 16 single-pole computer controlled relays. Printed circuit boards were fabricated to hold the various relays, and Plate 5 shows a close-up of one of the boards. Shown in Figure 7 is a diagram of one of these switching combinations. When the computer turns on the digitally controlled relay A, one of the 10-microfarad capacitors discharges through a normally open switch in relay A and sets the two single-pole latching relays. When the computer turns off relay A, the other 10 microfarad capacitor discharges through the other normally open switch in relay A which opens the two single-pole latching relays. These single-pole latching relays introduce less than a 400 nanovolt thermal emf into the signal and it is allowed to decay before a voltage measurement is made.

7. Computer

The computer used to run the heat capacity measurements was a Digital Equipment Corporation PDP 11/34 16-bit minicomputer with 124 K words of main MOS memory, 3 RL01 5-megabyte removable disk drives, a KW11-K programmable real-time clock, an IEC11-A IEEE-488 instrument bus controller, and a DRC-11 16-bit digital input-output card. The digital I/O card was used to control the multiplexer; the IEEE-488 controller card was used to program the voltage source and digital voltmeter. The IEEE 488 card was also used to input data from the digital voltmeter. The real-time clock was used to measure the time the heater was on . This clock has on accuracy of ± 0.01 per cent. The system clock, which

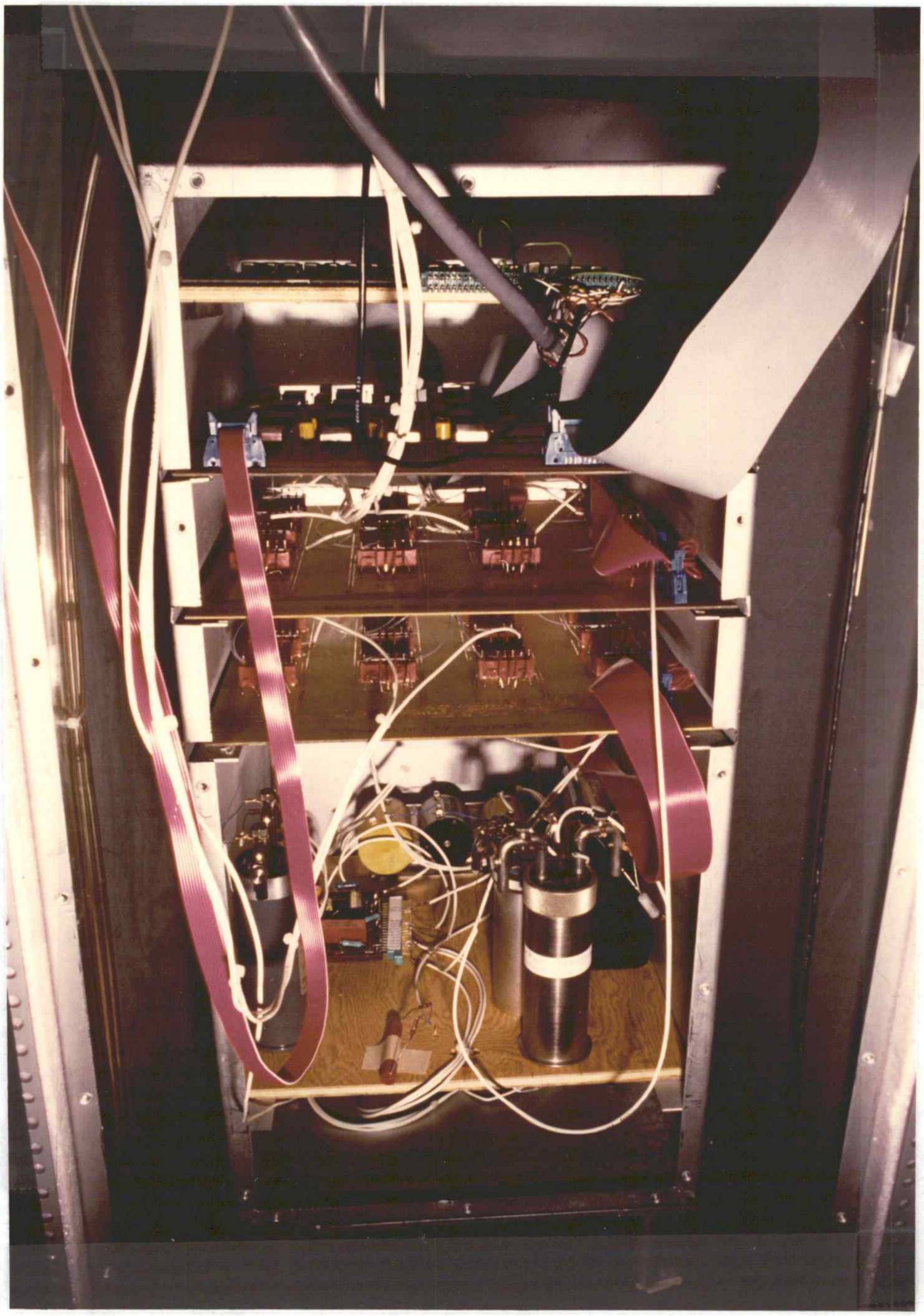

PLATE 4. Interior of multiplexer container.

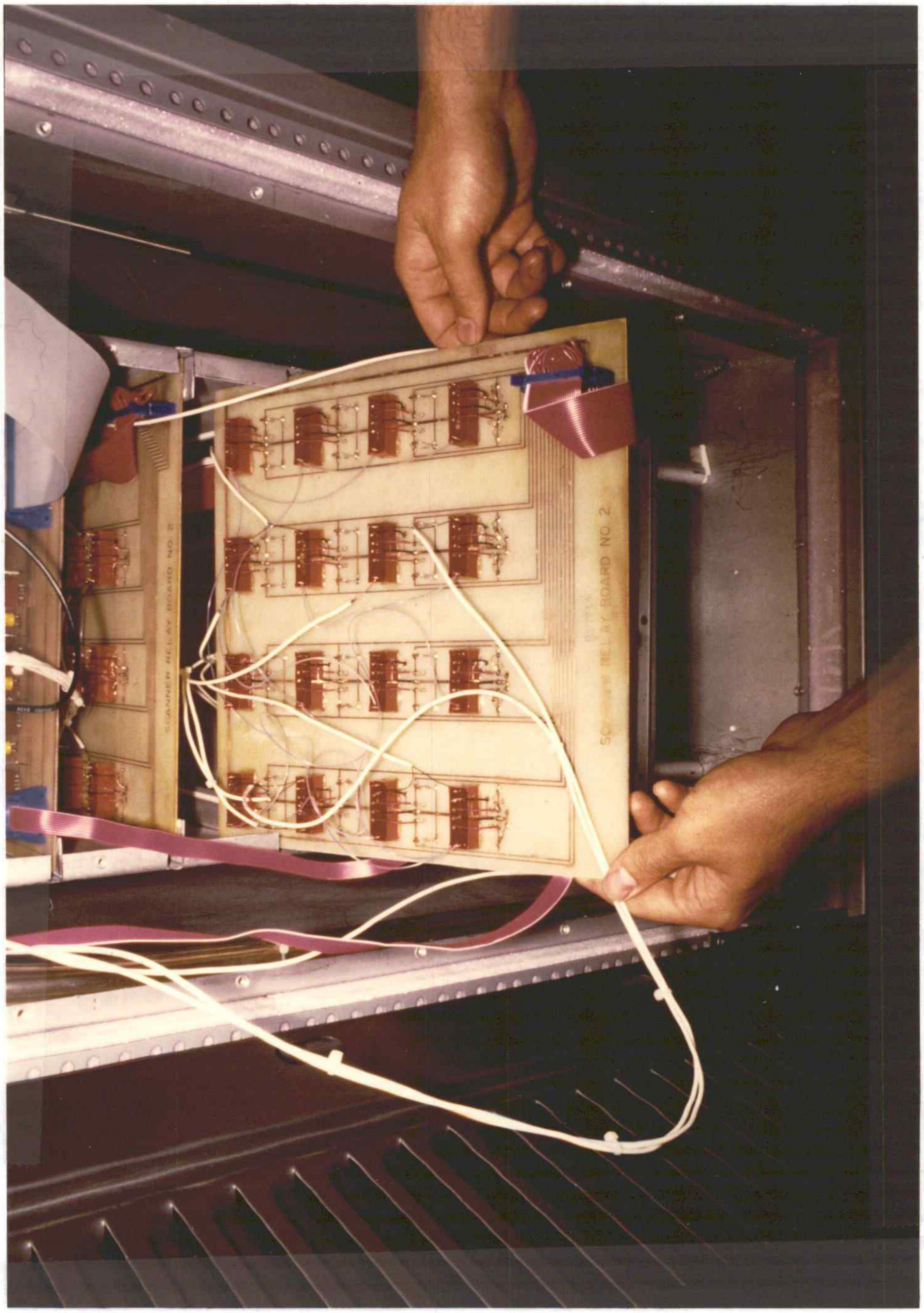

PLATE 5. Close-up of printed ciruit boards in multiplexer.

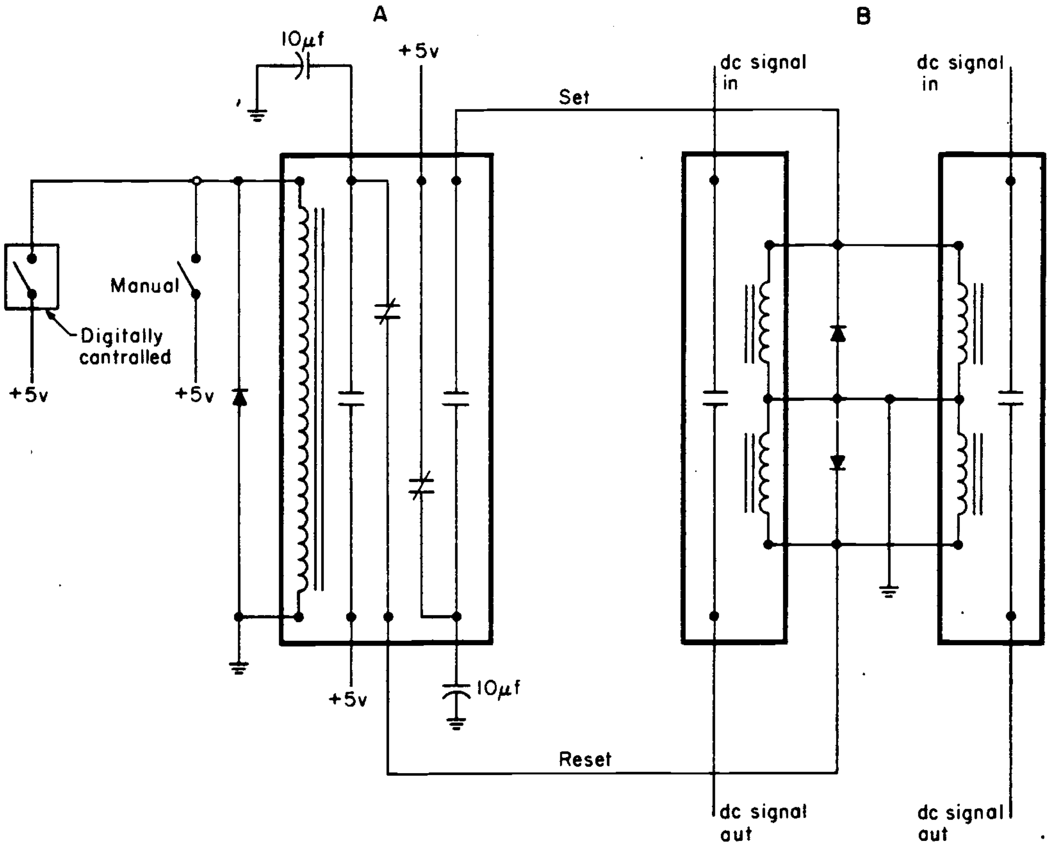

FIGURE 7. Schematic diagram of one switching element.

operates on the 60 Hz ac line frequency was used to measure the times between thermometer readings. The operating system used on the PDP 11/34 was RSX-11M Version 3.2 and all the programs were written in FORTRAN-IV Plus Version 2.5.

III. Procedure

A. Preparation

Before any heat capacity measurements were made, the following steps were taken:

1) the sample was loaded into the calorimeter and the calorimeter was placed into the cryostat;
2) a diffusion pump was used to obtain a pressure of 0.001 Pa;
3) the upper tank was filled with liquid nitrogen and the lower tank was filled with either liquid nitrogen for the 50 to 300 K range or with liquid helium for the 5 to 50 K range;
4) the calorimeter was cooled to the desired temperature; and
5) the shield controls were adjusted to adiabatic conditions.

B. Software

The overall scheme of the control and data aquisition algorithm is:

1) the temperature of the calorimeter is measured until equilibrium (explained in the temperature section below) is reached;
2) the heater is turned on;
3) the energy input to the calorimeter is measured;

4) the heater is turned off after a specified time;

5) the temperature of the calorimeter is measured until equilibrium is reached;

6) the temperature and the heat capacity are calculated and recorded; and

7) the entire cycle is repeated until a specified upper temperature is reached.

The control program is logically divided into 4 main groups: initialization, temperature measurement, energy input measurement, and heat capacity calculation. Actually three separate control programs are used for different temperature ranges, one for the 5 to 20 K range, one for the 15 to 60 K range, and one for the 50 to 310 K range.

1. Initialization

The program begins by asking for an initial value for the total C_p of the calorimeter and sample, in units of J K^{-1}. This C_p value is used to calculate the initial power input for the heater. A maximum time interval to reach equilibrium is also entered. This value is used to avoid excessively long equilibrium times if the temperature drift is too great during heater off times. Values for the number of moles of sample and the heat capacity correction for the helium exchange gas are read from a data file. Two ouput files are then opened. For each equilibrium period, one file receives the temperature, C_p, and various intermediate values used to calculate the C_p. The other file receives the time and temperature readings made during the equilibrium period.

These time and temperature values are used later, if necessary, to recalculate the temperature at the midpoint of the heating period. The calculations involved will be explained in the following section.

2. Temperature measurement

The temperature of the calorimeter is measured during the heater off time, or equilibrium time. Two arrays, each having 11 elements, are used to contain the time and temperature readings. All the elements in these two arrays are initially set to the first temperature-time measurement. The main program then enters a loop that maintains these two arrays as a moving record of the time and temperature. After each new measurement is made, the arrays are updated with the new values so that the last element of the arrays contains the latest readings and the first element of the arrays contains the oldest reading. The time interval between each temperature measurement is approximately 30 seconds.

After each new reading the main program checks to see if equilibrium has been reached. The time and temperature data are fit to a second order polynomial with a curve fitting routine that uses orthogonal polynomials of the type first described by Forsythe(10). Equilibrium occurs either when five sign changes of the second derivative of the polynomial take place or when the maximum time allowed to reach equilibrium has been reached. As long as neither of these two conditions are met, the program continues to update the time and temperature arrays.

The main calling program updates the time and temperature arrays with a call to subroutine RPt, or RGe, to measure the resistance of the platinum resistance thermomter (RPt) or the resistance of the germanium resistance thermometer (RGe). The thermometer resistance measurement subroutine consists of the 17 steps which follow.

1) The multiplexer switches are set so that the DVM measures the analog output of the nanovoltmeter with the nanovoltmeter's inputs shorted. This reading is called the zero offset and is used to correct subsequent measurements.

2) The multiplexer is set so that the DVM measures the voltage of the thermometer.

3) The dc voltage source is programmed so that its output will be the same as that measured in the previous step.

4) The system clock is read. This is the start of the time and temperature measurement cycle.

5) The multiplexer is set so that the nanovoltmeter measures the offset between the thermometer and the voltage source, and the DVM measures the analog output of the nanovoltmeter.

6) The actual voltage of the thermometer is calculated. This is the DVM reading of Step 2

plus the offset voltage of Step 5.

7) The voltage source is programmed to produce an an output of 99.9999mV which is approximately equal to the voltage across the 100-ohm standard resistor; if the germanium thermometer is being used this is the voltage across the 10 kilo-ohm standard resistor.

8) The multiplexer is set so that the nanovoltmeter measures the offset between the standard resistor and the voltage source.

9) The actual voltage across the standard resistor is calculated. This is equal to 0.0999999 volts plus the offset of Step 8.

10) The thermometer resistance is calculated. This is equal to the thermometer voltage divided by the thermometer current. The current is equal to the standard-resistor voltage divided by the known resistance of the standard resistor.

11) The multiplexer is set so that the direction of current flow from the thermometer-circuit constant current source is reversed.

12) Steps 5 through 10 are then repeated.

13) The system clock is read again to get the end time of the temperature-measurement cycle.

14) The resistance of the thermometer is

calculated. This equals one half the sum of the forward current resistance and the reverse-current resistance.

15) The time, measured in seconds since midnight, of the thermometer-resistance measurement is calculated. This equals one half of the sum of the times from Step 4 and Step 13.

16) The values of the forward current thermometer and standard resistor voltages, the reverse current standard resistor and thermometer voltages, and the thermometer resistance are printed on the terminal.

17) The subroutine then returns control to the main calling program with values for the thermometer resistance and the time of the measurements.

Once equilibrium has been reached, the time and temperature data are fit to a linear equation, which will be used in the C_p section to extrapolate to the mid-heating time; the data are also plotted on the terminal. The energy-input section of the program is then entered.

3. Energy Input

The energy input section begins by measuring the temperature difference between the two ends of the wire bundle that is attached to the calorimeter and the center shield. This temperature difference is calculated from the emf of a differential copper-constantan thermocouple. This temperature difference is also measured during and after the heating cycle. The initial and final temperature differences are used as a baseline for calculating an average heat loss from conduction during the heating cycle. If the program is executing the heating cycle for the first time, the energy input subroutine, named HEATER, is called. If the program is executing subsequent heating cycles, the C_p of the sample is calculated. This calculation will be explained in a later section.

Before the heater subroutine is called the main program calculates the power input needed for the particular heating cycle. This power input calculation is based on the rule of thumb which says that the temperature rise for a particular run should be approximately 10 per cent of the absolute temperature, but not more than 10 K. So at 20 K, a 2K temperature rise is desired, and at 200 K a 10 K temperature rise is desired. This approach permits both the characterization of the C_p versus T curve with a minimum of points and the detection of any thermal anomalies that may need closer scrutiny. Once the temperature rise is calculated, the heater constant current source setting is calculated. This setting is based on having the heater on for 10 minutes for operating temperatures above 50 K, and for 5 minutes if operating below 50 K. The current is calculated according to:

$$[(C_{ptotal})(\Delta T)/(\Delta t)(R_{heater})]^{1/2} \qquad (2)$$

where C_{ptotal} = the C_p of the sample plus the C_p of the empty calorimeter, J K^{-1}

ΔT = desired temperature rise, K

Δt = time heater is on , sec

R_{heater} = resistance of the heater, ohms

The resistance of the heater is temperature dependent and is approximated simply by adding the resistance at 5 K to the numerical value of the absolute temperature. The resistance at 5 K is 4500 ohms, so the resistance at 300 K would be approximately 4500 + 300 = 4800 ohms.

Once the current is calculated the subroutine HEATER is called. This subroutine consists of fourteen steps.

1) The multiplexer is set so that the heater constant-current source can be programmed by the voltage source, the dummy heater is on, and the DVM can measure the voltage across the heater standard resistor.

2) The voltage source is programmed to give the heater constant-current source the desired output.

3) The heater standard-resistor voltage is measured, and the current is calculated as a check to make sure the the current is not too large.

4) The current that is calculated in Step 3

determines which range on the DVM is selected, and the range that will give the highest resolution is chosen.

5) The multiplexer is set so that the DVM can measure the voltage across the heater.

6) The voltage across the heater with no current flowing is measured, and this voltage is called the zero offset and is subtracted from subsequent measurements.

7) The heater start time is read from the system clock, and the programmable clock is started.

8) The multiplexer is set so that the current now flows through the heater, and the DVM can measure the voltage across the standard resistor.

9) The DVM is programmed for maximum resolution, one microvolt; the standard resistor voltage is measured twice, and the readings are averaged.

10) The DVM is programmed for the resolution calculated in Step 4, and the heater voltage is measured 10 times and averaged.

11) The DVM is programmed for one microvolt resolution, and the differential themocouple voltage is measured.

12) The measurements of Steps 9-11 are printed on

the terminal.

13) The system clock is read and a check is made to see if the 10 minute heating period is over. If 10 minutes has not elapsed, the subroutine returns to Step 8. Otherwise, the programmable clock is stopped and read, and the heater is turned off.

14) The overall average of the heater current, heater voltage, elapsed time, and thermocouple voltage are printed on the terminal, and control is returned to the main calling program.

4. Heat capacity calculation

The calculation section consists of nine steps.

1) After the HEATER subroutine is completed, the energy input to the calorimeter is calculated. This is equal to the product of the heater current, the heater voltage, and the time of heating.

2) The midpoint of the heating time is calculated, and, with the linear coefficients calculated earlier in the last part of the temperature measurement section, the temperature at the heating mid time is calculated.

3) The program returns to the temperature measurement section where execution continues until equilibrium is reached. The time and temperature data are fit to a linear equation and the differential thermocouple is measured.

4) The average themocouple offset is calculated. This is equal to the average value measured during the heating period, minus the average of the initial and final measurements.

5) The coefficients from Step 3, above, are used to calculate the temperature at the mid-point of the heating period. The temperature rise of the heating cycle is the difference between this temperature and the one calculated in Step 3 above.

6) The total C_p is calculated as the total energy input from Step 1 divided by the temperature rise from Step 5.

7) The total C_p is then corrected to give the C_p of the sample. The corrections include subtracting the C_p of the helium exchange gas, the C_p of the empty calorimeter, and accounting for the heat leak during the heating period. The heat leak calculation is given in Appendix A.

8) The C_p and T values and their corresponding intermediate values are transferred to disk for later processing.

9) If the temperature of the calorimeter is greater than 303 K, the program stops; otherwise the entire measurement process is repeated.

C. Data Reduction

After the heat capacity data have been collected and curve fit, the C_p values are corrected for curvature by following the procedure described by Westrum et al. (1)

$$C_p = \langle C_p \rangle - (\Delta T^2/24)(d^2C_p/dT^2) \quad (2)$$

where $\langle C_p \rangle$ = measured (or mean) heat capcity

ΔT = temperature change of the C_p measurement

d^2C_p/dT^2 = second derivative of smoothed mean C_p curve, which for this small correction is assumed to be the true heat capacity curve.

The C_p versus T values are now in their final form and ready for publication.

IV. Results

The performance of the calorimeter was tested by measuring the heat capacity of benzoic acid, National Bureau of Standards standard reference material number 39i. The calorimeter was loaded with 64.21437 g of benzoic acid, 0.52583 moles, and backfilled with 6.7 kPa of helium. The first step was to measure the C_p of the empty capsule; this C_p varies from 0.26 J K^{-1} at 5 K to 38.37 J K^{-1} at 300 K . The C_p of the empty capsule was fit with three separate polynomials, and these were used to calculate the empty capsule C_p contribution during program execution.

The experimental C_p of benzoic acid is listed chronologically in Table 1. Listed in Appendix C are the intermediate values recorded during a given heating cycle. If there is a bad C_p value, these intermediate values help to locate where the error occured. In order to compare the heat capacity of benzoic acid to the accepted standard values from Furakawa et al. (11), a plot of heat capacity deviations is given in Figure 8. Also included for comparison are the values obtained from the manually operated calorimeter. Table 2 lists the smoothed values of C_p, S(T)-S(0), -[G(T)-H(T)]/T, and H(T)-H(0).

As can be seen from Table 1, there were repeated measurements made over the 200 to 300 K region. Each separate series was fit with a fourth order polynomial and the values at 273.15 K compared. The C_p at 273.15 K was (135.212±0.057) J K^{-1} mol^{-1} where 0.057 is twice the standard deviation of the single values. Therefore, the precision of the C_p values at 273.15 K is ±0.043 per cent.

Table 1. Experimental heat capacities of benzoic acid.

T K	C_p J mol^{-1} K^{-1}	T K	C_p J mol^{-1} K^{-1}	T K	C_p J mol^{-1} K^{-1}
210.37	106.913	229.11	115.061	12.01	3.418
219.84	111.041	238.17	119.105	12.92	4.122
229.07	115.011	247.04	123.007	13.91	4.925
238.09	119.075	255.74	126.963	14.99	5.876
246.92	122.921	264.26	130.848	16.04	6.839
255.58	126.820	272.61	134.659	16.96	7.746
264.05	130.685	280.80	138.428	17.79	8.587
272.37	134.599	288.86	142.173	18.54	9.329
280.55	138.389	296.79	145.733	19.39	10.376
288.57	142.068	206.99	105.348	20.26	11.248
296.46	145.652	216.39	109.499	21.06	12.177
304.22	149.326	225.76	113.516	22.60	13.829
158.21	85.697	234.89	117.761	24.77	16.138
168.45	89.641	243.82	121.569	26.92	18.246
178.66	93.728	252.56	125.391	29.37	21.186
188.90	97.901	261.12	129.362	32.08	24.108
198.99	102.070	269.50	133.306	34.93	26.745
208.82	106.253	277.73	136.993	38.07	29.653
218.37	110.430	285.82	140.613	41.52	32.535
227.66	114.434	293.77	144.340	45.30	35.290
236.74	118.507	301.59	148.022	49.43	38.276
245.62	122.411	4.51	0.187	53.99	41.979
254.31	126.288	4.76	0.029	64.83	48.409
262.82	130.070	5.11	0.198	70.95	51.488
271.17	133.941	5.59	0.189	77.43	54.461
279.36	137.695	5.87	0.223	84.51	57.507
287.40	141.456	6.16	0.367	92.26	60.611
295.30	145.182	6.53	0.517	100.81	63.979
303.07	148.641	7.22	0.265	110.00	67.460
160.26	86.515	7.35	0.789	119.40	70.994
170.11	90.217	7.77	0.960	128.84	74.503
180.32	94.350	8.35	1.213	138.25	78.027
190.53	98.533	8.98	1.515	147.65	81.619
200.58	102.741	9.65	1.870	157.54	85.372
210.35	106.882	10.37	2.308		
219.85	111.064	11.16	2.819		

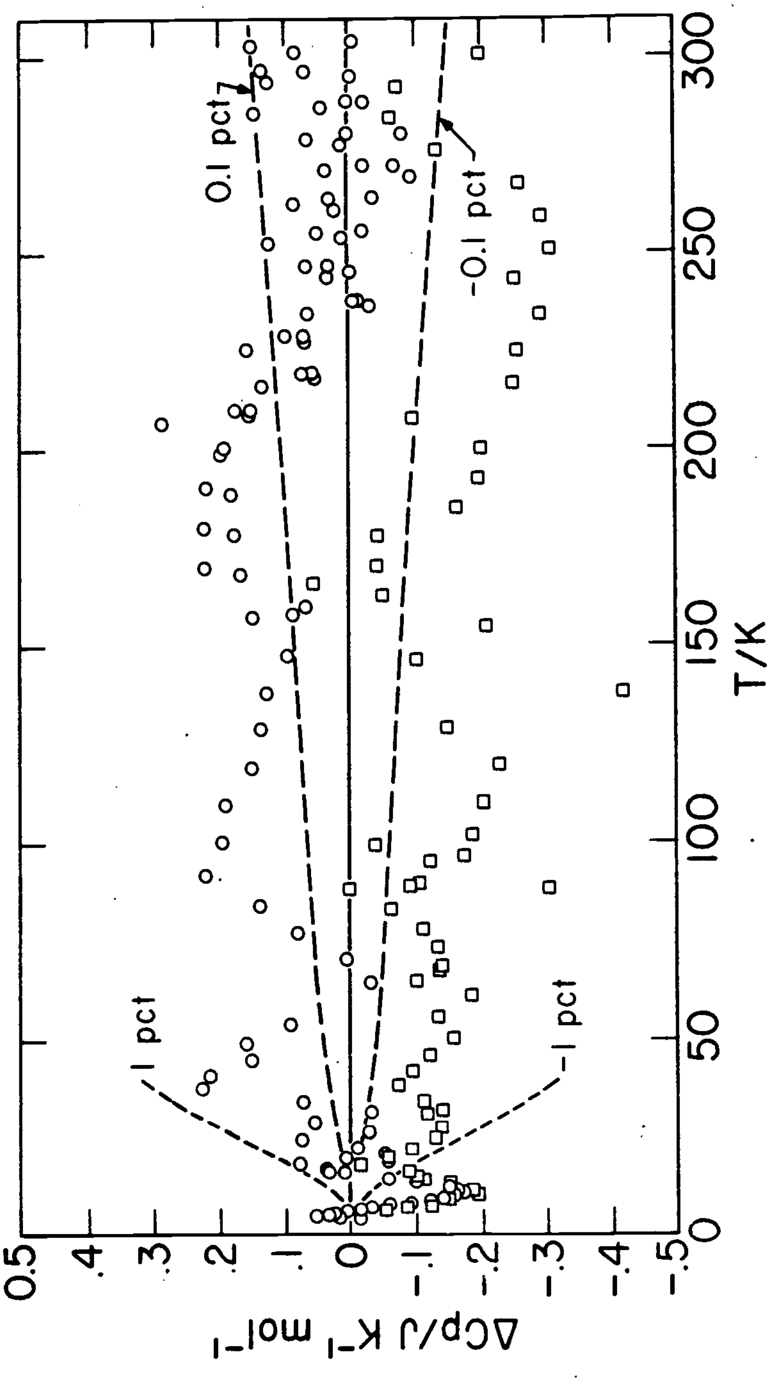

FIGURE 8. Benzoic acid heat capacity deviation plot. The oridinate values are the heat capacity from Furakawa et al. (11) minus the heat capacity determined in the present work. O automated calorimeter; □ manually operated calorimeter.

TABLE 2. Thermodynamic functions of benzoic acid.

T	Cp	S(T)-S(0)	-[G(T)-H(T)]/T	H(T)-H(0)
K	J K^{-1} mol^{-1}	J K^{-1} mol^{-1}	J K^{-1} mol^{-1}	J mol^{-1}
6.	0.398	0.140	0.035	0.629
10.	2.082	0.672	0.164	5.083
15.	5.886	2.187	0.560	24.401
20.	10.983	4.557	1.247	66.21
25.	16.525	7.602	2.204	134.94
30.	21.930	11.097	3.390	231.21
35.	26.940	14.860	4.758	353.58
40.	31.487	18.759	6.263	499.84
45.	35.599	22.709	7.871	667.7
50.	39.334	26.656	9.552	855.2
60.	45.757	34.421	13.053	1282.1
70.	51.035	41.875	16.642	1766.3
80.	55.733	49.002	20.245	2300.6
90.	59.979	55.816	23.822	2879.5
100.	63.934	62.342	27.350	3499.2
110.	67.719	68.614	30.818	4157.6
120.	71.427	74.665	34.221	4853.3
130.	75.124	80.528	37.558	5586.1
140.	78.853	86.232	40.832	6356
150.	82.643	91.801	44.045	7163
160.	86.511	97.258	47.201	8009
170.	90.462	102.620	50.304	8894
180.	94.496	107.905	53.357	9819
190.	98.609	113.124	56.366	10784
200.	102.795	118.288	59.333	11791
210.	107.049	123.406	62.262	12840
220.	111.364	128.485	65.157	13932
230.	115.736	133.532	68.020	15068
240.	120.163	138.551	70.855	16247
250.	124.642	143.547	73.663	17471
260.	129.171	148.523	76.446	18740
270.	133.748	153.484	79.208	20055
273.15	135.198	155.044	80.073	20478
280.	138.366	158.431	81.949	21415
290.	143.014	163.368	84.671	22822
298.15	146.811	167.384	86.877	24003
300.	147.673	168.295	87.377	24275

This value can also be checked by a "propagation-of-error" analysis of equation 1 using the following uncertainties: $\Delta i = \Delta v = \pm 0.001$ per cent; $\Delta(\Delta t) = \pm 0.01$ per cent; $\Delta m = \pm 0.001$ per cent; $\Delta(\Delta T) = \pm 0.005$ per cent. A value of ±0.023 per cent is calculated, which, given the assumptions in the "propagation-of-error" formula, checks closely with the experimental value. From Figure 8 the accuracy of the C_p values is ±0.1 per cent from 200 to 300 K; ±0.2 per cent from 50 to 200 K; ±1.0 per cent from 15 to 50 K; and ±2.0 per cent from 5 to 15 K.

V. Discussion and Recommendations

The results have shown that heat capacity measurements of equal or greater accuracy and precision than those obtained with the manual system can be made with the automated system described. The operator can now be freed from the repetitive tasks of logging the data and controlling the experiment.

There are improvements that could be made that would increase the precision of the measurements. The most important would be to decrease the uncertainty in the time measurement. The ±0.01 per cent uncertainty is the largest error in any of the quantities in equation 1. The other major improvement would be to reduce the noise and drift of the thermometer constant-current source by using an operational amplifier with more rigid specifications. This would decrease the uncertainty in the temperature measurement.

BIBLIOGRAPHY

1. Westrum, Jr., E. F., G. T. Furukawa, and J. P. McCullough. Adiabatic Low-Temperature Calorimetry from Experimental Thermodynamics, Calorimetry of Non-Reacting Systems, v 1. J. P. McCullough, D. W. Scott eds. 1968.
2. Andrews, J. T. S., P. A. Norton, and E. F. Westrum, Jr. An Adiabatic Calorimeter for Use at Superambient Temperatures. The Heat Capacity of Synthetic Sapphire (α-Al_2O_3) From 300 to 500 K. J. Chem. Thermodynamics, v 10, October 1978.
3. Chang, S. S. Automated Adiabatic Calorimetric System for Heat Capacity Measurement. Seventh Proc. on Thermo. Properties. AIME. 1977.
4. Martin, D. L., L. L. T. Bradley, W. J. Cazemier, and R. L. Snowdon. Automatic Calorimetry in the 3-30 K Range. The Specific Heat of α-Al_2O_3 Between 300 and 470 K. Rev. Sci. Instr., v 41, December 1970.
5. Martin, D. L., and R. L. Snowdon. An Automatic Data Acquisition System for Calorimetry: The Specific Heat of α-Al_2O_3 Between 300 and 470 K. Rev. Sci. Instr. v. 41, December 1970.
6. Moses, O., O. Ben-Aroya, and N. Lupu. Simple Calorimetric System for the Temperature Range 3 - 300 K With On-Line Computer. Rev. Sci. Instr., v. 48, August 1977.
7. Stuve, J. M., D. W. Richardson, and E. G. King. Low-Temperature

Heat Capacities and Enthalpy of Formation of Copper Oxysulfate. BuMines RI 8045, 1975.

8. Chang, S. S. A Self-Balancing Nanovolt Potentiometric System for Thermometry and Calorimetry. NBS-A, Phys. and Chem., v. 80A, July-August 1976.

9. Chang, S. S. Switch Arrangement and Power Averaging in Constant Current Calorimetric Heating. Rev. Sci. Instr., v. 40, 1969.

10. Forsythe, G. E. Generation and Use of Orthogonal Polynomials for Data-Fitting with a Digital Computer. J. Soc. Indust. Appl. Math., v. 5, June 1957.

11. Furukawa, G. T., R. E. McCoskey, and G. J. King. Calorimetric Properties of Benzoic Acid from 0 to 410 K. J. Res. Nat. Bur. Stand., v. 47, 1951.

12. Olofsson, G., S. Angus, G. T. Armstrong, A. N. Kornilov. A Report of IUPAC Commission 1.2 on Thermodynamics. Assignment and Presentation of Uncertainties of the Numerical Results of Thermodynamic Measurements. J. Chem. Thermodynamics, v. 13, 1981.

13. Ho, C. Y., R. W. Powell, and P. E. Liley. Thermal Conductivity of Copper. J. Phys. Chem. Ref. Data, v. 3 (Suppl. 1), 1974.

14. Sparks, L. L., R. L. Powell, and W. J. Hall. Reference Tables for Low-Temperature Thermocouples. NBS Monograph 124, 1972.

APPENDICES

Appendix A

Heat Leak Correction

The heat leak correction is based on the following equation:

$$q = (kA\Delta t\Delta T)/l$$

where q = heat loss, J

k = thermal conductivity of copper, J cm^{-1} s^{-1} K^{-1}

A = cross-sectional area of 14 wires, cm^2

l = length of wires, cm

Δt = time heater was on, sec

ΔT = average temperature difference of the ends of the wire leads, K

The subroutine cuk returns a value of the thermal conductivity of copper when given a temperature. This calculated conductivity is based on a table lookup routine using values for k from Ho et al. (13). The subroutine themf returns a value for ΔT in the above equation when given an emf difference of a Type T thermocouple and an absolute temperature. The values for the thermocouple voltages come from Sparks et al. (14).

Appendix B

This appendix contains the listings of all the programs used in the control of the calorimeter. The following table lists the program names and their functions.

Program Name	Function
check	Checks validity of data from the Fluke DVM.
cuk	Calculates the thermal conductivity of copper by linear interpolation of table values.
drout	Turns bits on or off in the 16-bit digital output card, the DR11C.
empty	Calculates the C_p of the empty capsule.
heater	Turns the electrical heater on and off and measures the heater power.
iec	Sends and receives commands and data over the IEEE-488 bus.
lcalpt	Main calling program.
linear	Calculates straight line for time and temperature data.
orth	Fits a quadratic equation to the time and temperature data.

Program Name	Function
pclock	Turns programmable clock on and off and reads its registers.
readf	Reads the Fluke DVM
rpt	Measures the resistance of the platinum resistance thermometer and the 100 ohm standard resistor.
setup	Attaches the IEC11-A bus controller and assigns the IEC driver a logical unit number.
taf	Power failure recovery subroutine.
temp	Calculates temperature from the resistance of the platinum resistance thermometer.
tempg	Calculates temperature from the resistance of the germanium resistance thermometer.
themf	Calculates Seebeck coefficient for a type T thermocouple.
tplot	Plots time and temperature data on the terminal.

```
c
c a subroutine to check the reading of the
c the fluke to see if it contains valid
c characters for the DECODE command
c
      subroutine check(flukin,flg)
      byte flukin(14)
      logical flg
      flg=.true.
c     write(5,4)(flukin(i),i=1,14)
    4 format(1x,14A1)
      if(flukin(1).ne.'+'.and.flukin(1).ne.'-')go to 2
      return
    2 write(5,3)(flukin(i),i=1,14)
    3 format(/,1x,'fluke reading error',14A1,/)
      flg=.false.
      return
      end
```

```
c
c Calculates the thermal conductivity of copper
c
      subroutine cuk(temp,xk)
      real*8 xt(62),k(62),temp,xk
      xt( 1)=   0.000
      xt( 2)=  23.464
      xt( 3)=  24.116
      xt( 4)=  24.659
      xt( 5)=  24.333
      xt( 6)=  25.419
      xt( 7)=  26.614
      xt( 8)=  27.266
      xt( 9)=  28.026
      xt(10)=  28.678
      xt(11)=  28.787
      xt(12)=  29.547
      xt(13)=  30.416
      xt(14)=  32.154
      xt(15)=  34.001
      xt(16)=  35.522
      xt(17)=  36.065
      xt(18)=  37.151
      xt(19)=  38.129
      xt(20)=  39.324
      xt(21)=  40.844
      xt(22)=  41.714
      xt(23)=  42.474
      xt(24)=  44.646
      xt(25)=  46.819
      xt(26)=  47.797
      xt(27)=  48.992
      xt(28)=  50.512
      xt(29)=  51.925
      xt(30)=  54.206
      xt(31)=  55.618
      xt(32)=  58.116
      xt(33)=  61.158
      xt(34)=  63.657
      xt(35)=  66.481
      xt(36)=  70.283
      xt(37)=  71.804
      xt(38)=  75.280
      xt(39)=  78.430
      xt(40)=  82.123
      xt(41)=  85.274
      xt(42)=  88.967
      xt(43)=  95.376
      xt(44)=  99.938
      xt(45)= 104.610
```

```
xt(46)= 110.910
xt(47)= 116.016
xt(48)= 125.032
xt(49)= 133.070
xt(50)= 145.019
xt(51)= 153.058
xt(52)= 176.305
xt(53)= 197.596
xt(54)= 238.875
xt(55)= 268.204
xt(56)= 302.097
xt(57)= 354.021
xt(58)= 397.907
xt(59)= 442.445
xt(60)= 483.398
xt(61)= 508.709
xt(62)= 529.891
 k( 1)=  0.000000
 k( 2)= 10.036133
 k( 3)= 10.148579
 k( 4)= 10.339426
 k( 5)= 10.472330
 k( 6)= 10.647843
 k( 7)= 10.918756
 k( 8)= 11.152197
 k( 9)= 11.390744
 k(10)= 11.419716
 k(11)= 11.450385
 k(12)= 11.479358
 k(13)= 11.494692
 k(14)= 11.484463
 k(15)= 11.380515
 k(16)= 11.254431
 k(17)= 11.124937
 k(18)= 10.963062
 k(19)= 10.864237
 k(20)= 10.627386
 k(21)= 10.407582
 k(22)= 10.218448
 k(23)= 10.015676
 k(24)=  9.678304
 k(25)=  9.049544
 k(26)=  8.678094
 k(27)=  8.442957
 k(28)=  8.165208
 k(29)=  7.890887
 k(30)=  7.587593
 k(31)=  7.355847
 k(32)=  7.129225
 k(33)=  6.824217
```

```
      k(34)=  6.565230
      k(35)=  6.278967
      k(36)=  6.057450
      k(37)=  5.839360
      k(38)=  5.624663
      k(39)=  5.428709
      k(40)=  5.183343
      k(41)=  5.009542
      k(42)=  4.857895
      k(43)=  4.670457
      k(44)=  4.540963
      k(45)=  4.450653
      k(46)=  4.355238
      k(47)=  4.281959
      k(48)=  4.200181
      k(49)=  4.149055
      k(50)=  4.089432
      k(51)=  4.057050
      k(52)=  3.985484
      k(53)=  3.927556
      k(54)=  3.855990
      k(55)=  3.828731
      k(56)=  3.803168
      k(57)=  3.787834
      k(58)=  3.782711
      k(59)=  3.770786
      k(60)=  3.757165
      k(61)=  3.743527
      k(62)=  3.729889
     do 3 i=1,62
     if(temp.gt.xt(i))go to 3
     ij=i-1
     go to 4
   3 continue
   4 xk=k(ij+1)-((k(ij+1)-k(ij))/(xt(ij+1)-xt(ij)))*(xt(ij+1)-temp)
     return
     end
```

```
c
c outputs a 16 bit binary number to the
c output register in the DR11C
c a 1 turns on a switch, a 0 turns off
c a switch
c
      SUBROUTINE DROUT (IOUT)
      COMMON /DR11C/ DRCSR,DROTBF,DRINBF
      INTEGER DRCSR,DROTBF,DRINBF,IOUT,IDS(2)
      do 10 i=1,10
      DRCSR=0
      DROTBF=IOUT
      DRCSR=2
      DRCSR=0
      DROTBF=IOUT
      DRCSR=2
   10 continue
      CALL WAIT (10,1,IDS)
c     Write(5,1)DROTBF
    1 FORMAT(' DROTBF=',O6)
      RETURN
      END
```

```
c
c calculates the heat capacity of the
c empty capsule
c
      subroutine empty(t,cempty)
      real*8 a(11),b(11),c(10),cempty,T
      data a/-0.2099952205900203 6E-01,
     &0.51271219211798478E-01,
     &-0.43068125438254253E-01,
     &0.19615866472478047E-01,
     &-0.51474361475591149E-02,
     &0.84149458319076206E-03,
     &-0.87461358769621408E-04,
     &0.57560732225678730E-05,
     &-0.23099489810575773E-06,
     &0.51313294369199313E-08,
     &-0.48149415552714812E-10/
      data b/-0.27168067260579932E+02,
     &0.96956254033208719E+01,
     &-0.15018767751000822E+01,
     &0.13319397355278289E+00,
     &-0.74648116297936615E-02,
     &0.27664610938540001E-03,
     &-0.68561277790855023E-05,
     &0.11223149012813701E-06,
     &-0.11632325993776311E-08,
     &0.69086862630129183E-11,
     &-0.17898226000332406E-13/
      data c/0.14518432616940280E+02,
     &-0.14008389884478441E+01,
     &0.53366086839724836E-01,
     &-0.82975162381982340E-03,
     &0.74721056524730017E-05,
     &-0.42478556038407633E-07,
     &0.15496765596869648E-09,
     &-0.35236660615947243E-12,
     &0.45498423116965431E-15,
     &-0.25489114122084163E-18/
      if(T.gt.63.)go to 50
      if(T.gt.17.)go to 40
      cempty=0.0
      do 10 i=1,10
      cempty=cempty+a(12-i)
   10 cempty=cempty*T
      cempty=cempty+a(1)
      return
   40 cempty=0.0
      do 20 i=1,10
      cempty=cempty+b(12-i)
   20 cempty=cempty*T
```

```
      cempty=cempty+b(1)
      return
   50 cempty=0.0
      do 30 i=1,9
      cempty=cempty+c(11-i)
   30 cempty=cempty*T
      cempty=cempty+c(1)
      return
      end
```

```
      SUBROUTINE HEATER(Htpwr,tdelt,Hvolt2,Htamp2,ton,vtherm)
c
c this subroutine turns on the heater at a specified
c power, Htpwr, for ten minutes.  It also measures the
c current through the heater and the voltage across the
c heater and the elapsed time of the heater on time.
c this subroutine returns to the main program the average
c heater voltage, the average heater current, the turn
c on time in seconds from midnight, and the length of
c time the heater was on.
c
      real*8 Htamps,tdelt,toff,ton,Htpwr,vsum,vtherm
      Real*8 Htamp1,Htamp2,Hvolt1,Hvolt2,dummy
      real*8 iloop,Azero,Vzero,iloopa,iloopv
      real*8 suma,isuma,sumv,isumv
      BYTE flukin(14),EDCout(9),EDCstd(9),EDC2(9),
     &ffr(14),flct(4),ftcl(4),elct(4),ffr12
      integer*4 out
      logical flg
      DATA flukin/14*0/

      DATA flct,ftcl/'_','?','@','$','_','?','D',' '/

      DATA elct/'_','?','@','#'/

      DATA EDC2/'+',5*0,'0','0','?'/

      DATA EDCout/6*0,'0','0','?'/

      DATA EDCstd/'+',6*'9','0','?'/

      DATA ffr/'%',',','H','0','F','3','T','B','0','V','R','1',',','?'/
c
c stop the programmable clock and zero its registers
c
      call pclock(0,dummy)
c
c set EDC to program the HP current source, fluke to measure
c heater std voltage, and dummy heater on.
c
      call drout("146677)
c
c set edc to calculated heater power.  EDC listen and controller to talk
c
      if(Htpwr.gt.0.018)Htpwr=0.018
      out=Htpwr*11000000
      encode(6,1,EDCout)out
    1 format(I6)
      do 80 i=1,6
   80 if(EDCout(i).eq.' ')EDCout(i)='0'
```

```
      do 2 i=1,6
    2 EDCout(8-i)=EDCout(7-i)
      EDCout(1)='+'
      EDCout(8)='1'
      EDCout(9)='?'
      WRITE(5,100)EDCout
  100 format(' EDCout=',9A1)
      call IEC (4,elct,"0430)
      call IEC (9,EDCout,"0400)
c
c check current to make sure not too large
c
      call IEC (4,flct,"0430)
      call IEC (14,ffr,"0400)
      call IEC (4,ftcl,"0430)
      do 13 ih=1,5
      do 3 i=1,2
    3 call IEC (14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..true.)go to 12
   13 continue
c
c no valid data after ih=5 tries, so return
c
      go to 98
   70 format(1x,14A1)
   12 DECODE (12,4,flukin)Htamps
    4 format(E15.6)
   10 type *,' Htamps=',Htamps
c
c heater current too large, return
c
      if (Htamps.gt.0.181)go to 99
c
c decide which range on fluke to give highest resolution
c
      ffrl2='1'
      if(Htamps.gt..005)ffrl2='2'
      if(Htamps.gt..04)ffrl2='3'
c
c set switches so fluke reads the zero offset when
c its inputs are shorted by heater
c
      call DROUT ("145677)
      do 200 i=1,2
  200 call IEC(14,flukin,"1000)
      call check(flukin,flg)
       if(flg.eq..true.)go to 201
      type *,'Azero no good for DECODE'
      Azero=0.0
```

```
      go to 202
  201 DECODE(12,4,flukin)Azero
  202 type *,'Azero=',Azero
      ffr(12)=ffr12
      call IEC(4,flct,"0430)
      call IEC(14,ffr,"0400)
      call IEC(4,ftcl,"0430)
      do 203 i=1,2
  203 call IEC(14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..true.)go to 204
      type *,'Vzero no good for DECODE'
      Vzero=0.0
      go to 205
  204 DECODE(12,4,flukin)Vzero
  205 type *,'Vzero=',Vzero
c
c get start time
c
      vsum=0.0
      Htamp2=0.0
      iloop=0.0
      iloopa=0.0
      iloopv=0.0
      Hvolt2=0.0
      tdelt=0.0
      ton=secnds(0.0)
      call pclock(1,dummy)
c
c this command executed so time between starting the clock and turning
c on the heater is the same as the time between turning off the clock
c and turning off the heater
c
      if(tdelt.ge.0.0.and.tdelt.lt.600.)go to 9
c
c start heater
c
    9 call DROUT ("152677)
      iloop=iloop+1
      iloopa=iloopa+1
      iloopv=iloopv+1
c
c measure Vstd
c
      ffr(12)='1'
      call IEC (4,flct,"0430)
      call IEC (14,ffr,"0400)
      call IEC (4,ftcl,"0430)
      call IEC (14,flukin,"1000)
      isuma=0.0
```

```
      suma=0.0
      do 6 i=1,2
      Htampl=0.0
      call IEC (14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..false.)go to 6
      DECODE (12,4,flukin)Htampl
      Htampl=Htampl-Azero
      isuma=isuma+1
      suma=suma+Htampl
    6 continue
      if(suma.gt.0)go to 300
      suma=0.0
      iloopa=iloopa-1
  300 Htamp2=(suma/isuma)+Htamp2
c
c measure Vht
c
      ffr(12)=ffrl2
      if(ffrl2.eq.'1')go to 11
      call IEC (4,flct,"0430)
      call IEC (14,ffr,"0400)
      call IEC (4,ftcl,"0430)
   11 call DROUT ("151677)
      call IEC (14,flukin,"1000)
      isumv=0.0
      sumv=0.0
      do 8 i=1,10
      Hvoltl=0.0
      call IEC (14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..false.)go to 8
      DECODE (12,4,flukin)Hvoltl
      Hvoltl=Hvoltl-Vzero
      isumv=isumv+1
      sumv=sumv+Hvoltl
    8 continue
      if(sumv.gt.0)go to 301
      sumv=0.0
      iloopv=iloopv-1
  301 Hvolt2=(sumv/isumv)+Hvolt2
c
c measure thermocouple
c
      FFR(12)='1'
      call drout("53677)
      call IEC(4,flct,"0430)
      call IEC(14,ffr,"0400)
      call IEC(4,ftcl,"0430)
      call IEC(14,flukin,"1000)
```

```
      call IEC(14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..false.)type *,'bad thermocouple read in heater'
      DECODE(12,4,flukin)vtherm
      vsum=vsum+vtherm
      FFR(12)=FFR12
c
c Decide if heat time elapsed
c if just after midnight, correct the elapsed time calculation
c
      tdelt=secnds(0.0)
      if(tdelt.lt.ton)tdelt=tdelt+86400.
      tdelt=tdelt-ton
c
c write intermediate results
c
      write(5,400)suma,isuma,sumv,isumv,vtherm,tdelt
  400 format(1x,f15.7,1x,f4.0,1x,f12.5,1x,f5.0,1x,f10.6,1x,f10.3)
      if(tdelt.ge.0.0.and.tdelt.lt.600.)go to 9
      call pclock(0,tdelt)
      if(tdelt.ge.0.0.and.tdelt.lt.600.)go to 9
      call DROUT("177677)
      write(5,401)iloop,iloopa,iloopv
  401 format(' iloop=',f4.0,' iloopa=',f4.0,' iloopv=',f4.0)
      Htamp2=(Htamp2/iloopa)/9.99979
      Hvolt2=Hvolt2/iloopv
      vtherm=vsum/iloop
c
c 24-sep-81 checked line clock, 60 Hz, against
c kwllk and found the kwllk too slow by
c 0.08%, so the following correction is applied
c to tdelt
c
      tdelt=tdelt*1.0008
      write(5,20) Hvolt2,Htamp2,ton,tdelt
   20 format(/1x,' Hvolt2=',f12.7,' Hamp2=',f12.7,' ton=',f15.4,
     &' tdelt=',f15.4,/)
      return
   99 type *,' tried to set heater to more than 18ma'
      call drout("177677)
      return
   98 type *,'too many bad reads on heater max current check'
      call drout("177677)
      return
      end
```

```
c This subroutine sends and receives both data and commands
c over the IEEE-488 bus.  The subroutine receives from the calling
c program the I/O function code (iocode), the number of
c bytes to be sent or received (iadr2), and the name of the
c array to hold the data.
c Essentially, a QIO is issued and the event flag is checked
c to see if the QIO is done. The subroutine also checks to
c see if a pre-determined amount of time has ellapsed
c (e.g. 30 seconds). If the event flag is no set within
c 30 seconds, then an IO.KIL is issued.
c
      subroutine IEC (iadr2,ioaray,iocode)
      byte ioaray(iadr2)
      integer iost(2),ids(2),adr(6),iadr2,idsc(2),idsw(2),idsr(2)
      integer flag
      adr(2)=iadr2
      call getadr (adr(1),ioaray(1))
      time=secnds(0.0)
      call qio (iocode,4,1,,iost,adr,ids)
c
c check event flag to see if qio done
c if not set in 30 seconds, then take drastic action
c
    6 time2=secnds(time)
      if(time2.gt.300.)go to 2
      call readef(1,idsr)
      IF (IOST(1).NE.1.OR.IDS(1).NE.1)GO TO 6
    2 if(iost(1).eq.1.and.ids(1).eq.1)return
c
c cancel qio with an IO.KIL
c
      WRITE(5,100)IOST(1),IDS(1)
  100 FORMAT(' IEC qio errors',2o6)
      write(5,110)iocode
  110 format(' I/O function code was',o6)
      call qio("12,4,1,,iost,,ids)
      call wait(2,2,ids)
      type *,'WARNING- qio in IEC not completed'
      type *,'IEC IO.KIL status',iost(1),ids(1)
      type *,'QIO event flag checked for',time2,' seconds'
c
c turn off all switches but "+"  switch for the constant
c current source
c
      type *,'DROUT sent "177677 every second'
    5 call DROUT ("177677)
      call wait(1,2,idsw)
      go to 5
      end
```

```
c
c  This program is the main control program for the
c  low temperature calorimeter when using the Pt
c  thermometer. Temperatures are taken and calculated
c  and equilibrium is decided on, thereby
c  resulting in the heater being turned on for
c  a certain time. Temperatures are taken and calculated again
c  and Cp calculated and the whole process repeated.
c
      program lcalpt
      real*8 TCp,Tlow,tdelt,tmid,Htamp2,PWR,Thigh,ton,vbeg,vend
      real*8 Cp,Htpwr,R,THT,Hvolt2,DT,tstart,tcheck,vtherm
      real*8 TK(11),t(11),time,TT,c0,c1,c2,Rhtr,ECp,CHe,xmoles
      real*8 totcp,dcp,dth,xk,tstop
      common /Ttime/ TK,t,c0,c1,c2,flag,mark
      external taf
      logical flag1,flag2,flag
      byte flukin(14),ffr(14),flct(4),ftcl(4)
      integer dsw, iost(2), ids(2)
      byte ioren
      data flct,ftcl/'_','?','@','$','_','?','D',' '/

      data ffr/'%',',','H','1','F','3','T','B','0','V','R','1',',','?'/
      data Rhtr/4100./
      data TK,t/11*0,11*0/
c
c Initialization section
c
      call enastr
      call pwrup(taf)
      ioren='$'
      ipts=11
      call setup(4)
c
c get first Cp estimate and max time between heater runs
c
      type *,'First Cp estimate'
      accept *,totcp
      type *,'Equilibrium time maximum, tstop='
      accept *,tstop
c
c get sample mole weight and Helium correction (in joules)
c
      open(unit=1,name='runspecs.dat',type='old',dispose='save')
      read(1,*)xmoles
      read(1,*)CHe
      write(5,213)xmoles,CHe
  213 format(1x,'runspes.dat: xmoles=',f9.6,'  CHe=',f9.6)
      close(unit=1)
      first=1
```

```
c
c read liquid helium tank thermocouple
c
    1 continue
      open(unit=2,name='control.dat',type='unknown',dispose='save',
     &access='append')
      write(2,212)
  212 format(1x,'***********************')
      close(unit=2)
      call DROUT("137677)
      call IEC(4,flct,"0430)
      call IEC(14,ffr,"0400)
      call IEC(4,ftcl,"0430)
      call IEC(14,flukin,"1000)
      call IEC(14,flukin,"1000)
      write(5,120)(flukin(i),i=1,14)
  120 format(/,14A1)
c
c Temperature measurement section
c
c
c  set up TK and t arrays with the first reading
c
      call RPT(R,time)
      call TEMP(R,TT)
      TK(1)=TT
      t(1)=time
      do 2 i=1,ipts
      TK(i)=TK(1)
    2 t(i)=t(1)
      tstart=secnds(0.0)
c
c  moving average and check for second derivative
c  changing sign.  After 5 sign changes assume
c  equilibrium.
c
      flag2=.true.
      ieq=0
    3 do 4 i=1,ipts-1
      TK(i)=TK(i+1)
    4 t(i)=t(i+1)
      call RPT (R,time)
      call TEMP (R,TT)
      TK(ipts)=TT
      t(ipts)=time
c
c  fit the temp vs. time curve to a quadratic equation
c
      mark=0
      call orth
```

```
      flag1=flag
      write(5,210)time,TT,flag1
  210 format('+',72x,f12.4,1x,f10.6,1x,L1)
      open(unit=2,name='control.dat',type='unknown',dispose='save',
     &access='append')
      write(2,211)time,TT
  211 format(1x,f12.4,1x,f10.6)
      close(unit=2)
c
c test sign of second derivative and if opposite from the last
c fit, increment sign change counter, ieq, by one.
c
      if(flag1.eq..true..and.flag2.eq..false..or.flag1.eq..false..and.
     & flag2.eq..true.)ieq=ieq+1
      flag2=flag1
c
c if sign has not changed at least 5 times then take another
c temperature, because not at equilibrium
c
      tcheck=secnds(tstart)
      if(tcheck.gt.tstop)go to 5
      if(ieq.lt.5) go to 3
c
c plot time and temperature arrays on the terminal
c
    5 call linear
      CALL TPLOT
c
c Energy input section
c
c Measure the thermocouple emf between the ends
c of the wrie leads
c
      Call drout("77677)
      call IEC(4,flct,"0430)
      call iec(14,ffr,"0400)
      call iec(4,ftcl,"0430)
      call iec(14,flukin,"1000)
      call iec(14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..false.)type *,'bad VEND'
      DECODE(12,59,flukin)VEND
   59 format(e15.6)
      type  *,'vend=',vend
c
c  if this is the first time through, then skip the Cp calculation.
c
      if(first.eq.1)go to 6
c
c calculation section
```

```
c
c
c calculate the high temperature at mid heating period
c delta T and the average temperature
c
      mark=1
      type *,'coefficients',c0,c1
      vtherm=vtherm-((vend+vbeg)/2.)
      type *,'vtherm=',vtherm
      Thigh=c0+(c1*tmid)
      DT=Thigh-Tlow
      TCp=(Tlow+Thigh)/2.
c
c Heat leak correction
c 0.0000819444 is cross sectional area of 14
c copper wires divided by their length (21.6 cm)
c
c cuk returns the termal conductivity of copper given a temperature
c
      call cuk(TCp,xk)
c
c themf returns the delta temperature given a delta emf
c of a copper-constantan thermocouple and the temperature
c
      call themf(TCp,vtherm,dth)
      dcp=0.0000819444*tdelt*dth*xk
      dcp=dcp/DT
c
c calculate the Cp of the sample
c
      totcp=PWR/DT
      call empty(TCp,Ecp)
      Cp=totcp-dcp-Ecp-CHe
      Cp=Cp/xmoles
      open(unit=1,name='cp.dat',type='unknown',access='append')
      write(1,14)TCp,Cp,dcp,DT,totcp,ECp,Tlow,Thigh,tmid,Hvolt2,Htamp2
     &,tdelt
   14 format(1x,f9.4,1x,f11.5,1x,f8.5,1x,f7.4,1x,f11.5,1x,f9.5,
     &1x,f9.4,1x,f9.4,1x,f11.4,1x,f10.6,1x,f10.7,1x,f10.4)
      close(unit=1,dispose='save')
      write(5,10)TCp,Cp,DT,dcp,totcp,ECp,Tlow,Thigh,tmid
   10 format(/,1x,'TCp=',f12.5,' Cp=',f12.5,' DT=',f12.5,' dcp=',
     &f12.5,/,' totcp=',f12.5,
     &' ECp=',f12.5,' Tlow=',f12.5,' Thigh=',f12.5,' tmid=',f15.4,/)
c
c if the last temperature is above 310 K then exit.
c
      if(Thigh.gt.303.)go to 9
c
c Energy input section
```

```
c
c
c decide how much power to run through the heater for 10 minutes
c
    6 THT=TK(ipts)*0.1
      if(THT.gt.10.)THT=10.
      Htpwr=totcp*THT/600.
c
c calculate approximate heater resistance
c
      Rhtr=TK(ipts) + 4100.0
      Htpwr=sqrt(Htpwr/Rhtr)
c
c turn on heater for 10 minutes
c
      vbeg=vend
      call HEATER (Htpwr,tdelt,Hvolt2,Htamp2,ton,vtherm)
      type *,'vtherm from heater',vtherm
c
c calculate energy input to sample
c
      PWR=Htamp2*Hvolt2*tdelt
c
c calculate mid heating period temperature and time
c
      tmid=ton+(tdelt/2.)
      mark=1
      Tlow=c0+(tmid*c1)
      first=0
      go to 1
c
c Finished
c
    9 continue
      call dsastr
      call wtqio ("2000,4,1,,iost,,ids)
      if(iost(1).ne.1.or.ids(1).ne.1)write(5,12)iost(1),ids(1)
   12 format(' ERROR--IEO: not detached',2o6)
      close(unit=2)
c
c keep setting switches so only thing happening is
c current through thermometers
c
   20 call DROUT("177677)
      go to 20
      end
```

```
c
c least squares linear curve fit
c TK=c0+c1*t
c
      subroutine linear
      common /Ttime/TK(11),t(11),c0,c1,c2,flag,mark
      logical flag
      INTEGER TBAD(11)
      real*8 avg,DIFF,N,SIGMA,MEAN
      real*8 t,TK,c0,c1,c2
      Real*8 sumt,sumTK,sumt2,sumtTK
      ipts=11
      do 4 i=1,ipts
    4 tbad(i)=0
      ibad=0
    2 sumt=0.0
      sumTK=0.0
      sumt2=0.0
      sumtTK=0.0
      n=0.0
      c0=0.0
      c1=0.0
      c2=0.0
      do 1 i=1,ipts
      if(tbad(i).eq.1)go to 1
      n=n+1.
      sumt=sumt+t(i)
      sumTK=sumTK+TK(i)
      sumt2=sumt2+t(i)*t(i)
      sumtTK=sumtTK+t(i)*TK(i)
    1 continue
      c1=sumtTK-(sumt*sumTK/n)
      c1=c1/(sumt2-(sumt*sumt)/n)
      c0=(sumTK-c1*sumt)/n
      c2=0.0
      if(ibad.eq.0)go to 11
      if(ibad.lt.6)return
      c0=mean
      c1=0.0
      type *,'linear returns mean only'
      return
c
c calculate standard deviation
c
   11 mean=sumTK/ipts
      sigma=0.0
      do 10 i=1,ipts
      diff=TK(i)-mean
   10 sigma=sigma+diff*diff
      sigma=sqrt(sigma/(n-1.))
```

```
c
c find bad points
c
      do 3 i=1,ipts
      avg=c0+c1*t(i)
      diff=abs(avg-TK(i))
      if(diff.lt.sigma)go to 3
      ibad=IBAD+1
      tbad(i)=1
      type *,'BAD ',t(i),TK(i)
    3 continue
      if (ibad.gt.0)go to 2
      return
      end
```

```
c
c  Orthogonal curve fitting routine for the
c  equation  TK=c0+c1*t+c2*t**2
c
c  if mark=1, then all the coefficients are calculated.
c  if mark=0, then only the c2 coefficient is calculated
c  and flag set to .true. if positive and set to .false.
c  if negative.
c
      subroutine orth
      common /Ttime/ TK(11),t(11),c0,c1,c2,flag,mark
      logical flag
      real*8 tavg,t,TK,p1(11),p12(11),b1,a2,c0,c1,c2
      real*8 p12sum,p22sum,p1TK,p2(11)
      flag=.true.
c
c  let the number of samples = m
c
      m=11
c
c  calculate t average (tavg)
c
      tavg=0.0
      do 1 i=1,m
    1 tavg=tavg+t(i)
      tavg=tavg/m
c
c  calculate the first orthogonal polynomial p1
c  also calculate p1*p1
c
      do 2 i=1,m
      p1(i)=t(i)-tavg
    2 p12(i)=p1(i)*p1(i)
c
c  calculate the intermediate coeficients b1 and a2
c
      p12sum=0.0
      a2=0.0
      do 3 i=1,m
      a2=t(i)*p12(i) + a2
    3 p12sum=p12(i) + p12sum
      a2=a2/p12sum
      b1=p12sum/m
c
c  calculate the second order coefficient
c
      c2=0
      do 4 i=1,m
      p2(i)=(t(i)-a2)*p1(i) - b1
    4 c2=p2(i)*TK(i) + c2
```

```
      if(c2.lt.0.0)flag=.false.
      if(mark.eq.1)go to 5
      return
c
c  calculate other coefficients
c
    5 c0=0.0
      c1=0.0
      p22sum=0.0
      p1TK=0.0
      do 6 i=1,m
      p22sum=p22sum + p2(i)*p2(i)
      p1TK= p1TK + p1(i)*TK(i)
    6 c0=c0 + TK(i)
      c2=c2/p22sum
      c1=p1TK/p12sum
      c0=c0/m
      c0=c0+c2*(a2*tavg-b1)-c1*tavg
      c1=c1-c2*(a2+tavg)
      return
      end
```

```
c
c Turns the KW11A programmable clock on and off
c and reads its registers
c
      subroutine pclock(iswtch,htime)
      common /kw11a/ null(2)
      common /kw11b/ kwasr,kwabr
      common /kw11c/ nul2(8)
      common /kw11d/ kwacr,kwbsr,kwbbr,kwbcr
      integer*4 isecs,ksecs
      real*8 htime
c
c iswtch=0 stops the clock
c iswtch=1 starts the clock
c iswtch=2 reads the clock
c
c configured as a 24-bit clock running at 10kHz
c with feed B to A
c
c     write(5,2)kwasr,kwabr,kwacr,kwbsr,kwbbr,kwbcr
    2 format(6(1x,o6))
      if(iswtch.eq.1)go to 3
      go to 1
c
c start the clock
c
c A & B Buffers/Preset Registers = 0
c
    3 kwabr=0
      kwbbr=0
c     write(5,2)kwasr,kwabr,kwacr,kwbsr,kwbbr,kwbcr
c
c set A status register for single interval
c mode, rate is B overflow, and enable counter
c A is set
c
      kwasr=1
c
c B status register for Feed B to A
c 10kHz rate and enable counter B set
c
      kwbsr="47
c     write(5,2)kwasr,kwabr,kwacr,kwbsr,kwbbr,kwbcr
c
c extra commands  to make the turn on time the same as
c the turn off time
c
      isecs=isecs*"400
      isecs=isecs+ksecs
      htime=isecs*0.0001
```

```
      kwb="46
      kwb=0
      return
c
c stop B and A clocks
c
c
c read A & B counter registers
c
c   1 write(5,2)kwasr,kwabr,kwacr,kwbsr,kwbbr,kwbcr
    1 continue
      ksecs=kwbcr
      isecs=kwacr
d     write(5,2)kwasr,kwabr,kwacr,kwbsr,kwbbr,kwbcr
      isecs=isecs*"400
c     write(5,2)isecs
      isecs=isecs+ksecs
c     write(5,2)isecs
      htime=isecs*0.0001
      if(iswtch.eq.2)return
      kwbsr="46
      kwasr=0
      return
      end
```

```
c
c subroutine to read the fluke
c
      subroutine readf(V,flukin)
      logical flg
      real*8 V,sum
      byte flukin(14)
      call IEC (14,flukin,"1000)
      do 401 j=1,10
      sum=0.0
      isum=0
      do 400 i=1,1
      call IEC (14,flukin,"1000)
      call check(flukin,flg)
      if(flg.eq..false.)go to 400
      isum=isum+1
      DECODE (12,41,flukin)V
   41 format(E15.6)
      sum=sum+V
  400 continue
  401 if(isum.gt.0)go to 402
  402 V=sum/isum
      return
      end
```

```
c
c measures the resistance of the platinum
c resistance thermometer
c
      subroutine RPt(R,time)
      real*8 Vstdl,vptl
      real*8 time,timel,time2
      Real*8 VPt, Vstd,Rplus,Rminus,R,Vofset,VPt2,Vzero
      integer adr(6),ids(2)
      byte flukin(14),EDCout(9),EDCstd(9),EDC2(9)
      byte ffr(14),flct(4),ftcl(4),elct(4)
      logical flg
      DATA flukin/14*0/
      DATA flct,ftcl/'_','?','@','$','_','?','D',' '/
      DATA elct/'_','?','@','#'/
      DATA EDC2/'+',5*0,'0','0','?'/
      DATA EDCout/6*0,'0','0','?'/
      DATA EDCstd/'+',6*'9','0','?'/
      DATA ffr/'%',',','H','1','F','2','T','B','0','V','R','1',',','?'/
c
c set switches and read zero offset so can correct further
c readings
c
      call DROUT ("177675)
      call IEC (4,flct,"0430)
      call IEC (14,ffr,"0400)
      call IEC (4,ftcl,"0430)
      call readf ( Vzero,flukin)
c
c  set switches to read VPt with fluke
c
      CALL DROUT ("177273)
C
C  SET FLUKE LISTNER, CONTROLLER TALKER -FLCT-
C
      call IEC (4,flct,"0430)
c
c  set fluke for fast read no filter
c
      call IEC(14,ffr,"0400)
c
c  set fluke talker, controller listener -ftcl-
c
      call IEC (4,ftcl,"0430)
c
c  read VPt with fluke
c
      call readf(VPt,flukin)
   51 format(' Vofset=',14A1)
c     WRITE(5,101)VPt
```

```
  101 format(1x,'VPt=',f12.7)
   41 format(E15.6)
      VPt2=VPt
c
c  set EDC listener, controller talker
c
      call IEC (4,elct,"0430)
c
c  set EDC output to offset VPt -Vofset-
c
      EDCout(1)='+'
      do 1 i=2,6
      EDCout(i)=flukin(i+3)
    1 EDC2(i)=EDCout(i)
c     write(5,100)EDCout
  100 format(1x,'EDCout=',9A1)
      call IEC (9,EDCout,"0400)
c
c   set switches so keithley 180 sees offset of EDC and VPt
c   of thermometer, and fluke sees analog output of keithley
c
      CALL DROUT ("177670)
C
C  set fluke to listener, controller to talker
C
c     call IEC (4,flct,"0430)
c
c  set fluke to slow read with filter(timeout disabled)
c
c     call IEC (14,ffr,"0400)
c
c  set fluke to talker, controller to listener
c
      call IEC (4,ftcl,"0430)
c
c  read VPt-Vedc offset (Vofset) from back of keithley with fluke
c
      time1=secnds(0.0)
      call readf(Vofset,flukin)
      Vofset=Vofset-Vzero
c     write(5,102)Vofset
  102 format(1x,'Vofset=',f12.7)
c
c  calculate VPt
c
      VPt=VPt+Vofset*0.001
      vpt1=VPt
c     write(5,108)VPt
  108 format(1x,'VPt+Vofset=',f12.7)
c
```

```
c   set EDC to Vstd
c
      EDCstd(1)='+'
      call IEC (4,elct,"0430)
      call IEC (9,EDCstd,"0400)
c     write(5,103)EDCstd
  103 format(1x,'EDCstd=',9A1)
c
c  set switches so keithley sees offset of EDC and Vstd
c
      CALL DROUT ("177664)
c
c  set fluke to talker, controller to listener
c
      call IEC (4,ftcl,"0430)
c
c   read Vstd-Vedc
c
      call readf(Vofset,flukin)
      Vofset=Vofset-Vzero
c     write(5,102)Vofset
c
c  calculate Vstd
c
      Vstd=0.0999999+Vofset*0.001
      vstdl=vstd
c     write(5,104)Vstd
  104 format(1x,'Vstd=',f12.7)
C
C  calculate Rplus
C
      Rplus=VPt/(Vstd/99.9995)
D     write(5,105)Rplus
  105 format(1x,'Rplus=',f15.7)
c
c  set switches to Vedc-VPt, reverse current
c
      CALL DROUT ("177570)
      call wait(1,2,ids)
C
C SET EDC TO -VPt
C
      EDC2(1)='-'
      call IEC (4,elct,"0430)
      call IEC (9,EDC2,"0400)
c     write(5,103)EDC2
c
c  read VPt-Vedc with fluke
c
      call IEC (4,ftcl,"0430)
```

```
      call readf(Vofset,flukin)
      Vofset=Vofset-Vzero
      VPt=-VPt2
      VPt=VPt+(Vofset*0.001)
c     write(5,101)VPt
c
c set switches for Vedc-Vstd
c
      CALL DROUT ("177564)
c
c  change polarity of EDC
c
      EDCstd(1)='-'
      call IEC (4,elct,"0430)
      call IEC (9,EDCstd,"0400)
c     write(5,103)EDCstd
c
c  read Vedc-Vstd on fluke
c
      call IEC (4,ftcl,"0430)
      call readf(Vofset,flukin)
      Vofset=Vofset-Vzero
c     write(5,102)Vofset
c
c  calculate Vstd
c
      Vofset=Vofset*0.001
      Vstd=-0.0999999+Vofset
c     write(5,104)Vstd
c     write(5,102)Vofset
c
c  calculate Rminus
c
      Rminus=VPt/(Vstd/99.9995)
      time2=secnds(0.0)
D     write(5,106)Rminus
  106 format(1x,'Rminus=',f15.7)
      R=(Rplus+Rminus)*0.5
c     write(5,107)R
  107 format(1x,'R=',f15.7)
      time=(time1+time2)*0.5
      write(5,200)vpt1,vstd1,vstd,vpt,R
  200 format(5(1x,f13.9))
c     open(unit=3,name='rpt.dat',type='unknown',
c    &dispose='save',access='append')
c     write(3,200)vpt1,vstd1,vstd,vpt,r
c     close(unit=3)
      call DROUT("177675)
      return
      end
```

```
c
c Attaches the IEC11-A card and assigns a LUN
c
      subroutine setup(ilun)
      integer dsw,lun,ids(2),iost(2)
      byte ioren
      ioren='$'
      call ASNLUN(ilun,'IE',0,dsw)
      if (dsw.ne.1)type *,' IE0: not assigned lun',ilun
      call wtqio("1420,ilun,1,,iost,,ids)
      if(iost(1).ne.1.or.ids(1).ne.1)type 31,iost(1),ids(1)
   31 format('error IE0: not attached',2o6)
      call IEC (1,ioren,"4000)
      call wtqio("5010,ilun,1,,iost,,ids)
      if(iost(1).ne.1.or.ids(1).ne.1)type 11,iost(1),ids(1)
   11 format('error--IE0: srqs not disabled',2o6)
      return
      end
```

```
      subroutine taf
c
c powerfailure subroutine
c
c taf= that's all folks
c
      type *,'power failure'
    1 call drout ("177677)
      call wait(2,2,ids)
      go to 1
      return
      end
```

```
      SUBROUTINE TEMP (R,T68)
C
C THIS SUBROUTINE CALCULATES THE TEMPERATURE IN DEGREES K GIVEN THE
C RESISTANCE OF A 25 OHM PLATINUM RESISTANCE THERMOMETER
C SPECIFICALLY FOR THE BUREAU OF MINES THERMOMETER SERIAL
C NUMBER1731678.
C
      REAL*8 A1,A2,A3,A4,B1,B2,B3,CONST,C1,C2,C3,C4,DIFF,T68,AC,BC,AB
      REAL*8 DWT68,D1,D2,R,RO,TMP,TPRIME,T100,WCCT68,WT68,TD,A
      DIMENSION A(21)
      DATA A1,A2,A3/-9.4980763E-05,-8.6209085E-05,-2.4945712E-04/
      DATA A4/-2.904E-07/
      DATA B1,B2,B3/9.9704681E-06,8.2723062E-07,6.3699122E-06/
      DATA C1,C2,C3/-8.3479218E-07,2.9242617E-09,-3.5269584E-08/
      DATA C4/-8.645954E-15/
      DATA D1,D2/2.0284733E-08,1.5680932E-10/
      DATA RO/25.557771/
      DATA A/38.59276,43.44837,39.10887,38.69352,32.56883,24.70158,
     *53.03828,77.35767,-95.75103,-223.52892,239.50285,524.64944,
     *-319.79981,-787.60686,179.54782,700.42832,29.48666,-335.24378,
     *-77.25660,66.76292,24.44911/
C
C DEFINITION OF VARIABLES
C
C THE VARIABLES CORRESPOND TO THOSE FOUND IN THE ARTICLE
C THE INTERNATIONAL PRACTICAL TEMPERATURE SCALE OF 1968
C AMENDED EDITION OF 1975, IN METROLOGIA 12,7-17(1976)
C AND TO THE ONES FOUND IN NBS MONOGRAPH 126 TITLED
C PLATINUM RESISTANCE THERMOMETRY
C
      WT68=R/RO
C
C ESTIMATE THE TEMPERATURE
C
      CONST=(LOG(WT68)/3.28)+1.
      TMP=A(1)
      DO 1 J=2,21
       I=J-1
    1 TMP=TMP+(A(J)*(CONST**I))
      T68=TMP
C
C DECIDE WHICH TEMPERATURE RANGE
C
      IF(TMP.GE.273.15)GO TO 100
      IF(TMP.GT.90.188)GO TO 200
      IF(TMP.GT.54.361)GO TO 300
      IF(TMP.GT.20.28)GO TO 400
C
C TEMPERATURE 13.81 TO 20.28 K
C
```

```
  500 DWT68=A1+T68*(B1+T68*(C1+T68*D1))
      WCCT68=WT68-DWT68
      T68=A(1)
      CONST=(LOG(WCCT68)/3.28)+1.
      DO 2 J=2,21
      I=J-1
    2 T68=T68+(A(J)*(CONST**I))
      DIFF=ABS(T68-TMP)
      TMP=T68
      IF(DIFF.GT.0.0000001)GO TO 500
      IF(T68.GT.20.28)GO TO 400
      RETURN
C
C TEMPERATURE RANGE 20.28 TO 54.361 K
C
  400 DWT68=A2+T68*(B2+T68*(C2+T68*D2))
      WCCT68=WT68-DWT68
      T68=A(1)
      CONST=(LOG(WCCT68)/3.28)+1.
      DO 3 J=2,21
      I=J-1
    3 T68=T68+(A(J)*(CONST**I))
      DIFF=ABS(T68-TMP)
      TMP=T68
      IF(DIFF.GT.0.0000001)GO TO 400
      IF(T68.LT.20.28)GO TO 500
      IF(T68.GT.54.361)GO TO 300
      RETURN
C
C TEMPERATURE RANGE 54.361 TO 90.188 K
C
  300 DWT68=A3+T68*(B3+T68*C3)
      WCCT68=WT68-DWT68
      T68=A(1)
      CONST=(LOG(WCCT68)/3.28)+1.
      DO 4 J=2,21
      I=J-1
    4 T68=T68+(A(J)*(CONST**I))
      DIFF=ABS(T68-TMP)
      TMP=T68
      IF(DIFF.GT.0.0000001)GO TO 300
      IF(T68.LT.54.361)GO TO 400
      IF(T68.GT.90.188)GO TO 200
      RETURN
C
C TEMPERATURE RANGE 90.188 TO 273.15 K
C
  200 TD=T68-273.15
      DWT68=(A4*TD)+(C4*(T68-373.15)*(TD*TD*TD))
      WCCT68=WT68-DWT68
```

```
      T68=A(1)
      CONST=(LOG(WCCT68)/3.28)+1.
      DO 5 J=2,21
      I=J-1
    5 T68=T68+(A(J)*(CONST**I))
      DIFF=ABS(T68-TMP)
      TMP=T68
      IF(DIFF.GT.0.0000001) GO TO 200
      IF(T68.LT.90.188)GO TO 300
      IF(T68.GT.273.15)GO TO 100
      RETURN
C
C TEMPERATURE
C
  100 AC=0.0039256764*1.014966942
      BC=0.0039256764*(-0.00014966942)
      AB=AC/BC
      TPRIME=(-AB/2.)-SQRT(((AB/2.)**2.)-((1./BC)*(1.-WT68)))
      T100=TPRIME/100.
      T68=TPRIME+0.045*(T100)*(T100-1.)*((TPRIME/419.58)-1.)*
     #((TPRIME/630.74)-1.)+273.15
      RETURN
      END
```

```
c
c subroutine for calculating temperature in K
c given resistance in ohms of germanium resistance
c thermometer SN 1321 (made by Solitron)
c Calibrated at Lawarance Livermore Labs1969
c
      subroutine tempg(R,T)
      real*8 sum,R, T,a(10)
      data a/1.08615565d-01,
     &2.85791049d-02,
     &-7.52460623d-01,
     &1.44254544d+00,
     &-1.33353641d+00,
     &7.46805948d-01,
     &-2.66478197d-01,
     &5.92374633d-02,
     &-7.42960886d-03,
     &3.98979083d-04/
      R=dlog10(R)
      sum=0.0
      do 1 i=1,9
      sum=sum+a(11-i)
    1 sum=sum*R
      sum=sum+a(1)
      T=1./sum
      return
      end
```

```
c
c calculation of seebeck coefficient for type T thermocouple
c given a temperature. a delta voltage is then divided by the
c seebeck coef and the delta temp is returned
c it is assumed v is in volts
c
      subroutine themf(t,v,dt)
      real*8 t,v,dt,sumt,b(14)
      data b/-3.9974007864d-01,
     &2.6329515981d-01,
     &-9.6491216443d-03,
     &3.8973308068d-04,
     &-9.8186150331d-06,
     &1.6059280063d-07,
     &-1.7932074012d-09,
     &1.4080710479d-11,
     &-7.8671373053d-14,
     &3.1144995156d-16,
     &-8.5433550766d-19,
     &1.5448411036d-21
     &,-1.6565456476d-24
     &,7.9795893156d-28/
      v=v*1000000.
c
c if voltage less than 1 microvolt, too much error possible
c so set to zero
c
      if(v.lt.1.0)v=0.0
      sumt=0.0
      do 1 i=1,13
      sumt=sumt+b(15-i)*(15-i)
    1 sumt=sumt*t
      sumt=sumt+b(1)
d     type *,'Seeb=',sumt
      dt=v/sumt
      return
      end
```

```
c
c this subroutine plots the T vs t data on the terminal
c
      subroutine tplot
      real*8 t(11),TK(11),Tp(11),Tsmall
      common /Ttime/ TK,t,c0,c1,c2,flag,mark
      INTEGER*4 IPRINT
      real*8 c0,c1,c2
      logical flag
      Tsmall=1.0e+38
      do 1 i=1,11
    1 if(TK(i).lt.Tsmall)Tsmall=TK(i)
      do 2 i=1,11
    2 Tp(i)=TK(i)-Tsmall
      write(5,3)
    3 format(//)
      write(5,4)
    4 format(1x,'0',100x,'10 mK')
      write(5,6)
    6 format(1x,10('!!!!!!!!!.'))
      do 7 i=1,11
      iprint=Tp(i)*10000.+1
      IF(IPRINT.GT.100)IPRINT=100
      write(5,8)
    8 format(<iprint>x,'*')
    7 continue
      write(5,3)
      return
      end
```

Appendix C

Experimental heat capacities and related data

T	Cp	dcp	0T	totcp	Ecp	Tlow	Thigh	tmid	Htvolt	Htamp	tdelt
K	$J\ K^{-1}\ mol^{-1}$	$J\ K^{-1}$	K	$J\ K^{-1}$	$J\ K^{-1}$	K	K	sec	volt	amp	sec
210.3701	107.04132	0.02345	9.6093	91.90294	35.59268	205.5655	215.1747	50571.7602	80.873132	0.0177783	614.2236
219.8407	111.17464	0.02295	9.3443	94.48106	35.99787	215.1686	224.5129	52532.8652	80.924573	0.0177615	614.2305
229.0657	115.14968	0.02332	9.1255	96.93517	36.36143	224.5030	233.6285	54638.0485	81.080107	0.0177650	614.1282
238.0865	119.21845	0.02262	8.9227	99.40358	36.69105	233.6251	242.5478	56448.9845	81.268571	0.0177729	614.0706
246.9245	123.06833	0.02219	8.7569	101.72775	36.99127	242.5460	251.3029	58149.4845	81.521340	0.0177942	614.1019
255.5769	126.97243	0.02146	8.5636	104.05311	37.26446	251.2951	259.8587	59739.5295	81.614423	0.0177811	614.0278
264.0501	130.84227	0.02104	8.3972	106.33619	37.51309	259.8515	268.2486	61623.3139	81.768399	0.0177815	614.1279
272.3707	134.76035	0.02075	8.2543	108.62365	37.74059	268.2436	276.4978	63470.3622	82.008688	0.0178008	614.1932
280.5469	138.55538	0.02055	8.1037	110.82903	37.95063	276.4951	284.5987	65611.5597	82.159967	0.0178001	614.1194
288.5745	142.23830	0.02012	7.9601	112.96170	38.14714	284.5944	292.5545	67789.5424	82.285053	0.0177933	614.1473
296.4595	145.82670	0.01783	7.8239	115.03295	38.33380	292.5475	300.3714	69893.7491	82.406681	0.0177848	614.0920
304.2185	149.50510	0.01472	7.7043	117.14212	38.51186	300.3664	308.0706	71740.6780	82.602023	0.0177919	614.0904
158.2131	85.80006	0.02040	10.2520	77.55124	32.41333	153.0871	163.3391	29550.9611	76.362878	0.0169443	614.4574
168.4464	89.74895	0.02078	10.2167	80.42359	33.20885	163.3381	173.5548	31731.7736	77.712514	0.0172038	614.5824
178.6634	93.84091	0.02079	10.2262	83.27079	33.90437	173.5503	183.7765	33765.1596	79.220253	0.0174975	614.3193
188.8962	98.01891	0.02131	10.2492	86.08127	34.51741	183.7716	194.0208	35873.6024	80.746038	0.0177916	614.1344
198.9940	102.19266	0.02111	9.9595	88.81242	35.05408	194.0143	203.9737	37647.7072	80.915396	0.0177994	614.1487
208.8157	106.38058	0.02082	9.6937	91.48286	35.52267	203.9689	213.6625	39826.0090	81.094452	0.0178067	614.1195
218.3659	110.56264	0.02115	9.4180	94.09661	35.93704	213.6569	223.0749	41710.3912	81.088839	0.0177959	614.1184
227.6644	114.57135	0.02138	9.1877	96.57570	36.30800	223.0706	232.2583	43374.0612	81.199289	0.0177928	614.1537
236.7443	118.64944	0.02128	8.9788	99.05552	36.64353	232.2549	241.2337	45405.6719	81.362335	0.0177973	614.2110
245.6245	122.55777	0.02107	8.7870	101.41537	36.94847	241.2310	250.0180	47327.0059	81.505044	0.0177960	614.3789
254.3104	126.43953	0.02062	8.5925	103.73331	37.22572	250.0142	258.6066	49174.8396	81.596394	0.0177827	614.2807
262.8171	130.22665	0.02024	8.4373	105.97662	37.47803	258.5985	267.0358	51210.8971	81.800765	0.0177934	614.3254
271.1674	134.10159	0.02027	8.2801	108.24480	37.70862	267.0273	275.3075	52914.8978	81.972048	0.0177972	614.3659
279.3622	137.86044	0.01995	8.1304	110.43329	37.92090	275.2970	283.4274	55055.9511	82.135664	0.0177998	614.1366

T	Cp	dcp	DT	totcp	Ecp	Tlow	Thigh	tmid	Htvolt	Htamp	tdelt
K	$J\ K^{-1}\ mol^{-1}$	$J\ K^{-1}$	K	$J\ K^{-1}$	$J\ K^{-1}$	K	K	sec	volt	amp	sec
287.4021	141.62606	0.01949	7.9762	112.61091	38.11891	283.4140	291.3903	57240.6026	82.225912	0.0177860	614.1740
295.3021	145.35597	0.01732	7.8441	114.75785	38.30673	291.3801	299.2242	59051.8918	82.390193	0.0177871	614.2523
303.0719	148.81931	0.01403	7.7083	116.75497	38.48601	299.2178	306.9261	61156.7059	82.473613	0.0177704	614.0759
160.2615	86.61858	0.02097	9.4828	78.15017	32.58128	155.5201	165.0029	26484.0497	73.694890	0.0163648	614.4979
170.1053	90.32574	0.02128	10.2099	80.84666	33.32812	165.0012	175.2094	28664.9264	77.856292	0.0172563	614.3880
180.3224	94.46335	0.02162	10.2278	83.70361	34.00906	175.2085	185.4363	30551.5051	79.374238	0.0175527	614.4750
190.5341	98.65102	0.02110	10.1964	86.50476	34.60872	185.4359	195.6323	32482.2147	80.672950	0.0177965	614.3630
200.5801	102.86479	0.02171	9.8962	89.24541	35.13304	195.6320	205.5282	34484.8659	80.802035	0.0177951	614.2318
210.3470	107.01034	0.02162	9.6292	91.88378	35.59165	205.5324	215.1616	36673.3210	80.950461	0.0177967	614.1420
219.8496	111.19714	0.02151	9.3763	94.49181	35.99824	215.1615	224.5378	38457.4103	81.055472	0.0177929	614.3206
229.1106	115.19958	0.02154	9.1399	96.96133	36.36313	224.5406	233.6805	40349.7561	81.150095	0.0177839	614.0826
238.1650	119.24858	0.02127	8.9572	99.42084	36.69382	233.6864	242.6436	42501.4669	81.423998	0.0178107	614.0667
247.0400	123.15477	0.02140	8.7845	101.77620	36.99505	242.6478	251.4323	44281.0655	81.653622	0.0178269	614.2013
255.7381	127.11533	0.02139	8.6066	104.13308	37.26936	251.4348	260.0413	46062.2463	81.831317	0.0178317	614.1958
264.2580	131.00493	0.02118	8.4292	106.42774	37.51897	260.0434	268.4726	47733.3714	81.953272	0.0178241	614.1412
272.6067	134.82061	0.02078	8.2603	108.66160	37.74683	268.4766	276.7369	49439.9411	82.052261	0.0178118	614.1478
280.8040	138.59447	0.02096	8.1286	110.85642	37.95706	276.7397	284.8683	51626.6816	82.284157	0.0178281	614.2616
288.8626	142.34372	0.02057	7.9900	113.02450	38.15406	284.8676	292.8576	53813.9972	82.458225	0.0178310	614.1975
296.7856	145.90858	0.01823	7.8605	115.08401	38.34141	292.8554	300.7159	55816.8788	82.617666	0.0178303	614.0936
206.9949	105.47472	0.02284	9.3165	90.92519	35.43931	202.3366	211.6532	29807.1392	79.164082	0.0174209	614.2432
216.3945	109.63069	0.02284	9.4821	93.52568	35.85446	211.6534	221.1355	31956.1875	81.078064	0.0178072	614.2422
225.7596	113.65219	0.02235	9.2518	96.01973	36.23437	221.1337	230.3855	33994.7072	81.219517	0.0178086	614.1800
234.8912	117.90225	0.02190	9.0100	98.59676	36.57706	230.3862	239.4040	35738.0924	81.285297	0.0177920	614.2551
243.8177	121.71526	0.02165	8.8207	100.91268	36.88822	239.4074	248.2281	37924.0406	81.450280	0.0177943	614.1516
252.5605	125.54153	0.02128	8.6611	103.20755	37.17150	248.2300	256.8911	40110.1064	81.697448	0.0178138	614.2129
261.1185	129 51700	0.02098	8.4609	105.55528	37.42911	256.8880	265.3489	42001.8359	81.743775	0.0177899	614.1405
269.5018	133.46571	0.02050	8.3018	107.86592	37.66388	265.3509	273.6527	43892.1140	81.928150	0.0177958	614.1968
277.7324	137.15809	0.02040	8.1529	110.02315	37.87964	273.6560	281.8089	45967.2675	82.078535	0.0177952	614.1365
285.8193	140.78173	0.02009	8.0215	112.12919	38.08056	281.8085	289.8300	48152.8646	82.268000	0.0178026	614.1276
293.7674	144.51301	0.01881	7.8833	114.28002	38.27066	289.8257	297.7090	50338.6313	82.407665	0.0177983	614.2313
301.5897	148.19961	0.01590	7.7631	116.39729	38.45232	297.7082	305.4713	52526.8719	82.616236	0.0178082	614.1736
4.5098	0.18691	0.00000	0.0645	0.12046	0.02090	4.4829	4.5448	80026.9206	0.334862	0.0000770	301.1068

T	Cp	dcp	DT	totcp	Ecp	Tlow	Thigh	tmid	Htvolt	Htamp	tdelt
K	$J\ K^{-1}\ mol^{-1}$	$J\ K^{-1}$	K	$J\ K^{-1}$	$J\ K^{-1}$	K	K	sec	volt	amp	sec
4.7572	0.02907	0.00000	0.4230	0.11416	0.02320	4.5448	4.9320	80980.8035	0.833629	0.0001924	301.0757
5.1074	0.19802	0.00000	0.3769	0.13209	0.02669	4.9320	5.2994	81935.7353	0.846392	0.0001953	301.1113
5.5881	0.18889	0.00000	0.5671	0.13261	0.03201	5.3046	5.7200	83659.6354	1.039566	0.0002402	301.1770
5.8706	0.22345	0.00000	0.4394	0.15423	0.03547	5.7200	6.0193	84318.9890	0.987000	0.0002280	301.1499
6.1640	0.36762	0.00000	0.3556	0.23393	0.03935	6.0192	6.3249	84977.5474	1.093224	0.0002526	301.1574
6.5258	0.51745	0.00000	0.4180	0.31795	0.04458	6.3249	6.7317	85634.3112	1.381967	0.0003194	301.0912
7.2192	0.26547	0.00000	0.9786	0.19702	0.05615	6.7317	7.2120	86290.9404	1.663573	0.0003847	301.2401
7.3476	0.79039	0.00000	0.2690	0.47541	0.05853	7.2131	7.4821	547.6767	1.354886	0.0003133	301.2867
7.7719	0.96107	0.00000	0.5611	0.57355	0.06692	7.4913	8.0524	1203.8439	2.149885	0.0004971	301.1211
8.3540	1.21469	0.00000	0.5926	0.71981	0.07982	8.0577	8.6503	1863.1517	2.450518	0.0005669	307.0369
8.9775	1.51703	0.00000	0.6410	0.89444	0.09546	8.6570	9.2980	2524.9795	2.844555	0.0006566	306.9590
9.6474	1.87229	0.00000	0.6972	1.10030	0.11452	9.2988	9.9960	3187.4090	3.289373	0.0007596	307.0183
10.3729	2.31069	0.00000	0.7497	1.35445	0.13815	9.9981	10.7478	3849.4858	3.783993	0.0008741	307.0048
11.1581	2.82192	0.00000	0.8140	1.65310	0.16798	10.7511	11.5652	4511.8370	4.355023	0.0010064	307.0402
12.0068	3.42254	0.00000	0.8809	2.00745	0.20650	11.5663	12.4472	5174.4588	4.991196	0.0011537	307.0846
12.9222	4.12714	0.00000	0.9460	2.42841	0.25696	12.4492	13.3952	5836.8453	5.689339	0.0013151	307.0237
13.9131	4.93107	0.00000	1.0343	2.91714	0.32296	13.3959	14.4302	6498.8640	6.519961	0.0015073	306.9946
14.9852	5.88353	0.00000	1.1147	3.50110	0.40609	14.4278	15.5426	7163.2513	7.409207	0.0017132	307.4695
16.0389	6.84724	0.00000	0.9952	4.09610	0.49434	15.5413	16.5365	7827.6606	7.568682	0.0017499	307.7881
16.9649	7.75575	0.00000	0.8711	4.65122	0.57175	16.5294	17.4005	8491.2769	7.549530	0.0017454	307.4873
17.7893	8.59764	0.00000	0.7882	5.15480	0.63263	17.3952	18.1834	9154.7658	7.561429	0.0017481	307.3988
18.5369	9.33985	0.00000	0.7219	5.62278	0.71033	18.1760	18.8979	9818.4309	7.558241	0.0017473	307.3618
19.3926	10.38880	0.00000	0.9070	6.27177	0.80776	18.9390	19.8461	12149.6078	8.940666	0.0020689	307.5496
20.2646	11.26183	0.00000	0.8307	6.83962	0.91654	19.8493	20.6800	13139.4091	8.941739	0.0020675	307.3182
21.0623	12.19143	0.00000	0.7640	7.43647	1.02458	20.6803	21.4443	14129.6637	8.940320	0.0020671	307.4270
22.6010	13.84541	0.00000	2.3073	8.53842	1.25682	21.4474	23.7547	16194.8616	11.770838	0.0027221	614.8561
24.7678	16.15728	0.00000	2.0406	10.13735	1.64010	23.7475	25.7881	17492.5188	12.064317	0.0027877	615.0727
26.9160	18.26836	0.00000	2.2847	11.69644	2.08911	25.7782	28.0277	18790.4899	13.715429	0.0031680	615.0149
29.3740	21.21125	0.00000	2.6919	13.84349	2.68870	28.0280	30.7199	20860.1798	16.203310	0.0037415	614.6916
32.0776	24.13698	0.00000	2.7176	16.13913	3.44591	30.7188	33.4364	22747.0714	17.650222	0.0040399	615.1077
34.9329	26.77734	0.00000	3.0094	18.41644	4.33484	33.4282	36.4376	24633.7363	19.788419	0.0045540	615.0077
38.0686	29.68881	0.00000	3.2797	20.99566	5.38313	36.4287	39.7084	26520.5508	22.053588	0.0050787	614.8008

T	Cp	dcp	DT	totcp	Ecp	Tlow	Thigh	tmid	Htvolt	Htamp	tdelt
K	J K^{-1} mol^{-1}	J K^{-1}	K	J K^{-1}	J K^{-1}	K	K	sec	volt	amp	sec
41.5159	32.57419	0.00000	3.6061	23.71956	6.58980	39.7129	43.3190	28405.9846	24.603329	0.0056600	614.2349
45.2968	35.33233	0.00000	3.9860	26.54345	7.96338	43.3038	47.2898	30291.5428	27.402733	0.0062856	614.2535
49.4294	38.32241	0.00000	4.3660	29.67542	9.52308	47.2464	51.6124	32177.5535	30.364239	0.0069440	614.4742
53.9947	42.02904	0.01379	4.7991	33.39037	11.27518	51.5952	56.3943	35051.1349	33.819839	0.0077097	614.5666
64.8344	48.46742	0.01428	6.0506	40.70295	15.20178	61.8091	67.8597	33594.5033	42.037682	0.0095353	614.4051
70.9481	51.54960	0.01852	6.1339	44.37587	17.24975	67.8811	74.0150	35774.1908	44.229064	0.0100174	614.3503
77.4262	54.52666	0.01753	6.7872	47.93731	19.24676	74.0326	80.8199	37953.9690	48.402707	0.0109406	614.4068
84.5134	57.57633	0.01961	7.3561	51.51449	21.21825	80.8353	88.1914	40133.9306	52.271013	0.0117964	614.5644
92.2614	60.68337	0.02124	8.1128	55.05772	23.12607	88.2050	96.3178	42313.7293	56.821135	0.0127968	614.2946
100.8061	64.05539	0.02232	8.9600	58.66188	24.95605	96.3261	105.2861	44492.7634	61.717621	0.0138622	614.3627
110.0015	67.54066	0.02314	9.4052	62.18589	26.64658	105.2989	114.7041	47864.3447	65.178478	0.0146039	614.4551
119.4049	71.07975	0.02418	9.3936	65.52702	28.12571	114.7081	124.1017	50047.5661	66.931214	0.0149659	614.4994
128.8363	74.59263	0.02406	9.4600	68.65020	29.40183	124.1063	133.5663	52229.6600	68.784888	0.0153490	615.1169
138.2458	78.12092	0.02379	9.3497	71.60923	30.50585	133.5709	142.9206	54417.0406	69.944950	0.0155802	614.3781
147.6544	81.71675	0.02377	9.4638	74.46431	31.47014	142.9225	152.3863	56598.2334	71.824822	0.0159684	614.4356
157.5436	85.47454	0.02391	10.0879	77.32769	32.35743	152.4997	162.5876	31502.4662	75.671537	0.0167794	614.3661