

DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

**Thermistor Calibrations, Techniques, and Computer Programs
1968–1989**

by

T. C. Urban¹ and W. H. Diment¹

Open-File Report 90-216

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature. Any use of trade names is for descriptive purposes only and does not imply endorsement by the USGS.

Although the programs have been extensively tested, the U.S. Geological Survey cannot guarantee that they will give accurate results for all applications or that they will work on all computer systems. No warranty, expressed or implied, is made by the USGS as to the accuracy and functioning of the programs and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

¹ U.S. Geological Survey, Denver, Colorado 80225

TABLE OF CONTENTS

ABSTRACT	1
INTRODUCTION	2
THERMISTORS	3
Drift	3
Probe and cable assembly	4
Thermistor calibration	6
Thermistor Calibration Programs	11
TABLE 1	15-16
ILLUSTRATIONS	17-21
Figure 1	17
Figure 2	18
Figure 3	19
Figure 4	20
Figure 5	21
REFERENCES	22
APPENDIX I	23
Program THERMISTOR	26
APPENDIX II	36
Program THMCAL - PL/1	37
APPENDIX III	40
Program THMCAL - BASIC	42

Thermistor Calibrations, Techniques, and Computer Programs 1968–1989

by

T. C. Urban and W. H. Diment

ABSTRACT

Thermistors are semiconducting devices composed of sintered oxides that are widely used either to measure temperatures or for temperature compensation. In our application, glass-encapsulated beads have been used in single probes and multielement cables to make precision temperature measurements in boreholes. Laboratory calibrations of the thermistors ranged from 0 °C to a maximum of 240 °C. The nominal resistance of these beads at 25 °C ranged from 2 to 500 kohms, depending upon the maximum expected temperature and the sensitivity required. All of these beads have a negative temperature coefficient of about $-4\%/^{\circ}\text{C}$ and have been calibrated to a relative accuracy of $\pm 0.001^{\circ}\text{C}$ and an absolute precision of 0.02°C over the temperature range of interest. Although thermistors are sensitive, they drift $0.003 - 0.006^{\circ}\text{C}/\text{month}$, usually in the direction of apparently lower temperatures. This drift necessitates repeated calibrations since temperature changes of 0.1°C or less in boreholes over time may be significant. In order to minimize the drift, the beads are obtained in quantity and selected on the basis of few or no bubbles in the glass or on the bead and low DC noise (1 or 2 μvolts). For a given quantity, typically 100, 10–15% of the beads will be selected given these criteria. The probes or cables are assembled and aged for several days over the temperature range for which they were designed. The probes or cables are then calibrated over the desired temperature range. The resistance (R) and temperature (T) are related by

$$T = \frac{C1 + C2 \times \ln R}{C3 + \ln R}$$

where C1, C2, and C3 are constants and ln is the natural logarithm. The constants are determined by a nonlinear least-squares fit to the equation. Examples are given and computer codes provided for statistical analysis of the data. Details of design of some probes and calibration systems are also given.

INTRODUCTION

In recent years it has become increasingly evident that precision temperature measurements repeated over time in boreholes can reveal useful information in such diverse areas as: climatic change, temporal changes of ground-water flow, and evolution of geothermal systems. The first requirement is that boreholes either be permanently implanted with temperature sensors or filled with penetrable fluid so that repeated temperature logs can be made. It is the latter case that we address here. The second requirement is that the temperature sensors are adequately calibrated at appropriate intervals of time, with respect to standards (traceable to the National Institute of Standards and Technology) over the temperature range of interest so that comparison of temperatures made over years is possible even when different instruments are used. Other requirements such as allowance for cable stretch are not discussed here.

Thermistors have been used for almost 50 years. Their initial widespread use was for temperature compensation in telephone circuits (Sillars, 1942; Robertson, *et al.*, 1966). In our application, we have used thermistors as precision temperature sensors to measure temperatures in boreholes for heat-flow, geothermal, and earthquake studies. Although thermistors are manufactured in a wide variety of forms, we have used glass-encapsulated beads in our probes. The bead assemblies, including the glass encapsulation, are small (0.1" outside diameter by 0.5+" long), quite rugged and not subject to chemical oxidation. Robertson, *et al.* (1966) reviewed the manufacturing process for thermistors, and their properties, drift, and equations used to relate resistance to temperature. The pressure effect on glass-encapsulated thermistors is negligible (Diment and Robertson, 1963). We will not repeat that information but will supplement it with some of our experiences with thermistors and the techniques we use to calibrate the probes and cables.

THERMISTORS

Drift. Thermistors generally have much higher temporal drift rates than do other types of temperature sensors. Frequent calibrations and drift corrections for data obtained between calibrations can usually compensate for the increase in resistance with time. However, some thermistors exhibit abnormally high drift rates between calibrations. Such abnormal drift rates can usually be attributed to mechanical or thermal shock. Since most of our thermistor probes are assembled from glass-encapsulated beads, the damage usually takes the form of a cracked glass shell and a chipped or broken bead. In the extreme case, the result is an open circuit which is obvious. In less catastrophic cases, an offset in resistance and, therefore, temperature occurs. This offset may or may not be noticeable in a temperature log since the damage may have happened before or after logging a hole.

Another form of step or sudden change in the calibration may be due to the thermal stresses on the bead during logging or calibration of the probe. We have noted that some glass-encapsulated beads contain bubbles in the glass coating the bead and the bead wires. For an undamaged bead without bubbles, the DC voltage measured across the bead without any source of power applied to the bead should be zero to a few microvolts. The small voltages appear to be generated by noisy lights, air conditioners, motors, and other unknown sources. For those beads with bubbles, the DC noise level can be, and usually is, significantly higher, on the order of tens of microvolts or even millivolts. If the objective is to measure temperatures to 0.001°C (this requires a resistance measurement at the temperature accurate to 0.1 ohms) then, for a $10\ \mu\text{A}$ current through the bead, the voltage difference for 0.001°C is $1\ \mu\text{V}$. If the DC noise level is on the order of $10\ \mu\text{V}$, the "noise" level of the probe is about 0.01°C . For higher noise-induced voltages, the errors become proportionally worse. We do not know the composition of the bubbles, but assume that they are either gas or fluid inclusions produced in the manufacturing process. Extraneous voltages could produce reactions that could cause weak transient currents. A more extreme case is a large bubble on the surface of the thermistor bead. Aside from being a possible DC noise source, the large bubble can be a stress point on the surface of the bead. We have observed, using a low-power microscope, material in bubbles that has apparently spalled from the wall of the bead. Gradual or sudden spalling of a large piece into a bubble could be a source of high drift rates or sudden shifts in calibration. The number of useable beads varies considerably from one lot to another. The simple solution to this problem is to

select beads with few bubbles and low DC-voltage noise levels. This selection process is simple if the thermistor probes are assembled in-house. However, if the assembled probes are purchased, the selection process must be specified and can add significantly to the cost of the probes. Depending upon the desired accuracy, the additional cost may not be warranted. If, however, repeated, high-precision temperature measurements are needed for the long-term evaluation and observation of changes in the thermal regime, the extra selection process is worth the expense.

Probe and cable assembly. Thermistor beads are quite rugged. However, in order to lower them into drill holes, the beads are either incorporated into multiconductor cables or installed in a probe which is then attached to the end of a cable. Raspet, *et al.* (1966) described the construction of multielement cables and we have generally followed that construction practice. For single thermistor probes, there are as many designs as there are investigators. We present one design which we have successfully used in wells with maximum temperatures of about 210°C and to total depths of about 7000 ft (≈ 2 km).

Figure 1 illustrates the basic design of the probe that we have used for 15 years. Variations in the design for high temperature and high pressure include adding a second o-ring and changing the 1/4-28 thread to 5/16-32 in order to add strength. Probes are constructed from either 416 stainless steel or brass. The 416 stainless is easy to machine and has good corrosion resistance. Probes made of brass are easily machined and work quite well in most "clean" boreholes, although in some environments (H₂S for example), the brass corrodes quite severely.

The thermistor bead was epoxied into the bottom of the well in the probe with Stycast 2642FT¹. Electrical connections were provided by two Teflon² insulated wires that protrude from the o-ring end of the probe. The void around the wires was, also, filled with Stycast 2642FT. The filling of the small void around the wires was facilitated by heating the Stycast to reduce viscosity. Heating of the outside of the probe helped increase the flow down the probe well and also aided in expelling trapped air bubbles from the Stycast. With practice all of the air can be removed from the well. Curing of the epoxy proceeds in stages, each lasting about 24 hours. For the high-temperature probes, the temperatures at each stage are 75°, 150°, and 225°C, with a minimum cure at 75°C for several days for the

¹ Stycast 2642FT is a trademark of Emerson and Cuming, Inc.

² Teflon is a trademark of E.I. du Pont de Nemours & Company.

low-temperature probes. The exact temperatures are not critical, only that they cover the operating range of the probe. Stycast 2642FT is used because it has high thermal conductivity ($1.34 \text{ W/m}\cdot\text{K}$), low electrical conductivity ($10^{16} \text{ ohm}\cdot\text{cm}$ at 21°C), and a thermal expansion (38×10^{-6} per $^\circ\text{C}$) close to that of copper (Emerson and Cuming, 1974). Its adhesion to metals and glass is excellent. We have used Stycast 2642FT for a number of years although other high-temperature epoxies would probably work as well.

Attachment of any sensor to the end of a cable is, and always will be, a problem. Dissimilar materials are involved and cable construction varies across a wide range of materials, strength, and sealing properties. An adapter fits between the cable and the probe to aid in assembly and to facilitate sealing of the electrical connections. In our truck, the cable end is terminated by an industry standard 1-in outside diameter, 4-conductor cablehead manufactured by Mineral Logging Systems of Ft. Worth, Texas. With slight variations, the same cablehead is available from several different manufacturers. Since the purpose of the cablehead is to allow several different tools to be easily attached to the cable, the adapter discussed above is split into the cablehead plus adapter. An example of such an adapter is shown in figure 2. A completely assembled probe and adapter is illustrated in figure 3. Although it would be possible to combine the pieces in figures 1 and 2 into one solid piece, thereby eliminating one o-ring connection, epoxying the bead into the probe would become much more difficult if not impossible without special tools and setups.

The wires emanating from the probe pass up the length of the adapter and are terminated on the 4-pin connector (see fig. 2) at pins 2 and 3. Pins 1 and 2 and pins 3 and 4 are jumpered to be common. In this manner a 4-wire resistance measurement can be made whereby a known current is conducted down the cable in two of the leads and the resulting voltage drop across the thermistor is measured with the other two leads. Although the 4 wires are joined into two leads several centimeters from the thermistor bead, the electrical resistance of those few centimeters is negligible. Modern multimeters can compensate for lead resistances of 1000 ohms or more. Cable resistance is about 8 ohms/100 m. Therefore, a 3 km (10,000 ft) cable has a lead resistance of only 240 ohms, well within the maximum lead resistance requirements of most multimeters.

The final piece in the construction of a thermistor probe is the o-ring. For low temperatures, o-rings constructed of a number of different materials will work if they are compatible with water, grease, drilling mud, and other substances found in drill holes. For high temperatures ($> 100-150\text{ }^{\circ}\text{C}$), we have used Viton³ o-rings with good success in most environments. Prolonged exposure to high temperature does, however, cause the o-rings to take a "set" into a roughly rectangular (rather than their normal circular) cross-section. This can lead to seal failure and loss of the probe in the sense that fluids contaminate the connectors, causing electrical noise and erratic measurements. Inspection of the exterior o-rings on the cablehead before and after each log is a requirement, especially in high temperature holes. After logs are obtained in very high-temperature, deep holes, the probe and cablehead need to be disassembled and inspected for o-ring deterioration or failure. Failure of the o-ring on the connector in the cablehead can be costly, not only in the loss of the connector, but reheading the cable is time consuming. The cost of machining a probe and adapter, assembly, and calibration is about \$1000. The cost of the o-rings is insignificant in comparison and routine inspection and replacement is more than worth the effort. Finally, as an extra precaution, the "o-rings" that we use between the probe and adapter are Quad-Ring Seals⁴. A Quad-Ring can be thought of as two o-rings moulded together so that the cross-section resembles that of a four-leaf clover. The advantage of this type of seal is that one of the lobes can be damaged and the seal will continue operating as if a single o-ring were present.

For cables that are not terminated by a cablehead, the adapter and cablehead are combined into a single unit and the whole assembly is usually permanently attached to the end of the cable. Sealing in this arrangement, as well as in the cablehead, is by a grease seal against the individual conductors. If a waterproof jacket surrounds the conductors in the cable, then a rubber moulding can be cast to seal the jacket to the adapter. Since the variations in this termination are numerous, the choice as to the method of attaching the probe (fig. 1) to the cable will be simply left as a generic "adapter".

Thermistor calibration. Once the thermistor probe and cable have been assembled and a suitable instrument, such as a multimeter, for measuring the thermistor resistance is in place, the next step is to relate the measured resistance (R) to the temperature (T). The functional relationship will be discussed shortly. For now, we will assume such a function relating T to R exists and all we simply have to do is to measure them.

³ Viton is a trademark of E.I. du Pont de Nemours & Company.

⁴ Quad-Ring is a trademark of Minnesota Rubber Company.

Twenty years ago thermistor resistances were measured with a 4- or 5-dial Wheatstone bridge and electronic null detector. Current was limited to about $150\ \mu\text{A}$ to limit self-heating. In recent years, 5-1/2 or 6-1/2 digit multimeters have replaced Wheatstone bridges for these measurements. Multimeters can measure resistance with the 4-wire technique described above. The 5-1/2 digit multimeters we use are Data Precision models 2500 or 3500 and the 6-1/2 digit instruments are Fluke 8502A multimeters. Depending upon the resistance to be measured, different range settings are used, each with a correspondingly different current. Unlike the Wheatstone bridge where the current is usually fixed over the entire range of resistances to be measured, the source current from the multimeters varies from $1\ \mu\text{A}$ to $100\ \mu\text{A}$ or more. Current steps are a factor of 10 between ranges for the 2500/3500 models. The current steps for the 8502A are more complex. We selected the 8502A for the low current ($9.8\ \mu\text{A}$) in the 100 Kohm range that most of our measurements were to be made. This was specified in order to minimize self-heating in the single bead thermistor. For the 8502A, the 100K range over-ranges at 256,000 ohms and resistances are measured to 0.1 ohms. Although similar multimeters from other manufacturers may be quite suitable for these measurements, the specifications for each instrument need to be examined quite closely. We have found that, for example for a 6-1/2 digit multimeter on the 100K range, the current may be 10 or 100 times higher than the 8502A depending upon the manufacturer of the instrument and the purpose for which it was intended. While for a current of 1 ma the bead will not exactly “glow in the dark”, errors on the order of several degrees will be introduced by self-heating. An estimate of the amount of self-heating can be made based upon the current and the resistance at a given temperature. A typical bead will have a dissipation constant in stirred water of $5\ \text{mW}/^\circ\text{C}$. The power generated is I^2R . Therefore, for a current of $10\ \mu\text{A}$ and a resistance of 100,000 ohms, the temperature rise ($\Delta T = I^2R/5$) would be 0.002°C . However, if the current were $100\ \mu\text{A}$, then $\Delta T = 0.2^\circ\text{C}$. If in the unusual case that the current through the thermistor was 1 mA, then a temperature rise of 20°C would be observed. Although all of these currents are low, the effect of self-heating can be quite dramatic due to the small bead size and low dissipation constant.

The measurement of the temperature corresponding to the measured thermistor resistance is derived from a temperature standard. Secondary temperature standards, until recent years, were primarily limited to platinum thermometers. Because of their low resistance (25 ohms at 0 °C), Mueller bridges were used to measure their resistance. Temperatures could be interpolated from these readings to 0.001 °C. Electronic thermometers had a variety of sensors but were only calibrated to 0.1 °C. These thermometers were not intended as secondary standards. A secondary standard was developed by Hewlett-Packard with their 2801A Quartz Thermometer. We used the 2801A for calibrations from 1968 until 1986 when it was replaced by a Guildline 9540 Platinum Resistance Thermometer. When used for temperature calibrations, these instruments are interfaced to a computer, which has resulted in a semi-automatic calibration of the thermistors.

Our calibration techniques have changed only slightly over the years. Initially we strapped the quartz thermometer to the temperature probe and immersed the assembly in a well stirred water or oil bath, depending upon the temperature range to be covered in the calibration. Readings of resistance and temperature were observed on the 2500 multimeter and the 2801A thermometer and recorded manually. Starting in 1978 the recording process was automated by the addition of a Tektronix 4051 Graphics Computer System, which input the resistance and temperature and printed the values. Ten readings were taken at a given temperature; one reading every 30 s. The temperature in the bath was then raised 10 °C and the process repeated. At each 10 °C temperature step the bath was allowed to increase in temperature by slowly heating the fluid. The total temperature rise was on the order of 0.5–1.0 °C. We found that this slight change in temperature had an advantage over recording a single resistance at constant temperature in that we could (1) calculate statistical parameters for the equation constants, and (2) prevent undue weight from being assigned to a specific temperature. By the latter we mean that if we only recorded one resistance and temperature at a given temperature step and used it ten times in determining the equation constants, then any significant error in either the resistance or temperature would have a significant impact on the equation constants. However, recording ten resistance-temperature pairs over a 0.5–1.0 °C range reduced the impact of one bad reading.

In 1979 we obtained the Fluke 8502A multimeters. These instruments are 6-1/2 digit meters which were a significant improvement over the 5-1/2 digit Data Precision 2500 in that the current was reduced from $100\ \mu\text{A}$ to $10\ \mu\text{A}$, in the range with a resolution of 0.1 ohms, while extending the over-range from 12,000 ohms to 265,000 ohms. Thus the effect of self-heating was reduced substantially while maintaining resolution. In 1986 the H-P Quartz Thermometer was replaced with the Guildline 9540 Platinum Thermometer.

The calibration of the thermistor probe involves setting a temperature and recording a resistance which is then fitted to an equation by a non-linear least-squares fit, discussed below. The starting temperature, $0\ ^\circ\text{C}$, is the ice point, which is obtained by stabilizing the temperature of a mixture of crushed ice and water in a Dewar flask. For best results we have used distilled water for both the ice and water. Tap water contains numerous minerals, gases, etc. and can cause an unstable ice point which can change with time. A stable ice point is useful for detecting deterioration in the probe. At a constant temperature, the resistance of a "bad" probe will drift, sometimes erratically, or be very noisy. Thus, the ice point offers a good, quick check on the stability of the probe. As at all temperatures, ten readings over five minutes are obtained.

Temperatures from 10° to 80°C are obtained in a water bath. Below room temperature ($\sim 22^\circ\text{C}$), the bath must be refrigerated. We have used a Lauda K2/R bath as a source of temperature-controlled water from 10° – 80°C . The water from the K2/R is circulated through a coil in an external container (fig. 4). The original external container was a battery jar. We recently replaced the battery jar with a Thermo-Cup⁵. The water in the external container is stirred vigorously with a mechanical stirrer. It is in this external container that the probe/thermometer assembly is immersed. Immersion directly into the K2/R is not practical due to the cycling of the heater and resultant sharp temperature fluctuations. In the external container, however, these spikes in temperature are damped and a smoother temperature rise is obtained. Again the temperature rise is limited to about 0.5 – 1.0°C during the duration of the test. This slow rate of increase (0.1 – $0.2^\circ\text{C}/\text{min}$) allows for the slight difference in the time constants of the probe ($\sim 2\ \text{s}$) and the quartz ($\sim 3\ \text{s}$) or platinum ($\sim 5\ \text{s}$) thermometers while still giving an acceptable temperature span for statistical analysis. Because of the thermal inertia of the system, raising the temperature from one ten-degree step to another is very time consuming. We installed 500 watt

⁵ Thermo-Cup is a trademark of Lab-Line Instruments.

heaters in the external container and the K2/R and only apply power to them to raise the temperature in each bath to 1.5°C less than the next temperature step and then turn the power off. For example, if the next temperature step was 40°C , then we apply power to the auxiliary heaters until the temperature in the two baths is $\sim 38.5^{\circ}\text{C}$. This allows for a rapid increase to the next step while the 1.5°C interval allows for the heaters to cool to the bath temperature.

For temperatures above 80°C , we use a Blue-M Bath with UCON HTF-100 Heat Transfer Fluid. The bath is manufactured with a pulse plate activated by an electromagnet for stirring the fluid. We found, however, that the high viscosity of the heat-transfer fluid at low temperatures ($< 90^{\circ}\text{C}$) prevents thorough mixing by the pulse plate. We, therefore, removed the pulse plate and electromagnet and installed a mechanical stirrer. The stirrer allows ample mixing at all temperatures, thus preventing the formation of "hot spots" in the fluid.

A further modification to the Blue-M Bath consists of incorporating a variable transformer in parallel with the thermostat controlling the temperature of the bath. A single-pole, double-throw switch (SPDT) selects either the variable transformer or the thermostat. The thermostat is used initially to raise the temperature of the bath to $\sim 85^{\circ}\text{C}$. The variable transformer is then selected and set to allow a gradual rise in temperature, similar to that of the water bath described above. The thermistor-calibration procedure for the oil bath is identical to that of the water bath. The temperature is set at a level ($\sim 90^{\circ}$, 100° , ...) and the resistance and temperature recorded for ten readings at 30 s intervals as the bath temperature slowly rises about $0.5\text{--}1.0^{\circ}\text{C}$. The setting on the variable transformer increases about 2–3% of full scale for each stage as the temperature is raised from 90° to 240°C . That is, the setting at 120°C might be 40, at 130°C it would be 42, etc. At some temperature stage, the setting would be increased by 3% instead of 2% in order to maintain the $0.5\text{--}1.0^{\circ}\text{C}$ rise over 5 minutes. Some art is involved in this in that the temperature of the bath is a function of the heat input into the fluid from the heating coil and the heat loss from the fluid by conduction, convection, and evaporation. Hence, variations in the AC line voltage can cause variations in the heating rate of the fluid and must be compensated by adjustments to the variable transformer. We recently installed a 1 KVA constant-voltage transformer with improvement in the smoothness of the power curve.

The oil bath has always been installed in an enclosure with an exhaust fan. Fumes from the evaporating oil are neither flammable nor toxic but can be unpleasant as they accumulate in a room. A small room with an exhaust fan sufficed to control the fumes for many years. In 1986 we moved into a new facility and installed a fume hood for the bath. The fume hood is more compact and easier to adjust for proper air flow around the bath. The latter affects the heat-exchange rate of the fluid. A very rapid air flow causes a high heat loss and requires a much higher heating rate. Because of the viscosity of the oil, a high heating rate can cause hot spots to form in the fluid.

Thermistor Calibration Programs. The resistance-temperature pairs were initially recorded manually. This invariably leads to a few errors as digits are inverted or mistakenly recorded. The introduction of the table-top computer/controller changed this by allowing for automatic data acquisition. In 1979, a Tektronix 4051 Graphic Computer System was set up to record the calibration data at each temperature stage. The program THERMISTOR (Appendix I) in the BASIC language was written to input and print the data. Since the 4051 has a high resolution graphics capability equivalent to a Tektronix 4010 terminal, an additional part of the program plots the data. The data is automatically scaled to fit the screen. Aside from the statistical analysis, the 0.5–1.0 °C span of the data at each temperature stage can be examined. Over this temperature range, the plot of resistance versus temperature is approximately linear, even though over a much larger range the relationship is decidedly nonlinear. “Bad points” appear as marked departures from linearity. The causes of these bad points vary from one bad point to another. They appear to be due to (1) variations in the time constants of the probes and temperature sensors (standards), (2) power-line glitches, (3) static electricity, (4) local radio station interference, (5) contact resistance and noise generation in the connectors, (6) power variations in the heaters causing rapid changes in the heating curve, and (7) unidentified causes. Calibrations appear to be better on weekends and there are fewer bad points, suggesting many of the problems are cultural in nature. The number of bad points varies but generally there are usually one or two out of ten readings. By plotting R versus T, these points may be identified and replaced with new data, which are again checked. In this manner spurious points can be eliminated immediately. It should be noted that many of the points that are discarded by this selection process could possibly be retained since they would in many cases be within the $\pm 0.02^\circ\text{C}$ absolute error. However, by eliminating them better calibrations can be achieved.

Once the calibration data were obtained, they were entered into a mainframe computer and the data were fit to the thermistor equation (2) using a general least-squares PL1 program, THMCAL (Appendix II). In 1981 THMCAL was rewritten in 4051 BASIC (Appendix III) and this is the version that we currently use.

Although many different equations have been used over the years by various investigators (see Robertson, *et al.*, 1966 for a discussion), we selected the following equation (Robertson, *et al.*, 1966, eq. 4) because we were able to obtain a better correspondence with the data and semiconductor theory:

$$\ln R = A + \frac{B}{T + C} \quad (1)$$

where R is the resistance in ohms, ln is the natural logarithm, A, B, and C are constants, and T is the temperature. T can be in degrees Celsius or Kelvin with the constant C adjusted by 273.15. Since we measure resistance with a multimeter and the temperature is the unknown, we solve equation 1 for T and combine some of the constants to yield:

$$T = \frac{C1 + C2 \times \ln R}{C3 + \ln R} \quad (2)$$

where

$$C1 = A \times C + B \quad (3)$$

$$C2 = -C \quad (4)$$

$$C3 = -A. \quad (5)$$

The least-squares method we use is based on the assumption that the observed values are normally distributed about the true values (e.g., Wolberg, 1967). Other assumptions, however, may be used. The method attempts to determine C1, C2, and C3 so as to minimize the residuals between the true and apparent values. Because the equations are nonlinear, the normal equations can not be solved directly for C1, C2, and C3. Therefore, the method of successive approximation (Hald, 1952; Wolberg, 1967) is used whereby equation 2 is transformed into a linear function of corrections to C1, C2, and C3 (see Wolberg, 1967).

The weighted sum of the squares of the residuals is denoted by S:

$$S = \sum_{i=1}^n [w_{t_i} \hat{R}_{t_i}^2 + w_{r_i} \hat{R}_{r_i}^2] \quad (6)$$

where $w = \frac{1}{\sigma^2}$, σ is the uncertainty in R or T, $\widehat{R}_{t_i} = T_i - t_i$, $\widehat{R}_{r_i} = R_i - r_i$, T_i is the measured temperature, t_i is the calculated temperature, R_i is the measured resistance, and r_i is the calculated resistance. In this case temperature is the dependent variable and resistance is the independent variable (eq. 2). The conditional function F^i is defined as

$$F^i = t_i - f(r_i; C1, C2, C3) \quad (7)$$

where the temperature function, f , is equation 2. If $C1$, $C2$, and $C3$ are the true values, then

$$F^i = 0 \quad \text{for } i = 1, \dots, n. \quad (8)$$

Derivatives of F are denoted by subscripts

$$F_{r_i}^i \equiv \frac{\partial F^i}{\partial r_i}, \quad F_{t_i}^i \equiv \frac{\partial F^i}{\partial t_i} = 1, \quad F_{C_k}^i \equiv \frac{\partial F^i}{\partial C_k}. \quad (9)$$

where $k=1,2,3$. Let C_{k_0} be the initial guess of C_k . The choice of the initial guess can be very critical in nonlinear cases. An inappropriate value for C_k can lead to slow convergence or even divergence in the solution. Fortunately, equation 2 converges quite rapidly for an initial $C1_0 = 3000$, $C2_0 = -300$, and $C3_0 = 5$.

If F_0^i is the estimated value of F^i , then

$$F_0^i = T_i - f(R_i; C1_0, C2_0, C3_0) \neq 0. \quad (10)$$

These values of F_0^i can be related to the derivatives of F^i , the residuals and the differences between C_{k_0} and C_k by using Lagrange multipliers. The details of the derivation may be found in Wolberg (1967, p. 39-43).

The resulting set of equations are

$$\left(\begin{array}{c} \left[\frac{F_{C1} F_{C1}}{L} \right] \\ \left[\frac{F_{C2} F_{C1}}{L} \right] \\ \left[\frac{F_{C3} F_{C1}}{L} \right] \end{array} \right) \left(\begin{array}{c} \left[\frac{F_{C1} F_{C2}}{L} \right] \\ \left[\frac{F_{C2} F_{C2}}{L} \right] \\ \left[\frac{F_{C3} F_{C2}}{L} \right] \end{array} \right) \left(\begin{array}{c} \left[\frac{F_{C1} F_{C3}}{L} \right] \\ \left[\frac{F_{C2} F_{C3}}{L} \right] \\ \left[\frac{F_{C3} F_{C3}}{L} \right] \end{array} \right) \left(\begin{array}{c} A_1 \\ A_2 \\ A_3 \end{array} \right) = \left(\begin{array}{c} \left[\frac{F_{C1} F_0}{L} \right] \\ \left[\frac{F_{C2} F_0}{L} \right] \\ \left[\frac{F_{C3} F_0}{L} \right] \end{array} \right) \quad (11)$$

where $A_k = C_{k_0} - C_k$, the brackets, $[]$, denote the summation over i , and the subscripts and superscripts, i , have been dropped. The weights, L , are

$$\begin{aligned} L_i &= \sigma_{T_i}^2 + (F_{R_i}^i \cdot \sigma_{R_i})^2 \\ &= \sigma_T^2 + \sigma_R^2 \cdot \frac{(C1 - C2 \cdot C3)^2}{R^2 (C3 + \ln R_i)^4}. \end{aligned} \quad (12)$$

The derivatives are

$$F_{C1} = \frac{\partial F}{\partial C1} = -\frac{1}{C3 + \ln R_i} \quad (13)$$

$$F_{C2} = \frac{\partial F}{\partial C2} = -\frac{\ln R_i}{C3 + \ln R_i} \quad (14)$$

and

$$F_{C3} = \frac{\partial F}{\partial C3} = \frac{C1 + C2 \cdot \ln R_i}{(C3 + \ln R_i)^2} \quad (15)$$

The corrections, A_k , are obtained by solving the set of simultaneous equations (eq. 11). These corrections are then added to the initial estimates to obtain new values for the equation constants. The calculations are repeated until the corrections fall below some predetermined value. Details of this method are available in a number of texts of which Wolberg (1967) is a good example. The THMCAL programs use this method to estimate C1, C2, and C3 and also provide some statistical analysis of the constants by calculating their standard deviations (Wolberg, 1967, p. 54-62):

$$\text{Variance} \equiv \sigma_{Ck}^2 \approx \frac{S}{n-3} E_{kk}^{-1} \text{ for } k = 1, 2, 3 \quad (16)$$

where S the weighted sum of the squares of the residuals, $n-3$ is the number of degrees of freedom, and E_{kk}^{-1} are the diagonal elements of the inverse of the square part of the matrix in equation 11. The standard deviation of the constants is the square root of equation 16. The number of iterations required to reduce the corrections to less than 1×10^{-6} is five. Hence, the convergence is quite rapid. If the number of iterations exceeds six, there is usually a significant error in one or more of the data points. We use a 30 °C temperature range and 40 points in the calculations, although on occasion we have obtained good fits up to a range of 100 °C and 110 points.

The computer codes in Appendices II and III deserve a few comments to clarify machine-dependent functions and some of the subroutines. In the PL1 version (Appendix II), THMCAL uses Crout's method (Nielsen, 1964, p. 185-188) in procedure "crmes" for inverting the matrix. The BASIC program (Appendix III) uses a built-in function INV to calculate the inverse of the matrix and DET to calculate the determinant. The algorithm is based on the Gauss-Jordan technique, performed in place, with full row and column pivoting (Tektronix, 1976a). Most of these programs are quite standard. However, machine-dependent routines would need to be modified, especially in the 4051 BASIC, to run on another system. An example of the output of THMCAL is listed in Table 1.

Table 1a. Example of output from THMCAL (Appendix III) listing the calibration data. The nominal temperature range is for reference only. The calibration resistance range is the range of resistances over which the calibration constants (e.g., Table 1b) are considered valid.

THERMISTOR CALIBRATION: PROBE 500-02 - 11 July 1989

NOMINAL TEMPERATURE RANGE: 0 - 30 °C

CALIBRATION RESISTANCE RANGE: 425513 - 2107375 OHMS

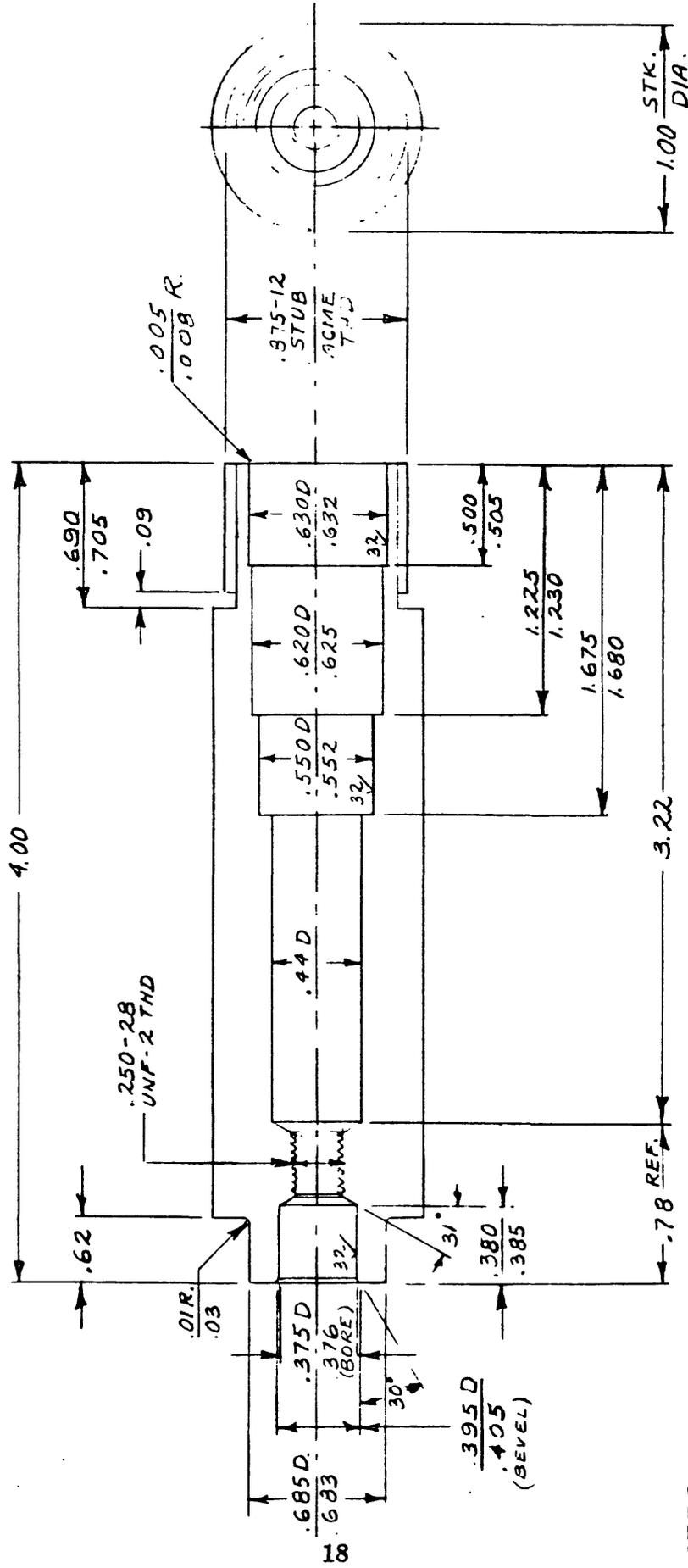
THERMISTOR RESISTANCE (OHMS)	OBSERVED TEMPERATURE (°C)	CALCULATED TEMPERATURE (°C)	TEMPERATURE DIFFERENCE (°C)
2107580	0.017	0.024	-0.007
2107508	0.017	0.025	-0.008
2107458	0.023	0.025	-0.002
2107412	0.023	0.026	-0.003
2107375	0.025	0.026	-0.001
2107342	0.027	0.026	0.001
2107294	0.031	0.027	0.004
2107273	0.033	0.027	0.006
2107261	0.033	0.027	0.006
2107249	0.033	0.027	0.006
1192166	10.209	10.210	-0.001
1190138	10.243	10.242	0.001
1187575	10.283	10.281	0.002
1184098	10.335	10.335	0.000
1180043	10.398	10.398	0.000
1175622	10.466	10.467	-0.001
1170918	10.542	10.541	0.001
1166020	10.618	10.619	-0.001
1160966	10.696	10.699	-0.003
1155882	10.778	10.780	-0.002
709960.0	20.003	20.004	-0.001
707600.0	20.067	20.068	-0.001
703157.0	20.188	20.191	-0.003
701021.0	20.252	20.250	0.002
698926.0	20.310	20.308	0.002
696881.0	20.366	20.365	0.001
694860.0	20.424	20.421	0.003
692858.0	20.478	20.477	0.001
690871.0	20.532	20.533	-0.001
688118.0	20.612	20.611	0.001
429471.0	30.020	30.022	-0.002
428700.0	30.060	30.059	0.001
427917.0	30.098	30.096	0.002
426323.0	30.174	30.173	0.001
425513.0	30.212	30.212	0.000
424720.0	30.250	30.250	0.000
423932.0	30.286	30.288	-0.002
423149.0	30.326	30.326	0.000
422351.0	30.364	30.364	0.000
421568.0	30.400	30.403	-0.003

Table 1b. Example output from THMCAL (Appendix III) listing calibration constants and their standard deviations. See text and Table 1a.

THERMISTOR CALIBRATION: PROBE 500-02 - 11 July 1989
NOMINAL TEMPERATURE RANGE: 0 - 30 °C
CALIBRATION RESISTANCE RANGE: 425513 - 2107375 OHMS

THERMISTOR EQUATION: $T = \frac{C1 + C2 \times \ln(R)}{C3 + \ln(R)}$

C1 = 5111.5764	S. DEV. C1 =	9.0408
C2 = -351.0106	S. DEV. C2 =	0.6215
C3 = 5.6447	S. DEV. C3 =	0.0329



NOTES -

1. .875-12 STUB ACME THD.
 PITCH = .0833 THD HT = .0300
 MAJOR DIA. = .875/.870
 MINOR DIA. = .815/.801
2. TOLERANCE X DIMS = ± .010 IN.
3. BREAK ALL SHARP CORNERS .003/.005 R

Figure 2. Mechanical drawing of thermistor probe adapter. All dimensions are in inches.

See Figure 1 also.

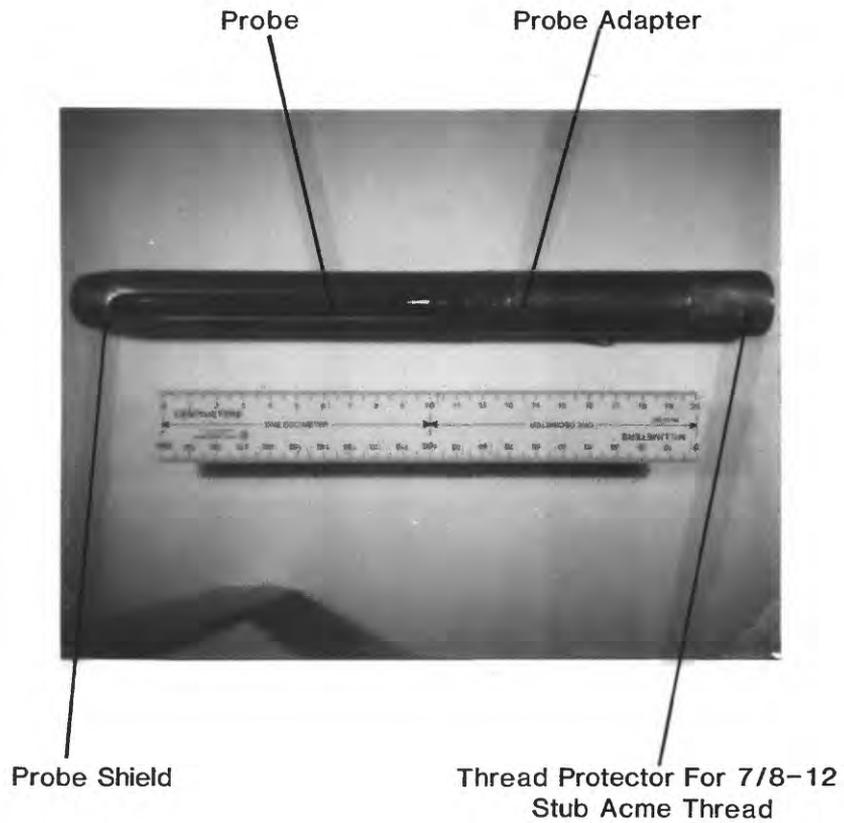


Figure 3. Photograph of assembled thermistor probe and adapter with addition of shield to protect probe from damage and thread protector on adapter. The shield is one of several designs that have been used.

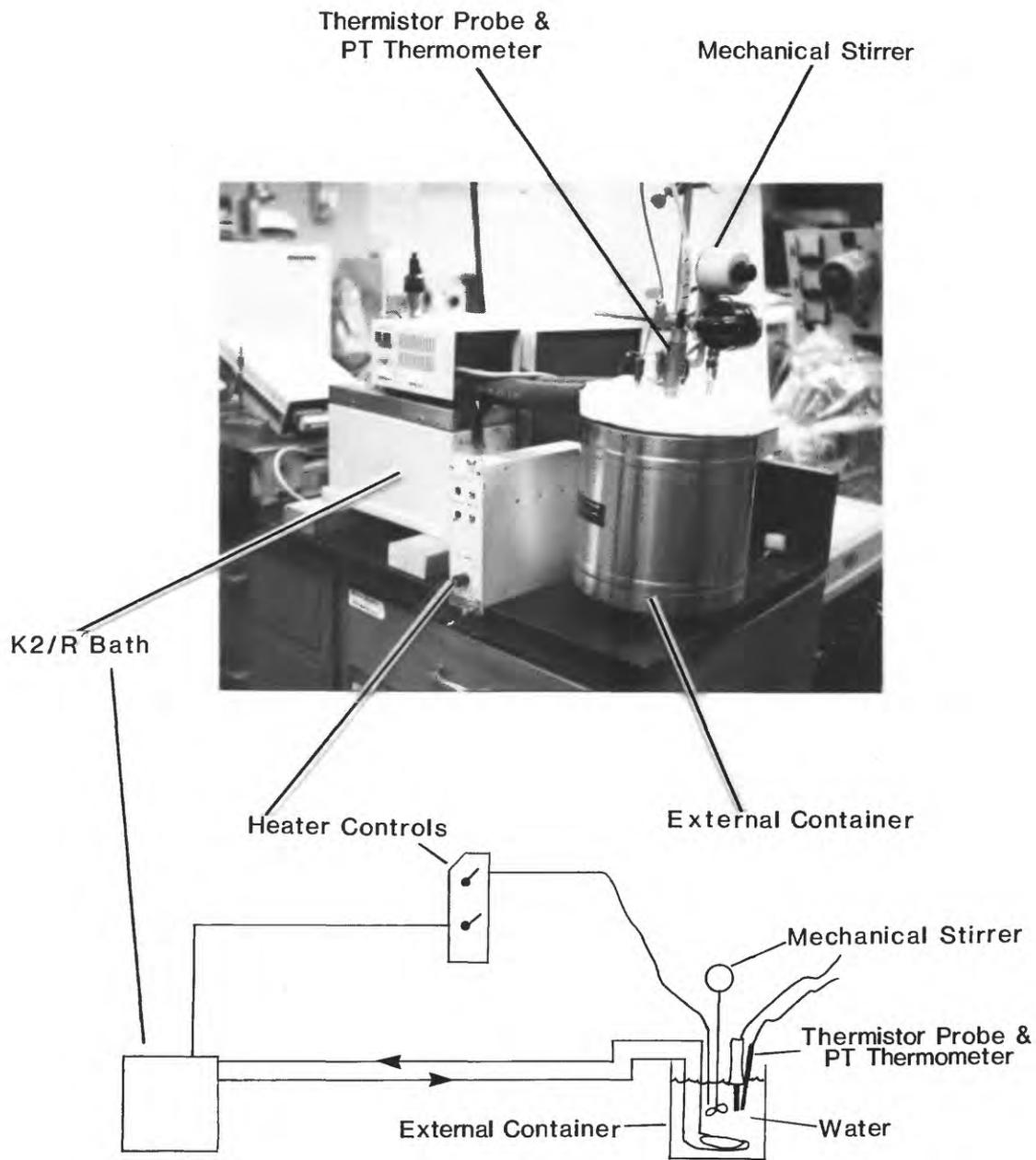


Figure 4. Photograph and schematic of the temperature calibration water bath. Water is circulated between the K2/R bath and the external container.

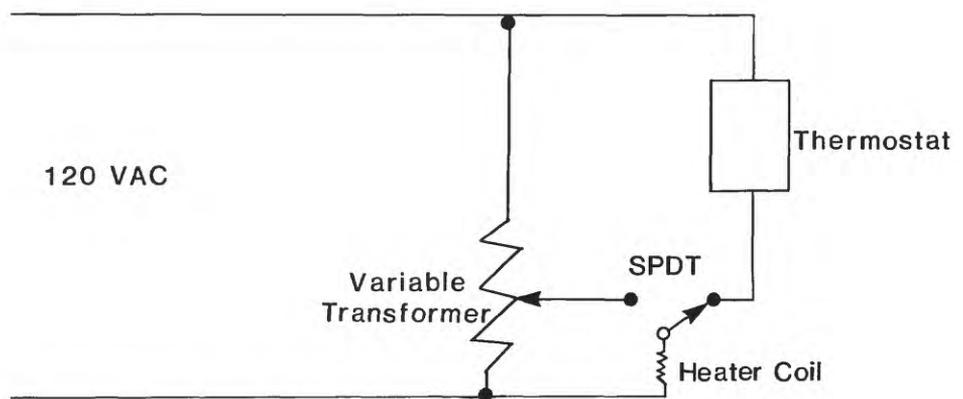
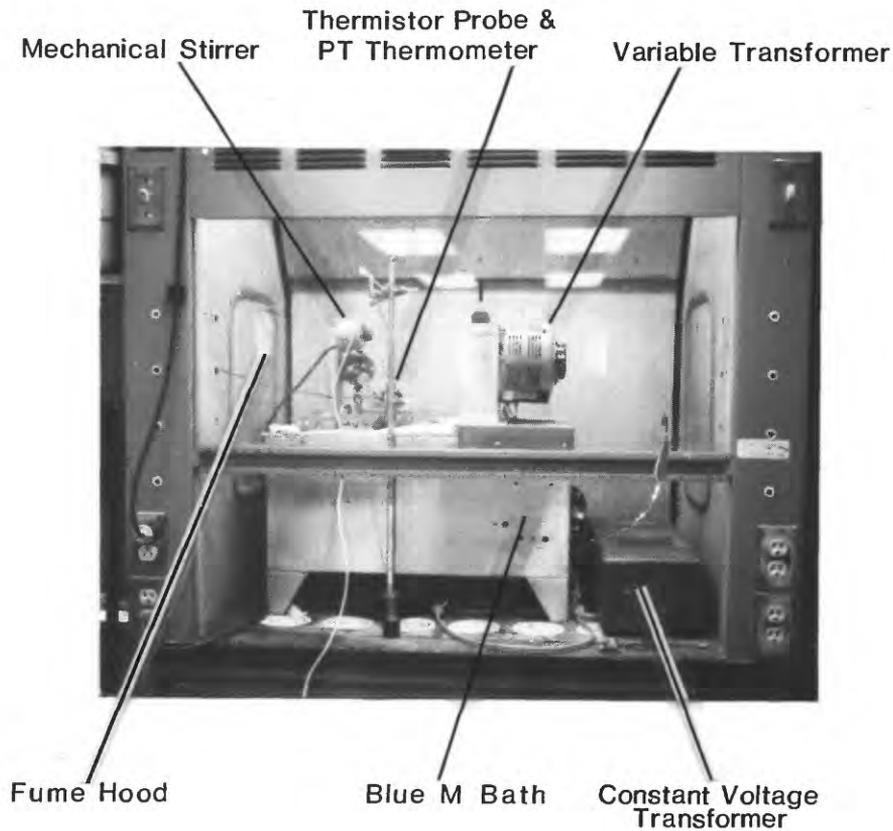


Figure 5. Photograph of the high temperature oil bath. Bath is located in fume hood in order to exhaust fumes at high temperatures. Schematic of power circuit showing relationship of thermostat and variable transformer to the heater coil.

REFERENCES

- Diment, W.H. and Robertson, E.C., 1963, Temperature, thermal conductivity, and heat flow in a drilled hole near Oak Ridge, Tennessee: *Journal of Geophysical Research*, v. 68, p. 5035-5047.
- Emerson and Cuming, 1974, Preliminary technical bulletin 7-2-12A, Stycast 2762FT: Emerson and Cuming, Inc., Canton, MA, 1 p.
- Hald, A. , 1952 , *Statistical theory with engineering applications* : John Wiley & Sons, New York , 783 p.
- Nielsen, K.L. , 1964 , *Methods in numerical analysis*, 2nd ed. : The MacMillan Company, New York , 408 p.
- Raspet, R., Swartz, J.H., Lillard, M.E., and Robertson, E.C., 1966, Preparation of thermistor cables used in geothermal investigations: *U.S. Geological Survey Bulletin* 1203-C, 11 p.
- Robertson, E.C., Raspet, R., Swartz, J.H., and Lillard, M.E., 1966, Properties of thermistors used in geothermal investigations: *U.S. Geological Survey Bulletin* 1203-B, 34 p.
- Sillars, R.W., 1942, Materials and devices of falling resistance-temperature characteristic: *Journal of Scientific Instruments*, p. 81-84
- Tektronix, 1976a, 4051R01 Matrix Functions Instruction Manual: Tektronix, Inc., Beaverton, Oregon, 64 p.
- Tektronix, 1976b, 4051 Graphic System Reference Manual: Tektronix, Inc., Beaverton, Oregon, 632p.
- Wolberg, J.R., 1967, *Prediction analysis*: D. Van Nostrand Company, Princeton, 291 p.

APPENDIX I

4051 BASIC Program THERMISTOR

Program THERMISTOR was originally written in 1979 in order to eliminate manual data input and recording of the thermistor calibration data. The program is written in Tektronix 4051 BASIC. Since the 4051 is a graphics computer/controller, many enhancements to BASIC are included for high-resolution graphics, equivalent to a Tektronix 4012 terminal, and for controlling the GPIB (IEEE 488) bus. In order to use this program on another computer, the "CALL", "WBYTE", disk and I/O (input/output) statements would need to be modified.

In the 4051, "CALL" is used to access various ROM (Read-Only-Memory) routines in plugin packs. "WBYTE" is a primitive output routine that writes a byte or series of bytes directly on the GPIB bus. WBYTE is generally used to control an external device, such as forcing a multimeter into LOCAL mode. It can also be used to insure that all the external devices are unaddressed (WBYTE @63,95:), that is that all the devices are in a quiescent state and not listening (63 \equiv unlisten) or talking (95 \equiv untalk). Various other values are device dependent or IEEE 488 bus messages. Device addresses are determined by adding 32 to the device number for a listen address or 64 for a talk address. Hence, the multimeter, device 12, is sent to local mode (IEEE message GTL) by WBYTE@44,1: (e.g., line 240 of program THERMISTOR). The unlisten command may also be included (i.e., WBYTE@63,44,1:).

In many implementations of BASIC, the I/O addressing symbol "@" is replaced by "#" and the ":" by ",". In the "format statements", such groups as "4D.3D" in line 1180 mean 4 digits are allowed to the left of a decimal point, followed by a decimal point and 3 digits to the right of the decimal point. Format statements are used following a PRINT USING and can be enclosed in quotation marks (line 2330), stored in a string such as K\$ (line 3090) or referenced by line number as in line 3200 to an IMAGE (format) statement (line 3210). "F" in a format statement allows for the full length of the string (FA) or digits (FD.FD). For specific information, consult the Tektronix 4051 Reference Manual (Tektronix, 1976b).

The 4051 has 10 user definable keys (UDK) which may be used directly or with the shift key to effectively enable 20 UDKs. Each "key", therefore, has 2 unique addresses which correspond to line numbers in a BASIC program. Thus UDK number 1 is associated with the line numbered 4, UDK 2 with the line numbered 8, UDK 20 (shifted UDK 10) with the line numbered 80. Any comment (REMARK or REM) or executable statement can be associated with each key and subsequent numbers. Because 4051 BASIC is an interpretative type of BASIC, the program can be interrupted and restarted at virtually any line number using the UDKs or "RUN linenumber". Most of the first 80 line numbers in THERMISTOR execute, via "GOTO" or "GOSUB", some subroutine or routine in the program, although a few statements define variables before branching. The remainder of the discussion covers some aspects in general of the program, although specific lines are not examined in detail.

Lines 100–350 — data input of thermistor resistance (R) and temperature (T). The numbers "@20" and "@12" are the primary GPIB bus addresses of the Guildline 9540 Thermometer and the Fluke 8502A Digital Multimeter, respectively. The routine reads R, T and the time and prints the numbers. R and T are also stored in vectors R9 and T9. The program then waits 30 s to take the next reading (actually 28 s since the I/O routines consume about 2 s). The loop is repeated for N readings (usually N=10) and then all of the instruments are returned to "local" mode.

Lines 360–540 — initialization of variables.

Lines 550–640 — reads the DC voltage across the thermistor with no applied current. A delay loop, lines 580–600 is included to allow for the charge due to cable capacitance to decay.

Lines 650–980 — sets up disk directory and filenames. The routine also creates the file if necessary.

Lines 990–1190 — changes Fluke 8502A multimeter ranges and sets print format.

Lines 1200–2390 — plots data either as points and line or line only. Data is automatically scaled to fit the screen of the 4051.

Lines 2400–2480 — disk file is opened for update ("U") and the data is written to the disk.

Lines 2490–2690 — deletes data points until no number is entered. Data point 1 through 10 is selected to be deleted, based on the screen plot, starting with the highest number first. The vector values are then moved down one. Thus, if data point 7 is to be discarded, then point 8 is written into location 7, point 9 into 8 and 10 into 9. The number of retained (valid) data points is adjusted as is the next point to be entered. New data points are then re-entered (UDK 7 \equiv line 28).

Lines 2700–2770 — inputs and prints on screen the platinum thermometer temperature (UDK 9).

Lines 2780–2830 — inputs and prints on screen the thermistor resistance (UDK 19 \equiv [shift] UDK 9).

Lines 2840–2970 — asks for a starting temperature, reads the platinum thermometer and inputs R and T (lines 100–350) when the specified temperature is reached.

Lines 2980–3130 — inputs data from the disk file and prints it on the screen or line printer.

Lines 3140–3220 — asks for a title for printer listing and prints it or prints the previously entered title. The printer is “paged” (“P”) before the title is printed.

Lines 3230–3320 — all of the data from the disk file is read into memory. Control is transferred to the plotting routine at line 1260.

Lines 3330–3380 — eliminates spaces in the Guildline 9540 thermometer input string so that it may be converted into a real number.

Lines 3390–3490 — mounts or dismounts a floppy disk.

Lines 3500–3550 — reads file header and determines the number of data points which actually have been stored in the file, excluding “N”, the number at the beginning of the file which is the final number of points in the file. This is used to determine if the data has been saved.

Lines 3560–3620 — lists the data that is in memory.

1 REM PGM: THERMISTOR (CALIBRATION) - DATA COLLECTION - 6/29/79
2 REM FLUKE 8502A & GUILDLINE 9540 - LEVEL 11 - 7/12/89
3 GO TO 370
4 REMARK FLUKE RANGE 100K
5 GOSUB 1060
6 RETURN
8 REM FLUKE RANGE 1000K
9 GOSUB 1130
10 RETURN
12 REM DC VOLTS - 8502A
13 GO TO 560
14 PRINT A\$
15 RETURN
16 REMARK FILE
17 GO TO 650
20 REM MOUNT/DISMOUNT DISK
21 GO TO 3390
24 REMARK DATA INPUT
25 GO TO 100
28 REM RE-INPUT DATA
29 GO TO 130
32 REMARK PLOT
33 Q=0
34 GO TO 1230
36 REMARK INPUT AND PRINT RTD THERMOMETER TEMPERATURE
37 GO TO 2700
40 REMARK STORE DATA
41 GO TO 2400
44 REMARK START_UP
45 GO TO 370
48 REM FLUKE RANGE 10K
49 GOSUB 990
50 RETURN
52 GO TO 3500
56 GO TO 3560
60 REM TEMPERATURE ALARM
61 GO TO 2840
64 REM TITLE
65 GO TO 3140
68 REMARK DELETE DATA POINT
69 GO TO 2490
72 Q=1
73 GO TO 1230
76 REMARK 8502 RESISTANCE
77 GO TO 2780
80 GO TO 2980
100 REMARK DATA INPUT ROUTINE

```

110 A9=1
120 I=1
130 PRINT @20:"R"
140 N1=N
150 INPUT @20:T$
160 GOSUB 3330
170 T=VAL(T$)
180 PRINT @12:" ";
190 INPUT @12:R
200 CALL "TIME",S$
210 S$=SEG(S$,11,8)
220 PRINT USING G$:I,R,T,S$
230 PRINT @U9: USING G$:I,R,T,S$
240 WBYTE @44,1:
250 R9(I)=R
260 T9(I)=T
270 I=I+1
280 IF I > N THEN 310
290 CALL "WAIT",28
300 GO TO 150
310 WBYTE @44,4:
320 WBYTE @63,95:
330 CALL "GTL";20
340 PRINT ""
350 END
360 REMARK INITIALIZATION ROUTINE
370 INIT
380 N=10
390 U=33
400 PRINT "PRINTER UNIT: ";
410 INPUT U9
420 SET KEY
430 DIM F$(300)
440 PRINT "CONSTANTS INITIALIZED"
450 PRINT
460 REMARK NUMBER OF READINGS AT A GIVEN TEMPERATURE
470 PRINT "CURRENT NUMBER OF READINGS = ";N;" - CHANGE (Y OR CR): ";
480 INPUT Q$
490 IF Q$="" THEN 520
500 PRINT "NEW NUMBER OF READINGS: ";
510 INPUT N
520 DELETE R9,T9
530 DIM R9(N),T9(N)
540 END

```

```

550 REMARK DC VOLTS
560 PRINT @12:"%,JVR1S7TL ";
570 INPUT @12:V
580 FOR IO=1 TO 25
590 B=1.234567989E+9
600 NEXT IO
610 INPUT @12:V
620 PRINT "V=";V
630 WBYTE @63,44,1:
640 RETURN
650 REMARK FILE INFORMATION
660 PRINT "PROBE LIBRARY (E.G., PROBE10): ";
670 INPUT J$
680 CALL "USERLIB",J$
690 PRINT "PROBE NUMBER (E.G., 711): ";
700 INPUT E$
710 E$="N"&E$
720 CALL "TIME",T$
730 E$=E$&"/"
740 DELETE H$
750 DIM H$(3)
760 H$=SEG(T$,4,3)
770 E$=E$&H$
780 H$=SEG(T$,1,2)
790 E$=E$&H$
800 H$=SEG(T$,8,2)
810 E$=E$&H$
820 PRINT "FILE EXTENSION (E.G., A, B, C...): ";
830 INPUT T$
840 T$="."&T$
850 E$=E$&T$
860 PRINT J$;"/";E$;" - OK Y/N: ";
870 INPUT Q$
880 IF Q$<>"Y" THEN 650
890 PRINT "CREATE DISK FILE - Y OR CR: ";
900 INPUT Q$
910 IF Q$=" " THEN 980
920 PRINT "NUMBER OF TEMPERATURE STEPS: ";
930 INPUT NO
940 CREATE E$;20 * NO * N + 100,0
950 OPEN E$;5,"F",F$
960 WRITE #5:2 * N * NO
970 CLOSE 5

```

```

980 RETURN
990 REM FLUKE RANGE 10K
1000 PRINT @12:"%,JZR3S7TL ";
1010 INPUT @12:A
1020 WBYTE @63,44,1:
1030 G$="4D,2X,7D.2D,2X,4D.3D,4X,8A"
1040 K$="4D,2X,7D.2D,2X,4D.3D"
1050 RETURN
1060 REMARK FLUKE RANGE 100K
1070 PRINT @12:"%,JZR4S7TL ";
1080 INPUT @12:A
1090 WBYTE @63,44,1:
1100 G$="4D,2X,7D.D,2X,4D.3D,4X,8A"
1110 K$="4D,2X,7D.D,2X,4D.3D"
1120 RETURN
1130 REMARK FLUKE RANGE 1000K
1140 PRINT @12:"%,JZR5S7TL ";
1150 INPUT @12:A
1160 WBYTE @63,44,1:
1170 G$="4D,2X,7D,2X,4D.3D,4X,8A"
1180 K$="4D,2X,7D,2X,4D.3D"
1190 RETURN
1200 REMARK PLOT DATA
1210 PRINT "POINTS & LINE(0) OR LINE ONLY(1): ";
1220 INPUT Q
1230 DELETE L0,H1
1240 PAGE
1250 DIM L0(2),H1(2)
1260 L0=10000000
1270 H1=-10000000
1280 FOR I=1 TO N
1290 L0(1)=L0(1) MIN R9(I)
1300 L0(2)=L0(2) MIN T9(I)
1310 H1(1)=H1(1) MAX R9(I)
1320 H1(2)=H1(2) MAX T9(I)
1330 NEXT I
1340 W1=L0(1)
1350 W2=H1(1)
1360 W3=L0(2)
1370 W4=H1(2)
1380 RESTORE 1390
1390 DATA 10,120,10,90
1400 READ V1,V2,V3,V4

```

```

1410 VIEWPORT V1 + 6 * 1.8,V2,V3 + 3 * 2.8,V4
1420 DELETE P
1430 DIM P(8)
1440 P(1)=W1
1450 P(2)=W2
1460 P(3)=INT((V2 - (V1 + 6 * 1.8))/(8 * 1.8)) MAX 1
1470 P(5)=W3
1480 P(6)=W4
1490 P(7)=INT((V4 - (V3 + 3 * 2.8))/(3 * 2.8)) MAX 1
1500 REMARK CALCULATE NEW LIMITS AND INTERVALS
1510 P5=3
1520 GOSUB 1920
1530 P5=7
1540 GOSUB 1920
1550 WINDOW P(1),P(2),P(5),P(6)
1560 V5=(P(6) - P(5))/(V4 - (V3 + 3 * 2.8))
1570 H5=(P(2) - P(1))/(V2 - (V1 + 6 * 1.8))
1580 MOVE (P(1)+P(2))/2,P(6)
1590 RMOVE -(LEN(J$) + LEN(E$) + 1) * H5 * 1.792/2,V5 * 2.816
1600 PRINT J$;"/";E$;
1610 S1=(P(2) - P(1))/(V2 - (V1 + 6 * 1.8))
1620 S2=(P(6) - P(5))/(V4 - (V3 + 3 * 2.8))
1630 REM AXES
1640 AXIS P(3),P(7),P(1)+P(3),P(5)+P(7)
1650 REMARK LABEL AXES
1660 P5=4
1670 A$=""
1680 GOSUB 2210
1690 P5=8
1700 A$=""
1710 GOSUB 2210
1720 FOR I=1 TO N
1730 IF I>1 AND Q=1 THEN 1830
1740 MOVE R9(I),T9(I)
1750 IF Q=1 THEN 1840
1760 RMOVE -S1 * 1.79/3,-S2 * 2.82/3
1770 I1=I
1780 IF I1 <= 9 THEN 1810
1790 I1=I1 - 10
1800 GO TO 1780
1810 PRINT I1;
1820 GO TO 1840
1830 DRAW R9(I),T9(I)

```

```

1840 NEXT I
1850 IF Q=1 THEN 1880
1860 Q=1
1870 GO TO 1720
1880 HOME
1890 PRINT A9
1900 A9=A9+1
1910 END
1920 REMARK P(P5) = MINIMUM NO. TICS
1930 P1=(P(P5-1)-P(P5-2))/P(P5)
1940 P2=10^INT(LGT(P1))
1950 P1=P1/P2
1960 IF P1 > 2 THEN 2000
1970 IF P1=1 THEN 2040
1980 P2=2 * P2
1990 GO TO 2040
2000 IF P1 > 5 THEN 2030
2010 P2=5 * P2
2020 GO TO 2040
2030 P2=10 * P2
2040 REMARK ADJUST DATA MIN
2050 P1=INT(P(P5-2)/P2)
2060 P3=P2 * (P1 + 2)
2070 IF P3 < P(P5 - 2) THEN 2100
2080 P3=P3-P2
2090 GO TO 2070
2100 P(P5-2)=P3
2110 REMARK ADJUST DATA MAX
2120 P1=INT(P(P5-1)/P2)
2130 P3=P2 * (P1 - 2)
2140 IF P(P5 - 1) < P3 THEN 2170
2150 P3=P3+P2
2160 GO TO 2140
2170 P(P5-1)=P3
2180 REMARK P(P5)=ADJUSTED TIC INTERVAL
2190 P(P5)=P2
2200 RETURN
2210 REMARK LABEL AXIS
2220 P4=P(P5-1)
2230 P(4)=P(1)
2240 P(8)=P(5)
2250 P3=ABS(P(P5-3)+P4) MAX ABS(P(P5-2)-P4)
2260 P3=INT(LGT(P3)+1.0E-8)

```

```

2270 P2=10^-P3
2280 P1=P(P5-2)-P4/2
2290 P(P5)=P(P5)+P4
2300 IF P(P5) > P1 THEN 2350
2310 MOVE P(4),P(8)
2320 PRINT A$;
2330 PRINT USING "-D.5D,S":P(P5) * P2
2340 GO TO 2290
2350 IF P3=0 THEN 2390
2360 P(P5)=P1
2370 MOVE P(4),P(8)
2380 PRINT USING "2A,+FD,S":" E";P3
2390 RETURN
2400 REMARK STORE DATA
2410 OPEN E$;5,"U",F$
2420 FOR I=1 TO N
2430 WRITE #5:R9(I),T9(I)
2440 NEXT I
2450 PRINT "DATA STORED IN DISK FILE: ";J$;"/";E$
2460 PRINT
2470 CLOSE 5
2480 END
2490 REMARK DELETE DATA POINT
2500 PRINT "INDEX OF DATA ITEM TO BE DELETED: ";
2510 INPUT I$
2520 IF I$="" THEN 2650
2530 I0=VAL(I$)
2540 IF I0 => 1 AND I0<N1 THEN 2580
2550 IF I0=N1 THEN 2620
2560 PRINT "INDEX OUT OF RANGE"
2570 GO TO 2500
2580 FOR J=I0+1 TO N1
2590 R9(J-1)=R9(J)
2600 T9(J-1)=T9(J)
2610 NEXT J
2620 N1=N1-1
2630 PRINT @U9:"DATA ITEM NUMBER: ";I0;" DELETED"
2640 GO TO 2500
2650 PRINT "TOTAL NUMBER OF VALID POINTS = ";N1
2660 I=N1+1
2670 IF I <= N THEN 2690
2680 PRINT "I > DIMENSION OF ARRAYS"
2690 END
2700 REMARK INPUT AND PRINT PLATINUM THERMOMETER TEMPEATURE

```

```

2710 PRINT @20:"R"
2720 INPUT @20:T$
2730 GOSUB 3330
2740 T6=VAL(T$)
2750 PRINT "T= ";T6
2760 CALL "GTL";20
2770 RETURN
2780 REMARK INPUT AND PRINT FLUKE 8502 RESISTANCE
2790 PRINT @12:" ";
2800 INPUT @12:R6
2810 PRINT "R= ";R6
2820 WBYTE @63,44,1:
2830 RETURN
2840 PRINT " TEMPERATURE ALARM"
2850 PRINT "SIGNAL AT: ";
2860 INPUT O5
2870 PRINT @20:"R"
2880 INPUT @20:T$
2890 GOSUB 3330
2900 T6=VAL(T$)
2910 IF T6<O5 THEN 2880
2920 CALL "TIME",S$
2930 S$=SEG(S$,11,8)
2940 PRINT "TEMPERATURE IS: ";T6;" - TIME IS ";S$
2950 CALL "GTL";20
2960 CALL "WAIT",2
2970 GO TO 100
2980 OPEN E$;5,"R",F$
2990 ON EOF (5) THEN 3120
3000 PRINT "UNIT - CRT(32) OR PRINTER(";U9;"): ";
3010 INPUT U0
3020 IF U0<>32 THEN 3040
3030 PAGE
3040 U1=12
3050 PRINT @U0,U1:"FILE: ";J$;"/";E$;"
3060 READ #5:N9
3070 I=1
3080 READ #5:F5,F6
3090 PRINT @U0,U1: USING K$:I,F5,F6
3100 I=I+1
3110 GO TO 3080
3120 CLOSE 5
3130 END
3140 REM TITLE - 8/7/80

```

```

3150 PRINT "TITLE OR CR: ";
3160 INPUT Q$
3170 IF Q$="" THEN 3190
3180 W$=Q$
3190 PRINT @U9: USING "P":
3200 PRINT @U9: USING 3210:W$
3210 IMAGE FA,L/
3220 RETURN
3230 DELETE R8,T8
3240 OPEN E$;6,"R",F$
3250 READ #6:N
3260 N=N/2
3270 DIM R8(N),T8(N)
3280 FOR I=1 TO N
3290 READ #6:R8(I),T8(I)
3300 NEXT I
3310 Q=1
3320 GO TO 1260
3330 REM ELIMINATE SPACES IN GUILDLINE 9540 INPUT STRING
3340 P0=POS(T$," ",1)
3350 IF P0=0 THEN 3380
3360 T$=REP(" ",P0,1)
3370 GO TO 3340
3380 RETURN
3390 REM MOUNT/DISMOUNT DISK
3400 PRINT "MOUNT DISK ON DRIVE 0 - Y/CR: ";
3410 INPUT Q$
3420 IF Q$="" THEN 3450
3430 CALL "MOUNT",0,F$
3440 RETURN
3450 PRINT "DISMOUNT DISK ON DRIVE 0 - Y/CR: ";
3460 INPUT Q$
3470 IF Q$="" THEN 3490
3480 CALL "DISMOUNT",0
3490 RETURN
3500 CALL "FILE",0,E$,F$
3510 F$=SEG(F$,88,4)
3520 P9=VAL(F$)
3530 P9=(P9-9)/18
3540 PRINT "NUMBER OF DATA POINTS STORED = ";P9
3550 END
3560 REM LIST DATA

```

```
3570 PRINT "LIST DATA ON CRT(32) OR PRINTER(";U9;"): ";
3580 INPUT U8
3590 FOR I=1 TO 10
3600 PRINT @U8: USING K$:I,R9(I),T9(I)
3610 NEXT I
3620 END
```

APPENDIX II

PL/1 Program THMCAL

Program THMCAL does a nonlinear, least-squares fit between thermistor resistances and temperatures as measured by a secondary temperature standard such as a platinum or quartz thermometer. This program should work with little or no modifications on any computer system with a PL/1 compiler. Depending upon the implementation, however, variables that are defined as "float decimal" may be more efficiently designated as "float binary". "SYSIN" and "SYSPRINT" are the standard input and output devices. The reference is probably to a keyboard and screen of a terminal, although in a batch process it could refer to a cardreader and line printer. The procedure for redefining sysin and sysprint as disk files varies with the operating system of the computer but is usually fairly trivial.

Input and output (I/O) statements are "get" and "put" with "edit" the formatted versions. Variables and files are declared with "dcl". Files are opened with "open" and closed with "close". If the input file is truncated at any point, then the program terminates ("on endfile ..."). After the data is read, initial values for C1, C2, and C3 are set along with "epsil", the minimum value for the A_k 's (eq. 11). The estimated temperatures (t_c) are calculated. A call to "zero" zeros the array s and its inverse e . The elements for the matrix in equation 11 are determined (s). A call to "crmes" solves the set of simultaneous equations by Crouts method (Nielsen, 1964, p. 185-188). C1, C2, and C3 are corrected and a new estimated temperature (t_c) calculated. Differences between the measured and calculated temperatures are also determined. The iteration counter is incremented and the corrections are each checked against epsil. If any of the corrections are NOT less than epsil, then the procedure is repeated up to 50 times. Once convergence is obtained, when all of the corrections are less than epsil, the 3×3 inverse matrix, c , is calculated, as is "sn", the weighted sum of the squares of the temperature differences. From c and sn, the estimated uncertainties (or standard deviations) of C1, C2, and C3 ($sc1$, $sc2$, and $sc3$) are determined (eq. 16).

Procedures "tl" and "hd" put the title and heading on each page. The calibration data, thermistor equation, constants and standard deviations of the constants are printed. At this point, the program jumps to label "next" to get another set of calibration data. If none exists, the program branches to "srt" and finishes. Otherwise another calibration is determined with the new data.

```

thmcal: proc;

dcl title char (60);
dcl (r(100),t(100),tc(100),d(100),s(3,4),e(3,4),c1,c2,c3,
      ca,b,a,zl,sigx,sigy,det,c(3,3),sn,sc1,sc2,sc3,epsil)
      float decimal (16);

dcl (n,i,k,ict)fixed binary;
dcl (sqrt,log,abs) builtin;
dcl (endpage,endfile) condition;
dcl sysin file stream input;
dcl sysprint file stream output;

open file(sysin) input stream;
open file(sysprint) output stream print pagesize(57) linesize(100);

on endfile(sysin) go to srt;

n=100;
next: get file(sysin) edit (title,n)(a(60),f(3));
      get file(sysin) edit ((r(i),t(i) do i=1 to n))(f(8,1),f(7,3));
      get file(sysin) edit (sigx,sigy)(f(4,2),x(1),f(6,3));

c1=3000; c2=-300; c3=5; epsil=0.000000005;

do i=1 to n; tc(i)=(c1 + c2 * log(r(i)))/(c3 + log(r(i))); end;

ict=0;
do k=1 to 50; call zero (s,e);
l10: do i=1 to n; zl=0;
      zl=sigy * sigy + sigx * sigx * (c1 - c2 * c3) * (c1 - c2 * c3)/(((c3 + log(r(i))) ** 4)*r(i)*r(i));
      s(1,1)=s(1,1) + (1/((c3 + log(r(i))) ** 2))/zl;
      s(1,2)=s(1,2) + (log(r(i))/((c3 + log(r(i))) ** 2))/zl;
      s(1,3)=s(1,3) + ((c1 + c2 * log(r(i)))/(c3 + log(r(i))) ** 3)/zl;
      s(1,4)=s(1,4) + ((tc(i) - t(i))/(c3 + log(r(i))))/zl;
      s(2,2)=s(2,2) + (log(r(i)) * log(r(i)))/((c3 + log(r(i))) ** 2)/zl;
      s(2,3)=s(2,3) + (log(r(i)) * (c1 + c2 * log(r(i)))/((c3 + log(r(i))) ** 3))/zl;
      s(2,4)=s(2,4) + ((tc(i) - t(i)) * log(r(i))/(c3 + log(r(i))))/zl;
      s(3,3)=s(3,3) + ((c1 + c2 * log(r(i))) ** 2)/((c3 + log(r(i))) ** 4)/l;
      s(3,4)=s(3,4) + ((t(i) - tc(i)) * (c1 + c2 * log(r(i)))/((c3 + log(r(i))) ** 2))/zl;

end;

s(2,1)=s(1,2); s(1,3)=-s(1,3); s(3,1)=s(1,3); s(2,3)=-s(2,3);
s(3,2)=s(2,3);

call crmes (s,e);

l11: ca=e(3,4); b=e(2,4)-e(2,3)*ca;
      a=e(1,4) - e(1,3) * ca - e(1,2) * b;

```

```

c1=c1-a; c2=c2-b; c3=c3-ca;
do i=1 to n; tc(i)=(c1 + c2 * log(r(i)))/(c3 + log(r(i)));
d(i)=t(i)-tc(i); end;
ict=ict+1;
if abs(a) < epsil then go to l2; go to l4;
l2: if abs(b) < epsil then go to l3; go to l4;
l3: if abs(ca) < epsil then go to l5;
l4: end;
l5: det=s(1,1) * (s(2,2) * s(3,3) - s(3,2) * s(2,3)) - s(1,2) * (s(2,1) * s(3,3) -
s(3,1) * s(2,3)) + s(1,3) * (s(2,1) * s(3,2) - s(3,1) * s(2,2));
c(1,1)=(s(2,2) * s(3,3) - s(2,3) * s(3,2))/det;
c(2,2)=(s(1,1) * s(3,3) - s(1,3) * s(3,1))/det;
c(3,3)=(s(1,1) * s(2,2) - s(2,1) * s(1,2))/det;
c(1,2)=(-s(2,1) * s(3,3) + s(2,3) * s(3,1))/det;
c(1,3)=(s(2,1) * s(3,2) - s(2,2) * s(3,1))/det;
c(2,1)=(-s(1,2) * s(3,3) + s(1,3) * s(3,2))/det;
c(2,3)=(-s(3,2) * s(1,1) + s(1,2) * s(3,1))/det;
c(3,1)=(s(1,2) * s(2,3) - s(1,3) * s(2,2))/det;
c(3,2)=(-s(2,3) * s(1,1) + s(2,1) * s(1,3))/det;
sn=0; zl=0;
do i=1 to n;
zl=sigy * sigy + sigx * sigx * ((c1 - c2 * c3) **2)/(((c3 + log(r(i))) **4) * (r(i) **2));
sn=sn + d(i) * d(i)/zl; end;

sc1=sqrt(sn * c(1,1)/(n - 3));
sc2=sqrt(sn * c(2,2)/(n - 3));
sc3=sqrt(sn * c(3,3)/(n - 3));

tl: proc;
put file(sysprint) page line(3) edit (title)(a(60));
put file(sysprint) skip(2); end;

hd: proc;
put file(sysprint) edit (" R", " T", "TC", " D")(4 (x(9),a(2)));
put file(sysprint) skip(2); end;

on endpage (sysprint) begin; call tl; call hd; end;

call tl; call hd;

do i=1 to n;
put file(sysprint) edit (r(i),t(i),tc(i),d(i))(skip,x(5),f(8,1),3 (x(4),f(7,3)));
end;

call tl;
put file(sysprint) edit ("Thermistor Equation: ",

```

```

        "T = (C1 + C2 * LN(R))/(C3 - LN(R))", "C1 = ",c1,"S. Dev. C1 = ",
        sc1,"C2 = ",c2,"S. Dev. C2 = ",sc2,"C3 = ",c3,
        "S. Dev. C3 = ",sc3)(x(6),a(21),a(26),skip(2),
        3 (skip,x(4),a(5),f(13,8),x(4),a(13),f(13,8)));
put file(sysprint) edit ("Number of Iterations: ",ict)
        (skip(3),x(6),a(22),f(2));
put file(sysprint) edit ("Experimental Error in R: ",sigx,
        "Experimental Error in T: ",sigy)(skip(3),x(6),a(26),
        f(4,1),skip,x(6),a(26),f(5,3));
go to next;
zero: proc (s,e);
        dcl (s(3,4),e(3,4))float decimal (16);
        dcl (ii,jj)fixed binary;
        do ii=1 to 3; do jj=1 to 4;
                s(ii,jj)=0; e(ii,jj)=0; end; end;
        return; end;
crmes: proc (s,e);
        dcl (s(3,4),e(3,4))float decimal (16);
        e(1,1)=s(1,1); e(2,1)=s(2,1); e(3,1)=s(3,1);
        e(1,2)=s(1,2)/s(1,1);
        e(1,3)=s(1,3)/s(1,1);
        e(1,4)=s(1,4)/s(1,1);
        e(2,2)=s(2,2) - e(1,2) * e(2,1);
        e(3,2)=s(3,2) - e(1,2) * e(3,1);
        e(2,3)=(s(2,3) - e(1,3) * e(2,1))/e(2,2);
        e(2,4)=(s(2,4) - e(1,4) * e(2,1))/e(2,2);
        e(3,3)=s(3,3) - e(2,3) * e(3,2) - e(1,3) * e(3,1);
        e(3,4)=(s(3,4) - e(2,4) * e(3,2) - e(1,4) * e(3,1))/e(3,3);
return; end;
srt: close file(sysin), file(sysprint);
end thmcal;

```

APPENDIX III

4051 BASIC Program THMCAL

In 1981 we rewrote PL/1 program THMCAL in BASIC to run on the Tektronix 4051. We added several additions to accommodate the file structure and the ROM (Read-Only-Memory) packs available in the 4051. The User-Definable-Keys (UDK) are 1 (initial startup and run calibration), 6 (subsequent calibrations of the same thermistor) and 10 (reprint calibration). As we described in Appendix I, the line number in the program is four times the UDK number.

Lines 100–280 — set up variables.

Lines 290–340 — select disk and printer.

Lines 350–510 — define parameters, titles, and format range for printing of resistances. "USERLIB" is a routine that selects a subdirectory path. Line 490 is the entry point for subsequent calibrations (entered by UDK 6).

Lines 520–540 — clears screen and prints titles on screen.

Lines 550–560 — opens disk file for input.

Lines 570–600 — subroutine that gets and prints time.

Lines 580–770 — read in calibration data (resistance and temperature). E\$ is used by the time routine in line 590. Data file is sequential. A check is made to determine if the correct number of points has been read. If an end-of-file (EOF) is reached before all the required number of points have been read, then control is passed to line 750 by the "ON" condition in line 610.

Lines 780–800 — constants are read. Pointer for READ is set to line 790. Initial values for C1, C2, and C3 are read as well as the value to check for convergence, E0, and the errors in resistance (S0) and temperature (S1).

Lines 810–1070 — array values calculated. Based on the initial values, the calculated temperatures (T0) are obtained. The iteration counter (I0) is initialized (line 850). The maximum number of iterations allowed is set by the FOR loop at 20 (line 860). Arrays S and E are zeroed (lines 870 and 880). Values for the S array are calculated (lines 890–1070). This sets up equation 11.

Lines 1080–1110 — inverse of the 3×4 array S is obtained using ROM routine INV (Tektronix, 1976a). The square part of the matrix, E, starting at the upper left-hand corner (1,1) is the inverse of the square part (3×3) of matrix S. The remaining column in E represents the solution of the set of simultaneous equations (lines 1090–1110). Actually INV is more general than the above indicates. The arrays can be 3×5 , 4×5 , 4×9 , etc. and each column beyond the square part is a solution to a set of linear equations (Tektronix, 1976a).

Lines 1120–1140 — new estimates of C1, C2, and C3 are obtained.

Lines 1150–1190 — new calculated temperatures and differences are calculated.

Line 1200 — iteration counter is incremented.

Lines 1210–1260 — values of corrections are checked against E0. If any of the corrections are greater than E0, then the next iteration is calculated. The iteration number is printed (line 1240). If all of the corrections are less than E0, then the program execution continues at line 1260 where the last iteration number is listed on the screen.

Lines 1270–1380 — standard deviations of the constants are calculated (eq. 16). The constants are rounded to H (i.e., 4 decimal) places.

Lines 1390–1400 — ending time is obtained and printed.

Lines 1410–1540 — nominal temperature range is determined from data, For example, $60\text{--}90^\circ\text{C}$. This could be obtained by additional input parameters in the data file.

Lines 1550–1560 — title and headings printed on printer.

Lines 15570–1870 — data and constants printed on printer. Specific print formats are selected depending upon significant figures and decimal places for resistance (lines 1580–1650). Only 50 points are allowed per page (lines 1670–1690). Thermistor constants are listed on a new page.

Lines 1880–2020 — title and heading subroutines with corresponding format statements. Appendix I explains some of the format types. For a detailed explanation, see Tektronix (1976b).

```

1 REM PROGRAM: THMCAL - 12/27/89
2 REM MODIFIED FROM PL/1 ON 10/6/81
3 END
4 PAGE
5 GO TO 100
24 PAGE
25 B=0
26 GO TO 490
40 B=1
41 GO TO 1550
100 REM DATA INPUT
110 INIT
120 DIM A$(7),B$(11),D$(7),V$(10),U$(11),G$(10),H$(9),K$(10),L$(6)
130 DIM M$(3),N$(10),O$(21),P$(34),S$(1),F$(300)
140 A$="NOMINAL"
150 B$="CALIBRATION"
160 D$="RANGE: "
170 V$="RESISTANCE"
180 U$="TEMPERTURE"
190 G$="THERMISTOR"
200 H$="OBSERVED"
210 K$="CALCULATED"
220 L$="(OHMS)"
230 M$="(C)"
240 N$="DIFFERENCE"
250 O$="THERMISTOR EQUATION: "
260 P$="T = [C1 + C2 × ln(R)]/[C3 + ln(R)]"
270 Q$="EXPERIMENTAL ERROR IN R: "
280 R$="EXPERIMENTAL ERROR IN T: "
290 PRINT "MOUNT DISK ON UNIT: ";
300 INPUT U0
310 CALL "MOUNT",U0,F$
320 CALL "UNIT",U0
330 PRINT "PRINTER UNIT NO. (E.G., 6, 51, ETC.): ";
340 INPUT U
350 PRINT "PRINT NUMBER OF ITERATIONS AND EXPERIMENTAL ERRORS -
Y/N: ";
360 INPUT S$
370 PRINT "PROBE NUMBER AND DATE (E.G., 10-711 — 19 JUNE 1986): "
380 INPUT W$
390 PRINT "MAX T ON 1000K RANGE (E.G., 10): ";
400 INPUT R8
410 R8=R8+9
420 PRINT "MIN T ON 10K RANGE (E.G., 160): ";

```

```

430 INPUT R9
440 W$="THERMISTOR CALIBRATION: PROBE "&W$
450 DIM S(3,4),E(3,4)
460 PRINT "DISK LIBRARY (E.G., PROBE10/N711): ";
470 INPUT J$
480 CALL "USERLIB",J$
490 DIRECTORY
500 PRINT "DISK FILE (E.G., JUN1986.A): ";
510 INPUT I$
520 PAGE
530 PRINT USING "25T,FA/": "THERMISTOR CALIBRATION"
540 PRINT USING "25T,FA,A,FA,/L": J$,"/",I$
550 OPEN I$;1,"R",F$
560 GO TO 610
570 CALL "TIME",T$
580 T$=SEG(T$,11,8)
590 PRINT E$;"TIME = ";T$
600 RETURN
610 ON EOF (1) THEN 750
620 E$="STARTING "
630 GOSUB 570
640 READ #1:N
650 N=N/2
660 DELETE R,T,T0,D
670 DIM R(N),T(N),T0(N),D(N)
680 N0=0
690 FOR I=1 TO N
700 READ #1:R(I),T(I)
710 N0=N0+1
720 NEXT I
730 CLOSE 1
740 IF N0=N THEN 780
750 PRINT "EOF ERROR N<";N0
760 CLOSE 1
770 END
780 RESTORE 790
790 DATA 3000,-300,5,1.0E-6,0.2,0.01,10000
800 READ C1,C2,C3,E0,S0,S1,H
810 FOR I=1 TO N
820 L=LOG(R(I))
830 T0(I) = (C1 + C2 * L)/(C3 + L)
840 NEXT I
850 I0=0
860 FOR K=1 TO 20

```

```

870 S=0
880 E=0
890 FOR I=1 TO N
900 Z=0
910 L=LOG(R(I))
920 Z = S1 * S1 + S0 * S0 * (C1 - C2 * C3) * (C1 - C2 * C3)/((C3 + L) ↑ 4 * R(I) * R(I))
930 S(1,1) = S(1,1) + 1/(C3 + L) ↑ 2/Z
940 S(1,2) = S(1,2) + L/(C3 + L) ↑ 2/Z
950 S(1,3) = S(1,3) + (C1 + C2 * L)/(C3 + L) ↑ 3/Z
960 S(1,4) = S(1,4) + (T0(I) - T(I))/(C3 + L)/Z
970 S(2,2) = S(2,2) + L * L/(C3 + L) ↑ 2/Z
980 S(2,3) = S(2,3) + L * (C1 + C2 * L)/(C3 + L) ↑ 3/Z
990 S(2,4) = S(2,4) + (T0(I) - T(I)) * L/(C3 + L)/Z
1000 S(3,3) = S(3,3) + (C1 + C2 * L) ↑ 2/(C3 + L) ↑ 4/Z
1010 S(3,4) = S(3,4) + (T(I) - T0(I)) * (C1 + C2 * L)/(C3 + L) ↑ 2/Z
1020 NEXT I
1030 S(2,1)=S(1,2)
1040 S(1,3)=-S(1,3)
1050 S(3,1)=S(1,3)
1060 S(2,3)=-S(2,3)
1070 S(3,2)=S(2,3)
1080 E=INV(S)
1090 C0=E(3,4)
1100 B=E(2,4)
1110 A=E(1,4)
1120 C1=C1-A
1130 C2=C2-B
1140 C3=C3-C0
1150 FOR I=1 TO N
1160 L=LOG(R(I))
1170 T0(I) = (C1 + C2 * L)/(C3 + L)
1180 D(I)=T(I)-T0(I)
1190 NEXT I
1200 I0=I0+1
1210 GO TO ABS(A)=>E0 OF 1240
1220 GO TO ABS(B)=>E0 OF 1240
1230 GO TO ABS(C0)<E0 OF 1260
1240 PRINT "ITERATION ";I0;" COMPLETED"
1250 NEXT K
1260 PRINT "LAST ITERATION (";I0;" ) - CALIBRATION COMPLETED"
1270 S5=0

```

```

1280 FOR I=1 TO N
1290 L=LOG(R(I))
1300 Z = S1 * S1 + S0 * S0 * (C1 - C2 * C3) ↑ 2 / ((C3 + L) ↑ 4 * R(I) ↑ 2)
1310 S5 = S5 + D(I) * D(I) / Z
1320 NEXT I
1330 G1 = SQR(S5 * E(1, 1) / (N - 3))
1340 G2 = SQR(S5 * E(2, 2) / (N - 3))
1350 G3 = SQR(S5 * E(3, 3) / (N - 3))
1360 C1 = INT(H * C1 + 0.5) / H
1370 C2 = INT(H * C2 + 0.5) / H
1380 C3 = INT(H * C3 + 0.5) / H
1390 E$="ENDING "
1400 GOSUB 570
1410 REM DETERMINE NOMINAL TEMPERATURE RANGE
1420 L0=N-9
1430 IF INT(T(L0)/10+0.5)=INT(T(N)/10) THEN 1460
1440 L0=L0+1
1450 GO TO 1430
1460 L0=L0+4
1470 A0=INT(R(L0))
1480 T9=INT(T(L0))
1490 L0=5
1500 A1=INT(R(L0))
1510 T8=INT(T(L0))*(INT(T(L0))>0)
1520 T8 = 10 * INT(T8/10)
1530 T9 = 10 * INT(T9/10)
1540 REM *****
1550 GOSUB 1900
1560 GOSUB 1950
1570 FOR I=1 TO N
1580 IF T(I)<R8 THEN 1630
1590 IF T(I)=>R9 THEN 1650
1600 REM PRINT FORMATS
1610 C$="13T,7D.D,26T,3D.3D,39T,3D.3D,53T,2D.3D"
1620 GO TO 1660
1630 C$="13T,7D,26T,3D.3D,39T,3D.3D,53T,2D.3D"
1640 GO TO 1660
1650 C$="13T,7D.2D,26T,3D.3D,39T,3D.3D,53T,2D.3D"
1660 PRINT @U: USING C$:R(I),T(I),T0(I),D(I)
1670 IF I<N AND I<>50 OR I=N THEN 1700
1680 GOSUB 1880
1690 GOSUB 1950
1700 NEXT I

```

```

1710 GOSUB 1880
1720 PRINT @U: USING 1730:O$,P$
1730 IMAGE /,13T,FA,FA,/
1740 PRINT @U: USING 1750:"C1 = ",C1,"S. DEV. C1 = ",G1
1750 IMAGE / ,16X,5A,4D.4D,8X,13A,4D.4D
1760 PRINT @U: USING 1750:"C2 = ",C2,"S. DEV. C3 = ",G2
1770 PRINT @U: USING 1750:"C3 = ",C3,"S. DEV. C3 = ",G3
1780 IF S$="N" THEN 1830
1790 PRINT @U: USING 1800:"NUMBER OF ITERATIONS: ",IO
1800 IMAGE /,L,18X,22A,2D
1810 PRINT @U: USING 1820:Q$,S0," OHMS",R$,S1," C"
1820 IMAGE /,2L,18X,26A,2D.D,5A/,18X,26A,D.3D,2A
1830 PRINT @U: USING "P,S":
1840 GO TO B OF 1870
1850 PRINT USING 1800:"NUMBER OF ITERATIONS: ",IO
1860 PRINT USING 1820:Q$,S0," OHMS",R$,S1," C"
1870 END
1880 REM TITLE
1890 PRINT @U: USING "P,15L,S":
1900 PRINT @U: USING 1920:W$,A$,U$,D$,T8," - ",T9," C"
1910 PRINT @U: USING 1930:B$,V$,D$,A0," - ",A1," OHMS"
1920 IMAGE /2L,13T,FA,/2L,13T,3(FA,X),FD,3A,FD,2A/
1930 IMAGE 13T,3(FA,X),FD,3A,FD,5A/L
1940 RETURN
1950 REM HEADING
1960 PRINT @U: USING 1990:G$,H$,K$,U$
1970 PRINT @U: USING 2000:V$,U$,U$,N$
1980 PRINT @U: USING 2010:L$,M$,M$,M$
1990 IMAGE 13T,10A,26T,8A,38T,10A,51T,11A
2000 IMAGE 13T,10A,25T,11A,38T,11A,51T,10A
2010 IMAGE 15T,6A,29T,3A,42T,3A,55T,3A/
2020 RETURN

```