Div 563

FILE COPY
DO NOT REMOVE
SEP 16 1981

NBSIR 80-2102

User's Manual for Factor-Jump Thermogravimetry Apparatus, and Associated Programs, Including A General Plotting Program

B. Dickens

Polymer Science and Standards Division
Center for Materials Research
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

March 1981

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

NBSIR 80-2102

USER'S MANUAL FOR FACTOR-JUMP THERMOGRAVIMETRY APPARATUS, AND ASSOCIATED PROGRAMS, INCLUDING A GENERAL PLOTTING PROGRAM

B. Dickens

Polymer Science and Standards Division
Center for Materials Research
U.S. Department of Commerce
National Bureau of Standards
Washington, DC 20234

March 1981

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

Table of Contents

TABLES

Figure

Introduction

A scheme for automation for thermogravimetry was described initially by Dickens, Pummer and Flynn [1] and then further by Dickens [2]. The performance of the apparatus is detailed by Dickens in references [2] and [3]. The apparatus includes a minicomputer which directs the course of the experiment. The computer programs were described initially in reference [4] and in more detail in reference [5]. The present report describes the operation of the automated equipment in minute detail. Several computer programs have been written to collect data using the automated thermogravimetry apparatus, to process thermogravimetric data, and to plot the results. Their operation and, to some extent, their philosophies, are described here.

These programs are:

1. Factor-jump thermogravimetry: TGRUNF.

2. Editing of thermogravimetric results: TGEDIT.

3. Calculation of trimmed means: TGTRIM.

4. General calculation of activation energies from rate data, provide normal probability plots: TGDEPG.

5. Examination of the trend of activation energies with time, extent of reaction, or sample weight (actually a generalized polynomial fitting program): POLGEP.

6. Make publication-type tables of activation energy data: TABLEP.

7. General program to operate interface so that non-routine operations can be easily carried out: BMDP.

8. Generalized plotting program for Versatec printer-plotter: PABS.

TGRUNF is a FORTRAN system of overlays which operates the thermogravimetry apparatus while running on an attached minicomputer. It collects data and provides a real-time calculation of activation energies using only one sample. Typically, 10 to 25 activation energies are obtained per experiment.

TGEDIT uses cut-off limits input by the operator to edit the output file from TGRUNF. Activation energies are discarded if the associated data do not exceed the chosen threshold values. The program also discards those parts of the data which fall outside the desired limits of degree of conversion (extent of reaction) and temperature.

TGTRIM is a FORTRAN program (in overlay form) which reads the edited output file of TGRUNF. The program places the activation energies in ascending numerical order and then computes the weighted and unweighted average activation energy as up to two-thirds of the outer values are chopped off.

TGDEPG is a data processing program (written in FORTRAN) which uses rate data and thermocouple emfs to calculate activation energies. Input data and activation energies are subjected to critical examination.

POLGEP is a FORTRAN program which reads data in almost any format (it may have trouble if there are alphabetic words among the data it reads in) and fits a polynomial of user-chosen degree to the data. The polynomial and its derivative can be evaluated at points specified by the user.

TABLEP is a FORTRAN program which reads the 'save file' of TGRUNF and provides a ready-for-publication type table.

BMDP is a BASIC program which allows the operator to set modules in the MIDAS interface by direct commands or from a previously made file on disc, and to read specified voltages available at the analog scanner for a specified number of times with a user-chosen time delay between readings.

PABS is a FORTRAN program which uses the VERSATEC plotting software and plotter to plot information selected under user control from files in more or less any (repetitive) format.

I. Apparatus Set-Up

The following procedure is intended to be an exhaustive checklist to enable the user to progress from a completely shutdown apparatus to data collection. Some items need not be performed by the user if the apparatus is already switched on and has not been disturbed since the normal termination of a successful run. In that case, go to step I.11 or I.12.

I.1: Rebooting the Computer Executive System

1. Turn on CRT terminal (switch at rear). Ensure that the keyboard is in "remote" with the "TTY" button (top row) depressed, and that it is not in "shift lock" mode.

2. Open doors on all drives of flexible discs.

3. Turn on flexible disc controller (switch at rear).

4. Turn on computer (key at right side of push-button panel).

5. Put disc with FLOXOS system on it in unit 0 with the slot in the disc cover to the rear. Both sides of the disc contain the same information.

6. Using the sequence: "DTA, characters for address, ADD, DTA, characters for contents, WRT", on the hexadecimal panel, enter the bootstrap loader:

ADDRESS	CONTENTS
0050	D500
0052	00CF
0054	4300
0056	0080
0078	16C1

   DTA, ADD and WRT are buttons on the panel. The address contents are numbers given in base 16, where $A_{16} = 10_{10}$ and $F_{16} = 15_{10}$. The computer is now ready to load the operating system into its memory from the disc in drive 0.

7. Flip the IPL switch on the lower front left of the disc controller. The red select light should go to drive 0. If it does not, try two or three times, then try with another system disc or dial another drive to be drive 0. Take care not to have two drives dialed to the same number. The disc drive assignments are changed using the thumbwheels at the front left. (It is rare indeed that you cannot read in the system from one of the discs on one of the drives. If all else fails, the disc ensemble may need service.)

8. Perform "DTA, 50, ADD" on the hexadecimal panel to enter the starting address of 50 for the boot strap program (i.e., it starts from the instruction given in address 50 in the computer).

9. Press RUN on the hexadecimal panel.

10. Flip the IPL switch on the disc controller. The executive system should read into the computer. *FLOXOS or *FLOXOS-R7.1 SYS-28 should appear on the CRT. When this fails to occur, try another disc in the drive, or the disc in another drive, which is then dialed to be drive zero.

11. Pressing "carriage return" on the CRT keyboard and getting the reply

```
*?
*
```

is a useful testing of successful loading. The commands required to use the operating system are given in the FLOXOS manual[6]. (Not FLOS, which is an old version of FLOXOS!)

12. If you are starting with the computer already running, remove old file assignments the safe way by typing the following list (after pressing "INI" if necessary to get back to the operating system):

```
*CH  F0
*CH  F1
*CH  F2
*CH  F3
*CH  D0
*CH  D1
*ZU
```

The F means flexible disc and the number corresponds to the digit dialed on the thumbwheel switch for each drive. The D means rigid disc; D0 is the removable one, D1 is fixed. The operating system provides the * as a prompting character.

I.2: Turning on the Automation System and the Thermogravimetry Apparatus

1. Power must be supplied to both the power supply used for the photocell in the electrobalance and to the electrobalance itself. These supplies are usually left switched on.

2. If necessary, attach the small power plug to the housing of the electrobalance on the Dupont 951 thermobalance. The bulb illuminating the photocell must light up (the light can be seen through the glass envelope). Unsteadiness of > 20 μV in the weight reading when the electrobalance is completely assembled is usually an indication that the contact between the

bulb and the bulb housing must be improved by sandpapering the bulb base. Normally the balance is steady to 1-2 μV.

3. Check that the FAN and DERIVATIVE switches on the electrobalance console are turned to OFF. You may wish to check that power is indeed being supplied to the body of the electrobalance by momentarily turning the fan on.

4. Make sure that the ice point reference cell is switched on, or, if you are using ice in a dewar, that the dewar is completely filled with ice and the thermocouple is in the middle of a tightly packed mass of crushed ice and water.

5. Turn on the MIDAS interface (switch on front left). The various outputs of the interface will be in a random state. Do not switch on the recipients of the control voltages yet (i.e., the flow controller, temperature controller, and pressure controller). Press the "clear" button on the front right of the interface if the error light is on.

6. Switch on the DANA 5900 digital voltmeter. Turn the range switch to "auto". (If "NO" is in the digital display, the voltmeter is not set to a high enough range.)

7. At this point, the MIDAS modules must be initialized. Although this could be done by attaching the CRT to the MIDAS cable, it is more convenient to do it with the programs BMPD (step 8) or TGRUNF (step 9).

8. To find the BMPD program (refer to table 1), type AC BMPD,2D1↲ on the CRT. ↲ means carriage return. Assign the usual output from the program to a write-only unit by typing AS 700. Then type DO MIDAS,D1↲ . (To erase the line when typing on the CRT, type # (shift 3). To erase the latest unerased character, type - (shift 0).)

   When the light has stopped flashing on the disc drive and BASIC has appeared on the CRT, type LOAD 2↲ and wait until BASIC reappears. Then type RUN, and answer the question about title with a carriage return. When you get the question about setting MIDAS, answer KOMP010D0:0;G0:0;↲ .

   This will set all digital-to-analog converters to 0 V dc output and will display the electrobalance reading on the digital voltmeter. (A list of the MIDAS com-

Table 1. Initiation of interface using BMPD and BASIC

(User input is underlined.)

```
*ZU
*AS 700
*AC BMPD.2D1
*DO MIDAS.D1
  BIAS 3572
  END 3882
  BIAS 3882
 BASIC 03-04
 LOAD 2
 BASIC 03-04
 RUN
 GIVE TITLE
  INITIALISE INTERFACE USING BMPD AND BASIC INTERPRTER
  INITIALISE INTERFACE USING BMPD AND BASIC INTERPRETER
 SET INITIAL CONDITIONS, Y OR N?
 Y
 GIVE COMMAND
 D0:0;G0:0;
 SET INITIAL CONDITIONS, Y OR N?
 N
 GIVE INPUT TO BE READ, FINISH WITH -1
 P   (TO STOP PROGRAM ON ILLEGAL INPUT)
 IN-ERR 590
 PAUSE
 *
```

mands appropriate to the thermogravimetry apparatus is given in table 2.) You may proceed to switching on the rest of the apparatus (step 11). If you have not given the illegal input described in table 1, the program will remain in a state where it can transmit commands to MIDAS. Later you can either press "INI" or enter the illegal output to return to the FLOXOS system, which allows you to run a different program.

9. To use the TGRUNF program to initialise the apparatus, first decide whether to use flexible discs or the rigid disc. The rigid disc is preferable.

   The programs which collect data and the raw unedited data files are usually kept on the fixed rigid disc D1. Typing LI D1↵ will give a listing on the CRT of the files on disc D1, but it will be hard to read because it goes by so quickly. A better way is to turn on the printer, and then type AS 362↵ and LI 3D1↵. This will produce a listing on the printer. Verify that the files TGMAIN and BMPD are on the fixed disc. If they are, go to step 9(c). If they are not, they will have to be copied on to it from the appropriate removable rigid disc (TGRUNF) as in 9(a). The removable disc is usually reserved for edited data files and for the programs which perform editing and other calculations on the data. Look in the rigid disc file for the number of the appropriate disc.

9a. To change the removable rigid disc, switch the drive to "OFF" if it is on and wait until the lights to out (about 60 s) before installing a new disc. Follow the procedure on pp. 2 to 8 in the manual "Control Data Cartridge Disk Drive, model 9427H", to install the disc in its drive. Switch the unit on and wait for the 'ready' light to go on (this takes about 60 s). The removable rigid disc is referred to as unit D0, the fixed rigid disc as unit D1. Type DO COPYTG,D0↵ on the CRT. When this procedure is finished, install the TGPROC disc as the removable disc. Type DO SETTG,D1↵ to assign the needed files and then go to 9c.

9b. To use flexible discs, put the TGRUNF flexible disc in flexible disc drive 1 and a TGS flexible disc in drive 2. Use the appropriate disc for the material you are giving to degrade i.e., keep all files for a given material on the same disc. Type DO SETUP,2F1↵ on the CRT (refer to table 3). Turn on the VERSATEC printer (knob at top right).

Table 2. MIDAS Commands

Reset → /

Begin commands → #

Set desired pressure → D _ _ _ _: (in Torr)

Set desired temperature → D _ _ _ _;
9999 = 40 mV (thermocouple emf)

Set flow N_2 → G _ _ _ _:
5000 = 500 scc/min N2

Set flow O_2 → G _ _ _ _;
5000 = 200 scc/min O2

Set time interval → IT _ _ *R
(switch on clock module set to "interval")

Set time → IT _ _ _ _ _ _ R
hr min s

Read time → IQ:;<=>?R (output is hr, min, s as a 6 digit number)

Set voltage to be measured → K _

0 = electrobalance 1 = thermocouple 2 = pressure 3 = N_2 flow specification

4 = O_2 flow specification 5 = actual N_2 flow 6 = actual O_2 flow

Set DVM, read voltage → MP_?Δ*T*U

_ = 0 means no control

_ = 3 means program control

? = 0 means no filter

? = 1 means use filter

Δ = 3 means use 0.1 V scale

Δ = 4 means use 1. V scale

Δ = 5 means use 10 V scale

Literals (i.e., CR, LF and titles) are rarely necessary, but are entered after a $ sign.

Table 3a. Setup of TGRUNF from flexible disc.

User input is underlined.

```
*DO SETTG.F1
LU  PU  NAME
 1  F2  NFILED
 3  62
 6  00
 7  00
 B  F1  TGSEETU
 C  F1  TGFACT
 D  F1  TGPLAT
 E  F1  TGPREC
 F  F1  TGSTAT
 ASSIGN SAVE FOR E'S (6) AND PARAMETERS (9) THEN TYPE 'RUN TGMAIN.F1'
  ?
*AL PE84V2.6F2.2.80
*AC PESAVE.9F2
*RUN TGMAIN.F1
  GIVE TITLE ..........
```

Table 3b. Setup of TGRUNF from rigid disc.

User input is underlined.

```
*DO SETTG.D1
LU  PU  NAME
 1  D1  NFILED
 2  62
 3  D1  PRINT
 6  00
 7  00
 9  D1  NFILEC
 B  D1  TGSETU
 C  D1  TGFACT
 D  D1  TGPLAT
 E  D1  TGPREC
 F  D1  TGSTAT
ASSIGN SAVE FILE FOR E'S (6) AND PARAMETERS (9) THEN TYPE 'RUN TGMAIN.D1'
  ?
*AL PE82V2.6D1.1.80
*AC PESAVE.9D1
*RUN TGMAIN.D1
 GIVE TITLE
  FACTOR JUMP THERMOGRAVIMTERY FOR PE IN VACUUM  PRESS< 10 MICRONS
   YEARE OF GRACE 1978 AD
```

9c. Next to the * displayed on the CRT, type RUN TGMAIN,XX↲ where XX = F1 (flexible discs) or D1 (rigid discs). (Note: from here on, the carriage return ↲ will be omitted. However information is not sent from the CRT to the computer until the carriage return is typed, so all lines must end with a carriage return).

Answer ↲ to all questions until you get to "SET MIDAS, Y or N?". Answer Y. Then type in KOMP010D0:D0;G0:G0; This will set all digital-to-analog converters to 0 V dc output and will display the electrobalance reading on the digital voltmeter. (A list of the MIDAS commands appropriate to the thermogravimetry apparatus is given in table 2.)

You may leave the program in this state for the time being. Later, you will type N to exit from the MIDAS setting routine and then will type B in answer to the question "FORWARD or BACK, F or B?". This will take you back to input of the title for the run.

10. *NOTE:* Programs can be interrupted catastrophically (i.e., the program must be read into the computer again before it can be rerun) by pressing "INI" on the hexadecimal keyboard on the computer. After pressing "INI", it is a good idea to type *SM 0 to zero the available computer memory. Programs can be *interrupted* more gently by typing @ when the program is "busy" on some calculation (but *not* when it is expecting input from you, because @ would then be read as an item of data and would surely give a FORMAT ERROR, and *not* when a new part of the program is being read from a flexible disc). The program can then be continued by typing *CO. BASIC programs (such as BMPD) in the process of calculation can also be interrupted by hitting the "ESCAPE" key on the right hand side of the CRT keyboard. TGRUNF is a FORTRAN program; hit @ rather than ESCAPE to stop it.

11. Turn on the L/N Electromax III controller (switch on front left). Room temperature in the furnace and 0 °C in the ice-point reference cell will give an on-scale deviation to the right. Deviation to the left means the system will heat up if the power supply is switched on, and at this stage means a malfunction somewhere since the MIDAS system should have been initialized to a programmed temperature of 0 °C.

12. If it is warranted by the results of step 11, turn on the Kepco power supply, which supplies power to the furnace as directed by the L/N Electromax III controller. The Kepco meters should show zero output.

13. If you are going to control the pressure, turn on the MKS pressure measuring system (switch at left front). The pressure will probably read near to 760.

14. If the exhaust valve and gas flow are going to be used, attach the flow controller to the end of the silvered furnace manifold, then turn on the Granville Phillips automatic pressure controller by depressing its power switch. The valve should automatically go to "OPEN." If the run is to be in vacuum, do not switch on the pressure measuring system or the pressure controller. Attach the pump assembly to the furnace manifold. If vacuum is currently being applied to the apparatus, you will have to open one of the valves to release it. *Note 1*: Runs can be conducted in air simply by leaving the upstream end of the silvered manifold open. *Note 2*: The pressure controller/ exhaust valve combination cannot handle temperatures > 410 °C. Cooling of gas in the side arm sets up oscillations which the controller is unable to overcome. These temperatures can be used, however, for pressures greater than atmospheric, if the gas flow is bubbled through a glass tube in a cylinder of mercury to give the required pressure. In this case, the exhaust valve is not used.

15. Turn on the flow controller if gas flows are needed. Use the front panel channel switch and the digital display (which should show zero output) to examine the output of each channel at this time. The displays give percentage of maximum capacity (500 scc/min N_2 on channel 1; 200 scc/min O_2 on channel 2).

16. Turn the electrobalance time constant to 1 s.

17. If the initialization process was successful, the DVM should now be displaying the electrobalance signal. If it is not, sending the command KOMP010 via the CRT is the way to proceed, provided the program is still in a state where it will accept MIDAS commands. If necessary, rerun BMPD or TGRUNF and display the electrobalance signal on the DVM*. If the pressure is

---

*There has been a history (now apparently corrected satisfactorily) of the DVM refusing to display the voltage applied while functioning satisfactorily in remote control. If this is the case, enter MT several times on the CRT as a command to MIDAS to trigger the DVM to take a reading, or, better, use a portable DVM to read the voltage at the input to the interface DVM.

less than ∿ 650 mm Hg, open a valve to the atmosphere to equilibrate the pressures inside and out. Place an empty pan on the balance arm or, better, use the arm with the quartz cup, and adjust the balance to -0.170 V by turning the "SUPPRESSION" knob on the front panel and changing riders on the rear of the balance arm (rarely necessary). Keep the total load on the balance < 0.5 g. The DVM on its most sensitive scale will read voltages between ± 0.16 V to 1 μV.

18. Place ∿ 10 to 20 mg of sample in the cup on the balance arm. Because 1 mg is equivalent to 20 mV, this means adding sample to the cup until the voltmeter changes from ∿ -0.17 V to between ∿ +0.03 V and +0.23 V. 0.23 V seems to work well. If you have decided to use more sample than this, it may be necessary to adjust the suppression so that the weight can be read on the most sensitive DVM scale for the most important part of the degradation process. (Transpose the weight of the empty pan accordingly.) The TGRUNF program will automatically select the appropriate DVM scale to read the sample weight. Note that, as a safety measure, the program will unconditionally stop when the voltage corresponding to the sample weight reaches -0.158 V. The small amount of sample between -0.158 V and -0.170 V is usually not worth studying, being the residual of degradation at the 90 to 95 percent stage. Smooth out the grease on the end of the furnace manifold and attach the reaction manifold to the balance housing.

19. Turn the electrobalance time constant to 5 s.

20. If a gas flow is to be used, check the gas cylinders for adequate back-up pressure and the drying tubes for further life. Turn the valves to give ∿ 10 psi in the lines to the flow controller.

21. If the exhaust valve is to be used, consider filling the traps which would protect the valve from contamination. The exhaust may be connected to the house vacuum or to a vacuum pump by choosing the appropriate position on the rear stop cock of the exhaust system. The pump is probably the safer choice.

22. It is a good idea to check the reaction manifold for tightness and remove the air inside, even when using a flow of gas, by pumping the system down to a vacuum using the vacuum pump. (You will have to set the exhaust stop cocks appropriately to do this if you later intend to use a gas flow. Direct the flow to the top manifold in this case.) Leaks are usually at the furnace manifold-balance housing junction. Leaks

exist if the pyrex reaction manifold is not perpendicular to the balance housing, if the O-ring is dirty, and perhaps if the swagelok system at the rear of the flow controller has been jarred. Establish to your satisfaction that the system is sufficiently leak tight (< 25 μm Hg?) by watching the rate of pressure increase when the pump is not connected with the system.

23. If vacuum is not the reaction condition, and you are using the exhaust valve rather than the mercury level to control the pressure, push the "CONTROL" switch on the pressure controller, set the rate knob to "MANUAL" and the "DAMPING" knob to "10." The reference knob should already be set to ∿ 0.6, and the range knob to "10 V." Turn the manual knob to "OPEN" to open the valve. The "OPEN" light flashes when the valve is fully open. Reset the stop cocks so that the line is open to the pump or house vacuum. If 100 percent N_2 is the reaction condition, turn off the valve at the top of the oxygen tank and use "D5000"; as a MIDAS command to open the O_2 channel flow control valve so that the O_2 line will be pumped out.

II. Factor-Jump Experiments Using TGRUNF

The factor-jump method [5,7] calculates activation energies using one sample and a series of short isothermal plateaus. The procedure has been programmed in FORTRAN as the program TGRUNF, which is a series of overlays. This program has been described briefly in references [4] and [5]. Reference [3] describes some considerations associated with successful running of TGRUNF and provides preliminary results on several polymers.

The following paragraphs describe the input to the program. Many of the considerations involved in choosing suitable values for the input are given in reference [3]. The remainder are given here. Examples of inputs and outputs are dotted throughout the text. For more detail, refer to the program listing in Appendix A. If TGRUNF is already available in the computer because of the set-up procedure given above, answer the question about input of MIDAS commands with N and reply with B to the question about whether you want to go forward or back to the beginning of the program. Then continue at step 7 below.

II.1 Detailed Procedure

[Assuming computer operating system is running.]

1. Place the overlay disc TGRUNF into disc drive 1 if you are using flexible discs. Ensure the TGRUNF rigid

disc files are available on the fixed disc if you are going to use the rigid disc (this is the more reliable mode of operation). See Section I.9(a).

2. Press "INI" on the hexadecimal panel to return to the FLOXOS executive system. * FLOXOS should print on the CRT. Consult section I.1 if pressing INI fails.

3. If you are using the flexible discs, you will need a second disc for the output Save file. This second disc also provides space for the initial parameters (NFILEC) and for temporary storage of data (NFILED). Put the appropriate disc in unit 2. Refer to Appendix A if you have to generate a new disc.

   In the case of the rigid disc, there is sufficient space to keep all this information and the programs on the one disc.

4. Type DO SETTG,XX where XX=F1 for the flexible disc and D1 for the rigid disc. Refer to table 3. Successful execution of these commands returns * to the CRT.

5. After * is printed on the CRT, make any needed assignments for the following files:

   3 = permanent record of progress of run. If you are using the rigid discs, this has already been assigned automatically to the rigid disc as file PRINT. If you are using the flexible discs, you may wish to assign it to a disc file on a flexible disc, probably in drive 0 (e.g., AC TGDAT1,3F0). If you use a flexible disc file, you will need to have the whole disc available. Also, it will be touch and go as to whether the sample or the disc space will be used up first. With the rigid disc there is plenty of space to assign the needed 75 to 100 cylinders. You may also assign the print file to the printer using AS 362, but this is not recommended because ∿ 8 s are wasted between data points waiting for the developer in the printer to pump up. The rigid disc is definitely the best choice.

   6 = output of activation energy data: energy, sigma(E), sample weight, rates of weight loss, temperatures, etc. The usual procedures for this are: (a) to use a file named with an initial letter code for the substance, then either the month, day, year, or the reaction conditions and run number, e.g., M63077 or MVAC5. The latter designation is probably better, e.g., AL MVAC5,6XX,2,80 allocates (creates) a file called MVAC5 of line length 80 characters, and two cylinders long (which should be adequate) on disc

drive XX (F2 for flexible disc, D1 for rigid disc). Note that all file names must be six characters or less. (See the Cautionary Remarks Appendix in this report and the FLOXOS write up for explanations of AL and AC, etc.) Two cylinders will hold 96 lines of output on the flexible disc, three times that amount on the rigid disc. Three overflow cylinders are automatically provided. Each activation energy and its associated data such as temperatures and rates take five lines. The title to the file takes two lines. This size file is almost always adequate in size.

7 = file to log raw data for later reprocessing, rarely used.

6. After assignments in 5 have been made, type LU, affirm that file assignments are similar to those in table 3 (with 6 and 9 included), then type RUN TGMAIN,XX where XX=F2 or D1.

7. When prompted by the program, enter the title of the experiment via the CRT.

8. Complete the choice of control parameters. The uses of the various parameters are indicated in table 4. *NOTE: The program only calculates activation energies when the temperature has been measured. It tests for input 1; therefore, always assign temperature to input 1 and the sample weight to input 0.*
The choice of a typical parameter block is given in tables 5 and 6. File 9 is assigned automatically in SETTG to be file NFILEC on drive F2 or D1. If you did not assign save files and print units because you were merely initializing the system, you may assign them when the program returns to the executive program to allow specification of the save file for the parameters. The file assignment commands are given in II.5 above.

9. Choose initial levels, P_i, and maximum values for each of the four factors (temperature, pressure, flow of N_2, flow of O_2). Typical values and the method of inputting them are shown in table 7. The values to use are discussed below.

9a. Temperature: The ideal starting temperature is one at which the sample will lose weight at a rate between 15 and 40 to 60 μV/s. This obviously depends on the sample size. The ideal temperature difference to use between adjacent temperature plateaus depends on keeping *both* rates of weight loss in the range 15 and

Table 4. Initialization of Program Parameters

Using Subroutine Prime

Requires N, I and VALUE in FORMAT (2I3, F10.0).
i.e. N -- right justified to column 3, no decimal point
I -- right justified to column 6, no decimal point
VALUE -- between columns 7 and 16 inclusive, include decimal point

To Change	N	I	Value	Details
NTHCP	1	0	J or K	-Specify thermocouple type.
TSCALE	2	0		-Specify scale to apply to bring DAC voltage into correspondence with thermocouple voltage at temperature controller.
SKIP	3	0	30 to 120	-time in seconds to wait between sets of readings, i.e., read weight every "skip" seconds.
FACTOR	4	1 to 8	order	-Specify order in which inputs on analog scanner are to be read as factors.
FACFUN	5	1 to 8	0	-Raw reading, r, untransformed.
			1	-Use log (r)
			2	-Use r^2.
			3	-Use 1/r.
DERIVS	6	1 to 8	0	-No derivative for the Ith factor
			1	-Calculate derivative and check precision for Ith factor.
INPUTS	7	1 to 8	input number on scanner	-Specify which inputs on analog scanner are to be read.
SCALES	8	1 to 8	scale factor	-Scale to be applied to voltage reading on input to put it in appropriate physical units
NRANGE	9	1 to 8	3	-Use 0.1v scale on DVM for reading this factor (DVM will over-range to 0.16 v)
			4	-use 1v scale
			5	-Use 10v scale
NREADS	10	1 to 8	≥1	Number of individual readings of an input to be taken in succession (without waiting "skip" seconds) and averaged together to give "reading" and its estimated standard deviation.
NTERMS	11	1 to 8	0 to 5	-Number of terms to use in polynomial which is to be fitted to the Ith factor.
CHI TEST	12	1 to 8		-Target value in chi square test for Ith factor.
PTIMES	13	1 to 8		-Maximum time allowed for a factor to steady down after change.
PRECF	14	1 to 8		-Target precision for factor extrapolation.
PRECR	15	1 to 8		-Target precisions for rate extrapolations.
NFACTS	16	0	≥1	-Number of factors to be measured in this experiment.
NRATES	17	0		-Number of rates to be considered.
TIMLIM	18	0	>0	-Maximum time allowed for measurement of any one plateau.

BAD PTS	19	0	>0	-Maximum number of data points to throw away before resetting steadiness indicators.
EQUIL	20	0	>0	-Time in seconds to allow after resetting of factors before data are measured.
CHECKS	21	0	>0	-Number of data points between checks of currently attained precision.
ENDSIG	22			-Minimum number of standard deviations adjacent readings must differ by if experiment is not to be terminated (tests for whether some sample is still left).
RATMIN	23			-Minimum value of rate of weight loss to allow. Failure of this test increases the temperature unconditionally by TJUMP.
RATMAX	24			-Maximum value of rate of weight loss to allow. Failure here unconditionally decreases the temperature by TJUMP.
TJUMP	25			-Number of degrees to jump temperature by unconditionally when so indicated by RATMIN and RATMAX tests.
I/O UNITS		0	0 to 15	-Assign logical units in program to actual peripherals on computer.
	26			-Input from user
	27			-Program instructions to user
	28			-Print file.
	29			-(Disc) file for buffered experimental (rarely) data (for later processing; this is used.)
	30			-User console (for duplicate of print file)
	31			-Unit to save initalized parameters on (Disc) file for data transfers between subroutines.
	32			
	33			-Summary of activation energy determinations.
M_i	34	1 to 5	1 to 5	-Exponents M_i of polynomial.
PRECE	35	0	>0	-Target precision for activation energy.
	0	0	0	-input required to stop initialization procedure.

Table 5. Assignment of Controlling Values for TGRUNF Program.

(Dialogue on terminal; user input is underlined.)

```
 READ PARAMETERS FROM SAVE FILE?
 0=NO, 1=YES, GIVE IN I1 FORMAT
1
 GIVE FILE NUMBER IN I2 FORMAT
 PROGRAM WILL THEN PAUSE, GIVE CONTINUE WHEN READY
 FILE WILL BE REWOUND FIRST
 9
PAUSE 5
  PAUSE
*CO  file assignments can be changed here if necessary.

 PARAMETERS INITIALISED AS FOLLOWS:
 NTHCP =              E    TSCALE =  243.73   SKIP=  30.
 FACTORS =       1       2       0       0       0       0       0       0
 FACFUN =        0       0       0       0       0       0       0       0
 DERIVS =        1       0       0       0       0       0       0       0
 INPUTS =        0       1       2       3       4       5       6       7
 SCALES =    1.000   1.000   0.010   1.000   1.000   1.000   1.000   1.000
 NRANGE =        3       3       5       5       5       5       5       5
 NREADS =        3       3       3       3       3       3       3       3
 NTERMS =        2       1       0       0       0       0       0       0
 CHI TEST=   2000.   1000.   1000.   1000.   1000.   1000.   1000.   1000.
 PTIMES =    300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 PRECF =    100.00  100.00  100.00  100.00  100.00  100.00  100.00  100.00
 PRECR =     30.00   30.00   30.00   30.00   30.00   30.00   30.00   30.00
 NFACTS =    2 NRATES =    1 TIMLIM   = 500. BAD PTS =   8 EQUIL = 200.
 CHECKS  =   75 END SIG =   5.0 RATMIN =   12.5 RATMAX =   40.0 TJUMP = 20.0
 TC TO D =    0.17023E-01   -0.22097E-06    0.54809E-11   -0.57670E-16
 IO UNITS=       5       5       3       7       8       1       6
 POLNOM EXP=     1       2       3       4       5
 PRECISION IN E ACT =   25.00
 DO YOU WANT TO CHANGE ANY CONTROL PARAMETERS?
 IF NOT, GIVE PARAMETER OF ZERO
 IF SO, ENTER PARAMETER NUMBER, ELEMENT IN PARAMETER ARRAY,
 NEW VALUE IN I3,I3,F7.3 FORMAT
 INTEGER VALUES WILL BE INTEGERISED, GIVE NEW VALUE WITH
 DECIMAL POINT
 12  3 99999.
ERR 03  9032 (See Note below.)
 ANOTHER CHANGE?
0
```

Note: This error message was generated because the program converts the third quantity in the line into an integer. 99999 is greater than the maximum integer allowed, but will be used in fact only as floating point. The message may be ignored in such cases.

Table 6. Typical Choice of Controlling Values for TGRUNF Program.

(Dialogue on terminal; user input underlined.)

```
 TEST OF RESPON & BIASVL ROUTINES 8/4/77    RESPONSE TIMES AND BIAS
 VOLTAGES WILL BE DETERMINED

 TYPE E THERMOCOUPLE=NI-CR/CU-NI,EXPRESSIONS GOOD
FROM -0.05 TO +0.04, 0 TO 400 DEGREES C.(TABLE A5.2.3,P.307
 NBS MONOGRAPH 125,1974)
 OUTPUT PARAMETERS ON SAVE FILE
 0=NO,1=YES, IN I1 FORMAT           'yes' would return control to
0                                   executive program and would allow
                                    reassignment of files if required.
 PARAMETERS INITIALISED AS FOLLOWS:
 NTHCP =               E   TSCALE =  243.73   SKIP=  30.
 FACTORS =       1       2       0       0       0       0       0       0
 FACFUN =        0       0       0       0       0       0       0       0
 DERIVS =        1       0       0       0       0       0       0       0
 INPUTS =        0       1       2       3       4       5       6       7
 SCALES =    1.000   1.000   0.010   1.000   1.000   1.000   1.000   1.000
 NRANGE =        3       3       5       5       5       5       5       5
 NREADS =        3       3       3       3       3       3       3       3
 NTERMS =        2       1       0       0       0       0       0       0
 CHI TEST=   2000.   1000.  99999.   1000.   1000.   1000.   1000.   1000.
 RTIMES =    300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 PRECF =    100.00  100.00  100.00  100.00  100.00  100.00  100.00  100.00
 PRECR =     30.00   30.00   30.00   30.00   30.00   30.00   30.00   30.00
 NFACTS =    2 NRATES =    1 TIMLIM  = 500. BAD PTS =  8 EQUIL = 200.
 CHECKS  =   75 END SIG =  5.0 RATMIN =  12.5 RATMAX =  40.0 TJUMP = 20.0
 TC TO D =   0.17023E-01  -0.22097E-06   0.54809E-11  -0.57670E-16
 IO UNITS=       5       5       3       7       8       1       6
 POLNOM EXP=     1       2       3       4       5
 PRECISION IN E ACT =  25.00
```

Table 7. Specification of starting values and maximum values of temperature, pressure, and the flow rates.

User input is underlined.

```
READY TO SET UP FACTOR LEVELS AND CHANGES
ONE PER LINE
GIVE INITIAL T AND MAXIMUM T IN DEGREES CELCIUS
F7.2 FORMAT, ONE PER LINE
300.
400.
GIVE INITIAL P AND MAXIMUM P IN MM HG IN F7.2 FORMAT
MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 MM HG
800.
900.
GIVE INITIAL N2 FLOW AND MAXIMUM FLOW IN SCC/MIN
MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 SCC/MIN
400.
500.
GIVE INITIAL O2 FLOW AND MAXIMUM IN SCC/MIN
MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 SCC/MIN
100.
200.
FACTOR LEVELS AND MAXIMA FOLLOW:
NO.   INITIAL   MAXIMUM
  1    300.00    400.00
  2    800.00    900.00
  3    400.00    500.00
  4    100.00    200.00
GIVE FACTOR NUMBER (1 TO 4) FOR FURTHER CHANGES
0 = NO CHANGES
0
```

40 to 60 μV/s. The actual choice depends on the activation energy. See table 8 for guidance. Values in the range 5 to 15 are appropriate.

When the rate of weight loss falls off continuously as the reaction proceeds, a sequence such as +8, -6, +8, -6... will probably keep the rate within the appropriate range. When the rate of weight loss goes through a maximum, use a sequence such as +8, -8, +8, -8...., unless you want to study the part of the degradation occurring after the maximum, in which case you should use a gradually ascending series.

Remember that gas flows above ∿ 415 °C are not stable when the exhaust valve is used. The hot gas cooling in the side arm of the furnace-specimen chamber introduces instabilities which the pressure controlling system cannot keep up with (partly because the pressure must go past the desired setting before the servo-driver exhaust valve reverses itself and partly because the valve action is slow).

9b. Pressure: Ordinarily, the pressure is kept constant at 800 to 810 mm Hg. Slow flow rates of N_2 (50 to 100 scc/min = 1/2 to 1 mm/s over the sample) at this pressure seem to give chemically meaningful activation energies, free of the effect of thermal history of the sample. Degradation in vacuum is generally unsatisfactory, partly because the activation energies seem to include the physical effect of evaporating performed small molecules and molecules up to different maximum molecular sizes depending on the temperature, and partly because the surface of the molten polymer is very viscous and the escaping volatiles form bubbles which impact shocks to the balance as they burst. However, changing pressure and/or flow rate may be a useful way of widening the range of oxygen concentration available at the degrading polymer. Slow flow rates (< 100 scc/min) and high pressures (> 600 mm Hg) make for slow re-establishment of control by the exhaust value. It is not worthwhile to attempt to give a table of equilibration times under various conditions because the times depend on the initial and final conditions. Ensure to your satisfaction that the system will stabilize within the time period specified by the EQUIL parameter (usually 150 to 250 s) or use the mercury valve described earlier.

9c. Gas flows: The flow meter is intended to be precise to within one percent of full scale, but may show a bias unless it is level. We have checked the absolute flow and feel that the meter can be trusted to the

Table 8. Ratios in rates for various values of E, ΔT and T.

	E(kcal/mole)					
$\Delta T/T$(°C)	15	20	30	40	50	60
10/250	1.32	1.44	1.73	2.08	2.49	2.99
15/250	1.51	1.73	2.28	2.99	3.94	5.18
10/350	1.21	1.29	1.47	1.67	1.90	2.17
15/350	1.34	1.47	1.79	2.17	2.63	3.19

above one percent. This implies a minimum flow of (say) 25 scc/min for N_2 and 10 scc/min for O_2. It is probably best to use O_2 already diluted with N_2 if you want reliable rates with less O_2 flow than this.

10. Choose the input values for the design matrix, D_{ij}, which is used to update the current values, C_i, of factor i according to

$$C_i = P_i + D_{ij}$$

where P_i is the previous level of factor i. Refer to table 9.

11. Enter one for implementation of steadiness checks (via χ^2 value obtained from fit of five latest readings to polynomial of maximum degree 2) or 0 for no steadiness checks. Refer to table 10. When polymers are degrading in vacuum, it is usually best to forego steadiness checks, which tell a horrible story, and which waste time better spent on collecting as much data as possible. For well-behaved conditions such as degradation in oxygen-containing or inert gas atmospheres, steadiness checks may be used, but may not be necessary even then.

12. Enter 1 to implement filtering of input to DVM (this is the usual mode).

13. Usually enter 0 to avoid reading other experiments while this one is in progress.

14. Usually enter 0 (when requested after next overlay has read in) to forego measurements of factor response time.

However, if you want to determine the factor response times, enter 1 instead of zero and then enter for each of the four factors (temperature, pressure, flow of N_2, flow of O_2) either 0 (response time will be read in as the next input value) or 1 (response time will be measured) as shown in table 11. In the latter mode of operation, the program will set initial values of the parameters, will wait until their levels stabilize, and will then set them to new values and wait again until their levels restabilize. A typical set of results is shown in table 12. The stabilization time is considered to be the response time. This part of the program is intended to be used to check the functioning of the apparatus as it is being tuned up after some major changes such as replacement of the

Table 9. Build-up of Design Matrix in TGRUNF Program.

(Dialogue on terminal; user input is underlined.)

```
READY TO BUILD DESIGN MATRIX
I.E., VECTORS FOR CHANGES IN FACTOR LEVELS
MAXIMUM OF 16 VECTORS, ONE PER LINE
GIVE LINE NUMBERS THEN 4 COMPONENTS
FORMAT(I2/(F6.2))
END WITH LINE NUMBER = 0
1
-10.
0.
0.
0.
 1   -10.000      0.000      0.000      0.000
2
12.
0.
0.
0.
 2    12.000      0.000      0.000      0.000
0
INPUT NUMBER OF VECTORS IN I2 FORMAT
2
DESIGN MATRIX FOLLOWS
 1   -10.000      0.000      0.000      0.000
 2    12.000      0.000      0.000      0.000
CHANCE FOR FURTHER CHANGES? 0=NO, 1=YES, IN I2 FORMAT
0
```

Table 10. Entering Steadiness Checks Choice into TGRUNF Program.

(Dialogue on terminal; user input is underlined.)

```
STEADINESS CHECKS TO BE MADE DURING DATA COLLECTION?
 GIVE 0 FOR NO, 1 FOR YES, I2 FORMAT
1
FILTER INPUTS TO DVM? 0=NO,1=YES (YES TAKES
0.5 SEC/READ) GIVE IN I2 FORMAT
1
READ OTHER EXPERIMENTS AT SAME TIME? 0=NO,1=YES,
GIVE IN I2 FORMAT
0
FACTOR LEVELS AND CHANGES FOLLOW:
  1   250.000      1.000
  2   400.000      0.000
  3   500.000      0.000
  4     0.000      0.000
DESIGN MATRIX FOLLOWS
 1   -10.000      0.000      0.000      0.000
 2    12.000      0.000      0.000      0.000
STEADINESS CHECK = 1 WHERE 0=NO,1=YES
INPUT FILTER TO DVM = 1 WHERE 0=NO,1=YES
READ OTHER EXPTS = 0 WHERE 0=NO,1=YES
```

Table 11. Procedure for Testing Factor Responses in TGRUNF.

(Dialogue on terminal; user input is underlined.)

```
MEASURE FACTOR RESPONSE TIMES? 0=NO, 1=YES
1

READ IN FACTOR RESPONSE TIMES OR MEASURE THEM?
ENTER 0 FOR READ
      1 FOR MEASURE
IN I1 FORMAT FOR TEMP     PARAMETER
1

READ IN FACTOR RESPONSE TIMES OR MEASURE THEM?
ENTER 0 FOR READ
      1 FOR MEASURE
IN I1 FORMAT FOR PRESS    PARAMETER
1

READ IN FACTOR RESPONSE TIMES OR MEASURE THEM?
ENTER 0 FOR READ
      1 FOR MEASURE
IN I1 FORMAT FOR N2 FLOW  PARAMETER
1

READ IN FACTOR RESPONSE TIMES OR MEASURE THEM?
ENTER 0 FOR READ
      1 FOR MEASURE
IN I1 FORMAT FOR O2 FLOW  PARAMETER
1

TARGET VALUES OF FACTORS IN RESPON ARE:
     1      1     250.0
     1      2     270.0
     2      1     400.0
     2      2     450.0
     3      1     200.0
     3      2     240.0
     4      1      50.0
     4      2      60.0
DO YOU WANT TO CHANGE ANY VALUES?
 IF SO, ENTER:
   FACTOR NUMBER, 1=T, 2=P, 3=FN2, 4=FO2
   1 OR 2 FOR FIRST OR SECOND VALUE OF A FACTOR
   VALUE ITSELF
ALL IN (2I3,F10.0)
%%  END WITH FACTOR NUMBER OF ZERO (0) %%
0
```

Table 12. Output of Factor Response Checking Routine in TGRUNF.

```
FACTOR RESPONSE TIMES  FOLLOW:
 FACTOR    LOW   HIGH   TIME  CHISQ  MEASURED? TIMES & CHI VALUES
FACSET                 (the heading applies only to the long lines below.)
    250.0          1 }
   14  *D 4185;      }  set temperature to low then high values.
FACSET               }
    270.0          1 }
   14  *D 4558;
TEMP        250.   270.   153.  213.5    1   224.     83.   309.    118.
FACSET
    400.0          2 }
   14  *D 4000:      }
FACSET               }
    500.0          3 }
   14  *G 5000:      }
FACSET               }  Set pressure, N2 flow and O2 flow under program
      0.0          4 }  control.  These must all be set so the pressure
   14  *G 0000;      }  response is measured under well defined conditions.
FACSET               }  The factors indicated by the last number in the
    450.0          2 }  line below FACSET.  The level of the factor is
   14  *D 4500:      }  the first number in these lines.
FACSET               }
    500.0          3 }
   14  *G 5000:      }
FACSET               }
      0.0          4 }
   14  *G 0000;
PRESS       400.   450.   178.50719.0    1   132.    224. 72384. 29054.
FACSET
    200.0          3 }
   14  *G 2000:      }  Set N2 flow
FACSET               }
    240.0          3 }
   14  *G 2400:
N2 FLOW     200.   240.   126.  500.4    1    82.    170.   218.    782.
FACSET
     50.0          4 }
   14  *G 1250;      }
                     }  Set O2 flow
$$$ RANGE CHANGE FROM  5 TO  4 ON INPUT  6 $$$
                     }
FACSET               }
     60.0          4 }
   14  *G 1500;
O2 FLOW      50.    60.    82.   95.4    1    85.     79.   103.     87.
```

furnace. It is rarely done just before the real experiment is begun but could be used to check how well previously untried experimental conditions would be controlled. In particular, it determines the probable extrapolation time between plateaus.

15. Enter your choice for the treatment of bias voltage. Refer to table 13.

    1 = set bias voltage equal to zero
    2 = read it in
    3 = measure it
    0 = terminate input, no more bias voltages need be considered.

    The usual input is 1 for each parameter for which a voltage will be read. This sets that particular bias voltage to zero. The parameters are:

    sample weight
    thermocouple reading
    pressure
    specified flow on channel 1 (N_2)
    specified flow on channel 2 (O_2)
    actual flow on channel 1
    actual flow on channel 2

    Alternatively, the bias voltages on some or all of these lines can be determined. Refer to table 13 for the sequence of operations.

    If you give 2 above, the bias voltage and its standard deviation must then be given on the next two lines, each with a decimal point and carriage return.

    If you enter 3 above, the program will set the parameter to a preselected value (100 °C for temperature, 400 mm Hg and 300 scc/min N_2 for pressure, 0 scc/min for the flows of N_2 and O_2) and will correct the actual reading by the preset value to obtain the bias voltage. It will indicate problems with unsteadiness (as defined by the values of PTIME and CHTEST in the control parameter table input in paragraph 8).

    After the bias voltages have been determined, the various factor levels are not reset. They will be set to your chosen values as soon as the next overlay is read in. A typical output for the determination of bias voltages is given in table 14. The results are saved and used in the program for the current run only. They must be given to the program again the next time it is run (the program starts by initializing all its parameters).

Table 13. Initialisation of TGRUNF Routine to Determine Bias Voltages.

(Dialogue on terminal; user inputs are underlined.)

```
THIS ROUTINE IS SET UP FOR:
WEIGHT AS "FACTOR" 1
TEMPERATURE AS FACTOR 2
PRESSURE AS FACTOR 3
FLOW CHANNEL 1 AS FACTOR 4
FLOW CHANNEL 2 AS FACTOR 5
FACTORS 2 TO 5 ARE PRESET FOR THE BIAS MEASUREMENTS

FOR EACH FACTOR, IS BIAS VOLTAGE TO BE:
SET = 0 (J=1)
MEASURED (J=2)
OR READ IN (J=3)?
TERMINATE INPUT WITH J=0
GIVE J FOR FACTOR  1 IN I1 FORMAT
2
GIVE J FOR FACTOR  2 IN I1 FORMAT
2
GIVE J FOR FACTOR  3 IN I1 FORMAT
2
GIVE J FOR FACTOR  4 IN I1 FORMAT
2
GIVE J FOR FACTOR  5 IN I1 FORMAT
2
GIVE J FOR FACTOR  6 IN I1 FORMAT
0

BIAS VOLTAGE DETERMINATIONS FOLLOW:

FACSET
    100.0          1
   14  :D 1541;
FACSET
    400.0          2
   14  :D 4000:
FACSET
    300.0          3
   14  :G 3000:
FACSET
    300.0          3
   14  :G 3000:
FACSET
    100.0          4
   14  :G 2500;
```

Table 14. Determination of bias voltages

User input is underlined.

```
THIS ROUTINE IS SET UP FOR:
WEIGHT AS "FACTOR" 1
TEMPERATURE AS FACTOR 2
PRESSURE AS FACTOR 3
FLOW CHANNEL 1 AS FACTOR 4
FLOW CHANNEL 2 AS FACTOR 5
FACTORS 2 TO 5 ARE PRESET FOR THE BIAS MEASUREMENTS

FOR EACH FACTOR, IS BIAS VOLTAGE TO BE:
SET = 0 (J=1)
MEASURED (J=2)
OR READ IN (J=3)?
TERMINATE INPUT WITH J=0
GIVE J FOR FACTOR  1 IN I1 FORMAT
2
GIVE J FOR FACTOR  2 IN I1 FORMAT
2
GIVE J FOR FACTOR  3 IN I1 FORMAT
0

BIAS VOLTAGE DETERMINATIONS FOLLOW:


FACSET
    100.0          1
   14  ♦D 1541;


$$$ RANGE CHANGE FROM  3 TO  4 ON INPUT  0 $$$


BIAS VOLTAGES AND SIGMAS FOR FACTOR INPUTS
FACTOR      VOLTAGE         SIGMA
     1 -0.1736267E+06  0.5773553E+01
     2  0.2924219E+02  0.3511883E+01
```

16. Set MIDAS as and if required by typing MIDAS-type commands in directly from the CRT (see table 15). The format of the commands is given in table 2.

17. Input the initial and final sample weights in microvolts. Refer to the first half of table 16. Although the factor-jump method does not require the initial and final sample weights to determine an activation energy, these quantities are necessary to calculate a degree of conversion and thus relate the activation energy in this sample to that obtained in other samples.

18. Give the appropriate information for preconditioning the sample. Refer to the second half of table 16. If you are using the exhaust valve, use the preconditioning time to adjust the value manually to obtain the desired pressure (using the CLOSE-OPEN knob on the Granville-Phillips pressure controller) and then turn the RATE knob on the controller fully clockwise to give the maximum response of the value to changes in pressure. Once the mercury valve has been set at a given immersion depth of the exhaust tube in the mercury, it needs no further attention until a new pressure is required.

This is the end of the operator input; the program now directs the flow of the experiment. Note, however, that the program will stop if the sample weight goes more negative than -0.158 V, if adjacent weight readings do not differ by (ENDSIG)(σ_W), or if the maximum allowed value of a parameter is exceeded. Any of these conditions provides a normal termination of a run. In fact the run continues until one of these conditions occurs.

III. Editing of Activation Energy File from TGRUNF

The program TGRUNF cannot influence the course of the experiment until it has obtained a result. For example, until it obtains a rate of weight loss below the minimum, it will not adjust the temperature categorically upward. Even though such a low rate of weight loss has been obtained, the TGRUNF program still calculates the activation energy. It is best therefore to set the allowed minimum rate above the lowest rate which still gives reliable results. The output file written by TGRUNF can later be "edited" using the program TGEDIT.

TGEDIT is written in FORTRAN and is listed in Appendix B. It allows corrections to be made to all temperatures in the file, with subsequent recalculation of activation energies. It allows activation energies to be discarded for the following reasons:

Table 15. Initialisation of Interface Using Commands Transmitted by TGRUNF, then Return to Beginning of TGRUNF Program.

(Dialogue on terminal; user input is underlined.)

```
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
Y
 GIVE COMMAND, 36 CHARACTERS MAXIMUM
 SHARP SIGN WILL BE ADDED BY COMPUTER
KOMP000
     38  #KOMP000
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
Y
 GIVE COMMAND, 36 CHARACTERS MAXIMUM
 SHARP SIGN WILL BE ADDED BY COMPUTER.
D0:0;
     38  #D0:0;
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
Y
 GIVE COMMAND, 36 CHARACTERS MAXIMUM
 SHARP SIGN WILL BE ADDED BY COMPUTER
G0:0;
     38  #G0:0;
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
N
 GO FORWARD OR BACK? ENTER =F= OR =B=
B
 GIVE TITLE FOR RUN, 2 LINES, 72 CHARS/LINE MAXIMUM
```

Table 16. Initialisation of MIDAS, specification of sample weight and bake-out parameters.

User input in underlined.

```
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
Y
 GIVE COMMAND, 36 CHARACTERS MAXIMUM
 SHARP SIGN WILL BE ADDED BY COMPUTER
K1MP000
    38  K1MP000
 SET INITIAL CONDITIONS ON MIDAS, Y OR N?
N
 GO FORWARD OR BACK? ENTER =F= OR =B=
F

 GIVE INITIAL AND FINAL WEIGHTS IN MICROVOLTS, WITH SIGN
  ON 2 LINES IN F15.0 FORMAT.
  INITIAL WEIGHT OF 0 WILL CAUSE PROGRAM TO USE FIRST
  WEIGHT READING IN DATA COLLECTION AS INITIAL WEIGHT
  I.E., AFTER BAKEOUT OF SAMPLE

 300000.
 -175000.

 INITIAL AND FINAL WEIGHTS ARE:        300000.        -175000.
 OK OR NOT, Y OR N?

 DO YOU WANT PRECONDITIONING OF THE SAMPLE? Y OR N?

Y
 GIVE BAKEOUT TIME, TEMPERATURE, PRESSURE, FLOW N2, FLOW O2
 EACH ON A SEPARATE LINE IN FORMAT F10.0

 1800.
 290.
 800.
 400.
 100.
 BAKE TIME IN SECONDS IS  1800.
 BAKE TEMPERATURE IN DEGREES CELCIUS IS   290.
 BAKE PRESSURE IN MM HG IS   800.
 FLOW OF N2 IN SCC/MIN IS   400.
 FLOW OF O2 IN SCC/MIN IS   100.

 OK OR NOT, Y OR N?
Y
```

(1) the extent of reaction does not fall within user-specified limits;

(2) a rate of weight loss is below a user-chosen minimum;

(3) the percentage error in at least one of the rates (i.e., 100 $\sigma(r)/r$ is above a user-chosen maximum);

(4) a temperature is below a user-chosen minimum

(5) a temperature is above a user-chosen maximum

(6) an estimated error in a temperature is above a user-chosen maximum; and

(7) the estimated error in the activation energy is above a user-chosen maximum.

Test 1 may be used to select activation energies when the degradation contains several steps or when the sample or the rate-determining process slowly changes. Tests 2 and 3 obviously ensure prespecified precision in the rates of weight loss. Tests 4 and 5 are designed to cull out activation energies over a small range in a degradation which consists of several different chemical steps. Test 6 ensures that the temperatures are adequately determined. Test 7 serves to discriminate against cases which were border-line but passed in tests 2, 3 and 6. In these cases the errors pile up in $\sigma(E)$.

The TGEDIT program has been found very useful in culling out untrustworthy measurement according to prespecified and more or less objective criteria. It produces an edited version of the input while leaving the original input file intact. The sequence of operations is shown in Appendix B.

IV. Estimation of Average Activation Energy by Trimmed Means: TGTRIM

One accepted [8,9,10] way of estimating the mean of a set of numbers, some of which may be in error, is to place the numbers in order of increasing magnitude and calculate the mean as more and more extreme values are removed from consideration. The system of FORTRAN overlays TGTRIM does just that for activation energies contained in an output file from TGRUNF. The listing of the program is given in Appendix C, which also contains examples of the input and output.

The program calculates the weighted and unweighted means and their estimated standard deviations for the surviving subset. It also calculates the corresponding values for the Winsorised subset, where the rejected values have been replaced by the nearest unrejected value. In this case, standard deviations are estimated using the number of degrees of freedom associated with the subset rather than the complete set. The most reasonable values to quote are the unweighted mean from the trimmed set and the unbiased (i.e., unweighted) estimate of the standard deviation from the corresponding Winsorised subset.

V. General Calculations of Activation Energies from Rate Data: TGACTE

The program TGDEPG [1] calculates activation energies from temperature and rate data. Also, it uses normal order statistics and probability plots in a partly intuitive and definitely exploratory attempt to provide a diagnostic procedure for removing atypical values. Examples of its use and the program listings are given in Appendix D. The activation energy and its standard deviation are obtained from the Arrhenius equation using data which pass a test based on assigned standard deviations. The program estimates an absolute scale for the standard deviations associated with data points, assuming in the present version* that the distributions of log $\hat{\sigma}$ [2], the data, and the derived E values are all normal. Some idea of the scale of standard deviations is required if data are to be examined for atypical points and if the standard deviation of the weighted average is to be meaningful (as is required when various weighted averages are to be compared). Of course, it is good practice to examine the conditions generating atypical data for insight into the functioning of the apparatus or changes in the rate limiting step during the experiment. It is also good practice to redo experiments which contain ambiguous results, and to replicate experiments in general to obtain an independent estimate of the precision. Although several methods for detection of outliers exist [12-18], none makes use of standard deviations on individual values. In particular, Tietjen and Moore have shown [16] by computer analysis of 5000 synthesized data sets that their test is among the most powerful for detecting inhomogeneity in univariate samples. Specifically, they mixed random selections from a population

*A version with the more appropriate $\underline{t}$ statistics has stubbornly proved to be too large for the computer. However, for more than 10 to 15 data, there is little to be gained by substituting $\underline{t}$ for normal statistics.

distributed normally with mean 0 and standard deviation 1 with selections from a normal population of mean λ and standard deviation 1, where $\lambda = 0,1,2,3$, and 4, in proportions ranging from $\sim$ 10 to 25 percent. They compare the sums of squares of deviations from the average value for the complete set of data and a subset. In this way they tend to avoid the "masking" effect several outliers on one end of a sample would have on the sample mean (for the subset the calculated mean will approach that of the population in favorable cases). The ratio of sums of squares of deviations for subset and complete set is then used in an F-like test to see if the subset and complete set differ significantly.

Their test has not been applied as a routine procedure here for several reasons:

1. "Unfiltered" TG data generally do not conform exactly to the required conditions of normality and univariance.

2. The high discrimination of their test may be a liability if the quantities being tested are only approximately normally distributed or do not all have the same standard deviation.

3. We wished to be able to make use of even approximate estimates of the standard deviations of the experimental quantities and derived quantities. This requires scaling of DS or DE values (see later for their definitions) to their expected values as calculated from order statistics.

4. The computer program must be kept as small as possible so it will fit in the laboratory computer.

5. Their procedure and the procedure in TGDEPG have given identical results on the few TG data sets subjected to both tests. However, because of their avoidance of the masking effect, their test is expected to be superior, especially when the data are exactly normal and univariate. For example, when applied at the normal operating level ($\sim$ 2 percent), the test in TGDEPG does not detect outliers in the eight results of uranium isotope analysis quoted in reference [15]; however, the methods in references [12-15 and 17-18] also fail to discriminate among the data.

Values of the activation energy E_i obtained as the reaction proceeds may reveal the existence of an induction period, vary randomly about the population mean in the

40 to 70 percent region of degradation, or, if a monatomic trend with extent of reaction or time is present, be somewhat smoothly distributed about the population mean, and change as the remains of the sample react. Also, in a passive data collection (where the data have no influence on the control of the experiment), averaging the E_i values is sometimes difficult because the $\hat{\sigma}_i$ values may vary widely. This is usually because of a fall off in the magnitude of the rates as the sample size decreases but may also arise because of lower viscosity as the sample degrades. The contribution of each E_i value to the average, $\bar{E}_w$, must therefore be weighted to reflect differing uncertainties and to maximize the reliability of the value obtained for $\bar{E}_w$. Further the standard deviation, $\hat{\sigma}_{\bar{E}}$, of the weighted average value, $\bar{E}_w$, must be estimated to reflect the spread of the individual E_i values about $\bar{E}_w$. First, however, an attempt should be made to identify unacceptable values of E_i and the associated estimated standard deviation, $\hat{\sigma}_{Ei}$.

In devising our procedure we have had to make several simplifying assumptions which we point out as the need arises. The procedure should be considered to be heuristic rather than rigorously derived.

We assume that in TG each experimental datum (sample weight, temperature) is a random (i.e., independent) selection from a population of potential experimental data which are distributed according to a normal type of distribution. We also assume that the same is true for results such as E_i which are derived from the experimental data even though the experimental data have been subjected to several transformations to provide values of E_i. Therefore, we proceed as though the majority of our experimentally derived results, E_i, belong to the same population and assume that differences between individual E_i values and the weighted mean, $\bar{E}_w$, follow the normal distribution law. The appropriateness of this assumption is indicated by the usual linearity of the probability plots which are produced by the TGDEPG program and to some extent by the small magnitude of $\hat{\sigma}_{\bar{E}}$, the standard deviation assigned to $\bar{E}_w$.

One possibility is to assume that all E_i values have the same standard deviation (i.e., that they are drawn from the same sample population) and are normally distributed. Then the appropriate statistics to define the collection of E_i values are the arithmetic average, $\bar{E}$, and the "unbiased" estimate of its standard deviation

$$\sigma(\bar{E}) = \{[\Sigma(\bar{E}-E_i)^2]/[n(n-1)]\}^{1/2},$$

where there are n values of E_i. The 95 percent confidence limits on the range of $\bar{E}$ are estimated from $\sigma(\bar{E})$ and the appropriate value from the t distribution with n-2 degrees of freedom.

The E_i values themselves also have estimated standard deviations, obtained in the case of our TG data by propagation of errors estimated from the fit of a polynomial to the raw data. When the individual standard deviations $\hat{\sigma}_{Ei}$ on the data E_i are different, each E_i is said to come from a different parent distribution (with different $\hat{\sigma}_{Ei}$ but the same mean, μ). In this case, we minimize

$$\chi^2 = \sum_i \left[\frac{E_i - \hat{\mu}}{\sigma_{Ei}}\right]^2$$

by varying $\hat{\mu}$. That is,

$$\frac{\partial\chi^2}{\partial\hat{\mu}} = 0 = \sum_i \frac{E_i - \hat{\mu}}{\sigma_{Ei}^2} ,$$

and our estimate of the average (weighted in this case) is given by

$$\mu = \sum_i \frac{E_i}{\sigma_{Ei}^2} \Big/ \sum_i \frac{1}{\sigma_{Ei}^2} = \bar{E}_w$$

Thus, the weighting factors, w_i, for combining the experimental data E_i are $w_i = \frac{1}{\sigma_i^2}$. From the propagation of error formula, the variance, $\sigma_{\bar{E}}^2$, of this weighted average is given by

$$\sigma_{\bar{E}}^2 = 1/\Sigma(1/\sigma_{Ei})^2 ,$$

so that the error estimated in the weighted average depends on the scale of the individual standard deviations. We stress that the equations for $\bar{E}_w$ and $\sigma_{\bar{E}}^2$ are best estimates *only* when the E_i values are normally distributed and when the σ_{Ei} values are well known. The error propagation path in the TG factor-jump procedure is tortuous (through extrapolation and derivatives) and the initial assessments of error in the primary data of sample weight and sample temperature may be crude. Also the sample behavior may change during the experiment. Because the absolute values of the derived standard deviations, $\hat{\sigma}_{Ei}$, may not adequately represent the observed variation in the E_i values we will need to estimate the scale, k, of these derived standard deviations i.e., $\sigma_{Ei} = k\hat{\sigma}_{Ei}$. An estimate of the absolute scale of the $\hat{\sigma}_{Ei}$ values can be obtained by relating the experimental distribution to the normal distribution. This also allows us to examine the data for atypical data points.

After calculation of $\hat{\sigma}_{Ei}$ (in the TGRUNF program) we can form the statistic

$$DE_i = \frac{E_i - \bar{E}_w}{k\hat{\sigma}_{Ei}} \quad . \qquad (1)$$

where k, the scale of $\hat{\sigma}_{Ei}$, is initially set equal to unity but is then refined. DE is essentially the quantity T in references [12-16]. If all values of σ_{Ei} were known exactly and all values of E_i came from populations with the same mean, $\bar{E}_w$, then the DE_i statistic would follow a normal distribution (N). In practice we have to estimate σ_{Ei}, and our estimate, $k\hat{\sigma}_{Ei}$, would, if replicated, vary according to a χ distribution. DE_i is then seen to be distributed according to N/χ, which is a t-distribution. For a reasonably large number of points (say > 10 or 15), the t distribution approximates a normal distribution. For various reasons, we therefore assume that the DE_i statistic is normally distributed and thus that the relative probability of observing the particular value E_i with standard deviation $k\sigma_{Ei}$ will be given by

$$P_i = \frac{1}{\sigma_{Ei} \quad 2\pi} \exp \quad - \frac{1}{2} \quad \frac{E_i - \bar{E}_w}{k\hat{\sigma}_{Ei}} \quad ^2 \quad . \qquad (2)$$

The probability of observing a value of E_i less than $\bar{E}_w + z\sigma_{Ei}$ is given by $\int_{-\infty}^{z} P_i \, dz$. This is illustrated for the second extremum by the shaded area in figure 1. The

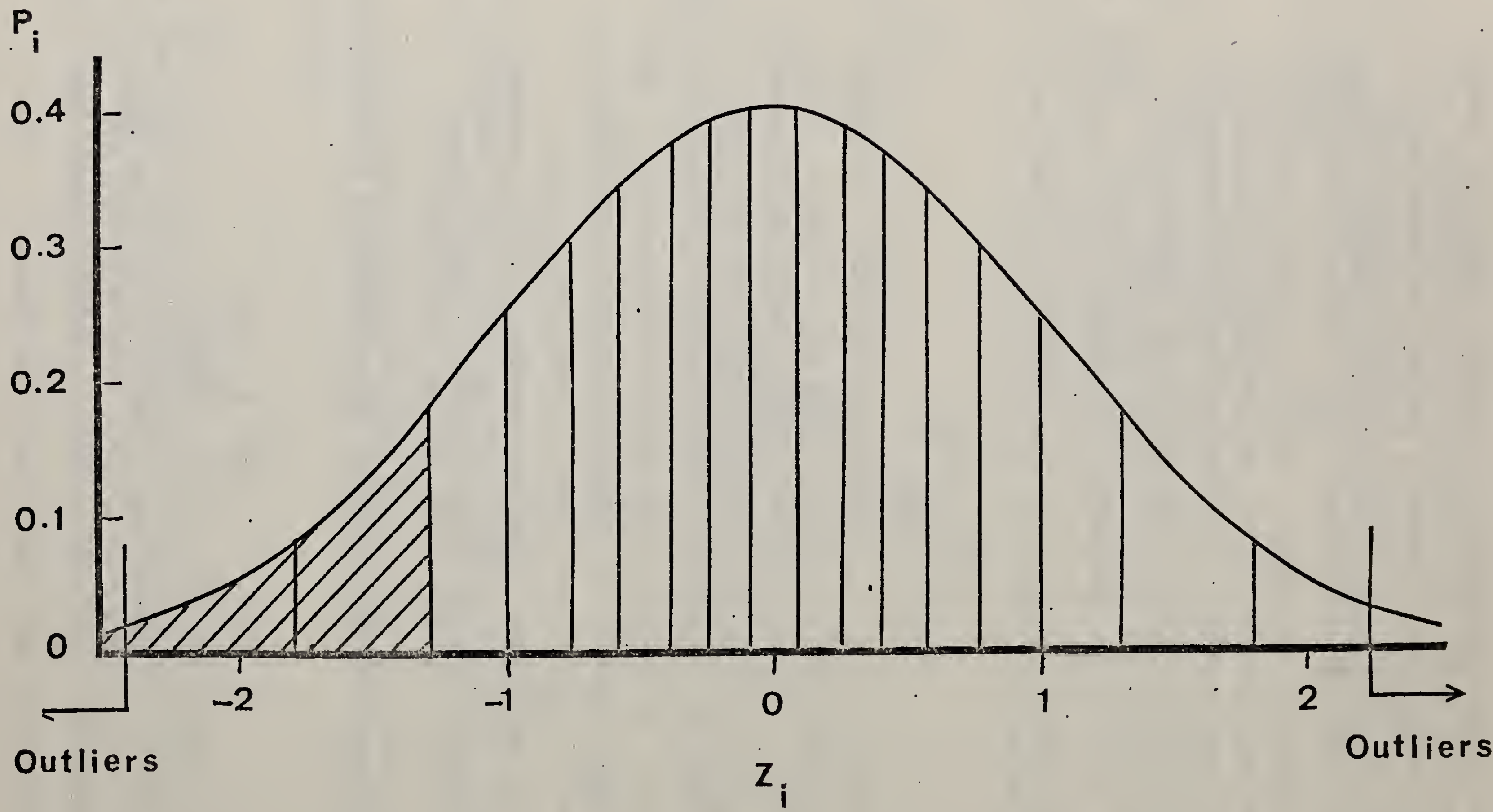


Figure 1

expected value z_i for each DE_i quantity may be obtained using order statistics (see reference [19]) if the DE_i quantities are first arranged in increasing magnitude. For n observations, we would expect, on the average, that the area under the probability curve in figure 1 would be divided into n + 1 equal areas. Therefore, the actual DE_i quantities should be related to the corresponding values, z_i, of the abscissa which divide the area under this curve into n + 1 equal parts

$$\text{i.e.,} \quad \int_{-\infty}^{z} P\,dz \sim i/(n+1) \quad .$$

In practice, we estimate the z_i values from the relationships [20]

$$p = \int_{-\infty}^{z_i} P_z dz = (i - \pi/8)/(n + 1 - \pi/4) \text{ for } i \geq 2 \text{ or } n \geq 10$$

$$\text{or } (3i - 1)/(3n + 1) \text{ for } i = 1 \text{ and/or } n < 10 \quad ,$$

and

$$z_i = 4.91(p^{0.14} - (1-p)^{0.14}) \quad .$$

If all E_i values come from parent populations with the same mean, ideally there should be a linear relationship between DE_i and z_i, with the line passing through the origin. A plot of DE_i vs z_i is known as a probability plot. Here we use the normal (i.e, Gaussian) probability plot (see references [21 and 22]). Values of DE_i belonging to a population with a mean different from that of the main body of data will probably be found as extreme values in the set of DE values. Hence, less weight should be given to the extreme values in determining the coefficients of the best linear relationship between DE_i and z_i. This could be achieved by considering only some inner set of DE_i values. A further consideration is that the scale of DE_i and z_i is non-linear, actually in some inverse relationship to the height of the probability function at point z_i, given that equal areas given approximately by $P_i(z_i - z_{i-1})$ should be delineated. The scale determination should therefore be weighted to make all contributions univariate. Exact weights would be $1/\sigma^2$ where σ^2 is the variance of z_i. Tables of σ^2_{zi} are available, but we have approximated these variances in practice by weighting the DE_i values according to the associated values of P_i (eq. 2). This incorporates the facts that the scale of DE_i is non-linear and values of DE_i nearer zero are expected to vary over a smaller range than values lying towards the extremes. The slope of the DE_i *versus* z_i relationship is the estimate of k, the scale of $\hat{\sigma}_{Ei}$, required to fit the DE_i values to a normal distribution. A plot of DE_i against z_i is produced by the computer program and has the advantage that the scaling and outlier rejection procedures are shown in a visual as well as an analytical fashion. A break in the plot is typically found between two different populations of DE_i values, although random fluctuations can sometimes produce surprisingly non-linear plots (see reference [23]).

The least probable values of E_i are necessarily associated with the extreme values of DE_i. These need not necessarily be associated with the largest or smallest values of E_i, however, because the size of σ_{Ei} has also been incorporated in DE_i. We must now decide whether to accept or reject the extreme values. From figure 1, we see that, on the average, the extremum DE_1 should bisect the area to the left of DE_2. If we set the cut off limit for DE_1 such that we will accept a DE_1 value that falls within the 90 percent of the area closest to DE_2, we will on average reject only 10 percent of the time an extremum which appears to be an outlier purely because of random fluctuations. Note this is *not* 10 percent of the data. For n values, it corresponds to approximately $[1/(n+1)]/10$ = 1 percent at *each* extreme for 20 data values; this constitutes a two-tail test at the 2 percent level. We expect to be able to reject most outliers which include additional non-random bias which makes their E_i values quite different from $\bar{E}_w$, and which, because of a small value of σ_{Ei}, would otherwise effect $\bar{E}_w$ significantly. Because the areas in figure 1 depend on the sample size, the cut off limit also depends on the sample size (cut off limits are applied to both negative and positive DE values). Thus, after determining the approximate scale, k, for $\hat{\sigma}_{Ei}$ in eq. 1 from a least squares fit of z_i to DE_i, we scale DE_i to z_i by applying the scale factor k to $\hat{\sigma}_{Ei}$. We determine the cut off limit for the DE_i values using the sample size and a prechosen chance of rejecting an acceptable extreme point along with atypical points, and we reject any extreme values of DE_i beyond the cut off limit. We then re-average the remaining E_i values to give a new value for $\bar{E}_w$, recalculate DE_i and rescale the remaining $\hat{\sigma}_{Ei}$ values by refitting DE_i to z_i. In this way we can reiterate the procedure (decreasing the cut off limit appropriately for each iteration) until no further outliers are discarded.

From the final set of scaled $\hat{\sigma}_{Ei}$ values we can calculate the standard deviation of the weighted average activation energy, $\bar{E}_w$. Here we re-emphasize that although we may have estimated the error in the rate and temperature measurements individually, (and perhaps, with other experimental setups, crudely from chart records, experience, or intuition) when we use the above procedure to estimate the σ_{Ei} values by rescaling $\hat{\sigma}_{Ei}$, we in fact assume only the *proportions* of the contributions of temperature and rate to the standard deviations of the individual activation energies E_i. Strictly speaking, we have also assumed that all deviations from the average arise from random error. In some cases, we have found a systematic trend of E_i with time or wide scatter among the E_i values. To the extent that these are assigned as random error, the associated $\hat{\sigma}_{Ei}$ values and the resulting estimate of $\sigma_{\bar{E}}$ are proportionately increased. This essentially states the extent to which the average value, $\bar{E}_w$, is meaningful when a trend or wide scatter is evident. Typically, one uses χ^2 or preferably the reduced value, $\chi^2_v = \chi^2/(n - v)$, as an estimate of the goodness-of-fit of a function (such as an average) to data. In the definition of χ^2_v, v quantities are determined from n data, and n - v is the number of degrees of freedom. Here we have rescaled the denominator, $\Sigma\sigma^2_{Ei} = k\Sigma(\hat{\sigma}_{Ei})^2$, of χ^2 so that χ^2_v should be approximately unity. When χ^2_v is not in the range 0.90 to 1.10, something may be wrong with the data set or the selected cut off limit. For example, the unbiased data may constitute a minor population in a very poor data set. The distribution will then be dominated by "bad" data. Thus, the "good" data may be outliers, partly because of their smaller $\hat{\sigma}_{Ei}$ values. In such cases the procedure will probably give poor results, but this should be neither surprising nor mitigating.

Extension of the Procedure

We have described how the DE_i population can be checked for atypical values. We now extend this concept to examining whether the σ_{Ei} values themselves constitute a homogeneous sample. This is especially important if one is examining data gathered and processed entirely by machine without operator intervention or assessment.

We could assume that the σ^2_{Ei} values follow a χ^2 distribution [24], but much of the reasoning (and computer programming) for the DE_i case can be reused if instead we examine the distribution of the residual

$$DS_i = \frac{\ln \hat{\sigma}_{Ei} - \overline{\ln \hat{\sigma}_{Ei}}}{k}$$

which we assume to be approximately normally distributed [25]. The scale factor, k, is initially set equal to 1. This actually assumes that the values of $\ln \sigma_{Ei}$ are randomly distributed, which is usually not the case. Nevertheless, the practice is defended on the grounds that it appears to be a satisfactory test for sample homogeneity. Box's test [26] gave the same results at the one percent level as this procedure.

In practice, the procedure consists of first checking the $\ln \hat{\sigma}_{Ei}$ distribution ($\sigma_{Ei} = k\hat{\sigma}_{Ei}$) for outliers without reference to the associated values of E_i. This finds points which have anomalously low values of $\ln \hat{\sigma}_{Ei}$ (these points would then dominate further calculations) and weeds out points with anomalously high values of $\ln \hat{\sigma}_{Ei}$ which have no effect on the final result yet would nominally contribute to the number of degrees of freedom. Explanations should be sought for either occurrence, but discarding points because of large $\ln \hat{\sigma}_{Ei}$ values is more common. Rejections tell the experimenter either that the change in conditions producing the data point was not large enough to be well known (for large $\ln \hat{\sigma}_{Ei}$) or that the rest of the data are not of the same high quality as the one with the anomalously low value of $\ln \hat{\sigma}_{Ei}$.

Precautionary Notes

This procedure is an attempt to estimate a reasonable value for the weighted average of the activation energy and its associated standard deviation using both individual estimates of the activation energy and their assigned standard deviations. It assumes that E_i and $\ln \sigma_{Ei}$ are both normally distributed and that a weighted fit between z_i and DE_i is appropriate in scaling $\hat{\sigma}_{Ei}$ values to obtain estimates of σ_{Ei}. Although the power of the method has not been tested on statistically assembled artificial data sets, the procedure appears to produce reasonable results when operating on TG data, provided that the "good" data dominate in the definition of the distribution. It evaluates the data in a way complementary to that of the trimmed mean program, which examines the means of E_i values without considering the associated $\hat{\sigma}_{Ei}$ values. Nevertheless, neither procedure should be used blindly.
The conditions generating data rejected as outliers should be examined for significance regarding the performance of the apparatus or changes in the sample behavior.
Further, it should be realized that the procedure here is to some extent intuitive, and its use is still experimental.

Because the process rests equally on the individual activation energies *and* their assigned standard deviation, the experiment should be designed to produce reasonable estimates of both these quantities as well as to keep the standard deviations small.

Finally, replication is advisable to allow use of analysis of variance to check that the measure of error between samples of the same material is consistent with the measure of error within the samples. If the error between samples is the larger, thermal history is probably playing an important role.

VI. Generalized Polynomial Fitting: POLGEP

POLGEP is a FORTRAN program which allows the user to read data in almost any format, to pick out from the "repeat unit" of data those items to which the polynomial will be fitted, to specify the degree and coefficients of the polynomial, and to choose whether the fit is to be weighted by $1/\sigma_i^2(y)$ or not. The general scheme of data input is essentially that of PABS (Section IX). The user can evaluate the fitted polynomial and its first derivative at any point. Estimates of the error are provided. The program is intended for such uses as extrapolating activation energies to zero sample weight in those cases where there is a trend with sample weight. A listing of the program and an example of its use are given in Appendix E.

VII. Publication Tables: TABLEP

The FORTRAN program TABLEP is provided to make tables of activation energies and associated data automatically. This saves time and avoids transcription errors. The listing and an example of its use are given in Appendix F.

VIII. General Program to Operate the MIDAS Interface: BMDP

BMDP is a BASIC program which allows the use of the MIDAS interface for quick checks of the apparatus, non-routine applications and so on. The results are written on file 7 in such a way that other BASIC programs will be able to read them. The program reads voltages on specified inputs on the analog scanner for a user-provided number of times, with a user-specified wait between each set of readings. The program is quite simple and is easily altered. The listing of the program and an example of its use are given in Appendix G.

IX. Generalized Plotting Program for Versatec Printer-Plotter: PABS

During the check-out of the apparatus and examination of data and of derived results, it became imperative to write a plotting program (incorporating software available for the VERSATEC printer-plotter). Because of the widely differing formats of numerical quantities to be plotted, we decided to make the program fairly general. The program PABS is the result. The listings are given in Appendix H.

PABS will read files in most formats. The user inputs the details in response to prompting questions. The first line is assumed to be a title. Any numbers of subsequent lines may be ignored. PABS reads in a repeating unit of up to 100 numbers, selects one of them to be x and selects up to 5 others to be y values at that value of x. This process is repeated until the end of the input file is reached or until a user-chosen number of points has been read in. Thus up to 5 curves are built up and plotted. Input is in free format, regardless of the number of lines required to contain the repeating unit. The user then specifies whether symbols should be plotted at the data points or not at all and whether blanks, a straight line or line interpolated using a user-chosen number of points is to be used to join adjacent points. The size of the plot is chosen, user-supplied scaling factors are applied to be the input data, and the plots are scaled automatically or by a user-provided scale factor.

The output of PABS must be processed by the program VCPY to produce a plot. An example of the procedure "automated" to plot several files is given in Appendix H.

References

[1] B. Dickens, W. J. Pummer and J. H. Flynn, "Automation of a thermogravimetry apparatus with a laboratory computer", in Analytical Pyrolysis, C. E. R. Jones and C. A. Cramers, Eds, Elsevier, Amsterdam (Holland), 1977.

[2] B. Dickens "Automation of factor-jump thermogravimetry for active computer control", Thermochim. Acta, 29, 41 (1979).

[3] B. Dickens, "Experiences in developing an automated factor-jump method of thermogravimetry": Thermochim. Acta, 29, 87 (1979).

[4] B. Dickens and J. H. Flynn, "Optimisation of thermogravimetric experiments under active computer control", Proc. First Eur. Symp. on Thermal Anal., Salford, 1976.

[5] B. Dickens, "Computer programs to implement factor-jump thermogravimetry," Thermochim. Acta, 29, 57 (1979).

[6] P. G. Stein, "Laboratory Automation Software for NBS 16-Bit Computers," Feb. 1977.

[7] J. H. Flynn and B. Dickens, "Steady-state parameter-jump methods and relaxation methods in thermogravimetry", Thermochim. Acta, 15, 1 (1976).

[8] J. W. Tukey, "The future of data analysis", Ann. Math. Statistics, 33 1 (1962).

[9] J. W. Tukey and D. H. McLaughlin, "Less vulnerable confidence and significance procedures for location based on a single sample: Trimming/winsorization 1", Sankhya, A25, 331 (1963).

[10] E. L. Crow and M. M. Siddiqui, "Robust Estimation of Location", J. Amer. Statistics Assoc., 62, 353 (1967).

[11] B. Dickens and J. H. Flynn, 5th North American Thermal Analysis Society Meeting, Peterborough, Ontario, Canada, 1975.

[12] F. E. Grubbs, Ann. Math. Stat., 21 (1950) 27.

[13] F. Proschan, Am. J. Phys., 21 (1953) 349.

[14] W. J. Dixon, Biometrics, 22 (1953) 74.

[15] F. E. Grubbs, Technometrics, 11 (1969) 1.

[16] A. L. Tietjen and R. H. Moore, Technometrics, 14 (1972) 583.

[17] F. E. Grubbs and G. Beck, Technometrics, 14 (1972) 847.

[18] G. L. Tietjen, R. H. Moore and R. J. Beckman, Technometrics, 15 (1973) 717.

[19] J. Mandel, The Statistical Analysis of Experimental Data, Wiley-Interscience, New York, (1964) 85.

[20] J. J. Filliben quoted in D. Hogben, S. T. Peavy, and R. N. Varner, OMNITAB II, User's Reference Manual, National Bureau of Standards Technical Note 552, (1971) 146.

[21] M. B. Wilk and R. Gnanadesikan, Biometrika, 55 (1968) 1.

[22] S. C. Abrahams and E. T. Keve, Acta Cryst. A27 (1971) 157.

[23] C. Daniel and F. S. Wood, Fitting Equations to Data, Wiley-Interscience, New York (1971).

[24] P. R. Bevington, Data Reduction and Error Analysis for the Physical Sciences, McGraw-Hill, New York, (1969) 187.

[25] C. Eisenhart, in Selected Techniques of Statistical Analysis, C. Eisenhart, M. W. Hastay and W. A. Wallis, Eds. McGraw-Hill, New York, p. 412, equation 31 (1974).

[26] G. P. E. Box, Biometrika, 36 (1949) 317.

Notes on the Program Appendices

Much use is made of 'transfer' files (see the FLOXOS manual). This is a convenient way of instituting automatic operation, both in assembling the programs from their component parts and in running the programs in production use. The files that a given program needs are evident from the SETXX files, which make routine file assignments, and from the examples of input, which show the assignment of particular input files. To run a program, use DO SETXX for that program, assign any needed input and output files and then type RUN YYY,DZ where YYY is the program name and the program is on disc DZ.

To change a program, edit the source file with the EDIT system program, and then form a binary file, usually with almost the same name as the source file but with a final B (for binary) in the name. Use the sequence

```
*AC  SOURCE,1D0
*AC  CALOUT,2D1 (this is a general scratch file)
*AS  300        (generally no output needed)
*RUN FORTV,D0
```

and then, after compilation (translation from FORTRAN into the CAL language)

```
*AC  CALOUT,1D1
*AC  BINARY,2D0
*RUN CAL
```

and finally, after translation from CAL into binary,

```
*WF  2
*CL
```

Errors in the compilation show up on unit 7, which is usually the terminal. The line numbers mentioned are those found in the EDIT program. Use of

```
*AS 362
*AS 762
```

for the compilation will print a listing with interspersed error messages. Always use

```
*AS  300
*AS  720
```

when about to run CAL. The listing is of no interest if you have started from FORTRAN.

The general order in the program Appendices is

SETXX
Example of input
Example of output
Listing of routines in program
Program remaking routine L0XXXX
File XXXXL5, associated with L0XXXX
Memory map of the program

Appendix A. TGRUNF, a FORTRAN to perform factor-jump thermogravimetry.

File SETTG, used as DO SETTG,D1 to make file assignments needed in a run of TGRUNF.

```
AC NFILED,1D1
AS 262
AC PRINT,3D1
AS 420
AS 520
AS 600
AS 700
AS 820
AC NFILEC,9D1
AS A20
AC TGSETU,BD0
AC TGFACT,CD0
AC TGPLAT,DD0
AC TGPREC,ED0
AC TGSTAT,FD0
LU
ASSIGN SAVE FILE FOR E'S (6) AND PARAMETERS-(9) THEN TYPE 'RUN TGMAIN,D1'
TR
```

User then types:

```
*AL   PMMA01,6D1,1,80
*AC   PMSAVE,9D1
*RUN  TGMAIN,D1
```

Example of the 'save' file written on unit 6 by TGRUNF. The file contains a two line title and then, for each activation energy, ordinal number, E, $\hat{\sigma}(E)$, time of determination, sample weight, percentage of reaction, temperature T for plateau 1, $\hat{\sigma}(T_1)$, rate 1, $\hat{\sigma}(r_1)$, extrapolated sample weight from plateau 1, $\hat{\sigma}(w_1)$, and number of data points taken during this plateau. The corresponding quantities are then given for plateau 2.

```
 NBS 706 IN 40% O2 P=800 MM HG 3/20/78

   1 22.422  0.177      938.   0.573756E+06   5.82
                     292.89  0.03    -0.3311E+02     0.4547E-01
                                    0.573473E+06   0.101013E+02    30
                     284.09  0.03    -0.2417E+02     0.3538E-01
                                    0.574039E+06   0.785965E+01    30
   2 23.191  0.139     1765.   0.554887E+06   8.19
                     284.14  0.03    -0.2122E+02     0.3538E-01
                                    0.555269E+06   0.785965E+01    30
                     295.03  0.03    -0.3171E+02     0.3244E-01
                                    0.554506E+06   0.720734E+01    30
   3 23.392  0.191     2591.   0.529632E+06  11.37
                     294.96  0.03    -0.2912E+02     0.3244E-01
                                    0.529354E+06   0.720734E+01    30
                     286.09  0.02    -0.2096E+02     0.4278E-01
                                    0.529912E+06   0.950372E+01    30
   4 23.157  0.175     3419.   0.512748E+06  13.49
                     286.19  0.02    -0.1978E+02     0.4278E-01
                                    0.513065E+06   0.950372E+01    30
                     297.06  0.03    -0.2944E+02     0.4581E-01
                                    0.512431E+06   0.101767E+02    30
   5 22.867  0.220     4246.   0.488936E+06  16.48
                     296.97  0.03    -0.2782E+02     0.4586E-01
                                    0.488741E+06   0.101987E+02    30
                     288.08  0.03    -0.2020E+02     0.4374E-01
                                    0.489231E+06   0.972691E+01    30
   6 22.935  0.167     5074.   0.472496E+06  18.55
                     288.18  0.03    -0.1946E+02     0.4374E-01
                                    0.472813E+06   0.972691E+01    30
                     299.13  0.02    -0.2883E+02     0.3539E-01
                                    0.472179E+06   0.787063E+01    30
   7 23.031  0.198     5901.   0.448867E+06  21.53
                     299.02  0.02    -0.2815E+02     0.3535E-01
                                    0.448602E+06   0.785362E+01    30
                     290.12  0.03    -0.2045E+02     0.4108E-01
                                    0.449133E+06   0.912550E+01    30
   8 23.086  0.172     6729.   0.432094E+06  23.64
                     290.15  0.03    -0.2001E+02     0.4108E-01
                                    0.432494E+06   0.912550E+01    30
                     301.13  0.03    -0.2969E+02     0.3885E-01
                                    0.431784E+06   0.863049E+01    30
```

9	23.764	0.231	7555.	0.407669E+06	26.71		
			303.99	0.03	-0.2931E+02	0.3985E-01	
					0.407390E+06	0.853049E+01	30
			291.91	0.03	-0.2097E+02	0.5051E-01	
					0.407948E+06	0.112212E+02	30
10	23.552	0.200	8382.	0.390441E+06	28.88		
			292.08	0.03	-0.2052E+02	0.5051E-01	
					0.390790E+06	0.112212E+02	30
			303.14	0.04	-0.3070E+02	0.4803E-01	
					0.390091E+06	0.106695E+02	30
10	23.507	0.224	9210.	0.365137E+06	32.06		
			303.01	0.04	-0.3033E+02	0.4808E-01	
					0.364843E+06	0.106926E+02	30
			293.90	0.04	-0.2181E+02	0.4185E-01	
					0.365431E+06	0.930671E+01	30
10	23.831	0.191	10037.	0.347203E+06	34.31		
			294.05	0.04	-0.2149E+02	0.4180E-01	
					0.347554E+06	0.928650E+01	30
			305.12	0.03	-0.3208E+02	0.6393E-01	
					0.346852E+06	0.142037E+02	30
10	24.055	0.246	10864.	0.320866E+06	37.63		
			305.03	0.03	-0.3151E+02	0.6393E-01	
					0.320559E+06	0.142037E+02	30
			295.95	0.03	-0.2256E+02	0.5145E-01	
					0.321172E+06	0.114309E+02	30
10	22.997	0.203	11691.	0.302300E+06	39.96		
			296.04	0.03	-0.2221E+02	0.5145E-01	
					0.302660E+06	0.114309E+02	30
			307.13	0.03	-0.3276E+02	0.6552E-01	
					0.301940E+06	0.147778E+02	30
10	24.370	0.217	12519.	0.275378E+06	43.35		
			306.98	0.03	-0.3232E+02	0.6652E-01	
					0.275030E+06	0.147778E+02	30
			297.92	0.03	-0.2312E+02	0.3731E-01	
					0.275726E+06	0.822791E+01	30
10	24.122	0.143	13346.	0.256475E+06	45.73		
			298.07	0.03	-0.2253E+02	0.3731E-01	
					0.256853E+06	0.828791E+01	30
			309.12	0.02	-0.3371E+02	0.3979E-01	
					0.256097E+06	0.893868E+01	30
10	23.990	0.166	14173.	0.229012E+06	49.18		
			309.02	0.02	-0.3257E+02	0.3979E-01	
					0.228687E+06	0.883668E+01	30
			300.00	0.03	-0.2350E+02	0.3254E-01	
					0.229337E+06	0.722918E+01	30
10	24.200	0.140	14999.	0.209822E+06	51.59		
			300.04	0.03	-0.2279E+02	0.3254E-01	
					0.210195E+06	0.722918E+01	30
			311.11	0.03	-0.3408E+02	0.4099E-01	
					0.209448E+06	0.310525E+01	30
10	23.929	0.177	15826.	0.182190E+06	55.07		
			310.97	0.03	-0.3260E+02	0.4099E-01	
					0.181877E+06	0.918525E+01	30
			301.99	0.03	-0.2363E+02	0.3449E-01	
					0.182503E+06	0.766166E+01	30

How to prepare a new flexible disc to be a TGS save file disc.

```
Put an old disc in drive F0
Put a new disc in drive F1
*ZU
*AC NFILEC,1F0
*AL NFILEC,2F1,1,80
*CP 1,2,80,A
 (You will get an end of medium message--90F0--ignore it.)
*AL NFILED,2F1,3,80
*ZU
The new disc is now ready.
```

Beginning of a typical output of the TGRUNF program. Examples of the user-input are given in Tables 3 to 16 of the main text. Users of the TGRUNF program will find much useful information in references 2, 3, and 5.

```
NBS 1475 PE IN VAC 4/4/78 3PM


TYPE E THERMOCOUPLE=NI-CR/CU-NI,EXPRESSIONS GOOD
FROM -0.05 TO +0.04, 0 TO 400 DEGREES C.(TABLE A5.2.3,P.307
NBS MONOGRAPH 125,1974)

PARAMETERS INITIALISED AS FOLLOWS:
NTHCP =            E    TSCALE =  243.73   SKIP=   0.
FACTORS =      1       2       0       0       0       0       0       0
FACFUN =       0       0       0       0       0       0       0       0
DERIVS =       1       0       0       0       0       0       0       0
INPUTS =       0       1       2       3       4       5       6       7
SCALES =   1.000   1.000   0.010   1.000   1.000   1.000   1.000   1.000
NRANGE =       3       3       5       5       5       5       5       5
NREADS =       3       3       3       3       3       3       3       3
NTERMS =       2       1       0       0       0       0       0       0
CHI TEST=  4000.   1500.   1000.   1000.   1000.   1000.   1000.   1000.
PTIMES =   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
PRECF =   100.00  100.00  100.00  100.00  100.00  100.00  100.00  100.00
PRECR =    30.00   30.00   30.00   30.00   30.00   30.00   30.00   30.00
NFACTS =   2 NRATES =    1 TIMLIM  = 600. BAD PTS = 15 EQUIL = 200.
CHECKS  =  75 END SIG =  0.0 RATMIN =   15.0 RATMAX =  45.0 TJUMP = 14.0
TC TO D =   0.17023E-01  -0.22097E-06   0.54809E-11  -0.57670E-16
IO UNITS=      5       5       3       7       8       1       6
POLNOM EXP=    1       2       3       4       5
PRECISION IN E ACT =  25.00
FACTOR LEVELS AND CHANGES FOLLOW:
  1  425.000     1.000
  2    0.000     0.000
  3    0.000     0.000
  4    0.000     0.000
DESIGN MATRIX FOLLOWS
 1     8.000       0.000       0.000       0.000
 2    -6.000       0.000       0.000       0.000
STEADINESS CHECK = 0 WHERE 0=NO,1=YES
INPUT FILTER TO DVM = 1 WHERE 0=NO.1=YES
READ OTHER EXPTS = 0 WHERE 0=NO,1=YES

BIAS VOLTAGES AND SIGMAS FOR FACTOR INPUTS
FACTOR       VOLTAGE          SIGMA
    1  0.0000000E+00  0.9999999E-09
    2  0.0000000E+00  0.9999999E-09

INITIAL AND FINAL WEIGHTS ARE:          187200.        -175000.

BAKE TIME IN SECONDS IS  1200.
BAKE TEMPERATURE IN DEGREES CELCIUS IS   420.
BAKE PRESSURE IN MM HG IS     0.
FLOW OF N2 IN SCC/MIN IS     0.
FLOW OF O2 IN SCC/MIN IS     0.

FACSET
    420.0        1
```

```
   14  #D 7445;
FACSET
     0.0          2
   14  #D 0000:
FACSET
     0.0          3
   14  #G 0000:
FACSET
     0.0          4
   14  #G 0000;
FACSET
   433.0          1
   14  #D 7698;
FACSET
     0.0          2
   14  #D 0000:
FACSET
     0.0          3
   14  #G 0000:
FACSET
     0.0          4
   14  #G 0000;
```

FACTOR LEVELS SET TO   433.000   0.000   0.000   0.000

****** MEASUREMENT FOR PLATEAU NUMBER 1 ******

NO.	F	I	READING	SIGMA	RD	TIME	BAD	RCHI	TESTCH
1	1	0	0.1498407E+06	0.2973396E+02	3	204.	0	0.0	4000.0
1	2	1	0.3131367E+05	0.8819175E+00	3	204.	0	0.0	1500.0
2	1	0	0.1494690E+06	0.2800589E+02	3	211.	0	0.0	4000.0
2	2	1	0.3131000E+05	0.5773508E+00	3	211.	0	0.0	1500.0
3	1	0	0.1490760E+06	0.3146951E+02	3	219.	0	0.0	4000.0
3	2	1	0.3131933E+05	0.3333346E+00	3	219.	0	0.0	1500.0
4	1	0	0.1486640E+06	0.3002217E+02	3	227.	0	0.0	4000.0
4	2	1	0.3132133E+05	0.3333346E+00	3	227.	0	0.0	1500.0
5	1	0	0.1482503E+06	0.3117863E+02	3	234.	0	0.0	4000.0
5	2	1	0.3131700E+05	0.5773508E+00	3	234.	0	0.0	1500.0
6	1	0	0.1478403E+06	0.3117063E+02	3	242.	0	0.0	4000.0
6	2	1	0.3131233E+05	0.3333346E+00	3	242.	0	0.0	1500.0
7	1	0	0.1474127E+06	0.3060132E+02	3	250.	0	0.0	4000.0
7	2	1	0.3131200E+05	0.0000000E+00	3	250.	0	0.0	1500.0
8	1	0	0.1470200E+06	0.2886748E+02	3	257.	0	9.0	4000.0
8	2	1	0.3131300E+05	0.0000000E+00	3	257.	0	0.0	1500.0
9	1	0	0.1466473E+06	0.2829208E+02	3	265.	0	0.0	4000.0
9	2	1	0.3131133E+05	0.3333346E+00	3	265.	0	0.0	1500.0
10	1	0	0.1462573E+06	0.2857928E+02	3	272.	0	0.0	4000.0
10	2	1	0.3130967E+05	0.3333346E+00	3	272.	0	0.0	1500.0
11	1	0	0.1458643E+06	0.2944670E+02	3	280.	0	0.0	4000.0
11	2	1	0.3130333E+05	0.3333346E+00	3	280.	0	0.0	1500.0
12	1	0	0.1454647E+06	0.2944670E+02	3	288.	0	0.0	4000.0
12	2	1	0.3131100E+05	0.0000000E+00	3	288.	0	0.0	1500.0
73	1	0	0.1211283E+06	0.2929208E+02	3	754.	0	0.0	4000.0
73	2	1	0.3127833E+05	0.8819175E+00	3	754.	0	0.0	1500.0
74	1	0	0.1207420E+06	0.2916043E+02	3	762.	0	0.0	4000.0
74	2	1	0.3127600E+05	0.0000000E+00	3	762.	0	0.0	1500.0
75	1	0	0.1203593E+06	0.2915662E+02	3	769.	0	0.0	4000.0
75	2	1	0.3128300E+05	0.5773508E+00	3	769.	0	0.0	1500.0

CHECKS OF PRECISION FOR PLATEAU 1

****

WEIGHT LOSS FOR THIS PLATEAU = 0.295E+05 MICROVOLTS
WITH AVERAGE RATE OF 0.522E+02 MICROVOLTS PER SECOND

TIME	P	YFIT	SIGMA P	P-YFIT	(P-YFIT)/SIGP	DEL P
-282.5	0.149841E+06	0.149861E+06	0.297E+02	-0.206E+02	-0.694E+00	0.000E+00
-275.5	0.149469E+06	0.149493E+06	0.280E+02	-0.236E+02	-0.841E+00	-0.372E+03
-267.5	0.149076E+06	0.149071E+06	0.315E+02	0.475E+01	0.151E+00	-0.393E+03
-259.5	0.148664E+06	0.148650E+06	0.300E+02	0.140E+02	0.466E+00	-0.412E+03
-252.5	0.148250E+06	0.148281E+06	0.312E+02	-0.313E+02	-0.100E+01	-0.414E+03
-244.5	0.147840E+06	0.147860E+06	0.312E+02	-0.202E+02	-0.647E+00	-0.410E+03
-236.5	0.147413E+06	0.147439E+06	0.306E+02	-0.268E+02	-0.874E+00	-0.425E+03
-229.5	0.147020E+06	0.147071E+06	0.299E+02	-0.511E+02	-0.177E+01	-0.393E+03
-221.5	0.146647E+06	0.146650E+06	0.283E+02	-0.305E+01	-0.108E+00	-0.373E+03
-214.5	0.146257E+06	0.146282E+06	0.286E+02	-0.249E+02	-0.870E+00	-0.390E+03
-206.5	0.145864E+06	0.145851E+06	0.294E+02	0.281E+01	0.955E-01	-0.393E+03
-198.5	0.145465E+06	0.145441E+06	0.294E+02	0.239E+02	0.809E+00	-0.400E+03
-191.5	0.145066E+06	0.145073E+06	0.306E+02	-0.669E+01	-0.218E+00	-0.398E+03
-183.5	0.144662E+06	0.144653E+06	0.294E+02	0.975E+01	0.331E+00	-0.404E+03
229.5	0.123050E+06	0.123063E+06	0.283E+02	-0.133E+02	-0.471E+00	-0.395E+03
236.5	0.122673E+06	0.122699E+06	0.277E+02	-0.261E+02	-0.940E+00	-0.377E+03
244.5	0.122296E+06	0.122283E+06	0.274E+02	0.129E+02	0.459E+00	-0.377E+03
252.5	0.121914E+06	0.121867E+06	0.303E+02	0.463E+02	0.153E+01	-0.382E+03
259.5	0.121510E+06	0.121504E+06	0.286E+02	0.613E+01	0.214E+00	-0.404E+03
267.5	0.121128E+06	0.121088E+06	0.283E+02	0.403E+02	0.142E+01	-0.381E+03
275.5	0.120742E+06	0.120672E+06	0.292E+02	0.696E+02	0.239E+01	-0.386E+03
282.5	0.120359E+06	0.120309E+06	0.292E+02	0.504E+02	0.173E+01	-0.383E+03

POLYNOMIAL COEFFICIENTS FOR FACTOR 1

COEFF	SIGMA
0.13503219E+06	0.49327717E+01
-0.52395359E+02	0.19859039E-01
0.66241832E-03	0.13429741E-03

REDUCED CHI SQUARE= 810.19
FTEST = 0.16E+07
RMUL = 0.10E+01
R(1) = -0.10E+01
R(2) = 0.15E-02

TIME	P	YFIT	SIGMA P	P-YFIT	(P-YFIT)/SIGP	DEL P
-282.5	0.313137E+05	0.313115E+05	0.882E+00	0.216E+01	0.244E+01	0.000E+00
-275.5	0.313100E+05	0.313113E+05	0.577E+00	-0.132E+01	-0.229E+01	-0.367E+01
-267.5	0.313193E+05	0.313111E+05	0.333E+00	0.822E+01	0.247E+02	0.933E+01
-259.5	0.313213E+05	0.313109E+05	0.333E+00	0.104E+02	0.313E+02	0.200E+01
-252.5	0.313170E+05	0.313107E+05	0.577E+00	0.630E+01	0.109E+02	-0.433E+01
-244.5	0.313123E+05	0.313105E+05	0.333E+00	0.185E+01	0.554E+01	-0.467E+01
259.5	0.312950E+05	0.312968E+05	0.115E+01	-0.182E+01	-0.157E+01	-0.113E+02
267.5	0.312783E+05	0.312966E+05	0.882E+00	-0.183E+02	-0.207E+02	-0.167E+02
275.5	0.312760E+05	0.312964E+05	0.000E+00	-0.204E+02	-0.204E+17	-0.233E+01
282.5	0.312830E+05	0.312962E+05	0.577E+00	-0.132E+02	-0.228E+02	0.700E+01

```
POLYNOMIAL COEFFICIENTS FOR FACTOR 2
        COEFF          SIGMA
 0.31303852E+05 0.93359089E+00
-0.27117468E-01 0.56411624E-02

 REDUCED CHI SQUARE=      65.37
 FTEST =  0.23E+02
 RMUL =  0.49E+00
 R(1) = -0.49E+00


 EXTRAPOLATION TIMES ABOUT MID POINT OF PLATEAU FOR POLYNOMIALS
 FIRST PLATEAU =       -200.      200.
 SECOND PLATEAU =       -482.      482.
 REAL TIMES ARE      200. SECONDS ON EITHER SIDE OF
       0.        0. FOR FIRST PLATEAU, AND       204.      769. FOR SECOND


 EXTRAPOLATED VALUES AND SIGMAS BACK AND FORWARD IN TIME
 THEN PRECISIONS, = BACK, FORWARD AND REQUESTED

 R1  -0.529E+02 0.131E+00  -0.517E+02 0.131E+00    404.     394.      30.
 F1   0.160E+06 0.331E+02   0.118E+06 0.331E+02   4651.    3325.     100.
 F2   0.313E+05 0.288E+01   0.313E+05 0.288E+01  10883.   10874.     100.
 FACSET
    425.0        1
   14  #D 7542;
 FACSET
      0.0        2
   14  #D 0000;
 FACSET
      0.0        3
   14  #G 0000;
 FACSET
      0.0        4
   14  #G 0000;


 FACTOR LEVELS SET TO   425.000     0.000     0.000     0.000


 ****** MEASUREMENT FOR PLATEAU NUMBER  2 ******

  NO. F I       READING          SIGMA RD        TIME BAD    RCHI  TESTCH


    1 1 0 0.1051397E+06 0.1502590E+02  3        1133.   0      0.0  4000.0
    1 2 1 0.3066500E+05 0.1154700E+01  3        1133.   0      0.0  1500.0
    2 1 0 0.1049323E+06 0.1507801E+02  3        1140.   0      0.0  4000.0
    2 2 1 0.3067833E+05 0.8819175E+00  3        1140.   0      0.0  1500.0
    3 1 0 0.1046993E+06 0.1760995E+02  3        1148.   0      0.0  4000.0
    3 2 1 0.3068167E+05 0.3333346E+00  3        1148.   0      0.0  1500.0
    4 1 0 0.1044570E+06 0.1616580E+02  3        1156.   0      0.0  4000.0
    4 2 1 0.3067667E+05 0.3333346E+00  3        1156.   0      0.0  1500.0


   73 1 0 0.8972800E+05 0.1558845E+02  3        1680.   0      0.0  4000.0
   73 2 1 0.3066267E+05 0.3333346E+00  3        1680.   0      0.0  1500.0
   74 1 0 0.8951031E+05 0.1587001E+02  3        1688.   0      0.0  4000.0
   74 2 1 0.3056300E+05 0.9000000E+00  3        1688.   0      0.0  1500.0
   75 1 0 0.8931669E+05 0.1501482E+02  3        1695.   0      0.0  4000.0
```

75 2 1 0.3066633E+05 0.3333346E+00 3 1695. 0 0.0 1500.0

CHECKS OF PRECISION FOR PLATEAU 2
****

WEIGHT LOSS FOR THIS PLATEAU = 0.158E+05 MICROVOLTS
WITH AVERAGE RATE OF 0.282E+02 MICROVOLTS PER SECOND

TIME	P	YFIT	SIGMA P	P-YFIT	(P-YFIT)/SIGP	DEL P
-281.0	0.105140E+06	0.105146E+06	0.150E+02	-0.631E+01	-0.453E+00	0.000E+00
-274.0	0.104932E+06	0.104941E+06	0.159E+02	-0.863E+01	-0.543E+00	-0.207E+03
-266.0	0.104699E+06	0.104706E+06	0.176E+02	-0.694E+01	-0.394E+00	-0.233E+03
-253.0	0.104467E+06	0.104472E+06	0.162E+02	-0.481E+01	-0.296E+00	-0.232E+03
-251.0	0.104252E+06	0.104267E+06	0.162E+02	-0.152E+02	-0.943E+00	-0.215E+03
-243.0	0.104024E+06	0.104033E+06	0.176E+02	-0.906E+01	-0.514E+00	-0.228E+03
-235.0	0.103802E+06	0.103799E+06	0.165E+02	0.281E+01	0.171E+00	-0.222E+03
-228.0	0.103574E+06	0.103595E+06	0.179E+02	-0.211E+02	-0.118E+01	-0.228E+03
-220.0	0.103348E+06	0.103362E+06	0.162E+02	-0.140E+02	-0.866E+00	-0.226E+03
-212.0	0.103122E+06	0.103130E+06	0.159E+02	-0.750E+01	-0.472E+00	-0.226E+03
-205.0	0.102914E+06	0.102926E+06	0.150E+02	-0.124E+02	-0.828E+00	-0.208E+03
-197.0	0.102714E+06	0.102694E+06	0.159E+02	0.201E+02	0.126E+01	-0.200E+03
-190.0	0.102497E+06	0.102491E+06	0.162E+02	0.606E+01	0.375E+00	-0.217E+03
-182.0	0.102278E+06	0.102259E+06	0.159E+02	0.187E+02	0.118E+01	-0.219E+03
-175.0	0.102071E+06	0.102057E+06	0.165E+02	0.138E+02	0.839E+00	-0.207E+03
-167.0	0.101841E+06	0.101826E+06	0.170E+02	0.155E+02	0.919E+00	-0.229E+03
-159.0	0.101617E+06	0.101595E+06	0.162E+02	0.221E+02	0.137E+01	-0.224E+03
-152.0	0.101391E+06	0.101393E+06	0.159E+02	-0.219E+01	-0.138E+00	-0.226E+03
-144.0	0.101168E+06	0.101163E+06	0.162E+02	0.538E+01	0.332E+00	-0.223E+03
-136.0	0.100957E+06	0.100933E+06	0.162E+02	0.241E+02	0.149E+01	-0.211E+03
-129.0	0.100734E+06	0.100732E+06	0.176E+02	0.181E+01	0.103E+00	-0.223E+03
-121.0	0.100527E+06	0.100502E+06	0.144E+02	0.243E+02	0.168E+01	-0.207E+03
-114.0	0.100309E+06	0.100302E+06	0.176E+02	0.731E+01	0.415E+00	-0.217E+03
-106.0	0.100080E+06	0.100073E+06	0.165E+02	0.700E+01	0.425E+00	-0.229E+03
-98.0	0.998580E+05	0.998444E+05	0.167E+02	0.136E+02	0.814E+00	-0.222E+03
-91.0	0.996393E+05	0.996446E+05	0.162E+02	-0.538E+01	-0.332E+00	-0.219E+03
-83.0	0.994196E+05	0.994165E+05	0.159E+02	0.313E+01	0.197E+00	-0.220E+03
-75.0	0.992130E+05	0.991887E+05	0.150E+02	0.243E+02	0.162E+01	-0.207E+03
-68.0	0.990077E+05	0.989897E+05	0.165E+02	0.180E+02	0.109E+01	-0.205E+03
-60.0	0.987993E+05	0.987624E+05	0.147E+02	0.369E+02	0.250E+01	-0.208E+03
-53.0	0.985847E+05	0.985638E+05	0.167E+02	0.209E+02	0.125E+01	-0.215E+03
-45.0	0.983670E+05	0.983371E+05	0.167E+02	0.299E+02	0.178E+01	-0.218E+03
-38.0	0.981390E+05	0.981391E+05	0.173E+02	-0.625E-01	-0.360E-02	-0.228E+03
-30.0	0.978930E+05	0.979129E+05	0.173E+02	-0.199E+02	-0.115E+01	-0.246E+03
-22.0	0.976780E+05	0.976870E+05	0.153E+02	-0.900E+01	-0.588E+00	-0.215E+03
-15.0	0.974783E+05	0.974896E+05	0.162E+02	-0.113E+02	-0.700E+00	-0.200E+03
-7.0	0.972446E+05	0.972643E+05	0.167E+02	-0.197E+02	-0.118E+01	-0.234E+03
0.0	0.970377E+05	0.970674E+05	0.150E+02	-0.298E+02	-0.198E+01	-0.207E+03
8.0	0.968283E+05	0.968427E+05	0.159E+02	-0.144E+02	-0.904E+00	-0.209E+03
16.0	0.966082E+05	0.966182E+05	0.153E+02	-0.994E+01	-0.649E+00	-0.220E+03
23.0	0.964042E+05	0.964221E+05	0.153E+02	-0.178E+02	-0.116E+01	-0.204E+03
31.0	0.961963E+05	0.961981E+05	0.159E+02	-0.181E+01	-0.114E+00	-0.208E+03
38.0	0.959836E+05	0.960024E+05	0.165E+02	-0.189E+02	-0.114E+01	-0.213E+03
45.0	0.957640E+05	0.957790E+05	0.162E+02	-0.150E+02	-0.928E+00	-0.220E+03
53.0	0.955510E+05	0.955837E+05	0.159E+02	-0.327E+02	-0.206E+01	-0.213E+03

61.0	0.953313E+05	0.953609E+05	0.165E+02	-0.296E+02	-0.180E+01	-0.220E+03
69.0	0.951133E+05	0.951384E+05	0.162E+02	-0.251E+02	-0.155E+01	-0.218E+03
76.0	0.948907E+05	0.949439E+05	0.162E+02	-0.532E+02	-0.329E+01	-0.223E+03
84.0	0.946963E+05	0.947218E+05	0.141E+02	-0.255E+02	-0.180E+01	-0.194E+03
92.0	0.944990E+05	0.945001E+05	0.147E+02	-0.113E+01	-0.764E-01	-0.197E+03
99.0	0.942953E+05	0.943063E+05	0.153E+02	-0.110E+02	-0.719E+00	-0.204E+03
107.0	0.940843E+05	0.940851E+05	0.147E+02	-0.875E+00	-0.594E-01	-0.211E+03
114.0	0.938766E+05	0.938919E+05	0.159E+02	-0.152E+02	-0.960E+00	-0.208E+03
122.0	0.936686E+05	0.936712E+05	0.153E+02	-0.256E+01	-0.167E+00	-0.208E+03
130.0	0.934700E+05	0.934509E+05	0.156E+02	0.191E+02	0.123E+01	-0.199E+03
137.0	0.932676E+05	0.932583E+05	0.156E+02	0.931E+01	0.597E+00	-0.202E+03
145.0	0.930563E+05	0.930385E+05	0.159E+02	0.178E+02	0.112E+01	-0.211E+03
152.0	0.928427E+05	0.928454E+05	0.171E+02	-0.359E+01	-0.216E+00	-0.214E+03
160.0	0.926323E+05	0.926271E+05	0.153E+02	0.512E+01	0.334E+00	-0.210E+03
168.0	0.924183E+05	0.924082E+05	0.159E+02	0.101E+02	0.638E+00	-0.214E+03
175.0	0.922183E+05	0.922168E+05	0.141E+02	0.150E+01	0.106E+00	-0.200E+03
183.0	0.920307E+05	0.919984E+05	0.144E+02	0.323E+02	0.224E+01	-0.188E+03
190.0	0.918336E+05	0.918075E+05	0.162E+02	0.261E+02	0.162E+01	-0.197E+03
198.0	0.916307E+05	0.915896E+05	0.153E+02	0.411E+02	0.268E+01	-0.203E+03
206.0	0.914203E+05	0.913720E+05	0.150E+02	0.483E+02	0.322E+01	-0.210E+03
213.0	0.912086E+05	0.911819E+05	0.155E+02	0.268E+02	0.172E+01	-0.212E+03
221.0	0.909983E+05	0.909648E+05	0.167E+02	0.335E+02	0.200E+01	-0.210E+03
228.0	0.907843E+05	0.907751E+05	0.173E+02	0.913E+01	0.526E+00	-0.214E+03
236.0	0.905610E+05	0.905586E+05	0.167E+02	0.238E+01	0.142E+00	-0.223E+03
243.0	0.903460E+05	0.903694E+05	0.162E+02	-0.234E+02	-0.145E+01	-0.215E+03
251.0	0.901433E+05	0.901534E+05	0.159E+02	-0.101E+02	-0.637E+00	-0.203E+03
259.0	0.899293E+05	0.899377E+05	0.147E+02	-0.850E+01	-0.577E+00	-0.214E+03
266.0	0.897280E+05	0.897493E+05	0.156E+02	-0.213E+02	-0.137E+01	-0.201E+03
274.0	0.895183E+05	0.895342E+05	0.159E+02	-0.159E+02	-0.100E+01	-0.210E+03
281.0	0.893166E+05	0.893462E+05	0.150E+02	-0.296E+02	-0.197E+01	-0.202E+03

POLYNOMIAL COEFFICIENTS FOR FACTOR 1

COEFF	SIGMA
0.97067437E+05	0.34519031E+01
-0.28114578E+02	0.13985898E-01
0.22657185E-02	0.95030075E-04

REDUCED CHI SQUARE= 397.19
FTEST = 0.97E+06
RMUL = 0.10E+01
R(1) = -0.10E+01
R(2) = 0.90E-02

TIME	P	YFIT	SIGMA P	P-YFIT	(P-YFIT)/SIGP	DEL P
-281.0	0.306660E+05	0.306652E+05	0.115E+01	0.820E+00	0.710E+00	0.000E+00
-274.0	0.306783E+05	0.306651E+05	0.833E+00	0.132E+02	0.149E+02	0.123E+02
-266.0	0.306817E+05	0.306651E+05	0.333E+00	0.166E+02	0.497E+02	0.334E+01
-258.0	0.306767E+05	0.306651E+05	0.333E+00	0.116E+02	0.348E+02	-0.508E+01
-251.0	0.306703E+05	0.306650E+05	0.333E+00	0.530E+01	0.159E+02	-0.634E+01
-243.0	0.306647E+05	0.306650E+05	0.333E+00	-0.328E+00	-0.984E+00	-0.566E+01
-235.0	0.306630E+05	0.306650E+05	0.000E+00	-0.196E+01	-0.196E+16	-0.167E+01
-228.0	0.306623E+05	0.306649E+05	0.333E+00	-0.259E+01	-0.778E+01	-0.672E+00

-220.0	0.306627E+05	0.306649E+05	0.333E+00	-0.221E+01	-0.664E+01	0.340E+00
-212.0	0.306610E+05	0.306648E+05	0.000E+00	-0.384E+01	-0.384E+16	-0.167E+01
-205.0	0.306590E+05	0.306648E+05	0.000E+00	-0.581E+01	-0.581E+16	-0.200E+01
-197.0	0.306600E+05	0.306648E+05	0.000E+00	-0.477E+01	-0.477E+16	0.100E+01
-190.0	0.306613E+05	0.306647E+05	0.333E+00	-0.341E+01	-0.102E+02	0.133E+01
-182.0	0.306627E+05	0.306647E+05	0.333E+00	-0.203E+01	-0.609E+01	0.134E+01
-175.0	0.306617E+05	0.306647E+05	0.333E+00	-0.300E+01	-0.899E+01	-0.100E+01
-167.0	0.306613E+05	0.306645E+05	0.333E+00	-0.330E+01	-0.989E+01	-0.340E+00
-159.0	0.306640E+05	0.306646E+05	0.000E+00	-0.586E+00	-0.586E+15	0.267E+01
-152.0	0.306643E+05	0.306646E+05	0.333E+00	-0.223E+00	-0.668E+00	0.329E+00
-144.0	0.306627E+05	0.306643E+05	0.333E+00	-0.184E+01	-0.553E+01	-0.166E+01
-136.0	0.306633E+05	0.306645E+05	0.333E+00	-0.115E+01	-0.345E+01	0.660E+00
-129.0	0.306650E+05	0.306644E+05	0.000E+00	0.559E+00	0.559E+15	0.167E+01
-121.0	0.306647E+05	0.306644E+05	0.333E+00	0.266E+00	0.797E+00	-0.332E+00
-114.0	0.306637E+05	0.306644E+05	0.333E+00	-0.699E+00	-0.210E+01	-0.100E+01
-106.0	0.306630E+05	0.306643E+05	0.000E+00	-0.133E+01	-0.133E+16	-0.668E+00
-98.0	0.306620E+05	0.306643E+05	0.000E+00	-0.229E+01	-0.229E+16	-0.100E+01
-91.0	0.306627E+05	0.306643E+05	0.333E+00	-0.159E+01	-0.477E+01	0.668E+00
-83.0	0.306637E+05	0.306642E+05	0.333E+00	-0.551E+00	-0.165E+01	0.100E+01
-75.0	0.306640E+05	0.306642E+05	0.000E+00	-0.180E+00	-0.180E+15	0.332E+00
-68.0	0.306630E+05	0.306641E+05	0.000E+00	-0.114E+01	-0.114E+16	-0.100E+01
-60.0	0.306630E+05	0.306641E+05	0.000E+00	-0.111E+01	-0.111E+16	0.000E+00
-53.0	0.306659E+05	0.306641E+05	0.000E+00	0.930E+00	0.930E+15	0.200E+01
-45.0	0.306623E+05	0.306640E+05	0.333E+00	-0.170E+01	-0.511E+01	-0.267E+01
-38.0	0.306603E+05	0.306640E+05	0.333E+00	-0.367E+01	-0.110E+02	-0.200E+01
-30.0	0.306607E+05	0.306640E+05	0.333E+00	-0.329E+01	-0.988E+01	0.340E+00
-22.0	0.306600E+05	0.306639E+05	0.000E+00	-0.392E+01	-0.392E+16	-0.666E+00
-15.0	0.306597E+05	0.306639E+05	0.333E+00	-0.422E+01	-0.127E+02	-0.332E+00
-7.0	0.306599E+05	0.306638E+05	0.000E+00	-0.485E+01	-0.485E+16	-0.668E+00
0.0	0.306537E+05	0.306638E+05	0.667E+00	-0.101E+02	-0.152E+02	-0.533E+01
8.0	0.306573E+05	0.306638E+05	0.632E+00	-0.645E+01	-0.732E+01	0.366E+01
16.0	0.306650E+05	0.306637E+05	0.000E+00	0.126E+01	0.126E+16	0.767E+01
23.0	0.306680E+05	0.306637E+05	0.000E+00	0.429E+01	0.429E+16	0.300E+01
31.0	0.306670E+05	0.306637E+05	0.000E+00	0.333E+01	0.333E+16	-0.100E+01
38.0	0.306667E+05	0.306636E+05	0.333E+00	0.304E+01	0.911E+01	-0.332E+00
46.0	0.306660E+05	0.306636E+05	0.000E+00	0.241E+01	0.241E+16	-0.668E+00
53.0	0.306633E+05	0.306636E+05	0.333E+00	-0.234E+00	-0.703E+00	-0.267E+01
61.0	0.306627E+05	0.306635E+05	0.333E+00	-0.855E+00	-0.257E+01	-0.660E+00
69.0	0.306640E+05	0.306635E+05	0.000E+00	0.516E+00	0.516E+15	0.133E+01
76.0	0.306640E+05	0.306634E+05	0.000E+00	0.551E+00	0.551E+15	0.000E+00
84.0	0.306637E+05	0.306634E+05	0.333E+00	0.258E+00	0.773E+00	-0.332E+00
92.0	0.306629E+05	0.306634E+05	0.000E+00	-0.137E+01	-0.137E+16	-0.167E+01
99.0	0.306657E+05	0.306633E+05	0.333E+00	0.233E+01	0.700E+01	0.367E+01
107.0	0.306667E+05	0.306633E+05	0.333E+00	0.337E+01	0.101E+02	0.100E+01
114.0	0.306650E+05	0.306633E+05	0.000E+00	0.173E+01	0.173E+16	-0.167E+01
122.0	0.306667E+05	0.306632E+05	0.333E+00	0.344E+01	0.103E+02	0.167E+01
130.0	0.306677E+05	0.306632E+05	0.333E+00	0.448E+01	0.134E+02	0.100E+01
137.0	0.306657E+05	0.306632E+05	0.333E+00	0.252E+01	0.755E+01	-0.200E+01
145.0	0.306620E+05	0.306631E+05	0.577E+00	-0.111E+01	-0.193E+01	-0.367E+01
152.0	0.306620E+05	0.306631E+05	0.000E+00	-0.108E+01	-0.108E+16	0.000E+00
160.0	0.306627E+05	0.306630E+05	0.333E+00	-0.375E+00	-0.112E+01	0.668E+00
168.0	0.306610E+05	0.306630E+05	0.577E+00	-0.200E+01	-0.347E+01	-0.167E+01
175.0	0.306540E+05	0.306630E+05	0.000E+00	-0.897E+01	-0.897E+16	-0.700E+01
183.0	0.306610E+05	0.306629E+05	0.577E+00	-0.193E+01	-0.334E+01	0.700E+01
190.0	0.306667E+05	0.306629E+05	0.333E+00	0.377E+01	0.113E+02	0.567E+01
198.0	0.306667E+05	0.306629E+05	0.333E+00	0.381E+01	0.114E+02	0.000E+00

```
206.0  0.306657E+05  0.306628E+05 0.333E+00 0.285E+01 0.855E+01-0.100E+01
213.0  0.306650E+05  0.306628E+05 0.000E+00 0.221E+01 0.221E+15-0.666E+00
221.0  0.306640E+05  0.306627E+05 0.000E+00 0.125E+01 0.125E+16-0.100E+01
228.0  0.306630E+05  0.306627E+05 0.000E+00 0.289E+00 0.289E+15-0.100E+01
236.0  0.306600E+05  0.306627E+05 0.000E+00-0.267E+01-0.267E+16-0.300E+01
243.0  0.306630E+05  0.306626E+05 0.000E+00 0.363E+00 0.363E+15 0.300E+01
251.0  0.306637E+05  0.306626E+05 0.333E+00 0.107E+01 0.320E+01 0.666E+00
259.0  0.306647E+05  0.306626E+05 0.333E+00 0.211E+01 0.632E+01 0.100E+01
266.0  0.306627E+05  0.306625E+05 0.333E+00 0.141E+00 0.422E+00-0.200E+01
274.0  0.306630E+05  0.306625E+05 0.000E+00 0.512E+00 0.512E+15 0.332E+00
281.0  0.306663E+05  0.306625E+05 0.333E+00 0.387E+01 0.116E+02 0.333E+01


POLYNOMIAL COEFFICIENTS FOR FACTOR 2
       COEFF          SIGMA
0.30663816E+05 0.47810167E+00
-0.48555061E-02 0.29056200E-02

REDUCED CHI SQUARE=     17.14
FTEST =  0.28E+01
RMUL =  0.19E+00
R(1) = -0.19E+00


EXTRAPOLATION TIMES ABOUT MID POINT OF PLATEAU FOR POLYNOMIALS
FIRST PLATEAU =      -464.      464.
SECOND PLATEAU =      -463.      463.
REAL TIMES ARE      182. SECONDS ON EITHER SIDE OF
     204.      769. FOR FIRST PLATEAU, AND      1133.      1695. FOR SECOND



EXTRAPOLATED VALUES AND SIGMAS BACK AND FORWARD IN TIME
THEN PRECISIONS, = BACK, FORWARD AND REQUESTED

R1  -0.302E+02 0.891E-01  -0.260E+02 0.891E-01    339.    292.     30.
F1   0.111E+05 0.217E+02   0.845E+05 0.217E+02   5196.   3934.    100.
F2   0.307E+05 0.143E+01   0.307E+05 0.143E+01  21479.  21476.    100.

ACTIVATION ENERGY FROM =ACTIVE=
     E  SIG E  FACTOR OR RATE
67.925  0.598  428.96  0.04  -0.5169E+02   0.1263E+00
               421.30  0.02  -0.3021E+02   0.8910E-01

CONTRIBUTION OF RATE AND TEMPERATURE TO VARIANCE(=SIGE**2)
      RATE         TEMP
 0.235E+00   0.123E+00
NUMBER OF POINTS USED TO GET POLYNOMIALS WAS  75 AND  75

E VALUES, TIMES AND SAMPLE WEIGHTS SO FAR:

 1 67.925  0.598      951.   0.110725E+06  21.11
FACSET
   433.0         1
```

```
   14  #D 7698:
FACSET
     0.0          2
   14  #D 0000:
FACSET
     0.0          3
   14  #G 0000:
FACSET
     0.0          4
   14  #G 0000:
```

FACTOR LEVELS SET TO   433.000   0.000   0.000   0.000

****** MEASUREMENT FOR PLATEAU NUMBER 3 ******

NO.	F	I	READING	SIGMA	RD	TIME	BAD	RCHI	TESTCH
1	1	0	0.7689969E+05	0.2338329E+02	3	2064.	0	0.0	4000.0
1	2	1	0.3131233E+05	0.3333346E+00	3	2064.	0	0.0	1500.0
2	1	0	0.7659131E+05	0.2367371E+02	3	2071.	0	0.0	4000.0
2	2	1	0.3131033E+05	0.3333346E+00	3	2071.	0	0.0	1500.0
3	1	0	0.7625269E+05	0.2511530E+02	3	2079.	0	0.0	4000.0
3	2	1	0.3130800E+05	0.0000000E+00	3	2079.	0	0.0	1500.0
4	1	0	0.7593431E+05	0.2511530E+02	3	2086.	0	0.0	4000.0
4	2	1	0.3130333E+05	0.3333346E+00	3	2086.	0	0.0	1500.0
5	1	0	0.7560169E+05	0.2425101E+02	3	2094.	0	0.0	4000.0
5	2	1	0.3130433E+05	0.3333346E+00	3	2094.	0	0.0	1500.0
6	1	0	0.7526400E+05	0.2492605E+02	3	2102.	0	0.0	4000.0
6	2	1	0.3130233E+05	0.3333346E+00	3	2102.	0	0.0	1500.0
7	1	0	0.7492600E+03	0.2482605E+02	3	2109.	0	0.0	4000.0
7	2	1	0.3130300E+05	0.0000000E+00	3	2109.	0	0.0	1500.0
8	1	0	0.7460500E+05	0.2367136E+02	3	2117.	0	0.0	4000.0
8	2	1	0.3130500E+05	0.0000000E+00	3	2117.	0	0.0	1500.0
9	1	0	0.7429531E+05	0.2396062E+02	3	2124.	0	0.0	4000.0
9	2	1	0.3130333E+03	0.3333346E+00	3	2124.	0	0.0	1500.0
10	1	0	0.7397231E+05	0.2397452E+02	3	2132.	0	0.0	4000.0
10	2	1	0.3130167E+05	0.3333346E+00	3	2132.	0	0.0	1500.0
11	1	0	0.7365031E+05	0.2309342E+02	3	2139.	0	0.0	4000.0
11	2	1	0.3130200E+05	0.0000000E+00	3	2139.	0	0.0	1500.0
12	1	0	0.7333300E+05	0.2424372E+02	3	2147.	0	0.0	4000.0
12	2	1	0.3130200E+05	0.0000000E+00	3	2147.	0	0.0	1500.0
13	1	0	0.7302031E+05	0.2338329E+02	3	2154.	0	0.0	4000.0
13	2	1	0.3130033E+05	0.3333346E+00	3	2154.	0	0.0	1500.0
14	1	0	0.7270731E+05	0.2453795E+02	3	2162.	0	0.0	4000.0
14	2	1	0.3130033E+05	0.3333346E+00	3	2162.	0	0.0	1500.0
15	1	0	0.7238200E+05	0.2309401E+02	3	2170.	0	0.0	4000.0
15	2	1	0.3129400E+05	0.5773508E+00	3	2170.	0	0.0	1500.0
16	1	0	0.7207600E+05	0.2281082E+02	3	2177.	0	0.0	4000.0
16	2	1	0.3129633E+05	0.8819175E+00	3	2177.	0	0.0	1500.0
17	1	0	0.7175600E+05	0.2426930E+02	3	2185.	0	0.0	4000.0
17	2	1	0.3130467E+05	0.3333346E+00	3	2185.	0	0.0	1500.0
18	1	0	0.7143331E+05	0.2396062E+02	3	2193.	0	0.0	4000.0
18	2	1	0.3130900E+05	0.0000000E+00	3	2193.	0	0.0	1500.0

TGRUNF summary obtained for latest 10 activation energies. The calculations are performed at the end of every temperature plateau.

```
E VALUES, TIMES AND SAMPLE WEIGHTS SO FAR:

  1 67.212  0.710      5979.   0.947759E+05  47.62
  2 61.139  0.457      6798.   0.763499E+05  51.19
  3 70.099  0.888      7612.   0.489558E+05  56.51
  4 63.637  0.691      8431.   0.328628E+05  59.64
  5 59.779  0.631      9243.   0.917692E+04  64.24
  6 63.345  0.516     10055.  -0.483650E+04  66.96
  7 69.759  0.791     10874.  -0.251777E+05  70.91
  8 66.657  0.399     11781.  -0.380109E+05  73.40
  9 66.280  0.658     12685.  -0.639526E+05  78.44
 10 63.573  0.366     13588.  -0.777644E+05  81.12

WEIGHTED MEAN E IS  64.447 + OR -   0.172
REDUCED CHISQ FOR E CALCULATION IS 28.410 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS   0.612
CALCULATED OVER    9. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 64.110 TO 64.784
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF E AND UNBIASED ESTIMATE OF
SIG E ARE 65.148 AND  1.088 WHICH GIVES CONFIDENCE RANGE
OF 63.016 TO 67.280
CHI-SQUARE FOR UNWEIGHTED E IS  30.26
```

FORTRAN listings for the routines in the program TGRUNF, which runs the computer-controlled factor-jump thermogravimetry apparatus.

```
$ASSM
TGRUN PROG
$FORT
$TRGT 16
C   ****************************************************************
C
C   TGRUN PROGRAM
C
C   ****************************************************************
C
C   MAIN PROGRAM TO RUN THERMOGRAVIMETRY EXPERIMENT
C   WRITTEN BY B.DICKENS
C   NATIONAL BUREAU OF STANDARDS
C   WASHINGTON D.C.
C   TELEPHONE (301) 921-3322
C
C   FOR DOCUMENTATION, SEE NBS INTERNAL REPORT $$$
C
C   THE MAIN PROGRAM NEEDS ALL COMMON BLOCKS
C
C     COMMON /BIAS/
C     COMMON /DERIVE/
C     COMMON /FACTR/
C     COMMON /IO/
C     COMMON /MIDASR/
C     COMMON /PARAMS/
C     COMMON /PLAT/
C     COMMON /STEPS/
C     COMMON /WHEN /
C
C   THE PROGRAM CALLS THE FOLLOWING SUBROUTINES:
C
C     ACTIVE
C     BAKEIT
C     BIASVL
C     EXTPOL
C     FACSET
C     INIT
C     OVERLY
C     PLATTO
C     PRECIS
C     PRIME
C     RESPON
C     TRAMID
C     WTAVER
C
C   8 FACTORS MAXIMUM
C
C
C   COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C   COMMON /BIAS/
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C   COMMON /DERIVE/
C
      EQUIVALENCE (PRECE,DUMP(131))
C
C   COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
```

```
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190)),(NVDC,DUMP(191))
C
C  COMMON /PARAMS/
C
      DIMENSION A01(8),A02(8),SIGA01(8),SIGA02(8)
     &,A1(4,8),A2(4,8),SIGA1(4,8),SIGA2(4,8)
     &,FACT1(2,8),SIGF1(2,8),RATE1(2,8),SIGR1(2,8)
     &,FACT2(2,8),SIGF2(2,8),RATE2(2,8),SIGR2(2,8)
     &,TIMEF(2),TIMEB(2)
      EQUIVALENCE (A01(1),DUMP(200)),(A02(1),DUMP(208))
     &, (SIGA01(1),DUMP(216)),(SIGA02(1),DUMP(224))
     &, (A1(1,1),DUMP(232)),(A2(1,1),DUMP(264))
     &, (SIGA1(1,1),DUMP(296)),(SIGA2(1,1),DUMP(328))
      EQUIVALENCE (FACT1(1,1),DUMP(360)),(SIGF1(1,1),DUMP(376))
     &,(RATE1(1,1),DUMP(392)),(SIGR1(1,1),DUMP(408))
     &,(FACT2(1,1),DUMP(424)),(SIGF2(1,1),DUMP(440))
     &,(RATE2(1,1),DUMP(456)),(SIGR2(1,1),DUMP(472))
     &,(KOUNT1,DUMP(488)),(KOUNT2,DUMP(489))
      EQUIVALENCE (TIMEF(1),DUMP(500)),(TIMEB(1),DUMP(502))
     &,(DAY,DUMP(504)),(HDTIME,DUMP(505))
     &,(SKIP,DUMP(506)),(RATMIN,DUMP(507)),(RATMAX,DUMP(508))
     &,(TJUMP,DUMP(509))
C
C  COMMON /PLAT/
C
      DIMENSION RCHI(8)
      EQUIVALENCE (LPLAT,DUMP(510)),(NELAPS,DUMP(511))
     &,(NFIRST,DUMP(512)),(NCHI,DUMP(513)),(NSTOP,DUMP(514))
     &,(START,DUMP(515)),(RCHI(1),DUMP(516))
     &,(NOTHER,DUMP(524)),(KSTORE,DUMP(525))
     &,(ENDSIG,DUMP(526))
     &,(TOIM4,DUMP(527)),(FACTIM,DUMP(528)),(EQUIL,DUMP(529))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C  COMMON /STEPS/
C
      DIMENSION F(4),DELF(4),DESIGN(4,16),FDER(4)
      EQUIVALENCE (F(1),DUMP(800)),(DELF(1),DUMP(804))
     &,(DESIGN(1,1),DUMP(808)),(NFVEC,DUMP(872)),(FDER(1),DUMP(873))
C
```

```
C  COMMON /WHEN/
C
      DIMENSION TPLAT1(2),TPLAT2(2),NTYME(3)
      EQUIVALENCE (BEGIN,DUMP(880)),(TPLAT1(1),DUMP(881))
     &,(TPLAT2(1),DUMP(883)),(NTYME(1),DUMP(885))
C
C
C
      DIMENSION TF(5,8),STF(5,8),TYME(5),FTIME(4)
     &,PRECFF(8),PRECFB(8),PRECRF(8),PRECRB(8)
     &,PSTOFF(8),PSTOFB(8),PSTORF(8),PSTORB(8)
     &,E(10),SIGE(10),TEMP(2,10),SIGT(2,10),ETIME(10),SAMPWT(10),C(10)
     &,ERATE(2,10),SIGER(2,10)
     &,EFACT(2,10),SIGF(2,10)
     &,NPTS(2,10)
C
C
C  OVERLAY NAMES
C
      INTEGER*2 ISET1(3),ISET2(3),ISET3(3),IFACT(3),IPLATT(3)
     &,IPRECS(3),ISTATS(3)
C
      DATA ISET1/'OS','ET','U1'/,ISET2/'OS','ET','U2'/
     &,ISET3/'OS','ET','U3'/
     &,IFACT/'OF','AC','TS'/,IPLATT/'OP','LA','TT'/
     &,IPRECS/'OP','RE','CS'/,ISTATS/'OS','TA','TS'/
C
C  ASSIGN OVERLAY LOGICAL UNITS
C
      NSETUP=11
      NFACT=12
      NPLATT=13
      NPRECS=14
      NSTATS=15
C
C  INITIALISE MIDAS ARRAY =OUT= TO CONTAIN SHARP SIGN
C
      OUT(2)=X'23'
C  MAX NUMBER OF ACTIVATION ENERGIES FOR RUNNING AVERAGE
      NENG=10
      NENG1=NENG-1
C
  732 CALL OVERLY(ISET1,NSETUP,0)
      CALL INIT
      CALL PRIME
      CALL DESYNE
C
C CHECK RESPONSE TIMES
C
      CALL OVERLY(ISET2,NSETUP,1)
      CALL RESPON(FTIME,NERROR)
      IF(NERROR.LE.0)GO TO 730
      WRITE(NTTYO,731)NERROR
  731 FORMAT(' ERROR VALUE OF',I2,' FROM RESPON SUBROUTINE')
      GO TO 732
C
C  SETTLE QUESTION OF BIAS VOLTAGES
C
  730 CALL OVERLY(ISET3,NSETUP,1)
      CALL BIASVL
C
C  INITIALISE MIDAS AND SAMPLE AS REQUIRED
C
      CALL OVERLY(IFACT,NFACT,0)
      CALL BAKEIT(N,NTHCP,BEGINW,FINALW)
C
```

```
C  OPTIONAL RETURN TO BEGINNING
C
      IF(N.EQ.'B')GO TO 732
C
C  WE NOW HAVE ALL THE FACTORS, CHANGES AND VECTORS.
C  BEGIN EXPERIMENT
C
      BEGIN=TIMER(NTYME)
C
C  ZERO COUNTERS
C    LOOP = NUMBER OF TIMES THROUGH DESIGN MATRIX
C    NPLAT = ORDINAL NUMBER OF PRESENT PLATEAU
C    IE =  ORDINAL NUMBER OF PRESENT ACTIVATION ENERGY
C
      LOOP=0
      NPLAT=0
      IE=0
      NSTOP=0
      I=0
C
C  LOOP OVER FACTOR CHANGING PROCEDURE BUILT INTO DESIGN MATRIX
C
  300 CONTINUE
      NPLAT=NPLAT+1
      IOK=0
      IF(NSTOP.EQ.5)IOK=1
C
C  HAVE FACTORS ALREADY BEEN SET AFTER EXIT FROM PLATTO (NSTOP=2)
C   OR MUST THEY BE RESET (RATES TOO LOW, NSTOP=4, N1=1)
C
      IF(NSTOP.EQ.4)GO TO 13
      IF(NSTOP.EQ.2)GO TO 21
   11 I=I+1
   12 N1=0
C
C  SET FACTORS
C
   13 CALL OVERLY(IFACT,NFACT,0)
   10 CALL FACSET(I,N1,N2,N3,N4)
      FACTIM=TIMER(NTYME)-BEGIN
C
C  MAKE THE LATER PLATEAU THE EARLIER PLATEAU
C
   20 CONTINUE
      IF(NSTOP.EQ.2)GO TO 47
   21 DO 45 K=1,2
      TPLAT1(K)=TPLAT2(K)
      DO 46 J=1,NFACTS
      AO1(J)=AO2(J)
      SIGAO1(J)=SIGAO2(J)
      PSTOFF(J)=0.
      PSTOFB(J)=0.
      PSTORF(J)=0.
      PSTORB(J)=0.
      A1(K,J)=A2(K,J)
      SIGA1(K,J)=SIGA2(K,J)
      FACT1(K,J)=FACT2(K,J)
      SIGF1(K,J)=SIGF2(K,J)
      RATE1(K,J)=RATE2(K,J)
      SIGR1(K,J)=SIGR2(K,J)
   45 CONTINUE
      KOUNT1=KOUNT2
C
C  GO TO FACTOR MEASUREMENT, RETURN ON ERRORS, OR WHEN IT IS TIME
C  TO MAKE CHECKS ON PRECISION OF FACTOR MEASUREMENTS AND DERIVED
C  QUANTITIES SUCH AS ACTIVATION ENERGY.
```

```
C
      CALL OVERLY(IPLATT,NPLATT,0)
   45 CALL PLATTO(AO2,SIGAO2,A2,SIGA2,KOUNT2,TF,STF,TYME,NERROR,NPLAT
     &,BEGINW)
      IF(NSTOP.EQ.1.AND.KOUNT2.LE.3)GO TO 2000
      IF(KOUNT2.GT.3)GO TO 2002
      NSTOP=5
      GO TO 100
C
C  SET NEW FACTOR LEVELS IF PLATEAU MEASUREMENT COMPLETED
C
 2002 IF(NSTOP.EQ.2)GO TO 11
   47 IF(NERROR.GT.0)GO TO 45
C
C  CHECK PRECISION LEVELS
C
      CALL OVERLY(IPRECS,NPRECS,0)
      CALL PRECIS(IFGO,NPLAT,PRECFF,PRECFB,PRECRF,PRECRB)
      IF(NSTOP.NE.0)GO TO 49
      IF(IFGO.EQ.0) GO TO 49
      NIX=0
      DO 542 J=1,NFACT
      IF(PRECFF(J).LT.PSTOFF(J))NIX=NIX+1
      PSTOFF(J)=PRECF(J)
      IF(PRECFB(J).LT.PSTOFB(J))NIX=NIX+1
      PSTOFB(J)=PRECFB(J)
      IF(PRECRF(J).LT.PSTORF(J))NIX=NIX+1
      PSTORF(J)=PRECRF(J)
      IF(PRECRB(J).LT.PSTORB(J))NIX=NIX+1
      PSTORB(J)=PRECRB(J)
  542 CONTINUE
      IF(NIX.EQ.0.AND.NSTOP.LE.0)GO TO 50
      IF(NFILEB.GT.6)WRITE(NFILEB,543)
      IF(NPRINT.GE.3)WRITE(NPRINT,543)
  543 FORMAT(/' *** TERMINATE THIS SERIES OF MEASUREMENTS BECAUSE'
     &/'       PRECISION IS DETERIORATING ***'/)
      NSTOP=3
      GO TO 49
C
C  RESUME PLATEAU FACTOR MEASUREMENT IF PRECISION IS INADEQUATE
C
   50 CALL OVERLY(IPLATT,NPLATT,0)
      CALL PLATCO(AO2,SIGAO2,A2,SIGA2,KOUNT2,TF,STF,TYME,NERROR,NPLAT
     &,BEGINW)
      IF(NSTOP.EQ.1.AND.KOUNT2.LE.3)GO TO 2000
      IF(KOUNT2.GT.3)GO TO 2001
      NSTOP=5
      GO TO 100
 2001 IF(NSTOP.EQ.2)GO TO 11
      GO TO 47
   49 CONTINUE
C
C  JUMP OVER EXTRAPOLATIONS IF THIS IS FIRST TIME THROUGH
C    OR IF PREVIOUS PLATEAU WAS NOT MEASURED PROPERLY
C
      IF(NPLAT.EQ.1.OR.IOK.EQ.1)GO TO 200
C
C  TIME GAP BETWEEN PLATEAUS AND EXTRAPOLATION TIMES COMPUTED
C  IN SUBROUTINE PRECIS
C
C  EXTRAPOLATE QUANTITIES
C
      NACTE=-1
      DO 15 J=1,NFACTS
      K=KANDU(J)
      IF(K.EQ.2)NACTE=J
```

```
      L=IFDER(J)
C
C  DO FOR PLATEAU NPLAT-1
C
      CALL EXTPOL(A01(K),A1(1,K),SIGA01(K),SIGA1(1,K)
     &,NTERMS(K)
     &,TIMEF(1),TIMEB(1),FACT1(2,K),SIGF1(2,K),FACT1(1,K)
     &,SIGF1(1,K),RATE1(2,K),SIGR1(2,K),RATE1(1,K),SIGR1(1,K)
     &,PRECA,PRECA,PRECA,PRECA,L,M)
   15 CONTINUE
C
C  IF TEMPERATURE WAS MEASURED, CALCULATE ACTIVATION ENERGY
C
  150 IF(NACTE.LT.0)GO TO 100
      IE=IE+1
      IF(IE.LE.NENG)GO TO 101
C
C  MOVE E'S DOWN IN LIST
C
      DO 102 J=1,NENG1
      JJ=J+1
      E(J)=E(JJ)
      SIGE(J)=SIGE(JJ)
      ETIME(J)=ETIME(JJ)
      SAMPWT(J)=SAMPWT(JJ)
      C(J)=C(JJ)
      DO 102 K=1,2
      TEMP(K,J)=TEMP(K,JJ)
      SIGT(K,J)=SIGT(K,JJ)
      ERATE(K,J)=ERATE(K,JJ)
      SIGER(K,J)=SIGER(K,JJ)
      EFACT(K,J)=EFACT(K,JJ)
      SIGF(K,J)=SIGF(K,JJ)
      NPTS(K,J)=NPTS(K,JJ)
  102 CONTINUE
      IE=IE-1
  101 CONTINUE
      ETIME(IE)=(TPLAT2(1)-TPLAT1(2))/2. + TPLAT1(2)
      SAMPWT(IE)=(FACT1(2,1)+FACT2(1,1))/2.
      C(IE)=100.*(BEGINW-SAMPWT(IE))/(BEGINW-FINALW)
      CALL OVERLY(ISTATS,NSTATS,0)
      CALL ACTIVE(E(IE),SIGE(IE),IE,TEMP(1,IE),TEMP(2,IE)
     &,SIGT(1,IE),SIGT(2,IE),FACT1(2,2),FACT2(1,2)
     &,SIGF1(2,2),SIGF2(1,2),VTEMP,RATE1(2,1),RATE2(1,1)
     &,SIGR1(2,1),SIGR2(1,1),VRATE,AT,NERROR)
      IF(NFILEB.GT.6)WRITE(NFILEB,882)KOUNT1,KOUNT2
      IF(NPRINT.GE.3)WRITE(NPRINT,882)KOUNT1,KOUNT2
  882 FORMAT(' NUMBER OF POINTS USED TO GET POLYNOMIALS WAS',I4,' AND'
     &,I4/)
      ERATE(1,IE)=RATE1(2,1)
      ERATE(2,IE)=RATE2(1,1)
      SIGER(1,IE)=SIGR1(2,1)
      SIGER(2,IE)=SIGR2(1,1)
      EFACT(1,IE)=FACT1(2,1)
      EFACT(2,IE)=FACT2(1,1)
      SIGF(1,IE)=SIGF1(2,1)
      SIGF(2,IE)=SIGF2(1,1)
      NPTS(1,IE)=KOUNT1
      NPTS(2,IE)=KOUNT2
      IF(NERROR.EQ.0)GO TO 85
      IE=IE-1
      GO TO 100
C
C  IF E PRECISION IS INADEQUATE CONTINUE, PROVIDED THERE IS ROOM
C    IN THE ARRAYS IN PLATTO AND REGRES
C
```

```
 85  IF(ABS((E(IE)/SIGE(IE))).GT.PRECE) GO TO 80
     IF(NSTOP.NE.0)GO TO 80
     IE=IE-1
     GO TO 50
  80 CONTINUE
C
C  TEST FOR NEED TO CHANGE TEMPERATURE BECAUSE RATE OF WEIGHT LOSS
C    IS EITHER BELOW MINIMUM OR ABOVE MAXIMUM ALLOWED
C    CHECK ONLY LATEST RATE (FORWARD IN TIME)
C
 200 RATES=ABS(RATE2(2,1))
     IF(RATES.GT.RATMIN.AND.RATES.LT.RATMAX)GO TO 881
     IF(RATES.LT.RATMIN)SIGNT=1.
     IF(RATES.GT.RATMAX)SIGNT=-1.
     IF(NSTOP.NE.2)I=I+1
     F(1)=F(1)-DELF(1)*DESIGN(1,I)+SIGNT*TJUMP
     IF(NSTOP.NE.1)NSTOP=4
     N1=1
 881 IF(IE.LT.1.OR.IOK.EQ.1)GO TO 100
     IF(NPRINT.GE.3)WRITE(NPRINT,89)
  89 FORMAT(/' E VALUES, TIMES AND SAMPLE WEIGHTS SO FAR:'/)
     IF(NFILEB.GT.6)WRITE(NFILEB,89)
     DO 84 J=1,IE
     IF(NPRINT.GE.3)WRITE(NPRINT,82)J,E(J),SIGE(J),ETIME(J),SAMPWT(J)
    &,C(J)
  82 FORMAT(I4,2F7.3,F10.0,E15.6,F7.2)
     IF(NFILEB.GT.6)WRITE(NFILEB,82)J,E(J),SIGE(J),ETIME(J),SAMPWT(J)
    &,C(J)
  84 CONTINUE
     IF(NFILEE.LT.6)GO TO 870
     WRITE(NFILEE,82)IE,E(IE),SIGE(IE),ETIME(IE),SAMPWT(IE),C(IE)
     WRITE(NFILEE,820)(TEMP(K,IE),SIGT(K,IE),ERATE(K,IE),SIGER(K,IE)
    &,EFACT(K,IE),SIGF(K,IE)
    &,NPTS(K,IE),K=1,2)
 820 FORMAT(20X,F8.2,F6.2,2E15.4/34X,2E15.6,I5)
C
C
 870 IF(IE.LE.1)GO TO 100
     CALL WTAVER(E,SIGE,IE,WTEAVE,SIGWTE)
C
C  PUT EOF BETWEEN PLATEAU DATA IN DATA LOGGING FILE
C
 100 END FILE NFILEA
C
C  END OF LOOP THROUGH VECTOR OF DESIGN MATRIX
C
     IF(NSTOP.EQ.1)GO TO 2000
1000 IF(I.LT.NFVEC)GO TO 300
     I=0
     LOOP=LOOP+1
C
C  END OF LOOP THROUGH WHOLE DESIGN MATRIX
C
     GO TO 300
C
C  END OF JOB, USUALLY BECAUSE OF INSUFFICIENT SAMPLE
C
2000 CONTINUE
     IF(NPRINT.GE.3)WRITE(NPRINT,2010)
     IF(NFILEB.GT.6)WRITE(NFILEB,2010)
2010 FORMAT(' CONDITIONS TOO IMPRECISE, STOP')
     CALL OVERLY(ISET1,NSETUP,0)
     CALL OVERLY(ISET2,NSETUP,1)
     CALL SETPAR(0.,1,NERROR,NTHCP,TC)
     CALL SETPAR(760.,2,NERROR,NTHCP,TC)
     CALL SETPAR(250.,3,NERROR,NTHCP,TC)
     CALL SETPAR(0.,4,NERROR,NTHCP,TC)
C
C  END OTHER FILES
C
     END FILE NFILEA
     END FILE NFILEB
     END FILE NPRINT
     END FILE NFILEE
     STOP
     END
```

```
$ASSM
ACTIVE PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE ACTIVE(E,SIGE,IE,TEMP1,TEMP2,SIGT1,SIGT2
     &,TREAD1,TREAD2,SIGTR1,SIGTR2,VTEMP
     &,RATE1,RATE2,SIGR1,SIGR2,VRATE,AT,NERROR)
C
C ****************************************************************
C  SUBROUTINE TO CALCULATE ACTIVATION ENERGY FROM TEMPERATURE AND
C  RATE DATA
C
C  E = ACTIVATION ENERGY
C  SIGE = STANDARD DEVIATION OF E
C  IE = ORDINAL NUMBER OF E IN LIST OF E VALUES
C  TEMP1 = FIRST TEMPERATURE (IN DEGREES K)
C  TEMP2 = SECOND TEMPERATURE
C  SIGT1, SIGT2, = STANDARD DEVIATIONS OF TEMP1 AND TEMP2
C  TREAD = VALUES OF CORRESPONDING THERMOCOUPLE EMF'S IN MICROVOLTS
C  VTEMP = CONTRIBUTION OF TEMP TO VARIANCE OF E
C
C  RATE1, RATE2, SIGR1, SIGR2 AND VRATE ARE SIMILAR VALUES FOR RATES
C  AT = POLYNOMIAL COEFFICIENTS FOR EMF TO DEGREE K CONVERSION
C  NERROR = ERROR INDICATOR
C
C
C  NEEDS THE FOLLOWING COMMON BLOCK AND FUNCTION:
C    COMMON /IO/
C    TDEGAB
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(1060)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C
      DIMENSION TEMP(2),SIGT(2),TREAD(2),SIGTR(2),RATE(2),SIGR(2),AT(4)
C
C
      R=1.987/1000.
      R2=R**2
      NERROR=0
      TREAD(1)=TREAD1
      TREAD(2)=TREAD2
      SIGTR(1)=SIGTR1
      SIGTR(2)=SIGTR2
      RATE(1)=RATE1
      RATE(2)=RATE2
```

```
      SIGR(1)=SIGR1
      SIGR(2)=SIGR2
          DO 30 J=1,2
C
C  GET TEMPERATURE IN DEGREES KELVIN, TREAD IS IN MICROVOLTS
C
              TEMP(J)=TDEGAB(TREAD(J),AT)
              T=TREAD(J)+SIGTR(J)
              S=TDEGAB(T,AT)
              SIGT(J)=S-TEMP(J)
              IF (TEMP(J).LT.1.) GO TO 40
              IF (ABS(RATE(J)).LT.1.E-06) GO TO 40
30            CONTINUE
C
C  GET ACTIVATION ENERGY
C
          TDIFF=TEMP(2)-TEMP(1)
          TDIFF=ABS(TDIFF)
          IF (TDIFF.LT.1.E-01) GO TO 40
          T=RATE(2)/RATE(1)
          IF (T.LT.1.E-06) GO TO 40
          IF (T.GT.1.E+06) GO TO 40
          IF (T.LT.1.) T=1./T
          RTEMP=ALOG(T)
          ETEMP=TEMP(2)*TEMP(1)/TDIFF
          E=R*RTEMP*ETEMP
          IF (ABS(E).GT.200..OR.ABS(E).LT.1.) GO TO 40
          GO TO 90
C
C  DISCARD THIS POINT FOR VARIOUS POSSIBLE REASONS
C
40        CONTINUE
          IF(NPRINT.GE.3)WRITE(NPRINT,50)IE
     IF(NFILE8.GT.6)WRITE(NFILE8,50)IE
   50 FORMAT(' ACT. ENERGY NUMBER',I3,' DISCARDED')
          NERROR=8
       RETURN
90        CONTINUE
          E2=E**2
C
C  GET ASSOCIATED SIGMA
C  1) FROM RATES
C
          VRATE=((SIGR(2)/RATE(2))**2+(SIGR(1)/RATE(1))**2)*E
     22/RTEMP**2
C
C  2) FROM TEMPERATURES
C
          VTEMP=((SIGT(1)/TEMP(1)**2)**2+(SIGT(2)/TEMP(2)**2)
     2**2)*E2*ETEMP**2
C
C  COMPUTE TOTAL SIGMA ON E
C
          SIGE=SQRT(VRATE+VTEMP)
C
C  OUTPUT INITIAL AND CALCULATED VALUES
```

```
C
      TEMP(1)=TEMP(1)-273.15
      TEMP(2)=TEMP(2)-273.15
      IF(NPRINT.LT.3)GO TO 200
      WRITE(NPRINT,52)
   52 FORMAT(/' ACTIVATION ENERGY FROM =ACTIVE='
     &/6X,'E',2X,'SIG E',2X,'FACTOR OR RATE '/)
      WRITE(NPRINT,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J=1,2)
100         FORMAT (2F7.3,F8.2,F6.2,2E13.4/14X,F8.2,F6.2,2E13.4)
110         CONTINUE
C
C
120   WRITE(NPRINT,130)
130   FORMAT (/' CONTRIBUTION OF RATE AND TEMPERATURE TO VARIANCE'
     2,'(=SIGE**2)'/'          RATE          TEMP'/)
      WRITE(NPRINT,140) VRATE,VTEMP
140   FORMAT (2E12.3)
  200 IF(NFILEB.LE.6)RETURN
      WRITE(NFILEB,52)
      WRITE(NFILEB,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J=1,2)
      WRITE(NFILEB,130)
      WRITE(NFILEB,140)VRATE,VTEMP
      TEMP1=TEMP(1)
      TEMP2=TEMP(2)
      SIGT1=SIGT(1)
      SIGT2=SIGT(2)
      RETURN
      END
```

```
$ASSM
AVSDN PROG
$FORT
$TRGT 16
C ***********************************************************************
C
      SUBROUTINE AVSDN(AVE,SD,Q,NVAL)
C
C ***********************************************************************
C  SUBROUTINE TO CALCULATE AVERAGE AND STANDARD DEVIATION
C  AVE = AVERAGE VALUE
C  Q = INPUT VALUES
C  SD = STANDARD DEVIATION OF POPULATION OF Q
C  NVAL = NUMBER OF VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
C  RETURNS SD = 1.E+06 IF ONLY ONE VALUE IS AVAILABLE
C
C
      DIMENSION Q(NVAL)
C
C  CALCULATE AVERAGE
C
      AVE=0.
      DO 10 I=1,NVAL
   10 AVE=AVE+Q(I)
      AVE=AVE/FLOAT(NVAL)
      IF(NVAL.GT.1) GO TO 50
      SD=1.0E+06
      RETURN
   50 SD=0.
C
C  CALCULATE STANDARD DEVIATION
C
      DO 40 I=1,NVAL
 40   SD=SD+(AVE-Q(I))**2
      SD=SQRT(SD/(FLOAT(NVAL-1)))
      RETURN
      END
```

```
$ASSM
BAKEIT PROG
$FORT
$TRGT 16
C  *************************************************************
C
      SUBROUTINE BAKEIT(N,NTHCP,BEGINW,FINALW)
C
C  *************************************************************
C
C  SUBROUTINE TO PRESET MIDAS AND PRECONDITION SAMPLE BY A BAKEOUT
C
C  NEEDS COMMON BLOCKS /IO/ AND /MIDASR/
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
      DIMENSION NTYME(3)
      EQUIVALENCE (NTYME(1),DUMP(885))
C
C  INITIALISE MIDAS IF REQUIRED
C
      OUT(1)=37
      OUT(2)=X'23'
      DO 210 I=3,20
  210 OUT(I)='  '
  455 WRITE(NTTYO,430)
  430 FORMAT(' SET INITIAL CONDITIONS ON MIDAS, Y OR N?')
      READ(NTTYI,440)NA
  440 FORMAT(A1)
      IF(NA.EQ.'N')GO TO 460
      WRITE(NTTYO,451)
  451 FORMAT(' GIVE COMMAND, 36 CHARACTERS MAXIMUM'
     &/' SHARP SIGN WILL BE ADDED BY COMPUTER')
      READ(NTTYI,420)JUNK2
      WRITE(NTTYO,423)OUT
  420 FORMAT(18A2)
  423 FORMAT(I6,2X,19A2)
      CALL TRAMID(OUT(2),19,REPLY(2),19,0.)
      GO TO 455
  460 CONTINUE
C
C  SET TIME
```

```
C
      OUT(1)=12
      OUT(3)='IT'
      OUT(4)='00'
      OUT(5)='00'
      OUT(6)='00'
      OUT(7)='R '
      OUT(8)='  '
      OUT(9)='  '
      CALL TRAMID(OUT(2),19,REPLY(2),19,0.)
      OUT(1)=38
C
C  OPTIONAL RETURN TO BEGINNING
C
  735 WRITE(NTTYO,733)
  733 FORMAT(' GO FORWARD OR BACK? ENTER =F= OR =B=')
      READ(NTTYI,440)N
      IF(N.EQ.'B')RETURN
C
C  INPUT INITIAL AND FINAL WEIGHTS
C
  530 WRITE(NTTYO,500)
  500 FORMAT(/' GIVE INITIAL AND FINAL WEIGHTS IN MICROVOLTS, WITH SIGN'
     &/'  ON 2 LINES IN F15.0 FORMAT.'
     &/'  INITIAL WEIGHT OF 0 WILL CAUSE PROGRAM TO USE FIRST'
     &/'  WEIGHT READING IN DATA COLLECTION AS INITIAL WEIGHT'
     &/'  -I.E., AFTER BAKEOUT OF SAMPLE'/)
      READ(NTTYI,510)BEGINW
      READ(NTTYI,510)FINALW
  510 FORMAT(F15.0)
      WRITE(NTTYO,520)BEGINW,FINALW
  520 FORMAT(/' INITIAL AND FINAL WEIGHTS ARE:',2F15.0)
      WRITE(NTTYO,120)
      READ(NTTYI,440)NA
      IF(NA.NE.'Y')GO TO 530
      IF(NFILEB.GT.6)WRITE(NFILEB,520)BEGINW,FINALW
      IF(NPRINT.GE.3)WRITE(NPRINT,520)BEGINW,FINALW
C
C  INITIAL CONDITIONING OF SAMPLE VIA BAKEOUT
C
  310 WRITE(NTTYO,300)
  300 FORMAT(/' DO YOU WANT PRECONDITIONING OF THE SAMPLE? Y OR N?')
      READ(NTTYI,440)N
      IF(N.EQ.'N') GO TO 400
      IF(N.NE.'Y')GO TO 310
  150 WRITE(NTTYO,100)
  100 FORMAT(/' GIVE BAKEOUT TIME, TEMPERATURE, PRESSURE, FLOW N2,'
     &,' FLOW O2'/' EACH ON A SEPARATE LINE IN FORMAT F10.0')
      READ(NTTYI,110)TIME,TEMP,PRESS,FN2,FO2
  110 FORMAT(F10.0)
      WRITE(NTTYO,140)TIME,TEMP,PRESS,FN2,FO2
  140 FORMAT(/' BAKE TIME IN SECONDS IS',F7.0
     &/' BAKE TEMPERATURE IN DEGREES CELCIUS IS',F7.0
     &/' BAKE PRESSURE IN MM HG IS',F7.0
     &/' FLOW OF N2 IN SCC/MIN IS',F7.0
     &/' FLOW OF O2 IN SCC/MIN IS',F7.0/)
```

```
      WRITE(NTTYO,120)
  120 FORMAT(' OK OR NOT, Y OR N?')
      READ(NTTYI,440)NA
      IF(NA.NE.'Y')GO TO 150
      IF(NFILEB.GT.6)WRITE(NFILEB,140)TIME,TEMP,PRESS,FN2,FO2
      IF(NPRINT.GE.3)WRITE(NPRINT,140)TIME,TEMP,PRESS,FN2,FO2
      CALL SETPAR(TEMP,1,NERROR,NTHCP,TC)
      CALL SETPAR(PRESS,2,NERROR,NTHCP,TC)
      CALL SETPAR(FN2,3,NERROR,NTHCP,TC)
      CALL SETPAR(FO2,4,NERROR,NTHCP,TC)
C
C  WAIT IN LOOP FOR BAKE TIME TO ELAPSE
C
      BTIME=TIMER(NTYME)+TIME
  200 TNOW=TIMER(NTYME)
      IF(TNOW.LT.BTIME)GO TO 200
C
C REZERO CLOCK
C
  400 OUT(1)=12
      OUT(3)='IT'
      OUT(4)='00'
      OUT(5)='00'
      OUT(6)='00'
      OUT(7)='R '
      OUT(8)='  '
      OUT(9)='  '
      CALL TRAMID(OUT(2),19,REPLY(2),19,0.)
      OUT(1)=39
      RETURN
      END
```

```
$ASSM
BIASVL PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE BIASVL
C
C ****************************************************************
C  SUBROUTINE TO DETERMINE OR READ BIAS VOLTAGES =BIASV= ON ANALOG SCANNER
C
C  NEEDS THE FOLLOWING COMMON BLOCKS, FUNCTIONS AND SUBROUTINES:
C    COMMON /BIAS/
C    COMMON /FACTR/
C    COMMON /IO/
C    COMMON /MIDASR/
C    RPAR
C    SETPAR
C    TRAMID
C    WAYT
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /BIAS/
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C  COMMON /MIDASR/ (PART OF)
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
```

```
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160))
     &,(NUM(1),DUMP(180)),(JUNK2(1),OUT(3)),(NVDC,DUMP(191))
C
C
      DIMENSION MBIAS(8),BASEBI(8)
C
C
  300 IWAIT=0
      NGO=0
      J=0
C
C  DETERMINE WHAT TO DO
C
      DO 60 K=1,8
      MBIAS(K)=0
C
C  INIITIALISE EACH BIAS VOLTAGE TO ZERO
C
      BIASV(K)=0.
      SIGBV(K)=1.E-09
      IF(K.EQ.1)WRITE(NTTYO,5)
    5 FORMAT(' THIS ROUTINE IS SET UP FOR:'
     &/' WEIGHT AS "FACTOR" 1'
     &/' TEMPERATURE AS FACTOR 2'
     &/' PRESSURE AS FACTOR 3'
     &/' FLOW CHANNEL 1 AS FACTOR 4'
     &/' FLOW CHANNEL 2 AS FACTOR 5'
     &/' FACTORS 2 TO 5 ARE PRESET FOR THE BIAS MEASUREMENTS'/
     &/' FOR EACH FACTOR, IS BIAS VOLTAGE TO BE:'
     &/' SET = 0 (J=1)'
     &/' MEASURED (J=2)'
     &/' OR READ IN (J=3)?'
     &/' TERMINATE INPUT WITH J=0')
  105 WRITE(NTTYO,15)K
   15 FORMAT(' GIVE J FOR FACTOR',I3,' IN I1 FORMAT')
      READ (NTTYI,25)J
   25 FORMAT(I1)
      IF(J.EQ.0)GO TO 350
      IF(J.LT.0)GO TO 105
      IF(J.GT.3)GO TO 105
      GO TO (60,20,30),J
C
C  SET UP ARRAY AND COUNTER TO CONTROL MEASUREMENT OF BIAS VOLTAGE
C    WHEN READ-IN IS COMPLETED
C
   20 CONTINUE
      MBIAS(K)=1
      IWAIT=IWAIT+1
      GO TO 60
C
C  READ IN  BIAS VOLTAGE
C
   30 WRITE(NTTYO,35)
   35 FORMAT(' GIVE BIAS VOLTAGE AND SIGMA IN F7.3 FORMAT (2 LINES)')
      READ(NTTYI,45)BIASV(K),SIGBV(K)
   45 FORMAT(F7.3)
```

```
      IF(SIGBV(K).LT.1.E-09)SIGBV(K)=1.E-09
   60 CONTINUE
      KK=8
      GO TO 360
  350 KK=K-1
      IF (IWAIT.GT.0.AND.NFILEB.GT.6) WRITE (NFILEB,355)
      IF (IWAIT.GT.0.AND.NPRINT.GE.3) WRITE (NPRINT,355)
  355 FORMAT (//' BIAS VOLTAGE DETERMINATIONS FOLLOW:'//)
C
C  ARE ANY MEARSUREMENTS TO BE MADE
C
  360 CONTINUE
      IF(IWAIT.EQ.0)GO TO 120
C
C  START CLOCK
C
      OUT(1)=12
      OUT(3)='IT'
      OUT(4)='00'
      OUT(5)='00'
      OUT(6)='00'
      OUT(7)='R '
      OUT(8)='  '
      CALL TRAMID(OUT(2),19,REPLY,19,0.)
      OUT(1)=38
C
C  MEASURE BIAS VOLTAGES
C
      DO 150 K=1,KK
      IF(MBIAS(K).EQ.0)GO TO 150
      J=K-1
C
C
      GO TO (150,70,80,90,100,150,150,150),K
C
C  SET TEMP
C
   70 CALL SETPAR(100.,1,NERROR,NTHCP,TC)
C
C  STORE TARGET VALUES IN ARRAY BASEBI IN MICROVOLTS
C
      BASEBI(K)=TC*1000.
      GO TO 150
C
C  SET PRESS = NEED A FLOW RATE FOR THIS, USE N2
C
   80 CALL SETPAR(400.,J,NERROR,NTHCP,TC)
      BASEBI(K)=FLOAT(NVDC)*1000.*SCALE(K)
      KJ=K+1
      J=KJ-1
      CALL SETPAR(300.,J,NERROR,NTHCP,TC)
      BASEBI(KJ)=FLOAT(NVDC)*1000.*SCALE(KJ)
      GO TO 150
C
C  SET FLOWS
C
```

```
   90 CALL SETPAR(300.,J,NERROR,NTHCP,TC)
      BASEBI(K)=FLOAT(NVDC)*1000.*SCALE(K)
      GO TO 150
  100 CALL SETPAR(100.,J,NERROR,NTHCP,TC)
      BASEBI(K)=FLOAT(NVDC)*1000.*SCALE(K)
  150 CONTINUE
      NTIME=0
  290 NERR=0
C
C  DETERMINE BIAS VOLTAGES
C
      DO 160 K=1,KK
      IF(MBIAS(K).EQ.0)GO TO 160
C
C  WAIT TILL READING IS STEADY
C
      CALL WAYT(CHTEST(K),PTIME(K),INPUT(K),NRANGE(K),SCALE(K),TIMEST
     &,CHISQR,0.,NREADS(K),NTERMS(K),NERROR)
      IF(NERROR.EQ.0)GO TO 160
      IF(NPRINT.GE.3)WRITE(NPRINT,170)INPUT(K),CHTEST(K)
     &,CHISQR,TIMEST,NTIME,NERROR
  170 FORMAT(' INPUT',I2,' HAS UNSTEADY READING'
     &/' CHTEST =',F7.2,' CHI READINGS =',F7.2,' TIME =',F7.2
     &/' PASS =',I3,' NERROR =',I3,' *********************')
      IF(NFILEB.GT.6)WRITE(NFILEB,170)INPUT(K),CHTEST(K),CHISQR,TIMEST
     &,NTIME,NERROR
  160 NERR=NERR+NERROR
      NTIME=NTIME+1
      IF(NERR.NE.0.AND.NTIME.LE.1)GO TO 290
      IF(NERR.NE.0.AND.NTIME.GT.1)NGO=1
  200 DO 250 K=1,KK
      IF(MBIAS(K).EQ.0)GO TO 250
      BIASV(K)=0.
      SIGBV(K)=1.E-09
      CALL RPAR(BIASV(K),SIGBV(K),INPUT(K),NRANGE(K),SCALE(K)
     &,NREADS(K),0.,NERROR)
      IF(NERROR.EQ.0)GO TO 250
      IF(NPRINT.GE.3)WRITE(NPRINT,180)INPUT(K),NERROR
  180 FORMAT(' DVM ERROR FOR INPUT',I2,' NERROR ='I2,' *********')
      IF(NFILEB.GT.6)WRITE(NFILEB,180)INPUT(K),NERROR
      GO TO 300
  250 CONTINUE
C
C  CORRECT FOR FACTOR LEVELS
C
      DO 260 K=1,KK
  260 IF(MBIAS(K).NE.0)BIASV(K)=BIASV(K)-BASEBI(K)
C
C  OUTPUT BIAS VOLTAGES AND SIGMAS
C
  120 IF(NPRINT.GE.3)WRITE(NPRINT,50)(I,BIASV(I),SIGBV(I),I=1,KK)
   50 FORMAT(/' BIAS VOLTAGES AND SIGMAS FOR FACTOR INPUTS'
     &/' FACTOR      VOLTAGE          SIGMA'
     &/(I7,2E15.7))
      IF(NFILEB.GT.6)WRITE(NFILEB,50)(I,BIASV(I),SIGBV(I),I=1,KK)
C
C  CYCLE ROUND AGAIN UNDER OPERATOR CONTROL BECAUSE OF INSTABILITIES
C
      IF(NGO.EQ.1)GO TO 300
      RETURN
      END
```

```
$ASSM
CHISQ PROG
$FORT
$TRGT 16
C ***********************************************************************
C
      SUBROUTINE CHISQ(P,SP,XSQ,N)
C
C ***********************************************************************
C  SUBROUTINE TO CALCULATE REDUCED CHI SQUARE FOR N INPUT VALUES.
C  P = INPUT VALUES
C  SP = STANDARD DEVIATIONS OF P VALUES
C  XSQ = CHI-SQUARE VALUE FOR P AND SP ARRAYS
C  N = NUMBER OF P VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
      DIMENSION P(N),SP(N)
      AVE=0.
C
C  CALCULATE AVERAGE
C
      DO 10 I=1,N
 10   AVE=AVE+P(I)
      AVE=AVE/FLOAT(N)
      XSQ=0.
      IF(N.LE.1)RETURN
C
C  CALCULATE CHI SQUARE
C
      DO 20 I=1,N
 20   XSQ=XSQ+((AVE-P(I))/SP(I))**2
      XSQ=XSQ/FLOAT(N-1)
      RETURN
      END
```

```
$ASSM
DEKODE PROG
$FORT
$TRGT 16
C ***************************************************************
C
      SUBROUTINE DEKODE(J,F,L,NSIDE)
C
C ***************************************************************
C  SUBROUTINE TO DECODE ARRAY J(L) INTO =F=, I.E., ONE COMPUTER WORD
C  HAS TO WORK IN FLOATING POINT BECAUSE AN INTEGER MAY OVERFLOW
C    L IS THE NUMBER OF BYTES TO DECODE
C    NSIDE = 0  START AT LHS OF FIRST WORD (EACH WORD CONTAINS 2 BYTES)
C    NSIDE = 1  START AT RHS OF FIRST WORD
C
C
      INTEGER*2 J(L),K,KK
C
C  TABLE OF DECIMAL DIGITS:
C
      DIMENSION TABLE(10)
      DATA TABLE/0.,1.,2.,3.,4.,5.,6.,7.,8.,9./
C
      N=0
      NEG=0
      F=0.
C
C  LOOP OVER CHARACTER TRANSLATION, 20 CHARACTERS MAXIMUM
C
      DO 10 I=1,20
      KK=J(I)
C
C  EXAMINE EACH BYTE IN WORD IN TURN
C
   20 DO 11 MM=1,2
      M=MM-1
C
C  NSIDE = 1 MEANS BEGIN WITH LESS SIGNIFICANT BYTE FROM FIRST WORD
C
      IF(I.NE.1)GO TO 25
      IF(NSIDE.EQ.0)GO TO 25
      MM=2
      M=1
   25 CONTINUE
      K=KK/256
      IF(M.EQ.1)K=KK-K*256
C
C  CHECK FOR NEGATIVE NUMBER
C
   30 IF(K.EQ.45)NEG=1
C
C  JUMP OUT IF CHARACTER IS BLANK, + OR -
C
      IF(K.EQ.43.OR.K.EQ.45.OR.K.EQ.32)GO TO 60
      K=K-47
      IF(K.LE.0.OR.K.GT.10)GO TO 11
C
C  TACK EACH DIGIT ONTO NUMBER, COUNT NUMBER OF DIGITS
C
      F=F*10.+TABLE(K)
   60 N=N+1
      IF(N.EQ.L)GO TO 50
   11 CONTINUE
   10 CONTINUE
C
C  APPLY ANY MINUS SIGN
C
   50 IF(NEG.EQ.1)F=-F
      RETURN
      END
```

```
$ASSM
DESYNE PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE DESYNE
C
C ****************************************************************
C
C  SUBROUTINE TO SET UP DESIGN MATRIX WHICH GOVERNS COURSE OF EXPERIMENT
C
C    NEEDS COMMON /IO/
C          COMMON /MIDASR/
C          COMMON /PLAT/
C      AND COMMON /STEPS/
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO;DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /MIDASR/ (PART OF)
C
      EQUIVALENCE (NFILTR,DUMP(190))
C
C  COMMON /PLAT/
C
      DIMENSION RCHI(8)
      EQUIVALENCE (LPLAT,DUMP(510)),(NELAPS,DUMP(511))
     &,(NFIRST,DUMP(512)),(NCHI,DUMP(513)),(NSTOP,DUMP(514))
     &,(START,DUMP(515)),(RCHI(1),DUMP(516))
     &,(NOTHER,DUMP(524)),(KSTORE,DUMP(525))
     &,(ENDSIG,DUMP(526))
C
C  COMMON /STEPS/
C
      DIMENSION F(4),TOPF(4),DESIGN(4,16),FDER(4)
      EQUIVALENCE (F(1),DUMP(800)),(TOPF(1),DUMP(804))
     &,(DESIGN(1,1),DUMP(808)),(NFVEC,DUMP(872)),(FDER(1),DUMP(873))
C
C  INPUT FACTOR LEVELS, STEP SIZES AND DESIGN MATRIX
C
      J=0
      WRITE(NTTYO,301)
 301  FORMAT(' READY TO SET UP FACTOR LEVELS AND CHANGES'
     &/' ONE PER LINE')
  305 WRITE(NTTYO,330)
  330 FORMAT(' GIVE INITIAL T AND MAXIMUM T IN DEGREES CELCIUS'
     & /' F7.2 FORMAT, ONE PER LINE')
      READ(NTTYI,310)F(1),TOPF(1)
  310 FORMAT(F7.2)
      IF(J.NE.0)GO TO 415
```

```
 306 WRITE(NTTYO,320)
 320 FORMAT(' GIVE INITIAL P AND MAXIMUM P IN MM HG IN F7.2 FORMAT'
   & /' MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 MM HG')
     READ(NTTYI,310)F(2),TOPF(2)
     IF (TOPF(2).LE.0.001) TOPF(2)=200.
     IF(J.NE.0)GO TO 415
 307 WRITE(NTTYO,335)
 335 FORMAT(' GIVE INITIAL N2 FLOW AND MAXIMUM FLOW IN SCC/MIN'
   & /' MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 SCC/MIN')
     READ(NTTYI,310)F(3),TOPF(3)
     IF (TOPF(3).LE.0.001) TOPF(3)=200.
     IF(J.NE.0)GO TO 415
 308 WRITE(NTTYO,340)
 340 FORMAT(' GIVE INITIAL O2 FLOW AND MAXIMUM IN SCC/MIN'
   & /' MAXIMUM VALUE OF 0 AUTOMATICALLY CHANGED TO 200 SCC/MIN')
     READ(NTTYI,310)F(4),TOPF(4)
     IF (TOPF(4).LT.0.001) TOPF(4)=200.
 415 WRITE(NTTYO,345)(I,F(I),TOPF(I),I=1,4)
 345 FORMAT(' FACTOR LEVELS AND MAXIMA FOLLOW:'
   & /' NO.   INITIAL   MAXIMUM'
   & /(I4,2F10.2))
 346 FORMAT(I4,2F710.3)
 405 WRITE(NTTYO,420)
 420 FORMAT(' GIVE FACTOR NUMBER (1 TO 4) FOR FURTHER CHANGES'
   &/' 0 = NO CHANGES')
     READ(NTTYI,75)J
  75 FORMAT(I1)
     IF(J.LE.0)GO TO 368
     IF(J.GT.4)GO TO 405
     GO TO (305,306,307,308),J
C
C  READ IN DESIGN MATRIX
C
 368 WRITE(NTTYO,350)
350  FORMAT(' READY TO BUILD DESIGN MATRIX'
    &/' I.E., VECTORS FOR CHANGES IN FACTOR LEVELS'
    &/' MAXIMUM OF 16 VECTORS, ONE PER LINE'
    &/' GIVE LINE NUMBERS THEN 4 COMPONENTS'
    &/' FORMAT(I2/(F6.2))'
    &/' END WITH LINE NUMBER = 0')
390   READ(NTTYI,360)I
360   FORMAT(I2)
     IF(I.EQ.0)GO TO 440
     READ(NTTYI,361)(DESIGN(J,I),J=1,4)
361  FORMAT(F6.2)
     WRITE(NTTYO,362)I,(DESIGN(J,I),J=1,4)
 362 FORMAT(I3,4F10.3)
     GO TO 390
 440 CONTINUE
     WRITE(NTTYO,365)
 365 FORMAT(' INPUT NUMBER OF VECTORS IN I2 FORMAT')
     READ(NTTYI,360)NFVEC
     WRITE(NTTYO,363)
 363 FORMAT(' DESIGN MATRIX FOLLOWS')
     WRITE(NTTYO,362)(I,(DESIGN(J,I),J=1,4),I=1,NFVEC)
     WRITE(NTTYO,365)
```

```
366 FORMAT(' CHANCE FOR FURTHER CHANGES? 0=NO, 1=YES, IN I2 FORMAT')
    READ(NTTYI,360)J
    IF(J.GT.0)GO TO 368
C
C  STEADINESS CHECKS TO BE MADE DURING DATA COLLECTION?
C
    WRITE(NTTYO,370)
370 FORMAT(' STEADINESS CHECKS TO BE MADE DURING DATA COLLECTION?'
   &/'  GIVE 0 FOR NO, 1 FOR YES, I2 FORMAT')
    READ(NTTYI,360)NCHI
C
C  FILTER INPUT TO DVM?
C
    WRITE(NTTYO,371)
371 FORMAT(' FILTER INPUTS TO DVM? 0=NO,1=YES (YES TAKES'
   &/' 0.5 SEC/READ) GIVE IN I2 FORMAT')
    READ(NTTYI,360)NFILTR
    IF(NFILTR.GT.0)NFILTR=1
C
C  READ OTHER EXPERIMENTS ALSO?
C
    WRITE(NTTYO,372)
372 FORMAT(' READ OTHER EXPERIMENTS AT SAME TIME? 0=NO,1=YES,'
   &/,' GIVE IN I2 FORMAT')
    READ(NTTYI,360)NOTHER
    IF(NPRINT.LT.3)GO TO 455
    WRITE(NPRINT,345)(I,F(I),TOPF(I),I=1,4)
    WRITE(NPRINT,363)
    WRITE(NPRINT,362)(I,(DESIGN(J,I),J=1,4),I=1,NFVEC)
    WRITE(NPRINT,374)NCHI
374 FORMAT(' STEADINESS CHECK =',I2,' WHERE 0=NO,1=YES')
    WRITE(NPRINT,375)NFILTR
375 FORMAT(' INPUT FILTER TO DVM =',I2,' WHERE 0=NO,1=YES')
    WRITE(NPRINT,376)NOTHER
376 FORMAT(' READ OTHER EXPTS =',I2,' WHERE 0=NO,1=YES')
455 CONTINUE
460 IF(NFILEB.LE.6)GO TO 610
    WRITE(NFILEB,345)(I,F(I),TOPF(I),I=1,4)
    WRITE(NFILEB,363)
    WRITE(NFILEB,362)(I,(DESIGN(J,I),J=1,4),I=1,NFVEC)
    WRITE(NFILEB,374)NCHI
    WRITE(NFILEB,375)NFILTR
    WRITE(NFILEB,376)NOTHER
610 CONTINUE
    RETURN
    END
```

```
$ASSM
ENKODE PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE ENKODE(NARRAY,NVAR)
C
C ************************************************************
C  SUBROUTINE TO PUT INTEGER =NVAR= INTO =NARRAY= IN HOLLERITH FORM
C    PRODUCES 4 DIGITS, 2 TO A WORD IN =NARRAY=
C
C  NEEDS COMMON /MIDASR/
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /MIDASR/
C
      INTEGER*2 NUM(10)
      EQUIVALENCE (NUM(1),DUMP(180))
C
C
      DIMENSION NARRAY(2)
C
C
      K1=NVAR/1000
      K2=(NVAR-K1*1000)/100
      K3=(NVAR-K1*1000-K2*100)/10
      K4=NVAR-K1*1000-K2*100-K3*10
      NARRAY(1)=NUM(K1+1)*256+NUM(K2+1)
      NARRAY(2)=NUM(K3+1)*256+NUM(K4+1)
      RETURN
      END
```

```
$ASSM
EXTPOL PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE EXTPOL(ATO,AT,SIGATO,SIGAT
     &,NTERMS
     &,TIMEF,TIMEB,FACTF,SIGFF,FACTB,SIGFB
     &,RATEF,SIGRF,RATEB,SIGRB,PRECFF,PRECFB,PRECRF,PRECRB,L,M)
C
C ****************************************************************
C  SUBROUTINE TO USE POLYNOMIAL OF P=ATO+AT(1)*TIME**M(1)+AT(2)*TIME**M(2)
C    TO EXTRAPOLATE TO VALUES OF BOTH P AND DP/DT TO TIMES TIMEF AND TIMEB
C
C    ATO = DEFINED ABOVE
C    SIGATO = STANDARD DEVIATION OF ATO
C    AT = DEFINED ABOVE
C    SIGAT = STANDARD DEVIATION OF AT
C    RCHI = CHI-SQUARE VALUE FOR FIT
C    TIMEF = TIME TO EXTRAPOLATE TO IN FORWARD DIRECTION
C    TIMEB = TIME TO EXTRAPOLATE TO IN BACKWARD DIRECTION
C    FACTF = VALUE OF FACTOR AS EXTRAPOLATED FORWARD
C    FACTB = VALUE OF FACTOR AS EXTRAPOLATED BACKWARD
C    RATEF AND RATEB = CORESPONDING VALUES FOR FIRST DERIVATIVES
C    SIGFF, SIGFB, SIGRF, SIGRB = STANDARD DEVIATIONS FOR ABOVE
C    PRECFF,PRECFB,PRECRF,PRECRB = PRECISION OF EXTRAPOLATED
C      RATES AND FACTOR LEVELS
C
C  NEDDS NO COMMON BLOCKS OR SUBROUTINES
C
      DIMENSION AT(NTERMS),SIGAT(NTERMS),M(NTERMS)
C
C  EXTRAPOLATE FACTOR LEVEL
C
      FACTF=ATO
      FACTB=ATO
      SIGFF=SIGATO**2
      SIGFB=SIGATO**2
      IF(NTERMS.LE.0)GO TO 50
      DO 40 I=1,NTERMS
      FACTF=FACTF+AT(I)*TIMEF**M(I)
      FACTB=FACTB+AT(I)*TIMEB**M(I)
      SIGFF=SIGFF+(SIGAT(I)*TIMEF**M(I))**2
      SIGFB=SIGFB+(SIGAT(I)*TIMEB**M(I))**2
   40 CONTINUE
   50 CONTINUE
      SIGFF=SQRT(SIGFF)
      SIGFB=SQRT(SIGFB)
C
C  EXTRAPOLATE DERIVATIVE
C  I.E., IF W=A+B*T+C*T**2,
C  THEN DW/DT=B+2*C*T
C
      RATEF=0.
      RATEB=0.
```

```
      SIGRF=1.E-04
      SIGRB=1.E-04
      IF(L.EQ.0)GO TO 120
      IF(NTERMS.LE.0)GO TO 110
      DO 100 I=1,NTERMS
      J=M(I)-1
      RATEF=RATEF+FLOAT(J+1)*AT(I)*TIMEF**J
      RATEB=RATEB+FLOAT(J+1)*AT(I)*TIMEB**J
      SIGRF=SIGRF+(FLOAT(J+1)*SIGAT(I)*TIMEF**J)**2
      SIGRB=SIGRB+(FLOAT(J+1)*SIGAT(I)*TIMEB**J)**2
  100 CONTINUE
C
      SIGRF=SIGRF-1.E-04
      SIGRB=SIGRB-1.E-04
      IF(SIGRF.LT.0.1E-04)SIGRF=0.1E-04
      IF(SIGRB.LT.0.1E-04)SIGRB=0.1E-04
C
  110 SIGRF=SQRT(SIGRF)
      SIGRB=SQRT(SIGRB)
C
C  EVALUATE PRECISION
C
  120 PRECFF=ABS(FACTF/SIGFF)
      PRECFB=ABS(FACTB/SIGFB)
      PRECRF=ABS(RATEF/SIGRF)
      PRECRB=ABS(RATEB/SIGRB)
      RETURN
      END
```

```
$ASSM
FACFUN PROG
$FORT
$TRGT 16
      SUBROUTINE FACFUN(W,S,I)
C  SUBROUTINE TO CALCULATE DESIRED FUNCTION OF FACTOR.
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES.
C
C  NO CHANGE IN INPUT
C
      IF(I.EQ.0)RETURN
C
      GO TO(10,20,30),I
C
C  FUNCTION IS LOG(W)
C
   10 S=ABS(S/W)
      W=ALOG(W)
      RETURN
C
C  FUNCTION IS W**2
C
   20 S=ABS(S*2.*W)
      W=W**2
      RETURN
C
C  FUNCTION IS 1/W
C
   30 S=ABS(S/(W**2))
      W=1./W
      RETURN
      END
```

```
$ASSM
FACSET PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE FACSET(I,N1,N2,N3,N4,NSTOP)
C
C ****************************************************************
C  SUBROUTINE TO SET FACTOR LEVELS FROM INITIAL VALUES, INTERVALS
C    AND VECTORS
C
C  NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C    COMMON /FACTR/
C    COMMON /IO/
C    COMMON /STEPS/
C    SETPAR
C
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /STEPS/
C
      DIMENSION F(4),TOPF(4),DESIGN(4,16),FDER(4)
      EQUIVALENCE (F(1),DUMP(800)),(TOPF(1),DUMP(804))
     &,(DESIGN(1,1),DUMP(808)),(NFVEC,DUMP(872)),(FDER(1),DUMP(873))
C
C
      NTIME=0
      IF(N1.NE.0)GO TO 15
      DO 10 J=1,4
      F(J)=F(J) + DESIGN(J,I)
      IF (F(J).GT.TOPF(J)) NSTOP = 6
```

```
$ASSM
BAKEIT PROG
$FORT
$TRGT 16
C   ****************************************************************
C
      SUBROUTINE BAKEIT(N,NTHCP,BEGINW,FINALW)
C
C   ****************************************************************
C
C   SUBROUTINE TO PRESET MIDAS AND PRECONDITION SAMPLE BY A BAKEOUT
C
C   NEEDS COMMON BLOCKS /IO/ AND /MIDASR/
C
C   COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C   COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C   COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
      DIMENSION NTYME(3)
      EQUIVALENCE (NTYME(1),DUMP(685))
C
C   INITIALISE MIDAS IF REQUIRED
C
      OUT(1)=37
      OUT(2)=X'23'
      DO 210 I=3,20
  210 OUT(I)='  '
  455 WRITE(NTTYO,430)
  430 FORMAT(' SET INITIAL CONDITIONS ON MIDAS, Y OR N?')
      READ(NTTYI,440)NA
  440 FORMAT(A1)
      IF(NA.EQ.'N')GO TO 460
      WRITE(NTTYO,451)
  451 FORMAT(' GIVE COMMAND, 36 CHARACTERS MAXIMUM'
     &/' SHARP SIGN WILL BE ADDED BY COMPUTER')
      READ(NTTYI,420)JUNK2
      WRITE(NTTYO,423)OUT
  420 FORMAT(18A2)
  423 FORMAT(I6,2X,19A2)
      CALL TRAMID(OUT(2),19,REPLY(2),19,0.)
      GO TO 455
  460 CONTINUE
C
C   SET TIME
```

```
**FMIDAS MIDAS CALLS TO INTEGRATE INTO FORTRAN
 TITLE CALL DEFINITION
 TARGT 16
 NOSQZ
 WIDTH 132
 LCNT  50
 SPACE 7
*******************************************************************
*
*  CALLS FROM BASIC:
*
*   CALL 1,U,O$,I$,L OR T$,S
*
*   U = PASLA UNIT NUMBER
*   O$ = OUTPUT STRING
*   I$ = MINPUT STRING
*   L  = MAXIMUM LENGTH OF MINPUT STRING
*   OR
*   T$ = NEW TERMINATION CHARACTER FOR MINPUT STRING
*        L DEFAULTS TO 255, T$ DEFAULTS TO CARRIAGE RETURN
*   S  = STATUS RETURNED WHERE:
*       -1 IS BAD PASLA UNIT NUMBER
*        0 IS NO ERRORS
*        1 IS PREMATURE TERMINATION
*        2 IS DEVICE UNAVAILABLE
*        4 MEANS MINPUT WAS RECEIVED AFTER MINPUT TERMINATION
*        8 IS OPERATOR INTERVENTION
*       16 IS FRAMING ERROR
*       32 IS PARITY ERROR
*       64 IS PASLA OVERFLOW ERROR
*
*******************************************************************
 SPACE 7
**********************************************************************
*
*   REGISTER USE
*
*   0 - WORKING REGISTER, INTERRUPT LEVEL
*   1 - USER AREA POINTER, BEGINNING ADDRESS
*   2 - WORKING REGISTER, INTERRUPT LEVEL
*   3 - WORKING REGISTER, USER INDEX
*   4 - NUMBER OF CHARACTERS LEFT TO BE SENT FROM O$
*   5 - ADDRESS OF NEXT CHARACTER OF I$
*   6 - TERMINATION CHARACTER OF I$
*   7 - MAXIMUM LENGTH OF I$
*   8 - POINTER TO PASLA TYPE 2 COMMAND
*   9 - RECEIVER DEVICE CODE
*  10 - TRANSMITTER DEVICE CODE
*  11 - ADDRESS OF NEXT CHARACTER OF O$
*  12 - ADDRESS OF BEGINNING OF I$
*  13 - ADDRESS OF RETURNED STATUS WORD
*  14 - RETURN ADDRESS TO BASIC
*  15 - ADDRESS OS CALL PARAMETER LIST
*
*  THIS IS CMIDAS AS CHANGED FOR FORTRAN
*  THE CHANGES WERE SUGGESTED BY BILL HALL, NBS BOULDER.
```

```
   10 CONTINUE
      IF (NSTOP.EQ.6) RETURN
   15 CALL SETPAR(F(1),1,N2,NTHCP,TC)
      CALL SETPAR(F(2),2,N3,NTHCP,TC)
      CALL SETPAR(F(3),3,N3,NTHCP,TC)
      CALL SETPAR(F(4),4,N4,NTHCP,TC)
      IF(NPRINT.GE.3)WRITE(NPRINT,20)F
      IF(NFILEB.GE.6)WRITE(NFILEB,20)F
   20 FORMAT(//' FACTOR LEVELS SET TO',4F10.3//)
      IF(N2+N3+N4.EQ.0)GO TO 200
      N1=1
      IF(NFILEB.GT.6)WRITE(NFILEB,30)N1,N2,N3,N4,LOOP,PLAT
     &,NFVEC,F,TOPF,((DESIGN(J,K),J=1,4),K=1,NFVEC)
      IF(NPRINT.GE.3)WRITE(NPRINT,30)N1,N2,N3,N4
     &,LOOP,NPLAT
     &,NFVEC,F,TOPF,((DESIGN(J,K),J=1,4),K=1,NFVEC)
   30 FORMAT(' STOPPED IN FACSET BECAUSE OF ERRORS IN SETTING FACTORS'
     &/' N1=',I3
     &/' ERROR FLAGS ARE',3I3
     &/' LOOP NUMBER    ',I3
     &/' PLATEAU NUMBER ',I3
     &/' FACTOR LEVELS',4E14.6
     &/' FACTOR STEPS ',4E14.6
     &/' FACTOR VECTORS',(/4E14.6))
      NTIME=NTIME+1
      IF(NTIME.LT.3)GO TO 15
      PAUSE 20
      GO TO 15
  200 N1=0
      RETURN
      END

$ASSM
FCTN PROG
$FORT
$TRGT 16
C
C     ******************************************************************
C
      FUNCTION FCTN (X,I,J,JTERMS,NPTS)
C
C     ******************************************************************
C
C  FUNCTION TO CALCULATE FITTED VALUES FOR REGRES SUBROUTINE
C
C
      DIMENSION X(NPTS),JTERMS(J)
      JEXP=JTERMS(J)
      FCTN=X(I)**JEXP
      RETURN
      END
```

```
*****************************************************************
 SPACE 3
CMDTB DC X'262E' PASLA 26
 DC X'286E' PASLA 28
 DC X'246E' PASLA 24
ECMDTB DO 3 REPEAT NEXT LINE FOR EACH PASLA DEFINED
 DC 0,0,0,P,R,G,*+4,I,U,0,V,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,P,R,G,*-50,0L
 TITLE RELEASE COMMAND FOR MIDAS UNIT
*
*
*  FORTRAN CHANGE
*
RELESE STM 0,RS
 LHI 14,RETAD
 AIS 15,2
 LH 4,0(15) GET THE DEVICE CODE
*
*
* RELESE LH 4,0(15) GET THE DEVICE CODE
*
*
*
 LH 10,0(4)
 SHI 10,X'4200' REMOVE EXPONENT
 LHI 8,CMDTB
 LHI 3,ECMDTB
RSRCH LB 9,0(8) GET PASLA NUMBER IN TABLE
 CHR 9,10
 BZS RFUNIT PASLA IN TABLE?
 AIS 8,2 NO, KEEP LOOKING
 AHI 3,70
 CLHI 8,ECMDTB
 BLS RSRCH
 BR 14 CAN'T CLEAR, SO RETURN
RFUNIT LH 6,18(3)
 CHR 1,6 ONLY USER CAN RELEASE
 BNZR 14 NOT THE USER
 XHR 0,0
 STH 0,18(3) RELEASE UNIT
 BR 14
 TITLE MIDAS DRIVER
*
*
*  FORTRAN CHANGE
*
MIDAS STM 0,RS
 LHI 14,RETAD
 AIS 15,2
 LH 13,8(15) STATUS WORD ADDRESS
*
*
* MIDAS LH 13,8(15) STATUS WORD ADDRESS
*
*
 LH 12,4(15) ADDRESS OF I$
 XHR 3,3
```

```
*    TELEPHONE 8-323-4433 (FTS)
*
*  THE CALL IS:  CALL MIDAS(U,O,FI,FL,S)
*           OR   CALL MIDAS(U,O,FI,T,S)
*    WHERE U,FL AND S ARE FLOATING POINT NUMBERS
*    AND O AND FI ARE HOLLERITH ARRAYS.
*  TO RELEASE THE UNIT ASSIGNED TO MIDAS:  CALL RELESE(U)
*    WHERE U IS THE UNIT NUMBER.
*    O IS THE STRING TO OUTPUT TO MIDAS
*    FI IS THE INPUT STRING
*    FL IS THE MAXIMUM LENGTH OF THE INPUT STRING
*    T IS AN OPTIONAL HOLLERITH CHARACTER FOR TERMINATION
*    S IS A STATUS WORD (WHICH HAS TO BE CHECKED FOR TRANSMISSION
*    COMPLETE)
*
*
*
*
***********************************************************************
 TITLE SUBROUTINE CONSTANTS
*
*
*  FORTRAN ENTRIES
*
 ENTRY MIDAS
 ENTRY RELESE
*
*
MMODE EQU X'8E' USER'S MMODE FLAG IN BASIC
*
*   THE FOLLOWING DEFINITION'S ARE NEEDED IN THE DC BELOW
*
G EQU X'D100' LOAD MULTIPLE INSTRUCTION
P EQU X'D009' STORE MULTIPLE INSTRUCTION
I EQU X'0300' BRANCH TO REGISTER 0
OL EQU X'0302' BRANCH TO REGISTER 2
*
*
*  FORTRAN CHANGE
RS DS 32
*
*
R DS 32 MIDAS INTERRUPT REGISTER SAVE AREA
 SPACE 7
****************************************************************
*
*   THIS TABLE SHOULD BE TAILORED TO THE INDIVIDUAL
*   SYSTEM WITH WHICH IT WILL BE USED
*
*   THE FIRST BYTE OF EACH CONSTANT IS THE PASLA
*   DEVICE CODE, AND THE SECOND BYTE IS THE
*   TYPE 2 COMMAND TO BE ASSOCIATED WITH IT.
*
*   REPEAT COUNT ON THE DO AT ECMDTB MUST CORRESPOND TO
*   THE NUMBER OF PASLAS DEFINED IN THE CMDTB TABLE.
*
```

```
*************************************************************************
 SPACE 3
CMDTB DC X'262E' PASLA 26
 DC X'286E' PASLA 28
 DC X'246E' PASLA 24
ECMDTB DO 3 REPEAT NEXT LINE FOR EACH PASLA DEFINED
 DC 0,0,0,P,R,G,*+4,I,U,0,V,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,P,R,G,*-50,0L
 TITLE RELEASE COMMAND FOR MIDAS UNIT
*
*
*  FORTRAN CHANGE
*
RELESE STM 0,RS
 LHI 14,RETAD
 AIS 15,2
 LH 4,0(15) GET THE DEVICE CODE
*
*
* RELESE LH 4,0(15) GET THE DEVICE CODE
*
*
*
 LH 10,0(4)
 SHI 10,X'4200' REMOVE EXPONENT
 LHI 8,CMDTB
 LHI 3,ECMDTB
RSRCH LB 9,0(8) GET PASLA NUMBER IN TABLE
 CHR 9,10
 BZS RFUNIT PASLA IN TABLE?
 AIS 8,2 NO, KEEP LOOKING
 AHI 3,70
 CLHI 8,ECMDTB
 BLS RSRCH
 BR 14 CAN'T CLEAR, SO RETURN
RFUNIT LH 6,18(3)
 CHR 1,6 ONLY USER CAN RELEASE
 BNZR 14 NOT THE USER
 XHR 0,0
 STH 0,18(3) RELEASE UNIT
 BR 14
 TITLE MIDAS DRIVER
*
*
*  FORTRAN CHANGE
*
MIDAS STM 0,RS
 LHI 14,RETAD
 AIS 15,2
 LH 13,8(15) STATUS WORD ADDRESS
*
*
* MIDAS LH 13,8(15) STATUS WORD ADDRESS
*
*
 LH 12,4(15) ADDRESS OF I$
 XHR 3,3
```

```
 STB 3,-1(12) CLEAR MINPUT STRING
 STH 3,0(13) CLEAR STATUS WORD
 STH 3,2(13)
 LH 4,0(15) GET DEVICE CODE
 LH 10,0(4)
 SHI 10,X'4200' REMOVE EXPONENT
 LHI 8,CMDTB
 LHI 3,ECMDTB
SEARCH LB 9,0(8) GET PASLA NUMBER
 CHR 9,10
 BZS RTUNIT PASLA IN TABLE?
 AIS 8,2 NO, KEEP LOOKING
 AHI 3,70
 CLHI 8,ECMDTB
 BLS SEARCH
NONO LHI 3,X'C110' PASLA NOT AVAILABLE--RETURN -1
 STH 3,0(13) DEVICE CODE ERROR
 BR 14
RTUNIT LH 6,18(3) UNIT ALREADY IN USE?
 BZS GRAB
 CHR 1,6 YES CHECK FOR RIGHT USER
 BZS FOUND
 BS NONO WRONG USER--PASLA NOT AVAILABLE
GRAB STH 1,18(3) GRAB THIS PASLA FOR THIS USER
FOUND AHR 10,10
 LH 7,X'D0'(10) SAVE VECTOR FOR RECEIVE
 STH 7,50(3)
 LHI 7,0(3) SET INTERRUPT POINTER
 STH 7,X'D0'(10)
 AIS 10,2
 LH 7,X'D0'(10) SAVE VECTOR FOR SEND
 STH 7,52(3)
 LHI 7,54(3) SET INTERRUPT POINTER
 STH 7,X'D0'(10)
 SRLS 10,1 SENDING DEVICE CODE
 LH 11,2(15) ADDRESS OF O$
 LH 4,6(15) MINPUT TERMINATION ADDRESS
 THI 4,1 STRING VARIABLE?
 BZS NEWLEN NO, MUST BE F.P. NUMBER
 LB 6,0(4) YES, LOAD TERMINATION CHARACTER
 LHI 7,X'FF' AND DEFAULT LENGTH
 B ISET
NEWLEN LH 6,0(4) GET LENGTH
 SRL 6,8 SPLIT F.P. WORD
 SRLS 7,8
 CLHI 6,X'43' >255
 BNLS DEFALT USE 255
 SHI 6,X'41'
 BNCS *+4 <1
 XHR 7,7
 BNZS *+4
 SRLS 7,4
 BS *+6
DEFALT LHI 7,X'FF'
 LIS 6,13 TERMINATE ON CARRIAGE RETURN
ISET LHR 5,12 SET MINPUT POINTER
```

```
 XHR 4,4 CLEAR OUTPUT COUNT
 STM 3,22(3) SET INTERRUPT REGISTERS
 STH 4,48(3) CLEAR ERROR CODE
 OC 9,1(8) ISSUE TYPE 2 COMMAND TO PASLA
 SSR 9,2 MAKE SURE DEVICE IS AVAILABLE
 BFFS 2,CLRIN
 LHI 15,X'4130' NOT, SO SET ERROR
 B MRESTOR
CLRIN RDR 9,2 CLEAR MINPUT REGISTER
 OC 9,WRTRD+1 SET MINPUT
 OC 10,WRTRD SET OUTPUT
 LB 4,-1(11) SET OUTPUT COUNT
 STB 4,25(3)
 SINT 0(10) START OUTPUT
WAIT LH 15,43(3) CHECK FOR TERMINATION FLAG
 THI 15,2
*
*
*  FORTRAN CHANGES
*
 BZ NOESC
*
*
* BZ STATOK DEVICE NOT UNAVAILABLE
*
*
PREMAT OHI 15,X'4201' PREMATURE TERMINATION
MRESTOR STH 15,0(13) RETURN STATUS
 LH 15,52(3) RETURN AFTER MRESTORING VECTORS
 AHR 9,9
 STH 15,X'D2'(9)
 LH 15,50(3)
 STH 15,X'D0'(9)
 BR 14
*
*
*  FORTRAN CHANGES
*
*STATOK LH 15,MMODE(1) GET USER MMODE FLAG
* SRLS 15,1
* BCS NOESC USER ENTERED ESCAPE?
 LIS 15,8
 B PREMAT
NOESC LH 7,30(3)
 LH 4,24(3) OUTPUT DONE?
 BNZ WAIT
 LB 15,-1(12) IF SO, CHECK INPUT
 SHR 15,7
 BNZ WAIT
 LH 15,48(3)
 BZ MRESTOR
 OHI 15,X'4200'
 B MRESTOR
*
*
*  FORTRAN CHANGES
```

```
*
RETAD LM 0,RS
 AH 15,0(15)
 BR 15
*
*
 TITLE INPUT INTERRUPT PROCESSOR
U LB 0,-1(12) GET LENGTH OF I$
 CHR 0,7 INPUT EXPECTED?
 BLS MINPUT
 LIS 0,4 NO
 OH 0,48(3)
 STH 0,48(3) IGNORE MINPUT
 SSR 9,2 PAY DUES TO THE PALS
 RDR 9,0
 B INRET
MINPUT SSR 9,2
 BTC 15,IERR ALL STATUS BITS SHOULD BE OFF
EINPUT RDR 9,2 MINPUT
 CHR 2,6 TERMINATION CHARACTER?
 BNZS SAVIT
 STH 0,30(3) DONE
 BS INRET
SAVIT STB 2,0(5)
 AIS 0,1
 STB 0,-1(12) INCREASE STRING POINTER
 AIS 5,1 AND MINPUT POINTER
 STH 5,26(3)
INRET LM 0,R
 LPSW 0(3)
IERR THI 2,2 NO CARRIER?
 BNZS NODEV
 THI 2,X'FB' NO RESPONSE?
 BNZS STATCK
NODEV LIS 0,2 SET DEVICE UNAVAILABLE
 OH 0,48(3)
 STH 0,48(3)
 BS INRET
STATCK NHI 2,X'E0' CHECK REST OF STATUS
 BZ INRET
 SRLS 2,1
 OH 2,48(3)
 STH 2,48(3) SAVE ERROR 48(3)
 B EINPUT
 TITLE OUTPUT INTERRUPT PROCESSOR
V SSR 10,2
 BFC 15,SENDIT
 BTFS 2,CANTTX NO CARRIER?
 THI 2,X'FB' NO RESPONSE
 BNZS TXSTAT
CANTTX LIS 0,2 CAN'T TRANSMIT--NO DEVICE
 OH 0,48(3)
 STH 0,48(3)
 BS OUTRET
TXSTAT THI 2,X'40' CLEAR TO SEND?
 BNZS CANTTX
```

```
 SSR 10,2 BUSY FLAG?
 BFFS 8,SENDIT
OUTRET LM 0,R
 LPSW 54(3) NO.RETURN
SENDIT LHR 4,4 MORE TO SEND?
 BZS OUTRET
 SIS 4,1 COUNT OUTPUT
 STH 4,24(3)
 WD 10,0(11) SEND IT
 AIS 11,1 BUMP POINTER
 STH 11,38(3)
 BS OUTRET
 TITLE MIDAS DRIVER SYMBOL TABLE
WRTRD DC X'6351' FASLA WRITE AND READ COMMANDS
 LIST
 END
```

```
$ASSM
INIT PROG
$FORT
$TRGT 16
C ******************************************************************
C
      SUBROUTINE INIT
C
C ******************************************************************
C  SUBROUTINE TO ASSIGN VALUES TO VARIABLES IN LABELLED COMMON
C  NEEDS ALL COMMON BLOCKS
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /BIAS/
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C  COMMON /DERIVE/
C
      EQUIVALENCE (PRECE,DUMP(131))
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
C  COMMON /PARAMS/
C
      DIMENSION A01(8),A02(8),SIGA01(8),SIGA02(8)
```

```
     &,A1(4,8),A2(4,8),SIGA1(4,8),SIGA2(4,8)
     &,FACT1(2,8),SIGF1(2,8),RATE1(2,8),SIGR1(2,8)
     &,FACT2(2,8),SIGF2(2,8),RATE2(2,8),SIGR2(2,8)
     &,TIMEF(2),TIMEB(2)
      EQUIVALENCE (A01(1),DUMP(200)),(A02(1),DUMP(208))
     &, (SIGA01(1),DUMP(216)),(SIGA02(1),DUMP(224))
     &, (A1(1,1),DUMP(232)),(A2(1,1),DUMP(264))
     &, (SIGA1(1,1),DUMP(296)),(SIGA2(1,1),DUMP(328))
      EQUIVALENCE (FACT1(1,1),DUMP(360)),(SIGF1(1,1),DUMP(376))
     &,(RATE1(1,1),DUMP(392)),(SIGR1(1,1),DUMP(408))
     &,(FACT2(1,1),DUMP(424)),(SIGF2(1,1),DUMP(440))
     &,(RATE2(1,1),DUMP(456)),(SIGR2(1,1),DUMP(472))
     &,(KOUNT1,DUMP(488)),(KOUNT2,DUMP(489))
      EQUIVALENCE (TIMEF(1),DUMP(500)),(TIMEB(1),DUMP(502))
     &,(DAY,DUMP(504)),(HDTIME,DUMP(505))
     &,(SKIP,DUMP(506)),(RATMIN,DUMP(507)),(RATMAX,DUMP(508))
     &,(TJUMP,DUMP(509))
C
C  COMMON /PLAT/
C
      DIMENSION RCHI(8)
      EQUIVALENCE (LPLAT,DUMP(510)),(NELAPS,DUMP(511))
     &,(NFIRST,DUMP(512)),(NCHI,DUMP(513)),(NSTOP,DUMP(514))
     &,(START,DUMP(515)),(RCHI(1),DUMP(516))
     &,(NOTHER,DUMP(524)),(KSTORE,DUMP(525))
     &,(ENDSIG,DUMP(526))
     &,(TOIM4,DUMP(527)),(FACTIM,DUMP(528)),(EQUIL,DUMP(529))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C  COMMON /STEPS/
C
      DIMENSION F(4),DELF(4),DESIGN(4,16),FDER(4)
      EQUIVALENCE (F(1),DUMP(800)),(DELF(1),DUMP(804))
     &,(DESIGN(1,1),DUMP(808)),(NFVEC,DUMP(872)),(FDER(1),DUMP(873))
C
C  COMMON /WHEN/
C
      DIMENSION TPLAT1(2),TPLAT2(2),NTYME(3)
      EQUIVALENCE (BEGIN,DUMP(880)),(TPLAT1(1),DUMP(881))
     &,(TPLAT2(1),DUMP(883)),(NTYME(1),DUMP(885))
C
C
C
C
C  INITIALISE COMMON VARIABLES
C
      DO 1 I=1,900
    1 DUMP(I)=0.
      NUM(1)=X'30'
      NUM(2)=X'31'
      NUM(3)=X'32'
      NUM(4)=X'33'
```

```
      NUM(5)=X'34'
      NUM(6)=X'35'
      NUM(7)=X'36'
      NUM(8)=X'37'
      NUM(9)=X'38'
      NUM(10)=X'39'
      DAY=0.
      SKIP=30.
      HDTIME=0.
      NSTOP=0
      NTHCP='E'
      TSCALE=243.73
C
C  GENERAL LOOP OVER PARAMETERS
C
      DO 10 I=1,8
      KANDU(I)=0
      IFUNCT(I)=0
      IFDER(I)=0
      INPUT(I)=I-1
      SCALE(I)=1.
      NRANGE(I)=5
      NREADS(I)=3
      NTERMS(I)=0
      CHTEST(I)=1000.
      PTIME(I)=300.
      PRECF(I)=100.
      PRECR(I)=50.
   10 CONTINUE
C
C  CHANGE THE EXCEPTIONS
C
      KANDU(1)=1
      KANDU(2)=2
      IFDER(1)=1
      SCALE(3)=0.01
      NRANGE(1)=3
      NRANGE(2)=3
      NTERMS(1)=2
      NTERMS(2)=1
      CHTEST(1)=2000.
      CHTEST(3)=99999.
C
C  SET THE REMAINING VALUES
C
      PRECE=25.
      NFACTS=2
      NRATES=1
      TIMLIM=600.
      NBADS=8
      EQUIL=200.
      INTVAL=200
      ENDSIG=9.
      RATMIN=15.
      RATMAX=45.
      TJUMP=16.
```

```
      DO 20 I=1,5
   20 M(I)=I
      NTTYI=5
      NTTYO=4
      NPRINT=3
      NFILEA=7
      NFILEB=8
      NFILED=1
      NFILEE=6
C
      RETURN
      END
```

```
$ASSM
MATINV PROG
$FORT
$TRGT 16
C     ****************************************************************************
C
      SUBROUTINE MATINV (ARRAY,NORDER,DET)
C
C     ****************************************************************************
C  SUBROUTINE FOR MATRIX INVERSION
C
      DOUBLE PRECISION ARRAY,AMAX,SAVE
      DIMENSION ARRAY(5,5),IK(5),JK(5)
      DET=1.
      DO 190 K=1,NORDER
C
C  FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX
C
          AMAX=0.
10        DO 30 I=K,NORDER
              DO 30 J=K,NORDER
              IF (DABS(AMAX)-DABS(ARRAY(I,J))) 20,20,30
20            AMAX=ARRAY(I,J)
              IK(K)=I
              JK(K)=J
30            CONTINUE
C
C  INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
C
          IF (AMAX) 50,40,50
40        DET=0.
          GO TO 260
50        I=IK(K)
          IF (I-K) 10,80,60
60        DO 70 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=ARRAY(I,J)
70            ARRAY(I,J)=-SAVE
80        J=JK(K)
          IF (J-K) 10,110,90
90        DO 100 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=ARRAY(I,J)
100           ARRAY(I,J)=-SAVE
C
C  ACCUMULATE ELEMENTS OF INVERSE MATRIX
C
110       DO 130 I=1,NORDER
              IF (I-K) 120,130,120
120           ARRAY(I,K)=-ARRAY(I,K)/AMAX
130           CONTINUE
          DO 160 I=1,NORDER
              DO 160 J=1,NORDER
              IF (I-K) 140,160,140
140           IF (J-K) 150,160,150
150           ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
```

```
160           CONTINUE
         DO 180 J=1,NORDER
              IF (J-K) 170,180,170
170           ARRAY(K,J)=ARRAY(K,J)/AMAX
180           CONTINUE
         ARRAY(K,K)=1./AMAX
190      DET=DET*AMAX
C
C  RESTORE ORDERING OF MATRIX
C
      DO 250 L=1,NORDER
         K=NORDER-L+1
         J=IK(K)
         IF (J-K) 220,220,200
200      DO 210 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=-ARRAY(I,J)
210           ARRAY(I,J)=SAVE
220      I=JK(K)
         IF (I-K) 250,250,230
230      DO 240 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=-ARRAY(I,J)
240           ARRAY(I,J)=SAVE
250      CONTINUE
260   RETURN
      END
```

```
$ASSM
OVERLY PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE OVERLY(NPROG,NUNIT,NREWIN,NTTYO)
C
C ****************************************************************
C  SUBROUTINE TO READ IN OVERLAY SEGMENT FROM UNIT NUNIT
C    TRY 3 TIMES IF OVERLAY IS AT BEGINNING OF FILE
C
      INTEGER*2 NPROG(3)
   30 NTIME=1
C
C  REWIND INPUT UNIT IF APPROPRIATE, READ IN OVERLAY SEGMENT
C
   10 IF(NREWIN.EQ.0)REWIND NUNIT
      CALL IFETCH(NPROG,NUNIT,ISTAT)
      IF(ISTAT.EQ.0)RETURN
      NTIME=NTIME+1
      IF(NTIME.LE.3.AND.NREWIN.EQ.0)GO TO 10
C
C  ERROR ROUTE
C
      WRITE(NTTYO,20)NPROG,NUNIT
   20 FORMAT(' CANNOT READ IN OVERLAY ',3A2,' FROM LOGICAL UNIT',I3)
C
C  REPEAT AFTER =CONTINUE= GIVEN BY OPERATOR
C
      PAUSE
      GO TO 30
      END
```

```
$ASSM
PLATTO PROG
$FORT
$TRGT 16
C ***********************************************************
C
      SUBROUTINE PLATTO(AO,SIGAO,A,SIGA,KOUNT,TF,STF,TYME,NERROR,NPLAT
     &,BEGINW)
C
C ***********************************************************
C  SUBROUTINE TO COLLECT WEIGHT DATA DURING A PLATEAU IN SOME
C    FACTOR OR COMBINATION OF FACTORS (WHICH MAY INCLUDE SOME FUNCTION
C    OF THE RATE OF WEIGHT LOSS) AND FIT A POLYNOMIAL
C    OF DEGREE NTERMS(K) FOR FACTOR K AS A FUNCTION OF TIME.
C  8 FACTORS MAXIMUM, 250 POINTS PER PLATEAU MAXIMUM (NOT COUNTING
C    PARTS DISCARDED BECAUSE OF INSTABILITY IN FACTOR LEVELS).
C
C  NEEDS THE FOLLOWING COMMON BLOCKS, FUNCTIONS AND SUBROUTINES:
C    COMMON /BIAS/
C    COMMON /FACTR/
C    COMMON /IO/
C    COMMON /MIDASR/
C    COMMON /PARAMS/
C    COMMON /PLAT/
C    COMMON /REGRS/
C    COMMON /WHEN/
C    BSET
C    FACFUN
C    REGRES
C    RPAR
C    TIMER
C
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /BIAS/
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
```

```
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
C  COMMON /PARAMS/ (PART OF)
C
      EQUIVALENCE (SKIP,DUMP(506))
C
C  COMMON /PLAT/
C
      DIMENSION RCHI(8)
      EQUIVALENCE (LPLAT,DUMP(510)),(NELAPS,DUMP(511))
     &,(NFIRST,DUMP(512)),(NCHI,DUMP(513)),(NSTOP,DUMP(514))
     &,(START,DUMP(515)),(RCHI(1),DUMP(516))
     &,(NOTHER,DUMP(524)),(KSTORE,DUMP(525))
     &,(ENDSIG,DUMP(526)),(TOIM4,DUMP(527)),(FACTIM,DUMP(528))
     &,(EQUIL,DUMP(529))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C  COMMON /WHEN/
C
      DIMENSION TPLAT1(2),TPLAT2(2),NTYME(3)
      EQUIVALENCE (BEGIN,DUMP(880)),(TPLAT1(1),DUMP(881))
     &,(TPLAT2(1),DUMP(883)),(NTYME(1),DUMP(885))
C
C
      LOGICAL ITEST
C
C
      DIMENSION AO(8),SIGAO(8),A(4,8),SIGA(4,8),TF(5,8),STF(5,8)
     &,TYME(5),TZERO(5),TEMP(5),STEMP(5)
     &,ATP(2),SIGATP(2)
      DIMENSION NWORD(8)
      DATA NWORD/'EB','TC','PR','F1','F2'
     &,'R1','R2','XX'/
C
C  INITIALISE
C
      LOOP=0
      NFIRST=0
      NSTOP=0
      NERROR=0
```

```
      NELAPS=0
      KSTORE=0
      DO 4 J=1,5
      TYME(J)=0.
      TEMP(J)=0.
      STEMP(J)=1.E-09
      DO 4 I=1,NFACTS
      TF(J,I)=0.
      STF(J,I)=1.E-09
      RCHI(I)=0.
    4 CONTINUE
C  MAKE KOUNT=-4 TO COLLECT 4 DATA POINTS QUICKLY TO USE IN
C     STEADINESS CHECKS WITHOUT BIASING THE DATA WITH
C     5 CLOSE POINTS IN THE UNSTEADY PART AT THE BEGINNING
C     OF THE PLATEAU
      KOUNT=-4
C
C  UNLESS NO STEADINESS CHECKS WILL BE MADE
C
      IF (NCHI.EQ.0) KOUNT = 0
C
C  PRINT HEADING
C
      IF(NPRINT.GE.3)WRITE(NPRINT,200)NPLAT
      IF(NFILEB.GE.6)WRITE(NFILEB,200)NPLAT
  200 FORMAT(' ****** MEASUREMENT FOR PLATEAU NUMBER',I3,' ******')
      IF(NPRINT.GE.3)WRITE(NPRINT,201)
      IF(NFILEB.GT.6)WRITE(NFILEB,201)
  201 FORMAT(/'  NO. F I        READING           SIGMA RD         TIME'
     &,' BAD    RCHI  TESTCH'//)
      REWIND NFILED
      LOOP=LOOP+1
C  WAIT UNTIL EQUILIBRATION TIME HAS ELAPSED
  830 START=TIMER(NTYME)-BEGIN
      IF((START-FACTIM).LT.EQUIL)GO TO 830
C
C
      ENTRY PLATCO
C
C
      IF(NFIRST.NE.0.AND.NPRINT.GE.3)WRITE(NPRINT,620)NPLAT
      IF(NFIRST.NE.0.AND.NFILEB.GT.6)WRITE(NFILEB,620)NPLAT
  620 FORMAT(/' ******   PLATEAU',I3,' CONTINUED    ******'/)
      NFIRST=1
C
C  BEGIN READ LOOP
C
  570 CONTINUE
C
C  MOVE ANY PREVIOUS READINGS DOWN IN SMALL ARRAY STORAGE LISTS
C
      DO 30 J=1,4
      TYME(J)=TYME(J+1)
      DO 30 I=1,NFACTS
      TF(J,I)=TF(J+1,I)
   30 STF(J,I)=STF(J+1,I)
```

```
C
      KOUNT=KOUNT+1
C
C  DELAY LOOP
C
      IF(KOUNT.LE.1)GO TO 26
   27 T=TIMER(NTYME)-BEGIN
      IF(T-TYME(5).LT.SKIP)GO TO 27
C
C  READ TIME, WEIGHT AND FACTOR LEVELS ALL TOGETHER AS RAPIDLY AS
C  POSSIBLE
C  THIS IS TIME FROM BEGINNING OF EXPERIMENT
C
   26 TYME(5)=TIMER(NTYME)-BEGIN
      IWT=0
      DO 20 J=1,NFACTS
      I=KANDU(J)
      IF(INPUT(I).EQ.0)IWT=J
      CALL RPAR(TF(5,J),STF(5,J),INPUT(I),NRANGE(I),SCALE(I)
     &,NREADS(I),BIASV(I),NERROR)
      IF(NERROR.EQ.0)GO TO 22
      IF(NPRINT.GE.3)WRITE(NPRINT,23)NERROR
      IF(NFILEB.GT.6)WRITE(NFILEB,23)NERROR
   23 FORMAT(' NERROR VALUE OF',I2,' IN PLATTO *****')
   22 STF(5,J)=STF(5,J)/SQRT(FLOAT(NREADS(I)))
      IFAC=IFUNCT(I)
      CALL FACFUN(TF(5,J),STF(5,J),IFAC)
C
C  STORE INITIAL WEIGHT IF NEEDED
C
      IF(NPLAT.NE.1)GO TO 20
      IF(ABS(BEGINW).GE.1.0E-6)GO TO 20
      IF(INPUT(I).NE.0.OR.LOOP.NE.1.OR.KOUNT.NE.-3)GO TO 20
      BEGINW=TF(5,J)
      IF(NFILEB.GT.6)WRITE(NFILEB,244)BEGINW
      IF(NPRINT.GE.3)WRITE(NPRINT,244)BEGINW
  244 FORMAT(/' &&&&&  INITIAL WEIGHT MEASURED IN PLATTO AS',F15.1
     &,' &&&&&'/)
   20 CONTINUE
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C  READ OTHER EXPERIMENTS HERE
C
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      IF(IWT.LE.0)GO TO 21
      IF(KOUNT.LT.2)GO TO 21
C
C  CHECK FOR END OF WEIGHT LOSS (WT2-WT1).LT.(ENDSIG*POOLED SIGMA)
C
      IF((TF(5,IWT)-TF(4,IWT))**2/(STF(5,IWT)**2+STF(4,IWT)**2)
     &.LT.ENDSIG)NSTOP=1
      IF(TF(5,IWT).LT.-0.159E+05)NSTOP=1
   21 CONTINUE
C
```

```
C  SET INITIAL TIME FOR THIS PLATEAU
C
      IF(KOUNT.LE.1)TPLAT2(1)=TYME(5)
C
C  CALCULATE CHI-SQUARE FOR SMALL ARRAYS IF ENOUGH READINGS
C    TO SEE IF FACTORS ARE AT EQUILIBRIUM
C
      NFLAG=0
      DO 48 K=1,NFACTS
      AO(K)=0.
      SIGAO(K)=1.E-09
      DO 48 J=1,2
      A(J,K)=0.
      SIGA(J,K)=1.E-09
   48 CONTINUE
      DO 40 K=1,NFACTS
      I=KANDU(K)
      JJ=INPUT(I)+1
      IF(NFILEA.GT.6)WRITE(NFILEA,101)NWORD(JJ),TF(5,K)
     &,STF(5,K),NREADS(I),TYME(5)
  101 FORMAT(2X,A2,2E14.7,I10,F10.0)
      RCHI(I)=0.
      IF(NCHI.EQ.0)GO TO 47
      IF(KOUNT.LE.6)GO TO 47
      DO 50 J=1,5
      TZERO(J)=TYME(J)-TYME(1)
      TEMP(J)=TF(J,K)
   50 STEMP(J)=STF(J,K)
      NTEMP=MIN0(NTERMS(I),2)
      CALL REGRES(TZERO,TEMP,STEMP,5,NTEMP,M,0,YFIT,ATO,ATP
     &,SIGATO,SIGATP,R,RMUL,RCHI(I),TEST)
      IF(RCHI(I).LT.CHTEST(I))GO TO 47
      CALL BSET(NFLAG,I)
C
      IF(NPRINT.GE.3)WRITE(NPRINT,610)I,INPUT(I),RCHI(I),CHTEST(I)
      IF(NFILEB.GT.6)WRITE(NFILEB,610)I,INPUT(I),RCHI(I),CHTEST(I)
  610 FORMAT(/' UNSTEADINESS IN FACTOR',I3,' ON INPUT',I3
     &/'  RCHI MEASURED =',F9.1,' TEST VALUE IS',F9.1
     &/' ACTUAL VALUES ARE'/'  TIME',3X,'READING',6X,'CALC',5X,'SIGMA')
      DO 810 J=1,5
      IF(NFILEB.GT.6)WRITE(NFILEB,820)TZERO(J),TEMP(J),YFIT(J),STEMP(J)
      IF(NPRINT.GE.3)WRITE(NPRINT,820)TZERO(J),TEMP(J),YFIT(J),STEMP(J)
  820 FORMAT(F6.0,3F10.1)
  810 CONTINUE
C
   47 IF(NPRINT.GE.3)WRITE(NPRINT,44)KOUNT,I,INPUT(I)
     &,TF(5,K),STF(5,K),NREADS(I),TYME(5),NELAPS
     &,RCHI(I),CHTEST(I)
      IF(NFILEB.GT.6)WRITE(NFILEB,44)KOUNT,I,INPUT(I)
     &,TF(5,K),STF(5,K),NREADS(I),TYME(5),NELAPS
     &,RCHI(I),CHTEST(I)
   44 FORMAT(I5,2I2,2E14.7,I3,F11.0,I4,2F8.1)
   40 CONTINUE
      TELAPS=TYME(5)-START
      IF(NSTOP.NE.0)GO TO 65
C
```

```
C  DO NOT SAVE INFORMATION WHEN FACTORS NOT AT EQUILIBRIUM
C
      IF(NFLAG.EQ.0)GO TO 60
      NELAPS=NELAPS+KOUNT+4
      KOUNT=-4
      IF (NCHI.EQ.0) KOUNT = 0
      REWIND NFILED
C
C  UPDATE CHTEST VALUES EVERY =NBAD= UNSTEADY READINGS
C
      NFIX=NELAPS/NBADS
      IF(NFIX.LE.KSTORE)GO TO 575
      KSTORE=KSTORE+1
C
C  UPDATE CHTEST AS 1.25*(LATEST RCHI VALUE)
C
C     IF(LOOP.GT.2)GO TO 100
C
      IF(NPRINT.GE.3)WRITE(NPRINT,70)
      IF(NFILEB.GT.6)WRITE(NFILEB,70)
   70 FORMAT(' NEW VALUES FOR CHTEST IN PLATTO'
     &/' FACTOR        OLD        NEW')
      DO 78 K=1,NFACTS
      I=KANDU(K)
      ITEST=BTEST(NFLAG,I)
      IF(ITEST)GO TO 576
      GO TO 78
  576 RCHI(I)=1.25*RCHI(I)
      IF(NPRINT.GE.3)WRITE(NPRINT,74)I,CHTEST(I),RCHI(I)
      IF(NFILEB.GT.6)WRITE(NFILEB,74)I,CHTEST(I),RCHI(I)
   74 FORMAT(I7,2F10.3)
      CHTEST(I)=RCHI(I)
   78 CONTINUE
C
C  CHECK WHETHER ALLOTTED TIME HAS ELAPSED
C
  575 IF(TELAPS.LT.TIMLIM)GO TO 570
C
C  RETURN ON ERROR BECAUSE ALL THE TIME HAS GONE
C    OR BECAUSE CHI VALUES HAVE BEEN UPDATED TWICE AND THERE ARE STILL
C    NO GOOD DATA POINTS
C
  100 NERROR=6
      IF(NPRINT.GE.3)WRITE(NPRINT,900)NERROR
  900 FORMAT(' NERROR VALUE OF',I2,' IN PLATTO')
      IF(NFILEB.GT.6)WRITE(NFILEB,900)NERROR
      GO TO 65
C
C  YES, CURRENT FACTOR LEVELS ARE AT EQUILIBRIUM, WRITE VALUES
C    ON TEMPORARY SCRATCH FILE =NFILED=
C
   60 IF(KOUNT.LE.0)GO TO 570
      WRITE(NFILED,600)TYME(5),(TF(5,I),STF(5,I),I=1,NFACTS)
  600 FORMAT(5E15.8)
C
C  IS THE EXPERIMENT UNDER  CONTROL?  CHECK EVERY 'INTVAL' POINTS
```

```
C
      IF(TELAPS.GT.TIMLIM)NSTOP=2
      IF(KOUNT.EQ.250)NSTOP=2
      IF(NSTOP.NE.0)GO TO 65
      KMOD=MOD(KOUNT,INTVAL)
      IF(KMOD.NE.0)GO TO 570
   65 TPLAT2(2)=TYME(5)
C
C  DISPLAY E BAL ON DVM SO THAT SUPPRESSION CAN BE CHANGED DURING OVERLAY
C    LOADING, RATE OF WEIGHT LOSS CAN CHECKED VISUALLY, ETC.
C
      OUT(1)=10
      OUT(2)=X'23'
      OUT(3)='K0'
      OUT(4)='MP'
      OUT(5)='00'
      OUT(6)='0 '
      DO 615 I=7,19
  615 OUT(I)='  '
      CALL TRAMID(OUT(2),19,REPLY(2),19,0.)
C
C
      IF (NSTOP.EQ.1) KOUNT = KOUNT-1
      RETURN
      END
```

```
$ASSM
PRECIS PROG
$FORT
$TRGT 16
C ***********************************************************************
C
      SUBROUTINE PRECIS(IFGO,NPLAT,PRECFF,PRECFB,PRECRF,PRECRB)
C
C ***********************************************************************
C  SUBROUTINE TO FIT POLYNOMIAL TO EXPERIMENTAL READINGS
C    AND ASSESS WHETHER PRECISION OF EXTRAPOLATED QUANTITIES IS ADEQUATE
C
C  NEEDS THE FOLLOWING COMMON BLOCKS, FUNCTIONS AND SUBROUTINES:
C    COMMON /FACTR/
C    COMMON /IO/
C    COMMON /PARAMS/
C    COMMON /REGRS/
C    COMMON /WHEN/
C    BSET
C    BTEST
C    EXTPOL
C    REGRES
C    RESIDS
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(2),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(120))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /PARAMS/
C
      DIMENSION A01(8),A02(8),SIGA01(8),SIGA02(8)
     &,A1(4,8),A2(4,8),SIGA1(4,8),SIGA2(4,8)
     &,FACT1(2,8),SIGF1(2,8),RATE1(2,8),SIGR1(2,8)
     &,FACT2(2,8),SIGF2(2,8),RATE2(2,8),SIGR2(2,8)
     &,TIMEF(2),TIMEB(2)
```

```
      EQUIVALENCE (A01(1),DUMP(200)),(A02(1),DUMP(208))
     &, (SIGA01(1),DUMP(216)),(SIGA02(1),DUMP(224))
     &, (A1(1,1),DUMP(232)),(A2(1,1),DUMP(264))
     &, (SIGA1(1,1),DUMP(296)),(SIGA2(1,1),DUMP(328))
      EQUIVALENCE (FACT1(1,1),DUMP(360)),(SIGF1(1,1),DUMP(376))
     &,(RATE1(1,1),DUMP(392)),(SIGR1(1,1),DUMP(408))
     &,(FACT2(1,1),DUMP(424)),(SIGF2(1,1),DUMP(440))
     &,(RATE2(1,1),DUMP(456)),(SIGR2(1,1),DUMP(472))
     &,(KOUNT1,DUMP(488)),(KOUNT2,DUMP(489))
      EQUIVALENCE (TIMEF(1),DUMP(500)),(TIMEB(1),DUMP(502))
     &,(DAY,DUMP(504)),(HDTIME,DUMP(505))
     &,(SKIP,DUMP(506))
C
C  COMMON /PLAT/
C
      EQUIVALENCE (NSTOP,DUMP(514))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C  COMMON /WHEN/
C
      DIMENSION TPLAT1(2),TPLAT2(2),NTYME(3)
      EQUIVALENCE (BEGIN,DUMP(880)),(TPLAT1(1),DUMP(881))
     &,(TPLAT2(1),DUMP(883)),(NTYME(1),DUMP(885))
C
C
C
      DIMENSION PRECFF(8),PRECFB(8),PRECRF(8),PRECRB(8)
C
      DIMENSION TYME(250),P(250),SP(250),PASS(16),NWORD(2)
      DATA NWORD/' YES',' NO'/
      DATA NR,NF/' R',' F'/
C
      IFGO=0
C
C
      IF(NPRINT.GE.3)WRITE(NPRINT,790)NPLAT
  790 FORMAT(/' CHECKS OF PRECISION FOR PLATEAU',I3
     &/' ****************************************************'/)
      IF(NFILEB.GT.6)WRITE(NFILEB,790)NPLAT
C
C  FIND MIDDLE OF PLATEAUS
C
      TMID1=(TPLAT1(1)+TPLAT1(2))/2.
      TMID2=(TPLAT2(1)+TPLAT2(2))/2.
      REWIND NFILED
C
C  READ FIRST FACTOR READINGS IN
C
      NFACT2=2*NFACTS
      DO 10 J=1,KOUNT2
      READ(NFILED,600)TYME(J),(PASS(I),I=1,NFACT2)
      P(J)=PASS(1)
```

```
      SP(J)=PASS(2)
  600 FORMAT(5E15.8)
C
C  PUT TIME ORIGIN IN MIDDLE OF PLATEAU
C
   10 TYME(J)=TYME(J)-TMID2
      K=1
C
C  OUTPUT WEIGHT LOSS IN THIS PLATEAU
C
      WTLOSS=ABS(P(KOUNT2)-P(1))
      RATE=WTLOSS/(TYME(KOUNT2)-TYME(1))
      IF(NFILEB.GT.6)WRITE(NFILEB,951)WTLOSS,RATE
      IF(NPRINT.GE.3)WRITE(NPRINT,951)WTLOSS,RATE
  951 FORMAT(/' WEIGHT LOSS FOR THIS PLATEAU =',E10.3,' MICROVOLTS'
     &/'  WITH AVERAGE RATE OF',E10.3,' MICROVOLTS PER SECOND'/)
      GO TO 20
C
C  CALCULATE REGRESSION FIT FOR VARIOUS FACTORS
C
   30 REWIND NFILED
      DO 35 J=1,KOUNT2
      READ (NFILED,600)DUMMY,(PASS(L),L=1,NFACT2)
      P(J)=PASS(KK)
   35 SP(J)=PASS(KK+1)
  20  I=KANDU(K)
      CALL REGRES(TYME,P,SP,KOUNT2,NTERMS(I),M,0,YFIT
     &,A02(I),A2(1,I),SIGA02(I),SIGA2(1,I),R,RMUL,RCHI,FTEST)
      CALL RESIDS(TYME,P,SP,YFIT,KOUNT2)
C
C  OUTPUT COEFFICIENTS OF REGRESSION FIT TO FACTOR VARIATION
C
      II=NTERMS(I)
      IF(NPRINT.LT.3)GO TO 750
      WRITE(NPRINT,700)I,A02(I),SIGA02(I)
      IF(II.GT.0)WRITE(NPRINT,702)(A2(JJ,I),SIGA2(JJ,I),JJ=1,II)
  700 FORMAT(/' POLYNOMIAL COEFFICIENTS FOR FACTOR',I2
     &/10X,'COEFF',10X,'SIGMA'
     &/(2E15.8))
  702 FORMAT(2E15.8)
      WRITE(NPRINT,952)RCHI,FTEST,RMUL
      IF(II.GT.0)WRITE(NPRINT,953)(IJ,R(IJ),IJ=1,II)
  952 FORMAT(/' REDUCED CHI SQUARE=',F10.2
     &/' FTEST =',E10.2/' RMUL =',E10.2)
  953 FORMAT(' R(',I1,') =',E10.2)
  750 IF(NFILEB.LE.6)GO TO 760
      WRITE(NFILEB,700)I,A02(I),SIGA02(I)
      IF(II.GT.0)WRITE(NFILEB,702)(A2(JJ,I),SIGA2(JJ,I),JJ=1,II)
      WRITE(NFILEB,952)RCHI,FTEST,RMUL
      IF(II.GT.0)WRITE(NFILEB,953)(IJ,R(IJ),IJ=1,II)
  760 CONTINUE
C
C
      KK=2*K+1
      K=K+1
      IF(K.LE.NFACTS)GO TO 30
```

```
C
C  CHECK EXTRAPOLATION PRECISION FOR ALL FACTORS FOR SECOND PLATEAU
C
C
C  CALCULATE EXTRAPOLATION TIMES
C
  65 TIMEX=(TPLAT2(1)-TPLAT1(2))/2.
C
C  SET EXTRAPOLATION TIME OF 200 SECONDS FOR CHECKING FIRST PLATEAU IN EXPT.
C  I.E., ASSUME 400 SECONDS BETWEEN PLATEAUS FOR FIRST PASS THROUGH
C
      IF(NPLAT.EQ.1)TIMEX=200.
C
C  EXTRAPOLATION TIMES ARE SYMMETRICAL
C
      TIMEB(1)=TPLAT1(1)-TMID1-TIMEX
      TIMEF(1)=TPLAT1(2)-TMID1+TIMEX
      TIMEB(2)=TPLAT2(1)-TMID2-TIMEX
      TIMEF(2)=TPLAT2(2)-TMID2+TIMEX
      IF(NPRINT.GE.3)WRITE(NPRINT,780)(TIMEB(I),TIMEF(I),I=1,2),TIMEX
     &,TPLAT1,TPLAT2
 780 FORMAT(//' EXTRAPOLATION TIMES ABOUT MID POINT OF PLATEAU'
     &,' FOR POLYNOMIALS'
     &/' FIRST PLATEAU =',2F10.0
     &/' SECOND PLATEAU =',2F10.0
     &/' REAL TIMES ARE',F9.0,' SECONDS ON EITHER SIDE OF'
     &/2F10.0,' FOR FIRST PLATEAU, AND',2F10.0,' FOR SECOND '/)
      IF(NFILEB.GT.6)WRITE(NFILEB,780)(TIMEB(I),TIMEF(I),I=1,2),TIMEX
     &,TPLAT1,TPLAT2
C
C  TEST THE TRENDS IN THE WEIGHT AND FACTORS TO SEE IF PRECISION
C    IN EXTRAPOLATED VALUES IS ADEQATE
C
      NGOODF=0
      NGOODR=0
      MERRTE=0
      MERFAC=0
C
C  OUTPUT HEADING FOR PRECISION LISTINGS
C
      IF(NPRINT.GE.3)WRITE(NPRINT,770)
      IF(NFILEB.GT.6)WRITE(NFILEB,770)
 770 FORMAT(//' EXTRAPOLATED VALUES AND SIGMAS BACK AND FORWARD'
     &,' IN TIME'
     &/' THEN PRECISIONS, = BACK, FORWARD AND REQUESTED'/)
C
      DO 80 J=1,NFACTS
      I=KANDU(J)
      L=IFDER(J)
C
C  OBTAIN EXTRAPOLATED QUANTITIES
C
      CALL EXTPOL(A02(I),A2(1,I),SIGA02(I),SIGA2(1,I)
     &,NTERMS(I)
     &,TIMEF(2),TIMEB(2),FACT2(2,I),SIGF2(2,I),FACT2(1,I),SIGF2(1,I)
     &,RATE2(2,I),SIGR2(2,I),RATE2(1,I),SIGR2(1,I)
```

```
     &,PRECFF(I),PRECFB(I),PRECRF(I),PRECRB(I),L,M)
C
C  TEST PRECISION IN RATES
C
      IF(L.EQ.0) GO TO 66
C
C  OUTPUT PRECISION IN RATES
C
      IF(NPRINT.GE.3)WRITE(NPRINT,775)NR,I,RATE2(1,I),SIGR2(1,I)
     &,RATE2(2,I),SIGR2(2,I),PRECRB(I),PRECRF(I),PRECR(I)
 775 FORMAT(A2,I1,2(2X,2E10.3),3F8.0)
      IF(NFILEB.GT.6)WRITE(NFILEB,775)NR,I,RATE2(1,I),SIGR2(1,I)
     &,RATE2(2,I),SIGR2(2,I),PRECRB(I),PRECRF(I),PRECR(I)
C
C  FLAG WHICH RATES ARE MEASURED ADEQUATELY
C
      MERIT=0
      IF((PRECRF(I)-PRECR(I)).GT.(-1.E-01).AND.(PRECRB(I)-PRECR(I))
     &.GT.(-1.E-01))MERIT=1
      IF(MERIT.EQ.1)CALL BSET(MERRTE,I)
      NGOODR=NGOODR+MERIT
C
C  TEST FACTOR LEVELS
C
  66 MERIT=0
C
C  OUTPUT PRECISION IN FACTORS
C
      IF(NPRINT.GE.3)WRITE(NPRINT,775)NF,I,FACT2(1,I),SIGF2(1,I)
     &,FACT2(2,I),SIGF2(2,I),PRECFB(I),PRECFF(I),PRECF(I)
      IF(NFILEB.GT.6)WRITE(NFILEB,775)NF,I,FACT2(1,I),SIGF2(1,I)
     &,FACT2(2,I),SIGF2(2,I),PRECFB(I),PRECFF(I),PRECF(I)
C
C  FLAG WHICH FACTORS ARE OF ADEQUATE PRECISION
C
      IF((PRECFF(I)-PRECF(I)).GT.(-1.E-01).AND.(PRECFB(I)-PRECF(I))
     &.GT.(-1.E-01))MERIT=1
      IF(MERIT.EQ.1)CALL BSET(MERFAC,I)
      NGOODF=NGOODF+MERIT
  80 CONTINUE
C
C  ARE ALL MEASUREMENTS OF ADEQUATE PRECISION?
C
C  IF SO, JUMP OVER PRECISION RESETTING
C
      IF(NGOODF.EQ.NFACTS.AND.NGOODR.EQ.NRATES)RETURN
C
C  IF NOT, CONTINUE MEASURING PLATEAU IN PLATTO UNLESS STORAGE
C    ARRAY (250 POINTS) OR TIME LIMIT (TIMLIM) HAS BEEN EXCEEDED
C    NSTOP IS FLAG, =0 FOR CONTINUE, =1 OR 2 FOR DO NOT GO BACK TO PLATTO
C
      IF(NSTOP.NE.0)GO TO 570
      IFGO=1
      RETURN
C
C  CHANGE PRECISION SPECIFICATIONS BECAUSE OF FAILING CASES
```

```
C
  570 IF(NPRINT.GE.3)WRITE(NPRINT,150)
      IF(NFILEB.GT.6)WRITE(NFILEB,150)
  150 FORMAT(//' NEW PRECISION LEVELS FROM PRECIS'
     &/' FACTOR   LEVEL     RATE     NEW?')
C
C
      DO 130 J=1,NFACTS
      MERIT=1

      MERYT=1
      I=KANDU(J)
      L=IFDER(J)
      IF(BTEST(MERFAC,I))GO TO 910
      MERIT=0
      PRECF(I)=((PRECFF(I)+PRECFB(I))/2.)*0.9
C
C
  910 IF(L.EQ.0)GO TO 135
      IF(BTEST(MERRTE,I))GO TO 135
      MERYT=0
      PRECR(I)=((PRECRF(I)+PRECRB(I))/2.)*0.9
  135 CONTINUE
C
C
      IF(NPRINT.GE.3)WRITE(NPRINT,160)I,PRECF(I)
     &,PRECR(I),NWORD(MERIT+1),NWORD(MERYT+1)
      IF(NFILEB.GT.6)WRITE(NFILEB,160)I,PRECF(I),PRECR(I),NWORD(
     &MERIT+1),NWORD(MERYT+1)
  160 FORMAT(I7,2F8.3,2X,2(2X,A4))
  130 CONTINUE
      RETURN
      END
```

```
$ASSM
PRIME PROG
$FORT
$TRGT 16
C ******************************************************************
C
      SUBROUTINE PRIME
C
C ******************************************************************
C  SUBROUTINE TO ALLOW CHANGES IN INITIAL VALUES OF CONTROL VARIABLES
C  NEEDS COMMON BLOCKS:
C     COMMON /BIAS/
C     COMMON /DERIVE/
C     COMMON /FACTR/
C     COMMON /IO/
C     COMMON /PARAMS/
C     COMMON /PLAT/
C     COMMON /MIDAST/
C     COMMON /REGRES/
C     COMMON /WHEN/
C
C  AND SUBROUTINE THMCPL
C
C  COMMON STORAGE IS DEFINED VIA ARRAY =DUMP=
      COMMON DUMP(900)
C
C  COMMON /BIAS/
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C  COMMON /DERIVE/
C
      EQUIVALENCE (PRECE,DUMP(131))
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119)),(TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
```

```
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
C
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160)),(NUM(1)
     &,DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
C  COMMON /PARAMS/
C
      DIMENSION A01(8),A02(8),SIGA01(8),SIGA02(8)
     &,A1(4,8),A2(4,8),SIGA1(4,8),SIGA2(4,8)
     &,FACT1(2,8),SIGF1(2,8),RATE1(2,8),SIGR1(2,8)
     &,FACT2(2,8),SIGF2(2,8),RATE2(2,8),SIGR2(2,8)
     &,TIMEF(2),TIMEB(2)
      EQUIVALENCE (A01(1),DUMP(200)),(A02(1),DUMP(208))
     &, (SIGA01(1),DUMP(216)),(SIGA02(1),DUMP(224))
     &, (A1(1,1),DUMP(232)),(A2(1,1),DUMP(264))
     &, (SIGA1(1,1),DUMP(296)),(SIGA2(1,1),DUMP(328))
      EQUIVALENCE (FACT1(1,1),DUMP(360)),(SIGF1(1,1),DUMP(376))
     &,(RATE1(1,1),DUMP(392)),(SIGR1(1,1),DUMP(408))
     &,(FACT2(1,1),DUMP(424)),(SIGF2(1,1),DUMP(440))
     &,(RATE2(1,1),DUMP(456)),(SIGR2(1,1),DUMP(472))
     &,(KOUNT1,DUMP(488)),(KOUNT2,DUMP(489))
      EQUIVALENCE (TIMEF(1),DUMP(500)),(TIMEB(1),DUMP(502))
     &,(DAY,DUMP(504)),(HDTIME,DUMP(505))
     &,(SKIP,DUMP(506)),(RATMIN,DUMP(507)),(RATMAX,DUMP(508))
     &,(TJUMP,DUMP(509))
C
C  COMMON /PLAT/
C
      DIMENSION RCHI(8)
      EQUIVALENCE (LPLAT,DUMP(510)),(NELAPS,DUMP(511))
     &,(NFIRST,DUMP(512)),(NCHI,DUMP(513)),(NSTOP,DUMP(514))
     &,(START,DUMP(515)),(RCHI(1),DUMP(516))
     &,(NOTHER,DUMP(524)),(KSTORE,DUMP(525))
     &,(ENDSIG,DUMP(526))
     &,(TOIM4,DUMP(527)),(FACTIM,DUMP(528)),(EQUIL,DUMP(529))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C  COMMON /WHEN/
C
      DIMENSION TPLAT1(2),TPLAT2(2),NTYME(3)
      EQUIVALENCE (BEGIN,DUMP(880)),(TPLAT1(1),DUMP(881))
     &,(TPLAT2(1),DUMP(883)),(NTYME(1),DUMP(885))
C
C
C
C
C
      DIMENSION NTITLE(72),FTIME(4)
C
```

```
C
C  READ AND WRITE TITLE
C
      WRITE(NTTYO,1)
    1 FORMAT(' GIVE TITLE FOR RUN, 2 LINES, 72 CHARS/LINE MAXIMUM')
      READ(NTTYI,2)NTITLE
    2 FORMAT(36A2)
      WRITE(NTTYO,2)NTITLE
C
C  OPTIONAL READ OF CONTROL PARAMETERS FROM SAVE FILE
C
      NFILEC=0
      WRITE(NTTYO,70)
   70 FORMAT(' READ PARAMETERS FROM SAVE FILE?'
     &/' 0=NO, 1=YES, GIVE IN I1 FORMAT')
      READ(NTTYI,75)J
   75 FORMAT(I1)
      IF(J.EQ.0)GO TO 80
      WRITE(NTTYO,77)
 77   FORMAT(' GIVE FILE NUMBER IN I2 FORMAT'
     &/' PROGRAM WILL THEN PAUSE, GIVE CONTINUE WHEN READY'
     &/' FILE WILL BE REWOUND FIRST')
      READ(NTTYI,78)NFILEC
 78   FORMAT(I2)
      PAUSE 5
      REWIND NFILEC
      READ(NFILEC,500)NTHCP,TSCALE,SKIP,KANDU,IFUNCT,IFDER,INPUT,SCALE
     &,NRANGE
     &,NREADS,NTERMS,CHTEST,PTIME,PRECF,PRECR,NFACTS,NRATES,TIMLIM,NBADS
     &,EQUIL,INTVAL,ENDSIG,RATMIN,RATMAX,TJUMP,AT
     &,NTTYI,NTTYO,NPRINT,NFILEA,NFILEB,NFILED,NFILEE
     &,M,PRECE
      WRITE(NTTYO,500)NTHCP,TSCALE,SKIP,KANDU,IFUNCT,IFDER,INPUT,SCALE
     &,NRANGE
     &,NREADS,NTERMS,CHTEST,PTIME,PRECF,PRECR,NFACTS,NRATES,TIMLIM,NBADS
     &,EQUIL,INTVAL,ENDSIG,RATMIN,RATMAX,TJUMP,AT
     &,NTTYI,NTTYO,NPRINT,NFILEA,NFILEB,NFILED,NFILEE
     &,M,PRECE
C
C  OPTIONAL CHANGE OF CONTROL PARAMETERS
C
   80 CONTINUE
      WRITE(NTTYO,10)
   10 FORMAT(' DO YOU WANT TO CHANGE ANY CONTROL PARAMETERS?'
     &/' IF NOT, GIVE PARAMETER OF ZERO'
     &/' IF SO, ENTER PARAMETER NUMBER, ELEMENT IN PARAMETER ARRAY,'
     &/' NEW VALUE IN I3,I3,F7.3 FORMAT'
     &/' INTEGER VALUES WILL BE INTEGERISED, GIVE NEW VALUE WITH'
     &/' DECIMAL POINT')
      GO TO 60
   50 WRITE(NTTYO,52)
   52 FORMAT(' ANOTHER CHANGE?')
   60 READ(NTTYI,62)N,I,VALUE
   62 FORMAT(2I3,F7.3)
      IF(N.LE.0)GO TO 400
      IF(N.GT.35)GO TO 50
```

```
      NVALUE=IFIX(VALUE+SIGN(0.5,VALUE))
      GO TO (100,103,106,104,108,109,110,120,130,135,136,140,150,160
     &,170,180,182,190
     &,192,193,194,196,197,198,199,200,210,220,230,240,250,255,256,300
     &,260)
     &,N
C
C  CHANGE THERMOCOUPLE TYPE
  100 WRITE(NTTYO,101)
  101 FORMAT(' GIVE THERMOCOUPLE TYPE AGAIN (AS A LETTER)')
      READ(NTTYI,102)NTHCP
  102 FORMAT(A1)
      GO TO 50
C  CHANGE SCALE OF THERMOCOUPLE TO DC VOLTAGE CONVERSION
  103 TSCALE=VALUE
      GO TO 50
C  CHANGE TIME TO SKIP BETWEEN READINGS
  106 SKIP=VALUE
      GO TO 50
C  SET UP WHICH EXPERIMENTAL VARIABLES ARE TO BE FACTORS (CHANGED)
  104 KANDU(I)=NVALUE
      GO TO 50
C  SET FUNCTION OF RAW READING TO BE USED (VIA FACFUN)
  108 IFUNCT(I)=NVALUE
      GO TO 50
C  FACTORS FOR WHICH DERIVATIVES ARE TO BE USED
  109 IFDER(I)=NVALUE
C  CHANGE ANALOG SCANNER INPUTS
  110 INPUT(I)=NVALUE
       GO TO 50
C  CHANGE SCALE VALUES
  120 SCALE(I)=VALUE
      GO TO 50
C  CHANGE NRANGE
  130 NRANGE(I)=NVALUE
      GO TO 50
C  CHANGE NUMBER OF DVM READINGS MAKING UP AVERAGE FOR EACH
C     FACTOR
  135 NREADS(I)=NVALUE
      GO TO 50
C  CHANGE NUMBER OF TERMS IN POLYNOMIAL FIT
  136 NTERMS(I)=MIN0(NVALUE,5)
      GO TO 50
C  CHANGE CHI SQUARE VALUES FOR EQUILIBRIUM IN FACTORS
  140 CHTEST(I)=VALUE
      GO TO 50
C  CHANGE MAX TIME FOR FACTOR TO TAKE TO STEADY DOWN
  150 PTIME(I)=VALUE
      GO TO 50
C  CHANGE MIN PRECISION IN FACTOR READINGS
  160 PRECF(I)=VALUE
      GO TO 50
C  CHANGE MIN PRECISION IN RATES
  170 PRECR(I)=VALUE
      GO TO 50
C  CHANGE NUMBER OF FACTORS
```

```
  180 NFACTS=NVALUE
      GO TO 50
C  CHANGE NUMBER OF RATES
  182 NRATES=NVALUE
      GO TO 50
C  CHANGE TIME LIMIT FOR PLATEAU
  190 TIMLIM=VALUE
      GO TO 50
C  CHANGE NUMBER OF BAD POINTS TO ALLOW BEFORE PLATEAU STABILISES
  192 NBADS=NVALUE
      GO TO 50
C  CHANGE EQUILIBRIUM TIME FOR SETTLING DOWN OF NEW FACTOR LEVELS
  193 EQUIL=VALUE
      GO TO 50
C  NUMBER OF READINGS BETWEEN CHECKS FOR ADEQUATE PRECISION
  194 INTVAL=NVALUE
      GO TO 50
C  NUMBER OF POOLED SIGMAS SUCCESSIVE WEIGHT READINGS MUST DIFFER BY
C    IF TERMINATION OF EXPERIMENT IS NOT TO OCCUR IN PLATTO
  196 ENDSIG=VALUE
      GO TO 50
C  MINIMUM RATE TO ALLOW BEFORE INCREASING TEMPERATURE BY TJUMP
  197 RATMIN=VALUE
      GO TO 50
C  MAXIMUM RATE TO ALLOW BEFORE DECREASING TEMP BY TJUMP
  198 RATMAX=VALUE
      GO TO 50
C  TEMPERATURE INCREMENT WHEN MIN RATE REACHED
  199 TJUMP=VALUE
      GO TO 50
C  CHANGE INPUT/OUTPUT UNITS
  200 NTTYI=NVALUE
      GO TO 50
  210 NTTYO=NVALUE
      GO TO 50
  220 NPRINT=NVALUE
      GO TO 50
  230 NFILEA=NVALUE
      GO TO 50
  240 NFILEB=NVALUE
      GO TO 50
C  CHANGE SAVE UNIT
  250 NFILEC=NVALUE
      GO TO 50
C  CHANGE SCRATCH FILE UNIT
  255 NFILED=NVALUE
      GO TO 50
C  CHANGE OUTPUT FILE FOR ACTIVATION ENERGIES
  256 NFILEE=NVALUE
      GO TO 50
C  CHANGE POLYNOMIAL EXPONENTS
  300 M(I)=NVALUE
      GO TO 50
C  CHANGE PRECISION IN ACTIVATION ENERGY
 260  PRECE=VALUE
      GO TO 50
```

```
  400 CONTINUE
C
C
      WRITE(NTTYO,500)NTHCP,TSCALE,SKIP,KANDU,IFUNCT,IFDER,INPUT,SCALE
     &,NRANGE
     &,NREADS,NTERMS,CHTEST,PTIME,PRECF,PRECR,NFACTS,NRATES,TIMLIM,NBADS
     &,EQUIL,INTVAL,ENDSIG,RATMIN,RATMAX,TJUMP,AT
     &,NTTYI,NTTYO,NPRINT,NFILEA,NFILEB,NFILED,NFILEE
     &,M,PRECE
      WRITE(NTTYO,430)
  430 FORMAT(' CHANCE FOR FURTHER CHANGES? (0=NO,1=YES)')
      READ(NTTYI,75)J
      IF(J.NE.0)GO TO 80
C
      IF(NPRINT.GE.3)WRITE(NPRINT,925)
  925 FORMAT('1')
      IF(NPRINT.GE.3)WRITE(NPRINT,2)NTITLE
      IF(NFILEA.GT.6)WRITE(NFILEA,2)NTITLE
      IF(NFILEB.GT.6)WRITE(NFILEB,2)NTITLE
      IF(NFILEE.GE.6)WRITE(NFILEE,2)NTITLE
C
C  GET THERMOCOUPLE COEFFICIENTS
C
      CALL THMCPL(NTHCP,AT)
C
C  OPTION OF STORING PRIMED VALUES ON SAVE FILE
C
      WRITE(NTTYO,450)
  450 FORMAT(' OUTPUT PARAMETERS ON SAVE FILE'
     &/' 0=NO,1=YES, IN I1 FORMAT')
      READ(NTTYI,75)J
      IF(J.EQ.0)GO TO 470
      WRITE(NTTYO,77)
      READ(NTTYI,78)NFILEC
      PAUSE 6
      REWIND NFILEC
      WRITE(NFILEC,500)NTHCP,TSCALE,SKIP,KANDU,IFUNCT,IFDER,INPUT,SCALE
     &,NRANGE
     &,NREADS,NTERMS,CHTEST,PTIME,PRECF,PRECR,NFACTS,NRATES,TIMLIM,NBADS
     &,EQUIL,INTVAL,ENDSIG,RATMIN,RATMAX,TJUMP,AT
     &,NTTYI,NTTYO,NPRINT,NFILEA,NFILEB,NFILED,NFILEE
     &,M,PRECE
  470 CONTINUE
C
C  OUTPUT CURRENT VALUES
C
      IF(NPRINT.LT.3)GO TO 455
      WRITE(NPRINT,500)NTHCP,TSCALE,SKIP,KANDU,IFUNCT,IFDER,INPUT,SCALE
     &,NRANGE
     &,NREADS,NTERMS,CHTEST,PTIME,PRECF,PRECR,NFACTS,NRATES,TIMLIM,NBADS
     &,EQUIL,INTVAL,ENDSIG,RATMIN,RATMAX,TJUMP,AT
     &,NTTYI,NTTYO,NPRINT,NFILEA,NFILEB,NFILED,NFILEE
     &,M,PRECE
  500 FORMAT(/' PARAMETERS INITIALISED AS FOLLOWS:'
     &/' NTHCP =  ',9X,A1,3X,'TSCALE =',F8.2,3X,'SKIP=',F5.0
     &/' FACTORS =',8I7
```

```
     &/' FACFUN = '.8I7
     &/' DERIVS = '.8I7
     &/' INPUTS = '.8I7
     &/' SCALES = '.8F7.3
     &/' NRANGE = '.8I7
     &/' NREADS = '.8I7
     &/' NTERMS = '.8I7
     &/' CHI TEST='.8F7.0
     &/' PTIMES = '.8F7.1
     &/' PRECF =  '.8F7.2
     &/' PRECR =  '.8F7.2
     &/' NFACTS = '.I3
     &.' NRATES = '.I3
     &.' TIMLIM  ='.F5.0
     &.' BAD PTS ='.I3
     &.' EQUIL ='.F5.0
     &/' CHECKS  ='.I4
     &.' END SIG ='.F5.1
     &.' RATMIN ='.F6.1
     &.' RATMAX ='.F6.1
     &.' TJUMP ='.F5.1
     &/' TC TO D ='.4E14.5
     &/' IO UNITS='.7I7
     &/' POLNOM EXP='.I5.4I7
     &/' PRECISION IN E ACT ='.F7.2)
 455 IF(NFILEA.LE.6)GO TO 460
     WRITE(NFILEA.2)NTITLE
 460 IF(NFILEB.LE.6)GO TO 610
     WRITE(NFILEB.500)NTHCP.TSCALE.SKIP.KANDU.IFUNCT.IFDER.INPUT.SCALE
     &.NRANGE
     &.NREADS.NTERMS.CHTEST.PTIME.PRECF.PRECR.NFACTS.NRATES.TIMLIM.NBADS
     &.EQUIL.INTVAL.ENDSIG.RATMIN.RATMAX.TJUMP.AT
     &.NTTYI.NTTYO.NPRINT.NFILEA.NFILEB.NFILED.NFILEE
     &.M.PRECE
 610 CONTINUE
     RETURN
     END
```

```
$ASSM
REGRES PROG
$FORT
$TRGT 16
C
C     ****************************************************************
C
      SUBROUTINE REGRES (X,Y,SIGMAY,NPTS,NTERMS,M,MODE,YFIT,A0,A,SIGMA0,
     2 SIGMAA,R,RMUL,CHISQR,FTEST)
C
C     ****************************************************************
C
C  NEEDS THE FOLLOWING FUNCTION AND SUBROUTINE:
C    FCTN
C    MATINV
C
C
      DOUBLE PRECISION ARRAY,SUM,YMEAN,SIGMA,CHISQ,XMEAN,SIGMAX
      DIMENSION X(NPTS),Y(NPTS),SIGMAY(NPTS),M(5),YFIT(NPTS)
     &,SIGMAA(5),R(5),A(5)
      DIMENSION WEIGHT(250),XMEAN(5),SIGMAX(5),ARRAY(5,5)
C
C  INITIALIZE SUMS AND ARRAYS
C
      SUM=0.
      YMEAN=0.
      SIGMA=0.
      CHISQ=0.
      RMUL=0.
      DO 10 I=1,NPTS
10        YFIT(I)=0.
      IF (NTERMS.LE.0) GO TO 30
      DO 20 J=1,NTERMS
          XMEAN(J)=0.
          SIGMAX(J)=0.
          R(J)=0.
          A(J)=0.
          SIGMAA(J)=0.
          DO 20 K=1,NTERMS
20        ARRAY(J,K)=0.
30    CONTINUE
C
C  ACCUMULATE WEIGHTED SUMS
C
      DO 110 I=1,NPTS
          IF (MODE) 40,70,80
40        IF (Y(I)) 60,70,50
50        WEIGHT(I)=1./Y(I)
          GO TO 90
60        WEIGHT(I)=1./(-Y(I))
          GO TO 90
70        WEIGHT(I)=1.
          GO TO 90
80        WEIGHT(I)=1./SIGMAY(I)**2
90        SUM=SUM+WEIGHT(I)
          YMEAN=YMEAN+WEIGHT(I)*Y(I)
```

```
         IF (NTERMS.LE.0) GO TO 110
         DO 100 J=1,NTERMS
100         XMEAN(J)=XMEAN(J)+WEIGHT(I)*FCTN(X,I,J,M,NPTS)
110      CONTINUE
      YMEAN=YMEAN/SUM
      IF (NTERMS.LE.0) GO TO 130
      DO 120 J=1,NTERMS
120      XMEAN(J)=XMEAN(J)/SUM
130   CONTINUE
      FNPTS=NPTS
      WMEAN=SUM/FNPTS
      DO 140 I=1,NPTS
140      WEIGHT(I)=WEIGHT(I)/WMEAN
C
C  ACCUMULATE MATRICES R AND ARRAY
C
      DO 150 I=1,NPTS
      SIGMA=SIGMA+WEIGHT(I)*(Y(I)-YMEAN)**2
      IF (NTERMS.LE.0) GO TO 160
      DO 150 J=1,NTERMS
      SIGMAX(J)=SIGMAX(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))**2
      R(J)=R(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))*(Y(I)-YMEAN)
         DO 150 K=1,J
      ARRAY(J,K)=ARRAY(J,K)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))
     2 *(FCTN(X,I,K,M,NPTS)-XMEAN(K))
  150 CONTINUE
160   CONTINUE
      FREE1=NPTS-1
      SIGMA=DSQRT(SIGMA/FREE1)
      IF (NTERMS.LE.0) GO TO 180
      DO 170 J=1,NTERMS
         SIGMAX(J)=DSQRT(SIGMAX(J)/FREE1)
         R(J)=R(J)/(FREE1*SIGMAX(J)*SIGMA)
         DO 170 K=1,J
         ARRAY(J,K)=ARRAY(J,K)/(FREE1*SIGMAX(J)*SIGMAX(K))
170      ARRAY(K,J)=ARRAY(J,K)
180   CONTINUE
C
C  INVERT SYMMETRIC MATRIX
C
      IF (NTERMS.LE.0) GO TO 200
      CALL MATINV (ARRAY,NTERMS,DET)
      IF (DET) 200,190,200
190   A0=0.
      SIGMA0=0.
      RMUL=0.
      CHISQR=0.
      FTEST=0.
      GO TO 330
C
C  CALCULATE COEFFICIENTS, FIT, AND CHI SQUARE
C
200   A0=YMEAN
      IF (NTERMS.LE.0) GO TO 230
      DO 220 J=1,NTERMS
         DO 210 K=1,NTERMS
```

```
210           A(J)=A(J)+R(K)*ARRAY(J,K)
          A(J)=A(J)*SIGMA/SIGMAX(J)
          AO=AO-A(J)*XMEAN(J)
          DO 220 I=1,NPTS
220       YFIT(I)=YFIT(I)+A(J)*FCTN(X,I,J,M,NPTS)
230   CONTINUE
      DO 240 I=1,NPTS
          YFIT(I)=YFIT(I)+AO
240       CHISQ=CHISQ+WEIGHT(I)*(Y(I)-YFIT(I))**2
      FREEN=NPTS-NTERMS-1
      CHISQR=CHISQ*WMEAN/FREEN
C
C  CALCULATE  UNCERTAINTIES
C
      IF (MODE) 250,260,250
250   VARNCE=1./WMEAN
      GO TO 270
260   VARNCE=CHISQR
  270 IF (NTERMS.LE.0) GO TO 290
      DO 280 J=1,NTERMS
          SIGMAA(J)=ARRAY(J,J)*VARNCE/(FREE1*SIGMAX(J)**2)
          SIGMAA(J)=SQRT(SIGMAA(J))
280       RMUL=RMUL+A(J)*R(J)*SIGMAX(J)/SIGMA
      FREEJ=NTERMS
      FTEST=(RMUL/FREEJ)/((1.-RMUL)/FREEN)
      RMUL=SQRT(RMUL)
      GO TO 300
290   CONTINUE
      FREEJ=0.
      FTEST=0.
      RMUL=0.
300   SIGMAO=VARNCE/FNPTS
      IF (NTERMS.LE.0) GO TO 320
      DO 310 J=1,NTERMS
          DO 310 K=1,NTERMS
310       SIGMAO=SIGMAO+VARNCE*XMEAN(J)*XMEAN(K)*ARRAY(J,K)/(FREE1*SIGMA
     2X(J)*SIGMAX(K))
320   SIGMAO=SQRT(SIGMAO)
330   RETURN
      END
```

```
$ASSM
RESIDS PROG
$FORT
$TRGT 16
C     ****************************************************************************
C
      SUBROUTINE RESIDS (T,P,SP,Y,N)
C
C     ****************************************************************************
C  SUBROUTINE TO LIST RESIDUALS FROM LEAST SQUARES FIT
C
C  NEEDS THE FOLLOWING COMMON BLOCK:
C    COMMON /IO/
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(1060)
C
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C
      DIMENSION T(N),P(N),SP(N),Y(N)
      IF(NPRINT.GE.3)WRITE(NPRINT,10)
      IF(NFILEB.GT.6)WRITE(NFILEB,10)
   10 FORMAT(//5X,'TIME',13X,'P',10X,'YFIT',3X,'SIGMA P'
     &,4X,'P-YFIT',1X,'(P-YFIT)/SIGP',' DEL P')
      DO 30 I=1,N
      TEMP1=P(I)-Y(I)
      TEMP2=TEMP1/AMAX1(SP(I),1.E-15)
      DEL=0.
      IF(I.GT.1)DEL=P(I)-P(I-1)
      IF(NPRINT.GE.3)WRITE(NPRINT,20)T(I),P(I)
     &,Y(I),SP(I),TEMP1,TEMP2,DEL
      IF(NFILEB.GT.6)WRITE(NFILEB,20)T(I),P(I),Y(I),SP(I),TEMP1,TEMP2
     &,DEL
   20     FORMAT (F10.1,2E14.6,4E10.3)
   30     CONTINUE
      IF(NPRINT.GE.3)WRITE(NPRINT,40)
      IF(NFILEB.GT.6)WRITE(NFILEB,40)
   40 FORMAT (//)
      RETURN
      END
```

```
$ASSM
RESPON PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE RESPON(FTIME,NERROR)
C
C ************************************************************
C  SUBROUTINE TO ESTIMATE FACTOR RESPONSE FTIME FOR USE IN
C  ESTIMATING PRECISION OF EXTRAPOLATED VALUE TO HALFWAY IN TIME
C  BETWEEN OLD AND NEW LEVELS OF FACTOR
C
C  NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C
C    COMMON /BIAS/
C    COMMON /FACTR/
C    COMMON /IO/
C    COMMON /MIDASR/
C    SETPAR
C    TRAMID
C    WAYT
C
C  INITIAL CHOICES FOR FACTOR LEVELS TO DETERMINE RESPONSE TIMES AT:
C          T - 250 AND 270 DEG.
C          P - 400 AND 450 MM HG
C          N2 - 200 AND 240 SCC/MIN
C          O2 - 50 AND 60 SCC/MIN
C  THESE VALUES MAY BE CHANGED BY USER INPUT
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /BIAS/BIASV(8),SIGBV(8)
C
      DIMENSION BIASV(8),SIGBV(8)
      EQUIVALENCE (BIASV(1),DUMP(1)),(SIGBV(1),DUMP(9))
C
C  COMMON /FACTR/
C
      DIMENSION KANDU(8),IFUNCT(8),IFDER(8),INPUT(8),SCALE(8)
     &,NRANGE(8),NREADS(8),NTERMS(8),CHTEST(8),PTIME(8)
     &,PRECF(8),PRECR(8),AT(4)
      EQUIVALENCE (NTHCP,DUMP(17)),(KANDU(1),DUMP(18))
     &,(IFUNCT(1),DUMP(26)),(IFDER(1),DUMP(34))
     &,(INPUT(1),DUMP(42)),(SCALE(1),DUMP(50))
     &,(NRANGE(1),DUMP(58)),(NREADS(1),DUMP(66))
     &,(NTERMS(1),DUMP(74)),(CHTEST(1),DUMP(82))
     &,(PTIME(1),DUMP(90)),(PRECF(1),DUMP(98))
     &,(PRECR(1),DUMP(106)),(NFACTS,DUMP(114))
     &,(NRATES,DUMP(115)),(TIMLIM,DUMP(116))
     &,(NBADS,DUMP(117)),(INTVAL,DUMP(118))
     &,(AT(1),DUMP(119))
C
C  COMMON /IO/
```

```
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),JUNK2(18)
      EQUIVALENCE (OUT(1),DUMP(140)),(REPLY(1),DUMP(160))
     &,(NUM(1),DUMP(180)),(JUNK2(1),OUT(3)),(NFILTR,DUMP(190))
C
C
      DIMENSION LABEL(2,4),FTIME(4),FVAL(2,4)
     &,NWHAT(4),CHI(4)
     &,T(4),TEMP1(2),TEMP2(2)
C
C
      DATA LABEL/'TEMP','    ','PRES','S   ','N2 F','LOW '
     &,'O2 F','LOW '/
      DATA FVAL/250.,270.,400.,450.,200.,240.,50.,60./
C
C
      NERROR=0
C
C  MEASURE RESPONSE TIMES TO CHECK CHI TEST VALUES, PTIMES, ETC.?
C
      WRITE(NTTYO,700)
  700 FORMAT(/' MEASURE FACTOR RESPONSE TIMES? 0=NO, 1=YES')
      READ(NTTYI,20)J
      IF(J.EQ.0)RETURN
C
C  READ IN FACTOR RESPONSE TIMES IF SO DESIRED
C
      DO 80 I=1,4
      WRITE(NTTYO,10)(LABEL(J,I),J=1,2)
   10 FORMAT(//' READ IN FACTOR RESPONSE TIMES OR MEASURE THEM?'/
     &' ENTER 0 FOR READ'/
     &'       1 FOR MEASURE'/
     &' IN I1 FORMAT FOR ',2A4,' PARAMETER ')
      READ(NTTYI,20)NWHAT(I)
   20 FORMAT(I1)
      IF(NWHAT(I).GT.0)GO TO 80
      WRITE(NTTYO,40)(LABEL(J,I),J=1,2)
   40 FORMAT(' FACTOR RESPONSE TIME FOR ',2A4,' IN F10.3 FORMAT?')
      READ(NTTYI,50)T(I)
   50 FORMAT(F7.3)
      CHI(I)=0
   80 CONTINUE
C
C  OPTIONALLY CHANGE TARGET VALUES
C
      WRITE(NTTYO,560)
  560 FORMAT(/' TARGET VALUES OF FACTORS IN RESPON ARE:')
      WRITE(NTTYO,500)((I,J,FVAL(J,I),J=1,2),I=1,4)
  500 FORMAT(2I7,F10.1)
  565 WRITE(NTTYO,510)
```

```
 510 FORMAT(' DO YOU WANT TO CHANGE ANY VALUES?'
    &/'  IF SO, ENTER:'
    &/'     FACTOR NUMBER, 1=T, 2=P, 3=FN2, 4=FO2'
    &/'     1 OR 2 FOR FIRST OR SECOND VALUE OF A FACTOR'
    &/'     VALUE ITSELF'
    &/' ALL IN (2I3,F10.0)'
    &/' %%  END WITH FACTOR NUMBER OF ZERO (0) %%')
     READ(NTTYI,520)I,J,FTEMP
 520 FORMAT(2I3,F10.0)
     IF(I.EQ.0)GO TO 550
     IF(I.GE.1.AND.I.LE.4.AND.J.GE.1.AND.J.LE.2)GO TO 540
     WRITE(NTTYO,530)
 530 FORMAT(/'  INCORRECT INPUT, TRY AGAIN')
     GO TO 510
 540 FVAL(J,I)=FTEMP
     GO TO 510
 550 WRITE (NTTYO,500) ((I,J,FVAL(J,I),J=1,2),I=1,4)
     WRITE (NTTYO,562)
 562 FORMAT(/' ARE THESE VALUES OK? 0=YES, 1=NO (I1 FORMAT)')
     READ (NTTYI,20)NGO
     IF (NGO.NE.0) GO TO 565
     IF(NFILEB.GT.6)GO TO 570
     WRITE(NFILEB,560)
     WRITE(NFILEB,500)((I,J,FVAL(J,I),J=1,2),I=1,4)
 570 IF(NPRINT.LT.3)GO TO 580
     WRITE(NPRINT,560)
     WRITE(NPRINT,500)((I,J,FVAL(J,I),J=1,2),I=1,4)
 580 CONTINUE
     IF(NFILEB.GT.6)WRITE(NFILEB,125)
     IF(NPRINT.GE.3)WRITE(NPRINT,125)
125  FORMAT(' FACTOR RESPONSE TIMES  FOLLOW:'
    &/'  FACTOR    LOW   HIGH   TIME  CHISQ  MEASURED? TIMES & CHI '
    &,'VALUES'/)
C
C  MEASURE ANY RESPONSE TIMES NOT READ IN
C
     DO 100 I=1,4
     IF(NWHAT(I).LE.0)GO TO 100
C
C  START CLOCK
C
     OUT(1)=12
     OUT(3)='IT'
     OUT(4)='00'
     OUT(5)='00'
     OUT(6)='00'
     OUT(7)='R '
     OUT(8)='  '
     CALL TRAMID(OUT(2),13,REPLY,13,0.)
     OUT(1)=38
     KK=I+1
C
C  FIX TO READ OUTPUT OF FLOW METERS RATHER THAN INPUT TO THEM
C
     IF(I.GT.2)KK=I+3
C
```

```
C  SET FACTOR AND WAIT TILL IT IS STEADY
C
      DO 85 J=1,2
      CALL SETPAR(FVAL(J,I),I,NERROR,NTHCP,TC)
      IF(NERROR.GT.0)RETURN
C
C  NEED GAS FLOW TO MAINTAIN PRESSURE
C
      IF(I.NE.2)GO TO 86
      CALL SETPAR(500.,3,NERROR,NTHCP,TC)
      CALL SETPAR(0.,4,NERROR,NTHCP,TC)
   86 CALL WAYT(CHTEST(KK),PTIME(KK),INPUT(KK),NRANGE(KK)
     &,SCALE(KK)
     &,TEMP1(J),TEMP2(J),BIASV(KK),NREADS(KK),NTERMS(KK),NERROR)
   85 CONTINUE
C
C  CALCULATE CHI SQUARE AND TIME FOR STEADINESS
C
      T(I)=(TEMP1(2)+TEMP1(1))/2.
      CHI(I)=(TEMP2(1)+TEMP2(2))/2.
C
C  OUTPUT RESULTS
C
      FTIME(I)=T(I)/2.
      IF(NERROR.GT.0.AND.NFILEB.GT.5)WRITE(NFILEB,87)
      IF(NERROR.GT.0.AND.NPRINT.GE.3)WRITE(NPRINT,87)
   87 FORMAT(/' CONDITIONS NOT MET FOR FOLLOWING FACTOR')
      IF(NPRINT.GE.3)WRITE(NPRINT,120)(LABEL(J,I),J=1,2)
     &,(FVAL(J,I),J=1,2),T(I),CHI(I),NWHAT(I),TEMP1,TEMP2
      IF(NFILEB.GT.5)WRITE(NFILEB,120)(LABEL(J,I),J=1,2)
     &,(FVAL(J,I),J=1,2),T(I),CHI(I),NWHAT(I),TEMP1,TEMP2
  100 CONTINUE
  120 FORMAT(1X,2A4,3F7.0,F7.1,I5,4F7.0)
      RETURN
      END
```

```
$ASSM
RMIDAS PROG
$FORT
$TRGT 16
C ******************************************************************
C
      SUBROUTINE RMIDAS(AVE,SD,NRANGE,INPUT,SCALE,NREADS,BIASV,NERROR)
C
C ******************************************************************
C  SUBROUTINE TO READ =NREADS= VALUES (MAX=500) OF
C  VOLTAGE ON DVM AND CALCULATE AVERAGE (AVE) AND STANDARD
C  DEVIATION  (SD) CHARACTERISING SAMPLE (Q).
C
C  AVE AND SD ARE DEFINED ABOVE
C  NRANGE = RANGE SETTING ON DVM
C  SCALE = CONVERSION FACTOR FROM MIDAS READING TO PHYSICAL QUANTITY
C    DEPENDS ON DVM RANGE AND ANY CONVERSION IN UNITS.
C  NREADS = NUMBER OF READINGS TO TAKE IN ASSESSING AVERAGE VALUE
C  BIASV = BIAS VOLTAGE ON VOLTAGE INPUT TO DVM
C  NERRROR = ERROR INDICATOR ( = 0 FOR NORMAL OPERATION, = 4 FOR
C    INCORRECT DVM FUNCTIONING (I.E., 'NO'), = 5 FOR DVM WILL NOT SET RANGE
C    SPECIFIED)
C
C  NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C    COMMON /IO/
C    COMMON /MIDASR/
C    AVSDN
C    DEKODE
C    TRAMID
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10),TRIG(20)
      EQUIVALENCE (OUT(1),DUMP(140))
     &,(REPLY(1),DUMP(160)),(NUM(1),DUMP(180))
     &,(NFILTR,DUMP(190))
C
C
C
      DIMENSION Q(500)
      NTYMER=0
  200 NERROR=0
      NTIME=1
      AVE=0.
      SD=1.E-09
```

```
      NREADS=MIN0(500,NREADS)
C
C  PREPARE TO SET DVM TO RANGE AND FUNCTION, THEN TRIGGER
C
      OUT(1)=20
      OUT(2)=X'23'
      OUT(3)='MP'
      OUT(4)='3'
      OUT(5)=NUM(NFILTR+1)
      OUT(6)=NUM(NRANGE+1)
      OUT(7)='*T'
      DO 110 I=8,20
  110 OUT(I)='  '
C
C  PREPARE TO UNLOAD DVM
C
      TRIG(1)=20
      TRIG(2)=X'23'
      TRIG(3)='MU'
      DO 120 I=4,20
  120 TRIG(I)='  '
C
C
  100 DO 10 I=1,NREADS
C
C  TRIGGER
C
      CALL TRAMID(OUT(2),19,REPLY(2),1,0.)
C
C  UNLOAD
C
      CALL TRAMID(TRIG(2),19,REPLY(2),19,9.)
C
C  DEKODE TO GET VOLTAGE
C
      CALL DEKODE(REPLY(2),Q(I),7,0)
C
C  CHECK FOR =NO= ON DVM
C
      CALL DEKODE(REPLY(5),F,1,1)
      IF(F.LT.0.1)GO TO 25
C
C  RANGE ON DVM NOT LARGE ENOUGH
C
      NR=NRANGE
      NRANGE=NRANGE+1
      IF(NRANGE.GT.7)NRANGE=7
      IF(NFILEB.GT.6)WRITE(NFILEB,300)NR,NRANGE,INPUT
      IF(NPRINT.GE.3)WRITE(NPRINT,300)NR,NRANGE,INPUT
  300 FORMAT(//' $$$ RANGE CHANGE FROM',I3,' TO',I3,' ON INPUT',I3
     &,' $$$'//)
      NTYMER=NTYMER+1
      IF(NTYMER.GT.2)GO TO 201
      GO TO 200
C
C  DEKODE TO GET RANGE
```

```
C
   25 CALL DEKODE(REPLY(6),F,1,0)
      NR=IFIX(F+0.5)
      IF(NR.EQ.NRANGE) GO TO 70
      NTIME=NTIME+1
C
C  TRY 3 TIMES TO READ MIDAS CORRECTLY
C
      IF(NTIME.LE.3) GO TO 100
      NERROR=5
      RETURN
 70   CONTINUE
      Q(I)=Q(I)*SCALE-BIASV
      IF(NRANGE.GT.3)Q(I)=Q(I)*10.**(NRANGE-3)
   10 CONTINUE
C
C  FIND AVERAGE AND SIGMA OF READINGS
C
      CALL AVSDN(AVE,SD,Q,NREADS)
C
C  CHECK TO SEE IF RANGE CAN BE MADE MORE SENSITIVE
C
      IF(NRANGE.EQ.3)GO TO 80
      F=ABS((AVE/SCALE)+BIASV)
      FSD=SD/SCALE
      BOUND=15000.
      IF(NRANGE.GT.3)BOUND=BOUND*10.**(NRANGE-3)
      IF(F.GT.(BOUND-5.*FSD)) GO TO 80
      NTYMER=NTYMER+1
      IF(NTYMER.GT.2)GO TO 201
      NR=NRANGE
      NRANGE=NRANGE-1
      IF(NFILEB.GT.6)WRITE(NFILEB,300)NR,NRANGE,INPUT
      IF(NPRINT.GE.3)WRITE(NPRINT,300)NR,NRANGE,INPUT
      GO TO 200
  201 NERROR=4
   80 RETURN
      END
```

```
$ASSM
RPAR PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE RPAR(AVE,SD,INPUT,NRANGE,SCALE,NREADS,BIASV,NERROR)
C
C ************************************************************
C  SUBROUTINE TO READ VALUE FROM MIDAS
C
C  AVE = AVERAGE VALUE OF SAMPLE OF READINGS FROM MIDAS
C  SD = STANDARD DEVIATION OF SAMPLE (NOT OF AVERAGE)
C  INPUT = INPUT NUMBER ON MIDAS ANALOGUE SCANNER
C  NRANGE = VOLTAGE RANGE TO USE ON DVM
C  SCALE = FACTOR TO CONVERT FROM MIDAS VOLTAGE TO PHYSICAL QUANTITY
C    DEPENDS ON DVM RANGE AND ANY CONVERSION IN UNITS
C  NREADS = NUMBER OF READINGS TO TAKE IN BUILDING UP SAMPLE
C  BIASV = BIAS VOLTAGE ON VOLTAGE INPUT TO DVM
C  NERROR = ERROR INDICATOR, RETURNED FROM SUBROUTINE RMIDAS
C
C  NEEDS COMMON /MIDASR/
C
C  NEEDS THE FOLLOWING SUBROUTINE:
C  RMIDAS
C  TRAMID
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20),NUM(10)
      EQUIVALENCE (OUT(1),DUMP(140))
     &,(REPLY(1),DUMP(160)),(NUM(1),DUMP(180))
C
C
C
      NTIME=1
   20 NERROR=0
C
C  SET ANALOG SCANNER TO SEND SIGNAL TO DVM
C
      OUT(1)=10
      OUT(2)=X'23'
      OUT(3)='K'
      OUT(4)=NUM(INPUT+1)
      DO 30 I=5,10
   30 OUT(I)='  '
      CALL TRAMID(OUT(2),10,REPLY(2),1,0.)
C
C  READ MIDAS
C
      CALL RMIDAS(AVE,SD,NRANGE,INPUT,SCALE,NREADS,BIASV,NERROR)


      IF(NERROR.EQ.0)RETURN
C
C  TRY 3 TIMES TO READ PARAMETER
C
      NTIME=NTIME+1
      IF(NTIME.LE.3)GO TO 20
      RETURN
      END
```

```
$ASSM
SETPAR PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE SETPAR(PAR,NPAR,NERROR,NTHCP,TC)
C
C ****************************************************************
C  SUBROUTINE TO SET TEMPERATURE, PRESSURE OR FLOWS TO VALUE =PAR=
C  NPAR = 1 FOR TEMPERATURE
C         2 FOR PRESSURE
C         3 FOR FLOW ON CHANNEL 1
C         4 FOR FLOW ON CHANNEL 2
C
C  NTHCP = THERMOCOUPLE TYPE
C  NTHCP=K : TYPE K THERMOCOUPLE
C  NTHCP=E : TYPE E THERMOCOUPLE
C  TC = THERMOCOUPLE VOLTAGE IN MICROVOLTS
C  NERROR = ERROR INDICATOR ( = 0 FOR NO ERROR, = 1 FOR WRONG
C    SPECIFICATION OF THERMOCOUPLE TYPE I.E., NOT E OR K,
C    =2 FOR NPAR OUT OF RANGE 1 TO 4)
C
C  NEEDS PART OF COMMON /FACTR/
C  COMMON /IO/
C   COMMON /MIDASR/
C  AND SUBROUTINES ENKODE AND TRAMID
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /FACTR/
C
      EQUIVALENCE (TSCALE,DUMP(130))
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
C  COMMON /MIDASR/
C
      INTEGER*2 OUT(20),REPLY(20)
     &,NUM(10)
      EQUIVALENCE
     &(OUT(1),DUMP(140)),(REPLY(1),DUMP(160))
     &,(NUM(1),DUMP(180)),(NVDC,DUMP(191))
C
C
      NERROR=0
      OUT(1)=14
      OUT(2)=X'23'
      DO 30 I=7,20
```

```
  30 OUT(I)='  '
     IF(NPAR.GT.4.OR.NPAR.LT.1)GO TO 600
     GO TO (100,200,300,400),NPAR
C
C  SET TEMPERATURE USING DEGREES CELCIUS
C
 100 CONTINUE
     OUT(3)='D'
     OUT(6)=';'
C
C  CONVERT TEMP TO THERMOCOUPLE READING IN MV
C
     IF(NTHCP.NE.'K') GO TO 10
C
C  TYPE K THERMOCOUPLE
C
C  POLYNOMIAL CALCULATED BY B. DICKENS FOR 25 TO 450 DEGREES CELCIUS
C    USING VALUES FROM TABLE A7.2.1, NBS MONOGRAPH 125, 1974
C
C  RESULTS GOOD TO ABOUT 15 MICROVOLTS
C
C  POLYNOMIAL FITTED ABOUT 233.94 TO EQUALISE ERROR AT ENDS
C
     T=PAR-233.94
     TC=0.95106799E+01 + T*(0.40565763E-01 + T*(0.21380738E-05
    & + T*(0.15233454E-07)))
C
     GO TO 40
 10  IF(NTHCP.NE.'E') GO TO 20
C
C  TYPE E THERMOCOUPLE
C
C  POLYNOMIAL CALCULATED BY B. DICKENS FOR 25 TO 450 DEGREES CELCIUS
C    USING VALUES IN TABLE A5.2.1, NBS MONOGRAPH 125, 1974 .
C
C  RESULTS GOOD TO ABOUT 5 MICROVOLTS
C
C  POLYNOMIAL FITTED ABOUT 247.67 TO EQUALISE ERRORS AT ENDS
C
     T=PAR-247.67
C
     TC= 0.16994019E+02 + T*(0.76160789E-01 + T*(0.29884472E-04
    & - T*(0.37459611E-07)))
     GO TO 40
 20  NERROR=1
     RETURN
 40  CONTINUE
C
C  MAKE MAXIMUM FURNACE TEMP = 450 C BECAUSE OF PYREX GLASS ENCLOSURE
C
     NVDC=MIN1(8500.,TC*TSCALE)
     GO TO 500
C
C  SET PRESSURE
C  PRESS = DESIRED PRESSURE IN TORR = MM HG
C
```

```
  200 CONTINUE
      PRESS=PAR
      OUT(3)='D'
      OUT(6)=':'
C
C  SCALE PRESSURE TO APPROPRIATE VOLTAGE
C
      NVDC=IFIX(PRESS*10.)
      GO TO 500
C
C  SET FLOW RATE 1 OR 2 AS SPECIFIED
C  BY VALUE OF NPAR(=3 FOR FLOW 1, =4 FOR FLOW 2).
C  FLOW = DESIRED FLOW RATE IN SCC/MIN
C
  300 CONTINUE
      FLOW=PAR
      OUT(3)='G'
      OUT(6)=':'
C
C  CHANNEL 1 IS FOR N2,MAX FLOW 500 SCC/MIN FOR 5000 MVDC
C  CONVERT FLOW TO VOLTAGE
C
      NVDC=MIN1(5000.*FLOW/500.,5000.)
      GO TO 500
  400 CONTINUE
      FLOW=PAR
      OUT(3)='G'
      OUT(6)=';'
C
C  CHANNEL 2 IS FOR O2,MAX FLOW 200 SCC/MIN FOR 5000 MVDC
C  CONVERT FLOW TO VOLTAGE
C
      NVDC=MIN1(5000.*FLOW/200.,5000.)
C
C  SET MIDAS
C
  500 NVDC=MAX0(NVDC,0)
      K1=NVDC/1000
      K2=(NVDC-K1*1000)/100
      K3=(NVDC-K1*1000-K2*100)/10
      K4=NVDC-K1*1000-K2*100-K3*10
      OUT(4)=NUM(K1+1)*256+NUM(K2+1)
      OUT(5)=NUM(K3+1)*256+NUM(K4+1)
      CALL TRAMID(OUT(2),19,REPLY(2),1,0.)
      IF (NFILEB.GT.6) WRITE(NFILEB,700)PAR,NPAR,OUT
      IF (NPRINT.GE.3) WRITE(NPRINT,700)PAR,NPAR,OUT
  700 FORMAT(' FACSET'/F10.1,I10/16,2X,19A2)
      RETURN
 600  NERROR=2
      RETURN
      END
```

```
$ASSM
TDEGAB PROG
$FORT
$TRGT 16
C ************************************************************************
C
      FUNCTION TDEGAB (T,AT)
C
C************************************************************************
C
C  FUNCTION TO CALCULATE TEMPERATURE IN DEGREES ABSOLUTE
C  FROM THERMOCOUPLE READING
C
      DIMENSION AT(4)
      TP=0
      DO 10 K=1,4
10        TP=TP+AT(K)*T**K
      TDEGAB=TP+273.15
      RETURN
      END
```

```
$ASSM
THMCPL PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE THMCPL (NTHCP,AT)
C
C ****************************************************************
C  SUBROUTINE TO PROVIDE COEFFICIENTS FOR CONVERSION OF
C  THERMOCOUPLE READINGS IN MICROVOLTS TO DEGREES CELCIUS
C
C  NEEDS COMMON BLOCK /IO/
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128))
C
      DIMENSION AT(4)
C
   20 IF(IABS(NTHCP-'K').GT.0)GO TO 60
C
C  TYPE K THERMOCOUPLE,VALUES FROM TABLE A7.2.3,NBS MONO.125
C
40    IF(NPRINT.GE.3)WRITE (NPRINT,50)
      IF(NFILEB.GT.6)WRITE(NFILEB,50)
      AT(1)=2.4383248E-02
      AT(2)=9.7830251E-09
      AT(3)=3.6276965E-12
      AT(4)=-2.5756438E-16
50    FORMAT (/50H TYPE K THERMOCOUPLE=NI-CR/NI-AL,EXPRESSIONS GOOD /,
     2' TO 400 DEGREES C.(TABLE A7.2.3,.P.384,.NBS MONO.125,1974)')
      GO TO 110
C
C  TYPE E THERMOCOUPLE
C
   60 IF(IABS(NTHCP-'E').GT.0)GO TO 90
      IF(NPRINT.GE.3)WRITE (NPRINT,70)
      IF(NFILEB.GT.6)WRITE(NFILEB,70)
70    FORMAT (/50H TYPE E THERMOCOUPLE=NI-CR/CU-NI,EXPRESSIONS GOOD /,
     2'FROM -0.05 TO +0.04, 0 TO 400 DEGREES C.(TABLE A5.2.3,P.307'/
     3 ' NBS MONOGRAPH 125,1974)')
      AT(1)=1.7022525E-02
      AT(2)=-2.2097240E-07
      AT(3)=5.4809514E-12
      AT(4)=-5.7669892E-17
      GO TO 110
C
C  ERROR ROUTE
C
   90 WRITE(NTTYO,100)
100   FORMAT (/40H WRONG LETTER FOR THERMOCOUPLE,TRY AGAIN)
      WRITE (NTTYO,10)
10    FORMAT (/' GIVE THERMOCOUPLE TYPE, K OR E')
      READ (NTTYI,30) NTHCP
30    FORMAT (A1)
      GO TO 20
110   RETURN
      END
```

```
$ASSM
TIMER PROG
$FORT
$TRGT 16
C ******************************************************************
C
      FUNCTION TIMER(NTYME)
C
C ******************************************************************
C  FUNCTION TO GIVE REAL TIME CLOCK READING IN SECONDS
C    FROM READING MIDAS CLOCK
C
C  NTYME = HOURS,MINUTES,SECONDS
C
C  NEEDS COMMON /PARAMS/
C
C  REQUIRES THE FOLLOWING SUBROUTINES:
C    DEKODE
C    TRAMID
C
      COMMON DUMP (900)
      EQUIVALENCE (DAY,DUMP(504)),(HDTIME,DUMP(505))
C
      DIMENSION NTYME(3)
C
C  SET UP COMMAND STREAM FOR MIDAS.  THIS HAS TO LOOK LIKE AN INTEGER*2
C    HOLLERITH STREAM WITH THE NUMBER OF CHARACTERS IN THE FIRST WORD
C
      INTEGER*2 T(10),R(10)
      DATA T/12,X'23','IQ','::','<=','>?','R ','  ','  ','  '/
      DATA NT,NR/9,9/
C
C  CALL MIDAS TO READ TIME, REPLY IS 6 CHARACTERS IN ARRAY R
C
      CALL TRAMID(T(2),NT,R(2),NR,6,)
      TIMER=0.
C
C  GET HOUR DIGITS AND CHANGE THEM INTO SECONDS
C
      CALL DEKODE(R(2),F,2,0)
      TIMER=F*3600.
      NTYME(1)=IFIX(F)
C
C  GET MINUTES
C
      CALL DEKODE(R(3),F,2,0)
      TIMER=TIMER+F*60.
      NTYME(2)=IFIX(F)
C
C  GET SECONDS
C
      CALL DEKODE(R(4),F,2,0)
      TIMER=TIMER+F
      NTYME(3)=IFIX(F)
C
C  CHECK IF CLOCK HAS GONE PAST 23:59:59 HRS
C
   10 RETURN
      END
```

```
$ASSM
TRAMID PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE TRAMID(O,NO,R,NR,FS)
C
C ************************************************************
C  SUBROUTINE TO CHECK IF MIDAS IS FREE AND IF SO WRITE
C    INSTRUCTION ARRAY =O= AND RECEIVE REPLY ARRAY =R=
C  MIDAS IS ON UNIT =U=
C  LENGTH OF REPLY ARRAY IS =FS=
C  O AND R ARE HOLLERITH ARRAYS
C
C  NEEDS COMMON /IO/ AND SUBROUTINE MIDAS
C
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYO,DUMP(124)),(NPRINT,DUMP(125)),(NFILEB,DUMP
     &(127))
C
C
      INTEGER*2 O(NO),R(NR),T(20),Q(20)
      U=36.
      T(1)=4
      T(2)=X'23'
      T(3)='/ '
      DO 10 I=4,20
   10 T(I)='  '
C
C  CHECK IF MIDAS IS FREE
C
      NDONE=0
   50 NTIME=0
   30 CALL MIDAS(U,T(2),Q(2),0.,S)
      IF(S.LT.0.1)GO TO 20
      NTIME=NTIME+1
      IF(NTIME.LT.5000)GO TO 30
      WRITE(NTTYO,40)S
      IF(NPRINT.GE.3)WRITE(NPRINT,40)S
      IF(NFILEB.GT.6)WRITE(NFILEB,40)S
   40 FORMAT(' MIDAS FOUND TO BE BUSY'
     &/' STATUS WORD IS',F4.0
     &/' CRASH ON, TRY CHECKING MIDAS AGAIN')
      GO TO 50
C
C  MIDAS IS NOT BUSY
C
   20 IF(NDONE.EQ.1)RETURN
C

C  WRITE AND READ MIDAS
C
      CALL MIDAS(U,O,R,FS,S)
      NDONE=1
C
C  WAIT TILL TRANSACTION IS COMPLETED
C
      GO TO 50
      END
```

```
$ASSM
WAYT PROG
$FORT
$TRGT 16
C ******************************************************************
C
      SUBROUTINE WAYT(PCHISQ,PTIME,INPUT,NRANGE,SCALE,TIMEST,
     &CHISQR,BIASV,NREADS,NTERMS,NERROR)
C
C ******************************************************************
C
C  SUBROUTINE TO WAIT UNTIL READING STEADIES DOWN
C
C  PCHISQ = EQUILIBRIUM VALUE FOR PARAMETER CHI-SQUARE
C  PTIME = MAXIMUM STEADYING TIME ALLOWED
C  INPUT = INPUT NUMBER ON MIDAS ANALOG SCANNER
C  NRANGE = INPUT VOLTAGE RANGE ON DVM
C  SCALE = CONVERSION FACTOR FROM MIDAS READING TO PHYSICAL QUANTITY
C    DEPENDS ON DVM RANGE AND ANY CONVERSION IN UNITS
C  TIMEST = TIME TO EQUILIBRATE AS OBSERVED HERE
C  CHISQR = EQUILIBRIUM CHI-SQUARE VALUE AS OBSERVED HERE
C  BIASV = BIAS VOLTAGE ON VOLTAGE INPUT TO DVM
C  NREADS = NUMBER OF READINGS TO TAKE IN ASSESSING AVERAGE VALUE
C  NTERMS = NUMBER OF TERMS IN POLYNOMIAL
C  NERROR = ERROR INDICATOR AS RETURNED FROM SUBROUTINE RPAR
C
C  NEEDS THE FOLLOWING COMMON BLOCKS, SUBROUTINES AND FUNCTIONS:
C    COMMON /REGRS/
C    REGRES
C    RPAR
C    TIMER
C
C  COMMON STORAGE DEFINED VIA ARRAY =DUMP=
C
      COMMON DUMP(900)
C
C  COMMON /REGRES/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
      DIMENSION P(5),SP(5),NTYME(3),TYME(5),A(5)
     &,SIGMA(5)
C
C
      NERROR=0
      BEGIN=TIMER(NTYME)
C
C  READ INPUT IN BLOCKS OF 5 READINGS, 5 SECONDS BETWEEN READINGS
C
 30   DO 20 K=1,5
      CALL RPAR(P(K),SP(K),INPUT,NRANGE,SCALE,NREADS,BIASV,NERROR)
      IF(NERROR.GT.0)RETURN
      TYME(K)=TIMER(NTYME)-BEGIN
C
C  WAIT 5 SECONDS BETWEEN READINGSS
```

```
C
   35 TNOW=TIMER(NTYME)-BEGIN
      IF((TNOW-TYME(K)).LT.5.)GO TO 35
   20 CONTINUE
      TIMEST=TYME(5)
      DO 40 I=1,5
   40 TYME(I)=TYME(I)-TYME(3)
      MODE=0
C
C  CALCULATE CHI SQUARE AS A CHECK FOR STEADINESS
C
      CALL REGRES(TYME,P,SP,5,NTERMS,M,MODE,YFIT,A0,A,SIGMA0,SIGMA
     &,R,RMUL,CHISQR,FTEST)
      IF(CHISQR.GT.PCHISQ.AND.TIMEST.LT.PTIME) GO TO 30
C
C  EITHER READING IS STEADY OR ELAPSED TIME IS TOO GREAT.
C
      IF(TIMEST.GT.PTIME)NERROR=3
      RETURN
      END
```

```
$ASSM
WTAVER PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE WTAVER (E,SIGE,NDATA,WTEAVE,SIGWTE)
C
C ****************************************************************
C  SUBROUTINE TO CALCULATE AVERAGE E (WTEAVE) AND STANDARD DEVIATION (SIGWTE)
C    FROM INPUT LIST E AND SIGE CONTAINING =NDATA=VALUES
C  ALSO CALCULATE UNWEIGHTED AVERAGE AND 95% CONFIDENCE RANGES
C
C  NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C    COMMON /IO/
C    COMMON /REGRS/
C
C  NEEDS SUBROUTINES:
C    AVSDN
C    CHISQ
C    REGRES
C
C  COMMON DEFINED BY ARRAY =DUMP=
      COMMON DUMP(900)
C
C
C  COMMON /IO/
C
      EQUIVALENCE (NTTYI,DUMP(123)),(NTTYO,DUMP(124))
     &,(NPRINT,DUMP(125)),(NFILEA,DUMP(126)),(NFILEB,DUMP(127))
     &,(NFILED,DUMP(128)),(NFILEE,DUMP(129))
C
C  COMMON /REGRS/
C
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE(M(1),DUMP(530)),(R(1),DUMP(535)),(YFIT(1),DUMP(540))
C
C
      DIMENSION E(NDATA),SIGE(NDATA),A(5),SIGA(5)
C
C  CALCULATE AVERAGE UNWEIGHTED E
C
      WTEAVE=0.
      SIGWTE=1.E-09
      EAVE=0.
      ESIG=1.E-09
      SIGAVE=1.E-09
      FREE=0.
      CHIWTE=0.
      CHIUNW=0.
      IF(NDATA.LE.0)RETURN
      IF(NDATA.GT.1)GO TO 200
      WTEAVE=E(1)
      SIGWTE=SIGE(1)
      SIGAVE=SIGE(1)
      EAVE=E(1)
```

```
      ESIG=SIGE(1)
      GO TO 300
  200 CONTINUE
      CALL AVSDN(EAVE,ESIG,E,NDATA)
      CALL CHISQ(E,SIGE,CHIUNW,NDATA)
C
C  CALCULATE AVERAGE SIGE
C
      CALL AVSDN(SIGAVE,SIGSIG,SIGE,NDATA)
C
C  CALCULATE WEIGHTED AVERAGE OF E
C
      CALL REGRES(E,E,SIGE,NDATA,0,M,1,YFIT,A0,A,SIGA0,SIGA
     &,R,RMUL,CHIWTE,FTEST)
      WTEAVE=A0
C
C  CALCULATE SIGMA FOR WEIGHTED E AVERAGE
C
      SIGWTE=0.
      DO 10 I=1,NDATA
   10 SIGWTE=SIGWTE+1./SIGE(I)**2
      SIGWTE=SQRT(1./SIGWTE)
      FDATA=NDATA
      FREE=NDATA-1
      ESIG=ESIG/SQRT(FDATA)
C
C  CALCULATE 95% CONFIDENCE RANGE
C
  300 T=1.96*SIGWTE
      WTEHI=WTEAVE+T
      WTELO=WTEAVE-T
      T=1.96*ESIG
      UNWEHI=EAVE+T
      UNWELO=EAVE-T
C
C  OUTPUT AVERAGES
C
      IF(NPRINT.LT.3) GO TO 100
      WRITE (NPRINT,30) WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
30    FORMAT (/' WEIGHTED MEAN E IS ',F7.3,' + OR - ',F7.3/
     2 ' REDUCED CHISQ FOR E CALCULATION IS',F7.3,' (SHOULD BE 1.)'/
     3 ' UNWEIGHTED MEAN OF SIGMAS IS ',F7.3,/
     4,' CALCULATED OVER ',F5.0,' DEGREES OF FREEDOM')
C
C  OUTPUT CONFIDENCE LIMITS
C
      WRITE (NPRINT,40) WTELO,WTEHI
40    FORMAT (/' ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY'/
     2 ' DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE',F7.3,' TO',
     3 F7.3/' FOR 95% CONFIDENCE LEVEL')
      WRITE (NPRINT,50) EAVE,ESIG,UNWELO,UNWEHI,CHIUNW
50    FORMAT (/' UNWEIGHTED ESTIMATE OF E AND UNBIASED ESTIMATE OF'/
     2 ' SIG E ARE',F7.3,' AND',F7.3,' WHICH GIVES CONFIDENCE RANGE'/
     3 ' OF',F7.3,' TO',F7.3
     &/' CHI-SQUARE FOR UNWEIGHTED E IS',F7.2)
  100 IF(NFILEB.LE.6)RETURN

      WRITE(NFILEB,30)WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
      WRITE(NFILEB,40)WTELO,WTEHI
      WRITE(NFILEB,50)EAVE,ESIG,UNWELO,UNWEHI,CHIUNW
      RETURN
      END
```

File LOTGRN, which automatically assigns all files needed to assemble the TGRUNF overlays and drives the LIBLDR program to make the overlays. To reassemble the program, type DO LOTGRN,DO.

```
BI 0
AC TGMAIN.1D0
AC TGRUNB.6D0
AC OVERLB.7D0
AC TIMEB.8D0
AC TRAMIB.9D0
AC DEKODB.AD0
AC MIDAS.BD0
AC REGREB.CD0
AC FCTNB.DD0
AC MATINB.ED0
AC FORLIB.FD0
AC TGRNL5.5D0
AS 362
RUN LIBLDR.D0
CL
AC TGSETU.1D0
AC INITB.6D0
AC PRIMEB.7D0
AC THMCPB.8D0
AC DESYNB.9D0
ST
AC RESPOB.6D0
AC WAYTB.7D0
AC SETPAB.8D0
AC RPARB.9D0
AC RMIDAB.AD0
AC AVSDNB.BD0
AC ENKODB.CD0
ST
AC BIASVB.6D0
AC SETPAB.7D0
AC ENKODB.8D0
AC RPARB.9D0
AC RMIDAB.AD0
AC AVSDNB.BD0
AC WAYTB.CD0
ST
CL
AC TGPLAT.1D0
AC PLATTB.6D0
AC FACFUB.7D0
AC RPARB.8D0
AC RMIDAB.9D0
AC AVSDNB.AD0
ST
CL
AC TGFACT.1D0
AC FACSEB.6D0
AC SETPAB.7D0
AC ENKODB.8D0
AC BAKEB.9D0
ST
CL
AC TGPREC.1D0
AC PRECIB.6D0
AC EXTPOB.7D0
AC RESIDB.8D0
ST
CL
AC TGSTAT.1D0
AC ACTIVB.6D0
AC WTAVEB.7D0
AC AVSDNB.8D0
AC CHISOB.9D0
AC TDEGBB.AD0
ST
CL
ZU
TR
```

File TGRNL5, which is assigned by LOTGRN during assembly of the TGRUNF overlays. This file contains the appropriate responses to the LIBLDR program.

```
OUT 1 TGRUN
BI 3576
BC 0E10
LO 6
LI 7
LI 8
LI 9
LI A
LI B
LI C
LI D
LI E
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 OSETU1
OV
LI 6
LI 7
LI 8
LI 9
RW F
ED F
MAP
MAP 3
XOUT
END
OUT 1 OSETU2
OV
LI 6
LI 7
LI 8
LI 9
LI A
LI B
LI C
RW F
ED F
MAP
MAP 3
XOUT
END
```

```
OUT 1 OSETU3
OV
LI 6
LI 7
LI 8
LI 9
LI A
LI B
LI C
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 OPLATT
OV
LI 6
LI 7
LI 8
LI 9
LI A
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 OFACTS
OV
LI 6
LI 7
LI 8
LI 9
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
```

```
OUT 1 OPRECS
OV
LI 6
LI 7
LI 8
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 OSTATS
OV
LI 6
LI 7
LI 8
LI 9
LI A
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
```

Memory map of the base segment, TGMAIN, of the TGRUNF program.

```
REL PROGS:
  3576 TGRUN      5202 OVERLY     5400 TIMER      5574 TRAMID
  5818 DEKODE     5DD6 REGRES     739E FCTN       7426 MATINV
  7C24 IFETCH     7C96 .D         7CA4 .COMEX     7E72 DLOG
  81E2 DEXP       83BA DSQRT      85BC $7         861A $9
  8652 DAINT      8716 @A         8730 @1         873E @S
  8760 DABS       877A .G         8790 @F         8896 @L
  88C6 @K         88F6 @M         8A92 @2         8AA0 @D
  8BBA @T         8BE2 @E         8C06 @0         8C22 @N
  8C7C @U         8C94 @V         8CB4 @P         8CE4 @B
  8CF4 .DARG      8D36 .DSWAP     8D92 .DRTN      8DBA .R
  8DF6 .A         8E9C SQRT       8FA4 ALOG       9082 EXP
  91B8 AINT       9230 IFIX       9236 IFIX2      9254 .Y2
  92D4 ABS        92EA .W         9338 .COMP      9364 $6
  93AE .RARG      93DA $8         9404 .5         940C .ZERO
  9414 .H         9466 .S         949C .P         951E .Q
  95CE .O         9616 .MES       968C .U         96BC .V
  96CA @R         96F0 @Z         9716 @H         A960

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY     5424 TIMER      5598 TRAMID     583C DEKODE
  5B86 RELESE     5BCA MIDAS      5DFA REGRES     73C2 FCTN
  744A MATINV     7C26 IFETCH     7C98 .D         7CA6 .COMEX
  7E74 DLOG       81E4 DEXP       83BC DSQRT      85BE $7
  861C $9         8654 DAINT      8718 @A         8732 @1
  8740 @S         8762 DABS       877C .G         8792 @F
  8898 @L         88C8 @K         88F8 @M         8A94 @2
  8AA2 @D         8BBC @T         8BE4 @E         8C08 @0
  8C24 @H         8C7E @U         8C96 @V         8CB6 @P
  8CE6 @B         8CF6 .DARG      8D38 .DSWAP     8D94 :DRTN
  8DBC .R         8DF8 .A         8E9E SQRT,      8FA6 ALOG
  9084 EXP        91BA AINT       9232 IFIX       9238 IFIX2
  9256 .Y2        92D6 ABS        92EC .W         933A .COMP
  9366 $6         93B0 .RARG      93DC $8         9408 .5
  9410 .ZERO      9416 .H         9468 .S         949E .P
  9520 .Q         95D0 .O         9618 .MES       969A .U
  96C2 .V         96CC @R         96F2 @Z         9764 @H

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
   INIT      PRIME     DESYNE    RESPON    BIASVL    BAKEIT
   FACSET    PLATTO    PRECIS    PLATCO    EXTPOL    ACTIVE
   WTAVER    SETPAR
```

Memory map for the initialization overlay, part 1 of TGSETU.

```
REL PROGS:
  3576 TGRUN    5202 OVERLY   5490 TIMER    5574 TRAMID
  5818 DEKODE   5DD6 REGRES   739E FCTN     7426 MATINV
  7C24 IFETCH   7C96 .D       7CA4 .COMEX   7E72 DLOG
  81E2 DEXP     83BA DSQRT    858C $7       861A $9
  8652 DAINT    8716 @A       8730 @1       873E @S
  8760 DABS     877A .G       8790 @F       8896 @L
  88C6 @K       88F6 @M       8A92 @2       8AA0 @D
  8BBA @T       8BE2 @E       8C06 @0       8C22 @N
  8C7C @U       8C94 @V       8CB4 @P       8CE4 @B
  8CF4 .DARG    8D36 .DSWAP   8D92 .DRTN    8DBA .R
  8DF6 .A       8E9C SQRT     8FA4 ALOG     90B2 EXP
  91B8 AINT     9230 IFIX     9236 IFIX2    9254 .Y2
  92D4 ABS      92EA .W       9338 .COMP    9364 $6
  93AE .RARG    93DA $8       9404 .5       940C .ZERO
  9414 .H       9466 .S       949C .P       951E .Q
  95CE .O       9616 .MES     968C .U       96BC .V
  96CA @R       96F0 @Z       9716 @H       AAC8 INIT
  AE3A PRIME    C362 THMCPL   C71E DESYNE   D4C4 MIN0
  D4CA MIN02    D4E0 .1       D4F2 $1       D50A $22
  D53E IABS     D544 IABS2    D57E SIGN     D5BC .RRARG
  D5FA

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY   5424 TIMER    5598 TRAMID   583C DEKODE
  5886 RELESE   5BCA MIDAS    5DFA REGRES   73C2 FCTN
  744A MATINV   7C26 IFETCH   7C98 .D       7CA6 .COMEX
  7E74 DLOG     81E4 DEXP     83BC DSQRT    858E $7
  861C $9       8654 DAINT    8718 @A       8732 @1
  8740 @S       8762 DABS     877C .G       8792 @F
  8898 @L       88C8 @K       88F8 @M       8A94 @2
  8AA2 @D       8BBC @T       8BE4 @E       8C08 @0
  8C24 @N       8C7E @U       8C96 @V       8CB6 @P
  8CE6 @B       8CF6 .DARG    8D38 .DSWAP   8D94 .DRTN
  8DBC .R       8DF8 .A       8E9E SQRT     8FA6 ALOG
  90B4 EXP      91BA AINT     9232 IFIX     9238 IFIX2
  9256 .Y2      92D6 ABS      92EC .W       933A .COMP
  9366 $6       93B0 .RARG    93DC $8       9408 .5
  9410 .ZERO    9416 .H       9468 .S       949E .P
  9520 .Q       95D0 .O       9618 .MES     969A .U
  96C2 .V       96CC @R       96F2 @Z       9764 @H
  AAEC INIT     AE5E PRIME    C386 THMCPL   C742 DESYNE
  D4C6 MIN0     D4CC MIN02    D4E2 .1       D4F4 $1
  D50C $22      D540 IABS     D546 IABS2    D580 SIGN
  D5BE .RRARG

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
    RESPON    BIASVL    BAKEIT    FACSET    PLATTO    PRECIS
    PLATCO    EXTPOL    ACTIVE    WTAVER    SETPAR
```

Memory map for the factor-response determining overlay, part 2 of TGSETU.

```
REL PROGS:
  3576 TGRUN     5202 OVERLY    5400 TIMER     5574 TRAMID
  5818 DEKODE    5DD6 REGRES    739E FCTN      7425 MATINV
  7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
  81E2 DEXP      83BA DSQRT     85DC $7        861A $9
  8652 DAINT     8716 @A        8730 @1        873E @S
  8760 DABS      877A .G        8790 @F        8896 @L
  88C6 @K        88F6 @M        8A92 @2        8AA0 @D
  8BBA @T        8BE2 @E        8C06 @O        8C22 @N
  8C7C @U        8C94 @V        8CB4 @P        8CE4 @B
  8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8DBA .R
  8DF6 .A        8E9C SQRT      8FA4 ALOG      9082 EXP
  91B8 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
  92D4 ABS       92EA .W        9338 .COMP     9364 $5
  93AE .RARG     93DA $8        9404 .5        940C .ZERO
  9414 .H        9466 .S        949C .P        951E .Q
  95CE .O        9616 .MES      968C .U        968C .V
  96CA @R        96F0 @Z        9716 @H        AAC8 RESPON
  BA12 WAYT      BCE4 SETPAR    C26C RPAR      C42E RMIDAS
  D322 AVSDN     D4AA ENKODE    D62C MIN02     D642 MAX0
  D648 MAX02     D65E .1        D670 MIN1      D676 MIN12
  D68C .4        D6AE $1        D6C6 $22       D6FA $3
  D726 FLOAT     D72C FLOAT2    D764

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY    5424 TIMER     5598 TRAMID    583C DEKODE
  5B86 RELESE    5BCA MIDAS     5DFA REGRES    73C2 FCTN
  744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
  7E74 DLOG      81E4 DEXP      83BC DSQRT     85DE $7
  861C $9        8654 DAINT     8718 @A        8732 @1
  8740 @S        8762 DABS      877C .G        8792 @F
  8898 @L        88C8 @K        88F8 @M        8A94 @2
  8AA2 @D        8BBC @T        8BE4 @E        8C08 @O
  8C24 @N        8C7E @U        8C96 @V        8CB6 @P
  8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
  8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
  9084 EXP       91BA AINT      9232 IFIX      9238 IFIX2
  9256 .Y2       92D6 ABS       92EC .W        933A .COMP
  9366 $6        93B0 .RARG     93DC $8        9408 .5
  9410 .ZERO     9416 .H        9468 .S        949E .P
  9520 .Q        95D0 .O        9618 .MES      969A .U
  96C2 .V        96CC @R        96F2 @Z        9764 @H
  AAEC RESPON    BA36 WAYT      BD08 SETPAR    C290 RPAR
  C452 RMIDAS    D346 AVSDN     D4CE ENKODE    D62E MIN02
  D644 MAX0      D64A MAX02     D660 .1        D672 MIN1
  D678 MIN12     D68E .4        D6B0 $1        D6C8 $22
  D6FC $3        D728 FLOAT     D72E FLOAT2

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
   BIASVL    BAKEIT    FACSET    PLATTO    PRECIS    PLATCO
   EXTPOL    ACTIVE    WTAVER
```

Memory map for TGSETU, part 3, which determines bias voltages.

```
REL PROGS:
   3576 TGRUN     5202 OVERLY    5400 TIMER     5574 TRAMID
   5818 DEKODE    5DD6 REGRES    739E FCTN      7425 MATINV
   7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
   81E2 DEXP      83BA DSQRT     858C $7        861A $9
   8652 DAINT     8716 @A        8730 @1        873E @S
   8760 DABS      877A .G        8790 @F        8896 @L
   88C6 @K        88F6 @M        8A92 @2        8AA0 @D
   8B3A @T        8BE2 @E        8C06 @O        8C22 @N
   8C7C @U        8C94 @V        8CB4 @P        8CE4 @B
   8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8DBA .R
   8DF6 .A        8E9C SQRT      8FA4 ALOG      90B2 EXP
   91B8 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
   92D4 ABS       92EA .W        9338 .COMP     9364 $6
   93AE .RARG     93DA $8        9404 .5        940E .ZERO
   9414 .H        9466 .S        949C .P        951E .Q
   95CE .O        9616 .MES      9698 .U        96BC .V
   96CA @R        96F0 @Z        9716 @H        AAC8 BIASVL
   B844 SETPAR    BDCC ENKODE    BF4E RPAR      C110 RMIDAS
   D004 AVSDN     D18C WAYT      D45E MIN02     D474 MAX0
   D47A MAX02     D490 .1        D4A2 MIN1      D4A8 MIN12
   D4BE .4        D4E0 $1        D4F8 $22       D52C $3
   D558 FLOAT     D55E FLOAT2    D596

ABS PROGS:
NONE

ENTRY-POINTS:
   5226 OVERLY    5424 TIMER     5598 TRAMID    583C DEKODE
   5B86 RELESE    5BCA MIDAS     5DFA REGRES    73C2 FCTN
   744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
   7E74 DLOG      81E4 DEXP      83BC DSQRT     858E $7
   861C $9        8654 DAINT     8718 @A        8732 @1
   8740 @S        8762 DABS      877C .G        8792 @F
   8898 @L        88C8 @K        88F8 @M        8A94 @2
   8AA2 @D        8BBC @T        8BE4 @E        8C08 @O
   8C24 @N        8C7E @U        8C96 @V        8CB6 @P
   8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
   8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
   90B4 EXP       91BA AINT      9232 IFIX      9238 IFIX2
   9256 .Y2       92D6 ABS       92EC .W        933A .COMP
   9366 $6        93B0 .RARG     93DC $8        9406 .5
   9410 .ZERO     9416 .H        9468 .S        949E .P
   9520 .Q        95D0 .O        9618 .MES      969A .U
   96C2 .V        96CC @R        96F2 @Z        9764 @H
   AAEC BIASVL    B868 SETPAR    BDF0 ENKODE    BF72 RPAR
   C134 RMIDAS    D028 AVSDN     D1B0 WAYT      D460 MIN02
   D476 MAX0      D47C MAX02     D492 .1        D4A4 MIN1
   D4AA MIN12     D4C0 .4        D4E2 $1        D4FA $22
   D52E $3        D55A FLOAT     D560 FLOAT2

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
   BAKEIT   FACSET   PLATTO   PRECIS   PLATCO   EXTPOL
   ACTIVE   WTAVER
```

Memory map for TGPLAT, the data collection overlay.

```
REL PROGS:
  3576 TGRUN     5202 OVERLY    5420 TIMER     5574 TRAMID
  5818 DEKODE    5DD6 REGRES    739E FCTN      7426 MATINV
  7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
  81E2 DEXP      833A DSQRT     85BC $7        861A $9
  8652 DAINT     8716 @A        8730 @1        873E @S
  8760 DABS      877A .G        8790 @F        8896 @L
  88C6 @K        88F6 @M        8A92 @2        8AA0 @D
  8B8A @T        8BE2 @E        8C06 @O        8C22 @N
  8C7C @U        8C94 @V        8C94 @P        8CE4 @B
  8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8D9A .R
  8DF6 .A        8E9C SQRT      8FA4 ALOG      9082 EXP
  91B8 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
  92D4 ABS       92EA .W        9338 .COMP     9364 $6
  93AE .RARG     93DA $8        9404 .5        940C .ZERO
  9414 .H        9466 .S        949C .P        951E .Q
  95CE .O        9616 .MES      968C .U        96BC .V
  96CA @R        96F0 @Z        9716 @H        AAC8 PLATTO
  C3B0 FACFUN    C4F6 RPAR      C699 RMIDAS    D5AC AVSDN
  D734 BTEST     D752 BSET      D788 .BITPS    D7E4 MIN0
  D7EA MIN02     D800 .1        D812 $1        D82A $22
  D85E FLOAT     D864 FLOAT2    D89C MOD       D8A2 MOD2
  D8D0 .IIARG    D90A .M        D98E

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY    5424 TIMER     5598 TRAMID    593C DEKODE
  5B86 RELESE    5ECA MIDAS     5DFA REGRES    73C2 FCTN
  744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
  7E74 DLOG      81E4 DEXP      833C DSQRT     853E $7
  861C $9        8654 DAINT     8718 @A        8732 @1
  8740 @S        8762 DABS      877C .G        8792 @F
  8898 @L        88C8 @K        88F8 @M        8A94 @2
  8AA2 @D        88BC @T        88E4 @E        8C08 @O
  8C24 @N        8C7E @U        8C96 @V        8C96 @P
  8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
  8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
  9084 EXP       91BA AINT      9232 IFIX      9238 IFIX2
  9256 .Y2       92D6 ABS       92EC .W        933A .COMP
  9366 $6        93B0 .RARG     93DC $8        9408 .5
  9410 .ZERO     9416 .H        9468 .S        949E .P
  9520 .Q        95D0 .O        9618 .MES      969A .U
  96C2 .V        96CC @R        96F2 @Z        9764 @H
  AAEC PLATTO    ADE2 PLATCO    C3D4 FACFUN    C51A RPAR
  C6DC RMIDAS    D5D0 AVSDN     D736 BTEST     D754 BSET
  D78A .BITPS    D7E6 MIN0      D7EC MIN02     D802 .1
  D814 $1        D82C $22       D860 FLOAT     D866 FLOAT2
  D89E MOD       D8A4 MOD2      D8D2 .IIARG    D90C .M

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
   BAKEIT    FACSET    PRECIS    EXTPOL    ACTIVE    WTAVER
```

Memory map for TGFACT, the factor setting overlay.

```
REL PROGS:
   3576 TGRUN     5202 OVERLY    5400 TIMER     5574 TRAMID
   5818 DEKODE    5DD6 REGRES    739E FCTN      7426 MATINV
   7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
   81E2 DEXP      83BA DSQRT     85BC $7        861A $9
   8652 DAINT     8716 @A        8730 @1        873E @S
   8760 DABS      877A .G        8790 @F        8896 @L
   88C6 @K        88F6 @M        8A92 @2        8AA0 @D
   8BBA @T        8BE2 @E        8C06 @O        8C22 @N
   8C7C @U        8C94 @V        8CB4 @P        8CE4 @B
   8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8DBA .R
   8DF6 .A        8E9C SQRT      8FA4 ALOG      9082 EXP
   91B8 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
   92D4 ABS       92EA .W        9338 .COMP     9364 $5
   93AE .RARG     93DA $8        9404 .5        940E .ZERO
   9414 .H        9466 .S        943C :P        951E .Q
   95CE .O        9616 .MES      968C .U        96BC .V
   96CA @R        96F0 @Z        9716 @H        AAC8 FACSET
   B01C SETPAR    B5A4 ENKODE    B726 BAKEIT    C224 MAX0
   C22A MAX02     C240 .1        C252 MIN1      C258 MIN12
   C26E .4        C290 $1        C2A8 $22       C2DC $3
   C308

ABS PROGS:
NONE

ENTRY-POINTS:
   5226 OVERLY    5424 TIMER     5598 TRAMID    583C DEKODE
   5B86 RELESE    5BCA MIDAS     5DFA REGRES    73C2 FCTN
   744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
   7E74 DLOG      81E4 DEXP      83BC DSQRT     85BE $7
   861C $9        8654 DAINT     8718 @A        8732 @1
   8740 @S        8762 DABS      877C .G        8792 @F
   8898 @L        88C8 @K        88F8 @M        8A94 @2
   8AA2 @D        8BBC @T        8BE4 @E        8C08 @O
   8C24 @N        8C7E @U        8C96 @V        8CB6 @P
   8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
   8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
   9084 EXP       91BA AINT      9232 IFIX      9238 IFIX2
   9256 .Y2       92D6 ABS       92EC .W        933A .COMP
   9366 $5        93B0 .RARG     93DC $8        9406 .5
   9410 .ZERO     9416 .H        9468 .S        943E .P
   9520 .Q        95D0 .O        9618 .MES      969A .U
   96C2 .V        96CC @R        96F2 @Z        9754 @H
   AAEC FACSET    B040 SETPAR    B5C8 ENKODE    B74A BAKEIT
   C226 MAX0      C22C MAX02     C242 .1        C254 MIN1
   C25A MIN12     C270 .4        C292 $1        C2AA $22
   C2DE $3

COMMON-BLOCKS:
   F1EE //

UNDEFINED:
    PRECIS     EXTPOL     ACTIVE     WTAVER
```

Memory map for TGPREC, the polynomial fitting and extrapolation overlay.

```
REL PROGS:
  3576 TGRUN     5202 OVERLY    5400 TIMER     5574 TRAMID
  5818 DEKODE    5DD6 REGRES    739E FCTN      7425 MATINV
  7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
  81E2 DEXP      83BA DSQRT     853C $7        861A $9
  8652 DAINT     8716 @A        8730 @1        873E @S
  8760 DABS      877A .G        8790 @F        8896 @L
  88C6 @K        88F6 @M        8A92 @2        8AA0 @D
  8B8A @T        8BE2 @E        8C06 @O        8C22 @N
  8C7C @U        8C94 @V        8C84 @P        8CE4 @B
  8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8DBA .R
  8DF6 .A        8E9C SQRT      8FA4 ALOG      9082 EXP
  91B8 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
  92D4 ABS       92EA .W        9338 .COMP     9364 $5
  93AE .RARG     93DA $3        9404 .5        940C .ZERO
  9414 .H        9466 .S        949C .P        951E .Q
  95CE .O        9616 .MES      968C .U        96BC .V
  96CA @R        96F0 @Z        9716 @H        AAC8 PRECIS
  CE96 EXTPOL    D3F0 RESIDS    D78C BTEST     D7AA BSET
  D7E0 .BITPS    D83C AMAX1     D850 .2        D862 $1
  D87A $3        D8A6 FLOAT     D8AC FLOAT2    D8E4

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY    5424 TIMER     5598 TRAMID    583C DEKODE
  5B86 RELESE    5BCA MIDAS     5DFA REGRES    73C2 FCTN
  744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
  7E74 DLOG      81E4 DEXP      83BC DSQRT     853E $7
  861C $9        8654 DAINT     8718 @A        8732 @1
  8748 @S        8762 DABS      877C .G        8792 @F
  8898 @L        88C8 @K        88F8 @M        8A94 @2
  8AA2 @D        8BBC @T        8BE4 @E        8C08 @O
  8C24 @N        8C7E @U        8C96 @V        8C86 @P
  8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
  8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
  9084 EXP       91BA AINT      9232 IFIX      9238 IFIX2
  9256 .Y2       92D6 ABS       92EC .W        933A .COMP
  9366 $5        93B0 .RARG     93DC $3        9408 .5
  9410 .ZERO     9416 .H        9468 .S        949E .P
  9520 .Q        95D0 .O        9618 .MES      969A .U
  96C2 .V        96CC @R        96F2 @Z        9764 @H
  AAEC PRECIS    CEBA EXTPOL    D414 RESIDS    D78E BTEST
  D7AC BSET      D7E2 .BITPS    D83E AMAX1     D852 .2
  D864 $1        D87C $3        D8A8 FLOAT     D8AE FLOAT2

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
   ACTIVE    WTAVER
```

Memory map for TGSTAT, the overlay which calculates the activation energy and associated statistics.

```
REL PROGS:
  3576 TGRUN     5202 OVERLY    5400 TIMER     5574 TRAMID
  5818 DEKODE    5DD6 REGRES    739E FCTN      7425 MATINV
  7C24 IFETCH    7C96 .D        7CA4 .COMEX    7E72 DLOG
  81E2 DEXP      838A DSQRT     858C $7        861A $9
  8652 DAINT     8716 @A        8730 @1        873E @S
  8760 DABS      877A .G        8790 @F        8895 @L
  88C6 @K        88F6 @M        8A92 @2        8AA0 @D
  8BBA @T        8BE2 @E        8C06 @O        8C22 @N
  8C7C @U        8C94 @V        8CB4 @P        8CE4 @B
  8CF4 .DARG     8D36 .DSWAP    8D92 .DRTN     8DBA .R
  8DF6 .A        8E9C SQRT      8FA4 ALOG      9082 EXP
  91B3 AINT      9230 IFIX      9236 IFIX2     9254 .Y2
  92D4 ABS       92EA .W        9338 .COMP     9364 $6
  93AE .RARG     93DA $8        9404 .5        940C .ZERO
  9414 .H        9456 .S        949C .P        951E .Q
  95CE .O        9616 .MES      968C .U        96BC .V
  96CA @R        96F0 @Z        9716 @H        AAC8 ACTIVE
  B35E WTAVER    BAD8 AVSDN     BC60 CHISQ     BDBE TDEGAB
  BE6C FLOAT     BE72 FLOAT2    BEAA

ABS PROGS:
NONE

ENTRY-POINTS:
  5226 OVERLY    5424 TIMER     5598 TRAMID    593C DEKODE
  5B86 RELESE    5BCA MIDAS     5DFA REGRES    73C2 FCTN
  744A MATINV    7C26 IFETCH    7C98 .D        7CA6 .COMEX
  7E74 DLOG      81E4 DEXP      838C DSQRT     858E $7
  861C $9        8654 DAINT     8718 @A        8732 @1
  8740 @S        8762 DABS      877C .G        8792 @F
  8898 @L        88C8 @K        88F8 @M        8A94 @2
  8AA2 @D        8BBC @T        8BE4 @E        8C08 @O
  8C24 @N        8C7E @U        8C96 @V        8CB6 @P
  8CE6 @B        8CF6 .DARG     8D38 .DSWAP    8D94 .DRTN
  8DBC .R        8DF8 .A        8E9E SQRT      8FA6 ALOG
  9084 EXP       91BA AINT      9232 IFIX      9238 IFIX2
  9256 .Y2       92D6 ABS       92EC .W        933A .COMP
  9366 $6        93B0 .RARG     93DC $8        9408 .5
  9410 .ZERO     9416 .H        9468 .S        949E .P
  9520 .Q        95D0 .O        9618 .MES      969A .U
  96C2 .V        96CC @R        96F2 @Z        9764 @H
  AAEC ACTIVE    B382 WTAVER    BAFC AVSDN     BC84 CHISQ
  BDE2 TDEGAB    BE6E FLOAT     BE74 FLOAT2

COMMON-BLOCKS:
  F1EE //

UNDEFINED:
NONE
```

Appendix B. TGEDIT, a FORTRAN program to edit factor-jump thermogravimetry results.

Example of how to run the TGEDIT program. User input is underlined.

```
*AC PEVAC3,7D1
*AL PEVAC3,C[illegible],1,80
*RUN TGEDIT,[illegible]
 CORRECTIONS TO TEMPERATURES OR NOT?
  GIVE Y FOR YES, N FOR NO
N

 GIVE:

 1)   MINIMUM DEGREE OF REACTION IN %
 2)   MAXIMUM DEGREE OF REACTION IN %
 3)   MINIMUM RATE (ABSOLUTE VALUE IS OK)
 4)   MAXIMUM % ERROR IN RATE (=100*SIGR/RATE)
 5)   MINIMUM TEMPERATURE IN DEGREES CELSIUS
 6)   MAXIMUM TEMPERATURE
 7)   MAXIMUM ERROR IN TEMPERATURE
 8)   MAXIMUM ERROR IN ACTIVATION ENERGY
  $$$ ALL ON SEPARATE LINES $$$

0.
100.
11.
2.
0.
500.
.1
1.75
  INPUT VALUES ARE:     0.00  100.00   11.00    2.00
                        0.00  500.00    0.10    1.75
  IS THIS OK? Y OR N
Y
```

A sample file, prepared by the TGRUNF program, and used as input to the TGEDIT program. The activation energies rejected by TGEDIT are marked in the first file with *. There has been some slight round-off in the E values.

```
NBS 706 IN 10% O2 P=200 3/19/78 AM

 1 27.620  0.324      973.   0.369732E+06  11.43
                    310.79  0.04    -0.5074E+02      0.5855E+00
                                     0.368127E+06   0.139414E+03    30
                    290.86  0.02    -0.2619E+02      0.1714E-01
                                     0.371336E+06   0.408180E+01    30
 2 24.710  0.074     1836.   0.349127E+06  14.78
                    291.08  0.02    -0.2418E+02      0.1583E-01
                                     0.349614E+06   0.351631E+01    30
                    302.15  0.01    -0.3695E+02      0.2636E-01
                                     0.348641E+06   0.585567E+01    30
 3 24.514  0.101     2663.   0.319670E+06  19.57
                    302.02  0.01    -0.3401E+02      0.2636E-01
                                     0.319295E+06   0.585567E+01    30
                    292.92  0.02    -0.2409E+02      0.1998E-01
                                     0.320043E+06   0.443929E+01    30
 4 25.546  0.084     3490.   0.300236E+06  22.73
                    293.05  0.02    -0.2266E+02      0.1998E-01
                                     0.300710E+06   0.443929E+01    30
                    304.13  0.02    -0.3504E+02      0.2146E-01
                                     0.299762E+05   0.476667E+01    30
 5 26.279  0.094     4317.   0.272061E+06  27.31
                    304.04  0.02    -0.3286E+02      0.2146E-01
                                     0.271682E+06   0.476667E+01    30
                    294.94  0.02    -0.2277E+02      0.1694E-01
                                     0.272440E+06   0.376348E+01    30
 6 25.837  0.085     5144.   0.253643E+06  30.30
                    295.04  0.02    -0.2161E+02      0.1694E-01
                                     0.254090E+06   0.376348E+01    30
                    306.14  0.02    -0.3351E+02      0.2452E-01
                                     0.253196E+06   0.547060E+01    30
 7 25.943  0.105     5971.   0.226476E+06  34.72
                    306.01  0.02    -0.3193E+02      0.2452E-01
                                     0.226113E+06   0.547060E+01    30
                    296.95  0.02    -0.2236E+02      0.1677E-01
                                     0.226840E+06   0.372499E+01    30
 8 26.508  0.089     6798.   0.208405E+06  37.66
                    297.10  0.02    -0.2122E+02      0.1677E-01
                                     0.208819E+06   0.372499E+01    30
                    308.18  0.02    -0.3315E+02      0.2356E-01
                                     0.207992E+06   0.523324E+01    30
 9 27.094  0.111     7624.   0.182019E+06  41.95
                    308.05  0.02    -0.3051E+02      0.2356E-01
                                     0.181669E+06   0.523324E+01    30
                    299.04  0.02    -0.2109E+02      0.1709E-01
                                     0.182370E+06   0.379681E+01    30
10 26.121  0.233     8451.   0.165029E+06  44.71
                    299.07  0.02    -0.1988E+02      0.1709E-01
                                     0.165426E+06   0.379681E+01    30
                    310.11  0.01    -0.3071E+02      0.1135E+00
                                     0.164632E+06   0.252085E+02    30
10 26.466  0.313     9279.   0.140820E+06  48.65
                    310.02  0.01    -0.2761E+02      0.1135E+00
                                     0.140517E+06   0.252085E+02    30
                    300.98  0.02    -0.1927E+02      0.1261E-01
```

				0.141123E+05	0.280176E+01	30
10 27.124	0.095	10106.	0.125360E+06	51.16		
		301.11	0.02	-0.1792E+02	0.1261E-01	
				0.125742E+06	0.280176E+01	30
		312.12	0.02	-0.2803E+02	0.2249E-01	
				0.124977E+06	0.499706E+01	30
10 27.814	0.124	10933.	0.103044E+06	54.79		
		312.05	0.02	-0.2572E+02	0.2252E-01	
				0.102739E+06	0.500788E+01	30
		302.98	0.02	-0.1765E+02	0.1409E-01	
				0.103349E+06	0.313343E+01	30
10 27.557	0.097	11761.	0.888427E+05	57.10		
		303.09	0.02	-0.1658E+02	0.1407E-01	
				0.891854E+05	0.312665E+01	30
		314.13	0.02	-0.2606E+02	0.2083E-01	
				0.884999E+05	0.462657E+01	30
10 27.717	0.119	12588.	0.683444E+05	60.43		
		314.01	0.02	-0.2332E+02	0.2083E-01	
				0.680816E+05	0.462657E+01	30
		304.99	0.02	-0.1629E+02	0.1310E-01	
				0.686073E+05	0.290953E+01	30
10 27.823	0.053	13456.	0.545482E+05	62.68		
		305.10	0.02	-0.1476E+02	0.1435E-01	
				0.552179E+05	0.345369E+01	30
		325.17	0.02	-0.3326E+02	0.2233E-01	
				0.538784E+05	0.537358E+01	30
10 27.702	0.130	14324.	0.277493E+05	67.03		
		324.99	0.02	-0.2761E+02	0.2038E-01	
				0.274595E+05	0.452704E+01	30
		315.92	0.02	-0.1929E+02	0.2220E-01	
				0.280391E+05	0.493123E+01	30
10 28.014	0.118	15151.	0.125341E+05	69.51		
		316.03	0.02	-0.1732E+02	0.2220E-01	
				0.129017E+05	0.493123E+01	30
		327.13	0.02	-0.2695E+02	0.2426E-01	
				0.121664E+05	0.539046E+01	30
10 27.920	0.122	15978.	-0.854382E+04	72.93		
		326.97	0.02	-0.2375E+02	0.2426E-01	
				-0.880101E+04	0.539046E+01	30
		317.95	0.01	-0.1662E+02	0.1125E-01	
				-0.828663E+04	0.249952E+01	30
10 27.399	0.084	16805.	-0.219179E+05	75.11		
		318.02	0.02	-0.1559E+02	0.1125E-01	
				-0.216032E+05	0.249952E+01	30
		329.14	0.02	-0.2398E+02	0.1483E-01	
				-0.222327E+05	0.329358E+01	30
10 28.136	0.098	17632.	-0.407054E+05	78.16		
		329.04	0.02	-0.2125E+02	0.1483E-01	
				-0.409376E+05	0.329358E+01	30
		319.94	0.01	-0.1481E+02	0.9052E-02	
				-0.404733E+05	0.201093E+01	30
* 10 28.334	0.048	18499.	-0.535494E+05	80.25		
		320.09	0.01	-0.1390E+02	0.9910E-02	
				-0.529289E+05	0.238214E+01	30
		340.19	0.02	-0.3057E+02	0.2355E-01	
				-0.541520E+05	0.566119E+01	30

10 28.830	0.107	19366.	-0.781598E+05	84.25		
		340.00	0.01	-0.2545E+02	0.2151E-01	
				-0.784489E+05	0.477898E+01	30
		330.96	0.01	-0.1786E+02	0.1266E-01	
				-0.778707E+05	0.281145E+01	30
10 27.673	0.083	20194.	-0.923056E+05	86.55		
		331.05	0.01	-0.1621E+02	0.1267E-01	
				-0.919698E+05	0.281753E+01	30
		342.18	0.02	-0.2460E+02	0.1768E-01	
				-0.926414E+05	0.393167E+01	30
* 10 27.220	0.114	21022.	-0.110599E+06	89.53		
		342.01	0.01	-0.1934E+02	0.1766E-01	
				-0.110820E+06	0.392317E+01	30
		333.05	0.01	-0.1391E+02	0.1005E-01	
				-0.110378E+06	0.223191E+01	30
* 10 12.927	87.368	21986.	-0.124156E+06	91.73		
		333.17	0.02	-0.1164E+02	0.1329E-01	
				-0.122699E+06	0.379578E+01	30
		352.99	0.05	-0.1635E+02	0.3753E+02	
				-0.125613E+06	0.566496E+04	4

NBS 706 IN 10% O2 P=200 3/19/78 AM

1	27.625	0.324	973.	0.369732E+06	11.43		
			310.79	0.04	-0.6074E+02	0.5855E+00	
					0.368127E+06	0.139414E+03	30
			290.86	0.02	-0.2619E+02	0.1714E-01	
					0.371336E+06	0.408180E+01	30
2	24.709	0.074	1836.	0.349127E+06	14.78		
			291.08	0.02	-0.2418E+02	0.1583E-01	
					0.349614E+06	0.351631E+01	30
			302.15	0.01	-0.3695E+02	0.2636E-01	
					0.348541E+06	0.585567E+01	30
3	24.520	0.101	2663.	0.319670E+06	19.57		
			302.02	0.01	-0.3401E+02	0.2636E-01	
					0.319236E+06	0.585567E+01	30
			292.92	0.02	-0.2409E+02	0.1998E-01	
					0.320043E+06	0.443929E+01	30
4	25.552	0.084	3490.	0.300236E+06	22.73		
			293.05	0.02	-0.2266E+02	0.1998E-01	
					0.300710E+06	0.443929E+01	30
			304.13	0.02	-0.3504E+02	0.2146E-01	
					0.299762E+06	0.476667E+01	30
5	26.266	0.094	4317.	0.272061E+06	27.31		
			304.04	0.02	-0.3285E+02	0.2146E-01	
					0.271682E+06	0.476667E+01	30
			294.94	0.02	-0.2277E+02	0.1694E-01	
					0.272440E+06	0.376348E+01	30
6	25.850	0.085	5144.	0.253643E+06	30.30		
			295.04	0.02	-0.2161E+02	0.1694E-01	
					0.254090E+06	0.376348E+01	30
			306.14	0.02	-0.3351E+02	0.2462E-01	
					0.253196E+06	0.547050E+01	30
7	25.938	0.105	5971.	0.226476E+06	34.72		
			306.01	0.02	-0.3199E+02	0.2462E-01	
					0.226113E+06	0.547050E+01	30
			296.95	0.02	-0.2236E+02	0.1677E-01	
					0.226840E+06	0.372499E+01	30
8	26.523	0.089	6798.	0.208405E+06	37.65		
			297.10	0.02	-0.2122E+02	0.1677E-01	
					0.208819E+06	0.372499E+01	30
			308.18	0.02	-0.3315E+02	0.2356E-01	
					0.207992E+06	0.523324E+01	30
9	27.084	0.111	7624.	0.182019E+06	41.95		
			308.05	0.02	-0.3051E+02	0.2356E-01	
					0.181669E+06	0.523324E+01	30
			299.04	0.02	-0.2109E+02	0.1709E-01	
					0.182370E+06	0.379681E+01	30
10	26.125	0.233	8451.	0.165029E+06	44.71		
			299.07	0.02	-0.1988E+02	0.1709E-01	
					0.165426E+06	0.379681E+01	30
			310.11	0.01	-0.3071E+02	0.1135E+00	
					0.164632E+06	0.252085E+02	30
11	26.469	0.313	9279.	0.140820E+06	48.65		
			310.02	0.01	-0.2761E+02	0.1135E+00	
					0.140517E+06	0.252085E+02	30
			300.98	0.02	-0.1927E+02	0.1261E-01	

					0.141123E+06	0.280176E+01	30
12	27.138	0.095	10106.	0.125360E+06	51.16		
			301.11	0.02	-0.1792E+02	0.1261E-01	
					0.125742E+06	0.280176E+01	30
			312.12	0.02	-0.2803E+02	0.2249E-01	
					0.124977E+06	0.499706E+01	30
13	27.815	0.124	10933.	0.103044E+06	54.79		
			312.05	0.02	-0.2572E+02	0.2252E-01	
					0.102739E+06	0.500788E+01	30
			302.98	0.02	-0.1765E+02	0.1409E-01	
					0.103349E+06	0.313343E+01	30
14	27.546	0.097	11761.	0.888427E+05	57.10		
			303.09	0.02	-0.1658E+02	0.1407E-01	
					0.891854E+05	0.312666E+01	30
			314.13	0.02	-0.2606E+02	0.2083E-01	
					0.884999E+05	0.462657E+01	30
15	27.755	0.119	12588.	0.683444E+05	60.43		
			314.01	0.02	-0.2332E+02	0.2083E-01	
					0.680816E+05	0.462657E+01	30
			304.99	0.02	-0.1609E+02	0.1310E-01	
					0.686073E+05	0.290953E+01	30
16	27.831	0.053	13456.	0.545482E+05	62.68		
			305.10	0.02	-0.1476E+02	0.1435E-01	
					0.552179E+05	0.345360E+01	30
			325.17	0.02	-0.3326E+02	0.2233E-01	
					0.538784E+05	0.537358E+01	30
17	27.682	0.130	14324.	0.277493E+05	67.03		
			324.99	0.02	-0.2761E+02	0.2032E-01	
					0.274595E+05	0.452784E+01	30
			315.92	0.02	-0.1929E+02	0.2220E-01	
					0.280391E+05	0.493123E+01	30
18	27.994	0.118	15151.	0.125341E+05	69.51		
			316.03	0.02	-0.1732E+02	0.2220E-01	
					0.129017E+05	0.493123E+01	30
			327.13	0.02	-0.2695E+02	0.2426E-01	
					0.121664E+05	0.539846E+01	30
19	27.898	0.122	15978.	-0.854382E+04	72.93		
			326.97	0.02	-0.2375E+02	0.2426E-01	
					-0.880101E+04	0.539846E+01	30
			317.95	0.01	-0.1662E+02	0.1125E-01	
					-0.828663E+04	0.249952E+01	30
20	27.398	0.084	16805.	-0.219179E+05	75.11		
			318.02	0.02	-0.1559E+02	0.1125E-01	
					-0.216032E+05	0.249952E+01	30
			329.14	0.02	-0.2398E+02	0.1483E-01	
					-0.222327E+05	0.329358E+01	30
21	28.160	0.098	17632.	-0.407054E+05	78.16		
			329.04	0.02	-0.2125E+02	0.1483E-01	
					-0.409376E+05	0.329358E+01	30
			319.94	0.01	-0.1481E+02	0.9052E-02	
					-0.404733E+05	0.201093E+01	30
22	28.837	0.107	19366.	-0.781598E+05	84.25		
			340.00	0.01	-0.2545E+02	0.2151E-01	
					-0.784489E+05	0.477898E+01	30
			330.96	0.01	-0.1786E+02	0.1266E-01	
					-0.778707E+05	0.281145E+01	30
23	27.688	0.083	20194.	-0.923056E+05	86.35		
			331.05	0.01	-0.1621E+02	0.1267E-01	
					-0.919698E+05	0.281753E+01	30
			342.18	0.02	-0.2490E+02	0.1768E-01	
					-0.926414E+05	0.393167E+01	30

Listing of the FORTRAN program TGEDIT. The source is in file EPROG.

```
$ASSM
TGEDIT PROG
$FORT
C
C  PROGRAM TO EDIT THE DATA FILES FROM FACTOR JUMP TG
C     RUNS IN A CONSISTENT MANNER.  DATA ARE SELECTED ONLY WHEN THEY
C     ARE ABOVE USER SPECIFIED LIMITS OF PRECISION OR WITHIN SPECIFIED
C     RANGES OF TEMPERATURE AND DEGREE OF REACTON.
C     CORRECTIONS CAN BE MADE TO THE TEMPERATURES.
C
C  THE PROGRAM READS THE SAVE FILE FROM THE FACTOR JUMP THERMOGRAVIMETRY
C  EXPERIMENT AND WRITES AN EDITED FILE IN THE SAME FORMAT
C
C  THE INPUT/OUTPUT UNITS ARE AS FOLLOWS:
C  ASK QUESTIONS ON LU 4
C  GET RESPONSES FROM LU 5
C  READ INPUT FILE FROM LU 7
C  WRITE OUTPUT FILE ON LU 8
C
      DIMENSION T(2),SIGT(2),ERATE(2),SIGER(2),EFACT(2),SIGF(2)
     &,NTITLE(35),N(2)
C
C  READ IN ANY TEMPERATURE CORRECTION
C
      CORREC=0.
    2 WRITE(4,1)
    1 FORMAT(/' CORRECTIONS TO TEMPERATURES OR NOT? '
     &/'  GIVE Y FOR YES, N FOR NO')
      READ(5,9)NETFIX
    9 FORMAT(A1)
      IF(NETFIX.EQ.'N')GO TO 103
      IF (NETFIX.NE.'Y') GO TO 2
    3 CONTINUE
      WRITE(4,5)
    5 FORMAT(///' GIVE TEMPERATURE INCREMENT')
      READ(5,6)CORREC
    6 FORMAT(F10.0)
      WRITE(4,7)CORREC
    7 FORMAT(' CORRECTION VALUE IS ',F10.2)
      WRITE(4,8)
    8 FORMAT(' IS THIS OK? Y OR N')
      READ(5,9)NGO
      IF(NGO.NE.'Y')GO TO 3
C
C  INPUT LIMITS TO BE USED IN EDITING
C
  103 WRITE(4,15)
   15 FORMAT(/' GIVE:'/
     &/' 1)  MINIMUM DEGREE OF REACTION IN %'
     &/' 2)  MAXIMUM DEGREE OF REACTION IN %'
     &/' 3)  MINIMUM RATE (ABSOLUTE VALUE IS OK)'
     &/' 4)  MAXIMUM % ERROR IN RATE (=100*SIGR/RATE)'
     &/' 5)  MINIMUM TEMPERATURE IN DEGREES CELSIUS'
     &/' 6)  MAXIMUM TEMPERATURE'
     &/' 7)  MAXIMUM ERROR IN TEMPERATURE'
     &/' 8)  MAXIMUM ERROR IN ACTIVATION ENERGY'
```

```
     &/'  $$$ ALL ON SEPARATE LINES $$$')
      READ(5,35)DOCMIN,DOCMAX,RCUT,SRCUT,TL,TH,STCUT,SECUT
   35 FORMAT(F10.0)
      WRITE(4,36)DOCMIN,DOCMAX,RCUT,SRCUT,TL,TH,STCUT,SECUT
   36 FORMAT(/' INPUT VALUES ARE:',4F8.2/18X,4F8.2)
      WRITE(4,8)
      READ(5,9)NGO
      IF(NGO.NE.'Y')GO TO 103
C
C  READ INPUT FILE, EDIT BLOCK BY BLOCK
C
      DO 90 I=1,2
      READ(7,10)NTITLE
      WRITE(8,10)NTITLE
   90 CONTINUE
   10 FORMAT(35A2)
      I=0
  100 READ(7,20,END=110)K,E,S,ET,SW,C,(T(J),SIGT(J),ERATE(J),SIGER(J)
     &,EFACT(J),SIGF(J),N(J),J=1,2)
C
C  APPLY TEMPERATURE CORRECTION
C
      IF (NETFIX.EQ.0)GO TO 105
      T(1)=T(1)+CORREC
      T(2)=T(2)+CORREC
      T1=T(1)+273.18
      T2=T(2)+273.18
      E=1.987*T1*T2/(T1-T2)*ALOG(ERATE(1)/ERATE(2))
      E=ABS(E/1000.)
  105 CONTINUE
C
C  EDIT FILE BASED ON INPUT LIMITS
C
      DO 108 K=1,2
      IF(ABS(ERATE(K)).LE.RCUT)GO TO 100
      IF ((100.*SIGER(K)/ABS(ERATE(K))).GT.SRCUT)GO TO 100
      IF(T(K).LT.TL.OR.T(K).GT.TH) GO TO 100
      IF (SIGT(K).GE.STCUT)GO TO 100
  108 CONTINUE
      IF (S.GE.SECUT) GO TO 100
      IF (C.LT.DOCMIN.OR.C.GT.DOCMAX) GO TO 100
C
C  OUTPUT SURVIVING DATA
C
      I=I+1
      WRITE(8,20)I,E,S,ET,SW,C,(T(J),SIGT(J),
     &ERATE(J),SIGER(J),EFACT(J),SIGF(J),N(J),J=1,2)
   20 FORMAT(I4,2F7.3,F10.0,E15.6,F7.2/
     &20X,F8.2,F6.2,2E15.4/34X,2E15.6,I5
     &/20X,F8.2,F6.2,2E15.4/34X,2E15.6,I5)
      GO TO 100
C
C  EOF ON INPUT FILE
C
  110 END FILE 8
      STOP
      END
```

Memory map of the TGEDIT program.

```
REL PROGS:
  3576 TGEDIT   4024 ALOG     4132 ABS     4148 .COMP
  4174 $6       41BE .RARG    41EA $8      4214 .5
  421C .S       4252 .0       429A .MES    4310 .U
  4340 .V       434E @Z       4374 @H      55BE

ABS PROGS:
NONE

ENTRY-POINTS:
  4026 ALOG     4134 ABS      414A .COMP   4176 $6
  41C0 .RARG    41EC $8       4218 .5      421E .S
  4254 .0       429C .MES     431E .U      4346 .V
  4350 @Z       43C2 @H

COMMON-BLOCKS:
NONE

UNDEFINED:
NONE
```

Appendix C. TGTRIM, a FORTRAN program to calculate trimmed means of the activation energies in the save file from factor-jump thermogravimetry.

Listing of the file SETRIM, used as DO SETRIM,DO to assign files as needed by the TRIMPG program.

```
AS 120
AS 220
AS 320
AS 420
AS 520
AS 662
AS 720
AS 820
AS 920
AS A20
AS B20
AC TRIM01.CD0
AC TRIM02.DD0
AS E20
AS F20
ASSIGN INPUT FILE (UNIT 7) THEN TYPE 'RUN TRIMPG.D0'
TR
```

Example of how to run the TGTRIM program. User responses are underlined.

```
*DO SETRIM
ASSIGN INPUT FILE (UNIT 7) THEN TYPE 'RUN TRIMPG.D0'
?
*AC PEVALL.ZD0
*RUN TRIMPG

 TGTRIM PROGRAM
READ INPUT DATA FROM FILE  7
FILE   6 IS FOR OUTPUT
GIVE NUMBER OF FILES TO EXAMINE (IN FREE FORMAT)
!
1
EXAMINE   1 FILES

GIVE 0 FOR RATE DATA INPUT
     1 FOR E/SIG E INPUT (OUTPUT OF TGRUNF)
        WITH DEGREES OF CONVERSION IN FILE
     2 FOR E/SIG E INPUT WITH D.O.C. TO BE CALCULATED
        NEEDS INITIAL AND FINAL SAMPLE WEIGHTS
        AS THIRD LINE OF FILE
!
1
FILE NUMBER  1

 TITLE OF INPUT FILE IS :


POOLED E'S FROM PE IN VAC 4/78  FROM PEVAC1.2.3.4 & PEBV1.2

 !
 !
 !
 !
```

Typical output from the TGTRIM program.

```
*************************************************************************

 TITLE OF INPUT FILE IS :

POLY ISO BUTYLENE IN N2  RUNS PIBE05 TO 08  DEC 1979
 SAMPLES IN E BAL RANGE .13 TO -.165

*************************************************************************
```

NUMBER	E	SIG E	D.O.C.	SWT
1	49.78	1.63	22.68	51072.50
2	47.47	0.95	28.54	35081.00
3	50.24	1.01	38.29	8459.12
4	48.19	0.54	44.20	-7675.77
5	47.57	0.69	53.67	-33521.80
6	49.63	0.52	59.21	-48630.70
7	50.07	0.58	67.55	-71421.81
8	47.64	0.47	72.19	-84090.87
9	48.29	0.67	78.85	-102272.00
10	48.54	0.34	82.83	-113127.00
11	45.15	0.83	90.13	-133044.00
12	52.51	0.72	12.64	58305.00
13	51.09	0.94	23.26	31141.30
14	49.30	0.55	29.29	15738.60
15	49.59	0.64	39.35	-9979.59
16	49.03	0.46	45.44	-25541.00
17	51.17	0.74	55.16	-50382.30
18	47.02	0.52	60.82	-64866.30
19	49.52	0.78	69.44	-86878.81
20	47.95	0.58	74.18	-98999.81
21	50.42	0.82	80.85	-116054.00
22	48.12	0.43	84.78	-126091.00
23	40.93	1.19	91.87	-144220.00
24	50.81	1.04	16.68	81641.88
25	48.16	0.70	20.53	70476.37
26	48.86	0.87	27.37	50618.90
27	47.83	0.59	31.73	37970.50
28	49.23	0.73	39.13	16515.20
29	48.59	0.53	43.73	3179.93
30	48.38	0.86	51.40	-19053.40
31	48.09	0.59	56.03	-32480.70
32	46.59	0.77	63.36	-53740.90
33	49.22	0.57	67.62	-66101.50
34	48.55	0.85	74.14	-84995.50
35	47.79	0.58	77.81	-95640.13
36	46.05	0.83	83.15	-111136.00
37	49.32	0.41	86.41	-120595.00
38	47.76	0.90	92.64	-138646.00
39	48.76	1.10	12.92	77352.62
40	50.43	0.70	17.17	65609.00
41	48.32	0.85	24.64	44983.20
42	48.70	0.59	29.41	31823.00
43	47.55	0.71	37.46	9606.20
44	50.28	0.56	42.40	-4034.69
45	47.42	0.87	50.59	-26625.90
46	48.14	0.63	55.50	-40167.80
47	50.24	0.87	63.13	-61235.30
48	46.47	0.54	67.51	-73319.38
49	47.49	0.95	74.08	-91448.37
50	49.52	0.47	78.26	-102998.00
51	45.30	0.76	86.71	-126309.00
52	48.53	0.37	90.56	-136939.00

52 DATA READ IN

AVERAGE VALUES BEFORE DATA TRIMMED

WEIGHTED MEAN E IS 48.566 + OR - 0.087
REDUCED CHISQ FOR E CALCULATION IS 4.855 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS 0.719
CALCULATED OVER 50. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 48.421 TO 48.712
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.493 AND 0.253
THIS GIVES A CONFIDENCE RANGE OF 48.069 TO 48.917
CHI-SQUARE FOR UNWEIGHTED E IS 4.87
T VALUE APPLIED IS 1.676

TRIMMED MEANS FOLLOW:
5 DATA CHOPPED OFF EACH END AND 42 DATA USED

E	SIG E	DOC
46.59	0.77	63.36
47.02	0.52	60.82
47.42	0.87	50.59
47.47	0.95	28.54
47.49	0.95	74.08
47.55	0.71	37.46
47.57	0.69	53.67
47.64	0.47	72.19
47.76	0.90	92.64
47.79	0.58	77.81
47.83	0.59	31.73
47.95	0.58	74.18
48.09	0.59	56.03
48.12	0.43	84.78
48.14	0.63	55.50
48.16	0.70	20.53
48.19	0.54	44.20
48.29	0.67	78.85
48.32	0.85	24.64
48.38	0.86	51.40
48.53	0.37	90.56
48.54	0.34	82.83
48.55	0.85	74.14
48.59	0.53	43.73
48.70	0.59	29.41
48.76	1.10	12.92
48.86	0.87	27.37
49.03	0.46	45.44
49.22	0.57	67.62
49.23	0.73	39.13
49.30	0.55	29.29
49.32	0.41	86.41
49.52	0.78	69.44
49.52	0.47	78.26
49.59	0.64	39.35
49.63	0.52	59.21
49.78	1.63	22.68
50.07	0.58	67.55
50.24	1.01	38.29
50.24	0.87	63.13
50.28	0.56	42.40
50.42	0.82	80.85

```
WEIGHTED MEAN E IS  48.614 + OR -   0.093
REDUCED CHISQ FOR E CALCULATION IS  2.071 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS   0.694
CALCULATED OVER   40. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 48.458 TO 48.771
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.612 AND  0.150
THIS GIVES A CONFIDENCE RANGE OF 48.359 TO 48.866
CHI-SQUARE FOR UNWEIGHTED E IS   2.07
T VALUE APPLIED IS  1.684

WINSORISED MEAN AND SIGMA FOLLOW:
(I.E., REPLACE REJECTED VALUES WITH NEAREST UNREJECTED VALUE)

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.592 AND  0.208
THIS GIVES A CONFIDENCE RANGE OF 48.243 TO 48.942
CHI-SQUARE FOR UNWEIGHTED E IS   2.81
T VALUE APPLIED IS  1.676
```

TRIMMED MEANS FOLLOW:
10 DATA CHOPPED OFF EACH END AND 32 DATA USED

E	SIG E	DOC
47.55	0.71	37.46
47.57	0.69	53.67
47.64	0.47	72.19
47.76	0.90	92.64
47.79	0.58	77.81
47.83	0.59	31.73
47.95	0.58	74.18
48.09	0.59	56.03
48.12	0.43	84.78
48.14	0.63	55.50
48.16	0.70	20.53
48.19	0.54	44.20
48.29	0.67	78.85
48.32	0.85	24.64
48.38	0.86	51.40
48.53	0.37	90.56
48.54	0.34	82.83
48.55	0.85	74.14
48.59	0.53	43.73
48.70	0.59	29.41
48.76	1.10	12.92
48.86	0.87	27.37
49.03	0.45	45.44
49.22	0.57	67.62
49.23	0.73	39.13
49.30	0.55	29.29
49.32	0.41	85.41
49.52	0.78	69.44
49.52	0.47	78.26
49.59	0.64	39.35
49.63	0.52	59.21
49.78	1.63	22.68

WEIGHTED MEAN E IS 48.586 + OR - 0.101
REDUCED CHISQ FOR E CALCULATION IS 1.223 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS 0.663
CALCULATED OVER 30. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 48.413 TO 48.758
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.577 AND 0.120
THIS GIVES A CONFIDENCE RANGE OF 48.373 TO 48.782
CHI-SQUARE FOR UNWEIGHTED E IS 1.22
T VALUE APPLIED IS 1.697

WINSORISED MEAN AND SIGMA FOLLOW:
(I.E., REPLACE REJECTED VALUES WITH NEAREST UNREJECTED VALUE)

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.611 AND 0.198
THIS GIVES A CONFIDENCE RANGE OF 48.279 TO 48.943
CHI-SQUARE FOR UNWEIGHTED E IS 1.28
T VALUE APPLIED IS 1.676

TRIMMED MEANS FOLLOW:
15 DATA CHOPPED OFF EACH END AND 22 DATA USED

E	SIG E	DOC
47.83	0.59	31.73
47.95	0.58	74.18
48.09	0.59	56.03
48.12	0.43	84.78
48.14	0.63	55.50
48.16	0.70	20.53
48.19	0.54	44.20
48.29	0.67	78.85
48.32	0.85	24.64
48.38	0.86	51.40
48.53	0.37	90.56
48.54	0.34	82.83
48.55	0.85	74.14
48.59	0.53	43.73
48.70	0.59	29.41
48.76	1.10	12.92
48.86	0.87	27.37
49.03	0.46	45.44
49.22	0.57	67.62
49.23	0.73	39.13
49.30	0.55	29.29
49.32	0.41	86.41

WEIGHTED MEAN E IS 48.577 + OR - 0.118
REDUCED CHISQ FOR E CALCULATION IS 0.689 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS 0.628
CALCULATED OVER 20. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 48.373 TO 48.781
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.551 AND 0.099
THIS GIVES A CONFIDENCE RANGE OF 48.380 TO 48.723
CHI-SQUARE FOR UNWEIGHTED E IS 0.69
T VALUE APPLIED IS 1.725

WINSORISED MEAN AND SIGMA FOLLOW:
(I.E., REPLACE REJECTED VALUES WITH NEAREST UNREJECTED VALUE)

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 48.565 AND 0.214
THIS GIVES A CONFIDENCE RANGE OF 48.205 TO 48.924
CHI-SQUARE FOR UNWEIGHTED E IS 1.74
T VALUE APPLIED IS 1.676

Listings of the routines in the FORTRAN program TGTRIM.

```
$ASSM
TGTRIM PROG
$FORT
$TRGT 16
C   ***************************************************************
C
C   TGTRIM, SET UP FOR MULTIPLE RUNS FROM ONE START.
C   PROGRAM TO EVALUATE TG DATA DERIVED FROM TEMPERATURE JUMPS
C
C   ******************************************************************
C
C   WRITTEN BY B. DICKENS, 311.02 NBS
C
C   TELEPHONE (301) 921-3322
C
C   PROGRAM NEEDS THE FOLLOWING SUBROUTINES AND FUNCTIONS:
C     OVERLY
C     PRIMTR
C     TRIM
C     TRIMRD
C
C
C   COMMON BLOCKS /IO/ AND /REGRS/ DEFINED VIA ARRAY 'DUMP'
C      COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C      COMMON /REGRS/ M(5),R(5),YFIT(250)
C
      COMMON  DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
      EQUIVALENCE (NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
C
      DIMENSION NTITLE(64),IDOLD(75),ID(75),E(75),SIGE(75),DOC(75)
      DIMENSION SWT(75),NTIME(75),TREAD(2,75),RATE(2,75),TEMP(2,75)
      DIMENSION SIGT(2,75),SIGR(2,75)
      EQUIVALENCE (DOC(1),NTIME(1))
C
C   OVERLAY NAMES AND FILE ASSIGNMENTS
C
      INTEGER*2 IPRIMS(3),ITRIMS(3)
      DATA IPRIMS/'OP','RI','MS'/, ITRIMS/'OT','RI','MS'/
      NPRIMS = 12
      NTRIMS = 13
      NDATA = 75
      NTTYO = 4
C
C   INITIALISE AND READ IN DATA
C
      CALL OVERLY (IPRIMS,NPRIMS,0,NTTYO)
      CALL PRIMTR (NLOOPY,NLOOP,INDATA)
  100 IF (NLOOPY.GT.0) CALL OVERLY (IPRIMS,NPRIMS,0,NTTYO)
      CALL TRIMRD (NTITLE,NLOOPY,IDOLD,E,SIGE,SWT,DOC,NTIME,TREAD
     & ,SIGT,RATE,SIGR,ID,INDATA,NDATA,NERROR)
      IF (NERROR.GT.0) GO TO 600
C
C   CALCULATE TRIMMED MEANS
C
      CALL OVERLY (ITRIMS,NTRIMS,0,NTTYO)
      CALL TRIM (E,SIGE,DOC,NDATA)
      WRITE (NPRINT,10)
   10 FORMAT ('1')
      IF (NLOOPY.LT.NLOOP) GO TO 100
  600 STOP
      END
```

```
$ASSM
ACTIV PROG
$FORT
$TRGT 16
C *****************************************************************
C
      SUBROUTINE ACTIV(E,SIGE,TEMP1,TEMP2,SIGT1,SIGT2
     &,TREAD1,TREAD2,SIGTR1,SIGTR2
     &,RATE1,RATE2,SIGR1,SIGR2,AT,NERROR)
C
C *****************************************************************
C  SUBROUTINE TO CALCULATE ACTIVATION ENERGY FROM TEMPERATURE AND
C  RATE DATA
C
C   *** FOR A SINGLE E VALUE ***
C
C  E  =  ACTIVATION ENERGY
C  SIGE  =  STANDARD DEVIATION OF E
C  IE  =  ORDINAL NUMBER OF E IN LIST OF E VALUES
C  TEMP1  =  FIRST TEMPERATURE (IN DEGREES K)
C  TEMP2  =  SECOND TEMPERATURE
C  SIGT1, SIGT2,  =  STANDARD DEVIATIONS OF TEMP1 AND TEMP2
C  TREAD  =  VALUES OF CORRESPONDING THERMOCOUPLE EMF'S IN MICROVOLTS
C  VTEMP  =  CONTRIBUTION OF TEMP TO VARIANCE OF E
C
C  RATE1, RATE2, SIGR1, SIGR2 AND VRATE ARE SIMILAR VALUES FOR RATES
C  AT  =  POLYNOMIAL COEFFICIENTS FOR EMF TO DEGREE K CONVERSION
C  NERROR  =  ERROR INDICATOR
C
C
C  NEEDS THE FOLLOWING COMMON BLOCK AND FUNCTION:
C    COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C    TDEGAB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION TEMP(2),SIGT(2),TREAD(2),SIGTR(2),RATE(2),SIGR(2),AT(4)
C
C
      R = 1.987/1000.
      R2 = R**2
      NERROR = 0
      TREAD(1) = TREAD1
      TREAD(2) = TREAD2
      SIGTR(1) = SIGTR1
      SIGTR(2) = SIGTR2
      RATE(1) = RATE1
      RATE(2) = RATE2
      SIGR(1) = SIGR1
      SIGR(2) = SIGR2
      DO 30 J = 1,2
C
C  GET TEMPERATURE IN DEGREES KELVIN. TREAD IS IN MICROVOLTS
C
      TEMP(J) = TDEGAB (TREAD(J),AT)
      T = TREAD(J)+SIGTR(J)
      S = TDEGAB(T,AT)
      SIGT(J) = S-TEMP(J)
      IF (TEMP(J).LT.1.) GO TO 40
      IF (ABS(RATE(J)).LT.1.E-06) GO TO 40
30    CONTINUE
C
C  GET ACTIVATION ENERGY
```

```
C
      TDIFF = TEMP(2)-TEMP(1)
      TDIFF = ABS(TDIFF)
      IF (TDIFF.LT.1.E-01) GO TO 40
      T = RATE(2)/RATE(1)
      IF (T.LT.1.E-06) GO TO 40
      IF (T.GT.1.E+06) GO TO 40
      IF (T.LT.1.) T = 1./T
      RTEMP = ALOG(T)
      ETEMP = TEMP(2)*TEMP(1)/TDIFF
      E = R*RTEMP*ETEMP
      IF (ABS(E).GT.200..OR.ABS(E).LT.1.) GO TO 40
      GO TO 90
C
C  DISCARD THIS POINT FOR VARIOUS POSSIBLE REASONS
C.
40    CONTINUE
      IF (NPRINT.GE.3) WRITE (NPRINT,50)IE
      IF (NFILEB.GT.6) WRITE (NFILEB,50)IE
   50 FORMAT (' ACT. ENERGY NUMBER',I3,' DISCARDED')
      NERROR = 8
      RETURN
90    CONTINUE
      E2 = E**2
C
C  GET ASSOCIATED SIGMA
C  1) FROM RATES
C
      VRATE = ((SIGR(2)/RATE(2))**2+(SIGR(1)/RATE(1))**2)*E
     &2/RTEMP**2
C
C  2) FROM TEMPERATURES
C
      VTEMP = ((SIGT(1)/TEMP(1)**2)**2+(SIGT(2)/TEMP(2)**2)
     &**2)*E2*ETEMP**2
C
C  COMPUTE TOTAL SIGMA ON E
C
      SIGE = SQRT(VRATE+VTEMP)
C
C  OUTPUT INITIAL AND CALCULATED VALUES
C
      TEMP(1) = TEMP(1)-273.15
      TEMP(2) = TEMP(2)-273.15
      IF (NPRINT.LT.3)GO TO 200
      WRITE (NPRINT,52)
   52 FORMAT (/' ACTIVATION ENERGY FROM  = ACTIVE = '
     &/6X,'E',2X,'SIG E',2X,'FACTOR OR RATE '/)
      WRITE (NPRINT,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J = 1,2)
100       FORMAT (2F7.3,F8.2,F6.2,2E13.4/14X,F8.2,F6.2,2E13.4)
110       CONTINUE
C
C
120.  WRITE (NPRINT,130)
130   FORMAT (/' CONTRIBUTION OF RATE AND TEMPERATURE TO VARIANCE'
     &,'( = SIGE**2)'/'     RATE     TEMP'/)
      WRITE (NPRINT,140) VRATE,VTEMP
140   FORMAT (2E12.3)
  200 IF (NFILEB.LE.6)RETURN
      WRITE (NFILEB,52)
      WRITE (NFILEB,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J = 1,2)
      WRITE (NFILEB,130)
      WRITE (NFILEB,140)VRATE,VTEMP
      TEMP1 = TEMP(1)
      TEMP2 = TEMP(2)
      SIGT1 = SIGT(1)
      SIGT2 = SIGT(2)
      RETURN
      END
```

```
$ASSM
AVSDN PROG
$FORT
$TRGT 16
C ***********************************************************************
C
      SUBROUTINE AVSDN(AVE,SD,Q,NVAL)
C
C ***********************************************************************
C  SUBROUTINE TO CALCULATE AVERAGE AND STANDARD DEVIATION
C  AVE = AVERAGE VALUE
C  Q = INPUT VALUES
C  SD = STANDARD DEVIATION OF POPULATION OF Q
C  NVAL = NUMBER OF VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
C  RETURNS SD = 1.E+06 IF ONLY ONE VALUE IS AVAILABLE
C
C
      DIMENSION Q(NVAL)
C
C  CALCULATE AVERAGE
C
      AVE=0.
      DO 10 I=1,NVAL
   10 AVE=AVE+Q(I)
      AVE=AVE/FLOAT(NVAL)
      IF(NVAL.GT.1) GO TO 50
      SD=1.0E+06
      RETURN
   50 SD=0.
C
C  CALCULATE STANDARD DEVIATION
C
      DO 40 I=1,NVAL
 40   SD=SD+(AVE-Q(I))**2
      SD=SQRT(SD/(FLOAT(NVAL-1)))
      RETURN
      END
```

```
$ASSM
CHISQ PROG
$FORT
$TRGT 16
C ************************************************************************
C
      SUBROUTINE CHISQ(P,SP,XSQ,N)
C
C ************************************************************************
C  SUBROUTINE TO CALCULATE REDUCED CHI SQUARE FOR N INPUT VALUES.
C  P = INPUT VALUES
C  SP = STANDARD DEVIATIONS OF P VALUES
C  XSQ = CHI-SQUARE VALUE FOR P AND SP ARRAYS
C  N = NUMBER OF P VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
      DIMENSION P(N),SP(N)
      AVE=0.
C
C  CALCULATE AVERAGE
C
      DO 10 I=1,N
 10   AVE=AVE+P(I)
      AVE=AVE/FLOAT(N)
      XSQ=0.
      IF(N.LE.1)RETURN
C
C  CALCULATE CHI SQUARE
C
      DO 20 I=1,N
 20   XSQ=XSQ+((AVE-P(I))/SP(I))**2
      XSQ=XSQ/FLOAT(N-1)
      RETURN
      END
```

```
$ASSM
FCTN PROG
$FORT
$TRGT 16
C
C     ********************************************************************************
C
      FUNCTION FCTN (X,I,J,JTERMS,NPTS)
C
C     ********************************************************************************
C
C  FUNCTION TO CALCULATE FITTED VALUES FOR REGRES SUBROUTINE
C
C
      DIMENSION X(NPTS),JTERMS(J)
      JEXP=JTERMS(J)
      FCTN=X(I)**JEXP
      RETURN
      END
```

```
$ASSM
EREAD PROG
$FORT
$TRGT 16
C  ******************************************************************
C
      SUBROUTINE EREAD(IDOLD,E,SIGE,SWT,DOC,NDATA,INDATA,NERROR)
C
C  ******************************************************************
C
C       SUBROUTINE TO READ IN E/SIGE FILE FROM TGRUNF
C     NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C     AND FUNCTION READX
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION IDOLD(1),E(1),SIGE(1),SWT(1),DOC(1)
C
C  NDATA IS INITIALISED NEAR DIMENSION STATEMENT IN MAIN PROGRAM
C
      NDO = NDATA
      NDATA = 0
C
C  READ IN E/SIG E FILE (5 RECORDS PER ACTIVATION ENERGY)
C
      DO 10 I = 1,NDO
      J = READX (IW,NERROR,3,IN,NTTYO,IEOF)
C
C  TEST FOR EOF
C
      IF (IEOF.GT.0)GO TO 20
      NDATA = NDATA+1
      IDOLD(NDATA) = I
      E(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SIGE(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      A = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SWT(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      IF (INDATA.EQ.1)DOC(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
C
C  SPACE OVER REST OF RECORDS FOR THIS ACTIVATION ENERGY
C
      DO 15 K = 1,4
   15 A = READX (IW,NERROR,3,IN,NTTYO,IEOF)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
$ASSM
MATINV PROG
$FORT
$TRGT 16
C      ************************************************************************
C
      SUBROUTINE MATINV (ARRAY,NORDER,DET)
C
C      ************************************************************************
C  SUBROUTINE FOR MATRIX INVERSION
C
      DOUBLE PRECISION ARRAY,AMAX,SAVE
      DIMENSION ARRAY(5,5),IK(5),JK(5)
      DET=1.
      DO 190 K=1,NORDER
C
C  FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX
C
          AMAX=0.
10        DO 30 I=K,NORDER
              DO 30 J=K,NORDER
              IF (DABS(AMAX)-DABS(ARRAY(I,J))) 20,20,30
20            AMAX=ARRAY(I,J)
              IK(K)=I
              JK(K)=J
30            CONTINUE
C
C  INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
C
          IF (AMAX) 50,40,50
40        DET=0.
          GO TO 260
50        I=IK(K)
          IF (I-K) 10,80,60
60        DO 70 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=ARRAY(I,J)
70            ARRAY(I,J)=-SAVE
80        J=JK(K)
          IF (J-K) 10,110,90
90        DO 100 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=ARRAY(I,J)
100           ARRAY(I,J)=-SAVE
C
C  ACCUMULATE ELEMENTS OF INVERSE MATRIX
C
110       DO 130 I=1,NORDER
              IF (I-K) 120,130,120
120           ARRAY(I,K)=-ARRAY(I,K)/AMAX
130           CONTINUE
          DO 160 I=1,NORDER
              DO 160 J=1,NORDER
              IF (I-K) 140,160,140
140           IF (J-K) 150,160,150
150           ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
```

```
160             CONTINUE
          DO 180 J=1,NORDER
             IF (J-K) 170,180,170
170          ARRAY(K,J)=ARRAY(K,J)/AMAX
180          CONTINUE
          ARRAY(K,K)=1./AMAX
190       DET=DET*AMAX
C
C  RESTORE ORDERING OF MATRIX
C
       DO 250 L=1,NORDER
          K=NORDER-L+1
          J=IK(K)
          IF (J-K) 220,220,200
200       DO 210 I=1,NORDER
             SAVE=ARRAY(I,K)
             ARRAY(I,K)=-ARRAY(I,J)
210          ARRAY(I,J)=SAVE
220       I=JK(K)
          IF (I-K) 250,250,230
230       DO 240 J=1,NORDER
             SAVE=ARRAY(K,J)
             ARRAY(K,J)=-ARRAY(I,J)
240          ARRAY(I,J)=SAVE
250       CONTINUE
260    RETURN
       END
```

```
$ASSM
NORPPF  PROG  D124:NORPPF.FTN
$FORT
      SUBROUTINE NORPPF(P,PPF)
C
C     PURPOSE--THIS SUBROUTINE COMPUTES THE PERCENT POINT
C              FUNCTION VALUE FOR THE NORMAL (GAUSSIAN)
C              DISTRIBUTION WITH MEAN = 0 AND STANDARD DEVIATION = 1.
C              THIS DISTRIBUTION IS DEFINED FOR ALL X AND HAS
C              THE PROBABILITY DENSITY FUNCTION
C              F(X) = (1/SQRT(2*PI))*EXP(-X*X/2).
C              NOTE THAT THE PERCENT POINT FUNCTION OF A DISTRIBUTION
C              IS IDENTICALLY THE SAME AS THE INVERSE CUMULATIVE
C              DISTRIBUTION FUNCTION OF THE DISTRIBUTION.
C     INPUT  ARGUMENTS--P      = THE SINGLE PRECISION VALUE
C                                (BETWEEN 0.0 AND 1.0)
C                                AT WHICH THE PERCENT POINT
C                                FUNCTION IS TO BE EVALUATED.
C     OUTPUT ARGUMENTS--PPF    = THE SINGLE PRECISION PERCENT
C                                POINT FUNCTION VALUE.
C     OUTPUT--THE SINGLE PRECISION PERCENT POINT
C             FUNCTION VALUE PPF.
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.
C     RESTRICTIONS--P SHOULD BE BETWEEN 0.0 AND 1.0, EXCLUSIVELY.
C     OTHER DATAPAC   SUBROUTINES NEEDED--NONE.
C     FORTRAN LIBRARY SUBROUTINES NEEDED--SQRT, ALOG.
C     MODE OF INTERNAL OPERATIONS--SINGLE PRECISION.
C     LANGUAGE--ANSI FORTRAN.
C     REFERENCES--ODEH AND EVANS, THE PERCENTAGE POINTS
C                 OF THE NORMAL DISTRIBUTION, ALGORTIHM 70,
C                 APPLIED STATISTICS, 1974, PAGES 96-97.
C               --EVANS, ALGORITHMS FOR MINIMAL DEGREE
C                 POLYNOMIAL AND RATIONAL APPROXIMATION,
C                 M. SC. THESIS, 1972, UNIVERSITY
C                 OF VICTORIA, B. C., CANADA.
C               --HASTINGS, APPROXIMATIONS FOR DIGITAL
C                 COMPUTERS, 1955, PAGES 113, 191, 192.
C               --NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS
C                 SERIES 55, 1964, PAGE 933, FORMULA 26.2.23.
C               --FILLIBEN, SIMPLE AND ROBUST LINEAR ESTIMATION
C                 OF THE LOCATION PARAMETER OF A SYMMETRIC
C                 DISTRIBUTION (UNPUBLISHED PH.D. DISSERTATION,
C                 PRINCETON UNIVERSITY), 1969, PAGES 21-44, 229-231.
C               --FILLIBEN, 'THE PERCENT POINT FUNCTION',
C                 (UNPUBLISHED MANUSCRIPT), 1970, PAGES 28-31.
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE
C                 DISTRIBUTIONS--1, 1970, PAGES 40-111.
C               --THE KELLEY STATISTICAL TABLES, 1948.
C               --OWEN, HANDBOOK OF STATISTICAL TABLES,
C                 1962, PAGES 3-16.
C               --PEARSON AND HARTLEY, BIOMETRIKA TABLES
C                 FOR STATISTICIANS, VOLUME 1, 1954,
C                 PAGES 104-113.
C     COMMENTS--THE CODING AS PRESENTED BELOW
C               IS ESSENTIALLY IDENTICAL TO THAT
C               PRESENTED BY ODEH AND EVANS
```

```
C                AS ALGORTIHM 70 OF APPLIED STATISTICS.
C                THE PRESENT AUTHOR HAS MODIFIED THE
C                ORIGINAL ODEH AND EVANS CODE WITH ONLY
C                MINOR STYLISTIC CHANGES.
C             --AS POINTED OUT BY ODEH AND EVANS
C                IN APPLIED STATISTICS,
C                THEIR ALGORITHM REPRESENTES A
C                SUBSTANTIAL IMPROVEMENT OVER THE
C                PREVIOUSLY EMPLOYED
C                HASTINGS APPROXIMATION FOR THE
C                NORMAL PERCENT POINT FUNCTION--
C                THE ACCURACY OF APPROXIMATION
C                BEING IMPROVED FROM 4.5*(10**-4)
C                TO 1.5*(10**-8).
C     WRITTEN BY--JAMES J. FILLIBEN
C                 STATISTICAL ENGINEERING LABORATORY (205.03)
C                 NATIONAL BUREAU OF STANDARDS
C                 WASHINGTON, D. C. 20234
C                 PHONE:  301-921-2315
C     ORIGINAL VERSION--JUNE      1972.
C     UPDATED         --SEPTEMBER 1975.
C     UPDATED         --NOVEMBER  1975.
C     UPDATED         --OCTOBER   1976.
C
C-----------------------------------------------------------------------
C
      DATA P0,P1,P2,P3,P4
     1/-.322232431088,-1.0,
     1 -.342242088547,-.204231210245E-1,
     1 -.453642210148E-4/
      DATA Q0,Q1,Q2,Q3,Q4
     1/.993484626060E-1,.588581570495,
     1 .531103462366,.103537752850,
     1 .38560700634E-2/
C
      IPR=6
C
C     CHECK THE INPUT ARGUMENTS FOR ERRORS
C
      IF(P.LE.0.0.OR.P.GE.1.0)GOTO50
      GOTO90
   50 WRITE(IPR,1)
      WRITE(IPR,46)P
      RETURN
   90 CONTINUE
    1 FORMAT(1H ,115H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THE
     1 NORPPF SUBROUTINE IS OUTSIDE THE ALLOWABLE (0,1) INTERVAL *****)
   46 FORMAT(1H , 35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)
C
C-----START POINT-------------------------------------------------------
C
      IF(P.NE.0.5)GOTO150
      PPF=0.0
      RETURN
C
  150 R=P
      IF(P.GT.0.5)R=1.0-R
      T=SQRT(-2.0*ALOG(R))
      ANUM=((((T*P4+P3)*T+P2)*T+P1)*T+P0)
      ADEN=((((T*Q4+Q3)*T+Q2)*T+Q1)*T+Q0)
      PPF=T+(ANUM/ADEN)
      IF(P.LT.0.5)PPF=-PPF
      RETURN
C
      END
```

```
$ASSM
OVERLY PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE OVERLY(NPROG,NUNIT,NREWIN,NTTYO)
C
C ************************************************************
C  SUBROUTINE TO READ IN OVERLAY SEGMENT FROM UNIT NUNIT
C    TRY 3 TIMES IF OVERLAY IS AT BEGINNING OF FILE
C
      INTEGER*2 NPROG(3)
   30 NTIME=1
C
C  REWIND INPUT UNIT IF APPROPRIATE, READ IN OVERLAY SEGMENT
C
   10 IF(NREWIN.EQ.0)REWIND NUNIT
      CALL IFETCH(NPROG,NUNIT,ISTAT)
      IF(ISTAT.EQ.0)RETURN
      NTIME=NTIME+1
      IF(NTIME.LE.3.AND.NREWIN.EQ.0)GO TO 10
C
C  ERROR ROUTE
C
      WRITE(NTTYO,20)NPROG,NUNIT
   20 FORMAT(' CANNOT READ IN OVERLAY ',3A2,' FROM LOGICAL UNIT',I3)
C
C  REPEAT AFTER =CONTINUE= GIVEN BY OPERATOR
C
      PAUSE
      GO TO 30
      END
```

```
$ASSM
PRIMTR PROG
$FORT
$TRGT 16
C   ***************************************************************
C
      SUBROUTINE PRIMTR(NLOOPY,NLOOP,INDATA)
C
C   *******************************************************************
C
C   SUBROUTINE TO INITIALISE TG DATA PROCESSING PROGRAMS
C
C   NEEDS SUBROUTINE READX
C
C     AND COMMON BLOCKS /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP(275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      NLOOPY=0
C
C    INITIALISE FILES, ETC.
C
      NTTYO = 4
      NTTYI = 5
      IN = 7
      NPRINT = 6
      NFILEB = 9
      DO 5 I = 1,5
    5 M(I) = I
C
C
      WRITE (NTTYO,10) IN
   10 FORMAT (/'  TGTRIM PROGRAM'
     & /' READ INPUT DATA FROM FILE',I3)
      WRITE (NTTYO,20) NPRINT
20    FORMAT (5H FILE,I4,14H IS FOR OUTPUT)
      LOOPY = 0
      WRITE (NTTYO,40)
40    FORMAT (' GIVE NUMBER OF FILES TO EXAMINE (IN FREE FORMAT)')
      NLOOP = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      WRITE (NTTYO,60) NLOOP
   60 FORMAT (' EXAMINE',I4,' FILES')
      WRITE (NTTYO,80)
   80 FORMAT (/' GIVE 0 FOR RATE DATA INPUT'
     &/'        1 FOR E/SIG E INPUT (OUTPUT OF TGRUNF)'
     &/'            WITH DEGREES OF CONVERSION IN FILE'
     &/'        2 FOR E/SIG E INPUT WITH D.O.C. TO BE CALCULATED'
     &/'            NEEDS INITIAL AND FINAL SAMPLE WEIGHTS'
     &/'            AS THIRD LINE OF FILE')
      INDATA = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
100   CONTINUE
      RETURN
      END
```

```
$ASSM
READEM PROG
$FORT
$TRGT 16
C   ****************************************************************
C
      SUBROUTINE READEM(NTIME,TREAD,SIGT,RATE,SIGR,ID,NDATA)
C
C   ****************************************************************
C
C   SUBROUTINE TO READ RAW DATA
C   NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION NTIME(1),TREAD(1,1),SIGT(1,1),RATE(1,1),
     & SIGR(1,1),ID(1)
C
C
      IF (NFILEB.GT.6) WRITE (NFILEB,10)
      IF (NPRINT.GE.3) WRITE (NPRINT,10)
10    FORMAT (46H       TIME        THCPL   SIG THCPL          RATE,
     & 12H    SIG RATE/)
      NERROR = 0
      NDO=NDATA
      NDATA = 0
      DO 50 I = 1,NDO
      NTIME(I) = READX(IW,NERROR,3,IN,NTTYO,IEOF)
      IF (IEOF.GT.0) GO TO 60
      TREAD(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGT(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      RATE(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGR(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      TREAD(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGT(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      RATE(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGR(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      IF (NFILEB.GT.6) WRITE (NFILEB,30)NTIME(I),(TREAD(J,I),SIGT(J,I)
     &,RATE(J,I),SIGR(J,I),J = 1,2)
      IF (NPRINT.GE.3) WRITE (NPRINT,30) NTIME(I),(TREAD(J,I),SIGT(J,I)
     &,RATE(J,I),SIGR(J,I),J = 1,2)
30    FORMAT (I10,4E12.4/10X4E12.4)
C
C   PUT THERMOCOUPLE READING IN MILLIVOLTS
C
      DO 40 J = 1,2
      TREAD(J,I) = TREAD(J,I)*1000.
40    SIGT(J,I) = SIGT(J,I)*1000.
      ID(I) = I
      NDATA = NDATA+1
50    CONTINUE
   60 RETURN
      END
```

```
$ASSM
REGRES PROG
$FORT
$TRGT 16
C
C     ****************************************************************************
C
      SUBROUTINE REGRES (X,Y,SIGMAY,NPTS,NTERMS,M,MODE,YFIT,A0,A,SIGMA0,
     2 SIGMAA,R,RMUL,CHISQR,FTEST)
C
C     ****************************************************************************
C
C  NEEDS THE FOLLOWING FUNCTION AND SUBROUTINE:
C    FCTN
C    MATINV
C
C
      DOUBLE PRECISION ARRAY,SUM,YMEAN,SIGMA,CHISQ,XMEAN,SIGMAX
      DIMENSION X(NPTS),Y(NPTS),SIGMAY(NPTS),M(5),YFIT(NPTS)
     &,SIGMAA(5),R(5),A(5)
      DIMENSION WEIGHT(250),XMEAN(5),SIGMAX(5),ARRAY(5,5)
C
C  INITIALIZE SUMS AND ARRAYS
C
      SUM=0.
      YMEAN=0.
      SIGMA=0.
      CHISQ=0.
      RMUL=0.
      DO 10 I=1,NPTS
10        YFIT(I)=0.
      IF (NTERMS.LE.0) GO TO 30
      DO 20 J=1,NTERMS
          XMEAN(J)=0.
          SIGMAX(J)=0.
          R(J)=0.
          A(J)=0.
          SIGMAA(J)=0.
          DO 20 K=1,NTERMS
20        ARRAY(J,K)=0.
30    CONTINUE
C
C  ACCUMULATE WEIGHTED SUMS
C
      DO 110 I=1,NPTS
          IF (MODE) 40,70,80
40        IF (Y(I)) 60,70,50
50        WEIGHT(I)=1./Y(I)
          GO TO 90
60        WEIGHT(I)=1./(-Y(I))
          GO TO 90
70        WEIGHT(I)=1.
          GO TO 90
80        WEIGHT(I)=1./SIGMAY(I)**2
90        SUM=SUM+WEIGHT(I)
          YMEAN=YMEAN+WEIGHT(I)*Y(I)
```

```
          IF (NTERMS.LE.0) GO TO 110
          DO 100 J=1,NTERMS
100           XMEAN(J)=XMEAN(J)+WEIGHT(I)*FCTN(X,I,J,M,NPTS)
110       CONTINUE
      YMEAN=YMEAN/SUM
      IF (NTERMS.LE.0) GO TO 130
      DO 120 J=1,NTERMS
120       XMEAN(J)=XMEAN(J)/SUM
130   CONTINUE
      FNPTS=NPTS
      WMEAN=SUM/FNPTS
      DO 140 I=1,NPTS
140       WEIGHT(I)=WEIGHT(I)/WMEAN
C
C  ACCUMULATE MATRICES R AND ARRAY
C
      DO 150 I=1,NPTS
      SIGMA=SIGMA+WEIGHT(I)*(Y(I)-YMEAN)**2
      IF (NTERMS.LE.0) GO TO 160
      DO 150 J=1,NTERMS
      SIGMAX(J)=SIGMAX(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))**2
      R(J)=R(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))*(Y(I)-YMEAN)
          DO 150 K=1,J
      ARRAY(J,K)=ARRAY(J,K)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))
     2 *(FCTN(X,I,K,M,NPTS)-XMEAN(K))
  150 CONTINUE
160   CONTINUE
      FREE1=NPTS-1
      SIGMA=DSQRT(SIGMA/FREE1)
      IF (NTERMS.LE.0) GO TO 180
      DO 170 J=1,NTERMS
          SIGMAX(J)=DSQRT(SIGMAX(J)/FREE1)
          R(J)=R(J)/(FREE1*SIGMAX(J)*SIGMA)
          DO 170 K=1,J
          ARRAY(J,K)=ARRAY(J,K)/(FREE1*SIGMAX(J)*SIGMAX(K))
170       ARRAY(K,J)=ARRAY(J,K)
180   CONTINUE
C
C  INVERT SYMMETRIC MATRIX
C
      IF (NTERMS.LE.0) GO TO 200
      CALL MATINV (ARRAY,NTERMS,DET)
      IF (DET) 200,190,200
190   A0=0.
      SIGMA0=0.
      RMUL=0.
      CHISQR=0.
      FTEST=0.
      GO TO 330
C
C  CALCULATE COEFFICIENTS, FIT, AND CHI SQUARE
C
200   A0=YMEAN
      IF (NTERMS.LE.0) GO TO 230
      DO 220 J=1,NTERMS
          DO 210 K=1,NTERMS
```

```
210            A(J)=A(J)+R(K)*ARRAY(J,K)
         A(J)=A(J)*SIGMA/SIGMAX(J)
         A0=A0-A(J)*XMEAN(J)
         DO 220 I=1,NPTS
220      YFIT(I)=YFIT(I)+A(J)*FCTN(X,I,J,M,NPTS)
230   CONTINUE
      DO 240 I=1,NPTS
         YFIT(I)=YFIT(I)+A0
240      CHISQ=CHISQ+WEIGHT(I)*(Y(I)-YFIT(I))**2
      FREEN=NPTS-NTERMS-1
      CHISQR=CHISQ*WMEAN/FREEN
C
C  CALCULATE  UNCERTAINTIES
C
      IF (MODE) 250,260,250
250   VARNCE=1./WMEAN
      GO TO 270
260   VARNCE=CHISQR
 270  IF (NTERMS.LE.0) GO TO 290
      DO 280 J=1,NTERMS
         SIGMAA(J)=ARRAY(J,J)*VARNCE/(FREE1*SIGMAX(J)**2)
         SIGMAA(J)=SQRT(SIGMAA(J))
280      RMUL=RMUL+A(J)*R(J)*SIGMAX(J)/SIGMA
      FREEJ=NTERMS
      FTEST=(RMUL/FREEJ)/((1.-RMUL)/FREEN)
      RMUL=SQRT(RMUL)
      GO TO 300
290   CONTINUE
      FREEJ=0.
      FTEST=0.
      RMUL=0.
300   SIGMA0=VARNCE/FNPTS
      IF (NTERMS.LE.0) GO TO 320
      DO 310 J=1,NTERMS
         DO 310 K=1,NTERMS
310      SIGMA0=SIGMA0+VARNCE*XMEAN(J)*XMEAN(K)*ARRAY(J,K)/(FREE1*SIGMA
     2X(J)*SIGMAX(K))
320   SIGMA0=SQRT(SIGMA0)
330   RETURN
      END
```

```
$ASSM
READX PROG
$FORT
      FUNCTION READX(IWORD,N,NP,NTTYI,NTTYO,IEOF)
C
C $$$$$$$$$$  VERSION OF MAR 9,1977 $$$$$$$$$$
C
C  FUNCTION READX USUALLY RETURNS THE NEXT DECIPHERIBLE NUMBER ON THE
C    INPUT FILE =NTTYI= AS THE VALUE OF THE VARIABLE READX.
C    ANY OUPUT, SUCH AS ERROR MESSAGES OR THE PROMPT CHARACTER,
C    IS ON UNIT NTTYO.
C
C  IW= ALPHANUMERIC INFORMATION OF THE FIRST 1 OR 2 ILLEGAL NON-NUMERICAL
C    INFORMATION ON INPUT FILE NTTYI.
C  N=0 MEANS NUMERICAL INFORMATION SUCCESSFULLY PROCESSED.
C  N=1 CHECK IW WORD FOR NON-NUMERICAL INFORMATION.
C        (READX=1. IN THIS CASE).
C  NP=1  SEND ! AS A PROMPT CHARACTER WHEN NEW RECORD REQUIRED.
C  NP=2  READ A NEW RECORD.
C  NP=3  DO BOTH THE ABOVE.
C  NTTYI=  INPUT UNIT
C  NTTYO=  OUTPUT UNIT FOR PROMPT CHARACTER AND ERROR MESSAGES.
C  IEOF=0   END OF FILE NOT YET REACHED ON UNIT NTTYI.
C  IEOF=1  END OF FILE FOUND ON NTTYI.
C
C  ILLEGAL INFORMATION:
C
C  IF FIRST CHARACTER IS NOT A BLANK, COMMA, POINT, MINUS, OR
C    A NUMBER, THE FIRST 2 ILLEGAL CHARACTERS ARE RETURNED IN =IW=.
C
C  LEGAL INFORMATION*
C
C  LEADING BLANKS ARE IGNORED.
C  LEADING COMMAS ARE IGNORED.
C  DECIMAL POINTS AND EXPONENTIAL TYPE NUMBERS ARE TREATED CORRCECTLY.
C  $ ON THE INPUT FILE IS PRESUMED TO PRECEED AN IN-LINE COMMENT AND
C    CAUSES READING OF A NEW RECORD.
C  NUMBERS ARE TERMINATED BY A BLANK, COMMA, OR BY THE END OF A RECORD.
C
C
C  READX WAS ORIGINALLY WRITTEN BY FRANK MCCRACKIN (NBS) AND WAS MODIFIED
C    BY FRED MOPSIK AND BY BRIAN DICKENS.  (JAN 1977)
C
C
C
      INTEGER BRANCH
      INTEGER*2 CD,DEC,IW2,N1,J,IEXP,IEXP1
      LOGICAL ISIGNE
      DIMENSION CD(80),DEC(10),TEN(10),TENTEN(10)
      DATA DEC/'0','1','2','3','4','5','6','7','8','9'/
      DATA TEN/1.,1.E1,1.E2,1.E3,1.E4,1.E5,1.E6,1.E7,1.E8,1.E9/
      DATA TENTEN/1.,1.E10,1.E20,1.E30,1.E40,1.E50,1.E60,3*1.E70/
      DATA N1/81/
      IF (NP.EQ.2.OR.NP.EQ.3) N1 = 81
      IEOF = 0
      N = 0
```

```
      SIGN = 1.
      X = 0.
      Y = 1.
      YJ = 0.
C
C   FIND FIRST CHARACTER
C
15    IF (N1.GT.80) GO TO 1000
      IF (CD(N1).EQ.'$') GO TO 1000
      ASSIGN 16 TO BRANCH
      GO TO 2000
16    GO TO (40,100,30,20,20,300),IP
20    N1 = N1+1
      GO TO 15
C   SET SIGN
30    SIGN = -1.
      N1 = N1+1
      GO TO 15
C
C   CALCULATE NUMBER
C
40    X = J
50    N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 50
      ASSIGN 70 TO BRANCH
      GO TO 2000
70    GO TO (75,100,205,200,270,270),IP
75    X = X*10.+J
      GO TO 50
C
C   CALCULATE FRACTION
C
100   CONTINUE
110   N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 110
      ASSIGN 125 TO BRANCH
      GO TO 2000
125   GO TO (130,3055,205,200,270,270),IP
130   YJ = 10.*YJ+J
      Y = 10.*Y
      GO TO 110
C
C   CALCULATE EXPONENT IEXP
C
200   ISIGNE = .FALSE.
      IEXP = 1
      IEXP1 = 1
      GO TO 210
205   ISIGNE = .TRUE.
      IEXP = 1
      IEXP1 = 1
210   N1 = N1+1
      IF (N1.GT.80) GO TO 250
      ASSIGN 220 TO BRANCH
```

```
      GO TO 2000
220   GO TO (230,250,250,250,250,250),IP
230   IEXP1 = IEXP
      IEXP = J+1
      GO TO 210
250   IF (ISIGNE) GO TO 260
      READX = SIGN*(X+YJ/Y)*TEN(IEXP)*TENTEN(IEXP1)
      N = 0
      RETURN
260   READX = SIGN*(X+YJ/Y)/(TEN(IEXP)*TENTEN(IEXP1))
      N = 0
      RETURN
270   READX = SIGN*(X+YJ/Y)
      N = 0
      RETURN
C
C   RETURN ALPHA ARG
C
300   IW2 = CD(N1)
      N1 = N1+1
      IF (N1.GT.80) GO TO 302
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',' .OR. CD(N1).EQ.'$') GO TO 302
      IW2 = IW2-X'20'+CD(N1)/X'100'
      N1 = N1+1
302   IWORD = IW2
      N = 1
      READX = 1.
      RETURN
3055  WRITE (NTTYO,3056)N1
3056  FORMAT(' 2d . AT COL.',I3)
      GO TO 270
1000  N1 = 1
      IF (NP.EQ.1.OR.NP.EQ.3) WRITE (NTTYO,1020)
1020  FORMAT(' !')
      READ (NTTYI,1010,END = 9000) CD
1010  FORMAT(80A1)
      GO TO 15
C
C   CHARACTER INTERP. ROUTINE
C
2000  DO 2010 J = 1,10
      IF (CD(N1).EQ.DEC(J)) GO TO 2020
2010  CONTINUE
      IP = 6
      IF (CD(N1).EQ.'.') IP = 2
      IF (CD(N1).EQ.'-') IP = 3
      IF (CD(N1).EQ.'+') IP = 4
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',') IP = 5
      GO TO 2030
2020  IP = 1
      J = J-1
2030  GO TO BRANCH
 9000 IEOF = 1
      RETURN
      END
```

```
$ASSM
SORT3 PROG
$FORT
$TRGT 16
C      ************************************************************************
C
       SUBROUTINE SORT3 (A,B,C,N)
C
C      *************************************************************************
C
C
C     THIS SUBROUTINE SORTS A 1-DIMENSIONAL ARRAY
C     (A) AND THE ARRAYS (B) AND (C) INTO AN ORDER BASED
C     ON THE ELEMENTS IN ARRAY (A).
C
       DIMENSION A(N),B(N),C(N)
C
       IF (N.LE.1) RETURN
       DO 20 I = 2,N
       IM1 = I-1
       DO 10 J = 1,IM1
       JJ = I-J
       IF (A(JJ).LE.A(JJ+1)) GO TO 20
       X = A(JJ)
       A(JJ) = A(JJ+1)
       A(JJ+1) = X
       X = B(JJ)
       B(JJ) = B(JJ+1)
       B(JJ+1) = X
       X = C(JJ)
       C(JJ) = C(JJ+1)
       C(JJ+1) = X
10     CONTINUE
20     CONTINUE
       RETURN
       END
```

```
$ASSM
TDEGAB PROG
$FORT
$TRGT 16
C      **********************************************************************
C
      FUNCTION TDEGAB (T,AT)
C
C      *******************************************************************
C
C  FUNCTION TO CALCULATE TEMPERATURE IN DEGREES ABSCLUTE
C  FROM THERMOCOUPLE READING
C
      DIMENSION AT(4)
C
      TP = 0
      DO 10 K = 1,4
10    TP = TP+AT(K)*T**K
      TDEGAB = TP+273.15
      RETURN
      END
```

```
$ASSM
THMPLS PROG
$FORT
$TRGT 16
C    *************************************************************************
C
      SUBROUTINE THMCPL (AT)
C
C     **************************************************************************
C
C   SUBROUTINE TO READ THERMOCOUPLE TYPE AND PROVIDE COEFFICIENTS FOR
C   CONVERSION OF THERMOCOUPLE READINGS IN MICROVOLTS TO DEGREES CELCIUS
C   NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION AT(4)
C
C
      WRITE (NTTYO,10)
10    FORMAT (/' GIVE THERMOCOUPLE TYPE, K OR E')
20    READ (NTTYI,30) NTHCP
30    FORMAT (A1)
      ITHCP = 4
      IF (NTHCP.EQ.'K')ITHCP = 1
      IF (NTHCP.EQ.'E')ITHCP = 2
      GO TO (40,60,80,90), ITHCP
C
C   TYPE K THERMOCOUPLE,VALUES FROM TABLE A7.2.3,NBS MONO.125
C
      IF (NFILEB.GT.6) WRITE (NFILEB,50)
40    IF (NPRINT.GE.3) WRITE (NPRINT,50)
50    FORMAT (/52H TYPE K THERMOCOUPLE = NI-CR/NI-AL,EXPRESSIONS GOOD /
     &,46H TO 400 DEGREES C.(TABLE A7,NBS MONO.125,1974))
      AT(1) = 2.4383248E-02
      AT(2) = 9.7830251E-09
      AT(3) = 3.6276965E-12
      AT(4) = -2.5756438E-16
      GO TO 110
C
C   TYPE E THERMOCOUPLE
C
      IF (NFILEB.GT.6) WRITE (NFILEB,70)
60    IF (NPRINT.GE.3) WRITE (NPRINT,70)
70    FORMAT (/52H TYPE E THERMOCOUPLE = NI-CR/CU-NI,EXPRESSIONS GOOD /
     &,58HFROM -0.05 TO +0.04, 0 TO 400 DEGREES C.(TABLE A5.2.3,NBS /,
     & 19HMONOGRAPH 125,1974))
      AT(1) = 1.7022525E-02
      AT(2) = -2.2097240E-07
      AT(3) = 5.4809314E-12
      AT(4) = -5.7669892E-17
      GO TO 110
C
C   DUMMY LOCATION
```

```
C
80     IF (NFILEB.GT.6) WRITE (NFILEB,100)
       IF (NPRINT.GE.3) WRITE (NPRINT,100)
       GO TO 20
C
C  ERROR ROUTE
C
   90 WRITE (NTTYO,100)
100    FORMAT (/40H WRONG SPECIFICATION FOR THERMOCOUPLE,TRY AGAIN)
       GO TO 20
110    RETURN
       END
```

```
$ASSM
TPPF  PROG  D124:TPPF.FTN
$FORT
      SUBROUTINE TPPF(P,NU,PPF)
C
C     PURPOSE--THIS SUBROUTINE COMPUTES THE PERCENT POINT
C              FUNCTION VALUE FOR THE STUDENT'S T DISTRIBUTION
C              WITH INTEGER DEGREES OF FREEDOM PARAMETER = NU.
C              THE STUDENT'S T DISTRIBUTION USED
C              HEREIN IS DEFINED FOR ALL X,
C              AND ITS PROBABILITY DENSITY FUNCTION IS GIVEN
C              IN THE REFERENCES BELOW.
C              NOTE THAT THE PERCENT POINT FUNCTION OF A DISTRIBUTION
C              IS IDENTICALLY THE SAME AS THE INVERSE CUMULATIVE
C              DISTRIBUTION FUNCTION OF THE DISTRIBUTION.
C     INPUT  ARGUMENTS--P      = THE SINGLE PRECISION VALUE
C                                (BETWEEN 0.0 (EXCLUSIVELY)
C                                AND 1.0 (EXCLUSIVELY))
C                                AT WHICH THE PERCENT POINT
C                                FUNCTION IS TO BE EVALUATED.
C                     --NU     = THE INTEGER NUMBER OF DEGREES
C                                OF FREEDOM.
C                                NU SHOULD BE POSITIVE.
C     OUTPUT ARGUMENTS--PPF    = THE SINGLE PRECISION PERCENT
C                                POINT FUNCTION VALUE.
C     OUTPUT--THE SINGLE PRECISION PERCENT POINT FUNCTION .
C             VALUE PPF FOR THE STUDENT'S T DISTRIBUTION
C             WITH DEGREES OF FREEDOM PARAMETER = NU.
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.
C     RESTRICTIONS--NU SHOULD BE A POSITIVE INTEGER VARIABLE.
C                 --P SHOULD BE BETWEEN 0.0 (EXCLUSIVELY)
C                   AND 1.0 (EXCLUSIVELY).
C     OTHER DATAPAC   SUBROUTINES NEEDED--NORPPF.
C     FORTRAN LIBRARY SUBROUTINES NEEDED--DSIN, DCOS, DSQRT, DATAN.
C     MODE OF INTERNAL OPERATIONS--DOUBLE PRECISION.
C     LANGUAGE--ANSI FORTRAN.
C     COMMENT--FOR NU = 1 AND NU = 2, THE PERCENT POINT FUNCTION
C              FOR THE T DISTRIBUTION EXISTS IN SIMPLE CLOSED FORM
C              AND SO THE COMPUTED PERCENT POINTS ARE EXACT.
C            --FOR OTHER SMALL VALUES OF NU (NU BETWEEN 3 AND 6,
C              INCLUSIVELY), THE APPROXIMATION
C              OF THE T PERCENT POINT BY THE FORMULA
C              GIVEN IN THE REFERENCE BELOW IS AUGMENTED
C              BY 3 ITERATIONS OF NEWTON'S METHOD FOR
C              ROOT DETERMINATION.
C              THIS IMPROVES THE ACCURACY--ESPECIALLY FOR
C              VALUES OF P NEAR 0 OR 1.
C     REFERENCES--NATIONAL BUREAU OF STANDARDS APPLIED MATHMATICS
C                 SERIES 55, 1964, PAGE 949, FORMULA 26.7.5.
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE
C                 DISTRIBUTIONS--2, 1970, PAGE 102,
C                 FORMULA 11.
C               --FEDERIGHI, 'EXTENDED TABLES OF THE
C                 PERCENTAGE POINTS OF STUDENT'S T
C                 DISTRIBUTION, JOURNAL OF THE
C                 AMERICAN STATISTICAL ASSOCIATION,
```

```
C                 1969, PAGES 683-638.
C              --HASTINGS AND PEACOCK, STATISTICAL
C                DISTRIBUTIONS--A HANDBOOK FOR
C                STUDENTS AND PRACTITIONERS, 1975,
C                PAGES 120-123.
C     WRITTEN BY--JAMES J. FILLIBEN
C                STATISTICAL ENGINEERING LABORATORY (205.03)
C                NATIONAL BUREAU OF STANDARDS
C                WASHINGTON, D. C. 20234
C                PHONE:  301-921-2315
C     ORIGINAL VERSION--OCTOBER   1975.
C     UPDATED         --NOVEMBER  1975.
C
C-----------------------------------------------------------------------
C
      DOUBLE PRECISION PI
      DOUBLE PRECISION SQRT2
      DOUBLE PRECISION DP
      DOUBLE PRECISION DNU
      DOUBLE PRECISION TERM1,TERM2,TERM3,TERM4,TERM5
      DOUBLE PRECISION DPPFN
      DOUBLE PRECISION DPPF,DCON,DARG,Z,S,C
      DOUBLE PRECISION B21
      DOUBLE PRECISION B31,B32,B33,B34
      DOUBLE PRECISION B41,B42,B43,B44,B45
      DOUBLE PRECISION B51,B52,B53,B54,B55,B56
      DOUBLE PRECISION D1,D3,D5,D7,D9
      DATA PI/3.14159265358979D0/
      DATA SQRT2/1.41421356D0/
      DATA B21/0.25D0/
      DATA B31,B32,B33,B34/0.01041666666667D0,5.0D0,16.0D0,3.0D0/
      DATA B41,B42,B43,B44,B45/0.00260416666667D0,3.0D0,19.0D0,17.0D0,
     1                         -15.0D0/
      DATA B51,B52,B53,B54,B55,B56/0.00001085069444D0,79.0D0,776.0D0,
     1                              1482.0D0,-1920.0D0,-945.0D0/
C
      IPR=6
C
C     CHECK THE INPUT ARGUMENTS FOR ERRORS
C
      IF(P.LE.0.0.OR.P.GE.1.0)GOTO50
      GOTO90
   50 WRITE(IPR,1)
      WRITE(IPR,46)P
      RETURN
   90 CONTINUE
    1 FORMAT(1H ,115H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THE
     1 TPPF   SUBROUTINE IS OUTSIDE THE ALLOWABLE (0,1) INTERVAL *****)
   46 FORMAT(1H , 35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)
C
C-----START POINT-------------------------------------------------------
C
      DNU=NU
      DP=P
      MAXIT=5
C
```

```
      IF(NU.GE.3)GOTO250
      IF(NU.EQ.1)GOTO100
      IF(NU.EQ.2)GOTO200
      WRITE(IPR,105)
  105 FORMAT(1H ,33HINTERNAL ERROR IN TPPF SUBROUTINE)
      PPF=0.0
      RETURN
C
C     TREAT THE NU = 1 (CAUCHY) CASE
C
  100 DARG=PI*DP
      PPF=-DCOS(DARG)/DSIN(DARG)
      RETURN
C
C     TREAT THE NU = 2 CASE
C
  200 TERM1=SQRT2/2.0D0
      TERM2=2.0D0*DP-1.0D0
      TERM3=DSQRT(DP*(1.0D0-DP))
      PPF=TERM1*TERM2/TERM3
      RETURN
C
C     TREAT THE NU GREATER THAN OR EQUAL TO 3 CASE
C
  250 CALL NORPPF(P,PPFN)
      DPPFN=PPFN
      D1=DPPFN
      D3=DPPFN**3
      D5=DPPFN**5
      D7=DPPFN**7
      D9=DPPFN**9
      TERM1=D1
      TERM2=B21*(D3+D1)/DNU
      TERM3=B31*(B32*D5+B33*D3+B34*D1)/(DNU**2)
      TERM4=B41*(B42*D7+B43*D5+B44*D3+B45*D1)/(DNU**3)
      TERM5=B51*(B52*D9+B53*D7+B54*D5+B55*D3+B56*D1)/(DNU**4)
      DPPF=TERM1+TERM2+TERM3+TERM4+TERM5
      PPF=DPPF
      IF(NU.GE.7)RETURN
      IF(NU.EQ.3)GOTO300
      IF(NU.EQ.4)GOTO400
      IF(NU.EQ.5)GOTO500
      IF(NU.EQ.6)GOTO600
      RETURN
C
C     AUGMENT THE RESULTS FOR THE NU = 3 CASE
C
  300 DCON=PI*(DP-0.5D0)
      DARG=DPPF/DSQRT(DNU)
      Z=DATAN(DARG)
      DO350IPASS=1,MAXIT
      S=DSIN(Z)
      C=DCOS(Z)
      Z=Z-(Z+S*C-DCON)/(2.0D0*C*C)
  350 CONTINUE
      PPF=DSQRT(DNU)*S/C
```

```
      RETURN
C
C     AUGMENT THE RESULTS FOR THE NU = 4 CASE
C
  400 DCON=2.0D0*(DP-0.5D0)
      DARG=DPPF/DSQRT(DNU)
      Z=DATAN(DARG)
      DO450IPASS=1,MAXIT
      S=DSIN(Z)
      C=DCOS(Z)
      Z=Z-((1.0D0+0.5D0*C*C)*S-DCON)/(1.5D0*C*C*C)
  450 CONTINUE
      PPF=DSQRT(DNU)*S/C
      RETURN
C
C     AUGMENT THE RESULTS FOR THE NU = 5 CASE
C
  500 DCON=PI*(DP-0.5D0)
      DARG=DPPF/DSQRT(DNU)
      Z=DATAN(DARG)
      DO550IPASS=1,MAXIT
      S=DSIN(Z)
      C=DCOS(Z)
      Z=Z-(Z+(C+(2.0D0/3.0D0)*C*C*C)*S-DCON)/((8.0D0/3.0D0)*C**4)
  550 CONTINUE
      PPF=DSQRT(DNU)*S/C
      RETURN
C
C     AUGMENT THE RESULTS FOR THE NU = 6 CASE
C
  600 DCON=2.0D0*(DP-0.5D0)
      DARG=DPPF/DSQRT(DNU)
      Z=DATAN(DARG)
      DO650IPASS=1,MAXIT
      S=DSIN(Z)
      C=DCOS(Z)
      Z=Z-((1.0D0+0.5D0*C*C+0.375D0*C**4)*S-DCON)/((15.0D0/8.0D0)*C**5)
  650 CONTINUE
      PPF=DSQRT(DNU)*S/C
      RETURN
C
      END
```

```
$ASSM
TRIM PROG
$FORT
$TRGT 16
C   ****************************************************************
C
      SUBROUTINE TRIM (E,SIGE,DOC,NDATA)
C
C   ******************************************************************
C
C   SUBROUTINE TO CALCULATE TRIMMED MEANS USING E(I)
C     UPTO 1/3 OF VALUES AT EACH END WILL BE CUT OFF
C
C    NEEDS THE FOLLOWING SUBROUTINES AND FUNCTIONS:
C     SORT3
C     WTAVEW
C
C
C   COMMON BLOCKS /IO/ AND /REGRS/ DEFINED VIA ARRAY 'DUMP'
C      COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C      COMMON /REGRS/ M(5),R(5),YFIT(250)
C
      COMMON  DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
C
      DIMENSION E(1),SIGE(1),DOC(1)
      IF (NDATA.GT.4) GO TO 10
      IF (NFILEB.GT.6) WRITE (NFILEB,50) NDATA
      IF (NPRINT.GE.3) WRITE (NPRINT,50) NDATA
   50 FORMAT(/' NO TRIMMED MEANS CAN BE CALCULATED BECAUSE THERE'
     &/' ARE ONLY',I2,' DATA'/)
      RETURN
   10 NTRIMS=NDATA/3
      IF (NTRIMS.LT.1) NTRIMS = 1
C
C   ESTIMATE AVERAGE VALUES WHEN NO DATA REJECTED
C
      IF (NFILEB.GT.6) WRITE (NFILEB,80)
      IF (NPRINT.GE.3) WRITE (NPRINT,80)
   80 FORMAT('1  AVERAGE VALUES BEFORE DATA TRIMMED' /)
      CALL WTAVEW (E,SIGE,NDATA,WTEAVE,SIGWTE,EAVE,ESIG,1,NDATA,0)
C
C   SORT ON BASIS OF E
C
      CALL SORT3 (E,SIGE,DOC,NDATA)
C
C   LOOP SEVERAL TIMES CUTTING ONE MORE E AT EACH END EACH TIME
C
      JUMP = NDATA/10
      IF (JUMP.LT.1) JUMP = 1
      DO 30 I=JUMP,NTRIMS,JUMP
      JTRIMS = NDATA -2*I
      NBOT = 1+I
      NTOP = NDATA-I
      IF (NFILEB.GT.6) WRITE (NFILEB,20) I,JTRIMS
      IF (NPRINT.GE.3) WRITE (NPRINT,20) I,JTRIMS
   20 FORMAT('1  TRIMMED MEANS FOLLOW:'
     &/I3,' DATA CHOPPED OFF EACH END AND ',I3,' DATA USED'
     &//'       E   SIG E     DOC'/)
      DO 90 J = NBOT,NTOP
      IF (NFILEB.GT.6) WRITE (NFILEB,40) E(J),SIGE(J),DOC(J)
```

```
      IF (NPRINT.GE.3) WRITE (NPRINT,40) E(J),SIGE(J),DOC(J)
   40 FORMAT (3F7.2)
   90 CONTINUE
C
C  CALCULATE TRIMMED MEAN AND SIGMA FROM SUBSET
C
      CALL WTAVEW (E(NBOT),SIGE(NBOT),JTRIMS,WTEAVE,SIGWTE,EAVE,ESIG
     & ,NBOT,NTOP,0)
C
C  CALCULATE WINSORISED MEAN AND SIGMA
C
      CALL WTAVEW(E,SIGE,NDATA,WTEAVE,SIGWTE,EAVE,ESIG,NBOT,NTOP,JTRIMS)
   30 CONTINUE
      RETURN
      END
```

```
$ASSM
TRIMRD PROG
$FORT
$TRGT 16
C   ****************************************************************
C
      SUBROUTINE TRIMRD(NTITLE,NLOOPY,IDOLD,E,SIGE,SWT,DOC,NTIME,TREAD
     & ,SIGT,RATE,SIGR,ID,INDATA,NDATA,NERROR)
C
C   ******************************************************************
C
C   NEEDS THE FOLLOWING SUBROUTINES AND FUNCTIONS:
C     ACTIV
C     EREAD
C     READEM
C     READX
C     AND COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION NTITLE(1),NTIME(1),TREAD(2,75),RATE(2,75)
      DIMENSION TEMP(2,75),SIGT(2,75),SIGR(2,75),E(1),SIGE(1)
      DIMENSION NSTAR(32),SWT(1),DOC(1)
      DIMENSION AT(4),IDOLD(1),ID(1)
      NLOOPY = NLOOPY+1
C
C
      WRITE (NTTYO,110) NLOOPY
 110  FORMAT (12H FILE NUMBER,I3)
C
C  READ AND WRITE TITLE OF INPUT FILE
C
      IF (NPRINT.GE.3) WRITE (NPRINT,10)
   10 FORMAT('1')
      READ (IN,120) (NTITLE(I),I=1,64)
 120  FORMAT (32A2/32A2)
      DO 130 I = 1,32
 130  NSTAR(I) = 2H**
      IF (NFILEB.GT.6) WRITE (NFILEB,140)
      IF (NFILEB.GT.6) WRITE (NFILEB,120) (NTITLE(I),I=1,64)
      IF (NPRINT.LT.3) GO TO 151
      WRITE (NPRINT,120) NSTAR
      WRITE (NPRINT,140)
 140  FORMAT (25H  TITLE OF INPUT FILE IS ://)
      WRITE (NPRINT,120) (NTITLE(I),I=1,64)
      WRITE (NPRINT,150)
 150  FORMAT (//)
      WRITE (NPRINT,120) NSTAR
      WRITE (NPRINT,150)
C
C   READ IN DATA
C
  151 IF (INDATA.GT.0)GO TO 85
C
C  GET THERMOCOUPLE COEFFICIENTS, READ IN DATA
C
      CALL THMCPL (AT)
      CALL READEM (NTIME,TREAD,SIGT,RATE,SIGR,ID,NDATA)
C
C  CALCULATE E ACT. AND ASSOCIATED SIGMA
C
      DO 170 I = 1,NDATA
      IDOLD(I) = ID(I)
```

```
      CALL ACTIV (E(I),SIGE(I),TEMP(1,I),TEMP(2,I),SIGT(1,I),SIGT(2,I)
     & ,TREAD(1,I),TREAD(2,I),SIGT(1,I),SIGT(2,I)
     & ,RATE(1,I),RATE(2,I),SIGR(1,I),SIGR(2,I)
     & ,AT,NERROR)
      IF (NERROR.GT.0) RETURN
  170 CONTINUE
      GO TO 300
C
C  READ IN E VALUES
C
   85 IF (INDATA.EQ.1) GO TO 88
      WRITE (NTTYO,89)
   89 FORMAT (/' READ INITIAL AND FINAL SAMPLE WEIGHTS, FREE FORMAT')
      WINITL = READX (IW,NERROR,3,IN,NTTYO,IEOF)
      WFINAL = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SAMPWT = WINITL-WFINAL
      IF (NFILEB.GT.6) WRITE(NFILEB,81) WINITL,WFINAL
      IF (NPRINT.GE.3) WRITE(NPRINT,81) WINITL,WFINAL
   81 FORMAT(/' INITIAL AND FINAL WEIGHTS READ FROM INPUT FILE AS'
     & /2F15.3)
C
C
   88 CALL EREAD (IDOLD,E,SIGE,SWT,DOC,NDATA,INDATA,NERROR)
      IF (INDATA.EQ.1) GO TO 87
      DO 86 I = 1,NDATA
   86 DOC(I) = ((WINITL-SWT(I))/SAMPWT)*100.
   87 CONTINUE
C
C  OUTPUT WHATEVER WAS INPUT
C
      IF (NFILEB.GT.6) WRITE (NFILEB,100)
      IF (NPRINT.GE.3) WRITE (NPRINT,100)
  100 FORMAT(/' NUMBER',9X,'E',5X,'SIG E',4X,'D.O.C.',12X,'SWT'/)
      IF (NFILEB.GT.6) WRITE (NFILEB,210) (IDOLD(I),E(I),SIGE(I),DOC(I)
     & ,SWT(I),I=1,NDATA)
      IF (NPRINT.GE.3) WRITE (NPRINT,210) (IDOLD(I),E(I),SIGE(I),DOC(I)
     & ,SWT(I),I=1,NDATA)
  210 FORMAT(I7,3F10.2,F15.2)
  300 IF (NFILEB.GT.6) WRITE (NFILEB,70) NDATA
      IF (NPRINT.GE.3) WRITE (NPRINT,70) NDATA
   70 FORMAT(//I6,' DATA READ IN'//)
      RETURN
      END
```

```
$ASSM
WTAVEW PROG
$FORT
$TRGT 16
C ************************************************************
C
      SUBROUTINE WTAVEW (E,SIGE,NDATA,WTEAVE,SIGWTE,EAVE,ESIG
     & ,NBOT,NTOP,NWINS)
C
C ************************************************************
C
C   *** WINSORISED VERSION ***
C
C   SUBROUTINE TO CALCULATE AVERAGE E (WTEAVE) AND STANDARD DEVIATION (SIGWTE)
C     FROM INPUT LIST E AND SIGE CONTAINING  = NDATA = VALUES
C   ALSO CALCULATE UNWEIGHTED AVERAGE AND 95% CONFIDENCE RANGES
C
C   NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C     COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C     COMMON /REGRS/ M(5),R(5),YFIT(250)
C
C   NEEDS SUBROUTINES:
C     AVSDN
C     CHISQ
C     REGRES
C     TPPF
C
C   NWINS.GT.0  MEANS ESTIMATE WINSORISED MEAN AND SIGMA,
C     I.E., REJECT ALL INPUT VALUES BEFORE NBOT IN LIST AND ALL THOSE
C     AFTER NTOP.  ALSO, REPLACE THE REJECTED VALUES WITH THE NEAREST
C     UNREJECTED VALUES (NBOT OR NTOP IN LIST AS THE CASE MAY BE).
C
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
      DIMENSION E(1),SIGE(1),A(5),SIGA(5)
C
C   CALCULATE AVERAGE UNWEIGHTED E
C
      WTEAVE = 0.
      SIGWTE = 1.E-09
      EAVE = 0.
      ESIG = 1.E-09
      SIGAVE = 1.E-09
      FREE = 0.
      CHIWTE = 0.
      CHIUNW = 0.
      IF (NDATA.LE.0)RETURN
      IF (NDATA.GT.1)GO TO 200
      WTEAVE = E(1)
      SIGWTE = SIGE(1)
      SIGAVE = SIGE(1)
      EAVE = E(1)
      ESIG = SIGE(1)
      RETURN
  200 CONTINUE
      IF (NWINS.LE.0) GO TO 201
C
C   GET READY TO CALCULATE WINSORISED MEAN AND SIGMA
C
      DO 70 J=1,NBOT
```

```
      E(J)=E(NBOT)
   70 SIGE(J)=SIGE(NBOT)
      DO 88 J=NTOP,NDATA
      E(J)=E(NTOP)
   88 SIGE(J)=SIGE(NTOP)
      IF (NFILEB.GT.6) WRITE(NFILEB,85)
      IF (NPRINT.GE.3) WRITE(NPRINT,85)
   85 FORMAT(//' WINSORISED MEAN AND SIGMA FOLLOW:'
     &/' (I.E., REPLACE REJECTED VALUES WITH NEAREST UNREJECTED VALUE)'
     &/)
  201 CONTINUE
      CALL AVSDN (EAVE,ESIG,E,NDATA)
      CALL CHISQ (E,SIGE,CHIUNW,NDATA)
C
C     CALCULATE AVERAGE SIGE
C
      CALL AVSDN (SIGAVE,SIGSIG,SIGE,NDATA)
C
C     WINSORISE
C
      IF (NWINS.GT.0) ESIG=ESIG*SQRT(FLOAT(NDATA-1)/(FLOAT(NWINS-1)
     & *FLOAT(NWINS)))
C
C     CALCULATE WEIGHTED AVERAGE OF E
C
      CALL REGRES (E,E,SIGE,NDATA,0,M,1,YFIT,A0,A,SIGA0,SIGA
     &,R,RMUL,CHIWTE,FTEST)
      WTEAVE = A0
C
C     CALCULATE SIGMA FOR WEIGHTED E AVERAGE
C
      SIGWTE = 0.
      DO 10 I = 1,NDATA
   10 SIGWTE = SIGWTE+1./SIGE(I)**2
      SIGWTE = SQRT(1./SIGWTE)
      FDATA = NDATA-1
      FREE = NDATA-2
      IF(NWINS.LE.0) ESIG = ESIG/SQRT(FDATA)
C
C     CALCULATE 95% CONFIDENCE RANGE
C
      NDOF = NDATA-2
      CALL TPPF (0.95,NDOF,CUTZ)
      T = CUTZ*SIGWTE
      WTEHI = WTEAVE+T
      WTELO = WTEAVE-T
      T= CUTZ*ESIG
      UNWEHI = EAVE+T
      UNWELO = EAVE-T
C
C     OUTPUT AVERAGES
C
      IF (NPRINT.LT.3) GO TO 100
      IF (NWINS.GT.0) GO TO 45
      WRITE (NPRINT,30) WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
30    FORMAT (/' WEIGHTED MEAN E IS ',F7.3,' + OR - ',F7.3/
     & ' REDUCED CHISQ FOR E CALCULATION IS',F7.3,' (SHOULD BE 1.)'/
     & ' UNWEIGHTED MEAN OF SIGMAS IS ',F7.3,/
     &,' CALCULATED OVER ',F5.0,' DEGREES OF FREEDOM')
C
C     OUTPUT CONFIDENCE LIMITS
C
      WRITE (NPRINT,40) WTELO,WTEHI
40    FORMAT (/' ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY'/
     & ' DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE',F7.3,' TO',
     & F7.3/' FOR 95% CONFIDENCE LEVEL')
```

```
   45 WRITE (NPRINT,50) EAVE,ESIG,UNWELO,UNWEHI,CHIUNW,CUTZ
50    FORMAT (/' UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
     & /' OF ITS SIGMA ARE',F7.3,' AND',F7.3
     & /' THIS GIVES A CONFIDENCE RANGE OF',F7.3,' TO',F7.3
     &/' CHI-SQUARE FOR UNWEIGHTED E IS',F7.2
     &/' T VALUE APPLIED IS',F7.3/)
  100 IF (NFILEB.LE.6)RETURN
      IF (NWINS.GT.0) GO TO 105
      WRITE (NFILEB,30)WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
      WRITE (NFILEB,40)WTELO,WTEHI
  105 WRITE (NFILEB,50)EAVE,ESIG,UNWELO,UNWEHI,CHIUNW,CUTZ
      RETURN
      END
```

File LOTRIM, which automatically assigns all files needed to assemble the TRIMPG overlays and drives the LIBLDR program to make the overlays. Use as DO LOTRIM,DO.

```
AC TRIMPG,1D0
AS 362
AC TRIML5,5D0
AC TGTRIB,7D0
AC OVERLB,8D0
AC FORLIB,FD0
RUN LIBLDR,D0
CL
AC TRIM01,1D0
AC PRIMTB,7D0
AC TRIMRB,8D0
AC THMCSB,9D0
AC READMB,AD0
AC ACTIVB,BD0
AC READXB,CD0
AC EREADB,DD0
AC TDEGBB,ED0
ST
CL
AC TRIM02,1D0
AC TRIMB,7D0
AC WTAVWB,8D0
AC TPFFB,9D0
AC NORPFB,AD0
AC SORT3B,BD0
AC AVSDNB,CD0
AC CHISQB,DD0
AC REGRSB,ED0
AC FCTNB,2D0
AC MATINB,4D0
ST
CL
ZU
TR
```

File TRIML5, which is assigned by LOTRIM during assembly of the TRIMPG overlays. This file contains the appropriate responses to the LIBLDR program.

```
OUT 1 TRIMPG
BI 3576
BC 044C
LO 7
LI 8
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 TRIM01
OV
LI 7
LI 8
LI 9
LI A
LI B
LI C
LI D
LI E
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 TRIM02
OV
LI 7
LI 8
LI 9
LI A
LI B
LI C
LI D
LI E
LI 2
LI 4
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
```

Memory map of the base segment of the TRIMPG program.

```
REL PROGS:
  3576 TGTRIM    4A6C OVERLY    4BCA IFETCH    4C3C .H
  4C8E .S        4CC4 .P        4D46 .Q        4DF6 .O
  4E3E .MES      4EB4 .U        4EE4 .V        4EF2 @R
  4F18 @H        6162

ABS PROGS:
NONE

ENTRY-POINTS:
  4A90 OVERLY    4BCC IFETCH    4C3E .H        4C90 .S
  4CC6 .P        4D48 .Q        4DF8 .O        4E40 .MES
  4EC2 .U        4EEA .V        4EF4 @R        4F66 @H

COMMON-BLOCKS:
  FBB2 //

UNDEFINED:
   PRIMTR    TRIMRD    TRIM
```

Memory map for the initialization segment TRIM01 of the TRIMPG overlay.

```
REL PROGS:
  3576 TGTRIM    4A6C OVERLY    4BCA IFETCH    4C3C .H
  4C8E .S        4CC4 .P        4D46 .Q        4DF6 .O
  4E3E .MES      4EB4 .U        4EE4 .V        4EF2 @R
  4F18 @H        62CA PRIMTR    6638 TRIMRD    7472 THMPLS
  786C READEM    7E82 ACTIV     86C4 READX     8E88 EREAD
  90EE TDEGAB    919C .R        91D8 .A        927E SQRT
  9386 ALOG      9494 EXP       959A AINT      9612 .Y
  9618 .Y2       9698 ABS       96AE .W        96FC .COMP
  9728 $6        9772 .RARG     979E $8        97C8 .5
  97D0 .ZERO     97D8

ABS PROGS:
NONE

ENTRY-POINTS:
  4A90 OVERLY    4BCC IFETCH    4C3E .H        4C90 .S
  4CC6 .P        4D48 .Q        4DF8 .O        4E40 .MES
  4EC2 .U        4EEA .V        4EF4 @R        4F66 @H
  62EE PRIMTR    665C TRIMRD    7496 THMCPL    7890 READEM
  7EA6 ACTIV     86E8 READX     8EAC EREAD     9112 TDEGAB
  919E .R        91DA .A        9280 SQRT      9388 ALOG
  9496 EXP       959C AINT      9614 .Y        961A .Y2
  969A ABS       96B0 .W        96FE .COMP     972A $6
  9774 .RARG     97A0 $8        97CC .5        97D4 .ZERO

COMMON-BLOCKS:
  FBB2 //

UNDEFINED:
   TRIM
```

Memory map for the segment TRIM02, which calculates trimmed means in the TRIMPG program.

```
REL PROGS:
  3576 TGTRIM    4A6C OVERLY    4BCA IFETCH    4C3C .H
  4C8E .S        4CC4 .P        4D46 .Q        4DF6 .O
  4E3E .MES      4EB4 .U        4EE4 .V        4EF2 @R
  4F18 @H        62CA TRIM      6872 WTAVEW    7320 TPPF
  81C6 NORPPF    849E SORT3     869C AVSDN     8824 CHISQ
  8982 REGRES    9F4A FCTN      9FD2 MATINV    A7D0 DCOS
  A84A .D        A858 .COMEX    AA26 DSIN      ABCC DLOG
  AF3C DEXP      B114 DSQRT     B316 DATAN     B680 $7
  B6DE $9        B716 DMOD      B788 DAINT     B84C @A
  B866 @1        B874 @S        B896 .X        B8C6 DABS
  B8E0 .G        B8F6 @F        B9FC @L        BA2C @K
  BA5C @M        BBF8 @2        BC06 @D        BD20 @T
  BD48 @E        BD6C @O        BD88 @N        BDE2 @U
  BDFA @V        BE1A @P        BE4A @B        BE5A .DARG
  BE9C .DDARG    BF32 .DSWAP    BF8E .DRTN     BFB6 .R
  BFF2 .A        C098 SQRT      C1A0 ALOG      C2AE EXP
  C3B4 AINT      C42C FLOAT     C432 FLOAT2    C46A .W
  C498 .COMP     C4E4 $6        C52E .RARG     C55A $8
  C584 .5        C58C .ZERO     C594 @H        D7DE

ABS PROGS:
NONE

ENTRY-POINTS:
  4A90 OVERLY    4BCC IFETCH    4C3E .H        4C90 .S
  4CC6 .P        4D48 .Q        4DF8 .O        4E40 .MES
  4EC2 .U        4EEA .V        4EF4 @R        62EE TRIM
  6896 WTAVEW    7344 TPPF      81EA NORPPF    84C2 SORT3
  86C0 AVSDN     8848 CHISQ     89A6 REGRES    9F6E FCTN
  9FF6 MATINV    A7D2 DCOS      A84C .D        A85A .COMEX
  AA28 DSIN      ABCE DLOG      AF3E DEXP      B116 DSQRT
  B318 DATAN     B682 $7        B6E0 $9        B718 DMOD
  B78A DAINT     B84E @A        B868 @1        B876 @S
  B898 .X        B8C8 DABS      B8E2 .G        B8F8 @F
  B9FE @L        BA2E @K        BA5E @M        BBFA @2
  BC08 @D        BD22 @T        BD4A @E        BD6E @O
  BD8A @N        BDE4 @U        BDFC @V        BE1C @P
  BE4C @B        BE5C .DARG     BE9E .DDARG    BF34 .DSWAP
  BF90 .DRTN     BFB8 .R        BFF4 .A        C09A SQRT
  C1A2 ALOG      C2B0 EXP       C3B6 AINT      C42E FLOAT
  C434 FLOAT2    C46C .W        C4BA .COMP     C4E6 $6
  C530 .RARG     C55C $8        C588 .5        C590 .ZERO
  C5E2 @H

COMMON-BLOCKS:
  FBB2 //

UNDEFINED:
NONE
```

Appendix D. TGDEPG, a FORTRAN program to calculate activation energies from rate and temperature data, to detect outliers and to make normal probability plots.

Listing of file SETDE, used as SETDE,D0 to assign several files needed by the TGDEPG program.

```
AS 120
AS 220
AS 320
AS 420
AS 520
AS 662
AS 720
AS 820
AS 920
AS A20
AC TGDE01.BD0
AC TGDE02.CD0
AS D20
AS E20
AS F20
ASSIGN INPUT (FILE 7) AND OUTPUT (FILE 8) THEN TYPE 'RUN TGDEPG.D0'
TR
```

Example of input for activation-energy-calculating-probability-plot program TGDEPG. User input is underlined. Start the procedure by typing DO SETGDE,DO.

```
  TGDEPG PROGRAM
READ INPUT DATA FROM FILE  7
 FILE   6 IS FOR OUTPUT
 PUT PROCESSED DATA ON FILE   8
 GIVE NUMBER OF FILES TO EXAMINE (IN FREE FORMAT)
 !
 1
 EXAMINE    1 FILES
 GIVE CHANCE OF REJECTING EXTREME VALUE IF
 FLUCTUATIONS ARE PURELY RANDOM
 (0.01 TO 0.40,  ZERO=0.10)
 FREE FORMAT
 !
 0
 CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM IS  0.10

 GIVE 0 FOR RATE DATA INPUT
      1 FOR E/SIG E INPUT (OUTPUT OF TGRUNF)
          WITH DEGREES OF CONVERSION IN FILE
      2 FOR E/SIG E INPUT WITH D.O.C. TO BE CALCULATED
 !
 1
 FILE NUMBER  1
 !
 !
 !
 !
 !
 !
 !
 !
 !
```

Because of (i) the complexity of the TGDEPG program itself, (ii) the fact that it is allowed to edit data files autonomously manner, and (iii) the controversy surrounding the procedure, several examples of output have been provided. The reader will have to decide for himself whether the results are acceptable or not. Comparison of the results from the programs TGTRIM and TGDEPG is undoubtedly worthwhile. TGTRIM calculates trimmed means using the magnitudes of the input values in the rejection test; TGDEPG also uses the standard deviations of the input values.

A TGDEPG run in which a coherent subset of activation energies was selected from an input set containing several widely divergent values.

****

TITLE OF INPUT FILE IS :

NBS 706 IN 50% O2 1/12/78 NEW FURNACE PRESS=800

****

CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM IS 15.04

PUT PROCESSED DATA ON FILE 8

NUMBER	E	SIG E	D.O.C.	SWT
1	25.05	0.79	2.56	482964.00
2	23.40	0.38	3.14	479109.00
3	23.28	0.37	9.56	436394.00
4	21.56	0.43	12.21	418809.00
5	21.29	0.30	14.04	406647.00
6	20.38	0.35	16.56	389892.00
7	21.51	0.57	18.34	378025.00
8	17.77	0.85	20.78	361836.00
9	18.68	0.42	22.59	349784.00
10	23.03	0.28	25.10	333058.00
11	21.67	0.28	26.92	320969.00
12	21.28	0.35	29.51	303777.00
13	21.42	0.20	31.43	290970.00
14	19.54	0.25	34.19	272644.00
15	21.96	0.21	36.21	259205.00
16	20.58	0.32	39.10	240006.00
17	21.83	0.28	41.23	225799.00
18	21.78	0.29	44.31	205316.00
19	21.70	0.24	46.59	190178.00
20	21.56	0.66	49.87	168359.00
21	22.32	0.55	52.25	152528.00
22	21.51	0.26	55.71	129496.00
23	23.97	0.21	58.22	112843.00
24	21.60	0.29	61.76	89302.31
25	21.96	0.31	64.29	72444.38
26	24.92	0.34	67.85	48830.30
27	21.77	0.21	70.33	32330.10
28	22.45	0.27	73.65	10258.70
29	23.01	0.26	75.86	-4499.84
30	21.62	0.40	78.67	-23158.90
31	21.97	0.30	80.44	-34945.20
32	24.47	0.39	82.50	-48610.70
33	23.29	0.26	83.73	-56834.90
34	20.79	0.79	85.32	-67363.69
35	99.73	31.67	86.13	-72753.69
36	43.29	2.16	86.56	-75655.38
37	33.30	1.79	86.90	-77861.00

PASS NUMBER 1

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 22.028
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 1.304

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.164) + 0.737(0.179)*Z
REDUCED CHI SQUARE = 0.317 CALCULATED OVER 37 POINTS

RESCALE DS VALUES BY FACTOR OF 0.737
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.616)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
32	82.5	5.85	2.14	24.471	0.390	-48611.			

1 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
36 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	-0.12	0.41	25.052	0.787	482964.			
2	3.1	-1.02	-2.14	23.400	0.385	479109.			
3	9.6	-0.93	-1.30	23.278	0.372	436394.			
4	12.2	-0.80	-1.16	21.559	0.425	418809.			
5	14.0	-0.71	-1.03	21.292	0.295	405647.			
6	16.6	-0.66	-0.82	20.378	0.345	389892.			
7	18.3	-0.58	-0.57	21.510	0.573	378025.			
8	20.8	-0.56	-0.41	17.773	0.849	361636.			
9	22.6	-0.51	-0.27	18.678	0.423	349784.			
10	25.1	-0.39	0.00	23.031	0.277	333658.			
11	26.9	-0.28	0.13	21.675	0.277	320969.			
12	29.5	-0.14	0.34	21.276	0.345	303777.			
13	31.4	-0.07	0.49	21.422	0.200	290970.			
14	34.2	-0.00	0.65	19.549	0.252	272644.			
15	36.2	0.34	0.73	21.962	0.215	259205.			
16	39.1	0.59	0.92	20.576	0.322	240006.			
17	41.2	0.83	1.03	21.831	0.282	225799.			
18	44.3	1.95	1.48	21.784	0.292	205316.			
19	46.6	-0.94	-1.48	21.697	0.236	190178.			
20	49.9	0.40	0.82	21.565	0.659	168359.			
21	52.3	2.20	1.72	22.321	0.545	152528.			
22	55.7	-0.95	-1.72	21.514	0.261	129496.			

23	58.2	-0.50	-0.20	23.971	0.213	112843.
24	61.8	-0.45	-0.07	21.605	0.288	89302.
25	64.3	0.84	1.16	21.964	0.305	72444.
26	67.8	-0.67	-0.92	24.919	0.341	48830.
27	70.3	-0.01	0.57	21.775	0.211	32330.
28	73.6	-0.53	-0.34	22.452	0.269	10259.
29	75.9	-0.62	-0.65	23.012	0.262	-4500.
30	78.7	-0.43	-0.13	21.619	0.405	-23159.
31	80.4	-0.30	0.07	21.974	0.297	-34945.
33	83.5	0.94	1.30	23.288	0.259	-48611.
34	85.7	-0.18	0.27	20.793	0.789	-56835.
35	86.3	-0.58	-0.49	99.726	31.669	-67364.
36	86.1	-0.66	-0.73	43.292	2.160	-72754.
37	86.6	-0.28	0.20	33.304	1.787	-75655.

PASS NUMBER 2

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.983
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 1.330

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.167) + 1.016(0.173)*Z
REDUCED CHI SQUARE = 0.598 CALCULATED OVER 36 POINTS

RESCALE DS VALUES BY FACTOR OF 0.749
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.607)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
35	86.3	5.75	2.13	99.726	31.669	-67364.			

1 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
35 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	0.82	1.01	25.052	0.787	482964.			
2	3.1	-0.14	0.39	23.400	0.385	479109.			
3	9.6	-0.18	0.32	23.278	0.372	436394.			
4	12.2	-0.01	0.63	21.559	0.425	418809.			
5	14.0	-0.49	-0.17	21.292	0.295	406647.			
6	16.6	-0.28	0.17	20.378	0.345	389892.			

7	18.3	0.39	0.80	21.510	0.573	378025.
8	20.8	0.92	1.28	17.773	0.849	361836.
9	22.6	-0.01	0.54	18.678	0.423	349784.
10	25.1	-0.58	-0.54	23.031	0.277	333058.
11	26.9	-0.58	-0.46	21.675	0.277	320969.
12	29.5	-0.28	0.24	21.276	0.345	303777.
13	31.4	-1.01	-2.13	21.422	0.200	290970.
14	34.2	-0.70	-1.01	19.540	0.252	272644.
15	36.2	-0.91	-1.28	21.962	0.215	259205.
16	39.1	-0.38	0.03	20.576	0.322	240006.
17	41.2	-0.55	-0.39	21.831	0.282	225799.
18	44.3	-0.51	-0.24	21.784	0.292	205316.
19	46.6	-0.79	-1.14	21.697	0.236	190178.
20	49.9	0.58	0.90	21.565	0.659	168359.
21	52.3	0.33	0.71	22.321	0.545	152528.
22	55.7	-0.66	-0.80	21.514	0.261	129496.
23	58.2	-0.93	-1.46	23.971	0.213	112843.
24	61.8	-0.52	-0.32	21.605	0.288	89302.
25	64.3	-0.45	-0.03	21.964	0.305	72444.
26	67.8	-0.30	0.10	24.919	0.341	48830.
27	70.3	-0.94	-1.70	21.775	0.211	32330.
28	73.6	-0.62	-0.63	22.452	0.269	10259.
29	75.9	-0.65	-0.71	23.012	0.252	-4500.
30	78.7	-0.07	0.46	21.619	0.405	-23159.
31	80.4	-0.48	-0.10	21.974	0.297	-34945.
33	83.5	-0.67	-0.90	23.288	0.259	-48611.
34	85.7	0.82	1.14	20.793	0.739	-56835.
36	86.3	2.17	1.70	43.292	2.160	-67364.
37	86.1	1.91	1.46	33.304	1.787	-72754.

PASS NUMBER 3

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.983
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.463

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.169) + 0.699(0.175)*Z
REDUCED CHI SQUARE = 0.100 CALCULATED OVER 35 POINTS

RESCALE DS VALUES BY FACTOR OF 0.524
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.598)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
36	86.3	3.33	2.12	43.292	2.160	-67364.			
37	86.3	2.97	1.69	33.304	1.787	-67364.			

2 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION

33 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	1.40	1.12	25.052	0.787	482964.			
2	3.1	0.04	0.44	23.400	0.385	479109.			
3	9.6	-0.03	0.36	23.278	0.372	436394.			
4	12.2	0.23	0.69	21.559	0.425	418809.			
5	14.0	-0.47	-0.14	21.292	0.295	406647.			
6	16.6	-0.17	0.21	20.378	0.345	369692.			
7	18.3	0.80	0.88	21.510	0.573	378025.			
8	20.8	1.55	1.45	17.773	0.849	361836.			
9	22.6	0.22	0.60	18.678	0.423	349784.			
10	25.1	-0.59	-0.52	23.031	0.277	333858.			
11	26.9	-0.59	-0.44	21.675	0.277	320969.			
12	29.5	-0.17	0.29	21.276	0.345	303777.			
13	31.4	-1.21	-2.12	21.422	0.200	290970.			
14	34.2	-0.77	-1.00	19.540	0.252	272644.			
15	36.2	-1.07	-1.27	21.962	0.215	259205.			
16	39.1	-0.30	0.07	20.576	0.322	240006.			
17	41.2	-0.56	-0.36	21.831	0.282	225799.			
18	44.3	-0.49	-0.21	21.724	0.292	205316.			
19	46.6	-0.90	-1.12	21.697	0.236	190178.			
20	49.9	1.06	1.00	21.565	0.659	168359.			
21	52.3	0.70	0.78	22.321	0.545	152528.			
22	55.7	-0.70	-0.78	21.514	0.261	129496.			
23	58.2	-1.09	-1.45	23.971	0.213	112843.			
24	61.8	-0.52	-0.29	21.605	0.288	89302.			
25	64.3	-0.41	0.00	21.964	0.305	72444.			
26	67.8	-0.19	0.14	24.919	0.341	48830.			
27	70.3	-1.11	-1.69	21.775	0.211	32330.			
28	73.6	-0.65	-0.60	22.452	0.269	10259.			
29	75.9	-0.70	-0.69	23.012	0.262	-4500.			
30	78.7	0.14	0.52	21.619	0.405	-23159.			
31	80.4	-0.46	-0.07	21.974	0.297	-34945.			
33	83.5	-0.72	-0.88	23.288	0.259	-48611.			
34	85.7	1.41	1.27	20.793	0.789	-56835.			

PASS NUMBER 4

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.960
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.371

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.174) + 0.733(0.181)*Z

REDUCED CHI SQUARE = 0.056 CALCULATED OVER 33 POINTS

RESCALE DS VALUES BY FACTOR OF 0.384
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.579)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
33 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	2.17	1.42	25.052	0.787	492964.			
2	3.1	0.31	0.55	23.400	0.385	479109.			
3	9.6	0.22	0.47	23.278	0.372	436394.			
4	12.2	0.57	0.84	21.559	0.425	418809.			
5	14.0	-0.38	-0.08	21.292	0.295	406647.			
6	16.6	0.03	0.31	20.378	0.345	389892.			
7	18.3	1.35	1.08	21.510	0.573	378025.			
8	20.8	2.37	2.09	17.773	0.849	361836.			
9	22.6	0.56	0.74	18.679	0.423	349784.			
10	25.1	-0.54	-0.47	23.031	0.277	333058.			
11	26.9	-0.54	-0.39	21.675	0.277	320969.			
12	29.5	0.03	0.39	21.276	0.345	303777.			
13	31.4	-1.39	-2.09	21.422	0.200	290970.			
14	34.2	-0.79	-0.96	19.540	0.252	272644.			
15	36.2	-1.20	-1.23	21.962	0.215	259205.			
16	39.1	-0.15	0.15	20.576	0.322	240006.			
17	41.2	-0.50	-0.31	21.831	0.282	225799.			
18	44.3	-0.41	-0.15	21.784	0.292	205316.			
19	46.6	-0.96	-1.08	21.697	0.236	190178.			
20	49.9	1.71	1.23	21.565	0.659	168359.			
21	52.3	1.22	0.96	22.321	0.545	152528.			
22	55.7	-0.70	-0.74	21.514	0.261	129496.			
23	58.2	-1.23	-1.42	23.971	0.213	112843.			
24	61.8	-0.44	-0.23	21.605	0.283	89302.			
25	64.3	-0.29	0.08	21.964	0.305	72444.			
26	67.8	-0.00	0.23	24.919	0.341	48830.			
27	70.3	-1.25	-1.66	21.775	0.211	32330.			
28	73.6	-0.62	-0.55	22.452	0.269	10259.			
29	75.9	-0.69	-0.64	23.012	0.262	-4500.			
30	78.7	0.45	0.64	21.619	0.405	-23159.			
31	80.4	-0.36	0.00	21.974	0.297	-34945.			
33	83.5	-0.72	-0.84	23.289	0.259	-48611.			
34	85.7	2.18	1.66	20.793	0.789	-56835.			

PASS NUMBER 5

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.960

UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.371
EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC
FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 1.754

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 1.754

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.579)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
6	16.6	-2.61	-1.23	20.378	0.605	389892.			
8	20.3	-2.81	-1.42	17.773	1.439	378025.			
9	22.3	-4.42	-1.66	18.678	0.742	378025.			
14	34.9	-5.47	-2.09	19.540	0.442	320969.			
23	58.6	5.38	2.09	23.971	0.374	190178.			
26	67.3	4.95	1.66	24.919	0.598	152528.			
33	83.8	2.92	1.42	23.288	0.454	48830.			

7 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
26 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	2.24	1.08	25.052	1.381	482964.			
2	3.1	2.13	0.84	23.400	0.675	479109.			
3	9.6	2.02	0.74	23.278	0.653	436394.			
4	12.2	-0.54	-0.23	21.559	0.746	418809.			
5	14.0	-1.29	-0.84	21.292	0.518	406547.			
7	18.6	-0.45	0.00	21.510	1.005	389892.			
10	25.3	2.20	0.96	23.031	0.486	378025.			
11	26.8	-0.59	-0.31	21.675	0.486	361836.			
12	29.6	-1.13	-0.74	21.276	0.605	349784.			
13	31.1	-1.53	-0.96	21.422	0.351	333858.			
15	36.9	0.01	0.31	21.962	0.377	320969.			
16	39.5	-2.45	-1.08	20.576	0.565	303777.			
17	41.4	-0.26	0.23	21.831	0.495	290970.			
18	44.2	-0.34	0.08	21.784	0.512	272644.			
19	46.2	-0.64	-0.39	21.697	0.414	259205.			
20	49.1	-0.34	0.15	21.565	1.156	240006.			
21	52.2	0.38	0.55	22.321	0.956	225799.			
22	55.3	-0.97	-0.64	21.514	0.458	205316.			
24	61.6	-0.70	-0.47	21.605	0.505	190178.			
25	64.9	0.01	0.39	21.964	0.535	168359.			
27	70.2	-0.50	-0.15	21.775	0.370	152528.			

28	73.7	1.04	0.64	22.452	0.472	129496.
29	75.2	2.29	1.23	23.012	0.460	112843.
30	78.8	-0.48	-0.08	21.619	0.711	89302.
31	80.3	0.03	0.47	21.974	0.521	72444.
34	85.8	-0.84	-0.55	20.793	1.384	48830.

PASS NUMBER 6

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.910
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.646
EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC
FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 0.944

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 1.656

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.500)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
16	39.5	-2.50	-2.00	20.576	0.533	303777.			
29	75.7	2.54	2.00	23.012	0.434	129496.			

2 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
24 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	2.41	1.28	25.052	1.303	482964.			
2	3.1	2.34	1.09	23.400	0.638	479109.			
3	9.6	2.22	0.93	23.278	0.615	436394.			
4	12.2	-0.50	-0.34	21.559	0.704	418809.			
5	14.0	-1.27	-1.28	21.292	0.488	406647.			
7	18.6	-0.42	-0.14	21.510	0.949	389892.			
10	25.3	2.44	1.55	23.031	0.459	378025.			
11	26.8	-0.51	-0.44	21.675	0.459	361836.			
12	29.6	-1.11	-1.09	21.276	0.571	349784.			
13	31.1	-1.47	-1.55	21.422	0.331	333058.			
15	36.9	0.15	0.55	21.962	0.356	320969.			
17	41.5	-0.17	0.24	21.831	0.467	303777.			
18	44.4	-0.26	0.14	21.784	0.484	290970.			
19	46.2	-0.55	-0.55	21.697	0.391	272644.			
20	49.2	-0.32	0.05	21.565	1.091	259205.			
21	52.1	0.46	0.67	22.321	0.902	240006.			

22	55.2	-0.92	-0.93	21.514	0.432	225799.
24	61.3	-0.64	-0.67	21.605	0.477	205316.
25	64.6	0.11	0.34	21.964	0.505	190178.
27	70.9	-0.39	-0.05	21.775	0.349	168359.
28	73.2	1.22	0.79	22.452	0.445	152528.
30	78.7	-0.43	-0.24	21.619	0.671	129496.
31	80.2	0.13	0.44	21.974	0.492	112843.
34	85.8	-0.86	-0.79	20.793	1.307	89302.

PASS NUMBER 7

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.898
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.620
EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC
FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 0.850

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 1.407

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.472)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	2.6	2.85	1.50	25.052	1.107	482964.			
2	3.2	2.77	1.24	23.400	0.542	482964.			
3	9.6	2.64	1.04	23.278	0.523	482964.			
10	25.4	2.91	1.96	23.031	0.390	418809.			

4 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
20 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
4	12.2	-0.57	-0.37	21.559	0.598	482964.			
5	14.0	-1.46	-1.50	21.292	0.415	479109.			
7	19.0	-0.48	-0.16	21.510	0.806	436394.			
11	26.4	-0.57	-0.49	21.675	0.390	418809.			
12	29.6	-1.28	-1.24	21.276	0.485	406647.			
13	31.6	-1.69	-1.96	21.422	0.281	389892.			
15	36.3	0.21	0.60	21.962	0.302	378025.			
17	41.8	-0.17	0.26	21.831	0.397	361836.			
18	44.6	-0.28	0.16	21.784	0.411	349784.			
19	46.1	-0.61	-0.60	21.697	0.332	333058.			
20	49.9	-0.36	0.05	21.565	0.927	320969.			

21	52.5	0.55	0.73	22.321	0.767	303777.
22	55.4	-1.05	-1.04	21.514	0.367	290970.
24	61.2	-0.72	-0.73	21.605	0.405	272644.
25	64.2	0.15	0.37	21.964	0.429	259205.
27	70.1	-0.41	-0.05	21.775	0.297	240006.
28	73.2	1.46	0.97	22.452	0.378	225799.
30	78.3	-0.49	-0.26	21.619	0.570	205316.
31	80.6	0.18	0.48	21.974	0.418	190178.
34	85.9	-1.00	-0.87	20.793	1.110	168359.

PASS NUMBER 8

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.723
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.504

EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC

FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 0.661

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 0.930

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.409)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
28	73.2	2.91	1.88	22.452	0.250	225799.			

1 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
19 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
4	12.2	-0.42	-0.45	21.559	0.395	482964.			
5	14.0	-1.57	-1.41	21.292	0.274	479109.			
7	19.0	-0.40	-0.31	21.510	0.533	436394.			
11	26.4	-0.19	0.06	21.675	0.258	418809.			
12	29.6	-1.39	-1.13	21.276	0.321	406647.			
13	31.6	-1.62	-1.88	21.422	0.186	389392.			
15	36.3	1.19	1.41	21.962	0.200	378225.			
17	41.8	0.41	0.59	21.831	0.262	361836.			
18	44.6	0.22	0.31	21.784	0.272	349784.			
19	46.1	-0.12	0.19	21.697	0.220	333058.			
20	49.9	-0.26	-0.06	21.565	0.613	320969.			
21	52.5	1.18	1.13	22.321	0.507	303777.			

22	55.4	-0.86	-0.74	21.514	0.243	290970.
24	61.2	-0.44	-0.59	21.605	0.268	272644.
25	64.2	0.85	0.74	21.964	0.284	259205.
27	70.1	0.26	0.45	21.775	0.196	240806.
30	78.2	-0.28	-0.19	21.619	0.377	225799.
31	80.3	0.91	0.92	21.974	0.276	205316.
34	85.6	-1.27	-0.92	20.793	0.734	190178.

PASS NUMBER 9

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.676
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.338
EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC
FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 0.925

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 0.861

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.391)

# DOC/TIME		DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
19 DATA LEFT

OUTPUT REMAINING DATA

# DOC/TIME		DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
4	12.2	-0.32	-0.40	21.559	0.366	482964.			
5	14.0	-1.51	-1.86	21.292	0.254	479109.			
7	19.0	-0.34	-0.55	21.510	0.493	436394.			
11	26.4	-0.01	0.13	21.675	0.238	418809.			
12	29.6	-1.35	-1.10	21.276	0.297	406547.			
13	31.6	-1.48	-1.38	21.422	0.172	389892.			
15	36.3	1.54	1.86	21.962	0.185	378025.			
17	41.8	0.64	0.70	21.831	0.243	361836.			
18	44.6	0.43	0.40	21.784	0.251	349784.			
19	46.1	0.10	0.26	21.697	0.203	333858.			
20	49.9	-0.20	-0.13	21.565	0.567	320969.			
21	52.5	1.37	1.38	22.321	0.469	303777.			
22	55.4	-0.72	-0.70	21.514	0.225	290970.			
24	61.2	-0.29	-0.26	21.605	0.248	272644.			
25	64.2	1.09	0.88	21.964	0.263	259205.			
27	70.1	0.54	0.55	21.775	0.182	240806.			
30	78.2	-0.16	-0.00	21.619	0.349	225799.			

```
31      80.3  1.16  1.10 21.974  0.256   205316.
34      85.6 -1.30 -0.88 20.793  0.679   190178.
```

WEIGHTED MEAN E IS 21.676 + OR -   0.059
REDUCED CHISQ FOR E CALCULATION IS 0.921 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS   0.313
CALCULATED OVER   18. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 21.561 TO 21.792
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 21.639 AND 0.075
THIS GIVES A CONFIDENCE RANGE OF 21.492 TO 21.786
CHI-SQUARE FOR UNWEIGHTED E IS   0.94

19 DATA WRITTEN ON FILE =   8 =

DE = -0.063+ 1.169*Z + 0.009*Z **2 + -0.090*Z **3
REDUCED CHI SQUARE IS 0.016 CALCULATED OVER 19 POINTS

3 CURVES GENERATED FOR DE/Z PLOTS
Z IS THE HORIZONTAL AXIS
DE IS THE FIRST CURVE SECOND CURVE IS LINEAR FIT OF DE TO Z
THIRD CURVE IS POLYNOMIAL FIT OF DE TO Z

```
  0.17406E+01  -                                                  B
               +
               +                                                  C
               +                                                  A
               +
               -                                            A
               +                                            B
               +
               +                                        A
               +                                     A  C
  0.10215E+01  -                                        B
               +                                     C
               +
               +                                     B
               +                                  C
               -                                  A
               +                                C B
               +                                A
               +                              A B
               +                              C
  0.30228E+00  -                              B
               +                             B
               +
               +                           B A
               +                           A
               -                         B
               +                         A
               +                       A
```

```
             +                     A
             +                  A AB
-0.41691E+00 -                    B
             +                    C
             +                  B
             +
             +                A C
             -
             +                C
             +             B
             +             C
             +
-0.11361E+01 -          B
             +          C
             +             A
             +          A
             +      B
             -A     A
             +
             +C
             +
             +
-0.18553E+01 -B
              +---------+---------+---------+---------+---------+
         0.1861E+010.1116E+010.3721E+000.3721E+000.1116E+010.1861E+01
```

A = DE V. Z
B = LINEAR FIT OF DE TO Z
C = POLYNOMIAL FIT OF DE TO Z
JOB COMPLETED

The saved file from the previous example.

NBS 706 IN 50% O2 1/12/78 NEW FURNACE PRESS=800

21.676	0.059	21.639	0.075	19
4	21.56	0.37	12.25	482964.00
5	21.29	0.25	14.01	479109.00
7	21.51	0.49	18.96	436394.00
11	21.67	0.24	26.36	418809.00
12	21.28	0.30	29.64	406647.00
13	21.42	0.17	31.56	389892.00
15	21.96	0.19	36.34	378025.00
17	21.83	0.24	41.78	361836.00
18	21.78	0.25	44.59	349784.00
19	21.70	0.20	46.10	333058.00
20	21.56	0.57	49.92	320969.00
21	22.32	0.47	52.51	303777.00
22	21.51	0.22	55.43	290970.00
24	21.60	0.25	61.19	272644.00
25	21.96	0.26	64.21	259205.00
27	21.77	0.18	70.10	240006.00
30	21.62	0.35	78.23	225799.00
31	21.97	0.26	80.31	205316.00
34	20.79	0.68	85.59	190178.00

An example where all rejections made were based on log $\hat{\sigma}_{Ei}$. Input $\hat{\sigma}_{Ei}$ values were rescaled.

```
*****************************************************************************
 TITLE OF INPUT FILE IS :

 NBS 706 IN 50% O2 1/24/78 RUN 3
P =800 MM HG

*****************************************************************************
```

CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM IS 15.06

PUT PROCESSED DATA ON FILE 8

NUMBER	E	SIG E	D.O.C.	SWT
1	21.49	0.32	5.08	466217.00
2	20.14	0.44	7.43	450583.00
3	20.90	0.31	9.19	438865.00
4	21.80	0.78	12.39	417580.00
5	20.74	0.64	14.47	403770.03
6	21.49	0.64	17.28	385060.00
7	19.24	0.45	19.29	371724.00
8	20.42	0.47	22.04	353466.00
9	20.50	0.43	24.02	340260.00
10	20.04	0.55	26.74	322175.00
11	20.72	0.36	28.75	308846.00
12	20.96	0.30	31.53	290307.00
13	19.53	0.33	33.61	276491.00
14	21.28	0.44	36.51	257211.00
15	22.02	0.31	38.67	242839.00
16	20.85	0.34	41.75	222349.00
17	22.43	0.30	44.06	207012.00
18	19.41	0.42	47.33	185242.00
19	26.21	2.48	50.44	164542.00
20	18.96	1.82	53.22	146055.00
21	23.07	0.44	55.78	129088.00
22	21.81	0.55	59.39	105082.00
23	21.81	0.42	62.06	87318.12
24	22.94	0.38	65.79	62529.70
25	21.99	0.28	68.48	44597.30
26	22.38	0.35	72.14	20277.10
27	21.87	0.39	74.69	3320.86
28	21.80	0.66	78.01	-18745.50
29	21.98	0.50	80.19	-33284.80
30	21.62	0.46	82.77	-50393.60
31	21.75	0.31	84.38	-61126.20
32	17.53	0.99	86.34	-74163.00
33	16.27	5.32	87.61	-82585.12
34	40.66	27.15	87.72	-83367.88

PASS NUMBER 1

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.316

UNWEIGHTED AVERAGE OF SIGMA VALUES IS 1.480

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.171) + 0.765(0.178)*Z
REDUCED CHI SQUARE = 0.317 CALCULATED OVER 34 POINTS

RESCALE DS VALUES BY FACTOR OF 0.765
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.589)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
33	87.6	2.91	1.68	16.269	5.323	-82585.			
34	87.6	5.04	2.11	40.664	27.149	-82585.			

2 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
32 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.1	-0.75	-0.86	21.491	0.323	466217.			
2	7.4	-0.34	0.04	20.140	0.444	450583.			
3	9.2	-0.79	-0.98	20.900	0.313	438865.			
4	12.4	0.40	0.98	21.805	0.780	417580.			
5	14.5	0.13	0.67	20.739	0.636	403770.			
6	17.3	0.13	0.76	21.489	0.636	385060.			
7	19.3	-0.30	0.18	19.243	0.455	371724.			
8	22.0	-0.26	0.33	20.425	0.470	353466.			
9	24.0	-0.38	-0.11	20.503	0.428	340260.			
10	26.7	-0.06	0.49	20.043	0.547	322175.			
11	28.8	-0.61	-0.49	20.720	0.359	308846.			
12	31.5	-0.85	-1.43	20.958	0.300	290307.			
13	33.6	-0.74	-0.76	19.532	0.326	276491.			
14	36.5	-0.33	0.11	21.280	0.445	257211.			
15	38.7	-0.80	-1.10	22.023	0.312	242839.			
16	41.8	-0.67	-0.67	20.854	0.343	222349.			
17	44.1	-0.87	-1.68	22.429	0.295	207012.			
18	47.3	-0.42	-0.26	19.410	0.418	185242.			
19	50.4	1.91	1.43	26.212	2.478	164542.			
20	53.2	1.51	1.25	18.959	1.823	146055.			
21	55.8	-0.35	-0.04	23.068	0.438	129088.			
22	59.4	-0.05	0.58	21.809	0.553	105082.			
23	62.1	-0.41	-0.18	21.809	0.420	87318.			
24	65.8	-0.55	-0.41	22.943	0.376	62530.			

25	68.5	-0.95	-2.11	21.992	0.277	44597.
26	72.1	-0.63	-0.58	22.385	0.354	20277.
27	74.7	-0.49	-0.33	21.969	0.395	3321.
28	78.0	0.17	0.86	21.800	0.656	-18746.
29	80.2	-0.19	0.41	21.977	0.495	-33285.
30	82.8	-0.28	0.26	21.618	0.464	-50394.
31	84.4	-0.82	-1.25	21.746	0.306	-61126.
32	86.3	0.71	1.10	17.533	0.987	-74163.

PASS NUMBER 2

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.317
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.558

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.177) + 0.595(0.184)*Z
REDUCED CHI SQUARE = 0.085 CALCULATED OVER 32 POINTS

RESCALE DS VALUES BY FACTOR OF 0.455
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.569)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
19	50.4	3.63	2.08	26.212	2.478	164542.			
20	53.4	2.95	1.65	18.959	1.823	164542.			

2 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
30 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.1	-0.85	-0.82	21.491	0.323	466217.			
2	7.4	-0.15	0.12	20.140	0.444	450583.			
3	9.2	-0.91	-0.94	20.900	0.313	433865.			
4	12.4	1.09	1.22	21.805	0.780	417580.			
5	14.5	0.64	0.82	20.739	0.636	403770.			
6	17.3	0.64	0.94	21.439	0.636	385060.			
7	19.3	-0.09	0.27	19.243	0.455	371724.			
8	22.0	-0.02	0.44	20.425	0.470	353466.			
9	24.0	-0.23	-0.04	20.503	0.428	340260.			
10	26.7	0.31	0.62	20.043	0.547	322175.			
11	28.8	-0.61	-0.44	20.720	0.359	308846.			

12	31.5	-1.01	-1.40	20.958	0.300	290307.
13	33.6	-0.83	-0.72	19.532	0.326	276491.
14	36.5	-0.14	0.19	21.280	0.445	257211.
15	38.7	-0.92	-1.07	22.023	0.312	242839.
16	41.8	-0.71	-0.62	20.854	0.343	222349.
17	44.1	-1.05	-1.65	22.429	0.295	207012.
18	47.3	-0.28	-0.19	19.410	0.418	185242.
21	55.4	-0.18	0.04	23.068	0.438	154542.
22	59.2	0.33	0.72	21.209	0.553	146055.
23	62.8	-0.27	-0.12	21.809	0.420	129088.
24	65.4	-0.51	-0.36	22.943	0.376	105082.
25	68.1	-1.18	-2.08	21.992	0.277	87318.
26	72.8	-0.64	-0.53	22.385	0.354	62530.
27	74.5	-0.40	-0.27	21.869	0.395	44597.
28	78.1	0.71	1.07	21.800	0.656	20277.
29	80.7	0.09	0.53	21.977	0.495	3321.
30	82.0	-0.05	0.35	21.618	0.464	-18746.
31	84.2	-0.96	-1.22	21.746	0.306	-33285.
32	86.8	1.61	1.40	17.533	0.987	-50394.

PASS NUMBER 3

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.316
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.452

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = -0.000(0.183) + 0.677(0.190)*Z
REDUCED CHI SQUARE = 0.027 CALCULATED OVER 30 POINTS

RESCALE DS VALUES BY FACTOR OF 0.308
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.547)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
32	86.8	2.70	2.05	17.533	0.987	-50394.			

1 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
29 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.1	-0.93	-0.77	21.491	0.323	466217.			
2	7.4	0.11	0.21	20.140	0.444	450583.			
3	9.2	-1.03	-0.89	20.900	0.313	439865.			

4	12.4	1.93	1.62	21.805	0.780	417580.
5	14.5	1.27	1.02	20.739	0.636	403770.
6	17.3	1.27	1.18	21.489	0.636	385060.
7	19.3	0.19	0.38	19.243	0.455	371724.
8	22.0	0.29	0.57	20.425	0.470	353466.
9	24.0	-0.01	0.04	20.503	0.428	340260.
10	26.7	0.78	0.77	20.043	0.547	322175.
11	28.8	-0.58	-0.38	20.720	0.359	308846.
12	31.5	-1.17	-1.36	20.958	0.300	290307.
13	33.6	-0.90	-0.67	19.532	0.326	276491.
14	36.5	0.11	0.29	21.280	0.445	257211.
15	38.7	-1.04	-1.02	22.023	0.312	242839.
16	41.8	-0.73	-0.57	20.854	0.343	222349.
17	44.1	-1.22	-1.62	22.429	0.295	207012.
18	47.3	-0.09	-0.12	19.410	0.418	185242.
21	55.4	0.06	0.12	23.068	0.438	164542.
22	59.2	0.82	0.89	21.809	0.553	146855.
23	62.8	-0.07	-0.04	21.809	0.420	129898.
24	65.4	-0.43	-0.29	22.943	0.376	105082.
25	68.1	-1.42	-2.05	21.992	0.277	87318.
26	72.8	-0.63	-0.47	22.385	0.354	62530.
27	74.5	-0.27	-0.21	21.869	0.395	44597.
28	78.1	1.37	1.36	21.800	0.656	20277.
29	80.7	0.46	0.67	21.977	0.495	3321.
30	82.0	0.25	0.47	21.618	0.464	-18746.
31	84.2	-1.10	-1.18	21.746	0.306	-33285.

PASS NUMBER 4

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.337
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.433

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.186) + 0.886(0.194)*Z
REDUCED CHI SQUARE = 0.031 CALCULATED OVER 29 POINTS

RESCALE DS VALUES BY FACTOR OF 0.273
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.536)

# DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
29 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.1	-0.94	-0.75	21.491	0.323	466217.			
2	7.4	0.23	0.26	20.140	0.444	450583.			
3	9.2	-1.05	-0.87	20.900	0.313	432865.			
4	12.4	2.29	2.04	21.805	0.780	417580.			
5	14.5	1.54	1.16	20.739	0.635	403770.			
6	17.3	1.54	1.35	21.489	0.636	385060.			
7	19.3	0.31	0.44	19.243	0.455	371724.			
8	22.0	0.43	0.64	20.425	0.470	353466.			
9	24.0	0.09	0.09	20.503	0.428	340260.			
10	26.7	0.99	0.87	20.043	0.547	322175.			
11	28.8	-0.55	-0.35	20.720	0.359	302846.			
12	31.5	-1.21	-1.35	20.958	0.300	290307.			
13	33.6	-0.91	-0.64	19.532	0.326	276491.			
14	36.5	0.23	0.35	21.280	0.445	257211.			
15	38.7	-1.07	-1.00	22.023	0.312	242839.			
16	41.8	-0.72	-0.54	20.854	0.343	222349.			
17	44.1	-1.27	-1.60	22.429	0.295	207812.			
18	47.3	0.00	-0.09	19.410	0.418	185242.			
21	55.4	0.18	0.17	23.068	0.438	164542.			
22	59.2	1.03	1.00	21.889	0.553	146055.			
23	62.8	0.02	-0.09	21.809	0.420	129088.			
24	65.4	-0.38	-0.26	22.943	0.376	105082.			
25	68.1	-1.50	-2.04	21.992	0.277	87318.			
26	72.8	-0.60	-0.44	22.385	0.354	62530.			
27	74.5	-0.20	-0.17	21.869	0.395	44597.			
28	78.1	1.65	1.60	21.800	0.656	20277.			
29	80.7	0.62	0.75	21.977	0.493	3321.			
30	82.0	0.39	0.54	21.618	0.464	-18746.			
31	84.2	-1.14	-1.16	21.746	0.306	-33285.			

PASS NUMBER 5

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 21.337
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.433

EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC

FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 2.365

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 2.365

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.536)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
29 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.1	0.20	-0.00	21.491	0.764	466217.			
2	7.4	-1.14	-1.16	20.140	1.050	450583.			
3	9.2	-0.59	-0.44	20.900	0.740	438865.			
4	12.4	0.25	0.09	21.805	1.844	417580.			
5	14.5	-0.40	-0.26	20.739	1.504	403770.			
6	17.3	0.10	-0.09	21.489	1.504	385060.			
7	19.3	-1.95	-1.35	19.243	1.076	371724.			
8	22.0	-0.82	-0.75	20.425	1.111	353466.			
9	24.0	-0.82	-0.87	20.503	1.012	340260.			
10	26.7	-1.00	-1.00	20.043	1.293	322175.			
11	28.8	-0.73	-0.64	20.720	0.849	308846.			
12	31.5	-0.53	-0.35	20.958	0.709	290307.			
13	33.6	-2.34	-2.04	19.532	0.771	276491.			
14	36.5	-0.05	-0.17	21.280	1.052	257211.			
15	38.7	0.93	0.87	22.023	0.738	242839.			
16	41.8	-0.60	-0.54	20.654	0.811	222349.			
17	44.1	1.57	1.35	22.429	0.698	207012.			
18	47.3	-1.95	-1.60	19.410	0.988	185242.			
21	55.4	1.67	1.60	23.068	1.036	164542.			
22	59.2	0.35	0.35	21.809	1.308	146055.			
23	62.8	0.48	0.44	21.809	0.993	129088.			
24	65.4	1.81	2.04	22.943	0.889	105082.			
25	68.1	1.00	1.00	21.992	0.655	87318.			
26	72.8	1.25	1.16	22.385	0.837	62530.			
27	74.5	0.57	0.75	21.869	0.934	44597.			
28	78.1	0.30	0.26	21.800	1.551	20277.			
29	80.7	0.55	0.54	21.977	1.170	3321.			
30	82.0	0.26	0.17	21.618	1.097	-18746.			
31	84.2	0.57	0.64	21.746	0.724	-33285.			

WEIGHTED MEAN E IS 21.337 + OR - 0.172
REDUCED CHISQ FOR E CALCULATION IS 1.140 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS 1.024
CALCULATED OVER 28. DEGREES OF FREEDOM

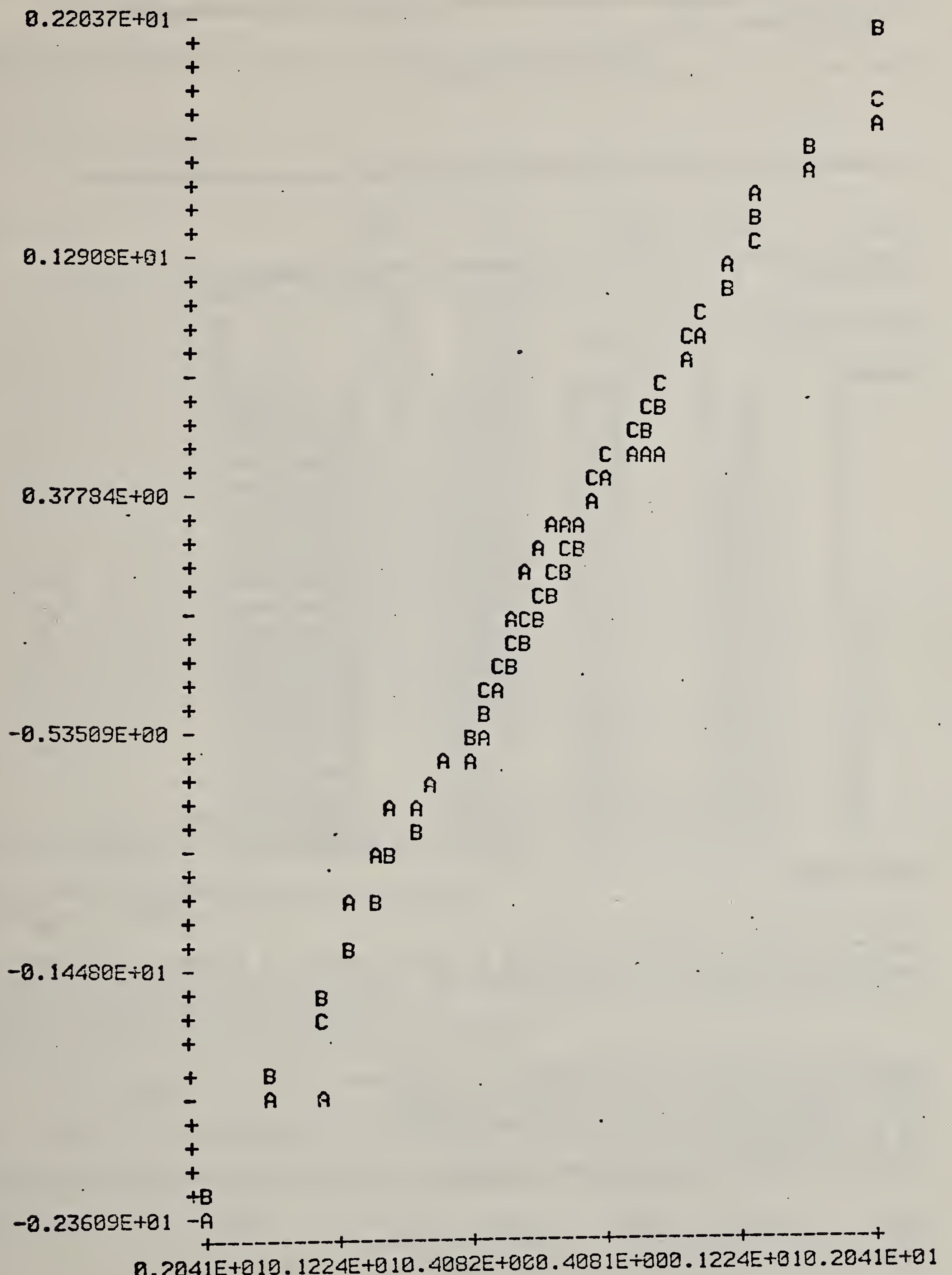
ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 21.000 TO 21.674
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 21.276 AND 0.185
THIS GIVES A CONFIDENCE RANGE OF 20.913 TO 21.639
CHI-SQUARE FOR UNWEIGHTED E IS 1.14

29 DATA WRITTEN ON FILE = 8 =

DE = 0.030+ 1.174*Z + -0.061*Z **2 + -0.030*Z **3
REDUCED CHI SQUARE IS 0.005 CALCULATED OVER 29 POINTS

3 CURVES GENERATED FOR DE/Z PLOTS
Z IS THE HORIZONTAL AXIS
DE IS THE FIRST CURVE SECOND CURVE IS LINEAR FIT OF DE TO Z
THIRD CURVE IS POLYNOMIAL FIT OF DE TO Z



A = DE V. Z
B = LINEAR FIT OF DE TO Z
C = POLYNOMIAL FIT OF DE TO Z
JOB COMPLETED

An example of an already coherent data set where no need for rejection was found.

```
***********************************************************************
TITLE OF INPUT FILE IS :

NBS 706 IN 40% O2 P=800 MM HG 3/20/78

***********************************************************************

CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM
IS 15.06

PUT PROCESSED DATA ON FILE  8
```

NUMBER	E	SIG E	D.O.C.	SWT
1	22.42	0.18	5.82	573756.00
2	23.19	0.14	8.19	554887.00
3	23.39	0.19	11.37	529632.00
4	23.16	0.18	13.49	512748.00
5	22.87	0.22	16.48	488986.00
6	22.93	0.17	18.55	472496.00
7	23.03	0.20	21.53	448867.00
8	23.09	0.17	23.64	432094.00
9	23.76	0.23	26.71	407669.00
10	23.55	0.20	28.38	390441.00
11	23.51	0.22	32.06	365137.00
12	23.83	0.19	34.31	347203.00
13	24.05	0.25	37.63	320866.00
14	23.00	0.20	39.96	302300.00
15	24.37	0.22	43.35	275378.00
16	24.12	0.14	45.73	256475.00
17	23.99	0.17	49.18	229012.00
18	24.20	0.14	51.59	209822.00
19	23.93	0.18	55.07	182190.00

```
PASS NUMBER  1

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 23.508
UNWEIGHTED AVERAGE OF SIGMA VALUES IS  0.188


EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
 DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
 K = SCALE FACTOR
 AND Z = NORMAL DISTRIBUTION QUANTILE


DS =  -0.000(  0.229) +    0.170(  0.244)*Z
REDUCED CHI SQUARE =  0.001 CALCULATED OVER    19 POINTS

RESCALE DS VALUES BY FACTOR OF    0.170
```

REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.391)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
19 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.8	-0.29	-0.26	22.422	0.177	573756.			
2	8.2	-1.70	-1.86	23.191	0.139	554887.			
3	11.4	0.16	-0.00	23.392	0.191	529632.			
4	13.5	-0.35	-0.40	23.157	0.175	512748.			
5	16.5	0.99	0.88	22.867	0.220	488986.			
6	18.5	-0.63	-0.70	22.935	0.167	472496.			
7	21.5	0.37	0.26	23.031	0.198	448867.			
8	23.6	-0.45	-0.55	23.086	0.172	432094.			
9	26.7	1.28	1.38	23.764	0.231	407669.			
10	28.9	0.43	0.40	23.552	0.200	390441.			
11	32.1	1.10	1.10	23.507	0.224	365137.			
12	34.3	0.16	0.13	23.831	0.191	347203.			
13	37.6	1.65	1.86	24.055	0.246	320866.			
14	40.0	0.52	0.55	22.997	0.203	302300.			
15	43.3	0.91	0.70	24.370	0.217	275378.			
16	45.7	-1.54	-1.10	24.122	0.143	256475.			
17	49.2	-0.66	-0.88	23.990	0.166	229012.			
18	51.6	-1.66	-1.38	24.200	0.140	209822.			
19	55.1	-0.29	-0.13	23.929	0.177	182190.			

PASS NUMBER 2

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 23.508
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 0.188

EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC

FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 3.844

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 3.844

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.391)

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
19 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	5.8	-1.60	-1.86	22.422	0.680	573756.			
2	8.2	-0.59	-0.40	23.191	0.534	554887.			
3	11.4	-0.16	-0.13	23.392	0.734	529632.			
4	13.5	-0.52	-0.26	23.157	0.673	512748.			
5	16.5	-0.76	-1.10	22.867	0.846	488986.			
6	18.5	-0.89	-1.38	22.935	0.642	472496.			
7	21.5	-0.63	-0.55	23.031	0.761	448867.			
8	23.6	-0.64	-0.70	23.086	0.661	432094.			
9	26.7	0.29	0.26	23.764	0.888	407669.			
10	28.9	0.06	0.13	23.552	0.769	390441.			
11	32.1	-0.00	-0.00	23.507	0.861	365137.			
12	34.3	0.44	0.49	23.831	0.734	347203.			
13	37.6	0.58	0.55	24.055	0.945	320866.			
14	40.0	-0.65	-0.88	22.997	0.780	302300.			
15	43.3	1.03	1.10	24.370	0.834	275378.			
16	45.7	1.12	1.38	24.122	0.550	256475.			
17	49.2	0.76	0.88	23.990	0.638	229012.			
18	51.6	1.29	1.86	24.200	0.538	209822.			
19	55.1	0.62	0.70	23.929	0.680	182190.			

WEIGHTED MEAN E IS 23.508 + OR - 0.159
REDUCED CHISQ FOR E CALCULATION IS 0.630 (SHOULD BE 1.)
UNWEIGHTED MEAN OF SIGMAS IS 0.724
CALCULATED OVER 18. DEGREES OF FREEDOM

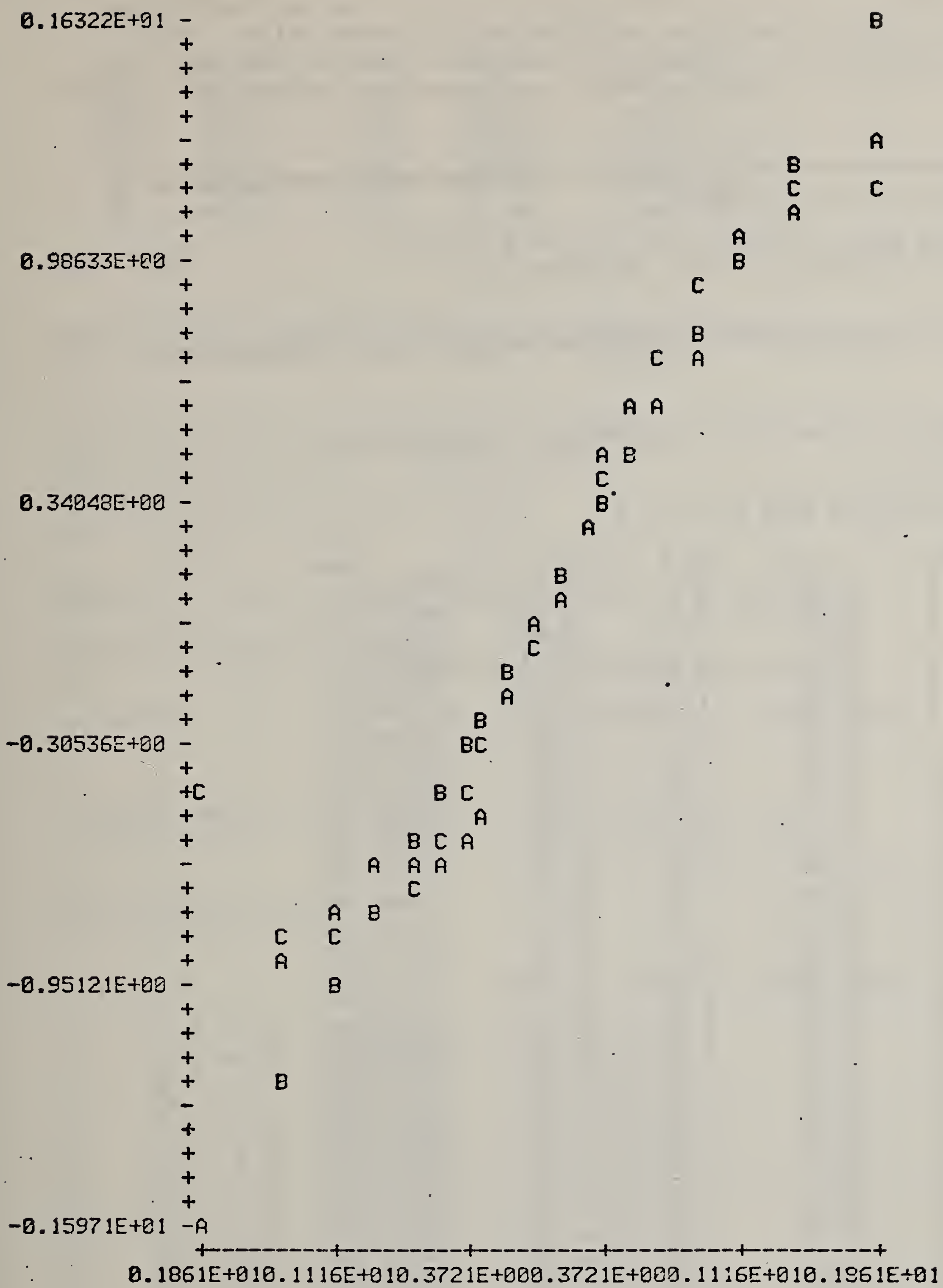
ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE 23.195 TO 23.821
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 23.495 AND 0.124
THIS GIVES A CONFIDENCE RANGE OF 23.251 TO 23.739
CHI-SQUARE FOR UNWEIGHTED E IS 0.63

19 DATA WRITTEN ON FILE = 8 =

DE = -0.030+ 1.090*Z + 0.117*Z **2 + -0.193*Z **3
REDUCED CHI SQUARE IS 0.001 CALCULATED OVER 19 POINTS

3 CURVES GENERATED FOR DE/Z PLOTS
Z IS THE HORIZONTAL AXIS
DE IS THE FIRST CURVE SECOND CURVE IS LINEAR FIT OF DE TO Z
THIRD CURVE IS POLYNOMIAL FIT OF DE TO Z



A = DE V. Z
B = LINEAR FIT OF DE TO Z
C = POLYNOMIAL FIT OF DE TO Z
JOB COMPLETED

An example of an incoherent data set where there was insufficient coherence to form a basis for rejection of incoherent values. The solution adopted was to examine only the values for the last few percent of reaction where the sample was expected to behave most ideally.

```
*****************************************************************************
TITLE OF INPUT FILE IS :

NBS IN 100% N2 P=8500 MM HG 3/22/78 BIG SAMPLE
PUMP OUT APPARATUS FIRST. RUN OVERNIGHT

*****************************************************************************
```

CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM IS 15.06

PUT PROCESSED DATA ON FILE 8

NUMBER	E	SIG E	D.O.C.	SWT
1	12.69	2.57	4.15	1085470.00
2	36.82	3.48	5.53	1067210.00
3	31.17	1.30	8.26	1031410.13
4	35.91	1.14	9.44	1015930.19
5	20.64	3.09	11.20	992725.00
6	38.86	1.12	15.10	941400.00
7	35.49	2.15	16.69	920566.00
8	30.65	2.71	19.24	887012.00
9	32.50	3.04	20.97	864279.00
10	46.12	1.09	23.48	831207.00
11	18.74	1.35	25.06	810437.00
12	34.84	0.93	27.81	774356.00
13	42.76	0.96	29.40	753337.00
14	24.64	1.08	32.05	718563.00
15	60.62	1.20	35.71	670464.00
16	21.43	0.73	38.32	636114.00
17	43.09	0.91	40.16	611956.00
18	21.01	1.22	42.76	577762.00
19	29.63	0.96	46.34	530689.00
20	29.73	0.73	48.82	498026.00
21	43.17	1.24	50.56	475073.00
22	31.28	1.07	53.03	442594.00
23	42.98	1.14	54.69	420855.00
24	39.52	1.05	57.07	389516.00
25	36.25	0.90	58.62	369091.00
26	43.91	0.79	60.87	339541.00
27	33.85	0.81	62.34	320223.00
28	33.09	0.77	64.43	292732.00
29	45.22	0.86	65.81	274631.00
30	33.92	0.61	67.78	248676.00
31	36.71	0.65	69.08	231541.00
32	36.89	0.62	70.91	207555.00
33	40.48	0.60	72.10	191946.00
34	36.94	0.44	73.78	169944.00
35	41.05	0.29	74.83	155931.00
36	46.10	0.87	77.16	125345.00
37	36.50	1.03	80.68	79011.37
38	53.48	0.87	82.69	52586.20
39	47.25	0.89	85.45	16312.40

40	40.06	0.65	86.97	-3690.95
41	47.16	0.59	89.05	-30987.50
42	46.37	0.29	90.17	-45714.70
43	45.71	0.40	92.45	-75782.38
44	46.80	0.51	95.03	-109618.02
45	45.04	0.25	96.15	-124405.00
46	45.75	0.61	97.80	-146053.00

PASS NUMBER 1

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 41.272
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 1.099

EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD
OUTLIERS ON BASIS OF DS STATISTIC, WHERE
DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K
K = SCALE FACTOR
AND Z = NORMAL DISTRIBUTION QUANTILE

DS = 0.000(0.147) + 0.584(0.152)*Z
REDUCED CHI SQUARE = 0.017 CALCULATED OVER 46 POINTS

RESCALE DS VALUES BY FACTOR OF 0.584 .
REJECTED OUTLIERS FOLLOW(ABS(DS VALUE) GREATER THAN 2.685)

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
46 DATA LEFT

OUTPUT REMAINING DATA

#	DOC/TIME	DS	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	4.1	-1.98	-1.82	12.694	2.575	1085470.			
2	5.5	-0.77	-1.07	36.818	3.484	1067210.			
3	8.3	1.76	1.28	31.172	1.298	1031410.			
4	9.4	2.28	2.22	35.911	1.141	1015930.			
5	11.2	0.59	0.97	20.645	3.093	992725.			
6	15.1	0.37	0.67	38.863	1.120	941400.			
7	16.7	2.08	1.82	35.490	2.146	920566.			
8	19.2	0.34	0.54	30.649	2.713	887012.			
9	21.0	1.45	1.17	32.505	3.039	864279.			
10	23.5	-0.72	-0.89	46.123	1.095	831207.			
11	25.1	1.85	1.42	18.745	1.349	810437.			
12	27.8	2.05	1.59	34.841	0.929	774356.			
13	29.4	0.30	0.48	42.758	0.958	753337.			
14	32.0	0.65	1.07	24.640	1.077	718563.			
15	35.7	0.02	0.08	60.623	1.200	670464.			
16	38.3	0.07	0.14	21.428	0.733	636114.			

17	40.2	0.27	0.42	43.092	0.913	611956.
18	42.8	0.45	0.74	21.011	1.220	577762.
19	46.3	-0.39	-0.48	29.631	0.961	530689.
20	48.8	-0.01	0.03	29.729	0.728	498026.
21	50.6	0.48	0.81	43.171	1.236	475073.
22	53.0	0.07	0.19	31.281	1.069	442594.
23	54.7	-1.26	-1.28	42.980	1.138	420855.
24	57.1	-0.74	-0.97	39.519	1.050	389516.
25	58.6	-0.69	-0.81	36.255	0.899	369091.
26	60.9	-0.67	-0.74	43.907	0.789	339541.
27	62.3	-0.53	-0.67	33.851	0.805	320223.
28	64.4	-0.40	-0.54	33.095	0.769	292732.
29	65.8	-0.31	-0.42	45.223	0.864	274631.
30	67.8	-0.26	-0.36	33.924	0.614	248676.
31	69.1	-0.23	-0.30	36.706	0.551	231541.
32	70.9	-0.11	-0.25	36.889	0.622	207555.
33	72.1	-0.04	-0.03	40.490	0.598	191945.
34	73.8	0.23	0.30	36.942	0.441	169844.
35	74.8	0.25	0.36	41.054	0.290	155931.
36	77.2	0.36	0.60	46.105	0.871	125345.
37	80.7	0.50	0.89	36.497	1.027	79011.
38	82.7	-0.59	-0.60	53.484	0.874	52586.
39	85.4	0.19	0.25	47.246	0.894	15312.
40	87.0	-0.09	-0.14	40.063	0.652	-3691.
41	89.0	-0.05	-0.08	47.162	0.587	-30988.
42	90.2	-0.09	-0.19	46.369	0.292	-45715.
43	92.4	-1.43	-1.42	45.711	0.399	-75782.
44	95.0	-1.97	-1.59	46.894	0.515	-109618.
45	96.1	-0.99	-1.17	45.037	0.254	-124405.
46	97.8	-2.21	-2.22	45.755	0.605	-146053.

PASS NUMBER 2

PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS 41.272
UNWEIGHTED AVERAGE OF SIGMA VALUES IS 1.099
EXAMINE E AND SIGE RELATIONSHIP TO DETECT
OUTLIERS ON DE STATISTIC
FIT OF DE TO Z
DE = (E(I)-E AVERAGE)/SIG E(I)
Z = NORMAL DISTRIBUTION QUANTILE
NUMBERS IN PARENTHESES ARE STANDARD DEVIATION

RESCALE SIGE VALUES BY FACTOR OF 10.667

INPUT SIGE VALUES NOW RESCALED BY FACTOR OF 10.667

REJECTED OUTLIERS FOLLOW(ABS(DE VALUE) GREATER THAN 2.685)

# DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE

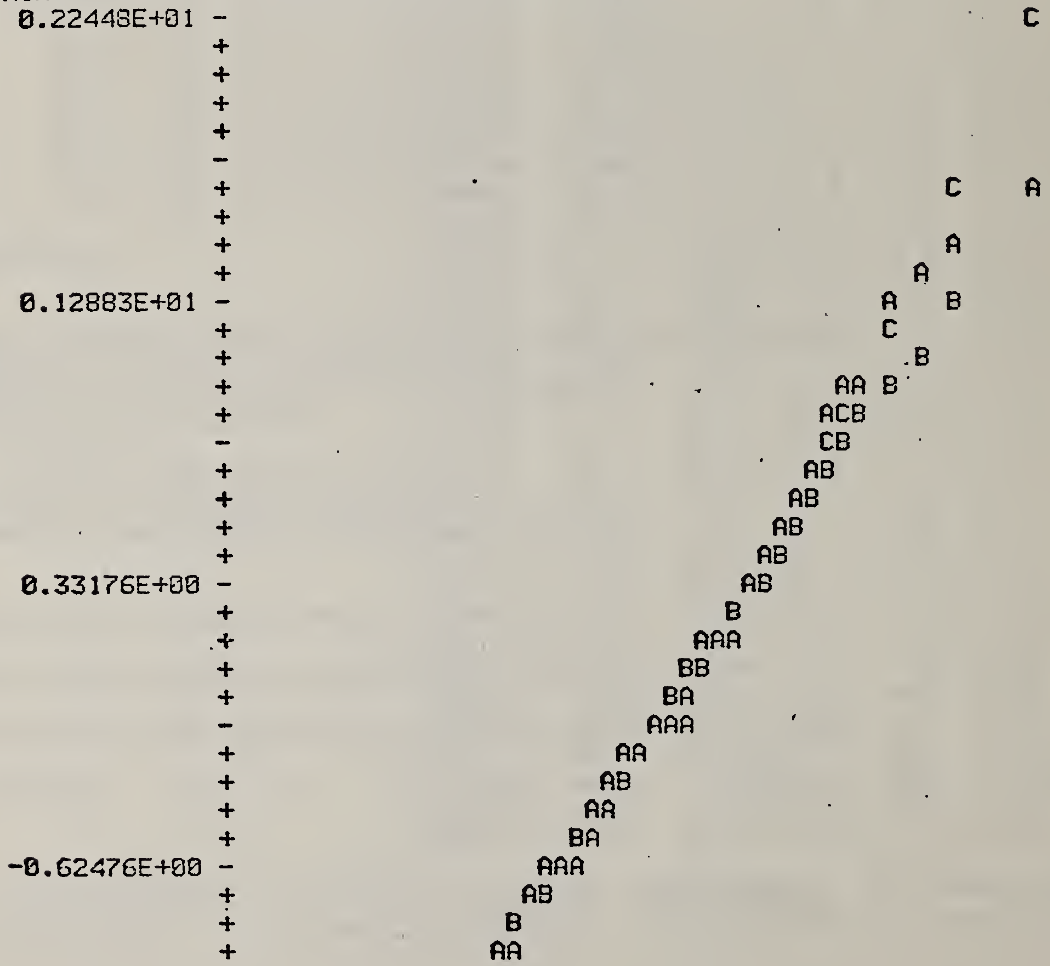
0 DATA DISCARDED AS OUTLIERS IN TAILS OF DISTRIBUTION
46 DATA LEFT

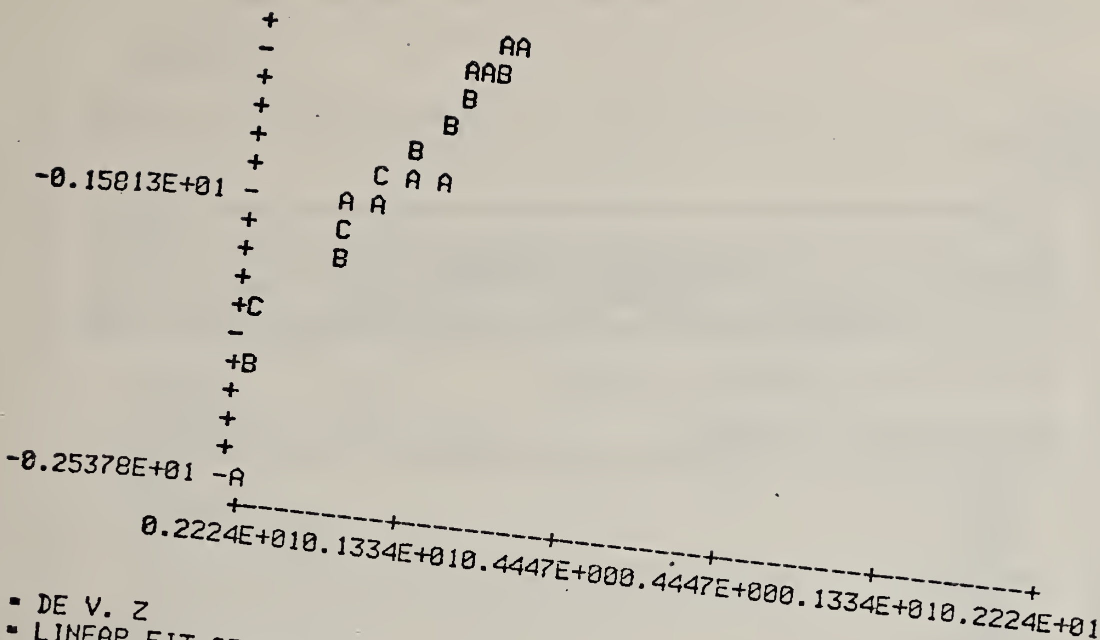
OUTPUT REMAINING DATA

#	DOC/TIME	DE	Z	E	SIGE	T/SWT	RATE	TEMP	RATE
1	4.1	-1.04	-0.97	12.694	27.468	1085470.			
2	5.5	-0.12	0.25	36.818	37.165	1067210.			
3	8.3	-0.73	-0.60	31.172	13.846	1031410.			
4	9.4	-0.44	-0.25	35.911	12.171	1015930.			
5	11.2	-0.63	-0.36	20.645	32.994	992725.			
6	15.1	-0.20	0.03	38.863	11.947	941400.			
7	16.7	-0.25	-0.03	35.490	22.892	920566.			
8	19.2	-0.37	-0.14	30.649	28.941	887012.			
9	21.0	-0.27	-0.08	32.505	32.418	864279.			
10	23.5	0.42	0.67	46.123	11.681	831207.			
11	25.1	-1.57	-1.82	18.745	14.390	810437.			
12	27.8	-0.65	-0.42	34.841	9.910	774356.			
13	29.4	0.15	0.48	42.758	10.219	753337.			
14	32.0	-1.45	-1.28	24.640	11.489	718563.			
15	35.7	1.51	1.82	60.623	12.801	670464.			
16	38.3	-2.54	-2.22	21.428	7.819	636114.			
17	40.2	0.19	0.54	43.092	9.739	611956.			
18	42.8	-1.56	-1.59	21.011	13.014	577762.			
19	46.3	-1.14	-1.17	29.631	10.251	530689.			
20	48.8	-1.49	-1.42	29.729	7.766	498026.			
21	50.6	0.14	0.42	43.171	13.185	475073.			
22	53.0	-0.83	-0.74	31.281	11.403	442594.			
23	54.7	0.14	0.35	42.980	12.139	420855.			
24	57.1	-0.16	0.14	39.519	11.201	389516.			
25	58.6	-0.52	-0.30	36.255	9.590	369091.			
26	60.9	0.31	0.60	43.907	8.417	339541.			
27	62.3	-0.86	-0.67	33.851	8.587	320223.			
28	64.4	-1.00	-0.83	33.095	8.203	292732.			
29	65.8	0.43	0.74	45.223	9.217	274631.			
30	67.8	-1.12	-1.07	33.924	6.550	248676.			
31	69.1	-0.66	-0.48	36.706	6.944	231541.			
32	70.9	-0.66	-0.54	36.889	6.635	207555.			
33	72.1	-0.12	0.19	40.480	6.379	191946.			
34	73.8	-0.92	-0.81	36.942	4.704	169844.			
35	74.8	-0.07	0.30	41.054	3.094	155931.			
36	77.2	0.52	0.81	46.105	9.291	125345.			
37	80.7	-0.44	-0.19	36.497	10.955	79311.			
38	82.7	1.31	1.42	53.484	9.323	52586.			
39	85.4	0.63	0.89	47.246	9.537	16312.			
40	87.0	-0.17	0.08	40.063	6.955	-3691.			
41	89.0	0.94	1.07	47.162	6.262	-30983.			
42	90.2	1.64	2.22	46.369	3.115	-45715.			
43	92.4	1.04	1.28	45.711	4.256	-75782.			
44	95.0	1.01	1.17	46.804	5.494	-109618.			
45	96.1	1.39	1.59	45.037	2.710	-124405.			
46	97.8	0.69	0.97	45.755	6.454	-146053.			

WEIGHTED MEAN E IS 41.272 + OR - 1.051
REDUCED CHISQ FOR E CALCULATION IS 0.870 (SHOULD BE 1.)

```
UNWEIGHTED MEAN OF SIGMAS IS  11.729
CALCULATED OVER   45. DEGREES OF FREEDOM

ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY
DISTRIBUTED. TRUE WEIGHTED MEAN LIES WITHIN RANGE 39.211 TO 43.332
FOR 95% CONFIDENCE LEVEL

UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE
OF ITS SIGMA ARE 37.454 AND  1.394
THIS GIVES A CONFIDENCE RANGE OF 34.721 TO 40.186
CHI-SQUARE FOR UNWEIGHTED E IS   1.16

 46 DATA WRITTEN ON FILE =   8 =


DE =  -0.264+  0.873*Z +  0.084*Z **2 +  0.014*Z **3
REDUCED CHI SQUARE IS  0.000 CALCULATED OVER  46 POINTS

3 CURVES GENERATED FOR DE/Z PLOTS
Z IS THE HORIZONTAL AXIS
DE IS THE FIRST CURVE SECOND CURVE IS LINEAR FIT OF DE TO Z
THIRD CURVE IS POLYNOMIAL FIT OF DE TO Z
```





A = DE V. Z
B = LINEAR FIT OF DE TO Z
C = POLYNOMIAL FIT OF DE TO Z
JOB COMPLETED

Listings of the routines in the FORTRAN program TGDEPG.

```
$ASSM
TGDEN PROG
$FORT
$TRGT 16
C   ************************************************************
C
C   TGDEN, SET UP FOR MULTIPLE RUNS FROM ONE START.
C   PROGRAM TO EVALUATE TG DATA DERIVED FROM TEMPERATURE JUMPS
C
C   **************************************************************
C
C   WRITTEN BY B. DICKENS, 311.02 NBS
C
C   TELEPHONE (301) 921-3322
C   PROGRAM NEEDS THE FOLLOWING SUBROUTINES AND FUNCTIONS:
C      DATARD
C      PINV
C      PLOT
C      PROBP2
C      REGRES
C      PLOT
C      SORT5
C      WTAVER
C
C
C   COMMON BLOCKS /IO/ AND /REGRS/ DEFINED VIA ARRAY 'DUMP'
C       COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C       COMMON /REGRS/ M(5),R(5),YFIT(250)
C
      COMMON  DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
C
      DIMENSION NTITLE(64),NTIME(50),TREAD(2,50),RATE(2,50)
      DIMENSION TEMP(2,50),SIGT(2,50),SIGR(2,50),E(50),SIGE(50)
      DIMENSION SWT(50),DOC(50)
      DIMENSION DEZ(10),DE(50),Z(50)
      DIMENSION DEPROB(50),ID(50),VRATE(50),VTEMP(50)
      DIMENSION SIGLOG(50)
      DIMENSION IDOLD(50),AZ(10),SIGAZ(10)
      DIMENSION PLOTX(100,6),PLOTY(100,6),NPTS(6)
C
C    ***  NOTE THAT THIS EQUIVALENCE MAY GIVE SOME SLIGHT ROUND-OFF
C         CHANGES IN NTIME (=DOC= IS USUALLY USED IN THE PROGAM
C         SO DOC SHOULD REMAIN UNCHANGED)    ***
      EQUIVALENCE (DOC(1),NTIME(1))
C
C   OVERLAY NAMES AND FILE ASSIGNMENTS
C
      INTEGER*2 IPRIMS(3),ISTTSS(3)
      DATA IPRIMS /'OP','RI','MS'/, ISTTSS /'OS','TT','SS'/
      NPRIMS=11
      NSTTSS=12
      NTTYO=4
      CALL OVERLY (IPRIMS,NPRIMS,0,NTTYO)
C
C   INITIALISE AND READ IN DATA
C
      CALL PRIMES (NLOOPY,NDATA,NLOOP,SAVER,SAVE,INDATA)
  100 IF (NLOOPY.GT.0) CALL OVERLY (IPRIMS,NPRIMS,0,NTTYO)
      CALL DATARD (SAVER,NTITLE,NLOOPY,IDOLD,E,SIGE,SWT,DOC,NTIME,TF
     & ,SIGT,RATE,SIGR,ID,INDATA,NDATA,NERROR)
      IF (NERROR.GT.0) GO TO 600
```

```
      CALL OVERLY (ISTTSS,NSTTSS,0,NTTYO)
      SIGSCL = 1.
C
C  NTYME IS NUMBER OF CURRENT PASS WHEN SEEKING TO REJECT
C  OUTLIERS AND RESCALE SIGE(I) VALUES.
C
      NTYME = 0
C
C  BEGIN BY CHECKING SIGMA DISTRIBUTION
C  ASSUME LN(SIGE(I)) IS NORMALLY DISTRIBUTED
C  (AS ORIGINALLY FORMULATED, THE PROGRAM WAS TO CHECK SIGE(I)**2
C  AGAINST A CHI-SQUARE DISTRIBUTION, BUT THAT VERSION
C  GOT TO BE TOO LARGE)
C
      DO 180 I = 1,NDATA
      DEPROB(I) = 1.
      SIGLOG(I) = ALOG(SIGE(I))
  180 CONTINUE
      NSIG = 0
      NFLAG2 = 'Z '
190   NTYME = NTYME+1
      IF (NFILEB.GT.6) WRITE(NFILEB,200) NTYME
      IF (NPRINT.GE.3) WRITE (NPRINT,200) NTYME
200   FORMAT (/12H PASS NUMBER,I3)
C
C  GET WEIGHTED MEAN OF ACTIVATION ENERGY
C  AND AVERAGE VALUE (UNWEIGHTED) OF SIGE(I).
C
      CALL REGRES (DEPROB,E,SIGE,NDATA,0,M,1,YFIT,WTEAVE,AZ(1)
     &,WTSIZE,SIGAZ(1),R,RMUL,RCHMN,TEST)
      CALL REGRES (DEPROB,SIGE,E,NDATA,0,M,0,YFIT,SIGAVE,AZ(1)
     &,AZ(2),SIGAZ(1),R,RMUL,RCHSIG,TEST)
      IF (NFILEB.GT.6) WRITE (NFILEB,210)WTEAVE,SIGAVE
      IF (NPRINT.GE.3) WRITE (NPRINT,210) WTEAVE,SIGAVE
210   FORMAT (//51H PRESENT STATUS:-AVERAGE WEIGHTED E IN KCAL/MOLE IS,
     & F7.3/38H UNWEIGHTED AVERAGE OF SIGMA VALUES IS,F7.3/)
C
      IF (NSIG.EQ.1) GO TO 250
C
C  EXAMINE DISTRIBUTION OF SIG E VALUES
C
C  CALCULATE QUANTITIES FOR NORMAL PROBABILITY PLOT
C
      IF (NFILEB.GT.6) WRITE (NFILEB,220)
      IF (NPRINT.GE.3) WRITE (NPRINT,220)
220   FORMAT (//45H EXAMINE LOG(SIGE(I)) DISTRIBUTION TO DISCARD/
     & 42H OUTLIERS ON BASIS OF DS STATISTIC, WHERE /
     & '  DS = (AVERAGE LOG SIGE - LOG SIGE(I))/K'/'  K = SCALE FACTOR'/
     & '  AND Z = NORMAL DISTRIBUTION QUANTILE'/)
C
      CALL REGRES (DEPROB,SIGLOG,E,NDATA,0,M,0,YFIT,SIGLAV
     &,AZ(1),SIGAZ(1),SIGAZ(2),R,RMUL,RCHSIG,TEST)
      IF(NPRINT.GT.3)WRITE(NPRINT,952)SIGLAV
  952 FORMAT(' AVERAGE OF LOG SIGMAS IS',F10.3//)
      NFLAG1 = 2HDS
      CALL PROBP2 (DE,Z,SIGLOG,DEPROB,SIGLAV,ID,NDATA)
      CALL REGRES (Z,DE,DEPROB,NDATA,1,M,1,YFIT,AZ(1),AZ(2)
     &,SIGAZ(1),SIGAZ(2),R,RMUL,CHISQR,TEST)
      IF (NFILEB.GT.6) WRITE (NFILEB,11) NFLAG1,AZ(1),SIGAZ(1)
     & ,AZ(2),SIGAZ(2),NFLAG2,CHISQR,NDATA
      IF (NPRINT.GE.3) WRITE (NPRINT,11) NFLAG1,AZ(1),SIGAZ(1)
     & ,AZ(2),SIGAZ(2),NFLAG2,CHISQR,NDATA
11    FORMAT (/1H ,A2,2H =,F8.3,1H(,F7.3,4H) + ,F8.3,1H(,F7.3,2H)*,A2/
     & 21H REDUCED CHI SQUARE =,F7.3,17H CALCULATED OVER ,I5,7H POINTS/
     & )
C
```

```
C  APPLY SCALE TO DS AND SIGLOG
C
      DO 230 I = 1,NDATA
      SIGLOG(I) = SIGLOG(I)/AZ(2)
230   DE(I) = DE(I)/AZ(2)
      SIGSCL = SIGSCL*AZ(2)
      IF (NFILEB.GT.6) WRITE (NFILEB,240) SIGSCL
      IF (NPRINT.GE.3) WRITE (NPRINT,240) SIGSCL
240   FORMAT (31H RESCALE DS VALUES BY FACTOR OF,F9.3)
      GO TO 330
250   CONTINUE
      NFLAG1 = 2HDE
      IF (NFILEB.GT.6) WRITE (NFILEB,260)
      IF (NPRINT.GE.3) WRITE (NPRINT,260)
260   FORMAT (42H EXAMINE E AND SIGE RELATIONSHIP TO DETECT/
     & 25H OUTLIERS ON DE STATISTIC/)
C
C  CALCULATE QUANTITIES FOR NORMAL PROBABILITY PLOT.
C
      CALL PROBP2 (DE,Z,E,SIGE,WTEAVE,ID,NDATA)
C
C   EXAMINE LINEARITY OF FIT OF DE TO Z
C
      IF (NFILEB.GT.6) WRITE (NFILEB,270)
      IF (NPRINT.GE.3) WRITE (NPRINT,270)
270   FORMAT (15H FIT OF DE TO Z/31H DE = (E(I)-E AVERAGE)/SIG E(I)/
     & 33H Z = NORMAL DISTRIBUTION QUANTILE/
     & 46H NUMBERS IN PARENTHESES ARE STANDARD DEVIATION/)
      IF (NTYME.EQ.1.AND.NFILEB.GT.6) WRITE (NFILEB,280)
      IF (NTYME.EQ.1.AND.NPRINT.GE.3) WRITE (NPRINT,280)
280   FORMAT (49H ESTABLISH DE/Z CORRESPONDENCE AND HENCE SCALE OF,
     & 15H SIGE(I) VALUES/)
C
C  USE PROBABILITIES FROM NORMAL DISTRIBUTION AND
C  DE AS WEIGHTS IN CALCULATING DE/Z FIT
C
      DO 290 I = 1,NDATA
      DEPROB(I) = (0.39894228/SIGE(I))*EXP(-(DE(I)**2)/2.)
C
C  MAKE DEPROB SUITABLE FOR ROUTINE WHICH EXPECTS
C  STANDARD DEVIATIONS
C
290   DEPROB(I) = 1./SQRT(DEPROB(I))
      CALL REGRES (Z,DE,DEPROB,NDATA,1,M,1,YFIT,AZ(1),AZ(2)
     &,SIGAZ(1),SIGAZ(2),R,RMUL,CHISQR,TEST)
C
C  APPLY SCALE FACTOR TO SIGE AND DE
C
      DO 300 I = 1,NDATA
      SIGE(I) = SIGE(I)*AZ(2)
300   DE(I) = DE(I)/AZ(2)
      SIGSCL = SIGSCL*AZ(2)
      IF (NFILEB.GT.6) WRITE (NFILEB,310) AZ(2)
      IF (NPRINT.GE.3) WRITE (NPRINT,310) AZ(2)
310   FORMAT (/33H RESCALE SIGE VALUES BY FACTOR OF,F7.3/)
      IF (NFILEB.GT.6) WRITE(NFILEB,320)SIGSCL
      IF (NPRINT.GE.3) WRITE (NPRINT,320) SIGSCL
320   FORMAT (45H  INPUT SIGE VALUES NOW RESCALED BY FACTOR OF,F7.3/)
C
C  PUT DE AND Z ARRAYS BACK IN CORRESPONDENCE WITH INPUT ORDER
C  I.E. SORT ON ID ARRAY
C
330   CALL SORT5 (ID,DE,Z,NDATA)
C
C  CHOP OFF OUTLIERS, I.E. BEYOND ABS(DE(I).GT.SIGCUT)
C
```

```
      KOUNT = 0
C
C   SIGCUT IS NORMAL DEVIATE WHICH DIVIDES END AREA OF NORMAL
C   DISTRIBUTION SO THAT THERE IS A  = SAVE =  CHANCE OF REJECTING A  = GOOD =
C   END VALUE AND  = (1-SAVE) =  CHANCE OF ACCEPTING IT.
C   PRESUMABLY TRUE OUTLIERS WILL BE MOSTLY REJECTED UNDER THIS
C   CRITERION
C
      FDATA = NDATA
      TESTP = ((2.-3.1415927/8.)/(FDATA+1.-3.1415927/4.))*(1.-SAVE)
      SIGCUT = ABS(PINV(TESTP))
      IF (NFILEB.GT.6) WRITE (NFILEB,340) NFLAG1,SIGCUT
      IF (NPRINT.GE.3) WRITE (NPRINT,340) NFLAG1,SIGCUT
340   FORMAT (' REJECTED OUTLIERS FOLLOW  (ABS(',A2,
     & 20H VALUE) GREATER THAN,F6.3,1H))
      IF (NFILEB.GT.6) WRITE (NFILEB,350) NFLAG1
      IF (NPRINT.GE.3) WRITE (NPRINT,350) NFLAG1
350   FORMAT (/'  # DOC/TIME     ',A2,'      Z     E   SIGE   T/SWT',
     & 29H         RATE   TEMP        RATE/)
C
C   OUTPUT REJECTED VALUES
C
      DO 390 I = 1,NDATA
360   IF (ABS(DE(I)).LE.SIGCUT) GO TO 390
      IF (INDATA.GT.0)GO TO 365
      IF (NFILEB.GT.6) WRITE (NFILEB,370) IDOLD(I),NTIME(I),DE(I)
     & ,Z(I),E(I),SIGE(I)
     & ,(TEMP(K,I),RATE(K,I),K=1,2)
      IF (NPRINT.GE.3) WRITE (NPRINT,370) IDOLD(I),NTIME(I),DE(I)
     & ,Z(I),E(I),SIGE(I)
     & ,(TEMP(K,I),RATE(K,I),K = 1,2)
370   FORMAT (I3,I9,2F6.2,2F7.3,F7.2,E11.3,F7.2,E11.3)
      GO TO 375
  365 IF (NFILEB.GT.6) WRITE (NFILEB,372)IDOLD(I),DOC(I),DE(I)
     & ,Z(I),E(I),SIGE(I),SWT(I)
      IF (NPRINT.GE.3) WRITE (NPRINT,372)IDOLD(I),DOC(I),DE(I)
     & ,Z(I),E(I),SIGE(I),SWT(I)
  372 FORMAT (I3,F9.1,2F6.2,2F7.3,F10.0)
  375 CONTINUE
      KOUNT = KOUNT+1
      NDATA = NDATA-1
      IF (I.GT.NDATA) GO TO 400
C
C   REMOVE  VALUES FROM ARRAYS AND CLOSE UP ARRAYS
C
      DO 380 J = I,NDATA
      IDOLD(J) = IDOLD(J+1)
      E(J) = E(J+1)
      SIGE(J) = SIGE(J+1)
      SIGLOG(J) = SIGLOG(J+1)
      DE(J) = DE(J+1)
      Z(J) = Z(J+1)
      SWT(J)=SWT(J+1)
      DOC(J)=DOC(J+1)
      DO 380 K = 1,2
      TEMP(K,J) = TEMP(K,J+1)
      RATE(K,J) = RATE(K,J+1)
      SIGT(K,J) = SIGT(K,J+1)
      SIGR(K,J) = SIGR(K,J+1)
380   CONTINUE
C
C   CHECK NEW INFORMATION WHICH IS NOW IN OLD POSITION
C
      GO TO 360
390   CONTINUE
400   CONTINUE
```

```
      DO 410 I = 1,NDATA
410   ID(I) = I
      IF (NFILEB.GT.6) WRITE (NFILEB,420) KOUNT,NDATA
      IF (NPRINT.GE.3) WRITE (NPRINT,420) KOUNT,NDATA
420   FORMAT (//I4,33H DATA DISCARDED AS OUTLIERS IN TAILS OF,
     & 13H DISTRIBUTION/I4,10H DATA LEFT/)
      IF (NDATA.LE.0) STOP
      IF (NFILEB.GT.6) WRITE (NFILEB,425)
      IF (NPRINT.GE.3) WRITE (NPRINT,425)
425   FORMAT (/22H OUTPUT REMAINING DATA/)
      IF (NFILEB.GT.6) WRITE (NFILEB,350) NFLAG1
      IF (NPRINT.GE.3) WRITE (NPRINT,350) NFLAG1
      DO 430 I = 1,NDATA
      IF (INDATA.EQ.0)GO TO 426
      IF (NFILEB.GT.6) WRITE (NFILEB,372) IDOLD(I),DOC(I),DE(I)
     & ,Z(I),E(I),SIGE(I),SWT(I)
      IF (NPRINT.GE.3) WRITE (NPRINT,372) IDOLD(I),DOC(I),DE(I)
     & ,Z(I),E(I),SIGE(I),SWT(I)
      GO TO 430
 426  IF (NFILEB.GT.6) WRITE (NFILEB,370) IDOLD(I),NTIME(I),DE(I)
     & ,Z(I),E(I),SIGE(I),(TEMP(K,I),RATE(K,I),K=1,2)
      IF (NPRINT.GE.3) WRITE (NPRINT,370) IDOLD(I),NTIME(I),DE(I)
     & ,Z(I),E(I),SIGE(I),(TEMP(K,I),RATE(K,I),K=1,2)
430   CONTINUE
C
C  BEGIN AGAIN IF OUTLIERS HAVE BEEN DISCARDED
C
      IF (KOUNT.GT.0) GO TO 190
C
C  IF NO OUTLIERS WERE REJECTED, CHECK TO SEE IF TEST JUST
C  CONCLUDED WAS ON E AND SIGE VALUES
C
      IF (NSIG.GE.1) GO TO 440
C
C  NOW TEST ON E AND SIGE VALUES
C
      NSIG = 1
      SIGSCL = 1.
      GO TO 190
440   CONTINUE
C
C  AVERAGE E VALUES IN VARIOUS WAYS FOR FINAL PASS
C  AND CALCULATE CONFIDENCE LIMITS
C
      CALL WTAVER (E,SIGE,NDATA,WTEAVE,SIGWTE,EAVE,ESIG)
C
C  OUTPUT RESULTS ON FILE FOR SUBSEQUENT PROCESSING
C
      WRITE (NTTYO,460) NFILEA
460   FORMAT (/7H FILE =,I4,26H = IS FOR TRANSMISSION OF /
     & 45H PROCESSED E/SIGMA E DATA TO SUBSEQUENT STEPS/)
      WRITE (NFILEA,120) NTITLE
  120 FORMAT(32A2/32A2)
      WRITE (NFILEA,470) WTEAVE,SIGWTE,EAVE,ESIG,NDATA
470   FORMAT (4F10.3,I5)
      IF (INDATA.EQ.0) WRITE (NFILEA,480)
     & (NTIME(I),E(I),SIGE(I),(TEMP(J,I),SIGT(J,I),RAT
     & E(J,I),SIGR(J,I),J = 1,2),IDOLD(I),I = 1,NDATA)
480   FORMAT (I5,2F7.3,F7.1,F4.1,2F6.3,F7.1,F4.1,2F6.3,I5)
      IF (INDATA.GT.0) WRITE (NFILEA,485)
     & (IDOLD(I),E(I),SIGE(I),DOC(I),SWT(I),I=1,NDATA)
  485 FORMAT (I5,3F10.2,F15.2)
      END FILE NFILEA
      IF (NFILEB.GT.6) WRITE (NFILEB,490) NDATA,NFILEA
      IF (NPRINT.GE.3) WRITE (NPRINT,490) NDATA,NFILEA
490   FORMAT (/I4,23H DATA WRITTEN ON FILE =,I4,2H =/)
```

```
C
C  GET COEFFICIENTS FOR LINEAR PLOT
C    USE E AND SIGE AS TEMPORARY STORAGE ARRAYS
C
      CALL REGRES (Z,DE,DEPROB,NDATA,1,M,1,E,AZ(1),AZ(2)
     & ,SIGAZ(1),SIGAZ(2),R,RMUL,CHISQR,TEST)
C
C  USE 4 TERM POLYNOMINAL TO OBTAIN WEIGHTED FIT OF DE TO Z
C
      CALL REGRES (Z,DE,DEPROB,NDATA,3,M,1,SIGE;DEZ(1)
     &,DEZ(2),SIGAZ(1),SIGAZ(2),R,RMUL,CHISQR,TEST)
      IF (NFILEB.GT.6) WRITE (NFILEB,450) NFLAG1,DEZ(1),DEZ(2)
     & ,NFLAG2,DEZ(3),NFLAG2,DEZ(4),NFLAG2,CHISQR,NDATA
      IF (NPRINT.GE.3) WRITE (NPRINT,450) NFLAG1,DEZ(1),DEZ(2)
     & ,NFLAG2,DEZ(3),NFLAG2,DEZ(4),NFLAG2,CHISQR,NDATA
450   FORMAT (/1H ,A2,3H = ,F7.3,1H+,F7.3,1H*,A2,1H+,F7.3,1H*,A2,3H**2,
     & 2H +,F7.3,1H*,A2,3H**3/22H REDUCED CHI SQUARE IS,F7.3,
     & ' CALCULATED OVER',I4,' POINTS')
      DO 495 I = 1,NDATA
      PLOTY(I,1) = DE(I)
      PLOTY(I,2) = E(I)
      PLOTY(I,3) = SIGE(I)
      DO 495 K = 1,3
      PLOTX(I,K) = Z(I)
  495 CONTINUE
      DO 570 I = 1,3
570   NPTS(I) = NDATA
C
C  PLOT FINAL RESULTS
C
      WRITE (NTTYO,520)
      IF (NFILEB.GT.6) WRITE (NFILEB,520)
      IF (NPRINT.GE.3) WRITE (NPRINT,520)
520   FORMAT (/' 3 CURVES GENERATED FOR DE/Z PLOTS'/
     & ' Z IS THE HORIZONTAL AXIS'/' DE IS THE FIRST CURVE'
     & 38H SECOND CURVE IS LINEAR FIT OF DE TO Z/
     & 41H THIRD CURVE IS POLYNOMIAL FIT OF DE TO Z/)
      CALL PLOT (3,NPTS,PLOTX,PLOTY)
580   CONTINUE
      IF (NFILEB.GT.6) WRITE (NFILEB,590)
      IF (NPRINT.GE.3) WRITE (NPRINT,590)
590   FORMAT (//12H A = DE V. Z/26H B = LINEAR FIT OF DE TO Z/
     & 30H C = POLYNOMIAL FIT OF DE TO Z/14H JOB COMPLETED/'1')
C
C  FINISH OFF SAVE FILE IN CASE OF PREMATURE TERMINATION
C
  600 IF (NERROR.GT.0)END FILE NFILEA
      IF (NLOOPY.LT.NLOOP) GO TO 100
      STOP
      END
```

```
$ASSM
ACTIV PROG
$FORT
$TRGT 16
C ***************************************************************
C
      SUBROUTINE ACTIV(E,SIGE,TEMP1,TEMP2,SIGT1,SIGT2
     &,TREAD1,TREAD2,SIGTR1,SIGTR2
     &,RATE1,RATE2,SIGR1,SIGR2,AT,NERROR)
C
C ***************************************************************
C  SUBROUTINE TO CALCULATE ACTIVATION ENERGY FROM TEMPERATURE AND
C  RATE DATA
C
C   *** FOR A SINGLE E VALUE ***
C
C  E  =  ACTIVATION ENERGY
C  SIGE  =  STANDARD DEVIATION OF E
C  IE  =  ORDINAL NUMBER OF E IN LIST OF E VALUES
C  TEMP1  =  FIRST TEMPERATURE (IN DEGREES K)
C  TEMP2  =  SECOND TEMPERATURE
C  SIGT1, SIGT2,  =  STANDARD DEVIATIONS OF TEMP1 AND TEMP2
C  TREAD  =  VALUES OF CORRESPONDING THERMOCOUPLE EMF'S IN MICROVOLTS
C  VTEMP  =  CONTRIBUTION OF TEMP TO VARIANCE OF E
C
C  RATE1, RATE2, SIGR1, SIGR2 AND VRATE ARE SIMILAR VALUES FOR RATES
C  AT  =  POLYNOMIAL COEFFICIENTS FOR EMF TO DEGREE K CONVERSION
C  NERROR  =  ERROR INDICATOR
C
C
C  NEEDS THE FOLLOWING COMMON BLOCK AND FUNCTION:
C    COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C    TDEGAB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION TEMP(2),SIGT(2),TREAD(2),SIGTR(2),RATE(2),SIGR(2),AT(4)
C
C
      R = 1.987/1000.
      R2 = R**2
      NERROR = 0
      TREAD(1) = TREAD1
      TREAD(2) = TREAD2
      SIGTR(1) = SIGTR1
      SIGTR(2) = SIGTR2
      RATE(1) = RATE1
      RATE(2) = RATE2
      SIGR(1) = SIGR1
      SIGR(2) = SIGR2
      DO 30 J = 1,2
C
C  GET TEMPERATURE IN DEGREES KELVIN, TREAD IS IN MICROVOLTS
C
      TEMP(J) = TDEGAB (TREAD(J),AT)
      T = TREAD(J)+SIGTR(J)
      S = TDEGAB(T,AT)
      SIGT(J) = S-TEMP(J)
      IF (TEMP(J).LT.1.) GO TO 40
      IF (ABS(RATE(J)).LT.1.E-06) GO TO 40
30    CONTINUE
C
C  GET ACTIVATION ENERGY
```

```
C
      TDIFF = TEMP(2)-TEMP(1)
      TDIFF = ABS(TDIFF)
      IF (TDIFF.LT.1.E-01) GO TO 40
      T = RATE(2)/RATE(1)
      IF (T.LT.1.E-06) GO TO 40
      IF (T.GT.1.E+06) GO TO 40
      IF (T.LT.1.) T = 1./T
      RTEMP = ALOG(T)
      ETEMP = TEMP(2)*TEMP(1)/TDIFF
      E = R*RTEMP*ETEMP
      IF (ABS(E).GT.200..OR.ABS(E).LT.1.) GO TO 40
      GO TO 90
C
C  DISCARD THIS POINT FOR VARIOUS POSSIBLE REASONS
C
40    CONTINUE
      IF (NPRINT.GE.3) WRITE (NPRINT,50)IE
      IF (NFILEB.GT.6) WRITE (NFILEB,50)IE
   50 FORMAT (' ACT. ENERGY NUMBER',I3,' DISCARDED')
      NERROR = 8
      RETURN
90    CONTINUE
      E2 = E**2
C
C  GET ASSOCIATED SIGMA
C  1) FROM RATES
C
      VRATE = ((SIGR(2)/RATE(2))**2+(SIGR(1)/RATE(1))**2)*E
     &2/RTEMP**2
C
C  2) FROM TEMPERATURES
C
      VTEMP = ((SIGT(1)/TEMP(1)**2)**2+(SIGT(2)/TEMP(2)**2)
     &**2)*E2*ETEMP**2
C
C  COMPUTE TOTAL SIGMA ON E
C
      SIGE = SQRT(VRATE+VTEMP)
C
C  OUTPUT INITIAL AND CALCULATED VALUES
C
      TEMP(1) = TEMP(1)-273.15
      TEMP(2) = TEMP(2)-273.15
      IF (NPRINT.LT.3)GO TO 200
      WRITE (NPRINT,52)
   52 FORMAT (/' ACTIVATION ENERGY FROM  = ACTIVE = '
     &/6X,'E',2X,'SIG E',2X,'FACTOR OR RATE '/)
      WRITE (NPRINT,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J = 1,2)
100       FORMAT (2F7.3,F8.2,F6.2,2E13.4/14X,F8.2,F6.2,2E13.4)
110       CONTINUE
C
C
120   WRITE (NPRINT,130)
130   FORMAT (/' CONTRIBUTION OF RATE AND TEMPERATURE TO VARIANCE'
     &,'( = SIGE**2)'/'     RATE     TEMP'/)
      WRITE (NPRINT,140) VRATE,VTEMP
140   FORMAT (2E12.3)
  200 IF (NFILEB.LE.6)RETURN
      WRITE (NFILEB,52)
      WRITE (NFILEB,100)E,SIGE,(TEMP(J),SIGT(J),RATE(J),SIGR(J),J = 1,2)
      WRITE (NFILEB,130)
      WRITE (NFILEB,140)VRATE,VTEMP
      TEMP1 = TEMP(1)
      TEMP2 = TEMP(2)
      SIGT1 = SIGT(1)
      SIGT2 = SIGT(2)
      RETURN
      END
```

```
$ASSM
AVSDN PROG
$FORT
$TRGT 16
C ******************************************************************
C
      SUBROUTINE AVSDN(AVE,SD,Q,NVAL)
C
C ******************************************************************
C  SUBROUTINE TO CALCULATE AVERAGE AND STANDARD DEVIATION
C  AVE = AVERAGE VALUE
C  Q = INPUT VALUES
C  SD = STANDARD DEVIATION OF POPULATION OF Q
C  NVAL = NUMBER OF VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
C  RETURNS SD = 1.E+06 IF ONLY ONE VALUE IS AVAILABLE
C
C
      DIMENSION Q(NVAL)
C
C  CALCULATE AVERAGE
C
      AVE=0.
      DO 10 I=1,NVAL
   10 AVE=AVE+Q(I)
      AVE=AVE/FLOAT(NVAL)
      IF(NVAL.GT.1) GO TO 50
      SD=1.0E+06
      RETURN
   50 SD=0.
C
C  CALCULATE STANDARD DEVIATION
C
      DO 40 I=1,NVAL
 40   SD=SD+(AVE-Q(I))**2
      SD=SQRT(SD/(FLOAT(NVAL-1)))
      RETURN
      END
```

```
$ASSM
CHISQ PROG
$FORT
$TRGT 16
C ***********************************************************
C
      SUBROUTINE CHISQ(P,SP,XSQ,N)
C
C ***********************************************************
C  SUBROUTINE TO CALCULATE REDUCED CHI SQUARE FOR N INPUT VALUES.
C  P = INPUT VALUES
C  SP = STANDARD DEVIATIONS OF P VALUES
C  XSQ = CHI-SQUARE VALUE FOR P AND SP ARRAYS
C  N = NUMBER OF P VALUES
C
C  NEEDS NO COMMON BLOCKS, FUNCTIONS OR SUBROUTINES
C
      DIMENSION P(N),SP(N)
      AVE=0.
C
C  CALCULATE AVERAGE
C
      DO 10 I=1,N
 10   AVE=AVE+P(I)
      AVE=AVE/FLOAT(N)
      XSQ=0.
      IF(N.LE.1)RETURN
C
C  CALCULATE CHI SQUARE
C
      DO 20 I=1,N
 20   XSQ=XSQ+((AVE-P(I))/SP(I))**2
      XSQ=XSQ/FLOAT(N-1)
      RETURN
      END
```

```
$ASSM
DATARD PROG
$FORT
$TRGT 16
C   ******************************************************************
C
       SUBROUTINE DATARD(SAVER,NTITLE,NLOOPY,IDOLD,E,SIGE,SWT,DOC,NTIME
     & ,TREAD,SIGT,RATE,SIGR,ID,INDATA,NDATA,NERROR)
C
C   ********************************************************************
C
C   NEEDS THE FOLLOWING SUBROUTINES AND FUNCTIONS:
C     ACTIV
C     EREAD
C     READIN
C     READX
C     AND COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION NTITLE(64),NTIME(50),TREAD(2,50),RATE(2,50)
      DIMENSION TEMP(2,50),SIGT(2,50),SIGR(2,50),E(50),SIGE(50)
      DIMENSION NSTAR(32),SWT(50),DOC(50)
      DIMENSION AT(4),IDOLD(50),ID(50),VRATE(50),VTEMP(50)
      NLOOPY = NLOOPY+1
C
C
      WRITE (NTTYO,110) NLOOPY
 110  FORMAT (12H FILE NUMBER,I3)
C
C  READ AND WRITE TITLE OF INPUT FILE
C
      READ (IN,120) NTITLE
 120  FORMAT (32A2/32A2)
      DO 130 I = 1,32
 130  NSTAR(I) = 2H**
      IF (NFILEB.GT.6) GO TO 131
      WRITE (NFILEB,140)NTITLE
      WRITE (NFILEB,90)SAVER
   90 FORMAT(/' CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM'
     & /' IS',F6.2)
      WRITE (NFILEB,30)NFILEA
   30 FORMAT(/' PUT PROCESSED DATA ON FILE',I3)
  131 IF (NPRINT.LT.3) RETURN
      WRITE (NPRINT,120) NSTAR
      WRITE (NPRINT,140)
 140  FORMAT (25H TITLE OF INPUT FILE IS ://)
      WRITE (NPRINT,120) NTITLE
      WRITE (NPRINT,150)
 150  FORMAT (//)
      WRITE (NPRINT,120) NSTAR
      WRITE (NPRINT,150)
      WRITE (NPRINT,90) SAVER
      WRITE (NPRINT,30) NFILEA
C
C   READ IN DATA
C
  151 IF (INDATA.GT.0)GO TO 85
C
C  GET THERMOCOUPLE COEFFICIENTS, READ IN DATA
C
      CALL THMCPL (AT)
      CALL READIN (NTIME,TREAD,SIGT,RATE,SIGR,ID,NDATA)
```

```
C
C  CALCULATE E ACT. AND ASSOCIATED SIGMA
C
      DO 170 I = 1,NDATA
      IDOLD(I) = ID(I)
      CALL ACTIV (E,SIGE,TEMP(1,I),TEMP(2,I),SIGT(1,I),SIGT(2,I)
     & ,TREAD(1,I),TREAD(2,I),SIGT(1,I),SIGT(2,I),VTEMP
     & ,RATE(1,I),RATE(2,I),SIGR(1,I),SIGR(2,I),VRATE
     & ,AT,NERROR)
      IF (NERROR.GT.0) RETURN
  170 CONTINUE
      RETURN
C
C  READ IN E VALUES
C
   85 IF (INDATA.EQ.1) GO TO 88
      WRITE (NTTYO,89)
   89 FORMAT (/' READ INITIAL AND FINAL SAMPLE WEIGHTS, FREE FORMAT')
      WINITL = READX (IW,NERROR,3,IN,NTTYO,IEOF)
      WFINAL = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SAMPWT = WINITL-WFINAL
      IF (NFILEB.GT.6) WRITE(NFILEB,81) WINITL,WFINAL
      IF (NPRINT.GE.3) WRITE(NPRINT,81) WINITL,WFINAL
   81 FORMAT(/' INITIAL AND FINAL WEIGHTS READ FROM INPUT FILE AS'
     & /2F15.3)
C
C  SET DIMENSION STATEMENT IN EREAD VIA NDATA
C
   88 NDATA = 50
      CALL EREAD (ID,E,SIGE,SWT,DOC,NDATA,INDATA,NERROR)
      IF (INDATA.EQ.1) GO TO 87
      DO 86 I = 1,NDATA
   86 DOC(I) = ((WINITL-SWT(I))/SAMPWT)*100.
   87 DO 180 I = 1,NDATA
  180 IDOLD(I)=ID(I)
C
C  OUTPUT WHATEVER WAS INPUT
C
      IF (NFILEB.GT.6) WRITE (NFILEB,100)
      IF (NPRINT.GE.3) WRITE (NPRINT,100)
  100 FORMAT(/' NUMBER',9X,'E',5X,'SIG E',4X,'D.O.C.',12X,'SWT'/)
      IF (NFILEB.GT.6) WRITE (NFILEB,210) (IDOLD(I),E(I),SIGE(I),DOC(I)
     & ,SWT(I),I=1,NDATA)
      IF (NPRINT.GE.3) WRITE (NPRINT,210) (IDOLD(I),E(I),SIGE(I),DOC(I)
     & ,SWT(I),I=1,NDATA)
  210 FORMAT(I7,3F10.2,F15.2)
      RETURN
      END
```

```
$ASSM
EREAD PROG
$FORT
$TRGT 16
C   ***************************************************************************
C
      SUBROUTINE EREAD(IDOLD,E,SIGE,SWT,DOC,NDATA,INDATA,NERROR)
C
C   ***************************************************************************
C
C      SUBROUTINE TO READ IN E/SIGE FILE FROM TGRUNF
C     NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C     AND FUNCTION READX
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(5))
C
      DIMENSION IDOLD(NDATA),E(NDATA),SIGE(NDATA),SWT(NDATA)
     &,DOC(NDATA)
C
C   NDATA IS INITIALISED NEAR DIMENSION STATEMENT IN MAIN PROGRAM
C
      NDO = NDATA
      NDATA = 0
C
C   READ IN E/SIG E FILE (5 RECORDS PER ACTIVATION ENERGY)
C
      DO 10 I = 1,NDO
      J = READX (IW,NERROR,3,IN,NTTYO,IEOF)
C
C   TEST FOR EOF
C
      IF (IEOF.GT.0)GO TO 20
      NDATA = NDATA+1
      IDOLD(NDATA) = I
      E(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SIGE(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      A = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      SWT(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
      IF (INDATA.EQ.1)DOC(NDATA) = READX (IW,NERROR,1,IN,NTTYO,IEOF)
C
C   SPACE OVER REST OF RECORDS FOR THIS ACTIVATION ENERGY
C
      DO 15 K = 1,4
   15 A = READX (IW,NERROR,3,IN,NTTYO,IEOF)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
$ASSM
FCTN PROG
$FORT
$TRGT 16
C
C     ****************************************************************************
C
      FUNCTION FCTN (X,I,J,JTERMS,NPTS)
C
C     ****************************************************************************
C
C  FUNCTION TO CALCULATE FITTED VALUES FOR REGRES SUBROUTINE
C
C
      DIMENSION X(NPTS),JTERMS(J)
      JEXP=JTERMS(J)
      FCTN=X(I)**JEXP
      RETURN
      END
```

```
$ASSM
MATINV PROG
$FORT
$TRGT 16
C     **************************************************************************
C
      SUBROUTINE MATINV (ARRAY,NORDER,DET)
C
C     **************************************************************************
C   SUBROUTINE FOR MATRIX INVERSION
C
      DOUBLE PRECISION ARRAY,AMAX,SAVE
      DIMENSION ARRAY(5,5),IK(5),JK(5)
      DET=1.
      DO 190 K=1,NORDER
C
C   FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX
C
          AMAX=0.
10        DO 30 I=K,NORDER
              DO 30 J=K,NORDER
              IF (DABS(AMAX)-DABS(ARRAY(I,J))) 20,20,30
20            AMAX=ARRAY(I,J)
              IK(K)=I
              JK(K)=J
30            CONTINUE
C
C   INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
C
          IF (AMAX) 50,40,50
40        DET=0.
          GO TO 260
50        I=IK(K)
          IF (I-K) 10,80,60
60        DO 70 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=ARRAY(I,J)
70            ARRAY(I,J)=-SAVE
80        J=JK(K)
          IF (J-K) 10,110,90
90        DO 100 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=ARRAY(I,J)
100           ARRAY(I,J)=-SAVE
C
C   ACCUMULATE ELEMENTS OF INVERSE MATRIX
C
110       DO 130 I=1,NORDER
              IF (I-K) 120,130,120
120           ARRAY(I,K)=-ARRAY(I,K)/AMAX
130           CONTINUE
          DO 160 I=1,NORDER
              DO 160 J=1,NORDER
              IF (I-K) 140,160,140
140           IF (J-K) 150,160,150
150           ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
```

```
160           CONTINUE
          DO 180 J=1,NORDER
              IF (J-K) 170,180,170
170           ARRAY(K,J)=ARRAY(K,J)/AMAX
180           CONTINUE
          ARRAY(K,K)=1./AMAX
190       DET=DET*AMAX
C
C  RESTORE ORDERING OF MATRIX
C
      DO 250 L=1,NORDER
          K=NORDER-L+1
          J=IK(K)
          IF (J-K) 220,220,200
200       DO 210 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=-ARRAY(I,J)
210           ARRAY(I,J)=SAVE
220       I=JK(K)
          IF (I-K) 250,250,230
230       DO 240 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=-ARRAY(I,J)
240           ARRAY(I,J)=SAVE
250       CONTINUE
260   RETURN
      END


$ASSM
PINV PROG
$FORT
$TRGT 16
C ************************************************************************
C
      FUNCTION PINV (Q)
C
C     ************************************************************************
C
C  FUNCTION TO CALCULATE NORMAL DISTRIBUTION QUANTILE FROM AREA,Q,UNDER
C  DISTRIBUTION CURVE.Q GIVEN BY FILLIBEN IN NBS TECHNICAL NOTE 552,P146
C
      PINV = 4.91*(EXP(0.14*ALOG(Q))-EXP(0.14*ALOG(1.-Q)))
C
      END
```

```
$ASSM
PLOT PROG
$FORT
$TRGT 16
C     ***************************************************************************
C
      SUBROUTINE PLOT (JM,IM,X,Y)
C
C     ***************************************************************************
C
C  PLOTTING SYMBOLS ARE A B C D E F IN THAT SEQUENCE.
C  THE SCALE-DEFINING POINTS, SAY, (0,0) AND (.10E+02, .10E-03)
C  SHOULD BE CONFINED TO THE FIRST CURVE ONLY.
C
C    JM  =  NUMBER OF CURVES
C    IM(JC)  =  NUMBER OF POINTS IN CURVE JC
C      (THIS NUMBER INCLUDES ANY LIMIT DEFINERS (FIRST CURVE ONLY))
C
C  NEEDS COMMON BLOCK /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
C
      DIMENSION X(100,6),Y(100,6),IM(6),P(51),YSC(6),XSC(6),S(6),IL(6)
C
      INTEGER R,C,RM1
C
      DATA BLANK,DASH,PLUS,S /1H ,1H-,1H+,1HA,1HB,1HC,1HD,1HE,1HF/
C
      YM = Y(1,1)
      Y0 = Y(1,1)
      XM = X(1,1)
      X0 = X(1,1)
      DO 40 J = 1,JM
C
C FIND LIMITS OF PLOT
C
      IMM = IM(J)
      DO 40 I = 1,IMM
      IF (Y(I,J).LT.YM) GO TO 10
      YM = Y(I,J)
10    IF (Y(I,J).GT.Y0) GO TO 20
      Y0 = Y(I,J)
20    IF (X(I,J).LT.XM) GO TO 30
      XM = X(I,J)
30    IF (X(I,J).GT.X0) GO TO 40
      X0 = X(I,J)
40    CONTINUE
      DYM = YM-Y0
      DXM = XM-X0
      YSC(1) = YM
      XSC(1) = X0
      DO 50 L = 2,6
50    YSC(L) = YSC(L-1)-DYM/5.
```

```
      DO 60 L = 2,6
60    XSC(L) = XSC(L-1)+DXM/5.
      DO 70 J = 1,JM
      IM1 = IM(J)-1
      DO 70 I = 1,IM1
C
C SORT IN DECREASING Y
C
      IP1 = I+1
      IMM = IM(J)
      DO 70 II = IP1,IMM
      IF (Y(II,J).LT.Y(I,J)) GO TO 70
      TEMP = Y(I,J)
      Y(I,J) = Y(II,J)
      Y(II,J) = TEMP
      TEMP = X(I,J)
C
C KEEP X AND Y PAIRED
C
      X(I,J) = X(II,J)
      X(II,J) = TEMP
70    CONTINUE
      R = 0
C
C START PLOT
C
      L = 0
      DO 80 J = 1,JM
80    IL(J) = 0
90    R = R+1
      RM1 = R-1
      AR1 = (RM1-.5)/50.
      YQ1 = YM-AR1*DYM
      AR2 = (RM1+.5)/50.
      YQ2 = YM-AR2*DYM
      DO 100 C = 1,51
100   P(C) = BLANK
C
C BLANK THE LINE
C
      J = JM
110   I = IL(J)
120   I = I+1
      IF (I.GT.IM(J)) GO TO 130
      IF (Y(I,J).GT.YQ1) GO TO 120
      IF (Y(I,J).LT.YQ2) GO TO 130
      C = 50.0*((X(I,J)-X0)/DXM)+1.5
      P(C) = S(J)
      GO TO 120
130   IL(J) = I-1
      J = J-1
      IF (J) 140,140,110
140   IF (MOD(RM1,5).EQ.0) GO TO 160
      WRITE (NPRINT,150) PLUS,P
      GO TO 190
150   FORMAT (1H ,14X,A1,51A1)
```

```
160   IF (MOD(RM1,10).EQ.0) GO TO 170
      WRITE (NPRINT,150) DASH,P
      GO TO 190
170   L = L+1
      WRITE (NPRINT,180) YSC(L),DASH,P
C
C PRINT ROW R
C
180   FORMAT (1H ,E13.5,1X,A1,51A1)
190   IF (R.LT.51) GO TO 90
      DO 210 C = 1,51
C
C FORM X AXIS
C
      ICM1 = C-1
      IF (MOD(ICM1,10).NE.0) GO TO 200
      P(C) = PLUS
      GO TO 210
200   P(C) = DASH
210   CONTINUE
      WRITE (NPRINT,220) P
C
C PRINT X AXIS
C
220   FORMAT (1H ,15X,51A1)
      WRITE (NPRINT,230) XSC
230   FORMAT (11X,6E10.4)
      RETURN
      END
```

```
$ASSM
PNORM PROG
$FORT
$TRGT 16
C       ******************************************************************
C
      SUBROUTINE PNORM (X,NDATA)
C
C     ******************************************************************
C
C  SUBROUTINE TO CALCULATE NORMAL DEVIATE FROM AREA OF
C  NORMAL PROBABILITY DISTRIBUTION
C  SEE OMNITAB 2 MANUAL,NBS TECHNICAL NOTE 552,(1971)P146.
C
      DIMENSION X(NDATA)
C
      PI = 3.1415927
      FDATA = NDATA
      DO 10 I = 1,NDATA
      FI = I
      XI = (FI-PI/8.)/(FDATA+1-PI/4.)
      X(I) = PINV(XI)
10    CONTINUE
C
C  CHANGE XI FOR FIRST AND LAST ITEMS IF NDATA.LE.10
C
      IF (NDATA.GT.10) RETURN
      XI = (3.-1.)/(3.*FI+1.)
      X(1) = PINV(XI)
      X(NDATA) = -X(1)
      RETURN
      END
```

```
$ASSM
PRIMES PROG
$FORT
$TRGT 16
C   ************************************************************
C
      SUBROUTINE PRIMES(NLOOPY,NDATA,NLOOP,SAVER,SAVE,INDATA)
C
C   *****************************************************************
C
C   SUBROUTINE TO INITIALISE TG ACTIVATION ENERGY ASSESSMENT
C     PROGRAM
C
C   NEEDS SUBROUTINE READX
C     AND COMMON BLOCKS /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C       AND /REGRS/ M(5),R(5),YFIT(250)
C
      COMMON DUMP(275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
      NLOOPY = 0
      NDATA = 50
C
C    INITIALISE FILES, ETC.
C
      NTTYI = 4
      NTTYO = 5
      IN = 7
      NPRINT = 6
      NFILEA = 8
      NFILEB = 9
      DO 5 I = 1,5
    5 M(I) = I
C
C
      WRITE (NTTYO,10) IN
   10 FORMAT (/'  TGACTE PROGRAM'
     & /'READ INPUT DATA FROM FILE',I3)
      WRITE (NTTYO,20) NPRINT
20    FORMAT (5H FILE,I4,14H IS FOR OUTPUT)
      WRITE (NTTYO,30) NFILEA
30    FORMAT (27H PUT PROCESSED DATA ON FILE,I4)
      WRITE (NTTYO,40)
40    FORMAT (' GIVE NUMBER OF FILES TO EXAMINE (IN FREE FORMAT)')
      NLOOP = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      WRITE (NTTYO,60) NLOOP
   60 FORMAT (' EXAMINE',I4,' FILES')
      WRITE (NTTYO,70)
70    FORMAT (43H GIVE CHANCE OF REJECTING EXTREME VALUE IF /
     & 31H FLUCTUATIONS ARE PURELY RANDOM/27H (0.01 TO 0.40,  ZERO=0.10)
     &/12H FREE FORMAT)
      SAVER = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      IF (SAVER.LT.1.E-05) SAVER = 0.10
      WRITE (NTTYO,90) SAVER
90    FORMAT (56H CHANCE OF WRONGLY REJECTING RANDOMLY PLACED EXTREMUM I
     &S,F6.2)
      SAVE = 1.-SAVER
      WRITE (NTTYO,80)
   80 FORMAT (/' GIVE 0 FOR RATE DATA INPUT'
     &/'        1 FOR E/SIG E INPUT (OUTPUT OF TGRUNF)'
     &/'             WITH DEGREES OF CONVERSION IN FILE'
     &/'        2 FOR E/SIG E INPUT WITH D.O.C. TO BE CALCULATED')
      INDATA = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
100   CONTINUE
      RETURN
      END
```

```
$ASSM
PROBP2 PROG
$FORT
$TRGT 16
C      ************************************************************************
C
      SUBROUTINE PROBP2 (DE,Z,E,SIGE,EAVE,ID,NDATA)
C
C     ************************************************************************
C
C  SUBROUTINE TO CALCULATE DE QUANTITIES, ORDER THEM AND
C  ASSIGN ASSOCIATED Z VALUES FROM ORDER STATISTICS ASSUMING
C  A NORMAL DISTRIBUTION..
C
C    NEEDS SUBROUTINES:
C    PNORM
C    SORT4
C
      DIMENSION DE(NDATA),Z(NDATA),E(NDATA),SIGE(NDATA),ID(NDATA)
C
      DO 10 I = 1,NDATA
10    DE(I) = (E(I)-EAVE)/SIGE(I)
C
C  ORDER ON DE
C  MAKE ID INTEGER IN SORT4 SUBROUTINE
C
      CALL SORT4 (DE,ID,NDATA)
C
C  CALCULATE NORMAL DISTRIBUTION QUANTILES
C
      CALL PNORM (Z,NDATA)
      RETURN
      END


$ASSM
PNORM PROG
$FORT
$TRGT 16
C      ************************************************************************
C
      SUBROUTINE PNORM (X,NDATA)
C
C     ************************************************************************
C
C  SUBROUTINE TO CALCULATE NORMAL DEVIATE FROM AREA OF
C  NORMAL PROBABILITY DISTRIBUTION
C  SEE OMNITAB 2 MANUAL,NBS TECHNICAL NOTE 552,(1971)P146.
C
      DIMENSION X(NDATA)
C
      PI = 3.1415927
      FDATA = NDATA
      DO 10 I = 1,NDATA
      FI = I
      XI = (FI-PI/8.)/(FDATA+1-PI/4.)
      X(I) = PINV(XI)
10    CONTINUE
C
C  CHANGE XI FOR FIRST AND LAST ITEMS IF NDATA.LE.10
C
      IF (NDATA.GT.10) RETURN
      XI = (3.-1.)/(3.*FI+1.)
      X(1) = PINV(XI)
      X(NDATA) = -X(1)
      RETURN
      END
```

```
$ASSM
READIN PROG
$FORT
$TRGT 16
C   ****************************************************************
C
      SUBROUTINE READIN(NTIME,TREAD,SIGT,RATE,SIGR,ID,NDATA)
C
C   ****************************************************************
C
C   SUBROUTINE TO READ RAW DATA
C   NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION NTIME(50),TREAD(2,50),SIGT(2,50),RATE(2,50),SIGR(2,50),
     & ID(50)
C
C
      IF (NFILEB.GT.6) WRITE (NFILEB,10)
      IF (NPRINT.GE.3) WRITE (NPRINT,10)
10    FORMAT (46H       TIME        THCPL   SIG THCPL          RATE,
     & 12H    SIG RATE/)
      NERROR = 0
      NDATA = 0
      DO 50 I = 1,50
      NTIME(I) = READX(IW,NERROR,3,IN,NTTYO,IEOF)
      TREAD(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGT(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      RATE(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGR(1,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      TREAD(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGT(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      RATE(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      SIGR(2,I) = READX(IW,NERROR,1,IN,NTTYO,IEOF)
      IF (NFILEB.GT.6) WRITE (NFILEB,30)NTIME(I),(TREAD(J,I),SIGT(J,I)
     &,RATE(J,I),SIGR(J,I),J = 1,2)
      IF (NPRINT.GE.3) WRITE (NPRINT,30) NTIME(I),(TREAD(J,I),SIGT(J,I)
     &,RATE(J,I),SIGR(J,I),J = 1,2)
30    FORMAT (I10,4E12.4/10X4E12.4)
C
C   PUT THERMOCOUPLE READING IN MILLIVOLTS
C
      DO 40 J = 1,2
      TREAD(J,I) = TREAD(J,I)*1000.
40    SIGT(J,I) = SIGT(J,I)*1000.
      ID(I) = I
      NDATA = NDATA+1
50    CONTINUE
      IF (NFILEB.GT.6) WRITE (NFILEB,70)NDATA
60    IF (NPRINT.GE.3) WRITE (NPRINT,70) NDATA
70    FORMAT (I6,20H DATA POINTS READ IN/)
      RETURN
      END
```

```
$ASSM
READX PROG
$FORT
      FUNCTION READX(IWORD,N,NP,NTTYI,NTTYO,IEOF)
C
C $$$$$$$$$  VERSION OF MAR 9,1977 $$$$$$$$$
C
C  FUNCTION READX USUALLY RETURNS THE NEXT DECIPHERIBLE NUMBER ON THE
C    INPUT FILE =NTTYI= AS THE VALUE OF THE VARIABLE READX.
C    ANY OUPUT, SUCH AS ERROR MESSAGES OR THE PROMPT CHARACTER,
C    IS ON UNIT NTTYO.
C
C  IW= ALPHANUMERIC INFORMATION OF THE FIRST 1 OR 2 ILLEGAL NON-NUMERICAL
C    INFORMATION ON INPUT FILE NTTYI.
C  N=0 MEANS NUMERICAL INFORMATION SUCCESSFULLY PROCESSED.
C  N=1 CHECK IW WORD FOR NON-NUMERICAL INFORMATION.
C        (READX=1. IN THIS CASE).
C  NP=1  SEND ! AS A PROMPT CHARACTER WHEN NEW RECORD REQUIRED.
C  NP=2  READ A NEW RECORD.
C  NP=3  DO BOTH THE ABOVE.
C  NTTYI=  INPUT UNIT
C  NTTYO=  OUTPUT UNIT FOR PROMPT CHARACTER AND ERROR MESSAGES.
C  IEOF=0   END OF FILE NOT YET REACHED ON UNIT NTTYI.
C  IEOF=1  END OF FILE FOUND ON NTTYI.
C
C  ILLEGAL INFORMATION:
C
C  IF FIRST CHARACTER IS NOT A BLANK, COMMA, POINT, MINUS, OR
C    A NUMBER, THE FIRST 2 ILLEGAL CHARACTERS ARE RETURNED IN =IW=.
C
C  LEGAL INFORMATION*
C
C  LEADING BLANKS ARE IGNORED.
C  LEADING COMMAS ARE IGNORED.
C  DECIMAL POINTS AND EXPONENTIAL TYPE NUMBERS ARE TREATED CORRCECTLY.
C  $ ON THE INPUT FILE IS PRESUMED TO PRECEED AN IN-LINE COMMENT AND
C    CAUSES READING OF A NEW RECORD.
C  NUMBERS ARE TERMINATED BY A BLANK, COMMA, OR BY THE END OF A RECORD.
C
C
C  READX WAS ORIGINALLY WRITTEN BY FRANK MCCRACKIN (NBS) AND WAS MODIFIED
C    BY FRED MOPSIK AND BY BRIAN DICKENS.  (JAN 1977)
C
C
C
      INTEGER BRANCH
      INTEGER*2 CD,DEC,IW2,N1,J,IEXP,IEXP1
      LOGICAL ISIGNE
      DIMENSION CD(80),DEC(10),TEN(10),TENTEN(10)
      DATA DEC/'0','1','2','3','4','5','6','7','8','9'/
      DATA TEN/1.,1.E1,1.E2,1.E3,1.E4,1.E5,1.E6,1.E7,1.E8,1.E9/
      DATA TENTEN/1.,1.E10,1.E20,1.E30,1.E40,1.E50,1.E60,3*1.E70/
      DATA N1/81/
      IF (NP.EQ.2.OR.NP.EQ.3) N1 = 81
      IEOF = 0
```

```
      N = 0
      SIGN = 1.
      X = 0.
      Y = 1.
      YJ = 0.
C
C   FIND FIRST CHARACTER
C
15    IF (N1.GT.80) GO TO 1000
      IF (CD(N1).EQ.'$') GO TO 1000
      ASSIGN 16 TO BRANCH
      GO TO 2000
16    GO TO (40,100,30,20,20,300),IP
20    N1 = N1+1
      GO TO 15
C   SET SIGN
30    SIGN = -1.
      N1 = N1+1
      GO TO 15
C
C   CALCULATE NUMBER
C
40    X = J
50    N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 50
      ASSIGN 70 TO BRANCH
      GO TO 2000
70    GO TO (75,100,205,200,270,270),IP
75    X = X*10.+J
      GO TO 50
C
C   CALCULATE FRACTION
C
100   CONTINUE
110   N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 110
      ASSIGN 125 TO BRANCH
      GO TO 2000
125   GO TO (130,3055,205,200,270,270),IP
130   YJ = 10.*YJ+J
      Y = 10.*Y
      GO TO 110
C
C   CALCULATE EXPONENT IEXP
C
200   ISIGNE = .FALSE.
      IEXP = 1
      IEXP1 = 1
      GO TO 210
205   ISIGNE = .TRUE.
      IEXP = 1
      IEXP1 = 1
210   N1 = N1+1
      IF (N1.GT.80) GO TO 250
```

```
      ASSIGN 220 TO BRANCH
      GO TO 2000
220   GO TO (230,250,250,250,250,250),IP
230   IEXP1 = IEXP
      IEXP = J+1
      GO TO 210
250   IF (ISIGNE) GO TO 260
      READX = SIGN*(X+YJ/Y)*TEN(IEXP)*TENTEN(IEXP1)
      N = 0
      RETURN
260   READX = SIGN*(X+YJ/Y)/(TEN(IEXP)*TENTEN(IEXP1))
      N = 0
      RETURN
270   READX = SIGN*(X+YJ/Y)
      N = 0
      RETURN
C
C   RETURN ALPHA ARG
C
300   IW2 = CD(N1)
      N1 = N1+1
      IF (N1.GT.80) GO TO 302
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',' .OR. CD(N1).EQ.'$') GO TO 302
      IW2 = IW2-X'20'+CD(N1)/X'100'
      N1 = N1+1
302   IWORD = IW2
      N = 1
      READX = 1.
      RETURN
3055  WRITE (NTTYO,3056)N1
3056  FORMAT(' 2d . AT COL.',I3)
      GO TO 270
1000  N1 = 1
      IF (NP.EQ.1.OR.NP.EQ.3) WRITE (NTTYO,1020)
1020  FORMAT(' !')
      READ (NTTYI,1010,END = 9000) CD
1010  FORMAT(80A1)
      GO TO 15
C
C   CHARACTER INTERP. ROUTINE
C
2000  DO 2010 J = 1,10
      IF (CD(N1).EQ.DEC(J)) GO TO 2020
2010  CONTINUE
      IP = 6
      IF (CD(N1).EQ.'.') IP = 2
      IF (CD(N1).EQ.'-') IP = 3
      IF (CD(N1).EQ.'+') IP = 4
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',') IP = 5
      GO TO 2030
2020  IP = 1
      J = J-1
2030  GO TO BRANCH
 9000 IEOF = 1
      RETURN
      END
```

```
$ASSM
REGRES PROG
$FORT
$TRGT 16
C
C     ************************************************************************
C
      SUBROUTINE REGRES (X,Y,SIGMAY,NPTS,NTERMS,M,MODE,YFIT,AO,A,SIGMAO,
     2 SIGMAA,R,RMUL,CHISQR,FTEST)
C
C     ************************************************************************
C
C  NEEDS THE FOLLOWING FUNCTION AND SUBROUTINE:
C    FCTN
C    MATINV
C
C
      DOUBLE PRECISION ARRAY,SUM,YMEAN,SIGMA,CHISQ,XMEAN,SIGMAX
      DIMENSION X(NPTS),Y(NPTS),SIGMAY(NPTS),M(5),YFIT(NPTS)
     &,SIGMAA(5),R(5),A(5)
      DIMENSION WEIGHT(250),XMEAN(5),SIGMAX(5),ARRAY(5,5)
C
C  INITIALIZE SUMS AND ARRAYS
C
      SUM=0.
      YMEAN=0.
      SIGMA=0.
      CHISQ=0.
      RMUL=0.
      DO 10 I=1,NPTS
10        YFIT(I)=0.
      IF (NTERMS.LE.0) GO TO 30
      DO 20 J=1,NTERMS
          XMEAN(J)=0.
          SIGMAX(J)=0.
          R(J)=0.
          A(J)=0.
          SIGMAA(J)=0.
          DO 20 K=1,NTERMS
20        ARRAY(J,K)=0.
30    CONTINUE
C
C  ACCUMULATE WEIGHTED SUMS
C
      DO 110 I=1,NPTS
          IF (MODE) 40,70,80
40        IF (Y(I)) 60,70,50
50        WEIGHT(I)=1./Y(I)
          GO TO 90
60        WEIGHT(I)=1./(-Y(I))
          GO TO 90
70        WEIGHT(I)=1.
          GO TO 90
80        WEIGHT(I)=1./SIGMAY(I)**2
90        SUM=SUM+WEIGHT(I)
          YMEAN=YMEAN+WEIGHT(I)*Y(I)
```

```
         IF (NTERMS.LE.0) GO TO 110
         DO 100 J=1,NTERMS
100         XMEAN(J)=XMEAN(J)+WEIGHT(I)*FCTN(X,I,J,M,NPTS)
110      CONTINUE
      YMEAN=YMEAN/SUM
      IF (NTERMS.LE.0) GO TO 130
      DO 120 J=1,NTERMS
120      XMEAN(J)=XMEAN(J)/SUM
130   CONTINUE
      FNPTS=NPTS
      WMEAN=SUM/FNPTS
      DO 140 I=1,NPTS
140      WEIGHT(I)=WEIGHT(I)/WMEAN
C
C  ACCUMULATE MATRICES R AND ARRAY
C
      DO 150 I=1,NPTS
      SIGMA=SIGMA+WEIGHT(I)*(Y(I)-YMEAN)**2
      IF (NTERMS.LE.0) GO TO 160
      DO 150 J=1,NTERMS
      SIGMAX(J)=SIGMAX(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))**2
      R(J)=R(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))*(Y(I)-YMEAN)
         DO 150 K=1,J
      ARRAY(J,K)=ARRAY(J,K)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))
     2 *(FCTN(X,I,K,M,NPTS)-XMEAN(K))
  150 CONTINUE
160   CONTINUE
      FREE1=NPTS-1
      SIGMA=DSQRT(SIGMA/FREE1)
      IF (NTERMS.LE.0) GO TO 180
      DO 170 J=1,NTERMS
         SIGMAX(J)=DSQRT(SIGMAX(J)/FREE1)
         R(J)=R(J)/(FREE1*SIGMAX(J)*SIGMA)
         DO 170 K=1,J
         ARRAY(J,K)=ARRAY(J,K)/(FREE1*SIGMAX(J)*SIGMAX(K))
170      ARRAY(K,J)=ARRAY(J,K)
180   CONTINUE
C
C  INVERT SYMMETRIC MATRIX
C
      IF (NTERMS.LE.0) GO TO 200
      CALL MATINV (ARRAY,NTERMS,DET)
      IF (DET) 200,190,200
190   A0=0.
      SIGMA0=0.
      RMUL=0.
      CHISQR=0.
      FTEST=0.
      GO TO 330
C
C  CALCULATE COEFFICIENTS, FIT, AND CHI SQUARE
C
200   A0=YMEAN
      IF (NTERMS.LE.0) GO TO 230
      DO 220 J=1,NTERMS
         DO 210 K=1,NTERMS
```

```
210          A(J)=A(J)+R(K)*ARRAY(J,K)
          A(J)=A(J)*SIGMA/SIGMAX(J)
          A0=A0-A(J)*XMEAN(J)
          DO 220 I=1,NPTS
220       YFIT(I)=YFIT(I)+A(J)*FCTN(X,I,J,M,NPTS)
230    CONTINUE
       DO 240 I=1,NPTS
          YFIT(I)=YFIT(I)+A0
240       CHISQ=CHISQ+WEIGHT(I)*(Y(I)-YFIT(I))**2
       FREEN=NPTS-NTERMS-1
       CHISQR=CHISQ*WMEAN/FREEN
C
C  CALCULATE  UNCERTAINTIES
C
       IF (MODE) 250,260,250
250    VARNCE=1./WMEAN
       GO TO 270
260    VARNCE=CHISQR
  270  IF (NTERMS.LE.0) GO TO 290
       DO 280 J=1,NTERMS
          SIGMAA(J)=ARRAY(J,J)*VARNCE/(FREE1*SIGMAX(J)**2)
          SIGMAA(J)=SQRT(SIGMAA(J))
280       RMUL=RMUL+A(J)*R(J)*SIGMAX(J)/SIGMA
       FREEJ=NTERMS
       FTEST=(RMUL/FREEJ)/((1.-RMUL)/FREEN)
       RMUL=SQRT(RMUL)
       GO TO 300
290    CONTINUE
       FREEJ=0.
       FTEST=0.
       RMUL=0.
300    SIGMA0=VARNCE/FNPTS
       IF (NTERMS.LE.0) GO TO 320
       DO 310 J=1,NTERMS
          DO 310 K=1,NTERMS
310       SIGMA0=SIGMA0+VARNCE*XMEAN(J)*XMEAN(K)*ARRAY(J,K)/(FREE1*SIGMA
     2X(J)*SIGMAX(K))
320    SIGMA0=SQRT(SIGMA0)
330    RETURN
       END
```

```
$ASSM
SORT4 PROG
$FORT
$TRGT 16
C       ************************************************************************
C
       SUBROUTINE SORT4 (A,K,N)
C
C       ************************************************************************
C
C
C     THIS SUBROUTINE SORTS A 1-DIMENSIONAL FLOATING POINT ARRAY
C     (A) AND AN ARRAY (K) INTO AN ORDER BASED ON THE ELEMENTS IN
C     IN ARRAY (A).
C
       DIMENSION A(N),K(N)
C
       IF (N.LE.1) RETURN
       DO 20 I = 2,N
       IM1 = I-1
       DO 10 J = 1,IM1
       JJ = I-J
       IF (A(JJ).LE.A(JJ+1)) GO TO 20
       X = A(JJ)
       A(JJ) = A(JJ+1)
       A(JJ+1) = X
       L = K(JJ)
       K(JJ) = K(JJ+1)
10     K(JJ+1) = L
20     CONTINUE
       RETURN
       END
```

```
$ASSM
SORT5 PROG
$FORT
$TRGT 16
C       ***********************************************************************
C
      SUBROUTINE SORT5 (K,B,C,N)
C
C     ***********************************************************************
C
C
C    THIS SUBROUTINE SORTS A 1-DIMENSIONAL ARRAY
C    (K) AND THE ARRAYS (B) AND (C) INTO AN ORDER BASED
C    ON THE ELEMENTS IN ARRAY (K).
C
      DIMENSION K(N),B(N),C(N)
C
      IF (N.LE.1) RETURN
      DO 20 I = 2,N
      IM1 = I-1
      DO 10 J = 1,IM1
      JJ = I-J
      IF (K(JJ).LE.K(JJ+1)) GO TO 20
      L = K(JJ)
      K(JJ) = K(JJ+1)
      K(JJ+1) = L
      X = B(JJ)
      B(JJ) = B(JJ+1)
      B(JJ+1) = X
      X = C(JJ)
      C(JJ) = C(JJ+1)
      C(JJ+1) = X
10    CONTINUE
20    CONTINUE
      RETURN
      END
```

```
$ASSM
THMPLS PROG
$FORT
$TRGT 16
C  ****************************************************************************
C
      SUBROUTINE THMCPL (AT)
C
C     ****************************************************************************
C
C  SUBROUTINE TO READ THERMOCOUPLE TYPE AND PROVIDE COEFFICIENTS FOR
C  CONVERSION OF THERMOCOUPLE READINGS IN MICROVOLTS TO DEGREES CELCIUS
C  NEEDS COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
C
      DIMENSION AT(4)
C
C
      WRITE (NTTYO,10)
10    FORMAT (/' GIVE THERMOCOUPLE TYPE, K OR E')
20    READ (NTTYI,30) NTHCP
30    FORMAT (A1)
      ITHCP = 4
      IF (NTHCP.EQ.'K')ITHCP = 1
      IF (NTHCP.EQ.'E')ITHCP = 2
      GO TO (40,60,80,90), ITHCP
C
C  TYPE K THERMOCOUPLE,VALUES FROM TABLE A7.2.3,NBS MONO.125
C
      IF (NFILEB.GT.6) WRITE (NFILEB,50)
40    IF (NPRINT.GE.3) WRITE (NPRINT,50)
50    FORMAT (/52H TYPE K THERMOCOUPLE = NI-CR/NI-AL,EXPRESSIONS GOOD /
     &,46H TO 400 DEGREES C.(TABLE A7,NBS MONO.125,1974))
      AT(1) = 2.4383248E-02
      AT(2) = 9.7830251E-09
      AT(3) = 3.6276965E-12
      AT(4) = -2.5756438E-16
      GO TO 110
C
C  TYPE E THERMOCOUPLE
C
      IF (NFILEB.GT.6) WRITE (NFILEB,70)
60    IF (NPRINT.GE.3) WRITE (NPRINT,70)
70    FORMAT (/52H TYPE E THERMOCOUPLE = NI-CR/CU-NI,EXPRESSIONS GOOD /
     &,58HFROM -0.05 TO +0.04, 0 TO 400 DEGREES C.(TABLE A5.2.3,NBS /,
     & 19HMONOGRAPH 125,1974))
      AT(1) = 1.7022525E-02
      AT(2) = -2.2097248E-07
      AT(3) = 5.4809314E-12
      AT(4) = -5.7669892E-17
      GO TO 110
C
C  DUMMY LOCATION
C
80    IF (NFILEB.GT.6) WRITE (NFILEB,100)
      IF (NPRINT.GE.3) WRITE (NPRINT,100)
      GO TO 20
C
C  ERROR ROUTE
C
   90 WRITE (NTTYO,100)
100   FORMAT (/40H WRONG SPECIFICATION FOR THERMOCOUPLE,TRY AGAIN)
      GO TO 20
110   RETURN
      END
```

```
$ASSM
WTAVER PROG
$FORT
$TRGT 16
C *****************************************************************
C
      SUBROUTINE WTAVER (E,SIGE,NDATA,WTEAVE,SIGWTE,EAVE,ESIG)
C
C *****************************************************************
C
C    ***    NOT THE WINSORISING VERSION    ***
C
C  SUBROUTINE TO CALCULATE AVERAGE E (WTEAVE) AND STANDARD DEVIATION (SIGWTE)
C    FROM INPUT LIST E AND SIGE CONTAINING  = NDATA = VALUES
C  ALSO CALCULATE UNWEIGHTED AVERAGE AND 95% CONFIDENCE RANGES
C
C  NEEDS THE FOLLOWING COMMON BLOCKS AND SUBROUTINES:
C    COMMON /IO/ NTTYI,NTTYO,IN,NPRINT,NFILEA,NFILEB
C    COMMON /REGRS/ M(5),R(5),YFIT(250)
C
C  NEEDS SUBROUTINES:
C    AVSDN
C    CHISQ
C    REGRES
C
      COMMON DUMP (275)
      EQUIVALENCE (NTTYI,DUMP(1)),(NTTYO,DUMP(2)),(IN,DUMP(3))
     & ,(NPRINT,DUMP(4)),(NFILEA,DUMP(5)),(NFILEB,DUMP(6))
      DIMENSION M(5),R(5),YFIT(250)
      EQUIVALENCE (M(1),DUMP(10)),(R(1),DUMP(15)),(YFIT(1),DUMP(20))
C
      DIMENSION E(NDATA),SIGE(NDATA),A(5),SIGA(5)
C
C  CALCULATE AVERAGE UNWEIGHTED E
C
      WTEAVE = 0.
      SIGWTE = 1.E-09
      EAVE = 0.
      ESIG = 1.E-09
      SIGAVE = 1.E-09
      FREE = 0.
      CHIWTE = 0.
      CHIUNW = 0.
      IF (NDATA.LE.0)RETURN
      IF (NDATA.GT.1)GO TO 200
      WTEAVE = E(1)
      SIGWTE = SIGE(1)
      SIGAVE = SIGE(1)
      EAVE = E(1)
      ESIG = SIGE(1)
      GO TO 300
  200 CONTINUE
      CALL AVSDN (EAVE,ESIG,E,NDATA)
      CALL CHISQ (E,SIGE,CHIUNW,NDATA)
C
C  CALCULATE AVERAGE SIGE
C
      CALL AVSDN (SIGAVE,SIGSIG,SIGE,NDATA)
C
C  CALCULATE WEIGHTED AVERAGE OF E
C
      CALL REGRES (E,E,SIGE,NDATA,0,M,1,YFIT,A0,A,SIGA0,SIGA
     &,R,RMUL,CHIWTE,FTEST)
      WTEAVE = A0
C
```

```
C  CALCULATE SIGMA FOR WEIGHTED E AVERAGE
C
      SIGWTE = 0.
      DO 10 I = 1,NDATA
   10 SIGWTE = SIGWTE+1./SIGE(I)**2
      SIGWTE = SQRT(1./SIGWTE)
      FDATA = NDATA
      FREE = NDATA-1
      ESIG = ESIG/SQRT(FDATA)
C
C  CALCULATE 95% CONFIDENCE RANGE
C
  300 T = 1.96*SIGWTE
      WTEHI = WTEAVE+T
      WTELO = WTEAVE-T
      T = 1.96*ESIG
      UNWEHI = EAVE+T
      UNWELO = EAVE-T
C
C  OUTPUT AVERAGES
C
      IF (NPRINT.LT.3) GO TO 100
      WRITE (NPRINT,30) WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
30    FORMAT (/' WEIGHTED MEAN E IS ',F7.3,' + OR - ',F7.3/
     & ' REDUCED CHISQ FOR E CALCULATION IS',F7.3,' (SHOULD BE 1.)'/
     & ' UNWEIGHTED MEAN OF SIGMAS IS ',F7.3,/
     &,' CALCULATED OVER ',F5.0,' DEGREES OF FREEDOM')
C
C  OUTPUT CONFIDENCE LIMITS
C
      WRITE (NPRINT,40) WTELO,WTEHI
40    FORMAT (/' ASSUMING ALL POSSIBLE SAMPLE MEANS ARE NORMALLY'/
     & ' DISTRIBUTED, TRUE WEIGHTED MEAN LIES WITHIN RANGE',F7.3,' TO',
     & F7.3/' FOR 95% CONFIDENCE LEVEL')
      WRITE (NPRINT,50) EAVE,ESIG,UNWELO,UNWEHI,CHIUNW
50    FORMAT (/' UNWEIGHTED ESTIMATE OF MEAN E AND UNBIASED ESTIMATE '
     & /' OF ITS SIGMA ARE',F7.3,' AND',F7.3
     & /' THIS GIVES A CONFIDENCE RANGE OF',F7.3,' TO',F7.3
     &/' CHI-SQUARE FOR UNWEIGHTED E IS',F7.2)
  100 IF (NFILEB.LE.6)RETURN
      WRITE (NFILEB,30)WTEAVE,SIGWTE,CHIWTE,SIGAVE,FREE
      WRITE (NFILEB,40)WTELO,WTEHI
      WRITE (NFILEB,50)EAVE,ESIG,UNWELO,UNWEHI,CHIUNW
      RETURN
      END
```

File LOTGDE, used as DO LOTGDE,D0 to assign files needed to make overlays in TGDEPG program. This file also starts the program LIBLDR which actually makes the overlays.

```
AC TGDEPG,1D0
AS 362
AC TGDEL5,5D0
AC TGDENB,7D0
AC OVERLB,8D0
AC FORLIB,FD0
RUN LIBLDR,D0
CL
AC TGDE01,1D0
AC PRIMSB,7D0
AC DATARB,8D0
AC THMCSB,9D0
AC READIB,AD0
AC ACTIVB,BD0
AC READXB,CD0
AC EREADB,DD0
AC TDEGBB,ED0
ST
CL
AC TGDE02,1D0
AC PINVB,7D0
AC PROB2B,8D0
AC SORT4B,9D0
AC SORT5B,AD0
AC WTAVEB,BD0
AC REGRSB,CD0
AC FCTNB,DD0
AC MATINB,ED0
ST
AC PNORMB,7D0
AC AVSDNB,8D0
AC CHISQB,9D0
AC PLOTB,AD0
AS B20
AS C20
AS D20
AS E20
ST
CL
AS 120
AS 320
AS 520
AS 720
AS 820
AS 920
AS A20
AS F20
AS 220
TR
```

File TGDEL5, which contains the appropriate responses to the program LIBLDR to make the overlays in the TGDEPG program.

```
OUT 1 TGDEPG
BI 3576
BC 044C
LO 7
LI 8
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 TGDE01
OV
LI 7
LI 8
LI 9
LI A
LI B
LI C
LI D
LI E
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
OUT 1 TGDE02
OV
LI 7
LI 8
LI 9
LI A
LI B
LI C
LI D
LI E
END
LI 7
LI 8
LI 9
LI A
RW F
ED F
MAP
MAP 3
XOUT
WF 1
END
```

Memory map for the base segment of the TGDEPG program.

```
REL PROGS:
  3576 TGDEN     80EE OVERLY    824C IFETCH    82BE .R
  82FA .A        83A0 SQRT      84A8 ALOG      85B6 EXP
  86BC AINT      8734 ABS       874A .W        8798 .COMP
  87C4 $6        880E .RARG     883A $8        8864 .5
  886C .ZERO     8874 .H        88C6 .S        88FC .P
  897E .Q        8A2E .D        8A76 .MES      8AEC .U
  8B1C .V        8B2A @R        8B50 @Z        8B76 @H
  9DC0

ABS PROGS:
NONE

ENTRY-POINTS:
  8112 OVERLY    824E IFETCH    82C0 .R        82FC .A
  83A2 SQRT      84AA ALOG      85B8 EXP       86BE AINT
  8736 ABS       874C .W        879A .COMP     87C6 $6
  8810 .RARG     883C $8        8868 .5        8870 .ZERO
  8876 .H        88C8 .S        88FE .P        8980 .Q
  8A30 .O        8A78 .MES      8AFA .U        8B22 .V
  8B2C @R        8B52 @Z        8BC4 @H

COMMON-BLOCKS:
  FBB2 //

UNDEFINED:
   PRIMES    DATARD    PROBP2    SORT5    PINV    WTAVER
   REGRES    PLOT
```

Memory map for the initialization segment of the TGDEPG program.

```
REL PROGS:
   3576 TGDEN     80EE OVERLY    824C IFETCH    82BE .R
   82FA .A        83A0 SQRT      84A8 ALOG      85B6 EXP
   86BC AINT      8734 ABS       874A .W        8798 .COMP
   87C4 $6        880E .RARG     883A $8        8864 .5
   886C .ZERO     8874 .H        88C6 .S        88FC .P
   897E .Q        8A2E .O        8A76 .MES      8AEC .U
   8B1C .V        8B2A @R        8B50 @Z        8B76 @H
   9F28 PRIMES    A412 DATARD    B316 THMPLS    B710 READIN
   BD9E ACTIV     C5E0 READX     CDA4 EREAD     D00A TDEGAB
   D0B8 .Y        D0BE .Y2       D13E

ABS PROGS:
NONE

ENTRY-POINTS:
   8112 OVERLY    824E IFETCH    82C0 .R        82FC .A
   83A2 SQRT      84AA ALOG      85B8 EXP       86BE AINT
   8736 ABS       874C .W        879A .COMP     87C6 $6
   8810 .RARG     883C $8        8868 .5        8870 .ZERO
   8876 .H        88C8 .S        88FE .P        8980 .Q
   8A30 .O        8A78 .MES      8AFA .U        8B22 .V
   8B2C @R        8B52 @Z        8BC4 @H        9F4C PRIMES
   A436 DATARD    B33A THMCPL    B734 READIN    BDC2 ACTIV
   C604 READX     CDC8 EREAD     D02E TDEGAB    D0BA .Y
   D0C0 .Y2

COMMON-BLOCKS:
   FBB2 //

UNDEFINED:
    PROBP2    SORT5    PINV    WTAVER    REGRES    PLOT
```

Memory map for the probability plot segment of the TGDEPG program.

```
REL PROGS:
  3576 TGDEN    80EE OVERLY   824C IFETCH   82BE .R
  82FA .A       83A0 SQRT     84A8 ALOG     85B6 EXP
  86BC AINT     8734 ABS      874A .W       8798 .COMP
  87C4 $6       880E .RARG    883A $8       8864 .5
  886C .ZERO    8874 .H       88C6 .S       88FC .P
  897E .Q       8A2E .O       8A76 .MES     8AEC .U
  8B1C .V       8B2A @R       8B50 @Z       8B76 @H
  9F28 PINV     9FEC PROBP2   A0DE SORT4    A292 SORT5
  A484 WTAVER   AC6E REGRES   C236 FCTN     C2BE MATINV
  CABC PNORM    CC5E AVSDN    CDE6 CHISQ    CF44 PLOT
  D96A .D       D978 .COMEX   DB46 DLOG     DE26 DEXP
  E08E DSQRT    E290 $7       E2EE $9       E326 DAINT
  E3EA @A       E404 @1       E412 @S       E434 DABS
  E44E .G       E464 @F       E56A @L       E59A @K
  E5CA @M       E766 @2       E774 @D       E88E @T
  E8B6 @E       E8DA @O       E8F6 @N       E950 @U
  E968 @V       E988 @P       E9B8 @B       E9C8 .DARG
  EA0A .DSWAP   EA66 .DRTN    EA8E FLOAT    EA94 FLOAT2
  EACC .Y       EAD2 .Y2      EB52 MOD      EB58 MOD2
  EB86 .IIARG   EBC0

ABS PROGS:
NONE

ENTRY-POINTS:
  8112 OVERLY   824E IFETCH   82C0 .R       82FC .A
  83A2 SQRT     84AA ALOG     85B8 EXP      863E AINT
  8736 ABS      874C .W       879A .COMP    87C6 $6
  8810 .RARG    883C $9       8868 .5       8870 .ZERO
  8876 .H       88C8 .S       88FE .P       8980 .Q
  8A30 .O       8A78 .MES     8AFA .U       8B22 .V
  8B2C @R       8B52 @Z       8BC4 @H       9F4C PINV
  A010 PROBP2   A102 SORT4    A2A6 SORT5    A4A8 WTAVER
  AC92 REGRES   C25A FCTN     C2E2 MATINV   CAE0 PNORM
  CC82 AVSDN    CE0A CHISQ    CF68 PLOT     D95C .D
  D97A .COMEX   DB48 DLOG     DEB8 DEXP     E090 DSQRT
  E292 $7       E2F0 $9       E328 DAINT    E3EC @A
  E406 @1       E414 @S       E436 DABS     E450 .G
  E466 @F       E56C @L       E59C @K       E5CC @M
  E768 @2       E776 @D       E890 @T       E8B8 @E
  E8DC @O       E8F8 @N       E952 @U       E96A @V
  E98A @P       E9BA @B       E9CA .DARG    EA0C .DSWAP
  EA68 .DRTN    EA90 FLOAT    EA96 FLOAT2   EACE .Y
  EAD4 .Y2      EB54 MOD      EB5A MOD2     EB88 .IIARG

COMMON-BLOCKS:
  FBB2 //

UNDEFINED:
NONE
```

Appendix E. POLGEP, a FORTRAN program to fit a user-chosen polynominal to data in any format.

File FIXPOL, a variant on the more usual SETXX type of file, which is used as DO FIXPOL,D0 to compile the main routine for the polynomial fitting program and assemble the program.

```
ZU
AC POLGEN.1D0
AC CALOUT.2D1
AS 300
RUN FORTV
AC CALOUT.1D1
AC POLGEB.2D0
RUN CAL
WF 2
CL
ZU
AC POLGEP.1D0
AC POLGEB.7D0
AC READXB.8D0
AC REGRSB.9D0
AC FCTNB.AD0
AC MATINB.BD0
AC EXTPOB.CD0
AC GENREB.DD0
AC FORLIB.FD0
AC POLGL5.5D0
RUN LIBLDR.D0
CL
ZU
TR
```

Example of the input to the POLGEP program. User-supplied information is underlined.

```
 POLGEN PROGRAM, GIVE NUMBER OF POLYNOMIAL TERMS
 AFTER CONSTANT TERM (FORMAT I1)
1
GIVE DEGREES OF POLYNOMIAL IN I1 FORMAT, 1 PER LINE
1


 POLGEN PROGRAM
FIT 1 TERM POLYNOMIAL TO DATA ON FILE 7
 TERMS IN POLYNOMIAL WILL BE: 1

 REPEAT SEQUENCE OF UP TO 100 ALLOWED IN INPUT FILE
PROGRAM WILL READ ALL FORMATS PROVIDED THERE ARE NO
 ALPHABETIC CHARACTERS

WEIGHTED OR UNWEIGHTED FIT IN LEAST SQUARES?
GIVE 1 OR 0 IN I1 FOMAT
0

LEAST SQUARES FIT WILL BE UNWEIGHTED
INPUT ROUTINE
GIVE: LU NUMBER OF INPUT FILE
      NUMBER OF LINES TO SKIP AS TEXT AFTER ALLOWING
        FIRST LINE TO BE A TITLE
      TOTAL NUMBER OF ARGUMENTS IN ONE LINE
      ORDINAL NUMBER OF ARGUMENT TO BE CALLED  = X(1) =
      DITTO FOR X(2),X(3),X(4),X(5),X(6), STOPPING WITH 0 IF < 6 AGUMENTS
      THEN NUMBER OF POINTS TO BE READ IN
      0 = UPTO EOF, MAXIMUM OF 500.
!
7 1 20 2 3 5 0 0

GIVE ONE SCALING FACTOR FOR EACH INPUT VALUE
!
1 1 1

TITLE OF INPUT FILE IS
POOLED E'S FROM PP IN VAC APRIL 78 FROM PPVAC 1,2,3,4,5,6
    1 LINES OF TEXT WILL BE SKIPPED AFTER TITLE
  500 POINTS MAX WILL BE READ IN
   20 ARGUMENTS PER LINE
ARGUMENTS ARE TO BE SAVED IN CALLING ARRAY X AS FOLLOWS:
X(1) IS ARGUMENT NUMBER  2
X(2) IS ARGUMENT NUMBER  3
X(3) IS ARGUMENT NUMBER  5

 SCALE FACTORS APPLIED TO INPUT VALUES ARE:
0.100000E+01 FOR INPUT VALUE NUMBER  2

0.100000E+01 FOR INPUT VALUE NUMBER  3

0.100000E+01 FOR INPUT VALUE NUMBER  5

  ARE ABOVE PARAMETERS OK? ANSWER 0 FOR NO OR 1 FOR YES
!
```

54 DATA READ FROM FILE 7

Y	SIG Y	X
0.643030E+02	0.903000E+00	0.383237E+05
0.584320E+02	0.120500E+01	0.190721E+05
0.687540E+02	0.105600E+01	-0.427110E+05
0.617260E+02	0.111400E+01	-0.539152E+05
0.590680E+02	0.671000E+00	-0.100918E+06
0.746330E+02	0.146500E+01	-0.126686E+06
0.602800E+02	0.255000E+00	0.130262E+06
0.586900E+02	0.352000E+00	0.972663E+05
0.606990E+02	0.545000E+00	0.547585E+05
0.618540E+02	0.495000E+00	0.367767E+05
0.563920E+02	0.666000E+00	0.125428E+05
0.613330E+02	0.504000E+00	-0.256341E+05
0.609680E+02	0.894000E+00	-0.599170E+05
0.535370E+02	0.543000E+00	-0.332576E+05
0.543790E+02	0.110400E+01	-0.637095E+05
0.604910E+02	0.554000E+00	-0.765823E+05
0.627630E+02	0.633000E+00	-0.996102E+05
0.393910E+02	0.106800E+01	-0.121597E+06
0.589750E+02	0.607000E+00	0.758076E+05
0.672830E+02	0.736000E+00	0.267564E+05
0.696680E+02	0.105200E+01	-0.231073E+05
0.632920E+02	0.600000E+00	-0.665374E+05
0.699310E+02	0.956000E+00	-0.105713E+06
0.732750E+02	0.946000E+00	-0.132862E+06
0.723400E+02	0.164200E+01	0.182921E+06
0.603300E+02	0.402000E+00	0.104814E+06
0.625450E+02	0.596000E+00	0.781281E+05
0.609600E+02	0.317000E+00	0.625377E+05
0.631620E+02	0.434000E+00	0.386835E+05
0.596740E+02	0.354000E+00	0.247912E+05
0.594900E+02	0.580000E+00	0.428625E+04
0.603240E+02	0.309000E+00	-0.917900E+04
0.624720E+02	0.581000E+00	-0.366857E+05
0.620170E+02	0.347000E+00	-0.516163E+05
0.635190E+02	0.752000E+00	-0.797934E+05
0.595910E+02	0.465000E+00	-0.939936E+05
0.665220E+02	0.747000E+00	-0.117902E+06
0.671330E+02	0.536000E+00	-0.145887E+06

0.584520E+02	0.429000E+00	0.250229E+06
0.533560E+02	0.466000E+00	0.228785E+06
0.624720E+02	0.353000E+00	0.214575E+06
0.572560E+02	0.493000E+00	0.191049E+06
0.609870E+02	0.380000E+00	0.176091E+06
0.622300E+02	0.717000E+00	0.151685E+06
0.596570E+02	0.540000E+00	0.136797E+06
0.618310E+02	0.462000E+00	0.113023E+06
0.589270E+02	0.370000E+00	0.981807E+05
0.497980E+02	0.499000E+00	0.747946E+05
0.651810E+02	0.420000E+00	0.603817E+05
0.578520E+02	0.693000E+00	0.384264E+05
0.569240E+02	0.551000E+00	0.252802E+05
0.533880E+02	0.852000E+00	0.539138E+04
0.636900E+02	0.503000E+00	-0.684837E+04
0.566540E+02	0.581000E+00	-0.335114E+05

POLYNOMIAL COEFFICIENTS ARE
0.60531662E+02 0.90993643E+00
-0.96534966E-05 0.78444409E-05
CHISQR = 0.332E+02
FTEST = 0.151E+01
RMUL = 0.168E+00
R =-0.168E+00
FITTED ABOUT MID POINT OF T= 0.781281E+05

38324.	64.3	60.9	1.0	-0.0	0.316E-02	0.633E+02	0.0
19072.	58.4	61.1	1.0	-0.0	0.316E-02	0.598E+02	0.0
-42711.	68.8	61.7	1.3	-0.0	0.316E-02	0.470E+02	0.0
-53915.	61.7	61.8	1.4	-0.0	0.316E-02	0.448E+02	0.0
100918.	59.1	62.3	1.7	-0.0	0.316E-02	0.372E+02	0.0
126686.	74.6	62.5	1.8	-0.0	0.316E-02	0.339E+02	0.0
130262.	60.3	60.0	1.0	-0.0	0.316E-02	0.602E+02	0.0
97266.	58.7	60.3	0.9	-0.0	0.316E-02	0.654E+02	0.0
54759.	60.7	60.8	0.9	-0.0	0.316E-02	0.655E+02	0.0
36777.	61.9	60.9	1.0	-0.0	0.316E-02	0.631E+02	0.0
12543.	56.9	61.2	1.0	-0.0	0.316E-02	0.585E+02	0.0
-25634.	61.3	61.5	1.2	-0.0	0.316E-02	0.504E+02	0.0
-59917.	61.0	61.9	1.4	-0.0	0.316E-02	0.437E+02	0.0
-33258.	53.5	61.6	1.3	-0.0	0.316E-02	0.488E+02	0.0
-63709.	54.4	61.9	1.4	-0.0	0.316E-02	0.431E+02	0.0
-76582.	60.5	62.0	1.5	-0.0	0.316E-02	0.409E+02	0.0
-99610.	62.8	62.2	1.7	-0.0	0.316E-02	0.374E+02	0.0
121597.	39.4	62.5	1.8	-0.0	0.316E-02	0.345E+02	0.0
75808.	59.0	60.6	0.9	-0.0	0.316E-02	0.665E+02	0.0
26756.	67.3	61.0	1.0	-0.0	0.316E-02	0.613E+02	0.0
-23107.	69.7	61.5	1.2	-0.0	0.316E-02	0.509E+02	0.0
-66537.	63.3	61.9	1.5	-0.0	0.316E-02	0.426E+02	0.0
105713.	69.9	62.3	1.7	-0.0	0.316E-02	0.365E+02	0.0
132862.	73.3	62.6	1.9	-0.0	0.316E-02	0.331E+02	0.0
182921.	72.3	59.5	1.2	-0.0	0.316E-02	0.485E+02	0.0
104814.	60.3	60.3	0.9	-0.0	0.316E-02	0.646E+02	0.0
78128.	62.5	60.5	0.9	-0.0	0.316E-02	0.665E+02	0.0
62538.	61.0	60.7	0.9	-0.0	0.316E-02	0.661E+02	0.0
38689.	63.2	60.9	1.0	-0.0	0.316E-02	0.634E+02	0.0
24791.	59.7	61.0	1.0	-0.0	0.316E-02	0.610E+02	0.0
4286.	59.5	61.2	1.1	-0.0	0.316E-02	0.568E+02	0.0
-9179.	60.3	61.4	1.1	-0.0	0.316E-02	0.539E+02	0.0

```
-36686.     62.5     61.6      1.3     -0.0  0.316E-02  0.491E+02     0.0
-51616.     62.0     61.8      1.4     -0.0  0.316E-02  0.453E+02     0.0
-79793.     63.5     62.1      1.5     -0.0  0.316E-02  0.404E+02     0.0
-93994.     59.6     62.2      1.6     -0.0  0.316E-02  0.382E+02     0.0
117902.     66.5     62.4      1.8     -0.0  0.316E-02  0.349E+02     0.0
145887.     67.1     62.7      2.0     -0.0  0.316E-02  0.317E+02     0.0
250229.     58.5     58.9      1.6     -0.0  0.316E-02  0.362E+02     0.0
228785.     53.4     59.1      1.5     -0.0  0.316E-02  0.396E+02     0.0
214575.     62.5     59.2      1.4     -0.0  0.316E-02  0.421E+02     0.0
191049.     57.3     59.4      1.3     -0.0  0.316E-02  0.463E+02     0.0
176091.     61.0     59.6      1.2     -0.0  0.316E-02  0.500E+02     0.0
151685.     62.2     59.3      1.1     -0.0  0.316E-02  0.555E+02     0.0
136797.     59.7     60.0      1.0     -0.0  0.316E-02  0.588E+02     0.0
113023.     61.8     60.2      1.0     -0.0  0.316E-02  0.633E+02     0.0
 98131.     58.9     60.3      0.9     -0.0  0.316E-02  0.653E+02     0.0
 74795.     49.8     60.6      0.9     -0.0  0.316E-02  0.665E+02     0.0
 60382.     65.2     60.7      0.9     -0.0  0.316E-02  0.659E+02     0.0
 38486.     57.9     60.9      1.0     -0.0  0.316E-02  0.633E+02     0.0
 25260.     56.9     61.0      1.0     -0.0  0.316E-02  0.610E+02     0.0
  5991.     53.4     61.2      1.1     -0.0  0.316E-02  0.571E+02     0.0
 -6849.     63.7     61.4      1.1     -0.0  0.316E-02  0.544E+02     0.0
-33511.     56.7     61.6      1.3     -0.0  0.316E-02  0.488E+02     0.0
 GIVE EXTRAPOLATION VALUE IN F10.0 FORMAT
 END WITH VALUE OF ZERO
 -175000.
-0.175E+06   0.629752E+02   0.218421E+01   28.83  -0.965358E-05   0.316226E-02
 GIVE EXTRAPOLATION VALUE IN F10.0 FORMAT
 END WITH VALUE OF ZERO
 0.
```

Example of the output of the POLGEP program.

```
 POLGEN PROGRAM
FIT 1 DEGREE POLYNOMIAL TO DATA ON FILE 7
 TERMS IN POLYNOMIAL WILL BE: 1

 REPEAT SEQUENCE OF UP TO 100 ALLOWED IN INPUT FILE
 PROGRAM WILL READ ALL FORMATS PROVIDED THERE ARE NO
 ALPHABETIC CHARACTERS

LEAST SQUARES FIT WILL BE UNWEIGHTED

TITLE OF INPUT FILE IS
POOLED E'S FROM PP IN VAC APRIL 78 FROM PPVAC 1,2,3,4,5,6
   1 LINES OF TEXT WILL BE SKIPPED AFTER TITLE
  500 POINTS MAX WILL BE READ IN
   20 ARGUMENTS PER LINE
ARGUMENTS ARE TO BE SAVED IN CALLING ARRAY X AS FOLLOWS:
X(1) IS ARGUMENT NUMBER  2
X(2) IS ARGUMENT NUMBER  3
X(3) IS ARGUMENT NUMBER  5
 SCALE FACTORS APPLIED TO INPUT VALUES ARE:
0.100000E+01 FOR INPUT VALUE NUMBER  2

0.100000E+01 FOR INPUT VALUE NUMBER  3

0.100000E+01 FOR INPUT VALUE NUMBER  5


    54 DATA READ FROM FILE  7
```

Y	SIG Y	X
0.643030E+02	0.903000E+00	0.383237E+05
0.584320E+02	0.120500E+01	0.190721E+05
0.687540E+02	0.105600E+01	-0.427110E+05
0.617260E+02	0.111400E+01	-0.539152E+05
0.590680E+02	0.571000E+00	-0.100916E+06
0.746330E+02	0.146500E+01	-0.126695E+06
0.602800E+02	0.255000E+00	0.130252E+06
0.586900E+02	0.352000E+00	0.972663E+05
0.606990E+02	0.545000E+00	0.547595E+05
0.618540E+02	0.495000E+00	0.357767E+05
0.568920E+02	0.666000E+00	0.125428E+05
0.613330E+02	0.504000E+00	-0.256341E+05
0.609680E+02	0.894000E+00	-0.539170E+05
0.535370E+02	0.543000E+00	-0.332576E+05
0.543790E+02	0.110400E+01	-0.637095E+05
0.604910E+02	0.554000E+00	-0.755823E+05
0.627630E+02	0.633000E+00	-0.996102E+05
0.393910E+02	0.106800E+01	-0.121597E+06
0.589750E+02	0.607000E+00	0.758076E+05
0.672830E+02	0.736000E+00	0.267564E+05
0.696680E+02	0.105200E+01	-0.231073E+05
0.632920E+02	0.600000E+00	-0.665374E+05
0.699310E+02	0.956000E+00	-0.105713E+06
0.732750E+02	0.946000E+00	-0.132862E+06
0.723400E+02	0.164200E+01	0.182921E+06
0.603300E+02	0.402000E+00	0.104914E+06
0.625450E+02	0.596000E+00	0.781281E+05

0.609600E+02	0.317000E+00	0.625377E+05
0.631620E+02	0.434000E+00	0.396295E+05
0.596740E+02	0.354000E+00	0.247912E+05
0.594960E+02	0.580000E+00	0.428620E+04
0.603240E+02	0.309000E+00	-0.917900E+04
0.624720E+02	0.581000E+00	-0.356257E+05
0.620170E+02	0.347000E+00	-0.515163E+05
0.635190E+02	0.752000E+00	-0.797934E+05
0.535910E+02	0.465000E+00	-0.939235E+05
0.635220E+02	0.747000E+00	-0.117903E+06
0.671330E+02	0.536000E+00	-0.145987E+06
0.584520E+02	0.420000E+00	0.250227E+06
0.533550E+02	0.466000E+00	0.220786E+06
0.624720E+02	0.357000E+00	0.214575E+06
0.572560E+02	0.493000E+00	0.191043E+06
0.609370E+02	0.380000E+00	0.176031E+06
0.622300E+02	0.717000E+00	0.151635E+06
0.536570E+02	0.540000E+00	0.135797E+06
0.610310E+02	0.462000E+00	0.113023E+06
0.509270E+02	0.370000E+00	0.981807E+05
0.497980E+02	0.409000E+00	0.747946E+05
0.651810E+02	0.420000E+00	0.603317E+05
0.578520E+02	0.693000E+00	0.324364E+05
0.569240E+02	0.551000E+00	0.252802E+05
0.533230E+02	0.852000E+00	0.599139E+04
0.636900E+02	0.563000E+00	-0.634337E+04
0.566540E+02	0.581000E+00	-0.335114E+05

POLYNOMIAL COEFFICIENTS ARE
0.60531662E+02 0.90993643E+00
-0.96534968E-05 0.78444409E-05
CHISQR = 0.332E+02
FTEST = 0.151E+01
RMUL = 0.168E+00
R =-0.169E+00
FITTED ABOUT MID POINT OF T= 0.781281E+05

X	Y	FF	SIGFF	RF	SIGRF	PRECFF	PRECRF
38323.7	64.303	60.916	0.962	-0.965E-05	0.316E-02	0.63E+02	0.31E-02
19072.1	58.432	61.102	1.021	-0.965E-05	0.316E-02	0.60E+02	0.31E-02
-42711.0	68.754	61.693	1.314	-0.965E-05	0.316E-02	0.47E+02	0.31E-02
-53915.2	61.726	61.806	1.379	-0.965E-05	0.316E-02	0.45E+02	0.31E-02
-100918.0	59.068	62.260	1.674	-0.965E-05	0.316E-02	0.37E+02	0.31E-02
-126686.0	74.633	62.509	1.846	-0.965E-05	0.316E-02	0.34E+02	0.31E-02
130262.0	60.289	60.028	0.998	-0.965E-05	0.316E-02	0.60E+02	0.31E-02
97266.3	58.690	60.347	0.922	-0.965E-05	0.316E-02	0.65E+02	0.31E-02
54758.6	60.699	60.757	0.928	-0.965E-05	0.316E-02	0.65E+02	0.31E-02
36776.7	61.854	60.931	0.966	-0.965E-05	0.316E-02	0.63E+02	0.31E-02
12542.8	56.892	61.165	1.045	-0.965E-05	0.316E-02	0.59E+02	0.31E-02
-25634.1	61.333	61.533	1.221	-0.965E-05	0.316E-02	0.50E+02	0.31E-02
-59917.0	60.968	61.854	1.414	-0.965E-05	0.316E-02	0.44E+02	0.31E-02
-33257.6	53.537	61.607	1.262	-0.965E-05	0.316E-02	0.49E+02	0.31E-02
-63789.5	54.379	61.901	1.437	-0.965E-05	0.316E-02	0.43E+02	0.31E-02
-76582.3	60.491	62.025	1.517	-0.965E-05	0.316E-02	0.41E+02	0.31E-02
-99610.2	62.763	62.247	1.665	-0.965E-05	0.316E-02	0.37E+02	0.31E-02
-121597.0	39.391	62.460	1.812	-0.965E-05	0.316E-02	0.34E+02	0.31E-02
75807.6	58.975	60.554	0.910	-0.965E-05	0.316E-02	0.67E+02	0.31E-02
26756.4	67.293	61.028	0.995	-0.965E-05	0.316E-02	0.61E+02	0.31E-02
-23107.3	69.668	61.509	1.208	-0.965E-05	0.316E-02	0.51E+02	0.31E-02
-66537.4	63.292	61.928	1.455	-0.965E-05	0.316E-02	0.43E+02	0.31E-02
-105713.0	69.931	62.306	1.705	-0.965E-05	0.316E-02	0.37E+02	0.31E-02
-137862.0	73.275	62.569	1.889	-0.965E-05	0.316E-02	0.33E+02	0.31E-02
182921.0	72.340	59.520	1.226	-0.965E-05	0.316E-02	0.49E+02	0.31E-02
104314.0	60.330	60.274	0.934	-0.965E-05	0.316E-02	0.65E+02	0.31E-02
78128.1	62.545	60.532	0.910	-0.965E-05	0.316E-02	0.67E+02	0.31E-02
62537.7	60.960	60.692	0.918	-0.965E-05	0.316E-02	0.66E+02	0.31E-02
38689.5	63.162	60.912	0.961	-0.965E-05	0.316E-02	0.63E+02	0.31E-02
24791.2	59.674	61.047	1.002	-0.965E-05	0.316E-02	0.61E+02	0.31E-02
4285.2	59.493	61.244	1.079	-0.965E-05	0.316E-02	0.57E+02	0.31E-02
-9179.0	60.324	61.374	1.139	-0.965E-05	0.316E-02	0.54E+02	0.31E-02
-36685.7	62.472	61.640	1.280	-0.965E-05	0.316E-02	0.48E+02	0.31E-02
-51616.3	62.017	61.784	1.365	-0.965E-05	0.316E-02	0.45E+02	0.31E-02
-79793.4	63.519	62.056	1.537	-0.965E-05	0.316E-02	0.40E+02	0.31E-02
-93993.5	59.591	62.193	1.628	-0.965E-05	0.316E-02	0.38E+02	0.31E-02
-117902.0	66.522	62.424	1.787	-0.965E-05	0.316E-02	0.35E+02	0.31E-02
-145837.0	67.133	62.694	1.979	-0.965E-05	0.316E-02	0.32E+02	0.31E-02
250229.0	58.452	58.870	1.628	-0.965E-05	0.316E-02	0.36E+02	0.31E-02
223785.0	53.356	59.077	1.492	-0.935E-05	0.316E-02	0.40E+02	0.31E-02
214575.0	62.472	59.214	1.405	-0.965E-05	0.316E-02	0.42E+02	0.31E-02
191043.0	57.256	59.442	1.270	-0.965E-05	0.316E-02	0.47E+02	0.31E-02
176091.0	60.987	59.586	1.191	-0.965E-05	0.316E-02	0.50E+02	0.31E-02
151685.0	62.230	59.822	1.077	-0.965E-05	0.316E-02	0.56E+02	0.31E-02
136797.0	59.657	59.965	1.020	-0.965E-05	0.316E-02	0.59E+02	0.31E-02
113023.0	61.831	60.195	0.950	-0.965E-05	0.316E-02	0.63E+02	0.31E-02
98180.7	58.927	60.338	0.923	-0.965E-05	0.316E-02	0.65E+02	0.31E-02
74794.6	49.798	60.564	0.910	-0.965E-05	0.316E-02	0.67E+02	0.31E-02
60381.7	65.181	60.703	0.921	-0.965E-05	0.316E-02	0.66E+02	0.31E-02
38486.4	57.652	60.914	0.962	-0.965E-05	0.316E-02	0.63E+02	0.31E-02
25288.2	56.924	61.042	1.000	-0.965E-05	0.316E-02	0.61E+02	0.31E-02
5931.4	53.398	61.228	1.072	-0.965E-05	0.316E-02	0.57E+02	0.31E-02
-6849.9	63.690	61.352	1.128	-0.965E-05	0.316E-02	0.54E+02	0.31E-02
-33511.4	56.654	61.609	1.263	-0.965E-05	0.316E-02	0.49E+02	0.31E-02

TIME	F	SIG F	PREC F	R	SIG R	PREC R
-0.175E+06	0.629752E+02	0.218421E+01	28.83	-0.965350E-05	0.316228E-02	0.00

Listing of the various routines in the POLGEP program.

```
$ASSM
POLGEN PROG
$FORT
$TRGT 16
C  POLGEN, A PROGRAM TO FIT A POLYNOMIAL TO DATA READ IN FROM A FREELY
C    FORMATTED FILE USING SUBROUTINE GENRED
C  POLGEN WILL CALCULATE VALUES FROM THE POLYNOMIAL
C    AT SPECIFIED VALUES OF THE ABSCISSA
C    ALSO, IT WILL CALCULATE THE DERIVATIVE AT THESE POINTS
C
      DIMENSION DATA(1500),TIME(1500),A1(5),SIGA1(5),A2(5),SIGA2(5)
      DIMENSION NTITLE(35),T(250),C(250),SIGC(250),YFIT(250)
     &,TR(250),M(5),A(5),SIGA(5),R(5)
     & ,X(10)
C
C  SET UP RUNNING CONDITIONS
C
      NTTYI=5
      NTTYO=4
      IN=7
      NOUT=6
      WRITE(NOUT,4)
    4 FORMAT('1')
      WRITE(NTTYO,5)
    5 FORMAT('  POLGEN PROGRAM, GIVE NUMBER OF POLYNOMIAL TERMS'
     &/'  AFTER CONSTANT TERM (FORMAT I1)')
      READ(NTTYI,6)NTERMS
      IF(NTERMS.LT.0)STOP
    6 FORMAT(I1)
      WRITE(NTTYO,70)
   70 FORMAT(' GIVE DEGREES OF POLYNOMIAL IN I1 FORMAT, 1 PER LINE')
      READ(NTTYI,6)(M(I),I=1,NTERMS)
      WRITE(NTTYO,20)NTERMS,IN,(M(I),I=1,NTERMS)
   20 FORMAT ('1'/'  POLGEN PROGRAM'/' FIT',I2,' DEGREE'
     &,' POLYNOMIAL TO DATA ON FILE',I2
     &/'  TERMS IN POLYNOMIAL WILL BE:',5I2/
     &/'  REPEAT SEQUENCE OF UP TO 100 ALLOWED IN INPUT FILE'
     &/'  PROGRAM WILL READ ALL FORMATS PROVIDED THERE ARE NO '
     &/'  ALPHABETIC CHARACTERS')
      WRITE(NTTYO,60)
   60 FORMAT(/' WEIGHTED OR UNWEIGHTED FIT IN LEAST SQUARES?'
     &/' GIVE 1 OR 0 IN I1 FOMAT')
      READ(NTTYI,6)NFIT
   40 FORMAT(I2)
      WRITE(NOUT,20)NTERMS,IN,(M(I),I=1,NTERMS)
      IF(NFIT.GT.0)WRITE(NTTYO,80)
      IF(NFIT.LE.0)WRITE(NTTYO,90)
      IF(NFIT.GT.0)WRITE(NOUT,80)
      IF(NFIT.LE.0)WRITE(NOUT,90)
   80 FORMAT(/' LEAST SQUARES FIT WILL BE WEIGHTED')
   90 FORMAT(/' LEAST SQUARES FIT WILL BE UNWEIGHTED')
C
C  READ IN DATA USING GENERAL READING SUBROUTINE
C
      CALL GENRED(X,IEOF,NTTYI,NTTYO,IN,NOUT)
      C(1)=X(1)
```

```
      SIGC(1)=X(2)
      T(1)=X(3)
      DO 100 I=2,250
      CALL GENREE(X,IEOF,NTTYI,NTTYO,IN,NOUT)
      IF(IEOF.GT.0)GO TO 105
      C(I)=X(1)
      SIGC(I)=X(2)
      T(I)=X(3)
  100 CONTINUE
      GO TO 110
  105 I=I-1
  110 NDATA=I
      WRITE(NTTYO,120)NDATA,IN
  120 FORMAT(//I6,' DATA READ FROM FILE',I3
     &//19X,'Y',15X,'SIG Y',19X,'X'/)
      WRITE(NTTYO,44)(C(I),SIGC(I),T(I),I=1,NDATA)
   44 FORMAT(3E20.6)
      WRITE(NOUT,120)NDATA,IN
      WRITE(NOUT,44)(C(I),SIGC(I),T(I),I=1,NDATA)
      IF(NDATA.GE.(NTERMS+1))GO TO 200
      WRITE(NTTYO,210)
  210 FORMAT(/' NOT ENOUGH DATA FOR THIS JOB')
C
C  FIT POLYNOMIAL AND OUTPUT COEFFICIENTS
C
  200 NHALF=NDATA/2
      TMID=T(NHALF)
      DO 400 I=1,NDATA
  400 TR(I)=T(I)-TMID
      CALL REGRES(TR,C,SIGC,NDATA,NTERMS,M,NFIT,YFIT,A0,A,SIGA0,SIGA
     &,R,RMUL,CHISQR,FTEST)
      WRITE(NOUT,140)A0,SIGA0
      WRITE(NTTYO,140)A0,SIGA0
  140 FORMAT(' POLYNOMIAL COEFFICIENTS ARE',/(2E15.8))
      IF(NTERMS.LE.0)GO TO 143
      WRITE(NOUT,141)(A(I),SIGA(I),I=1,NTERMS)
      WRITE(NTTYO,141)(A(I),SIGA(I),I=1,NTERMS)
  141 FORMAT(2E15.8)
  143 CONTINUE
      WRITE(NTTYO,150)CHISQR,FTEST,RMUL,(R(I),I=1,NTERMS)
  150 FORMAT(' CHISQR =',E10.3
     &/' FTEST =',E10.3
     &/' RMUL =',E10.3
     &/(' R =',3E10.3))
      WRITE(NTTYO,144)TMID
  144 FORMAT(' FITTED ABOUT MID POINT OF T=',E15.6)
      WRITE(NOUT,150)CHISQR,FTEST,RMUL,(R(I),I=1,NTERMS)
      WRITE(NOUT,144)TMID
      WRITE(NOUT,160)
  160 FORMAT(9X,'X',14X,'Y',13X,'FF',10X,'SIGFF'
     &,13X,'RF',10X,'SIGRF',9X,'PRECFF',9X,'PRECRF'/)
C
C  CALCULATE EXTRAPOLATED QUANTITIES
C
      L=1
      DO 205 I=1,NDATA
```

```
      CALL EXTPOL(A0,A,SIGA0,SIGA,NTERMS,TR(I),TR(I),FF,SIGFF,FB,SIGFB
     &,RF,SIGRF,RB,IGRB,PRECFF,PRECFB,PRECRF,PRECRB,L,M)
      WRITE(NOUT,172)T(I),C(I),FF,SIGFF,RF,SIGRF,PRECFF,PRECRF
  172 FORMAT(F10.1,3F15.3,2E15.3,2E15.2)
  205 WRITE(NTTYO,170)T(I),C(I),FF,SIGFF,RF,SIGRF,PRECFF,PRECRF
  170 FORMAT(F7.0,4F9.1,2E11.3,F8.1,F7.1)
C
C
      WRITE(NOUT,460)
  460 FORMAT(/'      TIME'
     &/12X,'F',8X,'SIG F',7X,'PREC F',12X,'R',8X,'SIG R',7X,'PREC R'
     &)
   17 WRITE(NTTYO,15)
   15 FORMAT(' GIVE EXTRAPOLATION VALUE IN F10.0 FORMAT'
     &/' END WITH VALUE OF ZERO')
      READ(NTTYI,11)TIME3
   11 FORMAT(F10.0)
      IF(ABS(TIME3).LT.0.0001) GO TO 999
      TIME5=TIME3-TMID
      TIME6=TIME5
      CALL EXTPOL(A0,A,SIGA0,SIGA,NTERMS,TIME6,TIME5,FF,SIGFF,FB,SIGFB
     &,RF,SIGRF,RB,SIGRB,PRECFF,PRECFB,PRECRF,PRECRB,L,M)
      WRITE(NTTYO,16)TIME3,FB,SIGFB,PRECFB,RB,SIGRB,PRECRB
   16 FORMAT(E10.3,2(2E15.6,F8.2))
      WRITE(NOUT,16)TIME3,FB,SIGFB,PRECFB,RB,SIGRB,PRECRB
      WRITE(NOUT,161)
  161 FORMAT(/)
      GO TO 17
  999 WRITE(NOUT,4)
      STOP
      END
```

```
$ASSM
EXTPOL PROG
$FORT
$TRGT 16
C ****************************************************************
C
      SUBROUTINE EXTPOL(ATO,AT,SIGATO,SIGAT
     &,NTERMS
     &,TIMEF,TIMEB,FACTF,SIGFF,FACTB,SIGFB
     &,RATEF,SIGRF,RATEB,SIGRB,PRECFF,PRECFB,PRECRF,PRECRB,L,M)
C
C ****************************************************************
C  SUBROUTINE TO USE POLYNOMIAL OF P=ATO+AT(1)*TIME**M(1)+AT(2)*TIME**M(2)
C    TO EXTRAPOLATE TO VALUES OF BOTH P AND DP/DT TO TIMES TIMEF AND TIMEB
C
C.   ATO = DEFINED ABOVE
C    SIGATO = STANDARD DEVIATION OF ATO
C    AT = DEFINED ABOVE
C    SIGAT = STANDARD DEVIATION OF AT
C    RCHI = CHI-SQUARE VALUE FOR FIT
C    TIMEF = TIME TO EXTRAPOLATE TO IN FORWARD DIRECTION
C    TIMEB = TIME TO EXTRAPOLATE TO IN BACKWARD DIRECTION
C    FACTF = VALUE OF FACTOR AS EXTRAPOLATED FORWARD
C    FACTB = VALUE OF FACTOR AS EXTRAPOLATED BACKWARD
C    RATEF AND RATEB = CORESPONDING VALUES FOR FIRST DERIVATIVES
C    SIGFF, SIGFB, SIGRF, SIGRB = STANDARD DEVIATIONS FOR ABOVE
C    PRECFF,PRECFB,PRECRF,PRECRB = PRECISION OF EXTRAPOLATED
C      RATES AND FACTOR LEVELS
C
C  NEDDS NO COMMON BLOCKS OR SUBROUTINES
C
      DIMENSION AT(NTERMS),SIGAT(NTERMS),M(NTERMS)
C
C  EXTRAPOLATE FACTOR LEVEL
C
      FACTF=ATO
      FACTB=ATO
      SIGFF=SIGATO**2
      SIGFB=SIGATO**2
      IF(NTERMS.LE.0)GO TO 50
      DO 40 I=1,NTERMS
      FACTF=FACTF+AT(I)*TIMEF**M(I)
      FACTB=FACTB+AT(I)*TIMEB**M(I)
      SIGFF=SIGFF+(SIGAT(I)*TIMEF**M(I))**2
      SIGFB=SIGFB+(SIGAT(I)*TIMEB**M(I))**2
   40 CONTINUE
   50 CONTINUE
      SIGFF=SQRT(SIGFF)
      SIGFB=SQRT(SIGFB)
C
C  EXTRAPOLATE DERIVATIVE
C  I.E., IF W=A+B*T+C*T**2,
C  THEN DW/DT=B+2*C*T
C
      RATEF=0.
      RATEB=0.
```

```
      SIGRF=1.E-04
      SIGRB=1.E-04
      IF(L.EQ.0)GO TO 120
      IF(NTERMS.LE.0)GO TO 110
      DO 100 I=1,NTERMS
      J=M(I)-1
      RATEF=RATEF+FLOAT(J+1)*AT(I)*TIMEF**J
      RATEB=RATEB+FLOAT(J+1)*AT(I)*TIMEB**J
      SIGRF=SIGRF+(FLOAT(J+1)*SIGAT(I)*TIMEF**J)**2
      SIGRB=SIGRB+(FLOAT(J+1)*SIGAT(I)*TIMEB**J)**2
  100 CONTINUE
C
      SIGRF=SIGRF-1.E-04
      SIGRB=SIGRB-1.E-04
      IF(SIGRF.LT.0.1E-04)SIGRF=0.1E-04
      IF(SIGRB.LT.0.1E-04)SIGRB=0.1E-04
C
  110 SIGRF=SQRT(SIGRF)
      SIGRB=SQRT(SIGRB)
C
C  EVALUATE PRECISION
C
  120 PRECFF=ABS(FACTF/SIGFF)
      PRECFB=ABS(FACTB/SIGFB)
      PRECRF=ABS(RATEF/SIGRF)
      PRECRB=ABS(RATEB/SIGRB)
      RETURN
      END

$ASSM
FCTN PROG
$FORT
$TRGT 16
C
C     ****************************************************************************
C
      FUNCTION FCTN (X,I,J,JTERMS,NPTS)
C
C     ****************************************************************************
C
C  FUNCTION TO CALCULATE FITTED VALUES FOR REGRES SUBROUTINE
C
C
      DIMENSION X(NPTS),JTERMS(J)
      JEXP=JTERMS(J)
      FCTN=X(I)**JEXP
      RETURN
      END
```

```
$ASSM
GENRED PROG
$FORT
C ****************************************************************************
C
      SUBROUTINE GENRED (X,IEOF,NTTYI,NTTYO,IN,NOUT)
C
C ************************************************************************
C
C  WRITTEN BY BRIAN DICKENS, NBS 921-3322
C $$$$$$ VERSION OF OCT 28, 1977 $$$$$$$$$$$$$$$$$$$$$$$$
C
C  THIS IS A SUBROUTINE FOR FREE FORMAT FREE SAMPLING OF DATA FILES
C  IT NEEDS THE FOLLOWING FUNCTION:
C    READX
C
      DIMENSION NTITLE(35),X(1),NX(100),TEMP(100),XSCALE(100)
C
      J=0
C
C  READ IN INPUT FILE INFORMATION
C
   11 WRITE (NTTYO,10)
   10 FORMAT(' INPUT ROUTINE'
     &/' GIVE: LU NUMBER OF INPUT FILE'
     &/'       NUMBER OF LINES TO SKIP AS TEXT AFTER ALLOWING '
     &/'         FIRST LINE TO BE A TITLE'
     &/'       TOTAL NUMBER OF ARGUMENTS IN ONE LINE'
     &/'       ORDINAL NUMBER OF ARGUMENT TO BE CALLED  = X(1) = '
     &/'       DITTO FOR X(2),X(3),X(4),X(5),X(6), STOPPING WITH 0 '
     &,'IF < 6 AGUMENTS'
     &/'       THEN NUMBER OF POINTS TO BE READ IN'
     &/'       0 = UPTO EOF, MAXIMUM OF 500.')
      IN = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      IF(IN.NE.NTTYO) GO TO 210
      WRITE(NTTYO,205)
  205 FORMAT(/' $$$  INPUT FILE SAME AS OUTPUT SCRATCH '
     &,'FILE $$$')
      GO TO 11
  210 REWIND IN
      NTEXT = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      NLINE = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NLINE.LE.100)GO TO 14
      WRITE (NTTYO,13)NLINE
   13 FORMAT (/' TOO MANY ARGUMENTS GIVEN FOR INPUT LINE'
     $/' YOU GAVE',I6,' MAX IS 100'/' TRY AGAIN')
      GO TO 11
   14 DO 15 I = 1,100
      NX(I) = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NX(I).LE.0)GO TO 16
   15 CONTINUE
      GO TO 17
   16 I = I-1
   17 CONTINUE
      MX = I
      NTOTAL = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
```

```
      IF (NTOTAL.LE.0)NTOTAL = 500
      IF (MX.LT.1)GO TO 11
C
      WRITE (NTTYO,138)
  138 FORMAT (/' GIVE ONE SCALING FACTOR FOR EACH INPUT VALUE')
      DO 139 J = 1,MX
      XSCALE(J) = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
  139 CONTINUE
C
C  BEGIN READING
C
      READ(IN,25)NTITLE
   25 FORMAT (35A2)
      WRITE (NTTYO,26)NTITLE,NTEXT,NTOTAL,NLINE,(I,NX(I),I = 1,MX)
   26 FORMAT (/' TITLE OF INPUT FILE IS'/35A2
     &/I6,' LINES OF TEXT WILL BE SKIPPED AFTER TITLE'
     &/I6,' POINTS MAX WILL BE READ IN'
     &/I6,' ARGUMENTS PER LINE'
     &/' ARGUMENTS ARE TO BE SAVED IN CALLING ARRAY X AS FOLLOWS:'
     &/(' X(',I1,') IS ARGUMENT NUMBER',I3))
      WRITE (NTTYO,141)(XSCALE(J),NX(J),J = 1,MX)
  141 FORMAT (/' SCALE FACTORS APPLIED TO INPUT VALUES ARE:'
     &,(/E12.6,' FOR INPUT VALUE NUMBER',I3))
      WRITE(NTTYO,250)
  250 FORMAT(/'  ARE ABOVE PARAMETERS OK? ANSWER 0 FOR NO OR 1 FOR YES')
      NTEST=READX(IW,NERROR,3,NTTYI,NTTYO,IEOF)
      IF(NTEST.NE.1) GO TO 11
      WRITE (NOUT,26)NTITLE,NTEXT,NTOTAL,NLINE,(I,NX(I),I = 1,MX)
      WRITE(NOUT,141)(XSCALE(J),NX(J),J=1,MX)
C
C  SKIP TEXT
C
      IF (NTEXT.LE.0)GO TO 810
      DO 820 I = 1,NTEXT
  820 READ(IN,25)NTITLE
  810 CONTINUE
C
C  READ IN A BANK OF VALUES
C
      ENTRY GENREE
      J = 0
      NEWREC = 1
  150 DO 20 I = 1,NLINE
      IF (I.EQ.1)NEWREC = 3
      TEMP(I) = READX (IW,NERROR,NEWREC,IN,NTTYO,IEOF)
      NEWREC = 1
      IF (IEOF.GT.0)RETURN
      IF (NERROR.GT.0)GO TO 150
   20 CONTINUE
      J = J+1
C
C  FILL IN X ARRAY
C
      DO 30 I = 1,MX
      L = NX(I)
      X(I) = TEMP(L)
      X(I) = X(I)*XSCALE(I)
   30 CONTINUE
      IF (J.EQ.NTOTAL)IEOF=1
      RETURN
      END
```

```
$ASSM
MATINV PROG
$FORT
$TRGT 16
C     ****************************************************************************
C
      SUBROUTINE MATINV (ARRAY,NORDER,DET)
C
C     ****************************************************************************
C  SUBROUTINE FOR MATRIX INVERSION
C
      DOUBLE PRECISION ARRAY,AMAX,SAVE
      DIMENSION ARRAY(5,5),IK(5),JK(5)
      DET=1.
      DO 190 K=1,NORDER
C
C  FIND LARGEST ELEMENT ARRAY(I,J) IN REST OF MATRIX
C
          AMAX=0.
10        DO 30 I=K,NORDER
              DO 30 J=K,NORDER
              IF (DABS(AMAX)-DABS(ARRAY(I,J))) 20,20,30
20            AMAX=ARRAY(I,J)
              IK(K)=I
              JK(K)=J
30            CONTINUE
C
C  INTERCHANGE ROWS AND COLUMNS TO PUT AMAX IN ARRAY(K,K)
C
          IF (AMAX) 50,40,50
40        DET=0.
          GO TO 260
50        I=IK(K)
          IF (I-K) 10,80,60
60        DO 70 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=ARRAY(I,J)
70            ARRAY(I,J)=-SAVE
80        J=JK(K)
          IF (J-K) 10,110,90
90        DO 100 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=ARRAY(I,J)
100           ARRAY(I,J)=-SAVE
C
C  ACCUMULATE ELEMENTS OF INVERSE MATRIX
C
110       DO 130 I=1,NORDER
              IF (I-K) 120,130,120
120           ARRAY(I,K)=-ARRAY(I,K)/AMAX
130           CONTINUE
          DO 160 I=1,NORDER
              DO 160 J=1,NORDER
              IF (I-K) 140,160,140
140           IF (J-K) 150,160,150
150           ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
```

```
160             CONTINUE
           DO 180 J=1,NORDER
             IF (J-K) 170,180,170
170             ARRAY(K,J)=ARRAY(K,J)/AMAX
180             CONTINUE
           ARRAY(K,K)=1./AMAX
190        DET=DET*AMAX
C
C  RESTORE ORDERING OF MATRIX
C
      DO 250 L=1,NORDER
           K=NORDER-L+1
           J=IK(K)
           IF (J-K) 220,220,200
200        DO 210 I=1,NORDER
              SAVE=ARRAY(I,K)
              ARRAY(I,K)=-ARRAY(I,J)
210           ARRAY(I,J)=SAVE
220        I=JK(K)
           IF (I-K) 250,250,230
230        DO 240 J=1,NORDER
              SAVE=ARRAY(K,J)
              ARRAY(K,J)=-ARRAY(I,J)
240           ARRAY(I,J)=SAVE
250        CONTINUE
260   RETURN
      END
```

```
$ASSM
READX PROG
$FORT
      FUNCTION READX(IWORD,N,NP,NTTYI,NTTYO,IEOF)
C
C $$$$$$$$$  VERSION OF MAR 9,1977 $$$$$$$$$
C
C  FUNCTION READX USUALLY RETURNS THE NEXT DECIPHERIBLE NUMBER ON THE
C    INPUT FILE =NTTYI= AS THE VALUE OF THE VARIABLE READX.
C    ANY OUPUT, SUCH AS ERROR MESSAGES OR THE PROMPT CHARACTER,
C    IS ON UNIT NTTYO.
C
C  IW= ALPHANUMERIC INFORMATION OF THE FIRST 1 OR 2 ILLEGAL NON-NUMERICAL
C    INFORMATION ON INPUT FILE NTTYI.
C  N=0 MEANS NUMERICAL INFORMATION SUCCESSFULLY PROCESSED.
C  N=1 CHECK IW WORD FOR NON-NUMERICAL INFORMATION.
C       (READX=1. IN THIS CASE).
C  NP=1  SEND ! AS A PROMPT CHARACTER WHEN NEW RECORD REQUIRED.
C  NP=2  READ A NEW RECORD.
C  NP=3  DO BOTH THE ABOVE.
C  NTTYI=  INPUT UNIT
C  NTTYO=  OUTPUT UNIT FOR PROMPT CHARACTER AND ERROR MESSAGES.
C  IEOF=0   END OF FILE NOT YET REACHED ON UNIT NTTYI.
C  IEOF=1  END OF FILE FOUND ON NTTYI.
C
C  ILLEGAL INFORMATION:
C
C  IF FIRST CHARACTER IS NOT A BLANK, COMMA, POINT, MINUS, OR
C    A NUMBER, THE FIRST 2 ILLEGAL CHARACTERS ARE RETURNED IN =IW=.
C
C  LEGAL INFORMATION*
C
C  LEADING BLANKS ARE IGNORED.
C  LEADING COMMAS ARE IGNORED.
C  DECIMAL POINTS AND EXPONENTIAL TYPE NUMBERS ARE TREATED CORRCECTLY.
C  $ ON THE INPUT FILE IS PRESUMED TO PRECEED AN IN-LINE COMMENT AND
C    CAUSES READING OF A NEW RECORD:
C  NUMBERS ARE TERMINATED BY A BLANK, COMMA, OR BY THE END OF A RECORD.
C
C
C  READX WAS ORIGINALLY WRITTEN BY FRANK MCCRACKIN (NBS) AND WAS MODIFIED
C    BY FRED MOPSIK AND BY BRIAN DICKENS.  (JAN 1977)
C
C
C
      INTEGER BRANCH
      INTEGER*2 CD,DEC,IW2,N1,J,IEXP,IEXP1
      LOGICAL ISIGNE
      DIMENSION CD(80),DEC(10),TEN(10),TENTEN(10)
      DATA DEC/'0','1','2','3','4','5','6','7','8','9'/
      DATA TEN/1.,1.E1,1.E2,1.E3,1.E4,1.E5,1.E6,1.E7,1.E8,1.E9/
      DATA TENTEN/1.,1.E10,1.E20,1.E30,1.E40,1.E50,1.E60,3*1.E70/
      DATA N1/81/
      IF (NP.EQ.2.OR.NP.EQ.3) N1 = 81
      IEOF = 0
      N = 0
```

```
      SIGN = 1.
      X = 0.
      Y = 1.
      YJ = 0.
C
C   FIND FIRST CHARACTER
C
15    IF (N1.GT.80) GO TO 1000
      IF (CD(N1).EQ.'$') GO TO 1000
      ASSIGN 16 TO BRANCH
      GO TO 2000
16    GO TO (40,100,30,20,20,300),IP
20    N1 = N1+1
      GO TO 15
C   SET SIGN
30    SIGN = -1.
      N1 = N1+1
      GO TO 15
C
C   CALCULATE NUMBER
C
40    X = J
50    N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 50
      ASSIGN 70 TO BRANCH
      GO TO 2000
70    GO TO (75,100,205,200,270,270),IP
75    X = X*10.+J
      GO TO 50
C
C   CALCULATE FRACTION
C
100   CONTINUE
110   N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 110
      ASSIGN 125 TO BRANCH
      GO TO 2000
125   GO TO (130,3055,205,200,270,270),IP
130   YJ = 10.*YJ+J
      Y = 10.*Y
      GO TO 110
C
C   CALCULATE EXPONENT IEXP
C
200   ISIGNE = .FALSE.
      IEXP = 1
      IEXP1 = 1
      GO TO 210
205   ISIGNE = .TRUE.
      IEXP = 1
      IEXP1 = 1
210   N1 = N1+1
      IF (N1.GT.80) GO TO 250
      ASSIGN 220 TO BRANCH
```

```
      GO TO 2000
220   GO TO (230,250,250,250,250,250),IP
230   IEXP1 = IEXP
      IEXP = J+1
      GO TO 210
250   IF (ISIGNE) GO TO 260
      READX = SIGN*(X+YJ/Y)*TEN(IEXP)*TENTEN(IEXP1)
      N = 0
      RETURN
260   READX = SIGN*(X+YJ/Y)/(TEN(IEXP)*TENTEN(IEXP1))
      N = 0
      RETURN
270   READX = SIGN*(X+YJ/Y)
      N = 0
      RETURN
C
C   RETURN ALPHA ARG
C
300   IW2 = CD(N1)
      N1 = N1+1
      IF (N1.GT.80) GO TO 302
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',' .OR. CD(N1).EQ.'$') GO TO 302
      IW2 = IW2-X'20'+CD(N1)/X'100'
      N1 = N1+1
302   IWORD = IW2
      N = 1
      READX = 1.
      RETURN
3055  WRITE (NTTYO,3056)N1
3056  FORMAT(' 2d . AT COL.',I3)
      GO TO 270
1000  N1 = 1
      IF (NP.EQ.1.OR.NP.EQ.3) WRITE (NTTYO,1020)
1020  FORMAT(' !')
      READ (NTTYI,1010,END = 9000) CD
1010  FORMAT(80A1)
      GO TO 15
C
C   CHARACTER INTERP. ROUTINE
C
2000  DO 2010 J = 1,10
      IF (CD(N1).EQ.DEC(J)) GO TO 2020
2010  CONTINUE
      IP = 6
      IF (CD(N1).EQ.'.') IP = 2
      IF (CD(N1).EQ.'-') IP = 3
      IF (CD(N1).EQ.'+') IP = 4
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',') IP = 5
      GO TO 2030
2020  IP = 1
      J = J-1
2030  GO TO BRANCH
 9000 IEOF = 1
      RETURN
      END
```

```
$ASSM
REGRES PROG
$FORT
$TRGT 16
C
C     ************************************************************************
C
      SUBROUTINE REGRES (X,Y,SIGMAY,NPTS,NTERMS,M,MODE,YFIT,A0,A,SIGMA0,
     2 SIGMAA,R,RMUL,CHISQR,FTEST)
C
C     ************************************************************************
C
C  NEEDS THE FOLLOWING FUNCTION AND SUBROUTINE:
C    FCTN
C    MATINV
C
C
      DOUBLE PRECISION ARRAY,SUM,YMEAN,SIGMA,CHISQ,XMEAN,SIGMAX
      DIMENSION X(NPTS),Y(NPTS),SIGMAY(NPTS),M(5),YFIT(NPTS)
     &,SIGMAA(5),R(5),A(5)
      DIMENSION WEIGHT(250),XMEAN(5),SIGMAX(5),ARRAY(5,5)
C
C  INITIALIZE SUMS AND ARRAYS
C
      SUM=0.
      YMEAN=0.
      SIGMA=0.
      CHISQ=0.
      RMUL=0.
      DO 10 I=1,NPTS
10        YFIT(I)=0.
      IF (NTERMS.LE.0) GO TO 30
      DO 20 J=1,NTERMS
          XMEAN(J)=0.
          SIGMAX(J)=0.
          R(J)=0.
          A(J)=0.
          SIGMAA(J)=0.
          DO 20 K=1,NTERMS
20        ARRAY(J,K)=0.
30    CONTINUE
C
C  ACCUMULATE WEIGHTED SUMS
C
      DO 110 I=1,NPTS
          IF (MODE) 40,70,80
40        IF (Y(I)) 60,70,50
50        WEIGHT(I)=1./Y(I)
          GO TO 90
60        WEIGHT(I)=1./(-Y(I))
          GO TO 90
70        WEIGHT(I)=1.
          GO TO 90
80        WEIGHT(I)=1./SIGMAY(I)**2
90        SUM=SUM+WEIGHT(I)
          YMEAN=YMEAN+WEIGHT(I)*Y(I)
```

```
         IF (NTERMS.LE.0) GO TO 110
         DO 100 J=1,NTERMS
100          XMEAN(J)=XMEAN(J)+WEIGHT(I)*FCTN(X,I,J,M,NPTS)
110      CONTINUE
      YMEAN=YMEAN/SUM
      IF (NTERMS.LE.0) GO TO 130
      DO 120 J=1,NTERMS
120      XMEAN(J)=XMEAN(J)/SUM
130   CONTINUE
      FNPTS=NPTS
      WMEAN=SUM/FNPTS
      DO 140 I=1,NPTS
140      WEIGHT(I)=WEIGHT(I)/WMEAN
C
C  ACCUMULATE MATRICES R AND ARRAY
C
      DO 150 I=1,NPTS
      SIGMA=SIGMA+WEIGHT(I)*(Y(I)-YMEAN)**2
      IF (NTERMS.LE.0) GO TO 160
      DO 150 J=1,NTERMS
      SIGMAX(J)=SIGMAX(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))**2
      R(J)=R(J)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))*(Y(I)-YMEAN)
         DO 150 K=1,J
      ARRAY(J,K)=ARRAY(J,K)+WEIGHT(I)*(FCTN(X,I,J,M,NPTS)-XMEAN(J))
     2 *(FCTN(X,I,K,M,NPTS)-XMEAN(K))
  150 CONTINUE
160   CONTINUE
      FREE1=NPTS-1
      SIGMA=DSQRT(SIGMA/FREE1)
      IF (NTERMS.LE.0) GO TO 180
      DO 170 J=1,NTERMS
         SIGMAX(J)=DSQRT(SIGMAX(J)/FREE1)
         R(J)=R(J)/(FREE1*SIGMAX(J)*SIGMA)
         DO 170 K=1,J
         ARRAY(J,K)=ARRAY(J,K)/(FREE1*SIGMAX(J)*SIGMAX(K))
170      ARRAY(K,J)=ARRAY(J,K)
180   CONTINUE
C
C  INVERT SYMMETRIC MATRIX
C
      IF (NTERMS.LE.0) GO TO 200
      CALL MATINV (ARRAY,NTERMS,DET)
      IF (DET) 200,190,200
190   A0=0.
      SIGMA0=0.
      RMUL=0.
      CHISQR=0.
      FTEST=0.
      GO TO 330
C
C  CALCULATE COEFFICIENTS, FIT, AND CHI SQUARE
C
200   A0=YMEAN
      IF (NTERMS.LE.0) GO TO 230
      DO 220 J=1,NTERMS
         DO 210 K=1,NTERMS
```

```
210             A(J)=A(J)+R(K)*ARRAY(J,K)
           A(J)=A(J)*SIGMA/SIGMAX(J)
           A0=A0-A(J)*XMEAN(J)
           DO 220 I=1,NPTS
220        YFIT(I)=YFIT(I)+A(J)*FCTN(X,I,J,M,NPTS)
230     CONTINUE
        DO 240 I=1,NPTS
           YFIT(I)=YFIT(I)+A0
240        CHISQ=CHISQ+WEIGHT(I)*(Y(I)-YFIT(I))**2
        FREEN=NPTS-NTERMS-1
        CHISQR=CHISQ*WMEAN/FREEN
C
C  CALCULATE  UNCERTAINTIES
C
        IF (MODE) 250,260,250
250     VARNCE=1./WMEAN
        GO TO 270
260     VARNCE=CHISQR
  270   IF (NTERMS.LE.0) GO TO 290
        DO 280 J=1,NTERMS
           SIGMAA(J)=ARRAY(J,J)*VARNCE/(FREE1*SIGMAX(J)**2)
           SIGMAA(J)=SQRT(SIGMAA(J))
280        RMUL=RMUL+A(J)*R(J)*SIGMAX(J)/SIGMA
        FREEJ=NTERMS
        FTEST=(RMUL/FREEJ)/((1.-RMUL)/FREEN)
        RMUL=SQRT(RMUL)
        GO TO 300
290     CONTINUE
        FREEJ=0.
        FTEST=0.
        RMUL=0.
300     SIGMA0=VARNCE/FNPTS
        IF (NTERMS.LE.0) GO TO 320
        DO 310 J=1,NTERMS
           DO 310 K=1,NTERMS
310        SIGMA0=SIGMA0+VARNCE*XMEAN(J)*XMEAN(K)*ARRAY(J,K)/(FREE1*SIGMA
      2X(J)*SIGMAX(K))
320     SIGMA0=SQRT(SIGMA0)
330     RETURN
        END
```

File FIXPOL, a transfer file which is used as DO FIXPOL,DO to compile the main routine for the polynomial fitting program and to assemble the program.

```
ZU
AC POLGEN.1D0
AC CALOUT.2D1
AS 300
RUN FORTV
AC CALOUT.1D1
AC POLGEB.2D0
RUN CAL
WF 2
CL
ZU
AC POLGEP.1D0
AC POLGEB.7D0
AC READXB.8D0
AC REGRSB.9D0
AC FCTNB.AD0
AC MATINB.BD0
AC EXTPOB.CD0
AC GENREB.DD0
AC FORLIB.FD0
AC POLGL5.5D0
RUN LIBLDR.D0
CL
ZU
TR
```

File POLG5, which contains the appropriate responses to the program LIBLDR to make the POLGEP program.

```
OUT 1 POLGEP
BI 3572
LO 7
LI 8
LI 9
LI A
LI B
LI C
LI D
ED F
MAP
XOUT
WF 1
END
```

Memory map of the POLGEP program.

REL PROGS:

3572 POLGEN	8936 READX	90FA REGRES	A6C2 FCTN
A74A MATINV	AF48 EXTPOL	B4A2 GENRED	C570 .D
C57E .COMEX	C74C DLOG	CABC DEXP	CC94 DSQRT
CE96 $7	CEF4 $9	CF2C DAINT	CFF0 @A
D00A @1	D018 @S	D03A DABS	D054 .G
D06A @F	D170 @L	D1A0 @K	D1D0 @M
D36C @2	D37A @D	D494 @T	D4BC @E
D4E0 @O	D4FC @N	D556 @U	D56E @V
D58E @P	D5BE @B	D5CE .DARG	D610 .DSWAP
D66C .DRTN	D694 .R	D6D0 .A	D776 SQRT
D87E ALOG	D98C EXP	DA92 AINT	DB0A FLOAT
DB10 FLOAT2	DB48 .Y	DB4E .Y2	DBCE ABS
DBE4 .W	DC32 .COMP	DC5E $6	DCA8 .RARG
DCD4 $8	DCFE .5	DD06 .ZERO	DD0E .S
DD44 .P	DDC6 .M	DE4A .Q	DEFA .O
DF42 .MES	DFB8 .U	DFE8 .V	DFF6 @R
E01C @H	F265		

ABS PROGS:
NONE

ENTRY-POINTS:

895A READX	911E REGRES	A5E6 FCTN	A76E MATINV
AF6C EXTPOL	B4C6 GENRED	BEA0 GENREE	C572 .D
C580 .COMEX	C74E DLOG	CABE DEXP	CC96 DSQRT
CE98 $7	CEF6 $9	CF2E DAINT	CFF2 @A
D00C @1	D01A @S	D03C DABS	D056 .G
D06C @F	D172 @L	D1A2 @K	D1D2 @M
D36E @2	D37C @D	D496 @T	D4BE @E
D4E2 @O	D4FE @N	D558 @U	D570 @V
D590 @P	D5C0 @B	D5D0 .DARG	D612 .DSWAP
D66E .DRTN	D696 .R	D6D2 .A	D778 SQRT
D880 ALOG	D98E EXP	DA94 AINT	DB0C FLOAT
DB12 FLOAT2	DB4A .Y	DB50 .Y2	DBD0 ABS
DBE6 .W	DC34 .COMP	DC60 $6	DCAA .RARG
DCD6 $8	DD02 .5	DD0A .ZERO	DD10 .S
DD46 .P	DDC8 .M	DE4C .Q	DEFC .O
DF44 .MES	DFC6 .U	DFEE .V	DFF8 @R
E06A @H			

COMMON-BLOCKS:
NONE

UNDEFINED:
NONE

Appendix F. TABLER, a FORTRAN program to make publication-ready tables from the factor-jump thermogravimetry save file.

Example of output of TABLEP program. The input is a SAVE file from TGRUNF assigned to logical unit 5. The output is on units 6 and 7.

```
Use  *AC   SAVE,5D1
     *AS   662
     *AS   720
     *RUN  TABLEP,D1
```

DOC	E	SIG E	T1	R1	T2	R2	SWT (MG)
74	44.8	0.4	366.86(3)	-60.6(2)	358.93(3)	-39.0(1)	14.5
77	44.1	0.3	359.04(3)	-34.4(1)	369.03(3)	-59.4(1)	12.7
81	43.5	0.4	368.78(3)	-48.5(1)	360.84(4)	-31.7(1)	10.2
84	44.9	0.3	360.92(4)	-26.7(1)	370.97(4)	-46.5(1)	8.8
87	44.2	0.4	370.84(4)	-36.9(1)	362.81(4)	-23.8(0)	6.9
89	45.0	0.3	363.01(4)	-19.6(0)	373.02(3)	-33.9(0)	5.8
92	45.2	0.4	372.74(3)	-25.4(0)	364.84(4)	-16.4(0)	4.4
95	44.6	0.4	380.78(4)	-20.3(1)	390.87(4)	-34.2(1)	2.5
74	43.1	0.4	356.03(4)	-32.5(1)	366.02(4)	-55.7(1)	15.0
78	41.3	0.4	365.82(4)	-47.9(1)	357.86(5)	-31.8(1)	12.5
80	43.4	0.3	357.92(5)	-27.0(1)	367.94(4)	-46.4(1)	11.2
84	43.2	0.4	367.85(4)	-39.4(1)	359.82(5)	-25.6(0)	9.1
86	44.6	0.3	360.05(4)	-21.6(0)	369.97(3)	-37.4(1)	8.0
89	43.8	0.4	369.88(3)	-30.5(1)	361.84(4)	-19.8(0)	6.4
90	45.7	0.4	361.97(4)	-16.0(0)	371.99(4)	-28.1(0)	5.6
92	46.4	0.5	371.84(4)	-22.0(0)	363.90(4)	-14.0(0)	4.4
93	45.8	0.5	363.98(4)	-11.6(0)	389.14(7)	-28.3(2)	3.8
39	44.3	0.4	371.99(3)	-47.4(1)	382.04(2)	-90.4(2)	14.2
54	44.8	0.4	381.86(2)	-74.2(2)	373.86(2)	-48.5(1)	10.5
63	46.3	0.2	374.09(2)	-41.4(1)	384.04(3)	-71.3(1)	8.4
76	44.7	0.3	383.86(3)	-51.5(1)	375.83(3)	-33.7(1)	5.5
82	45.8	0.3	375.90(3)	-25.2(1)	385.99(3)	-43.4(1)	4.1
89	45.5	0.4	385.81(2)	-25.8(1)	377.81(2)	-16.8(0)	2.5
92	45.3	0.2	377.89(2)	-11.4(0)	393.98(2)	-25.6(1)	1.8
15	45.7	1.0	369.91(2)	-27.4(2)	380.02(2)	-47.7(5)	15.2
56	44.0	0.5	381.87(2)	-61.0(2)	373.87(2)	-49.2(1)	7.9
66	46.0	0.2	374.04(2)	-33.5(1)	384.04(2)	-57.7(1)	6.1
78	45.2	0.4	383.88(2)	-41.1(1)	375.78(4)	-26.7(1)	3.8
84	46.3	0.3	376.06(4)	-19.0(1)	386.10(2)	-32.8(0)	2.7
91	44.8	0.4	385.83(2)	-17.6(0)	377.84(2)	-11.5(0)	1.5

Listing of the TABLEP program.

```
$ASSM
TABLER PROG
$FORT
C
C  TABLER, A PROGRAM TO MAKE A TABLE FROM TGRUNF OUTPUT
C     IN THE FORM OF THE SAVE FILE
C     THIS PROGRAM WILL DEDUCE THE ACTUAL WEIGHT OF THE SAMPLE IN MG
C     FROM THE DEGREES OF CONVERSION AND THE ELECTROBALANCE READINGS IN
C     MICROVOLTS FOR EACH SINGLE SAMPLE RUN (THIS IS DETECTED BY THE
C     FACT THAT THE DEGREE OF CONVERSION GOES FROM HIGH TO LOW AS ONE
C     SET OF DATA ENDS AND ANOTHER BEGINS).
C
C  NEEDS FUNCTION READX
C
C  X IS DUMMY VARIABLE
C
      DIMENSION NTITLE(80),A(12,100)
C
C  INITIALISE UNITS AND PROGRAM OUTPUT
C
      NTTYO=4
      INFILE=5
      NPRINT=6
      WRITE(NTTYO,200)
  200 FORMAT(//' TABLER PROGRAM'
     &/'  READ DATA FROM FILE 5'
     &/'  PRINT RESULTS ON TERMINAL AND ON FILE 6'//)
      READ(INFILE,5)NTITLE
    5 FORMAT(40A2)
      WRITE(NPRINT,30)
      WRITE(NTTYO,30)
   30 FORMAT('1'///' DOC',5X,'E ',' SIG E',5X,'T1',10X,'R1',8X,'T2'
     & ,10X,'R2',10X,'SWT'/67X,'(MG)'/)
C
C  READ IN DATA
C
      LOW=1
   80 DO 10 I=LOW,100
      X=READX(IW,NERROR,3,5,4,IEOF)
      IF(IEOF.GT.0) GO TO 50
      A(1,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(2,I)=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      A(3,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(4,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(5,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(6,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(7,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(8,I)=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      A(9,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(10,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(11,I)=READX(IW,NERROR,1,5,4,IEOF)
      A(12,I)=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      X=READX(IW,NERROR,1,5,4,IEOF)
      IF (I.EQ.1) GO TO 10
      J=I-1
      IF (A(3,I).GT.A(3,J)) GO TO 50
   10 CONTINUE
C
C  CALCULATE INITIAL AND FINAL SAMPLE WEIGHTS FROM FIRST AND LAST DEGREES
C     OF CONVERSION FOR A SINGLE SAMPLE RUN IN THE DATA STREAM
```

```
C     ( USUALLY SEVERAL SINLGE SAMPLE RUNS ARE COMBINED TO MAKE UP
C       THE COMPLETE SET OF DATA.)
C
   50 NDATA=I-1
      DOC1=A(4,1)
      DOC2=A(4,NDATA)
      SWT1=A(3,1)
      SWT2=A(3,NDATA)
      FACTOR=(SWT1-SWT2)/(DOC2-DOC1)
      WINITL=SWT1+DOC1*FACTOR
      WFINAL=SWT2-(100.-DOC2)*FACTOR
C
C   CALCULATE SAMPLE WEIGHTS IN MG FROM INPUT WEIGHT
C    AND CALCULAED FINAL WEIGHT OF SAMPLE
C
      DO 60 J=1,NDATA
      SWT=(A(3,J)-WFINAL)/(20.*1000.)
      IST1=A(6,J)*100.+0.5
      IST2=A(10,J)*100.+0.5
      ISR1=A(8,J)*10.+0.5
      ISR2=A(12,J)*10.+0.5
      IDOC=A(4,J)
      E=A(1,J)
      SE=A(2,J)
      T1=A(5,J)
      R1=A(7,J)
      T2=A(9,J)
      R2=A(11,J)
      WRITE(NPRINT,20)IDOC,E,SE,T1,IST1,R1,ISR1,T2,IST2,R2,ISR2,SWT
      WRITE(NTTYO,20)IDOC,E,SE,T1,IST1,R1,ISR1,T2,IST2,R2,ISR2,SWT
   20 FORMAT(I4,F7.1,F6.1,2(F8.2,'(',I2,')',F7.1,'(',I1,')'),F10.1)
   60 CONTINUE
C
C   SET UP FIRST MEMBER OF ARRRAY OF DATA FOR NEXT SINGLE SAMPLE RUN
C
      DO 70 J=1,12
   70 A(J,1)=A(J,I)
      LOW=2
      IF(IEOF.EQ.0) GO TO 80
  100 WRITE(NPRINT,110)
  110 FORMAT('1')
      END FILE NPRINT
      END FILE NTTYO
      STOP
      END
```

File FIXTAB, used as DO FIXTAB,D0 to recompile the TABLEP program source and to make the file assignments needed to reassemble the TABLEP program.

```
ZU
AC TABLER,1D0
AC CALOUT,2D1
AS 300
RUN FORTV
AC CALOUT,1D1
AC TABLEB,2D0
RUN CAL
WF 2
CL
ZU
AC TABLEP,1D0
AC TABLL5,5D0
AC TABLEB,7D0
AC READXB,8D0
AC FORLIB,FD0
RUN LIBLDR,D0
ZU
TR
```

File TABLL5, which contains the appropriate responses to the program LIBLDR to reassemble the TABLEP program.

```
OUT 1 TABLEP
BI 3572
LO 7
LI 8
ED F
MAP
XOUT
WF 1
END
```

Memory map of the TABLEP program.

```
REL PROGS:
  3572 TABLER    5432 READX     5BF6 .Y        5BFC .Y2
  5C7C .W        5CCA .ZERO     5CD2 .S        5D08 .P
  5D8A .Q        5E3A .O        5E82 .MES      5EF8 .U
  5F28 .V        5F36 @Z        5F5C @H        71A6

ABS PROGS:
NONE

ENTRY-POINTS:
  5456 READX     5BF8 .Y        5BFE .Y2       5C7E .W
  5CCE .ZERO     5CD4 .S        5D0A .P        5D8C .Q
  5E3C .O        5E84 .MES      5F06 .U        5F2E .V
  5F38 @Z        5FAA @H

COMMON-BLOCKS:
NONE

UNDEFINED:
NONE
```

Appendix G. BMPD, a BASIC program for general operation of the computer-interface combination.

Example of how to use the BMPD BASIC program. User input is underlined.

```
*AC BMPD,2D1
*AC DATAFL,7D1
*DO MIDAS,D1
  BIAS 3572
  END 3882
  BIAS 3882
BASIC 03-04
LOAD 2
BASIC 03-04
RUN
GIVE TITLE
  TEST OF E BALANCE
SET INITIAL CONDITIONS, Y OR N?
Y
GIVE COMMAND
1T000000R
SET INITIAL CONDITIONS, Y OR N?
N
GIVE INPUT TO BE READ, FINISH WITH -1
0
GIVE INPUT TO BE READ, FINISH WITH -1
1
GIVE INPUT TO BE READ, FINISH WITH -1
-1
 GIVE NUMBER OF READINGS & TIME BETWEEN READINGS
 100,10
NUMBER OF READINGS IS 100
NUMBER OF INPUTS IS 2
TIME DELAY BETWEEN BATCHES OF READINGS IS 10  SECONDS
INPUTS ARE :
  0
  1
1 .126658      .1454E-2      9
2 .126637      .1455E-2      19
3 .126666      .1454E-2      29
4 .12664       .1454E-2      39
5 .126658      .1455E-2      49
6 .126637      .1454E-2      59
7 .12672       .1455E-2      69
8 .126691      .1454E-2      79
```

Sample output file from BMPD program.

```
  TEST OF E BALANCE
* IT000000R
 NUMBER OF READINGS IS 100
 NUMBER OF INPUTS IS 2
 TIME DELAY BETWEEN READINGS IS 10  SECONDS
INPUTS ARE :
0
1
1 .126658 .1454E-2 9
2 .126637 .1455E-2 19
3 .126666 .1454E-2 29
4 .12664 .1454E-2 39
5 .126658 .1455E-2 49
6 .126637 .1454E-2 59
7 .12672 .1455E-2 69
8 .126691 .1454E-2 79
```

Listing of the BASIC program BMPD.

```
10 REM
20 REM   PROGRAM TO OPERATE MIDAS INTERFACE IN GENERAL FASHION
30 REM     AND WRITE RESULTS ON A DISC FILE IF REQUIRED
40 REM
50 REM     WRITTEN BY BRIAN DICKENS (NBS)
60 REM
70 REM   INITIALISATION AND DEFINITION OF STRINGS
80 REM
90 Z9=0
100 P9=7
110 DIM J$(20)
120 DIM Z$(6)
130 DIM W$(1)
140 W$="M"
150 DIM E$(30)
160 E$="U
170 DIM O$(1)
180 DIM L$(80),A$(1),C$(20),R$(9),T$(20)
190 DIM X(5),Y(8),P$(8),Q$(8)
200 DIM Z(8)
210 PRINT ON (4)"GIVE TITLE"
220 IF Z9<2300 THEN 260
230 WFM P9
240 P9=P9+1
250 Z9=0
260 INPUT ON (5)L$
270 PRINT L$
280 PRINT ON (P9)L$
290 P$="01234567"
300 Q$="33555555"
310 O$="<35>"
320 T$="<35>IQ:;<=>?R"
330 U=36
340 X(1)=0
350 X(2)=0
360 X(3)=.1E-5
370 X(4)=0
380 X(5)=.1E-3
390 REM
400 REM                    PRE-SET MIDAS
410 REM
420 PRINT ON (4)"SET INITIAL CONDITIONS, Y OR N?"
430 INPUT ON (5)A$
440 IF A$="N" THEN 560
450 PRINT ON (4)"GIVE COMMAND"
460 INPUT ON (5)C$
470 PRINT O$;C$
480 PRINT ON (P9)O$;C$
490 L=0
500 CALL 1,U,O$,R$,L,K
510 CALL 1,U,C$,R$,L,K
520 GOTO 420
530 REM
540 REM                    GET READY TO READ INPUTS ON ANALOG SCANNER
550 REM
560 C$="<35>K0MP303*T"
```

```
570 FOR N=1 TO 8
580 PRINT ON (4)"GIVE INPUT TO BE READ, FINISH WITH -1"
590 INPUT ON (5)Y(N)
600 Y(N)=Y(N)+1
610 IF Y(N)<=0 THEN 630
620 NEXT N
630 N=N-1
640 PRINT ON (4)" GIVE NUMBER OF READINGS & TIME BETWEEN READINGS"
650 INPUT ON (5)M,S
660 PRINT ON (4)"NUMBER OF READINGS IS";M
670 PRINT ON (4)"NUMBER OF INPUTS IS";N
680 PRINT ON (4)"TIME DELAY BETWEEN BATCHES OF READINGS IS";S;" SECONDS"
690 PRINT ON (P9)" NUMBER OF READINGS IS";M
700 PRINT ON (P9)" NUMBER OF INPUTS IS";N
710 PRINT ON (P9)" TIME DELAY BETWEEN READINGS IS";S;" SECONDS"
720 PRINT ON (4)"INPUTS ARE :"
730 PRINT ON (P9)"INPUTS ARE :"
740 FOR J=1 TO N
750 I=Y(J)
760 PRINT P$(I,I)
770 PRINT ON (P9)P$(I,I)
780 NEXT J
790 REM
800 REM                    PERFORM READINGS
810 REM
820 FOR N1=1 TO M
830 L=6
840 Z$=""
850 REM
860 REM                    GET TIME
870 REM
880 CALL 1,U,T$,Z$,L,K
890 R1=VAL(Z$)
900 GOSUB 1840
910 W=R1
920 REM
930 REM                   LOOP OVER INPUTS
940 REM
950 FOR J=1 TO N
960 I=Y(J)
970 REM
980 REM                    SET UP TO TRIGGER DVM
990 REM
1000 C$(1,2)="<35>K"
1010 C$(3,3)=P$(I,I)
1020 C$(4,7)="MP31"
1030 C$(8,8)=Q$(I,I)
1040 C$(9,10)="*T"
1050 C$(11,20)="
1060 L=0
1070 REM
1080 REM                   TRIGGER
1090 REM
1100 CALL 1,U,C$,J$,L,K
1110 REM
1120 REM                   WAIT
```

```
1130 REM
1140 CALL 1,U,W$,J$,L,K
1150 IF K=3 THEN 1140
1160 L=9
1170 R$=""
1180 REM
1190 REM                    UNLOAD
1200 REM
1210 CALL 1,U,E$,R$,L,K
1220 REM
1230 REM                    WAIT
1240 REM
1250 L=0
1260 CALL 1,U,W$,J$,L,K
1270 IF K=3 THEN 1260
1280 REM
1290 REM                    DECODE DVM READING
1300 REM
1310 R=VAL(R$)
1320 N5=0
1330 IF R<0 THEN N5=1
1340 R=ABS(R)
1350 R3=INT(R/100)
1360 R1=R-R3*100
1370 R2=R1-INT(R1/10)*10
1380 R=R3*X(R2)
1390 IF N5=1 THEN R=-R
1400 Z(J)=R
1410 NEXT J
1420 REM
1430 REM                    PRINT 1, 2 OR 3 VALUES
1440 REM
1450 IF N>1 THEN 1490
1460 PRINT ON (P9)N1;Z(1);W
1470 PRINT ON (4) N1;Z(1),W
1480 GOTO 1620
1490 IF N>2 THEN 1530
1500 PRINT ON (P9)N1;Z(1);Z(2);W
1510 PRINT ON (4) N1;Z(1),Z(2),W
1520 GOTO 1620
1530 IF N>3 THEN 1580
1540 PRINT ON (P9)N1;Z(1);Z(2);Z(3);W
1550 PRINT ON (4) N1,Z(1),Z(2),Z(3),W
1560 GOTO 1620
1570 REM
1580 REM                GENERAL PRINTOUT
1590 REM
1600 PRINT ON (4) N1;Z(1);Z(2);Z(3);Z(4);Z(5);Z(6);Z(7);Z(8);W
1610 PRINT ON (P9)N1;Z(1);Z(2);Z(3);Z(4);Z(5);Z(6);Z(7);Z(8);W
1620 REM
1630 REM
1640 REM
1650 L=0
1660 CALL 1,U,C$,R$,L,K
1670 CALL 1,U,W$,J$,L,K
1680 IF K=3 THEN 1670
```

```
1690 REM
1700 REM                WAIT LOOP TILL TIME TO BEGIN NEXT BATCH OF READING
1710 REM
1720 R$=""
1730 L=6
1740 REM
1750 REM                GET TIME
1760 REM
1770 CALL 1,U,T$,R$,L,K
1780 R1=VAL(R$)
1790 GOSUB 1840
1800 IF (R1-W)<S THEN 1770
1810 NEXT N1
1820 GOTO 1930
1830 REM
1840 REM        SUBROUTINE TO TIME IN SECONDS
1850 REM
1860 R2=INT(R1/10000)
1870 R3=INT((R1-10000*R2)/100)
1880 R4=INT(R1-10000*R2-100*R3)
1890 R1=R2*3600+R3*60+R4
1900 RETURN
1910 REM
1920 REM                END OF PROGRAM
1930 REM
1940 WFM P9
1950 GOTO 210
1960 END
```

Listing of MIDAS file, used as DO MIDAS,D1 to allow BASIC programs such as BMPD access to the MIDAS interface between the computer and experiment.

```
AC MIDOBJ.1D1
BI 0
LO 1
AS 120
RN BASIC.D1
TR
 END
```

Appendix H. PABS, a FORTRAN program for generalized plotting.

File SETPAB, used as DO SETPAB,D1 to make file assignments and to give instructions for plotting programs.

```
AS 120
AS 220
AS 320
AS 420
AS 520
AS 620
AS 762
AC LU8,8D1
AC LU9,9D1
AC VPLOT,AD1
AS B20
AS C20
AS E20
AS F20
 ASSIGN INPUT FILE AS UNIT B THEN 'RUN PABS' THEN 'RUN VCPY'
TR
```

Sample showing how to run the general plotting program PABS. User input is underlined. Begin with DO SETPAB,DO then assign input file as unit B (e.g., AC INPUT,BDO), then type RUN PABS,DO.

```
PLOT PROGRAM:
READS FROM FILE 11 AND WRITES INTERMEDIATE FILE ON
FILE  8
RECORD LENGTH OF FILE  8 SHOULD BE 16 IN THIS VERSION
INPUT IS IN FREE FORMAT
GIVE: NUMBER OF LINES TO SKIP AS TEXT AFTER ALLOWING
        FIRST LINE TO BE A TITLE
      TOTAL NUMBER OF ARGUMENTS IN ONE LINE
      ORDINAL NUMBER OF ARGUMENT TO BE CALLED  = X =
      DITTO FOR Y1,Y2,Y3,Y4,Y5, STOPPING WITH 0 IF <5
      THEN NUMBER OF POINTS TO BE READ IN
      0 = UPTO EOF, MAXIMUM OF 500.
!
 1 20 5 2 6 9 16 7 0 0

GIVE NUMBER OF POINTS (IF ANY) TO BE USED IN INTERPOLATING
0 =  NO INTERPOLATION, BUT DRAW LINES BETWEEN POINTS PLOTTED
-1 =  DRAW NO LINES AT ALL, JUST PLOT POINTS
-2 =  DRAW NO POINTS AT ALL, JUST DRAW LINES
-3 AND BELOW =  NO POINTS BUT DRAW SMOOTHED CURVES
                WITH -N POINTS USED IN INTERPOLATION
 FREE FORMAT

 !
0

GIVE LENGTH OF X AND Y AXES IN INCHES
 0  =  7. FOR X
 0  =  9.48 FOR Y
X GOES ALONG LENGTH OF ROLL OF PAPER, Y GOES ACROSS
GIVE X VALUE FIRST
!
0 0
GIVE AXIS LABELS AS TWO SETS OF 12 CHARACTERS
ONE PER LINE WITH Y AXIS FIRST


SAMPLE WT
GIVE SCALING FACTORS FOR INPUT ARRAYS
FIRST FOR X ARRAY AND THEN FOR EACH Y ARRAY
!
-.001 1 1 -1 -1 .1

ENTER 1 FOR VERSATEC SCALING, NICE NUMBERS ON AXES
      0 FOR FILL THE PAGE REGARDLESS OF AXIS NUMBERING
     -1 FOR READ IN YOUR OWN PLOT SCALE FACTOR
IF FILL PAGE, GIVE CONTRACTION FACTOR (1.= EXACT FIT OF PLOT
 TO PLOT AREA), INPUT OF 0 GIVES 1.0
 GIVE YOUR PLOT SCALE IF YOU SAID YOU HAD ONE
ALSO GIVE 1 FOR MODE DUMPS OR 0 FOR NO MODE DUMPS
!
1 0

TITLE OF INPUT FILE IS
NBS 1475 PE IN VAC 4/4/78
    1 LINES OF TEXT WILL BE SKIPPED AFTER TITLE
```

```
  500 POINTS MAX WILL BE READ IN
   20 ARGUMENTS PER LINE
X IS ARGUMENT NUMBER  5
Y(1)  IS ARGUMENT NUMBER  2
Y(2)  IS ARGUMENT NUMBER  6
Y(3)  IS ARGUMENT NUMBER  9
Y(4)  IS ARGUMENT NUMBER 16
Y(5)  IS ARGUMENT NUMBER  7
|
|
|
```

Sample input file for PABS program. This file corresponds to the example of user input and helps to explain the strange input used there.

```
NBS 1475 PE IN VAC 4/4/78

 1 66.185  0.374     3559.   0.233949E+06   0.98
                    410.55  0.01     -0.2769E+02       0.6270E-01
                                     0.234211E+06   0.139407E+02    82
                    418.38  0.02     -0.4807E+02       0.5493E-01
                                     0.233688E+06   0.121656E+02    81
 2 60.586  0.415     4379.   0.196729E+06   9.99
                    418.31  0.02     -0.4280E+02       0.5493E-01
                                     0.196430E+06   0.121656E+02    81
                    412.36  0.02     -0.2919E+02       0.4754E-01
                                     0.197029E+06   0.104863E+02    81
 3 65.795  0.424     5196.   0.174084E+06  15.48
                    412.57  0.02     -0.2610E+02       0.4754E-01
                                     0.174446E+06   0.104863E+02    81
                    420.42  0.03     -0.4508E+02       0.9491E-01
                                     0.173722E+06   0.209297E+02    80
 4 64.515  0.626     6013.   0.139261E+06  23.91
                    420.22  0.02     -0.4003E+02       0.9491E-01
                                     0.138954E+06   0.209297E+02    80
                    414.34  0.02     -0.2682E+02       0.5814E-01
                                     0.139567E+06   0.128613E+02    81
 5 65.877  0.451     6833.   0.118442E+06  28.95
                    414.46  0.02     -0.2391E+02       0.5821E-01
                                     0.118781E+06   0.128892E+02    81
                    422.32  0.01     -0.4123E+02       0.6803E-01
                                     0.118102E+06   0.196128E+02    82
 6 67.490  0.739     7657.   0.868738E+05  36.59
                    422.24  0.01     -0.3503E+02       0.8792E-01
                                     0.866649E+05   0.195786E+02    82
                    416.38  0.01     -0.2313E+02       0.8543E-01
                                     0.870826E+05   0.189917E+02    82
 7 60.651  0.546     8480.   0.684536E+05  41.05
                    416.35  0.01     -0.2155E+02       0.8543E-01
                                     0.686979E+05   0.189917E+02    82
                    424.33  0.02     -0.3576E+02       0.5838E-01
                                     0.682093E+05   0.128953E+02    81
 8 70.924  0.654     9299.   0.413622E+05  47.61
                    424.13  0.02     -0.3034E+02       0.5845E-01
                                     0.411567E+05   0.129233E+02    81
                    418.31  0.02     -0.1972E+02       0.5194E-01
                                     0.415677E+05   0.114989E+02    81
 9 64.354  0.507    10118.   0.258236E+05  51.37
                    418.34  0.02     -0.1787E+02       0.5169E-01
                                     0.261622E+05   0.114740E+02    81
                    426.22  0.02     -0.3029E+02       0.6269E-01
                                     0.254650E+05   0.138282E+02    81
10 62.347  0.652    10935.   0.272182E+04  55.97
                    426.13  0.02     -0.2607E+02       0.6269E-01
                                     0.246120E+04   0.138282E+02    81
                    420.25  0.02     -0.1782E+02       0.4549E-01
                                     0.298245E+04   0.101117E+02    82
11 68.952  0.298    11857.  -0.121868E+05  60.58
                    420.33  0.02     -0.1426E+02       0.5608E-01
                                    -0.118062E+05   0.149867E+02    82
                    436.03  0.02     -0.4317E+02       0.7241E-01
```

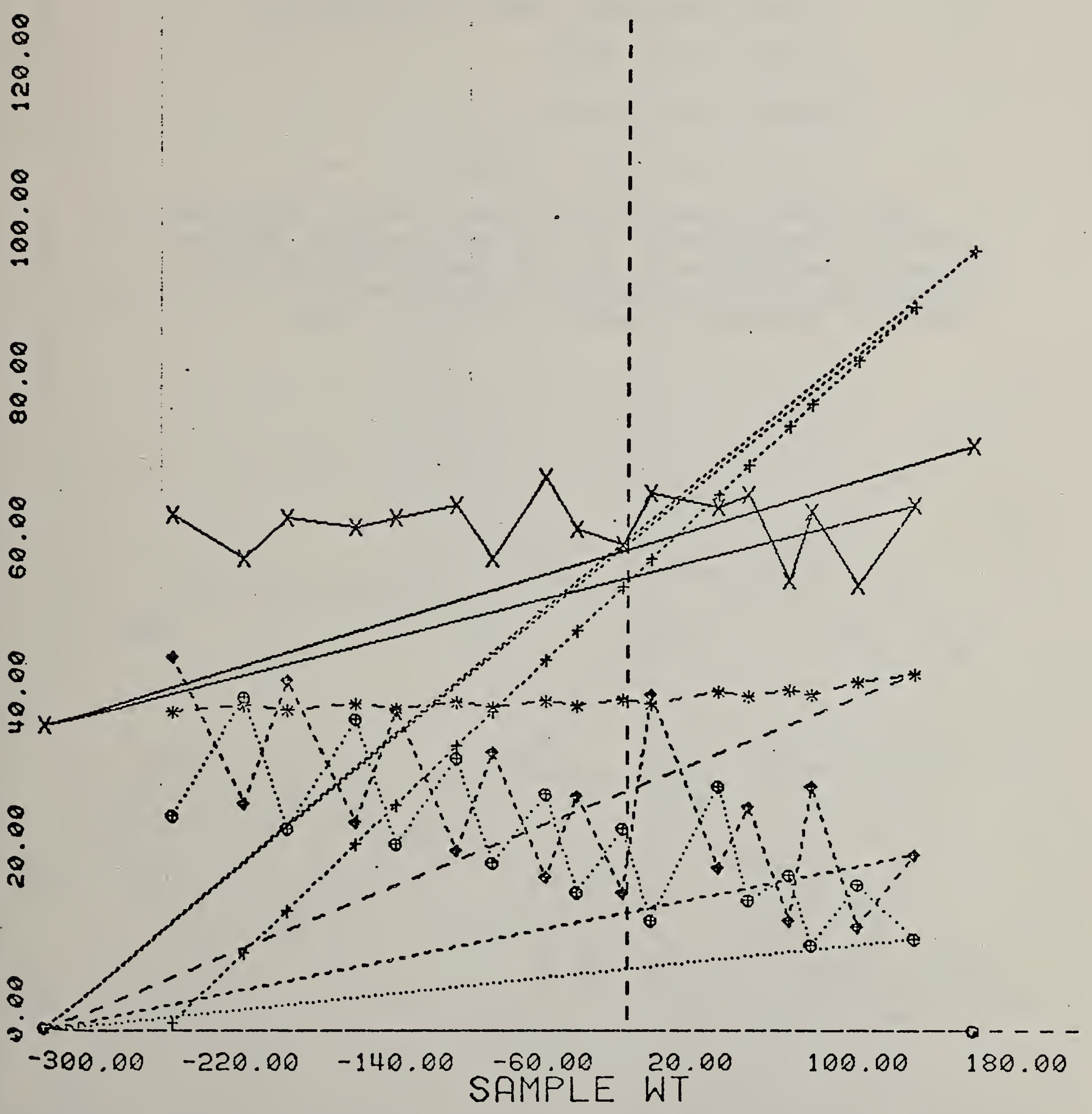
```
                                  -0.125674E+05   0.192291E+02   81
12 67.287  0.633     12773.  -0.464356E+05  68.87
                     435.88  0.02     -0.3134E+02     0.5665E-01
                                  -0.466910E+05   0.129360E+02   81
                     430.02  0.02     -0.2105E+02     0.5772E-01
                                  -0.461803E+05   0.127505E+02   81
13 68.851  0.546     13591.  -0.618970E+05  72.61
                     430.14  0.01     -0.1680E+02     0.5772E-01
                                  -0.616601E+05   0.127505E+02   81
                     437.96  0.01     -0.2888E+02     0.6594E-01
                                  -0.621340E+05   0.146792E+02   82
14 57.766  0.853     14416.  -0.821599E+05  77.52
                     437.84  0.01     -0.2003E+02     0.6602E-01
                                  -0.822975E+05   0.147109E+02   82
                     431.98  0.02     -0.1426E+02     0.5200E-01
                                  -0.820224E+05   0.114701E+02   81
15 66.534  0.431     15330.  -0.939407E+05  80.37
                     432.04  0.02     -0.1110E+02     0.6395E-01
                                  -0.936093E+05   0.169244E+02   81
                     447.86  0.01     -0.3146E+02     0.1026E+00
                                  -0.942721E+05   0.272634E+02   81
16 57.284  0.912     16247.  -0.117152E+06  85.59
                     447.72  0.01     -0.1872E+02     0.8340E-01
                                  -0.117293E+06   0.184715E+02   81
                     441.69  0.01     -0.1336E+02     0.3890E-01
                                  -0.117012E+06   0.861646E+01   81
17 67.444  0.768     18184.  -0.145636E+06  92.89
                     457.31  0.02     -0.1185E+02     0.6269E-01
                                  -0.146325E+06   0.167265E+02   81
                     467.66  0.07     -0.2260E+02     0.6370E-01
                                  -0.144946E+06   0.169782E+02   81
0   39. 0 0  0.300E+06    0.0
0 0 0 0
0 0 0
0 0 0 0
0 0 0
0   75. 0 0 -0.175E+06  100.0
0 0 0 0
0 0 0
0 0 0 0
0 0 0
```

Output of PABS program corresponding to sample input.

```
TITLE OF INPUT FILE IS
NBS 1475 PE IN VAC 4/4/78
   1 LINES OF TEXT WILL BE SKIPPED AFTER TITLE
  500 POINTS MAX WILL BE READ IN
   20 ARGUMENTS PER LINE
X IS ARGUMENT NUMBER  5
Y(1) IS ARGUMENT NUMBER  2
Y(2) IS ARGUMENT NUMBER  6
Y(3) IS ARGUMENT NUMBER  9
Y(4) IS ARGUMENT NUMBER 16
Y(5) IS ARGUMENT NUMBER  7

SCALE FACTORS APPLIED TO INPUT ARRAYS ARE:
0.100000E-02 FOR X ARRAY
0.100000E+01 FOR Y ARRAY NUMBER  1

0.100000E+01 FOR Y ARRAY NUMBER  2

0.100000E+01 FOR Y ARRAY NUMBER  3

0.100000E+01 FOR Y ARRAY NUMBER  4

0.100000E+00 FOR Y ARRAY NUMBER  5
LET VERSATEC SCALE PLOT
NO DUMPS OF MODE TABLE

SCALED PLOT ARRAYS
```

X	Y1	Y2	Y3	Y4	Y5
-0.233949E+03	0.661850E+02	0.980000E+00	0.276900E+02	0.480700E+02	0.418550E+02
-0.196729E+03	0.605860E+02	0.999000E+01	0.428000E+02	0.291900E+02	0.418310E+02
-0.174084E+03	0.657950E+02	0.154800E+02	0.261000E+02	0.450900E+02	0.412570E+02
-0.139261E+03	0.645150E+02	0.239100E+02	0.400300E+02	0.266200E+02	0.420220E+02
-0.118442E+03	0.658770E+02	0.289500E+02	0.239100E+02	0.412300E+02	0.414460E+02
-0.868738E+02	0.674900E+02	0.355900E+02	0.350300E+02	0.231300E+02	0.422240E+02
-0.684536E+02	0.606510E+02	0.410500E+02	0.215500E+02	0.357600E+02	0.416350E+02
-0.413622E+02	0.709240E+02	0.476100E+02	0.303400E+02	0.197200E+02	0.424130E+02
-0.258236E+02	0.643540E+02	0.513700E+02	0.178700E+02	0.302900E+02	0.418340E+02
-0.272182E+01	0.623470E+02	0.569700E+02	0.260700E+02	0.178200E+02	0.426130E+02
0.121868E+02	0.689520E+02	0.605800E+02	0.142600E+02	0.431700E+02	0.428330E+02
0.464356E+02	0.672870E+02	0.628700E+02	0.313400E+02	0.210500E+02	0.435880E+02
0.618970E+02	0.688510E+02	0.726100E+02	0.168000E+02	0.288800E+02	0.430140E+02
0.621599E+02	0.577660E+02	0.775200E+02	0.200300E+02	0.142500E+02	0.437940E+02
0.939487E+02	0.665340E+02	0.883700E+02	0.111000E+02	0.314600E+02	0.432040E+02
0.117152E+03	0.572840E+02	0.859300E+02	0.107200E+02	0.133600E+02	0.447720E+02
0.145636E+03	0.674440E+02	0.928900E+02	0.113500E+02	0.226800E+02	0.457310E+02
-0.300000E+03	0.390000E+02	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
0.175000E+03	0.750000E+02	0.100000E+03	0.000000E+00	0.000000E+00	0.000000E+00

This is an example of a plot by PABS from a prescaled TGRUNF file. The input file was made with the program FCOPYP (see later in this Appendix). The 'extra' lines are a result of the scaling method. The plot is intended to be retraced; the utility of the method is that many plots, all on the same scale so that they can easily be compared, can be made entirely under computer control.



The following is an unscaled PABS plot made using the same data as in the scaled plot example. There are no extraneous lines but the plot is on a semiarbitrary scale, the sole requirement being rounded numbers in the axis markings.

The symbols are:

x curve 1 (activation energy here)

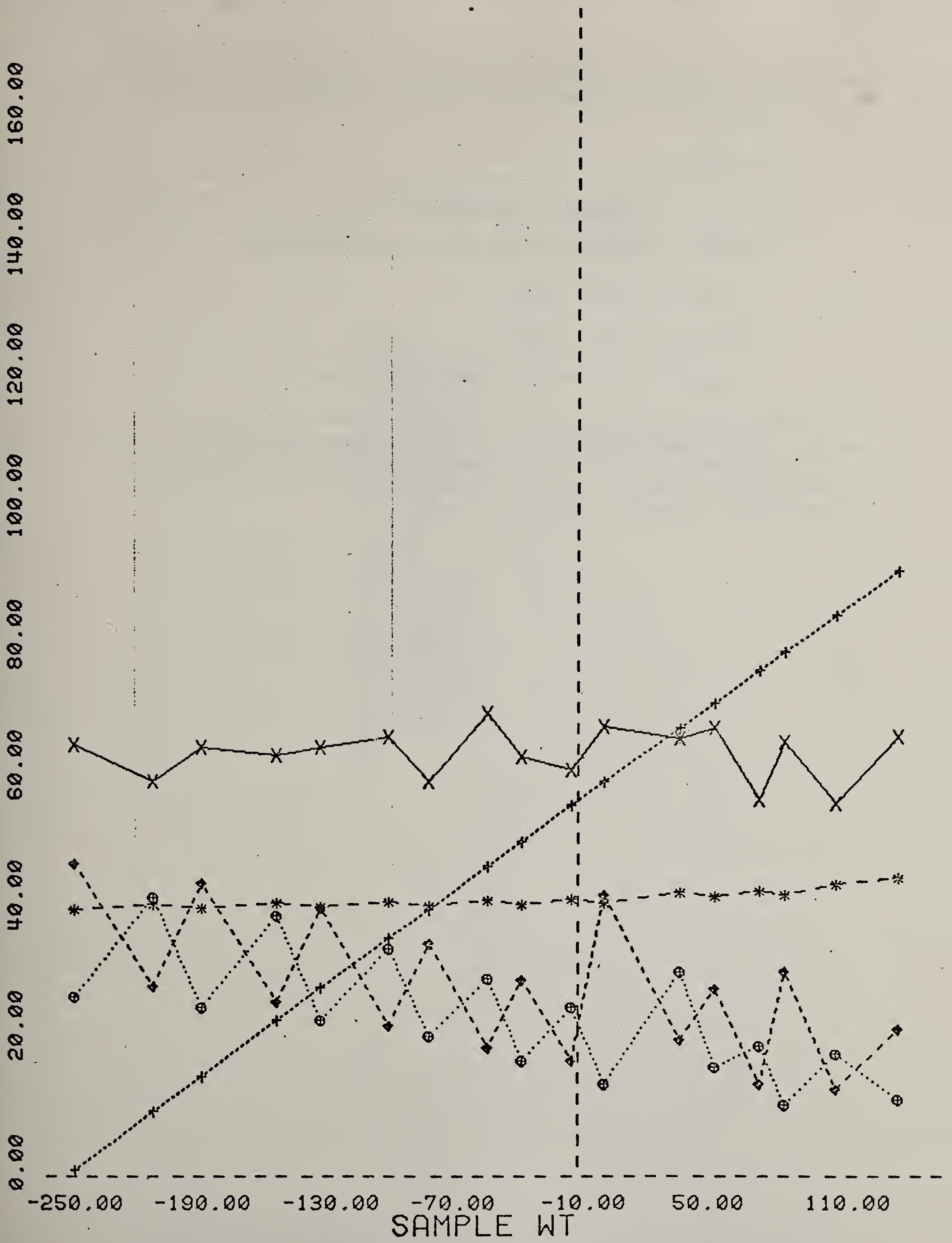
+ curve 2 (percentage extent of reaction here)

+ curve 3 (rate 1 here)

⊕ curve 4 (rate 2 here)

* curve 5 (one tenth of temperature 1 here)

The program will only plot 5 curves; therefore temperature 2 is omitted. The temperature plotted has been divided by 10 to make it fit on the same scale as E, etc. The sample weight has been multiplied by -0.001 to transform it from microvolts to millivolts and to make the chronological sequence from left to right.


0.00
20.00
40.00
60.00
80.00
100.00
120.00
140.00
160.00
-250.00
-190.00
-130.00
-70.00
-10.00
50.00
110.00
SAMPLE WT

An unscaled PABS plot made using the same data as in the input example. There are no extraneous lines but the plot is on a semi-arbitrary scale, the sole requirement being rounded numbers in the axis markings.

The symbols are:

x curve 1 (activation energy here)

+ curve 2 (percentage extent of reaction here)

+ curve 3 (rate 1 here)

⊕ curve 4 (rate 2 here)

* curve 5 (one tenth of temperature 1 here)

The program will only plot 5 curves; therefore temperature 2 is omitted. The temperature plotted has been divided by 10 to make it fit on the same scale as E, etc. The sample weight has been multiplied by -0.001 to transform it from microvolts to millivolts and to make the chronological sequence from left to right.

This file demonstrates a somewhat long way of automatically plotting several files without user intervention. The sequence AS 520 to RUN VCPY is repeated for each plot, and the fifth line, AC PVCAEP,B, is replaced by a reference to the file of interest. These 11 lines can be read into the system program EDIT and be written out to another file after line 5 has been replaced. In this way you can build up a large file to make many plots. This example as given will produce 2 plots, one of PVCAEP and one of PVCCEP. (If no disc reference is given, as in AC PVCAEP,B, the default option is used. In this example, the default disc is DO.)

```
AS 520
AS 762
AC 5PABS,5D0
AC LU8,8D0
AC PVCAEP,B
RUN PABS6
AC VCOPY5,5
AC BLANKS,F
CP F,5,80,A
AC LU9,9
RUN VCPY
AS 520
AS 762
AC 5PABS,5D0
AC LU8,8D0
AC PVCCEP,B
RUN PABS6
AC VCOPY5,5
AC BLANKS,F
CP F,5,80,A
AC LU9,9
RUN VCPY
ZU
TR
```

The following seven tables show how to produce scaled plots of many activation files energy automatically. This table shows a driving file, PLOTTH, which will plot the data in files PPO%05 and PPVAC1. The file is used as follows:

```
*AC PLOTTH, EDO
*TR E
```

Contents of the file PLOTTH

```
AC COPYZU.CD0
TR C
AC PP0%05.7D0
RUN FCOPYP.D0
AC DSETPA.CD0
TR C
AC DOPABS.CD0
TR C
AC COPYZU.CD0
TR C
AC PPVAC1.7D0
RUN FCOPYP.D0
AC DSETPA.CD0
TR C
AC DOPABS.CD0
TR C
ZU
TR
```

Contents of the file COPYZU, needed for automatic scaled plotting.

```
AS 120
AS 220
AS 320
AS 420
AC FCOPY5.5D0
AS 620
AC JUNK.8D1
AS 920
AS A20
AS B20
AS D20
AS F20
TR E
```

Listing of the FORTRAN program FCOPYP, needed for automatic plotting. This program copies the input file and adds user specified values as the initial and final values of both the weight and the activation energy. The source language is in file FCOPYR.

```
$ASSM
FCOPYR PROG
$FORT
C
C  PROGRAM TO MAKE TG DATA SUMMARY FILES PLOT ON STANDARD SCALE
C
      DIMENSION NLINE(40)
      DOC1=0.
      DOC2=100.
      WRITE(4,60)
   60 FORMAT(' GIVE ON SEPARATE LINES:'
     &/'   INITIAL SAMPLE WEIGHT'
     &/'   FINAL SAMPLE WEIGHT'
     &/'   LOWEST ACTIVATION ENERGY'
     &/'   HIGHEST ACTIVATION ENERGY')
      READ(5,70) SWT1,SWT2,EL,EH
   70 FORMAT(F10.0)
      DO 10 I=1,1000
      READ(7,20,END=30)NLINE
   20 FORMAT(40A2)
      WRITE(6,20)NLINE
      WRITE(8,20)NLINE
   10 CONTINUE
   30 WRITE(8,40)EL,SWT1,DOC1
   40 FORMAT(' 0 ',F5.0,' 0 0 ',E10.3,F7.1
     &/' 0 0 0 0'
     &/' 0 0 0'
     &/' 0 0 0 0'
     &/' 0 0 0')
      WRITE (8,40)EH,SWT2,DOC2
      END FILE 8
      STOP
      END
```

Memory map for the FCOPYR program.

```
REL PROGS:
  3576 FCOPYR   3890 .S      38C6 .MES    393C .U
  396C .V       397A @Z      39A0 @H      4BEA

ABS PROGS:
NONE

ENTRY-POINTS:
  3892 .S       38C8 .MES    394A .U      3972 .V
  397C @Z       39EE @H

COMMON-BLOCKS:
NONE

UNDEFINED:
NONE
```

Sample input to the FCOPYP program to produce a scaled plot. The quantities are initial and final sample weights and initial and final activation energies. They set extreme values for the x and y axes respectively.

```
300000.
-175000.
 39.
75.
```

Contents of the file DSETPA, used in automatic plotting.

```
AS 120
AS 220
AS 320
AS 420
AS 520
AS 620
AS 762
AC LU8,8D0
AC LU9,9D0
AC VPLOT,AD0
AS B20
AS D20
AS F20
TR E
```

Contents of the file DOPABS, needed in automatic plotting.

```
AC 5PABS,5D0
AS 762
AC LU8,8D0
AC JUNK,BD1
RUN PABS,D0
AC VCOPY5,5D0
AC BLANKS,FD0
CP F,5,80,A
RW 5
AC LU9,9D0
RUN VCPY,D0
TR E
```

Contents of the file 5PABS, needed for automatic plotting.

```
 1 20 5 2 6 9 16 7 0 0
 0
0 0

SAMPLE WT
-.001 1 1 -1 -1 .1
1 0
```

Several blank lines are needed in file VCOPY5 to provide replies to the VCPY program.

Listings of the FORTRAN routines in the program PABS.

```
$ASSM
VPLOT PROG
$FORT
$TRGT 16
C
C  PROGRAM TO PLOT UPTO 5 CURVES ON VERSATEC PLOTTER. DEO VOLENTE
C  PRESENT VERSION HAS 500 POINTS MAX IN EACH CURVE
C
C  WRITTEN BY BRIAN DICKENS. NBS 921-3322
C $$$$$$ VERSION OF FEBRUARY. 1978 $$$$$$$$$$$$$$$$$$$$$$$$$
C
C VERSION 5 UPDATED TO INCORPORATE CHANGES SUGGESTED BY FLOYD MAUER (NBS)
C
C  IT NEEDS THE FOLLOWING SUBROUTINES:
C
C     INTERF
C     MAXMIN
C     MODEP
C     READX
C  AND THE VERSATEC ROUTINES:
C     AXES
C     DRAW
C     MODE
C     NOTE
C     SCAN
C
      INTEGER*2 LABELX(6),LABELY(6)
      DIMENSION NTITLE(35),X(500),Y(500,5),NY(5),TEMP(100)
     &,YSCALE(5),LMASK(5),LSYMBL(5)
     &,XSTEP(1001),SMOOTH(1001)
      DATA LMASK/X'FFFF',X'3333',X'1111',X'0F0F',X'00FF'/
      DATA LSYMBL/24,43,38,33,42/
C
C  INITIALISE I/O UNITS
C
   11 NTTYO = 4
      NTTYI = 5
      NOUT = 7
      IN = 11
C
C  UNIT 8 IS WRITTEN BY SUBROUTINES NOTE (FOR SYMBOL PLOTTING) AND
C     DRAW (FOR LINE DRAWING)
C
      NPLOT = 8
      REWIND NPLOT
      WRITE (NOUT,1)
    1 FORMAT ('1')
C
      SAME = 9999.
      DO 5 I = 1,5
    5 NY(I) = 0
      XLENG = 0.
      YLENG = 0.
C
C  READ IN INPUT FILE INFORMATION
C
```

```
      WRITE (NTTYO,10)IN,NPLOT,NPLOT
   10 FORMAT (' PLOT PROGRAM:'
     &/' READS FROM FILE',I3,' AND WRITES INTERMEDIATE FILE ON'
     &/' FILE',I3
     &/' RECORD LENGTH OF FILE',I3,' SHOULD BE 16 IN THIS VERSION'
     &/' INPUT IS IN FREE FORMAT'
     &/' GIVE: NUMBER OF LINES TO SKIP AS TEXT AFTER ALLOWING '
     &/'          FIRST LINE TO BE A TITLE'
     &/'       TOTAL NUMBER OF ARGUMENTS IN ONE LINE'
     &/'       ORDINAL NUMBER OF ARGUMENT TO BE CALLED  = X = '
     &/'       DITTO FOR Y1,Y2,Y3,Y4,Y5, STOPPING WITH 0 IF <5'
     &/'       THEN NUMBER OF POINTS TO BE READ IN'
     &/'       0 = UPTO EOF, MAXIMUM OF 500.')
      NTEXT = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      NLINE = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NLINE.LE.100) GO TO 14
      WRITE (NTTYO,13)NLINE
   13 FORMAT (/' TOO MANY ARGUMENTS GIVEN FOR INPUT LINE'
     &/' YOU GAVE',I6,' MAXIMUM IS 100'/' TRY AGAIN')
      GO TO 11
   14 NX = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      DO 15 I = 1,5
      NY(I) = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NY(I).LE.0) GO TO 16
   15 CONTINUE
      GO TO 17
   16 I = I-1
   17 CONTINUE
      MY = I
      NTOTAL = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NTOTAL.LE.0) NTOTAL = 500
      IF (MY.LT.1) GO TO 11
C
C  DECIDE HOW TO PLOT CURVES AND WHETHER TO INTERPOLATE
C
      WRITE (NTTYO,18)
   18 FORMAT (/' GIVE NUMBER OF POINTS (IF ANY) TO BE USED IN'
     &,' INTERPOLATING'
     &/' 0 =  NO INTERPOLATION, BUT DRAW LINES BETWEEN POINTS PLOTTED'
     &/' -1 =  DRAW NO LINES AT ALL, JUST PLOT POINTS'
     &/' -2 =  DRAW NO POINTS AT ALL, JUST DRAW LINES'
     &/' -3 AND BELOW =  NO POINTS BUT DRAW SMOOTHED CURVES'
     &/'                 WITH -N POINTS USED IN INTERPOLATION'
     &/'  FREE FORMAT'//)
      NPTS = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
C
C  DECIDE ON AXIS LENGTHS
C
      WRITE (NTTYO,36)
   36 FORMAT (/' GIVE LENGTH OF X AND Y AXES IN INCHES'
     &/'  0  =  7. FOR X'
     &/'  0  =  9.48 FOR Y'
     &/' X GOES ALONG LENGTH OF ROLL OF PAPER, Y GOES ACROSS'
     &/' GIVE X VALUE FIRST'/)
      XLENG = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      YLENG = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
```

```
C
C  GET AXIS LABELS
C
      WRITE (NTTYO,27)
   27 FORMAT (' GIVE AXIS LABELS AS TWO SETS OF 12 CHARACTERS'
     &/' ONE PER LINE WITH Y AXIS FIRST'/)
      READ(NTTYI,28)LABELY,LABELX
   28 FORMAT (6A2)
C
C  GET SCALING FACTORS FOR INPUT ARRAYS
C
      WRITE (NTTYO,138)
  138 FORMAT (/' GIVE SCALING FACTORS FOR INPUT ARRAYS'
     &/' FIRST FOR X ARRAY AND THEN FOR EACH Y ARRAY')
      XSCALE = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      DO 139 J = 1,MY
      YSCALE(J) = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
  139 CONTINUE
C
C  DECIDE ON PLOT SCALING
C
      WRITE (NTTYO,21)
   21 FORMAT (/' ENTER 1 FOR VERSATEC SCALING, NICE NUMBERS ON AXES'
     &/'        0 FOR FILL THE PAGE REGARDLESS OF AXIS NUMBERING'
     &/'       -1 FOR READ IN YOUR OWN PLOT SCALE FACTOR'
     &/' IF FILL PAGE, GIVE CONTRACTION FACTOR (1.= EXACT FIT OF PLOT'
     &/'  TO PLOT AREA), INPUT OF 0 GIVES 1.0'
     &/'  GIVE YOUR PLOT SCALE IF YOU SAID YOU HAD ONE'
     &/' ALSO GIVE 1 FOR MODE DUMPS OR 0 FOR NO MODE DUMPS')
      NICE = READX (IW,NERROR,3,NTTYI,NTTYO,IEOF)
      IF (NICE.GT.1.OR.NICE.LT.-1) NICE = 1
      IF (NICE.EQ.0) CTRACT = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (NICE.EQ.-1) SCALE = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
      IF (ABS(CTRACT).LT.1.E-12) CTRACT = 1.0
      MODER = READX (IW,NERROR,1,NTTYI,NTTYO,IEOF)
C
C  OUTPUT STATUS QUO
C
      READ(IN,25)NTITLE
   25 FORMAT (35A2)
      WRITE (NTTYO,26)NTITLE,NTEXT,NTOTAL,NLINE,NX,(I,NY(I),I = 1,MY)
      WRITE (NOUT,26)NTITLE,NTEXT,NTOTAL,NLINE,NX,(I,NY(I),I = 1,MY)
   26 FORMAT (/' TITLE OF INPUT FILE IS'/35A2
     &/I6,' LINES OF TEXT WILL BE SKIPPED AFTER TITLE'
     &/I6,' POINTS MAX WILL BE READ IN'
     &/I6,' ARGUMENTS PER LINE'
     &/' X IS ARGUMENT NUMBER',I3
     &/(' Y(',I1,') IS ARGUMENT NUMBER',I3))
      WRITE (NOUT,141)XSCALE,(YSCALE(J),J,J = 1,MY)
  141 FORMAT (/' SCALE FACTORS APPLIED TO INPUT ARRAYS ARE:'
     &/E12.6,' FOR X ARRAY'
     &,(/E12.6,' FOR Y ARRAY NUMBER',I3))
      IF (NICE.EQ.0) WRITE (NOUT,31)
      IF (NICE.NE.0) WRITE (NOUT,32)
      IF (MODER.EQ.0) WRITE (NOUT,33)
      IF (MODER.NE.0) WRITE (NOUT,34)
```

```
   31 FORMAT (' MAKE PLOT FILL PLOT PAGE')
   32 FORMAT (' LET VERSATEC SCALE PLOT')
   33 FORMAT (' NO DUMPS OF MODE TABLE')
   34 FORMAT (' GIVE DUMPS OF MODE TABLE')
      IF (NICE.EQ.0) WRITE (NOUT,35)CTRACT
   35 FORMAT (' CONTRACTION FACTOR APPLIED IS',F10.3)
      IF (NICE.EQ.-1) WRITE (NOUT,136)SCALE
C
C  SKIP TEXT
C
      IF (NTEXT.LE.0) GO TO 810
      DO 820 I = 1,NTEXT
  820 READ(IN,25)NTITLE
  810 CONTINUE
  136 FORMAT (' PLOT SCALE FACTOR READ IN IS',F10.4)
C
C  READ IN LINE CONTAINING PLOT VALUES
C
      J = 0
      NPROMP = 1
  150 DO 20 I = 1,NLINE
      IF (I.EQ.1) NPROMP = 3
      TEMP(I) = READX (IW,NERROR,NPROMP,IN,NTTYO,IEOF)
      NPROMP = 1
      IF (IEOF.GT.0) GO TO 45
      IF (NERROR.GT.0) GO TO 150
   20 CONTINUE
      J = J+1
C
C  FILL IN PLOT ARRAYS
C
      X(J) = TEMP(NX)
      X(J) = X(J)*XSCALE
      DO 30 I = 1,MY
      L = NY(I)
      Y(J,I) = TEMP(L)
      Y(J,I) = Y(J,I)*YSCALE(I)
   30 CONTINUE
      IF (J.EQ.NTOTAL) GO TO 45
      IF (J.LT.500) GO TO 150
   45 NE = J
C
C  OUTPUT PLOT ARRAYS
C
      WRITE (NOUT,411)
  411 FORMAT (/' SCALED PLOT ARRAYS'/12X,'X',11X,'Y1',11X,'Y2',11X,'Y3'
     &,11X,'Y4',11X,'Y5'//)
      DO 400 I = 1,J
      WRITE (NTTYO,410)X(I),(Y(I,K),K = 1,MY)
      WRITE (NOUT,410)X(I),(Y(I,K),K = 1,MY)
  410 FORMAT (6E13.6)
  400 CONTINUE
      MCOUNT = 1
      IF (MODER.GT.0) CALL MODEP (MCOUNT)
C
C  SET PLOT DIMENSIONS
```

```
C
      IF (XLENG.GT.0.) CALL MODE (7,XLENG,SAME,SAME)
      IF (YLENG.GT.0.) CALL MODE (7,SAME,YLENG,SAME)
      MCOUNT = 2
      IF (MODER.GT.0) CALL MODEP (MCOUNT)
C
C  FORM TRANSFER FILE FOR PLOTS ON LOGICAL UNIT 8
C
      MYZ = MY-1
      IF (MYZ.LT.1) GO TO 12
C
C  VERSATEC SCALING, ALWAYS DONE.
C
      DO 200 J = 1,MYZ
  200 CALL SCAN (X,Y(1,J),NE,440)
C
C  MAX AND MIN FOR FINAL PLOT, GET OVERALL SCALE
C
   12 CALL SCAN (X,Y(1,MY),-NE,440)
      MCOUNT = 3
      IF (MODER.GT.0) CALL MODEP (MCOUNT)
      CALL MAXMIN (XMAX,XMIN,X,NE)
C
C  TO SCALE HERE OR NOT TO SCALE HERE?
C
      IF (NICE.NE.0) GO TO 301
      CALL MODE (-2,X1,X2,X3)
      XSCALE = ((XMAX-XMIN)/(X1-X2))/CTRACT
      CALL MODE (8,SAME,XSCALE,SAME)
      CALL MODE (-3,Y1,Y2,Y3)
      DO 440 J = 1,MY
      CALL MAXMIN (YMAX,YMIN,Y(1,J),NE)
  440 YSCALE(J) = ((YMAX-YMIN)/(Y1-Y2))/CTRACT
      CALL MAXMIN (YMAX,YMIN,YSCALE,MY)
      CALL MODE (9,SAME,YMAX,SAME)
      GO TO 331
C
C  APPLY READ IN SCALE FACTOR
C
  301 CONTINUE
      IF (NICE.NE.-1) GO TO 331
      CALL MODE (8,SAME,SCALE,SAME)
      CALL MODE (9,SAME,SCALE,SAME)
  331 CONTINUE
      MCOUNT = 4
      IF (MODER.GT.0) CALL MODEP (MCOUNT)
C
C  SET SYMBOL CODES AND DRAW SYMBOLS (VIA SUBROUTINE NOTE)
C
      IF (NPTS.LE.-2) GO TO 307
      CALL MODE (5,SAME,SAME,440.)
      DO 300 J = 1,MY
      CALL NOTE (X,Y(1,J),LSYMBL(J),-NE)
  300 CONTINUE
  307 IF (NPTS.EQ.-1) GO TO 1001
      IF (NPTS.LE.-3) NPTS = -NPTS
```

```
      IF (NPTS.GT.0) GO TO 309
C
C  SET LINE CODES AND DRAW LINES (VIA SUBROUTINE DRAW)
C
      DO 308 J = 1,MY
      L = LMASK(J)
      CALL MODE (10,FLOAT(L),SAME,SAME)
      CALL DRAW (X,Y(1,J),NE,441)
  308 CONTINUE
      GO TO 1001
  309 CONTINUE
C
C  DRAW SMOOTH CURVES HERE
C  FIRST, CALCULATE LENGTH OF STEP
C
      NSTEPS = MIN0(NE*5+1,1001)
      STEP = (XMAX-XMIN)/FLOAT(NSTEPS-1)
C
C  FILL IN X ARRAY
C
      XSTEP(1) = XMIN
      DO 50 I = 2,NSTEPS
   50 XSTEP(I) = XSTEP(I-1)+STEP
C
C  INTERPOLATE Y ARRAY AT EVERY XSTEP VALUE
C
C  DO OVER ALL  = MY =  CURVES
C
      NHALF = NPTS/2
      DO 1000 K = 1,MY
C
C  DO OVER ALL XSTEP POINTS
C
      I = 1
      DO 900 L = 1,NSTEPS
C
C  FIND X POINT CORRESPONDING TO XSTEP POINT UNDER CONSIDERATION
C
      DO 800 J = I,NE
      IF (X(J).GE.XSTEP(L)) GO TO 700
  800 CONTINUE
      J = NE
C
C  TAKE =NHALF= POINTS ON BOTH SIDES OF X(J) IF POSSIBLE
C
  700 I = J
      J = J-NHALF
      IF (J.LT.1) J = 1
      IF ((J+NPTS).GT.NE) J = NE-NPTS
C
C  INTERPOLATE
C
      CALL INTERF (X(J),Y(J,K),XSTEP(L),SMOOTH(L),NPTS)
  900 CONTINUE
C
C  SET LINE CODE
```

```
C
      L = LMASK(K)
      CALL MODE (10,FLOAT(L),SAME,SAME)
      MCOUNT = 5
      IF (MODER.GT.0) CALL MODEP (MCOUNT)
C
C  DRAW LINE
C
      CALL DRAW (XSTEP,SMOOTH,NSTEPS,441)
 1000 CONTINUE
 1001 CONTINUE
C
C  LABEL AXES
C
      CALL AXES (12.2,LABELX,12.2,LABELY)
C
C  FINISH PLOT
C
      CALL DRAW (0.,0.,1,9000)
      CALL DRAW (0.,0.,0,9999)
      WRITE (NOUT,1)
      STOP
      END

$ASSM
INTERF PROG
$FORT
$TRGT 16
      SUBROUTINE INTERF(X,Y,XHIT,S,NPTS)
C
C  SUBROUTINE TO INTERPOLATE FOR VALUE OF Y AT XHIT
C  GOSPEL ACCORDING TO DORN AND MCCRACKIN, "NUMERICAL METHODS WITH FORTRAN
C  CASE STUDIES", WILEY, 1972, PAGE 287
C
      DIMENSION X(NPTS),Y(NPTS)
C
C  XHIT  =  VALUE OF ARRAY X TO INTERPOLATE TO.
C  Y  =  ARRAY USED TO GET INTERPOLATED VALUE  = S =  FOR X VALUE OF XHIT
C  NPTS  =  NUMBER OF POINTS USED IN INTERPOLATION.
C  LINE NUMBERS CORRESPOND TO NUMBERS IN DORN & MCCRACKIN FLOW CHART
C
    2 I = 1
    3 S = 0
    4 J = 1
    5 P = 1.
    6 IF (J-I) 7,8,7
    7 P = P*(XHIT-X(J))/(X(I)-X(J))
    8 IF (J-NPTS) 9,10,9
    9 J = J+1
      GO TO 6
   10 Z = P
   11 S = S+Z*Y(I)
   12 IF (I-NPTS) 13,14,13
   13 I = I+1
      GO TO 4
   14 RETURN
      END
```

```
$ASSM
MAXMIN PROG
$FORT
$TRGT 16
      SUBROUTINE MAXMIN(XMAX,XMIN,X,N)
C
C  SUBROUTINE TO FIND MAX AND MIN VALUES OF X USING N VALUES
C
      DIMENSION X(N)
      XMAX = -1.E16
      XMIN = 1.E16
      DO 10 I = 1,N
      IF (XMAX.LT.X(I)) XMAX = X(I)
      IF (XMIN.GT.X(I)) XMIN = X(I)
   10 CONTINUE
      RETURN
      END

$ASSM
MODEP PROG
$FORT
$TRGT 16
      SUBROUTINE MODEP(M)
C
C  SUBROUTINE TO PRINT CURRENT VALUES IN MODE TABLE OF VERSATEC PLOTTING
C    SOFTWARE
C
      NOUT = 7
      WRITE (NOUT,5)M
    5 FORMAT (//' MODE TABLE AT STAGE NUMBER',I3//)
      DO 20 I = 1,9
      J = -I
      CALL MODE (J,X,Y,Z)
   20 WRITE (NOUT,10)I,X,Y,Z
   10 FORMAT (I6,3F10.3)
      I = 10
      CALL MODE (-10,X,Y,Z)
      WRITE (NOUT,30)I,X,Y,Z
   30 FORMAT (I6,2Z10,F10.3//)
      RETURN
      END
```

```
$ASSM
READX PROG
$FORT
      FUNCTION READX(IWORD,N,NP,NTTYI,NTTYO,IEOF)
C
C $$$$$$$$$$  VERSION OF MAR 9,1977 $$$$$$$$$$
C
C  FUNCTION READX USUALLY RETURNS THE NEXT DECIPHERIBLE NUMBER ON THE
C     INPUT FILE =NTTYI= AS THE VALUE OF THE VARIABLE READX.
C     ANY OUPUT, SUCH AS ERROR MESSAGES OR THE PROMPT CHARACTER,
C     IS ON UNIT NTTYO.
C
C   IW= ALPHANUMERIC INFORMATION OF THE FIRST 1 OR 2 ILLEGAL NON-NUMERICAL
C     INFORMATION ON INPUT FILE NTTYI.
C  N=0 MEANS NUMERICAL INFORMATION SUCCESSFULLY PROCESSED.
C  N=1 CHECK IW WORD FOR NON-NUMERICAL INFORMATION.
C        (READX=1. IN THIS CASE).
C  NP=1  SEND ! AS A PROMPT CHARACTER WHEN NEW RECORD REQUIRED.
C  NP=2  READ A NEW RECORD.
C  NP=3  DO BOTH THE ABOVE.
C  NTTYI=  INPUT UNIT
C  NTTYO=  OUTPUT UNIT FOR PROMPT CHARACTER AND ERROR MESSAGES.
C   IEOF=0   END OF FILE NOT YET REACHED ON UNIT NTTYI.
C   IEOF=1  END OF FILE FOUND ON NTTYI.
C
C   ILLEGAL INFORMATION:
C
C   IF FIRST CHARACTER IS NOT A BLANK, COMMA, POINT, MINUS, OR
C     A NUMBER, THE FIRST 2 ILLEGAL CHARACTERS ARE RETURNED IN =IW=.
C
C   LEGAL INFORMATION*
C
C   LEADING BLANKS ARE IGNORED.
C   LEADING COMMAS ARE IGNORED.
C   DECIMAL POINTS AND EXPONENTIAL TYPE NUMBERS ARE TREATED CORRCECTLY.
C   $ ON THE INPUT FILE IS PRESUMED TO PRECEED AN IN-LINE COMMENT AND
C     CAUSES READING OF A NEW RECORD.
C   NUMBERS ARE TERMINATED BY A BLANK, COMMA, OR BY THE END OF A RECORD.
C
C
C   READX WAS ORIGINALLY WRITTEN BY FRANK MCCRACKIN (NBS) AND WAS MODIFIED
C     BY FRED MOPSIK AND BY BRIAN DICKENS.  (JAN 1977)
C
C
C
       INTEGER BRANCH
       INTEGER*2 CD,DEC,IW2,N1,J,IEXP,IEXP1
       LOGICAL ISIGNE
       DIMENSION CD(80),DEC(10),TEN(10),TENTEN(10)
       DATA DEC/'0','1','2','3','4','5','6','7','8','9'/
       DATA TEN/1.,1.E1,1.E2,1.E3,1.E4,1.E5,1.E6,1.E7,1.E8,1.E9/
       DATA TENTEN/1.,1.E10,1.E20,1.E30,1.E40,1.E50,1.E60,3*1.E70/
       DATA N1/81/
       IF (NP.EQ.2.OR.NP.EQ.3) N1 = 81
       IEOF = 0
```

```
      N = 0
      SIGN = 1.
      X = 0.
      Y = 1.
      YJ = 0.
C
C   FIND FIRST CHARACTER
C
15    IF (N1.GT.80) GO TO 1000
      IF (CD(N1).EQ.'$') GO TO 1000
      ASSIGN 16 TO BRANCH
      GO TO 2000
16    GO TO (40,100,30,20,20,300),IP
20    N1 = N1+1
      GO TO 15
C   SET SIGN
30    SIGN = -1.
      N1 = N1+1
      GO TO 15
C
C   CALCULATE NUMBER
C
40    X = J
50    N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 50
      ASSIGN 70 TO BRANCH
      GO TO 2000
70    GO TO (75,100,205,200,270,270),IP
75    X = X*10.+J
      GO TO 50
C
C   CALCULATE FRACTION
C
100   CONTINUE
110   N1 = N1+1
      IF (N1.GT.80) GO TO 270
      IF (CD(N1).EQ.'E') GO TO 110
      ASSIGN 125 TO BRANCH
      GO TO 2000
125   GO TO (130,3055,205,200,270,270),IP
130   YJ = 10.*YJ+J
      Y = 10.*Y
      GO TO 110
C
C   CALCULATE EXPONENT IEXP
C
200   ISIGNE = .FALSE.
      IEXP = 1
      IEXP1 = 1
      GO TO 210
205   ISIGNE = .TRUE.
      IEXP = 1
      IEXP1 = 1
210   N1 = N1+1
      IF (N1.GT.80) GO TO 250
```

```
      ASSIGN 220 TO BRANCH
      GO TO 2000
220   GO TO (230,250,250,250,250,250),IP
230   IEXP1 = IEXP
      IEXP = J+1
      GO TO 210
250   IF (ISIGNE) GO TO 260
      READX = SIGN*(X+YJ/Y)*TEN(IEXP)*TENTEN(IEXP1)
      N = 0
      RETURN
260   READX = SIGN*(X+YJ/Y)/(TEN(IEXP)*TENTEN(IEXP1))
      N = 0
      RETURN
270   READX = SIGN*(X+YJ/Y)
      N = 0
      RETURN
C
C   RETURN ALPHA ARG
C
300   IW2 = CD(N1)
      N1 = N1+1
      IF (N1.GT.80) GO TO 302
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',' .OR. CD(N1).EQ.'$') GO TO 302
      IW2 = IW2-X'20'+CD(N1)/X'100'
      N1 = N1+1
302   IWORD = IW2
      N = 1
      READX = 1.
      RETURN
3055  WRITE (NTTYO,3056)N1
3056  FORMAT(' 2d . AT COL.',I3)
      GO TO 270
1000  N1 = 1
      IF (NP.EQ.1.OR.NP.EQ.3) WRITE (NTTYO,1020)
1020  FORMAT(' !')
      READ (NTTYI,1010,END = 9000) CD
1010  FORMAT(80A1)
      GO TO 15
C
C   CHARACTER INTERP. ROUTINE
C
2000  DO 2010 J = 1,10
      IF (CD(N1).EQ.DEC(J)) GO TO 2020
2010  CONTINUE
      IP = 6
      IF (CD(N1).EQ.'.') IP = 2
      IF (CD(N1).EQ.'-') IP = 3
      IF (CD(N1).EQ.'+') IP = 4
      IF (CD(N1).EQ.' ' .OR. CD(N1).EQ.',') IP = 5
      GO TO 2030
2020  IP = 1
      J = J-1
2030  GO TO BRANCH
 9000 IEOF = 1
      RETURN
      END
```

Driving program LOPABS which assigns files and runs program LIBLDR to make the PABS plotting program.

```
AC PABS,1D0
AS 220
AS 362
AC PLIB,4D0
AC PABSL5,5D0
AC FORLIB,6D0
AC VPLTB,7D0
AC INTERB,8D0
AC MAXMNB,9D0
AC MODEPB,AD0
AC READXB,BD0
AS C20
AS D20
AS E20
AS F20
RUN LIBLDR,D0
ZU
TR
```

Contents of file PABSL5, which supplies appropriate responses to the LIBLDR program to make the PABS plotting program.

```
OUT 1 PABS
LC 110
BI 3576
LO 7
LI 8
LI 9
LI A
LI B
LI 4
ED 4
ED 6
MAP
MAP 3
XOUT
WF 1
END
```

Memory map of the PABS program.

```
REL PROGS:
  3576 VPLOT    A0AE INTERF   A218 MAXMIN   A310 MODEP
  A4B2 READX    AC76 VPCM1    AC76 AXES     B12A NOTE
  B73A MODE     C09C DRAW     C9E8 SCAN     CF20 FIND
  D2FA VERS     D552 IRAM     D568 IOR2     D5AA IAND2
  D5CC COS      D5EC SIN      D6B2 .I       D6B8 .I2
  D770 .R       D7AC .A       D852 ALOG     D960 EXP
  DA66 MIN0     DA6C MIN02    DA82 .1       DA94 AMOD
  DABE AINT     DB36 $1       DB4E $22      DB82 FLOAT
  DB88 FLOAT2   DBC0 .Y       DBC6 .Y2      DC46 MOD2
  DC74 ABS      DC8A IABS2    DCC4 .W       DD12 .COMP
  DD3E .IIARG   DD78 .RRARG   DDB6 $6       DE00 .RARG
  DE2C $8       DE56 .5       DE5E .ZERO    DE66 .S
  DE9C .P       DF1E .Q       DFCE .O       E016 .MES
  E08C .U       E0BC .V       E0CA @R       E0F0 @Z
  E116 @W       E3C4 @H       F60E

ABS PROGS:
NONE

ENTRY-POINTS:
  A0D2 INTERF   A23C MAXMIN   A334 MODEP    A4D6 READX
  AC9A AXES     B14E NOTE     B75E MODE     C0C0 DRAW
  CA0C SCAN     CF44 FIND     D31E VERS     D552 IRAM
  D58A IOR2     D5AC IAND2    D5CE COS      D5EE SIN
  D6B4 .I       D6BA .I2      D772 .R       D7AE .A
  D854 ALOG     D962 EXP      DA68 MIN0     DA6E MIN02
  DA84 .1       DA96 AMOD     DAC0 AINT     DB39 $1
  DB50 $22      DB84 FLOAT    DB8A FLOAT2   DBC2 .Y
  DBC8 .Y2      DC48 MOD2     DC76 ABS      DC8C IABS2
  DCC6 .W       DD14 .COMP    DD40 .IIARG   DD7A .RRARG
  DDB8 $6       DE02 .RARG    DE2E $8       DE5A .5
  DE62 .ZERO    DE68 .S       DE9E .P       DF20 .Q
  DFD0 .O       E018 .MES     E09A .U       E0C2 .V
  E0CC @R       E0F2 @Z       E118 @W       E412 @H

COMMON-BLOCKS:
  FEEE VPCM1.

UNDEFINED:
NONE
```

H32

Appendix I. Cautionary Remarks

I. Problems with the Apparatus

(a) Occasional delays of two to three weeks in obtaining cylinders of compressed gas.

(b) Drying tubes: no problems.

(c) Gas-flow controller: generally no problems. Cables from interface must not be tugged because miniplug connections at end of cables are not very strong.

(d) Electrobalance: because this is the most sensitive piece of apparatus, it has to perform to the highest specifications.

(i) The contacts round the lamp in the photocell must occasionally be sanded to remove oxidation. The symptoms are unsteadiness and large standard deviations (100 μV?) in the balance.

(ii) Even normal handling of the balance pan can make the flag in the photocell scrape on the photocell housing. The symptom is slow recovery of the balance after it has been disturbed greatly (by a pellet of sample for example). Bend the flag to clear the housing.

(iii)Degradation products sometimes build up on the balance housing side of the expansion joint. Open up the balance housing and carefully burn the products off with a torch or bunsen burner.

(iv) When the balance reading is stable for short periods (30 s) and then changes by up to 300 μV to a new level of temporary stability, it probably needs factory service.

(v) Keep o-ring seal between balance housing and furnace clean and greased with silicone stop-cock grease.

(vi) Run balance with a time constant of 5 s, adjust suppression to give empty bucket weight of -0.16 V to -0.175 V (program stops when sample weight goes below -0.158 V).

(vii)a sample weight of +0.10 to 0.20 V usually seems appropriate.

For 0.10 V, this is (0.10 V + 0.17 V)/0.02(V/mg) = 13.5 mg of sample.

(e) Furnace: the furnace is usually trouble-free. As it ages (> 9 months), occasional flashes during the initial, high voltage (> 75 V) stages of heat-up may indicate weak connections. Generally, run it until it fails. Have a spare on hand.

The best maximum settings on the Kepco power supply are 80 to 85 V and 1.5 to 1.7 amps (these depend on the furnace resistance being ∿56Ω). The best settings on the Leeds and Northorp Electromax III controller are proportional = 120, reset = 50 (maximum possible), approach = 200 (maximum possible) and rate = 0. Note that the proportional range has been increased by a factor of 10 by placing a 100 KΩ resistor between terminals D6 and D7.

With the diffusion pump assembly, the furnace and balance housing will pump down to 3 to 5 μm of Hg. This is achieved by using a flat flange on the balance end of the pyrex furnace housing, by rotating the furnace housing backwards and forwards a few degrees to seal the o-ring well, and, during assembly of the furnace, by expoxying the lead-in wires which supply power to the furnace. A greased Viton-A gasket pressed on a hot press is used to join the two parts of the furnace housing. The thermocouple leads into the balance housing are also sealed with expoxy cement. With the direct access to the mechanical pump, ∿ 10μm of Hg pressure can be attained. This is probably satisfactory for most "vacuum" degradations.

(f) Pressure control: The Granville-Phillips Automatic Pressure Controller is generally trouble-free. Note that the servo-driven exhaust-valve is intended for use on an input line, whereas here it is used on the output line. All the open/close connections inside the controller have been changed. In addition, the number of corrective pulses per unit time for a given pressure deviation has been greatly increased by changing R1-21 (fig. 4-2, Granville Phillips Manual) from 27 KΩ to ∿ 11.7 KΩ. The settings for the APC are Rate = 10, Damping = 10, Reference = 0.8, Range = 10 V.

The MKS pressure measuring unit is also trouble-free. Settings are range multiplier = 1, head range = 1000, and with no capacitors in the scalar module.

The pressure sensor itself performs very well when allowance is made for the non-ideality of the system to be measured. At one point, the head was stuck at a 'zero' of 10.3 mm Hg (at which the pressure is really <5 μm Hg) because there was mercury loose inside the head. In practice this presented no problem, but the unit was later serviced by the factory. The zero depends on the position of the head and is not constant from experiment to experiment.

The servo-driven valve is usually trouble-free. A recent problem was a sudden inability of the apparatus to control the pressure over a long period of time. The pressure would oscillate and at some high pressure (900 to 1000 mm Hg) would blow out one of the greased glass joint connections. In diagnosing the possible cure, we tried rinsing the passage through the valve with toluene, then pumping it down. We also tried heating it for 24 hr at 105 °C in a vacuum oven. These treatments had no effect. Another symptom was slow pump-down through the valve--the valve was either blocked or not fully open. The oil level in the driver was readjusted. The valve then performed well. As it turned out, the new runs were being made at a new temperature, > 400 °C, and part of the trouble arose from the ebb and flow of hot gas into the cool side arm of the furnace. The cooling produced additional pressure changes which were much faster than the pressure controller could handle. Insulating the furnace arm did not cure the problem, but using a mercury/glass tube exhaust system did. Since the exhaust bubbles through the mercury, the pressure will be greater than atmospheric (unless you evacuate the space above the mercury).

(g) Vacuum pumps: no problems.

(h) MIDAS interface: generally no problems, occasionally the computer program will hang in a wait loop, presumably waiting for more transmission from the MIDAS controller. One solution is to pull out the plug at the end of the cable from the computer to MIDAS and then re-insert it. This allows the program to continue, but by this time much or all of the sample has degraded. Nonetheless, the program will, if all the sample has gone, come to a normal halt. The problem has often been cured by pushing all the pins in the plug in the back of the MIDAS firmly through the body of the plug and by firmly seating the plug into the MIDAS connection. The problem may also arise from the MIDAS backplane, which, as of this writing, is out for repair.

(i) Digital volt meter.

This unit has been plagued with troubles. Recently, it worked satisfactorily under program control but not under manual control from the front panel. Previously, it had problems in obeying the range switch--it would move the decimal point but not the digits. After being sent to two local repair companies for a total of three months, it was finally repaired satisfactorily at the factory.

(j) Computer: hardware problems include one bad memory cell long ago and faults in the CPU board and read errors in the hard disc drive. The operating system will sometimes not allow the restart of the overlay programs if they had previously been interrupted and had not come to a normal halt. The symptom is II... (illegal instruction) after you have typed in RUN TGMAIN,D1 (and have correctly assigned all the needed units). The solution is to type SM 0 and then RUN TGMAIN,D1. Sometimes an error message results after the first write to the MIDAS interface if SM 0 is not used. (This zeros the compute memory.) Rewinding the output files (via RW 6 and RW 3) and typing SM 0 and RUN TGMAIN,D1 is the way to proceed.

(k) Rigid discs: Generally trouble-free. Occasionally they cannot be read. These disc drive errors seem to have been due to dirt on the center post, and on bad electrical contact between the static strip and the spinning spindle at the bottom of the drive. Both of these problems can be cleaned by swabbing the parts with isopropyl alcohol.

(l) Flexible disc drives: some are good, some are not. The same applies to the flexible discs themselves. Do not use these unless you have to. Drives B and C seem to be the best, but their level of performance is far below that of the rigid discs. For occasional use they are excellent. When the drives are in good condition, they are excellent even in extended use, but it seems to be difficult to keep drives in good condition. After a few months of continuous use they seem to die with some ailment which requires they be returned to the factory. Within weeks the pressure pad may need to be replaced.

(m) Printer-plotter: generally trouble-free. Watch the level of the printer-liquid, and give up on trying to get the paper to stack neatly (this seems to depend on the paper shipment more than on you). Leaving about an inch of space between the edge of the basket and the paper control spring thing is the best you can do.

(n) CRT terminal: generally trouble free except for intermittant switch bounce (where for example, hitting the A key once sometimes gives AA instead of A). Switch bounce depends in some way on the environment. We have replacement switches for this keyboard. Clean bouncing switches with switch cleaning spray.

(o) Brief discussion of useful FLOXOS commands.

(i) The initial loading of the operating system (FLOXOS) was covered in section I.1. If the FLOXOS system is supposed to be in the computer but for some reason is not running, you can enter DTA, 02D0, ADD and then press the RUN button for a "warm restart" which should provide a running FLOXOS program will all the previous file assignments still operational.

(ii) CH F0: Used when disc F0 is being removed; all current file assignments to that disc will be dropped. The disc will not be written on further and you can use all the information on it by reaccessing the files (use other disc names for F0 as required).

(iii) AL NEWFIL,2D1,1,80: Used to create a new file called NEWFIL on disc D1. This name will be written into the disc directory and no other file with the same name can exist on that disc. If this is already the case, FLOXOS will return a message which says "NA-ERR," i.e., not assigned. In the example above, NEWFIL will be assigned as logical unit 2, and will be written on or read from whenever a computer program refers to unit 2. A space of one cylinder will be reserved on disc D1, and each line written to the file will have a maximum length of 80 characters. This is enough for 144 lines. Data files are usually made with 80 character lines. Programs are stored on files with lines of length 108.

(iv) AC MYFILE,6D0: A file called MYFILE already exists on disc D0. This command makes it logical unit 6 so a computer program can have access to it.

(v) AS 362: This assigns the printer (physical unit 62) to be logical unit 3. Then information written to unit 3 (programs refer to logical units only) will appear on physical unit 62, which is the printer. The CRT is physical unit 20 (an RS232 port on the back of the computer). The assignment 362 can be cancelled by assigning 3 back to the CRT, i.e., AS 320. The other physical unit which is important is number 24, which is attached to the MIDAS interface. 24 is base 16; BASIC and FORTRAN programs refer to this in base 10 as unit 36 (2x16+4). A third (currently unused) RS232 port on the back of the computer is physical unit 22.

(vi) CP 1,2,80,A: copy all (A) of file 1 (previously assigned) to file 2 (also previously assigned). Both files have 80 character lines. Use this command to copy data from the disc to the printer. For example:

```
AC DATA,1D1
AS 362
CP 1,3,80,A
```

CP 1,2,108,A would be used to copy program files, which have a record length of 108.

(vii) DO CMMDS,D0: This tells the operating system FLOXOS to read the instructions written on a file CMMDS which, in this example, is expected to be available on disc D0. These commands can be built up using the EDIT program supplied with FLOXOS.

(viii) EDIT: A text editing program which is the usual channel for getting information from the operator (you) to a file on a disc. Refer to the EDIT manual. EDIT reads unit 1 and writes unit 2. Several other files are used for print out of text or prompt characters, etc. EDIT is easy to use but cannot be summarized briefly.

(ix) RUN TGMAIN,D1: Loads the program TGMAIN from disc D1 into the memory of the computer and starts it running. This is the usual method of starting FORTRAN programs. Any files, including overlay files, needed by the program must be assigned before the RUN command is issued. (This is done via AL and AC commands which are usually already set up in a transfer file such as SETTG if the same files are always needed by a program.)

(x) BASIC: The BASIC interpreter itself is run by typing RUN BASIC. However, it is usually used here to control MIDAS with the BASIC program BMPD. The procedure is the following:

```
AL DATA,7D1,2,80 or AC DATA,7D1
AC BMPD,2D1 (or F1 etc.)
DO MIDAS,D1
```

then, after the BASIC interpreter has replied "BASIC..."

```
LOAD 2
RUN
etc.
```

The readings from the MIDAS interface are written on unit 7, assigned above as DATA in this example.

(xi) CL: completes the transfer of information to a file. Not always needed, cannot always be neglected. Refer to the FLOXOS/FLOS manuals.

(xii) RW 6: Rewind the file assigned to unit 6 so that the next reading or writing takes place at the beginning of the file. AC and AL start at the beginning of the file.

(xiii) WF 6: Write an end-of-file (a sort of magnetic end of information mark) on the file assigned to unit 6.

(xiv) FORTRAN: Refer to the program appendices to see how to compile and execute FORTRAN programs.

Appendix J. Cable diagrams for interface.

TABLE J1

REMOTE CONTROL OF DVM CABLING DIAGRAM FROM MIDAS BCD I/O
J2 PLUG TO DVM J202 PLUG

Comment on DVM function	DVM Function	Pin on DVM Connector J202	Wire Code cable number, color	Pin on MIDAS Connector J2	MIDAS Function	Comment on MIDAS Function
Program	front panel control				P0	
	hold	A10	1Y(ellow)	4	P1	P output register
	system control	B6	1O(range)	3	P2	
	super fast	A7	1R(ed)	2	P4	
	program storage	B15	1Br(own)	1	P8	
Function	DC measurement				Q0	Q output register
	filter	B7	1Gr(ay)	8	Q1	
	AC	A8	1V(iolet)	7	Q2	
	Ohms	A9	1B(lue)	6	Q4	
	Ratio	B8	1G(reen)	5	Q8	
Range	Autorange				R0	R output register
		B12	2R	12	R1	
		B13	2Br	11	R2	
		B14	1Bl(ack)	10	R4	
		B11	1W(hite)	9	R8	
		A15	2O	17	Strobe	S output register
		A14	2Y	18	Ground	
		B10	2G	28	Trigger	

REMOTE READING OF DVM,
CABLE DIAGRAM FROM MIDAS J1 PLUG TO DVM J201 PLUG

DVM Function	DVM Function (decade-bit)	Pin on DVM Connector J201	Wire Code (Cable number, color)	Pin on MIDAS Connector J1	MIDAS Function (decade-bit)	Comment on MIDAS Function
Sign of reading	Sign	A10	1Br(own)	45	P-1	0 for +, 1 for -
				33	1-2	ground =.0
				40	1-4	NC* = 1
				44	1-8	NC = 1
"half digit" (0 or 1 because of over-ranging capability)	100K-1	B8	3Y(ellow)	32	2-1	0 or 1 for decading half digit
				36	2-2	grounded = 0
				39	2-4	grounded = 0
				43	2-8	grounded = 0
Second digit	10K-1	B6	3R(ed)	31	3-1	Second Digit
	10K-4	B7	3O(range)	38	3-4	
	10K-8	39	3G(reen)	42	3-8	
	10K-2	B10	3B(lue)	35	3-2	
Third digit	1K-4	B12	3V(iolet)	37	4-4	Third Digit
	1K-1	B13	3Gr(ay)	30	4-1	
	1K-8	B14	3W(hite)	41	4-8	
	1K-2	B15	3Bl(ack)	34	4-2	
Fourth digit	100-4	B16	2Br	13	5-4	Fourth Digit
	100-1	B17	2R	5	5-1	
	100-8	B18	2O	17	5-8	
	100-2	B19	2Y	9	5-2	
Fifth digit	10-4	B20	2G	14	6-4	Fifth Digit
	10-1	B21	2B	6	6-1	
	10-8	B22	2W	18	6-8	
	10-2	A22	2Bl	10	6-2	
Sixth digit	1-1	A21	2Gr	7	7-1	Sixth Digit
	1-4	A20	2V	15	7-4	
	1-8	19	1Bl	19	7-8	
	1-2	18	1Gr	11	7-2	
				8	8-1	NC — some could be grounded to give an appropriate MIDAS character for spacing (see below)
				12	8-2	NC
				16	8-4	NC
				20	8-8	NC
DVM Overloaded or illegal function specified	NC	A12	1O	28	9-1	0 = ok 1 = NO
				3	9-2	grounded (=0)
				23	9-3	
				21	9-4	
Range Specification	B	17	1V	2	10-2	DVM Range Specification
	C	16	1B	22	10-4	
	A	15	1G	4	10-1	
	D	14	1Y	26	10-8	
Common Ground	Ground	B1	3Br	1	Ground	Common ground
	Ohms	B2				
	+ 5V	B3				
	System Control	B4				
	Filter	B5				
	Signal Integrate	B11				
Read to Transmit Data	Data Ready	A11	1R	48	NC	Start reading data
				47	NC	grounded

*NC = not connected

TABLE J3

MIDAS BCD I/O module jumper positions:

input sense selection = 1
output sence selection = 0
patch to "no E" in output reading

BCD character table hardwired into MIDAS.

BCD	Character
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	:
11	;
12	space
13	-
14	>
15	skip (no printing character)

for positive-true input, open pin is 1.

for ground-true input, grounded pin is 1.

DC volt range specification for Dana 5900 DVM.

Decimal Value	Range
3	0.1V
4	1V
5	10V
6	100V
7	1000V

Example of programming:

Ø begin command stream
M BCD I/O module address in crete
P begin by filling register P
3 3 = 1+2; 1 = hold
        2 = system control
1 1 = 1+0, 1 = filtered input
        0 = DC volts
3 range specification 3 = 0.1V (over-ranges to 0.16V)
* wait after next character is received, continue when BDC I/O module sends "continue" signal to controller
T trigger the DVM to take a reading
U unload the DVM reading.

Sample of reading as output after above commands:

```
  A    BC
-12345603
```

A = DC voltage in microvolts
B = 0 for satisfactory DVM performance
    1 for illegal function or overload
C = 0.1V range used

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 80-2102	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE
User's Manual for Factor-Jump Thermogravimetry Apparatus and Associated Computer Programs, Including a General Plotting Program

5. Publication Date
March 1981

6. Performing Organization Code

7. AUTHOR(S)
Brian Dickens

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as 9.

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The publication describes the operation of the computerized thermogravimetry apparatus. Detailed instructions are given to enable the user to bootstrap the computer and run the various programs. The items covered are 1) the thermogravimetry program TGRUNF, 2) the data file editing program TGEDIT, 3) estimation of trimmed means by TGTRIM, 4) general calculation of activation energy from rate data; probability plots by TGDEPG, 5) the generalized polynominal fitting program POLGEP, 6) publication tables by TABLEP, 7) the BASIC program BMPD which operates the computer interface in a general fashion and 8) user-directed and automated plotting using the generalized plotting program PABS. Appendices on cautionary remarks and cable diagrams are included.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Automated thermogravimetry; data analysis; generalized plotting programs; interdata 7-16 computer; polynominal fitting; user manual.

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
☐ Unlimited	UNCLASSIFIED	337
☐ For Official Distribution. Do Not Release to NTIS	20. SECURITY CLASS (THIS PAGE)	22. Price
☑ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-	UNCLASSIFIED	
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161		

USCOMM-DC