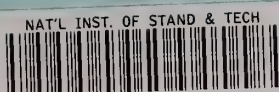


Reference

NBS
Publi-
cations



A11106 034188

NBSIR 83-2703

FASTMENU: A Set of FORTRAN Programs for Analyzing Surface Texture

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Manufacturing Engineering
Washington, DC 20234

July 1983



U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU
OF STANDARDS
LIBRARY
AUG 1 1983

NBSIR 83-2703

**FASTMENU: A SET OF FORTRAN
PROGRAMS FOR ANALYZING SURFACE
TEXTURE**

T. V. Vorburger

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Manufacturing Engineering
Washington, DC 20234

July 1983

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

A
d
m
p
2
a
r
c

ABSTRACT

A set of FORTRAN programs for surface texture analysis is described. These programs were developed for use with a minicomputer that is interfaced to stylus type instruments. The programs 1) perform data acquisition from the stylus instruments, 2) store the data on magnetic disk, and 3) perform statistical analyses for parameters such as the roughness average R_a , rms roughness R_q , and for the autocorrelation function and amplitude density function.

TABLE OF CONTENTS

1.	Introduction	1
2.	The Menu	4
3.	ROUGHNES	8
	3.1 Summary	8
	3.2 Operating System Commands	9
	3.3 ROUGHNES FORTRAN Program	10
	3.4 Flowchart for ROUGHNES	22
	3.5 Example of ROUGHNES Printout	28
4.	STEPHGHT	29
	4.1 Summary	29
	4.2 Operating System Commands	30
	4.3 STEPHGHT FORTRAN Program	31
	4.4 Flowchart for STEPHGHT	39
	4.5 Example of STEPHGHT Printout	45
5.	WHATSON	46
	5.1 Summary	46
	5.2 Operating System Commands	47
	5.3 WHATSON FORTRAN Program	48
	5.4 Flowchart for WHATSON	50
	5.5 Example of WHATSON Printout	51
6.	AVRGRA	52
	6.1 Summary	52
	6.2 Operating System Commands	53
	6.3 AVRGRA FORTRAN Program	54
	6.4 Flowchart for AVRGRA	57
	6.5 Example of AVRGRA Printout	59
7.	SMORGAS	60
	7.1 Summary	60
	7.2 Operating System Commands	62
	7.3 SMORGAS FORTRAN Program	63
	7.4 Flowchart for SMORGAS	69
	7.5 Example of SMORGAS Printout	72
8.	PLOTSVIL	73
	8.1 Summary	73
	8.2 Operating System Commands	74
	8.3 PLOTSVIL FORTRAN Program	75
	8.4 Flowchart for PLOTSVIL	78
	8.5 Example of PLOTSVIL Plot	79

9.	ADF	80
9.1	Summary	80
9.2	Operating System Commands	81
9.3	ADF FORTRAN Program	82
9.4	Flowchart for ADF	88
9.5	Example of ADF Plot	91
10.	ACF	92
10.1	Summary	92
10.2	Operating System Commands	93
10.3	ACF FORTRAN Program	94
10.4	Flowchart for ACF	98
10.5	Example of ACF Plot	101
11.	PSD	102
11.1	Summary	102
11.2	Operating System Commands	103
11.3	PSD FORTRAN Program	104
11.4	Flowchart for PSD	116
11.5	Example of PSD Plot	119
12.	References	120

1. Introduction

Surface texture needs are becoming better understood and more carefully specified for a wide range of industrial parts like ship hulls and propellers [1], automobiles [2], and x-ray mirrors [3]. Consequently, the measurement of surface texture is becoming more sophisticated as its importance for industrial products increases. More and more, the measurement of surfaces by stylus techniques involves digitization of the data and statistical time-series analysis both of which require a computer or microprocessor connected on line to the stylus instrument. Important components of these systems are the programs or software which direct the sequence of measurement operations.

At NBS, we have developed a system called FAST (Facility to Analyze Surface Texture) [4], which includes two commercial stylus instruments interfaced to a minicomputer that controls the data acquisition and analysis through a set of nine FORTRAN programs called FASTMENU. Each of these programs can be executed with a command from the computer console. Their names and functions are as follows:

ROUGHNES is used when measuring surface profiles for characterizing and calibrating surface roughness. As the stylus traverses the rough surface, the time varying electrical signal is digitized under program control and stored in the computer memory. The roughness average parameter R_a is calculated and printed for each profile, and the digitized profile data are stored on magnetic disks for subsequent analysis.

STEPHGHT, like ROUGHNES, controls data acquisition and storage, but this program is used for measurement and calibration of step heights. The step heights are calculated and printed during the execution of the program.

WHATSON is a utility program for examining the header information of data files stored on a disk. This program is helpful for keeping track of the various kinds of data and for determining which files may be deleted.

AVRGRA is used primarily in calibration reports. It calculates the average and the random uncertainty of a set of step height values or R_a values. The calibration uncertainty of the instrument and the resulting total uncertainty of the measurement are also calculated and all of the results are printed.

SMORGAS is a program that calculates seven parameters from stored profile data. These are R_a ; the rms roughness - R_q ; the peak-valley roughness - R_{tm} ; the average slope - S_a ; the average wavelength - D_a ; the peak count wavelength D_{pc} ; and the skewness - Q . The operator has the options of fitting the data to a least squares mean line, of choosing the sample length for R_{tm} , and of choosing the point to point spacing of the average slope calculation.

PLOTSVIL generates plots of the surface profiles stored on disk.

ADF, ACF and PSD are used to calculate and plot the amplitude density function, autocorrelation function, and power spectral density respectively for any or all the profiles in a file. These functions may also be printed as arrays of numbers if the operator chooses.

These programs were written in FORTRAN 77 for use on a Perkin Elmer-Interdata* 7/32 minicomputer with 256 Kbytes of memory storage, and magnetic disk storage consisting of 4 disks each with 5 Mbytes capacity. PSD is the largest by far of the nine programs. It requires 111 Kbytes of memory storage. The next largest module, ACF, requires 69 Kbytes.

The system also includes a system console, a printer, a Versatec printer-plotter, and an Analog-to-digital converter (ADC) housed in an interface (built at NBS) called the NBS Bus. The stylus instruments, a Talystep and a Talysurf 4, track the undulations of the surface profile and produce a time varying output voltage which can be filtered using one of several high pass electronic filters. A frequency generator interfaced to the NBS Bus controls the rate at which the output signal is digitized and hence the point-to-point spacing of the data.

The present arrangement is quite similar to one developed at NBS several years ago [5] with a much smaller minicomputer system. Whereas the previous software was written in machine code [6] because of the limited memory size, the present system size allows the use of the high level compiled language, FORTRAN 77. It should be noted that a few of the commands to be shown here are not part of the FORTRAN 77 language but come from run time libraries supplied by Perkin Elmer and Versatec.

One convenience of the present storage system is that each of the four disks serves a different function. The FORTRAN programs for surface texture measurement are stored on a removable disk called SRF, and the basic operating system programs are stored on the fixed, system disk, SYS. A second removable disk contains the surface profile data and, to date, approximately 12 disk cartridges have been filled with close to 60 Mbytes of profile information. The fourth disk is used as a scratch disk to hold the large temporary files which are needed for plotting graphs.

The software works as follows. Each of the nine commands listed above calls a file from the SYS disk. This file is a set of operating system commands that loads the appropriate program from

* Certain commercial equipment are identified in this report to specify adequately the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the equipment identified is necessarily the best available for the purpose.

SRF, assigns various logical units and executes the program. After the program is completed, the operating system shows the menu again or refers the operator to it, so that the operator can be reminded of the nine commands that are available in the set.

The remainder of this report is organized as follows. Chapter 2 discusses the three operating system files that introduce the operator to the system. Chapters 3-11 describe the structure of the programs for carrying out the nine commands. Each chapter describes one command and consists of five parts:

- 1) a short summary of the program (what it does),
- 2) a copy of the operating system file which calls the program,
- 3) a copy of the FORTRAN program,
- 4) a flowchart of the FORTRAN program, and
- 5) a sample of the hardcopy output, i.e. printouts or plotted graphs.

During this software development project I was greatly aided by F. E. Scire, who thoroughly tested many aspects of the system and made a number of valuable suggestions, and by E. C. Teague, who wrote the data gathering subroutine ADCIO. I am also grateful to D. E. Gilsinn for several enlightening discussions, to S. A. Morris for preparation of the manuscript, to L. Greenspan, M. Cadoff, and B. Rust for reviewing it, and to the GAMS Support Services Group for providing the portable FFT programs from the Center for Applied Mathematics software library. The work was supported in part by the David Taylor Naval Ship Research and Development Center, Annapolis, Maryland.

2. The Menu

When the operator types the command "FAST", the text shown in fig. 2.1 appears on the console screen. The instructions point to two other commands, FASTMENU (fig. 2.2) and FASTNITE (fig. 2.3).

FASTMENU is the heart of the system. It lists the main programs in the package along with a short description for each, which reminds the operator about its function. FASTNITE is a supplementary set of instructions which explains how several of the analysis programs may be run in a batch mode so that the operator does not have to sit at the console and choose options. To do this correctly, the operator must know the sequence of responses that are needed for each program and type these ahead of time in a disk file. The text in FASTNITE simply explains how to instruct the program to look for the disk file for input parameters rather than the console.

***** FAST*****

WELCOME TO THE WORLD OF SURFACE TEXTURE MEASUREMENT!

USE THE COMMAND "FASTMENU" TO FIND OUT ABOUT PROFILE MEASUREMENT, ROUGHNESS AND STEP HEIGHT CALIBRATIONS, AND THE ASSOCIATED ANALYSIS PROGRAMS.

USE THE COMMAND "FASTNITE" TO FIND OUT HOW TO RUN SOME OF THE ANALYSIS PROGRAMS IN A BATCH MODE, FOR EXAMPLE, OVERNIGHT.

Figure 2.1

```

~          ***** FASTMENU *****
~
~ HERE ARE THE COMMANDS FOR MEASURING AND ANALYZING SURFACE PROFILE DATA.
~
~ ROUGHNEE IS USED FOR MEASURING SURFACE ROUGHNESS, PARTICULARLY RA.
~
~ STEPHGT IS USED FOR MEASURING STEP HEIGHTS.
~
~ WHATEVER VOL: GIVES A RUNDOWN OF THE DATA FILES PRESENT ON A DISK (VOL:) -
~ BY LISTING THE FIRST TEN LINES OF EACH FILE.
~
~ AURORA CALCULATES THE AVERAGE FOR A SET OF RA OR STEP HEIGHT DATA AND
~ THE VARIOUS UNCERTAINTIES AS WELL.
~
~ SMORGAS CALCULATES THE FOLLOWING SURFACE PARAMETERS: RA, RQ, RTM, AVE.
~ SLOPE, AVE. WAVELENGTH, PEAK-COUNT WAVELENGTH, AND SKEWNESS.
~
~ ACF THESE COMMANDS YIELD PLOTS FOR THE AUTOCORRELATION FUNCTION,
~ ADF AMPLITUDE DENSITY FUNCTION, AND POWER SPECTRAL DENSITY.
~ PSD
~
~ PLOTEVIL YIELDS PLOTS OF SURFACE PROFILES.
~
~ TO FIND OUT HOW TO USE "ACF", "ADF", AND "SMORGAS" IN AN OVERNIGHT, BATCH
~ MODE, TYPE "FASTNITE".

```

Figure 2.2

* ***** FASTNITE *****
*
* THE COMMANDS "SMORGAS", "PLOTSVIL", "ACF", AND "ADF"
* ALLOW THE OPTION OF TYPING THE CONSOLE PARAMETERS INTO A
* FILE AHEAD OF TIME. THIS ENABLES YOU TO ANALYZE DATA OVER-
* NIGHT WITHOUT SITTING BY THE SYSTEM CONSOLE.
*
* IN THE CASE OF "SMORGAS", THE COMMAND WOULD BE
*
* SMORGAS (FILENAME).
*
* LOGICAL UNIT 5 IS THEN ASSIGNED TO THE FILE SPECIFIED BY
* "FILENAME" RATHER THAN TO THE CONSOLE.
*
* "PLOTSVIL", "ACF", AND "ADF" WORK THE SAME WAY.

Figure 2.3

3. ROUGHNES

3.1 Summary

The program ROUGHNES controls the data acquisition of roughness profiles and calculates the roughness average R_a for each. The file of operating system commands (sec. 3.2), loads the program, assigns the logical units, and starts the execution. The program has a main part and four subroutines - KCAL1, RA1, STPHGT, ADCIO. The main program is used to create and assign the new file that will hold the profile data. The program also calls for header information to be entered into the file and used as a label during printouts. The main program then calls two subprograms, first KCAL1, then RA1.

KCAL1 is used to calculate a calibration constant KCAL for the apparatus. To do this, it calls the subroutine ADCIO, which controls the digitization of a surface profile of a calibration step [5] as the step is being measured in the stylus instrument. KCAL1 then calls STPHGT which fits straight lines to the profile and calculates a step in terms of quantization levels. Since the true height of the calibration step is known in micrometers (um), the subroutine is then able to calculate KCAL in units of um/quantization level. This calibration procedure is performed three times and the average KCAL is calculated and stored.

The subroutine KCAL1 also allows the option to read the calibration constant from a previously created file instead of recalibrating the instrument.

Control then passes to the RA1 subroutine which measures the surface profiles and calculates R_a using both the raw data and the calibration constant. To do this, RA1 calls ADCIO which digitizes the roughness profile. Each profile consists of 4000 digitized points. It is measured three times, and the R_a value is calculated for each time. Then an average of the three values is calculated and stored, and the third profile is stored on magnetic disk. The set of three profiles is called a position. For all of the data-taking and calibration runs, the operator has the option of rejecting the data profile if it seems incorrect or of beginning an entire position over again.

3.2 Operating System Commands

```
1      ***** ROUGHNES *****
2      *
3      LO .BG, SRF:ROUGHNES.TSK
4      T .BG
5      CLOSE ALL
6      AS 4,L4:
7      AS 5,L1:
8      AS 6,C:
9      XDE SRF:SPILL.DAT
10     AL SRF:SPILL.DAT,IN,80/5/5
11     AS 8,SRF:SPILL.DAT
12     $W* YOU SHOULD NOW BE RECEIVING INFORMATION ON THE SURFACE CONSOLE.
13     ST
14     *
15     * RETURN TO "FASTMENU"
16     *
17     FASTMENU
18     $EXIT
```

The above set of commands perform the following functions in the Perkin Elmer 7/32 computer.

LO loads the machine code program, SRF:ROUGHNES.TSK, into the background partition (.BG) of the memory.

T .BG is used to address the task in .BG for further information and direction.

CLOSE ALL closes all the logical units of the task, just in case any were still open.

The next three AS statements are used to assign logical units. L4: is the printer, L1: the surface console, and C: the system console.

XDE deletes the temporary print file, SRF:SPILL.DAT, which is left over from the previous running of ROUGHNES.

AL creates SRF:SPILL.DAT again. AS assigns it as logical unit 8.
Note: The file that holds the profile data is created later on by SRF:ROUGHNES.TSK itself.

\$W prints a message to the system console.

ST begins execution of the program.

* in lines 14-16 allows uninterpreted comments to be inserted into the command stream.

After the execution of the program is finished, FASTMENU (Fig. 2.2) is called and the main programs are listed on the system console again.

3.3 ROUGHNESS FORTRAN Program

```

1      C ***** SRF:ROUGHNES.FTN CALLED BY "ROUGHNES" *****
2      C
3      C
4      C THIS PROGRAM BEGINS THE SERIES FOR SURFACE ROUGHNESS
5      C MEASUREMENTS. THE MAIN PROGRAM ALLOWS LABELING DATA CON-
6      C CERNING THE DATE, THE SPECIMEN, AND THE INSTRUMENT TO BE READ
7      C INTO A FILE WITH THE FORMAT "VOL:ADATE.DAT". THIS FILE WILL
8      C EVENTUALLY CONTAIN ALL OF THE PROFILE DATA GENERATED DURING
9      C THE RUN AND IS CREATED BY THE OPERATOR IN THIS MAIN PROGRAM.
10     C      T. VORBURGER 4/13/78 (REVISED 9/82)
11     C
12     C
13     C      INTEGER STAT1,STAT2,STAT,FDBUS(2)
14     C      INTEGER*2 DATFIL(8),BUFFER(4000)
15     C      COMMON /FUDGE/BUFFER
16     C      DIMENSION SPEC(20),RINSTR(20)
17     C      DATA DATFIL(7),DATFIL(8) /'.D','AT'/
18     C      DATA FDBUS/'BUS: ' /
19     C
20     C
21     1      FORMAT(' ***** NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY',
22     1' IS READY TO GO. *****'///
23     1' TYPE THE NAME OF THE DATA FILE FOR TODAY IN THE' /
24     2' FOLLOWING FORMAT: "VOL:ADATE", WHERE VOL IS THE' /
25     3' THREE CHARACTER VOLUME NAME AND ADATE IS THE EIGHT' /
26     4' CHARACTER FILE NAME.' )
27     2      FORMAT(1H ,20A4)
28     3      FORMAT(' NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY' )
29     4      FORMAT(' TYPE THE SPECIMEN ID INFORMATION.' )
30     5      FORMAT(20A4)
31     6      FORMAT(' I CAN" T OPEN THIS FILE. TRY AGAIN.' )
32     7      FORMAT(1H ,9A2)
33     8      FORMAT(' TYPE "TALYSTEP" OR "TALYSURF" AND THE VERTICAL' /
34     1' AND HORIZONTAL MAGNIFICATIONS.' )
35     9      FORMAT(' TYPE THE POINT-TO-POINT SPACING IN UM/PT.' /
36     1' INCLUDE THE DECIMAL POINT EXPLICITLY.' )
37     10     FORMAT(F10.4)
38     11     FORMAT(' THE FIRST FOUR RECORDS IN ' ,8A2/
39     1' ARE AS FOLLOWS:' )
40     12     FORMAT(8A2)
41     13     FORMAT(' THE POINT-TO-POINT SPACING IS ' ,F8.4,
42     1' UM.' )
43     17     FORMAT(' I CAN" T CREATE A FILE WITH THAT NAME.'/'COME ON TURKEY.',
44     1' DO IT OVER AND GET IT RIGHT THIS TIME!' )
45     19     FORMAT(' NBS BUS ASSIGN ERROR. STATUS WAS:' ,2X,I4)
46     C
47     C
48     C FIRST, THE OPERATOR IS ASKED TO TYPE THE FILENAME, THAT WILL
49     C BE CREATED TO HOLD THE DATA, AND THE SAMPLE IDENTIFICATION.
50     C
51     C
52     92     WRITE(5,1)
53     READ(5,12) (DATFIL(J),J=1,6)
54     CALL CFILW(DATFIL,2,80,5,5,0,0,STAT1)
55     IF (STAT1.LT.1) GO TO 90
56     WRITE(5,17) STAT1
57     GO TO 92
58     90     CALL OPENW(1,DATFIL,4,0,0,STAT2)

```

```

59          IF (STAT2 .LT. 1) GO TO 93
60          WRITE(5,6)
61          GO TO 92
62      C
63      C
64      C NOW, THE OPERATOR IS ASKED TO TYPE THE HEADER INFOR-
65      C MATION CONCERNING THE SPECIMEN, THE INSTRUMENT, AND
66      C THE SPACING OF THE DATA POINTS.
67      C
68      C
69      93      WRITE(5,4)
70             READ(5,5) SPEC
71             WRITE(5,8)
72             READ(5,5) RINSTR
73             WRITE(5,9)
74             READ(5,10) PTTOPT
75             WRITE(5,11) DATFIL
76             WRITE(5,7) DATFIL
77             WRITE(5,2) SPEC
78             WRITE(5,2) RINSTR
79             WRITE(5,10) PTTOPT
80             WRITE(1,12) DATFIL
81             WRITE(1,5) SPEC
82             WRITE(1,5) RINSTR
83             WRITE(1,10) PTTOPT
84             WRITE(8,3)
85             WRITE(8,7) DATFIL
86             WRITE(8,2) SPEC
87             WRITE(8,2) RINSTR
88             WRITE(8,13) PTTOPT
89      C
90      C
91      C AT THIS POINT WE OPEN THE NBS BUS AS LU-3.
92      C
93      C
94      50      CALL OPENW(3,FDBUS,4,0,0,STAT)
95             IF (STAT .EQ. 0) GO TO 105
96             WRITE (5,19) STAT
97             PAUSE 1
98             GO TO 50
99      C
100     C
101     C FINALLY, THIS PROGRAM CALLS THE KCAL1 AND RA1
102     C SUBROUTINES ONE AT A TIME.
103     C
104     C
105     105     CALL KCAL1(DATFIL)
106            CALL RA1(DATFIL)
107            CLOSE(1)
108            CLOSE(3)
109            CLOSE(8)
110            STOP
111            END
112     C
113     C
114     C
115     C
116     C

```

```

117      SUBROUTINE KCAL1(DATFIL)
118      C
119      C
120      C THIS ROUTINE CALCULATES THE CALIBRATION CONSTANT FOR THE
121      C SURFACE ROUGHNESS FACILITY. THERE ARE TWO OPTIONS.
122      C THE OPERATOR MAY USE THE INFORMATION AND CALIBRATION
123      C CONSTANTS ALREADY STORED IN A PREVIOUS "VOL:ADATE.DAT"
124      C FILE OR MAY CHOOSE TO RECALIBRATE THE INSTRUMENT.
125      C THE RESULTS ARE RECORDED IN THE CURRENT "VOL:ADATE.DAT" FILE.
126      C      T. VORBURGER 4/17/78 (REVISED 9/82)
127      C
128      C
129      INTEGER*2 DATFIL(8),Q1,Q2,OLDFIL(8)
130      DIMENSION SPEC(20),RINSTR(20),TALY(2)
131      DIMENSION NKCAL(3),IE1(3),IE2(3),H1(3),H3(3),H5(3)
132      DIMENSION H7(3),H4(3)
133      REAL KCAL(3)
134      INTEGER STAT2,FDBUS(2)
135      INTEGER*2 BUFFER(4000)
136      COMMON /FUDGE/BUFFER
137      DATA IBLANK/' '
138      DATA FDBUS/'BUS: '
139      DATA OLDFIL(7),OLDFIL(8)/'D','AT'/
140      C
141      1      FORMAT(' WE NOW BEGIN THE RECALIBRATION PROCEDURE.')
142      2      FORMAT(1H '/' ADJUST THE FREQUENCY GENERATOR TIMING FOR'/
143      1' STEP MEASUREMENT ON THE *** ',2A4,' ***.'/
144      1' AND SET THE FILTER FOR "STEPS" MODE.')
145      3      FORMAT(40A2)
146      4      FORMAT(' BAD DATA? WE"LL START THE STEP CALIBRATION OVER.')
147      5      FORMAT(20A4)
148      6      FORMAT(' WHAT IS THE HEIGHT OF THE CALIBRATING STEP IN UM?'/
149      2' INCLUDE THE DECIMAL POINT.')
150      7      FORMAT(F10.4)
151      8      FORMAT(' CALIBRATING STEP = ',F10.4,' UM'/////)
152      9      FORMAT(4X,'H1',8X,'H3',8X,'H5',8X,'H7'/4F10.4)
153      10     FORMAT(' STEP HEIGHT = ',F10.4,' UM')
154      11     FORMAT(' KCAL = ',E13.6,' UM/QUANTIZATION LEVEL')
155      12     FORMAT(' LOOK OK? TYPE "YES", "NO", OR "ST"(FOR START OVER).')
156      13     FORMAT(3(/10X,'TRACE',I3/10X,'THE EXTREMA ARE',I3,' AND ',
157      1I3,' MM.'/10X,4X,'H1',8X,'H3',8X,'H5',8X,'H7'/10X,4F10.4/10X,
158      2'STEP HEIGHT = ',F10.4,' UM'/10X,'KCAL = ',E13.6,
159      3' UM/QUANTIZATION LEVEL'))
160      14     FORMAT(E13.6)
161      15     FORMAT(8A2)
162      16     FORMAT('/' THE FIRST SIX RECORDS IN TODAY'S DATA FILE ARE:/'
163      11X,8A2/1X,20A4/1X,20A4/F10.4,' UM/PT(SPACING)'/F10.4,' UM = ',
164      2' THE HEIGHT OF THE CALIBRATING STEP.'/E13.6,
165      3' UM/QUANTIZATION LEVEL = KCAL')
166      17     FORMAT(20A4/20A4/F10.4/F10.4/E13.6)
167      18     FORMAT(' DO YOU WISH TO RECALIBRATE THE INSTRUMENT?'/
168      1' "Y" OR "N"?')
169      19     FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
170      1' ***** THEN HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
171      22     FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
172      1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
173      29     FORMAT(' WHAT IS THE NAME OF THE FILE THAT YOU',
174      1' WANT THE KCAL FROM?'/ USE THE TWELVE CHARACTER',

```



```

175         2' FORMAT "VOL:ADATE", '/' WHERE "VOL" IS THE THREE',
176         3' CHARACTER VOLUME NAME '/' AND "ADATE" IS THE',
177         4' EIGHT CHARACTER FILE NAME.')
178     30     FORMAT(' ERROR CODE', I3, ' SOMETHING'S WRONG.',
179         1' I CAN'T OPEN THIS FILE.'/' TRY AGAIN.')
180     32     FORMAT(' THE CALIBRATION CONSTANT KCAL HAS ',
181         1' BEEN OBTAINED FROM THE FILE ', 9A2)
182     C
183     C
184     C IN THE NEXT SEQUENCE, THE OPERATOR IS ASKED TO DECIDE
185     C WHETHER TO RECALIBRATE THE INSTRUMENT OR TO USE A
186     C CALIBRATION CONSTANT FROM ANOTHER FILE.
187     C
188     C
189     122     WRITE(5,18)
190             READ(5,3) Q1
191             IF(Q1.EQ.'N') GO TO 120
192             GO TO 126
193     120     WRITE(5,29)
194             READ (5,3) (OLDFIL(J),J=1,6)
195             CALL OPENW (2,OLDFIL,0,0,0,STAT2)
196             IF (STAT2 .LT. 1) GO TO 121
197             WRITE (5,30) STAT2
198             GO TO 122
199     121     READ (2,7,REC=5) CALSTP
200             READ (2,14) AVKCAL
201             WRITE (8,32) (OLDFIL(J),J=1,6)
202             WRITE(8,8) CALSTP
203             CALL CLOSE (2,STAT2)
204             WRITE (1,7,REC=5) CALSTP
205             WRITE (1,14) AVKCAL
206             GO TO 127
207     C
208     C
209     126     WRITE(5,1)
210             READ(1,5,REC=3) TALLY
211             WRITE(5,2) TALLY
212             IF(TALLY(2).EQ.'SURF') GO TO 104
213             HORSPC = 10.
214             GO TO 105
215     104     HORSPC = 6.6667
216     C
217     C
218     C KCAL ROUTINE: THE DATA ARE READ INTO THE COMPUTER VIA
219     C THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED STEP
220     C PROFILE IS FITTED BY TWO STRAIGHT LINES IN THE STEP
221     C FITTING SUBROUTINE, STEPHGT. THE RESULT IS PASSED BACK TO
222     C THE MAIN PROGRAM AS THE VARIABLE Y4, WHICH IS DIVIDED
223     C THE THE HEIGHT OF THE CALIBRATING STEP TO YIELD A VALUE
224     C FOR KCAL IN UM/QUANTIZATION LEVEL. THE AVERAGE VALUE OF KCAL
225     C IS STORED IN AVKCAL AND WRITTEN INTO THE FILE "VOL:ADATE.DAT".
226     C
227     C
228     105     WRITE(5,6)
229             READ(5,7) CALSTP
230             WRITE(1,7,REC=5) CALSTP
231     C
232     106     SUM1 = 0.

```



```

233          DO 102 K=1,3
234          NKCAL(K) = K
235      C
236      C
237      C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
238      C
239      103      WRITE (5,19)
240          READ (5,5) JUNK
241      C
242      C
243          CALL ADCIO(1000,Y'2307',4,3,80)
244      C
245      C
246      C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
247      C OF UNDERFLOW AND OVERFLOW.
248      C
249          L1 = 0
250          L2 = 0
251          DO 75 I = 1, 1000
252              IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
253              IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
254      75      CONTINUE
255          WRITE (5, 22) L1, L2
256      C
257      C
258          CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
259          KCAL(K) = CALSTP/Y4
260          H1(K) = KCAL(K)*Y1
261          H3(K) = KCAL(K)*Y3
262          H4(K) = KCAL(K)*Y4
263          H5(K) = KCAL(K)*Y5
264          H7(K) = KCAL(K)*Y7
265          WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
266          WRITE(5,10) H4(K)
267          WRITE(5,11) KCAL(K)
268          WRITE(5,12)
269          READ(5,3) Q2
270          IF( Q2.EQ.'NO') GO TO 103
271          IF(Q2.NE.'ST') GO TO 102
272          WRITE(5,4)
273          GO TO 106
274      102      SUM1 = SUM1+KCAL(K)
275          AVKCAL = SUM1/3.
276          WRITE(1,14) AVKCAL
277          WRITE(8,13) (NKCAL(K),IE1(K),IE2(K),H1(K),H3(K),
278              1H5(K),H7(K),H4(K),KCAL(K),K=1,3)
279      C
280      C
281      C THE LABELING INFORMATION CONTAINED IN THE FILE IS
282      C READ BACK AND PRINTED OUT FOR THE OPERATOR TO CHECK.
283      C THIS INCLUDES THE FILE NAME, THE SPECIMEN AND INSTRUMENT
284      C INFORMATION, THE POINT-TO-POINT SPACING, THE CALIBRATING
285      C STEP HEIGHT, AND KCAL(K).
286      C
287      C
288      127      READ(1,15,REC=1)DATFIL
289          READ(1,17)SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
290          WRITE(5,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL

```

```

291      WRITE(8,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
292      RETURN
293      END
294      C
295      C
296      C
297      C
298      SUBROUTINE STPHGT(Z1,Z3,Z4,Z5,Z7,E1,E2,HORSPC)
299      C
300      C
301      C STEP FITTING ROUTINE: THIS ROUTINE FITS A STRAIGHT
302      C LINE TO THE SAMPLE PROFILE ON EACH SIDE OF THE STEP
303      C WHICH IS RECORDED IN THE ARRAY BUFFER/STPDAT. THE
304      C OPERATOR MUST CHOOSE EXTREMA, E1 AND E2, WHICH ARE
305      C SYMMETRICAL ABOUT THE STEP LOCATION. THE RESULT IS
306      C THE STEP HEIGHT IN QUANTIZATION LEVELS WHICH IS
307      C STORED IN THE VARIABLE Z4. Z1,3,5,7 STORE THE HEIGHT
308      C DIFFERENCE BETWEEN THE LINES AT OTHER PLACES THAN AT
309      C THE STEP LOCATION. HORSPC IS A CONVERSION FACTOR WHICH
310      C RELATES POSITION ON THE CHART RECORD IN MM TO POSITION
311      C IN THE DATA ARRAY,BUFFER.
312      C
313      INTEGER*2 BUFFER(1000)
314      COMMON /FUDGE/BUFFER
315      INTEGER E1,E2,E3,E4,X1,X2
316      DIMENSION Y1(250),Y2(250)
317      1  FORMAT(' TYPE EXTREMUM FOR LEFT HAND LINE IN MM.'/
318      1' USE I3 FORMAT.')

```

```

349      V1 = 0.
350      V2 = 0.
351      DO 101 J=1,250
352      S1 = S1+(X1+J)**2
353      S2 = S2+(X2+J)**2
354      T1 = T1+Y1(J)
355      T2 = T2+Y2(J)
356      U1 = U1+(X1+J)
357      U2 = U2+(X2+J)
358      V1 = V1+(X1+J)*Y1(J)
359      V2 = V2+(X2+J)*Y2(J)
360 101   C
361      D3 = 250.*S1 - U1**2
362      D4 = 250.*S2 - U2**2
363      RM1 = (250.*V1-U1*T1)/D3
364      RM2 = (250.*V2-U2*T2)/D4
365      B1 = (S1*T1-U1*V1)/D3
366      B2 = (S2*T2-U2*V2)/D4
367      B3 = B2-B1
368      O1 = 0.125*E4+.875*E3
369      O3 = 0.375*E4+.625*E3
370      O4 = 0.5*(E4+E3)
371      O5 = 0.625*E4+0.375*E3
372      O7 = 0.875*E4+0.125*E3
373      Z1 = (RM2-RM1)*O1+B3
374      Z3 = (RM2-RM1)*O3+B3
375      Z4 = (RM2-RM1)*O4+B3
376      Z5 = (RM2-RM1)*O5+B3
377      Z7 = (RM2-RM1)*O7+B3
378      RETURN
379      END
380   C
381   C
382   C
383   C
384      SUBROUTINE RA1(DATFIL)
385   C
386   C
387   C THIS PROGRAM RECORDS THE AVERAGE ROUGHNESS RA FOR A 4000
388   C POINT SURFACE PROFILE. THE DATA ARE READ INTO THE ARRAY
389   C BUFFER VIA THE SUBROUTINE ADCIO. THE AVERAGE ROUGHNESS
390   C FOR EACH OF THREE TRACES AT EACH POSITION IS CALCULATED
391   C AND WRITTEN ONTO THE SURFACE CONSOLE AND THE PRINTER.
392   C THE AVERAGE OF RA'S FOR EACH POSITION IS WRITTEN
393   C ON THE FILE "SRF:ADATE.DAT". THE LAST PROFILE FOR EACH
394   C POSITION IS ALSO RECORDED IN "SRF:ADATE.DAT" ALONG WITH
395   C THE TOTAL NUMBER OF POSITIONS MEASURED FOR THE DAY.
396   C
397   C
398   C
399   C
400      INTEGER*2 BUFFER(4000),DATFIL(8),NTRACE(3),NPOS,QUES1,
401      1QUES2
402      INTEGER SUM1,COMENT(10)
403      DIMENSION RA(3),REGURG(20)
404      REAL MEAN,KCAL
405      COMMON /FUDGE/BUFFER
406   C

```

TVTV 4/22/78

```

407 C
408 1 FORMAT(// ' SURFACE ROUGHNESS RA MEASUREMENTS' /
409 1' ***** ADJUST THE TIMING AND SELECT THE APPROPRIATE FILTER.')
410 2 FORMAT(40A2)
411 3 FORMAT(E13.6)
412 4 FORMAT(' POSITION',I3,' TRACE',I3,' RA =',F10.4,' UM')
413 5 FORMAT(' LOOK OK? "YES","NO", OR "ST"(START POSITION OVER).')
414 6 FORMAT(16I5)
415 7 FORMAT('/ POS. ',I2,' AVERAGE RA =',F9.4,' UM; ',
416 110A4.3('/ TRACE ',I3,' RA =',F10.4,' UM' ))
417 8 FORMAT(' DO ANOTHER POSITION?' )
418 9 FORMAT(' BAD DATA. YOU CHOOSE TO START THE POSITION' ,
419 1' OVER.' )
420 10 FORMAT(I2,F10.4,10A4)
421 11 FORMAT(' RA ROUTINE COMPLETED.' )
422 12 FORMAT(20A4)
423 13 FORMAT(/// ' SURFACE ROUGHNESS RA MEASUREMENTS' )
424 14 FORMAT(1H1)
425 15 FORMAT(1H ,40A2)
426 16 FORMAT(1H ,20A4)
427 17 FORMAT(' DO YOU HAVE ANY COMMENTS FOR THIS POSITION?'/
428 1' YOU ARE ALLOWED 40 CHARACTERS.'/
429 2' IF NOT, JUST HIT RETURN.')
430 18 FORMAT(10A4)
431 19 FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
432 1' ***** THEN HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
433 22 FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
434 1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
435 C
436 C
437 C FIRST, THE PROGRAM OPENS THE DATA FILE FOR THE DAY. ITS NAME
438 C HAS THE FORMAT "VOL:ADATE.DAT", AND IT IS STORED IN THE
439 C VARIABLE DATFIL. THEN THE VALUE OF THE CONSTANT KCAL
440 C IS READ FROM THE DATA FILE. THEN THE VARIABLE NPOS,
441 C GIVING THE NUMBER OF POSITIONS MEASURED, IS INDEXED AND
442 C RECORDED IN THE DATA FILE FOR THE DAY.
443 C
444 C
445 WRITE(5,1)
446 WRITE(8,13)
447 READ(1,3,REC=6) KCAL
448 NPOS = 1
449 WRITE(1,15) NPOS
450 C
451 C NOW, WE BEGIN THE PROCESS OF READING PROFILE DATA AND
452 C CALCULATING RA. THIS IS DONE THREE TIMES FOR EACH
453 C POSITION. THE OPERATOR HAS THE OPTION OF ACCEPTING OR
454 C REJECTING EACH RA OR OF BEGINNING THE POSITION OVER.
455 C
456 105 SUM3 = 0.
457 DO 100 K=1,3
458 NTRACE(K) = K
459 C
460 C
461 C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
462 C
463 103 WRITE (5,19)
464 READ (5,12) JUNK

```



```

465 C
466 C
467 CALL ADCIO(4000,Y'2307',4,3,40)
468 C
469 C
470 C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
471 C OF UNDERFLOW AND OVERFLOW.
472 C
473 L1 = 0
474 L2 = 0
475 DO 75 I = 1, 4000
476 IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
477 IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
478 75 CONTINUE
479 WRITE (5, 22) L1, L2
480 C
481 C
482 SUM1 = 0.
483 DO 101 J=1,4000
484 101 SUM1 = SUM1 + BUFFER(J)
485 MEAN = SUM1/4000.
486 SUM2 = 0.
487 DO 102 J=1,4000
488 DIFF = BUFFER(J) - MEAN
489 102 SUM2 = SUM2 + ABS(DIFF)
490 RA(K) = KCAL*SUM2/4000.
491 WRITE(5,4) NPOS,NTRACE(K),RA(K)
492 WRITE(5,5)
493 READ(5,2) QUES1
494 IF (QUES1.EQ.'NO') GO TO 103
495 IF (QUES1.NE.'ST') GO TO 100
496 WRITE(5,9)
497 GO TO 105
498 100 SUM3 = SUM3 + RA(K)
499 C
500 C THE AVERAGE IS CALCULATED, AND THE RESULTS ARE WRITTEN
501 C ON THE SURFACE CONSOLE, THE PRINTER, AND THE DATA FILE
502 C FOR THE DAY. THE PROFILE DATA FROM TRACE 3 FOR EACH
503 C POSITION ARE ALSO WRITTEN INTO THE DATA FILE. THE OPER-
504 C ATOR HAS THE OPTION OF GOING TO A NEW POSITION OR OF
505 C ENDING THE PROGRAM.
506 C
507 AVRA = SUM3/3.
508 WRITE(5,17)
509 READ(5,18) COMENT
510 WRITE(1,10) NPOS,AVRA,COMENT
511 WRITE(1,6) BUFFER
512 IF (NPOS.EQ.6 .OR. NPOS.EQ.16 .OR. NPOS.EQ.26
513 1 .OR. NPOS.EQ.36 .OR. NPOS.EQ.46 .OR. NPOS.EQ.56
514 2 .OR. NPOS.EQ.66 .OR. NPOS.EQ.76 .OR. NPOS.EQ.86)
515 3 WRITE(8,14)
516 WRITE(8,7) NPOS,AVRA,COMENT,(NTRACE(K),RA(K),K=1,3)
517 WRITE(5,8)
518 READ(5,2) QUES2
519 IF(QUES2.EQ.'NO') GO TO 104
520 NPOS = NPOS + 1
521 GO TO 105
522 104 WRITE(1,10,REC=7) NPOS

```



```

523      REWIND 8
524      WRITE(4,14)
525      106 READ(8,12,END=108) REGURG
526          IF (REGURG(1) .NE. '1 ') GO TO 107
527          WRITE(4,14)
528          GO TO 106
529      107 WRITE(4,16) REGURG
530          GO TO 106
531      108 WRITE(5,11)
532          RETURN
533          END
534      C
535      C
536      C
537      C
538      C
539          SUBROUTINE ADCIO (N, MDEADR, ADR, LU, TOC)
540      C
541      C THIS ROUTINE SETS UP THE NBS BUS FOR ACQUIRING A RECORD OF N DATA
542      C POINTS VIA THE A TO D CONVERTER (ADC). THE SETUP OF THE ADC
543      C ASSUMES THAT IT WILL BE EXTERNALLY CLOCKED AT THE REQUIRED
544      C RATE. THE PROGRAM IS BASED ON THE BUS DRIVER WRITTEN BY
545      C CHARLES CODLING OF COMPUTER CONSULTANTS WHICH WAS IN TURN
546      C A MODIFICATION OF THE MUXBUS DRIVER WRITTEN BY RICHARD
547      C FREEMIRE OF NBS.
548      C
549      C THE PROGRAM ALERTS THE OPERATOR OF ANY PROBLEM IN OPENING
550      C THE BUS BY PROMPTS ON THE USER'S TERMINAL WHICH IS
551      C ASSUMED TO BE ASSIGNED TO LOGICAL UNIT 5.
552      C
553      C          WRITTEN BY E. CLAYTON TEAGUE 4/10/78
554      C          REVISED 1/14/83
555      C
556      C -----
557      C INPUT VARIABLE DEFINITIONS.
558      C
559      C N - NUMBER OF DATA POINTS TO BE ACQUIRED. 2N BYTES WILL
560      C BE ACQUIRED.
561      C MDEADR- HEX CODE TO INDICATE WHICH CHANNEL OF THE ADC
562      C WILL BE USED IN THE DATA ACQUISITION, WHAT TYPE
563      C OF CONVERSION TO MAKE ETC. THE USER SHOULD CONSULT
564      C A MANUAL ON THE NBS BUS TO DETERMINE WHAT THIS
565      C CODE SHOULD BE.
566      C ADR - SUBADDRESS OF THE ADC ON THE NBS BUS.
567      C LU - LOGICAL UNIT ASSIGNMENT OF THE NBS BUS.
568      C BUFFER - STARTING ADDRESS OF THE STORAGE AREA FOR
569      C ACQUIRED DATA. IT IS THE USER'S RESPONSIBILITY
570      C TO INSURE THAT 2N BYTES OF SPACE ARE ALLOTTED.
571      C TOC - TIME OUT CONSTANT TO BE USED FOR THE DATA ACQUISITION.
572      C
573      C
574      C          INTEGER DUMMY, ADR, TOC
575      C          INTEGER PBLK(8), ISTAT
576      C          INTEGER*2 BUFFER(4000)
577      C          LOGICAL FLAG
578      C          COMMON /FUDGE/BUFFER
579      C          DATA PBLK/8*0/
580      C

```

```

581 C SET UP BUS FOR I/O BY INITIALIZING BUSY BIT.
582 C
583 30 FLAG = .FALSE.
584 40 PBLK(6) = 0
585 C
586 C CLEAR BUS AND CHECK STATUS
587 C
588 CALL SYSIO (PBLK, Y'C0', LU, DUMMY, 0, Y'C0')
589 CALL SYSIO (PBLK, Y'C0', LU, DUMMY, 0, Y'B0')
590 CALL ILBYTE(ISTAT, PBLK, 2)
591 IF (ISTAT.EQ.2) GO TO 50
592 WRITE (5, 930) ISTAT
593 PAUSE 10
594 GO TO 40
595 C
596 C SELECT ADC SUBADDRESS ON BUS AND CHECK STATUS.
597 C
598 50 IADR = ADR + Y'C0'
599 CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, IADR)
600 C
601 CALL ILBYTE (ISTAT, PBLK, 2)
602 IF (ISTAT.EQ.136.OR.ISTAT.EQ.0)GO TO 60
603 WRITE (5, 940) ISTAT
604 PAUSE 11
605 GO TO 50
606 C
607 C SET UP MODE AND ADDRESS REGISTER OF ADC.
608 C
609 60 IMDE = MDEADR * Y'10000'
610 CALL SYSIO(PBLK, Y'38', LU, IMDE, 2, 0)
611 CALL ILBYTE(ISTAT,PBLK,2)
612 IF(ISTAT .EQ. 0) GO TO 68
613 * WRITE(5,950) ISTAT
614 PAUSE 12
615 GO TO 60
616 C
617 C LOAD TIME OUT CONSTANT(TOC) FOR READING. FIRST HALFWORD
618 C IS TOC, SECOND HALFWORD MUST BE NONZERO TO LOAD VALUE.
619 C DEFAULT TOC IS 10 SECONDS.
620 C
621 68 PBLK(6) = 0
622 PBLK(6) = 1 + TOC*Y'10000'
623 M = 2 * N
624 CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, IADR)
625 CALL ILBYTE (ISTAT, PBLK, 2)
626 IF(ISTAT .NE. Y'88') WRITE (5, 945) ISTAT
627 C
628 C READ 2N BYTES FROM CHANNEL SPECIFIED BY IMDEADR INTO
629 C BUFFER AND CHECK STATUS OF THE ADC CARD.
630 C
631 CALL SYSIO(PBLK, Y'58', LU, BUFFER, M, 0)
632 CALL ILBYTE(ISTAT, PBLK, 2)
633 IF (ISTAT .EQ. 0) GO TO 70
634 FLAG = .TRUE.
635 WRITE (5, 960) ISTAT
636 GO TO 40
637 C
638 930 FORMAT(' STATUS DURING CLEAR OF BUS WAS:' ,2X,Z4)

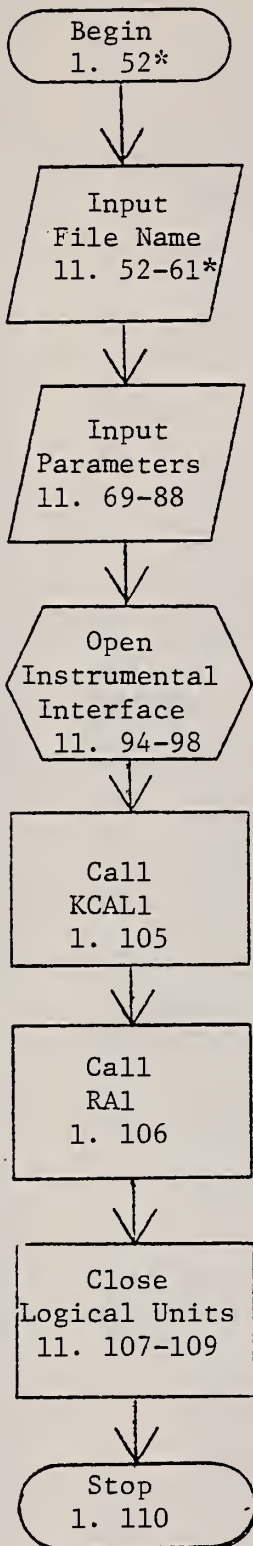
```

```

639      940  FORMAT(' IN SELECTING ADC ON NBS BUS.' /
640      1 ' STATUS WAS:' ,2X,Z4)
641      945  FORMAT(' STATUS JUST BEFORE READ OF A/D WAS:' /
642      1 ' ,2X,Z4)
643      950  FORMAT(' STATUS DURING LOADING MODE AND ADDRESS REG. WAS:' ,2X,Z4)
644      960  FORMAT(' IN READ FROM ADC. STATUS WAS:' ,2X,Z4)
645      70   CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, Y'B0')
646      IF (FLAG) GO TO 30
647      RETURN
648      END

```

3.4 Flowchart for ROUGHNES MAIN PROGRAM



The operator is prompted to type the name of the file that will be created to hold the profile data.

The operator is prompted to type the I.D., the type of stylus instrument and its magnification settings, and the horizontal point spacing of the data. These parameters are then output to the console, the data file, and a print file.

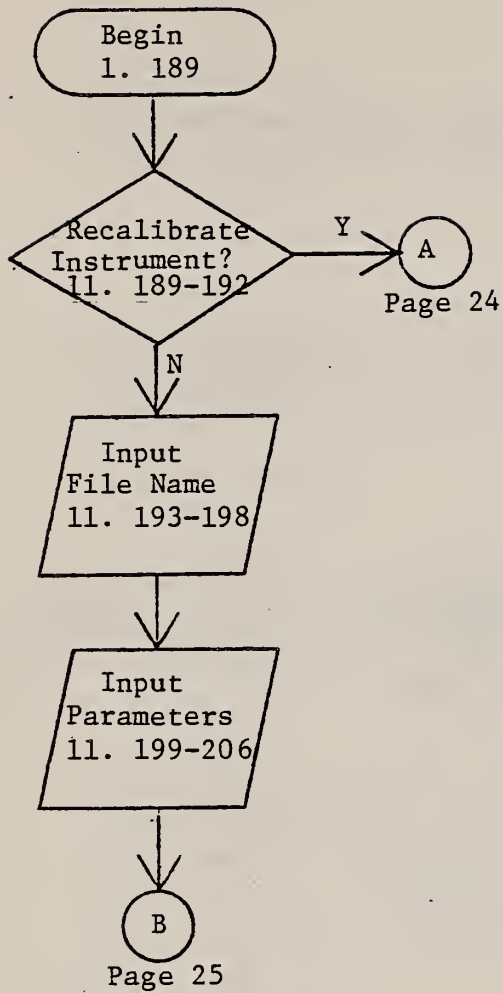
The NBS Bus is opened as a logical unit to the program. It functions as the interface between the computer and the stylus instrument.

The subroutine to measure the instrument calibration constant is called.

The subroutine to measure the roughness profiles and calculate R_a is called.

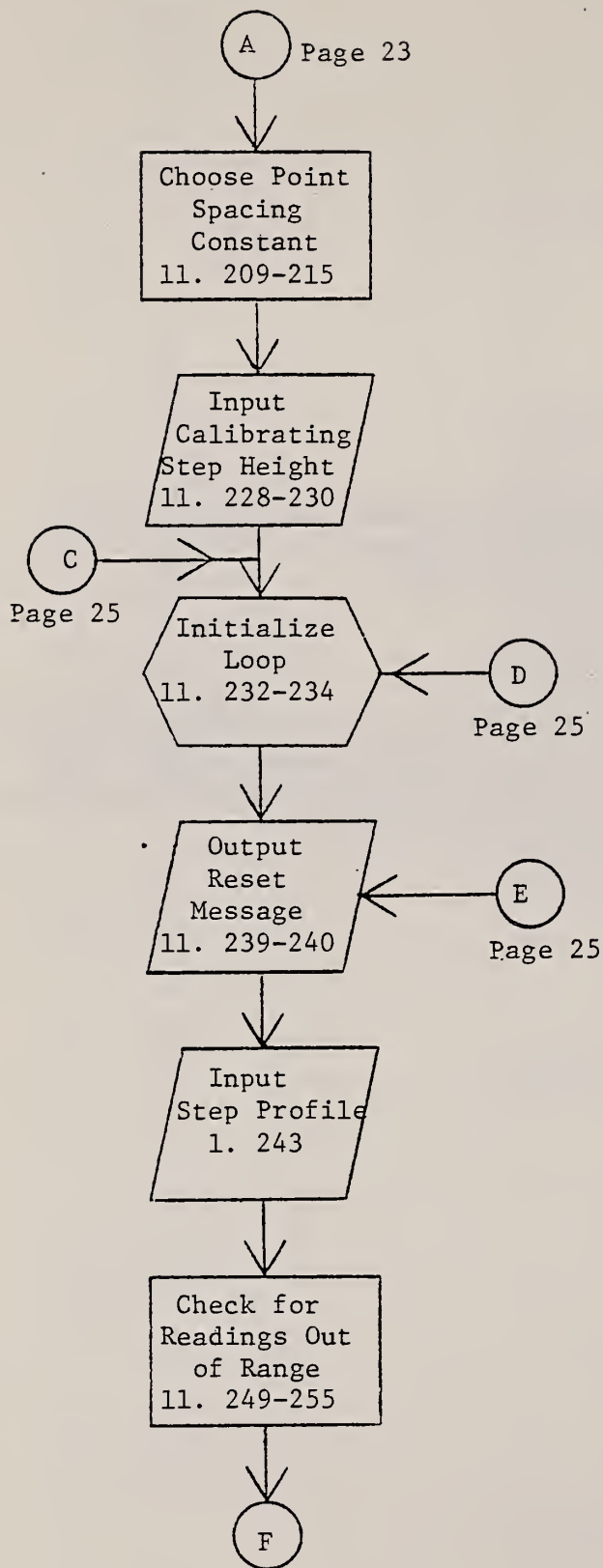
* "1.52" means "line 52"; "11. 52-61" means "lines 52-61".

KCAL1 SUBROUTINE



The operator is prompted to type the name of a previously created data file that holds the calibration data.

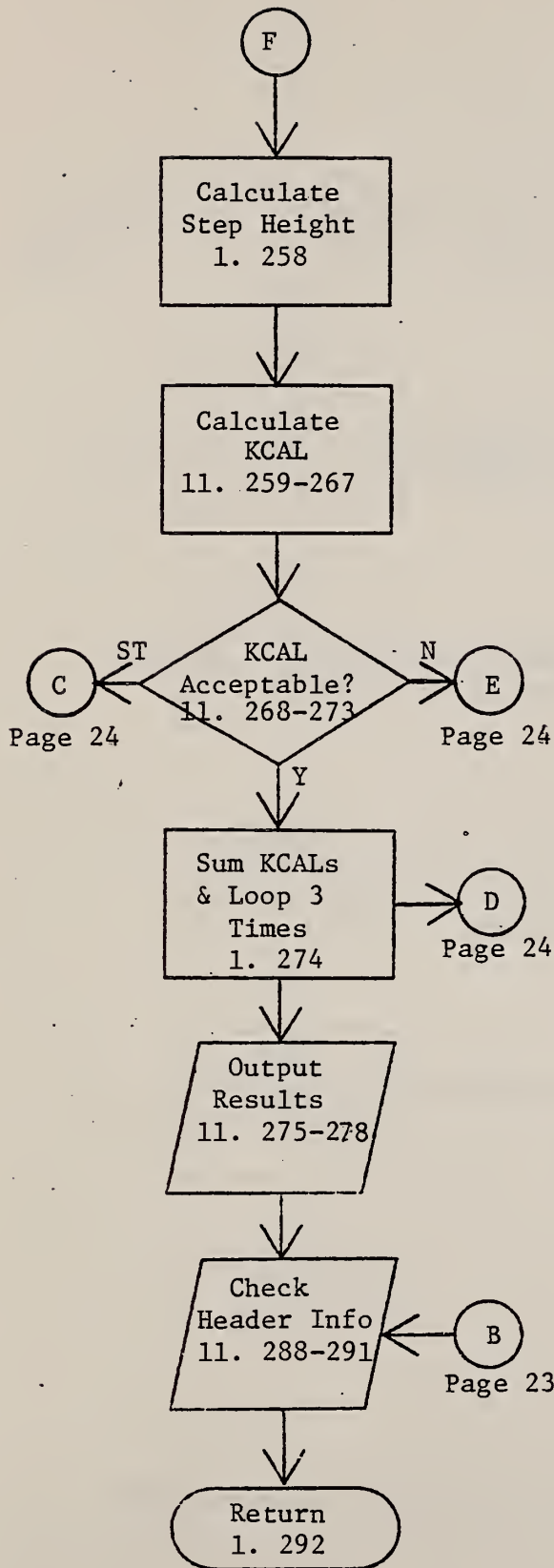
That file is read for the calibration constant KCAL and the calibrating step height. The values are then written into the current data file and the print file.



This constant depends on the chart speed and the data rate.

The operator is prompted to type the height of the calibrating step.

The calibrating step will be measured three times.



The subroutine STPHGT is called. It fits straight lines to the low and high sides of the step, extrapolates the lines to the middle of the step, and calculates the height difference in quantization levels at the middle of the step and at several other places along the profile.

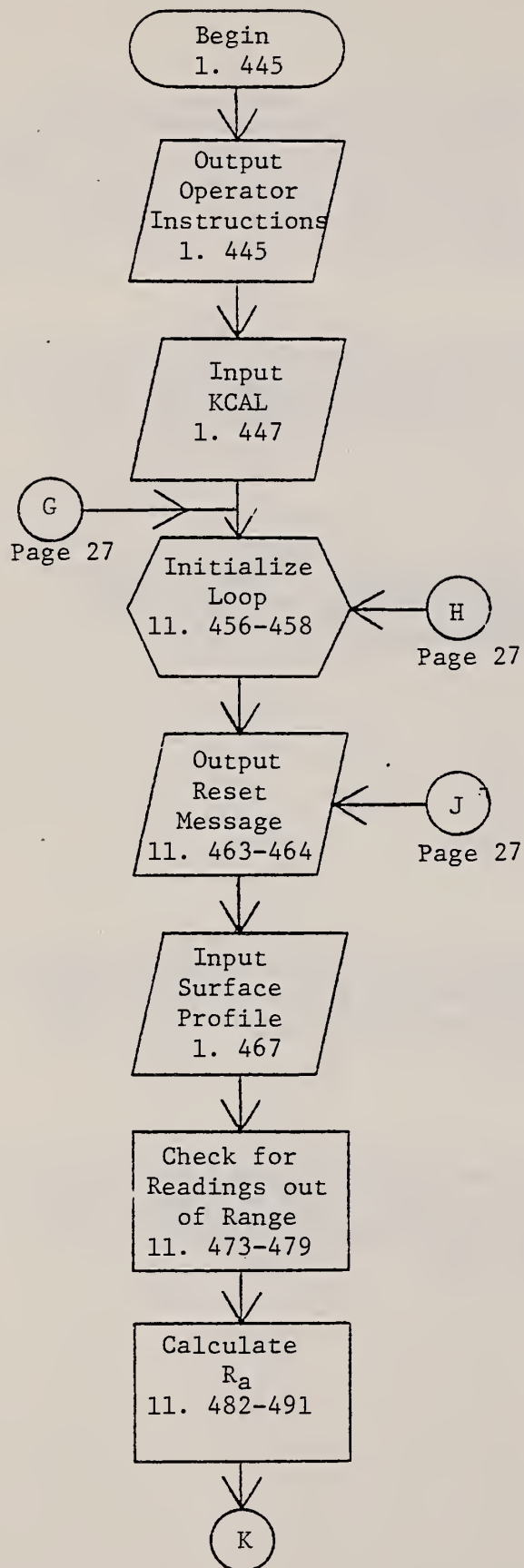
The calibration constant is calculated and displayed. The height differences discussed above are calculated and displayed in um.

If the operator types "ST", the program reinitializes the loop and begins the calibration procedure over again.

The average of three KCALs is calculated and written to the data file. KCAL and other step height parameters are written to the print file.

The first six lines of the data file are read back in and output to the console, so the operator can review the information, and to the print file.

RA1 SUBROUTINE

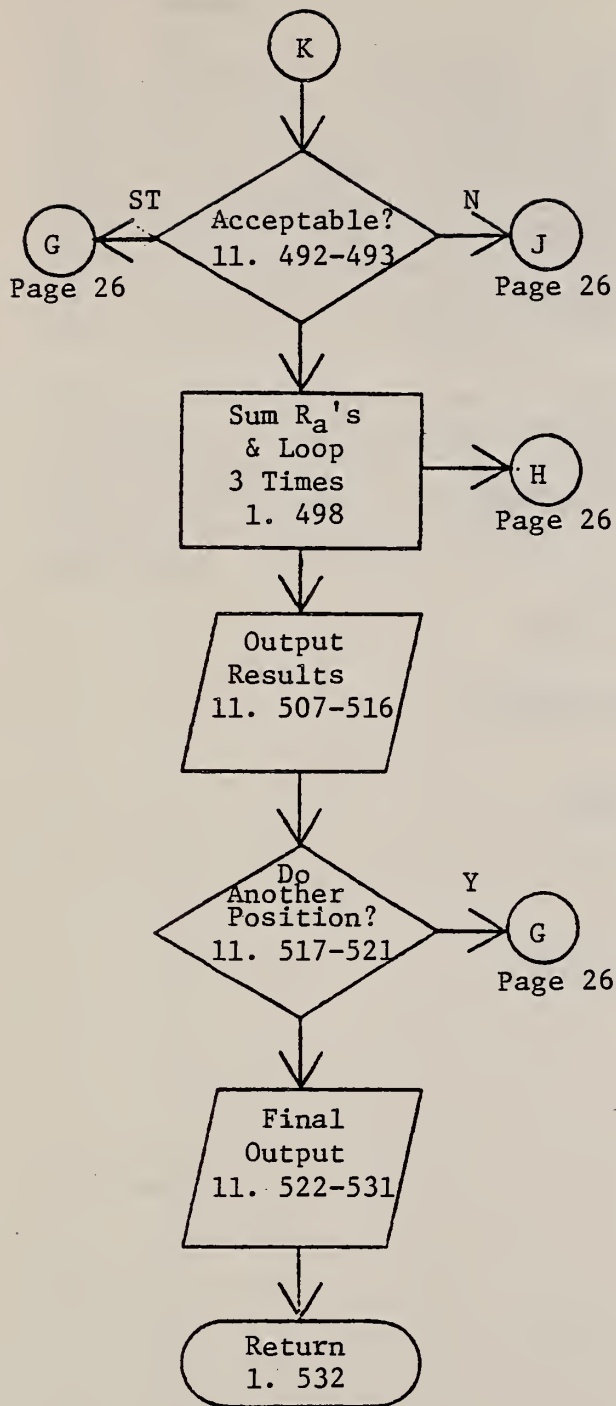


Read the calibration constant from the data file.

Each surface profile will be measured three times in this loop.

Subroutine ADCIO is called here.

The result is output to the console.



The average R_a is calculated. The operator may type in a label, which is then stored in the data file along with the third profile for each position. The measured R_a and other relevant information is output to the data file and the printer.

The final number of measurement positions is written on the data file and the print file is output to the printer.

3.5 Example of ROUGHNES Printout

NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY
SD8:B27SEP82.DAT
TEST OF TALYSURF WITH 3 UM PROTOTYPE SRM
TALYSURF 2,000X VERT. 20X HORIZ.
THE POINT-TO-POINT SPACING IS 12.7300 UM.

TRACE 1
THE EXTREMA ARE 52 AND 62 MM.
H1 H3 H5 H7
12.7358 12.7319 12.7281 12.7242
STEP HEIGHT = 12.7300 UM
KCAL = 0.664203E-02 UM/QUANTIZATION LEVEL
TRACE 2
THE EXTREMA ARE 54 AND 60 MM.
H1 H3 H5 H7
12.7321 12.7307 12.7293 12.7279
STEP HEIGHT = 12.7300 UM
KCAL = 0.660992E-02 UM/QUANTIZATION LEVEL
TRACE 3
THE EXTREMA ARE 52 AND 60 MM.
H1 H3 H5 H7
12.7326 12.7309 12.7291 12.7274
STEP HEIGHT = 12.7300 UM
KCAL = 0.660349E-02 UM/QUANTIZATION LEVEL

THE FIRST SIX RECORDS IN TODAY'S DATA FILE ARE:
SD8:B27SEP82.DAT
TEST OF TALYSURF WITH 3 UM PROTOTYPE SRM
TALYSURF 2,000X VERT. 20X HORIZ.
12.7300 UM/PT(SPACING)
12.7300 UM = THE HEIGHT OF THE CALIBRATING STEP.
0.661848E-02 UM/QUANTIZATION LEVEL = KCAL

SURFACE ROUGHNESS RA MEASUREMENTS

POS. 1 AVERAGE RA = 2.9968 UM;
TRACE 1 RA = 3.0215 UM
TRACE 2 RA = 2.9616 UM
TRACE 3 RA = 3.0074 UM

4. STEPHGHT

4.1 Summary

The program STEPHGHT serves an important function in calibrations. It controls the data acquisition of stepped profiles and calculates the step heights. The main program is almost identical to the main program in ROUGHNES. It calls the subroutine STEPCAL, the first part of which calculates a calibration constant KCAL by calling ADCIO and STPHGT. The last part of STEPCAL then measures unknown steps. Each step position is measured three times, an average is calculated, and the third step profile is stored on the disk.

Nearly all of the procedures and options in STEPHGHT are similar to those of ROUGHNES. A key difference is that the surface profile consists of 1000 rather than 4000 data points. However, the profiles are stored in a 4000 point array with 3000 trailing zeros so that both kinds of profiles can be read by the same programs.

4.2 Operating System Commands

```
1  ***** STEPHGHT *****
2
3  LD .BG, SRF:STEPHGHT.TSK
4  T .BG
5  CLOSE ALL
6  AS 4,L4:
7  AS 5,L1:
8  AS 6,C:
9  XDE SRF:SPILL.DAT
10 AL SRF:SPILL.DAT,IN,80/5/5
11 AS 8,SRF:SPILL.DAT
12 $W* YOU SHOULD NOW BE RECEIVING INFORMATION ON THE SURFACE CONSOLE.
13 ST
14 *
15 * RETURN TO "FASTMENU".
16 *
17 FASTMENU
18 $EXIT
```

4.3 STEPHGHT FORTRAN Program

```

1      C ***** SRF:STEPHGHT.FTN CALLED BY "STEPHGHT" *****
2      C
3      C THIS PROGRAM BEGINS THE SERIES FOR STEP HEIGHT
4      C MEASUREMENTS. THE MAIN PROGRAM ALLOWS LABELING DATA CON-
5      C CERNING THE DATE, THE SPECIMEN, AND THE INSTRUMENT TO BE READ
6      C INTO A FILE WITH THE FORMAT "VOL:ADATE.DAT". THIS FILE IS
7      C CREATED BY THE OPERATOR AT THE BEGINNING OF THE MAIN
8      C PROGRAM AND WILL EVENTUALLY CONTAIN ALL OF THE PROFILE
9      C DATA GENERATED DURING THE RUN.
10     C      T. VORBURGER 4/13/78 (REVISED 9/82)
11     C
12     C
13     C
14     C      INTEGER STAT1,STAT2,STAT,FDBUS(2)
15     C      INTEGER*2 DATFIL(8),Q1,BUFFER(4000)
16     C      COMMON /FUDGE/BUFFER
17     C      DIMENSION SPEC(20),RINSTR(20)
18     C      DATA DATFIL(7),DATFIL(8) /'.D','.AT'/
19     C      DATA FDBUS/'BUS: ' /
20     C
21     C
22     1      FORMAT(' ***** NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY',
23     1' IS READY TO GO. *****' ///
24     1' TYPE THE NAME OF THE DATA FILE FOR TODAY IN THE' /
25     2' FOLLOWING FORMAT: "VOL:ADATE", WHERE VOL IS THE' /
26     3' THREE CHARACTER VOLUME NAME AND ADATE IS THE EIGHT' /
27     4' CHARACTER FILE NAME.' )
28     2      FORMAT(1H ,20A4)
29     3      FORMAT(1H1/' NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY.',
30     1' STEP HEIGHT MEASUREMENTS')
31     4      FORMAT(' TYPE THE SPECIMEN ID INFORMATION.' )
32     5      FORMAT(20A4)
33     7      FORMAT(1H ,9A2)
34     8      FORMAT(' TYPE "TALYSTEP" OR "TALYSURF" AND THE VERTICAL' /
35     1' AND HORIZONTAL MAGNIFICATIONS.' )
36     9      FORMAT(' TYPE THE POINT-TO-POINT SPACING IN UM/PT.' /
37     1' INCLUDE THE DECIMAL POINT EXPLICITLY.' )
38     10     FORMAT(F10.4)
39     11     FORMAT(' THE FIRST FOUR RECORDS IN ' ,8A2/
40     1' ARE AS FOLLOWS:' )
41     12     FORMAT(8A2)
42     13     FORMAT(' THE POINT-TO-POINT SPACING IS ' ,F8.4,
43     1' UM.' )
44     17     FORMAT(' ERROR STATUS = ' ,I2/' COME ON TURKEY.',
45     1' DO IT OVER AND GET IT RIGHT THIS TIME!' )
46     19     FORMAT(' NBS BUS ASSIGN ERROR. STATUS WAS:' ,2X,I4)
47     C
48     C
49     C FIRST, THE OPERATOR IS ASKED TO TYPE THE FILENAME, THAT WILL
50     C BE CREATED TO HOLD THE DATA, AND THE SAMPLE IDENTIFICATION.
51     C
52     C
53     92     WRITE(5,1)
54     READ(5,12) (DATFIL(J),J=1,6)
55     CALL CFILW(DATFIL,2,80,5.5,0.0,STAT1)
56     IF (STAT1.LT.1) GO TO 90
57     WRITE(5,17) STAT1
58     GO TO 92

```

```

59      90      CALL OPENW(1,DATFIL,4,0,0,STAT2)
60      C
61      C
62      C NOW, THE OPERATOR IS ASKED TO TYPE THE HEADER INFOR-
63      C MATION CONCERNING THE SPECIMEN, THE INSTRUMENT, AND
64      C THE SPACING OF THE DATA POINTS.
65      C
66      C
67      WRITE(5,4)
68      READ(5,5) SPEC
69      WRITE(5,8)
70      READ(5,5) RINSTR
71      WRITE(5,9)
72      READ(5,10) PTTOPT
73      WRITE(5,11) DATFIL
74      WRITE(5,7) DATFIL
75      WRITE(5,2) SPEC
76      WRITE(5,2) RINSTR
77      WRITE(5,10) PTTOPT
78      WRITE(1,12) DATFIL
79      WRITE(1,5) SPEC
80      WRITE(1,5) RINSTR
81      WRITE(1,10) PTTOPT
82      WRITE(8,3)
83      WRITE(8,7) DATFIL
84      WRITE(8,2) SPEC
85      WRITE(8,2) RINSTR
86      WRITE(8,13) PTTOPT
87      C
88      C
89      C AT THIS POINT WE OPEN THE NBS BUS AS LU-3.
90      C
91      C
92      50      CALL OPENW(3,FDBUS,4,0,0,STAT)
93      IF (STAT .EQ. 0) GO TO 105
94      WRITE (5,19) STAT
95      PAUSE 1
96      GO TO 50
97      C
98      C
99      C FINALLY, THIS PROGRAM CALLS THE STEPCAL SUBROUTINE.
100     C
101     C
102     105     CALL STEPCAL(DATFIL)
103     CLOSE(1)
104     CLOSE(3)
105     CLOSE(8)
106     STOP
107     END
108     C
109     C
110     C
111     C
112     C
113     SUBROUTINE STEPCAL(DATFIL)
114     C THIS ROUTINE CALCULATES THE CALIBRATION CONSTANT AND STEP
115     C HEIGHTS FROM THE PROFILE DATA. ALL RESULTS
116     C ARE RECORDED IN THE FILE "VOL:ADATE.DAT".

```



```

117 C
118 C
119 C
120 INTEGER*2 DATFIL(8),OLDFIL(8),Q1,Q2,Q3,Q4,Q5
121 DIMENSION SPEC(20),RINSTR(20),TALY(2),COMENT(10)
122 DIMENSION NKCAL(3),IE1(3),IE2(3),H1(3),H3(3),H5(3)
123 DIMENSION H7(3),H4(3),XSTEP(3),REGURG(20)
124 REAL KCAL(3)
125 INTEGER*2 BUFFER(4000)
126 INTEGER NTRACE(3)
127 INTEGER STAT,STAT2,STAT4,STATUS,FDBUS(2)
128 COMMON /FUDGE/BUFFER
129 DATA IBLANK/' '/
130 DATA OLDFIL(7),OLDFIL(8)/'D','AT'/
131 DATA FDBUS/'BUS:'/'
132 C
133 1 FORMAT(' WE NOW BEGIN THE STEP MEASUREMENT OR RECAL',
134 1'IBRATION PROCEDURE.')
135 2 FORMAT(' WE WILL NOW START THE RECALIBRATION PROCEDURE.'/
136 1' WHAT IS THE HEIGHT OF THE CALIBRATING STEP IN UM?'/
137 2' INCLUDE THE DECIMAL POINT EXPLICITLY.')
138 3 FORMAT(40A2)
139 4 FORMAT(' DO YOU WISH TO RECALIBRATE THE INSTRUMENT?'/
140 1' "Y" OR "N"?')
141 5 FORMAT(20A4)
142 6 FORMAT(1H1/' ADJUST THE FREQUENCY GENERATOR TIMING '/
143 1' FOR STEP MEASUREMENT ON THE *****',2A4,' *****',/
144 2' AND SET THE FILTER FOR "STEPS" MODE.'////)
145 7 FORMAT(F10.4)
146 8 FORMAT(1H ,20A4)
147 9 FORMAT(4X,'H1',8X,'H3',8X,'H5',8X,'H7'/4F10.4)
148 10 FORMAT('/' POSITION',I3,' TRACE',I3,
149 1' STEP HEIGHT =' ,F10.4,' UM')
150 11 FORMAT(' KCAL = ',E13.6,' UM/QUANTIZATION LEVEL')
151 12 FORMAT(' LOOK OK?'/ TYPE "YES" IF IT LOOKS OK,/'
152 1' TYPE "NO" IF YOU WANT TO REDO THIS TRACE,/'
153 2' AND TYPE "ST" IF YOU WANT TO START THE KCAL SEQUENCE ',
154 3'OR THE POSITION OVER.')
155 13 FORMAT(3(/10X,'TRACE',I3/10X,'THE EXTREMA ARE',I3,' AND ',
156 1I3,' MM.'/10X,4X,'H1',8X,'H3',8X,'H5',8X,'H7'/10X,4F10.4/10X,
157 2'STEP HEIGHT = ',F10.4,' UM'/10X,'KCAL = ',E13.6,
158 3' UM/QUANTIZATION LEVEL'))
159 14 FORMAT(E13.6)
160 15 FORMAT(9A2)
161 16 FORMAT('/' THE FIRST SIX RECORDS IN TODAY'S FILE ARE:/'
162 11X,8A2/1X,20A4/1X,20A4/F10.4,' UM/PT(SPACING)'/F10.4,' UM = ',
163 2' THE HEIGHT OF THE CALIBRATING STEP.'/E13.6,
164 3' UM/QUANTIZATION LEVEL = KCAL')
165 17 FORMAT(20A4/20A4/F10.4/F10.4/E13.6)
166 18 FORMAT('/' STEP MEASUREMENT PROCEDURE:/' WE WILL MEASURE THE'
167 1' STEP THREE TIMES AT EACH POSITION AND TAKE THE AVERAGE.'/
168 2' *****ADJUST THE TIMING.')
169 19 FORMAT(' BAD DATA. YOU CHOOSE TO START THE POSITION ',
170 1' OR THE KCAL SEQUENCE OVER.')
171 21 FORMAT('/' POS.',I3,' AVE. STEP HEIGHT =' ,F8.4,
172 1' UM; ',10A4,3(/ TRACE ',I3,' STEP HEIGHT =' ,F10.4,' UM'))
173 22 FORMAT(' DO ANOTHER POSITION?')
174 23 FORMAT(I2,F10.4,10A4)

```

```

175      24      FORMAT(' END OF STEP HEIGHT ROUTINE.')
176      25      FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
177      1' ***** THEN HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
178      26      FORMAT(16I5)
179      28      FORMAT('/' KCAL ROUTINE TRACE',I3,
180      1' STEP HEIGHT = ',F10.3,' UM')
181      29      FORMAT(' WHAT IS THE NAME OF THE FILE THAT YOU ',
182      1' WANT THE KCAL FROM?'/ ' USE THE TWELVE CHARACTER',
183      2' FORMAT "VOL:ADATE",'/ ' WHERE "VOL" IS THE THREE',
184      3' CHARACTER VOLUME NAME'/ ' AND "ADATE" IS THE',
185      4' EIGHT CHARACTER FILE NAME.')
186      30      FORMAT(' ERROR CODE',I3/' SOMETHING'S WRONG.',
187      1' I CAN'T OPEN THIS FILE.'/ ' TRY AGAIN.')
188      31      FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
189      1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
190      32      FORMAT('/////' THE CALIBRATION CONSTANT KCAL HAS ',
191      1' BEEN OBTAINED FROM THE FILE ',9A2)
192      33      FORMAT('/' WE NOW START THE RECALIBRATION PROCEDURE.')
193      34      FORMAT(' CALIBRATING STEP = ',F10.4,' UM'/////)
194      35      FORMAT(' DO YOU HAVE ANY COMMENTS FOR THIS POSITION?'/
195      1' YOU ARE ALLOWED 40 CHARACTERS.'/ ' IF NOT, JUST HIT "RETURN".')
196      36      FORMAT(10A4)
197      37      FORMAT(1H1)
198      C
199      C
200      WRITE(5,1)
201      READ(1,5,REC=3)TALY
202      WRITE(5,6) TALY
203      IF(TALY(2).EQ.'SURF') GO TO 104
204      HORSPC = 10.
205      GO TO 105
206      104      HORSPC = 6.6667
207      C
208      C
209      C IN THE NEXT SEQUENCE, THE OPERATOR IS ASKED TO DECIDE
210      C WHETHER TO RECALIBRATE THE INSTRUMENT OR TO USE A
211      C CALIBRATION CONSTANT FROM ANOTHER FILE.
212      C
213      C
214      105      WRITE(5,4)
215      READ(5,3) Q1
216      IF(Q1.EQ.'N') GO TO 120
217      GO TO 126
218      120      WRITE(5,29)
219      READ (5,3) (OLDFIL(J),J=1,6)
220      CALL OPENW (2,OLDFIL,0,0,0,STAT2)
221      IF (STAT2 .LT. 1) GO TO 121
222      WRITE (5,30) STAT2
223      GO TO 120
224      121      READ (2,7,REC=5) CALSTP
225      READ (2,14) AVKCAL
226      WRITE (8,32) (OLDFIL(J),J=1,6)
227      WRITE(8,34) CALSTP
228      CALL CLOSE (2,STAT2)
229      WRITE (1,7,REC=5) CALSTP
230      WRITE (1,14) AVKCAL
231      GO TO 127
232      C

```

```

233      C
234      C KCAL ROUTINE: THE DATA ARE READ INTO THE COMPUTER VIA
235      C THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED STEP
236      C PROFILE IS FITTED BY TWO STRAIGHT LINES IN THE STEP
237      C FITTING SUBROUTINE,STPHGT. THE RESULT IS PASSED BACK TO
238      C THE MAIN PROGRAM AS THE VARIABLE Y4, WHICH IS DIVIDED
239      C BY THE HEIGHT OF THE CALIBRATING STEP TO YIELD A VALUE
240      C FOR KCAL IN UM/QUANTIZATION LEVEL.
241      C
242      C
243      126      WRITE(5,2)
244              WRITE (8,33)
245              READ(5,7) CALSTP
246              WRITE(1,7,REC=5) CALSTP
247      C
248      110      SUM1 = 0.
249              DO 102 K=1,3
250              NKCAL(K) = K
251      C
252      C
253      C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
254      C
255      103      WRITE (5,25)
256              READ (5,5) JUNK
257      C
258      C
259              CALL ADCIO(1000,Y*2307*,4,3,BUFFER,40)
260      C
261      C
262      C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
263      C OF UNDERFLOW AND OVERFLOW.
264      C
265              L1 = 0
266              L2 = 0
267              DO 75 I = 1, 1000
268              IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
269              IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
270      75      CONTINUE
271              WRITE (5, 31) L1, L2
272      C
273      C
274              CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
275              KCAL(K) = CALSTP/Y4
276              H1(K) = KCAL(K)*Y1
277              H3(K) = KCAL(K)*Y3
278              H4(K) = KCAL(K)*Y4
279              H5(K) = KCAL(K)*Y5
280              H7(K) = KCAL(K)*Y7
281              WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
282              WRITE (5,28) K,H4(K)
283              WRITE(5,11) KCAL(K)
284              WRITE(5,12)
285              READ(5,3) Q2
286              IF (Q2 .EQ. 'NO') GO TO 103
287              IF (Q2 .NE. 'ST') GO TO 102
288              WRITE (5,19)
289              GO TO 110
290              IF ( Q2.EQ.'NO') GO TO 103

```

```

291      102      SUM1 = SUM1+KCAL(K)
292      C
293      C
294      C THE AVERAGE VALUE FOR KCAL(K) IS STORED IN AVKCAL AND
295      C WRITTEN INTO THE FILE "VOL:ADATE.DAT". THEN ALL OF
296      C THE LABELING INFORMATION CONTAINED IN THE FILE IS
297      C READ BACK AND PRINTED OUT FOR THE OPERATOR TO CHECK.
298      C THIS INCLUDES THE FILE NAME, THE SPECIMEN AND INSTRUMENT
299      C INFORMATION, THE POINT-TO-POINT SPACING, THE CALIBRATING
300      C STEP HEIGHT, AND KCAL(K).
301      C
302      C
303      AVKCAL = SUM1/3.
304      WRITE(1,14) AVKCAL
305      WRITE(8,13) (NKCAL(K),IE1(K),IE2(K),H1(K),H3(K),
306      1H5(K),H7(K),H4(K),KCAL(K),K=1,3)
307      127      READ(1,15,REC=1)DATFIL
308      READ(1,17)SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
309      WRITE(5,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
310      WRITE(8,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
311      C
312      C
313      C STEP MEASUREMENT SEQUENCE: KCAL IS READ FROM THE FILE
314      C "VOL:ADATE.DAT". THEN, THE VARIABLE NPOS, WHICH GIVES
315      C THE NUMBER OF POSITIONS MEASURED IS INITIALIZED AND RECORDED
316      C IN "VOL:ADATE.DAT".
317      C
318      C
319      WRITE(5,18)
320      READ(1,14,REC=6) AVKCAL
321      NPOS = 1
322      WRITE(1,3) NPOS
323      C
324      C
325      C NOW THE STEP PROFILE DATA ARE READ INTO THE COMPUTER
326      C VIA THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED
327      C STEP PROFILE IS FITTED BY TWO STRAIGHT LINES IN THE
328      C STEP FITTING SUBROUTINE,STPHGT. THE RESULT IS PASSED
329      C BACKED TO THIS PROGRAM AS THE VARIABLE Y4, WHICH IS
330      C MULTIPLIED BY AVKCAL TO YIELD A VALUE FOR THE STEP
331      C HEIGHT, XSTEP, IN UM. THIS IS DONE THREE TIMES FOR
332      C EACH POSITION. THE OPERATOR HAS THE OPTION OF
333      C ACCEPTING OR REJECTING EACH XSTEP OR OF BEGINNING
334      C THE POSITION OVER.
335      C
336      C
337      204      SUM3 = 0.
338      DO 201 K=1,3
339      NTRACE(K) = K
340      C
341      C
342      C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
343      C
344      202      WRITE (5,25)
345      READ (5,5) JUNK
346      C
347      C
348      CALL ADCIO(1000,Y'2307',4,3,BUFFER,40)

```



```

349      C
350      C
351      C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
352      C OF UNDERFLOW AND OVERFLOW.
353      C
354          L1 = 0
355          L2 = 0
356          DO 76 I = 1, 1000
357              IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
358              IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
359      76      CONTINUE
360          WRITE (5, 31) L1, L2
361      C
362      C
363          CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
364          XSTEP(K) = AVKCAL*Y4
365          H1(K) = AVKCAL*Y1
366          H3(K) = AVKCAL*Y3
367          H5(K) = AVKCAL*Y5
368          H7(K) = AVKCAL*Y7
369          WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
370          WRITE(5,10) NPOS,NTRACE(K), XSTEP(K)
371          WRITE(5,12)
372          READ(5,3) Q3
373          IF (Q3.EQ.'NO') GO TO 202
374          IF (Q3.NE.'ST') GO TO 201
375          WRITE(5,19)
376          GO TO 204
377      201      SUM3 = SUM3+XSTEP(K)
378      C
379      C
380      C THE AVERAGE STEP HEIGHT IS NOW CALCULATED AND THE RESULTS
381      C ARE WRITTEN ON THE SURFACE CONSOLE, THE PRINTER, AND THE
382      C FILE "VOL:ADATE.DAT". THE PROFILE DATA FROM TRACE 3 FOR
383      C EACH POSITION ARE ALSO WRITTEN INTO THE DATA FILE. THE
384      C OPERATOR HAS THE OPTION OF GOING TO A NEW POSITION OR
385      C OF ENDING THE PROGRAM.
386      C
387      C
388          AVSTEP = SUM3/3.
389          WRITE(5,35)
390          READ(5,36) COMENT
391          WRITE(1,23) NPOS,AVSTEP,COMENT
392          WRITE(1,26) BUFFER
393          IF (NPOS.EQ.6 .OR. NPOS.EQ.16 .OR. NPOS.EQ.26
394              1.OR. NPOS.EQ.36 .OR. NPOS.EQ.46 .OR. NPOS.EQ.56
395              2.OR. NPOS.EQ.66 .OR. NPOS.EQ.76 .OR. NPOS.EQ.86) WRITE(8,37)
396          WRITE(8,21) NPOS,AVSTEP,COMENT,(NTRACE(K),XSTEP(K),K=1,3)
397          WRITE(5,22)
398          READ(5,3) Q4
399          IF (Q4.EQ.'NO') GO TO 206
400          NPOS = NPOS+1
401          GO TO 204
402      206      WRITE(1,23,REC=7) NPOS
403          REWIND 8
404      207      READ(8,5,END=208) REGURG
405          IF (REGURG(1).NE.'1 ') GO TO 209
406          WRITE(4,37)

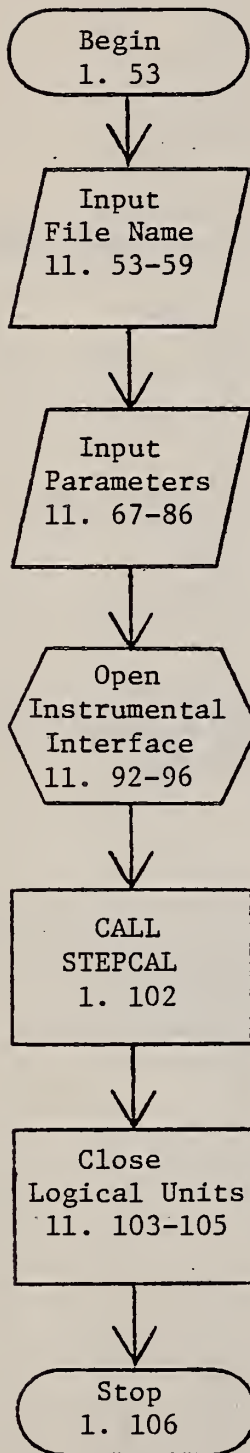
```



```
407      GO TO 207
408      209  WRITE(4,8) REGURG
409      GO TO 207
410      208  WRITE(5,24)
411      RETURN
412      END
```

The remaining subroutines, STPHGT and ADCIO, are shown in the ROUGHNES program.

4.4 Flowchart for STEPHGHT MAIN PROGRAM



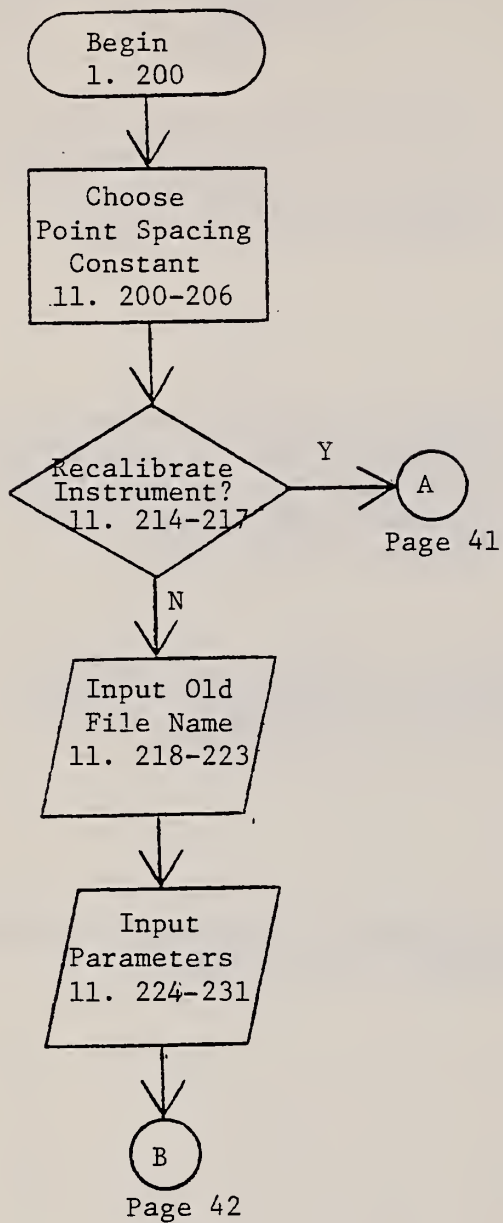
The operator is prompted to type the name of the file that will be created to hold the profile data.

The operator is prompted to type the specimen I.D., the type of stylus instrument and its magnification settings, and the horizontal point spacing of the data. These parameters are then output to the console, the data file, and a print file.

The NBS Bus is opened as a logical unit to the program. It functions as the interface between the computer and the stylus instrument.

The subroutine to determine the calibration constant and measure step heights is now called.

STEPCAL SUBROUTINE



This constant depends on the chart speed and the data input rate.

The operator is prompted to make this choice.

The operator is then asked to type the name of a previously created data file that holds the calibration data.

That file is then read for the calibration constant KCAL and calibrating step height. The values are also written into the current data file and the print file.

(A) Page 40

Input
Calibrating
Step Height
11. 243-246

The operator is prompted to type in the height of the calibrating step.

Initialize
Loop
11. 248-250

(C)
Page 42

The calibrating step will be measured three times.

Output
Reset
Message
11. 255-256

Input
Calibrating
Step Profile
1. 259

Subroutine ADCIO is called here.

Check for
Readings out
of Range
11. 265-271

Calculate
Step Height
1. 274

Calculate
KCAL.
11. 275-283

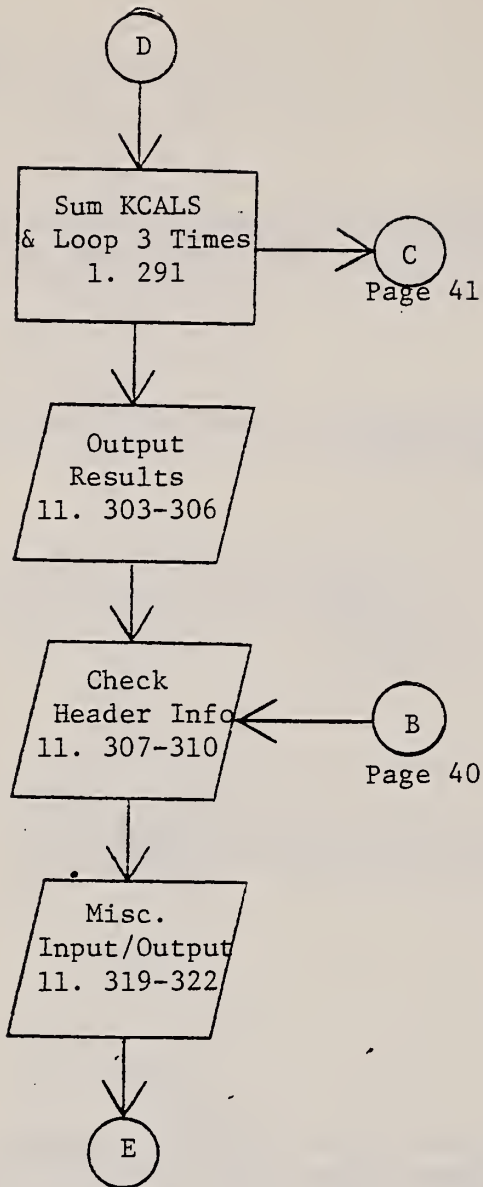
The subroutine STPHGT is called. It fits straight lines to the low and high sides of the step, extrapolates the lines to the middle of the step, and calculates the height difference in quantization levels at the middle of the step and at several other places along the profile.

The calibration constant KCAL is calculated and displayed. The height differences discussed above are calculated and displayed in um.

KCAL
Acceptable?
11. 284-287

If the operator types "ST", the program reinitializes the loop and begins the calibration procedure over again.

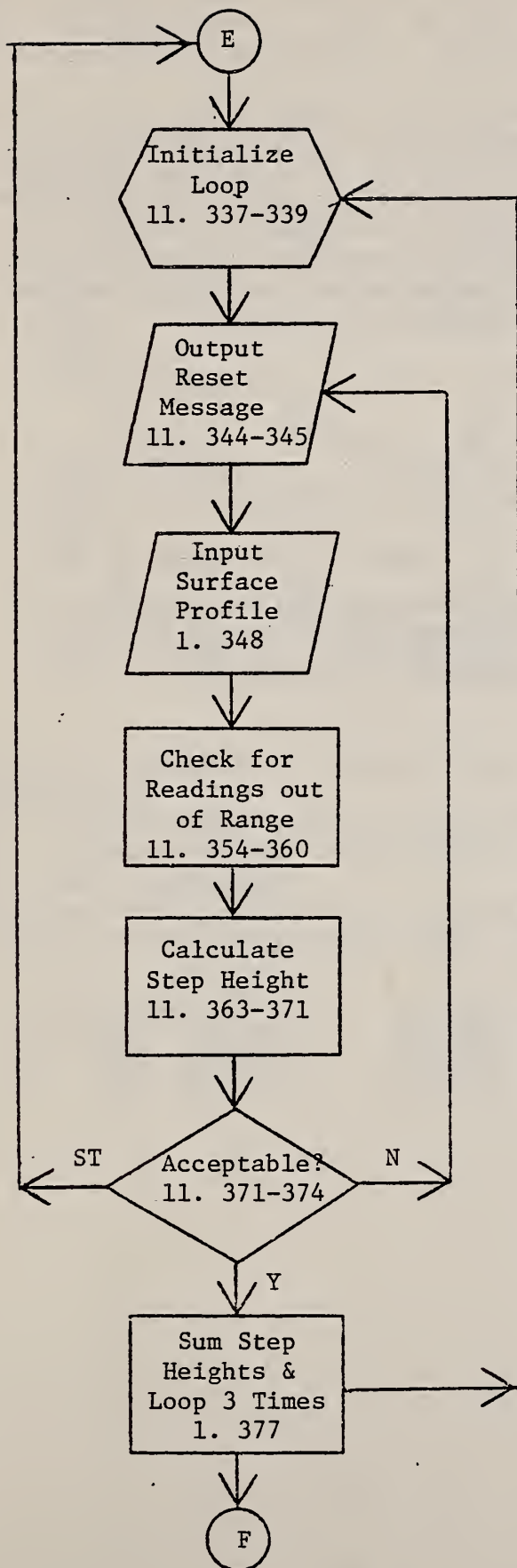
(D)



The average of 3 KCALS is calculated and written to the data file. The calibration constant and other step height parameters are written to the print file.

The first six lines of the data file are read back in and output to the console, so the operator can review the information, and to the printer.

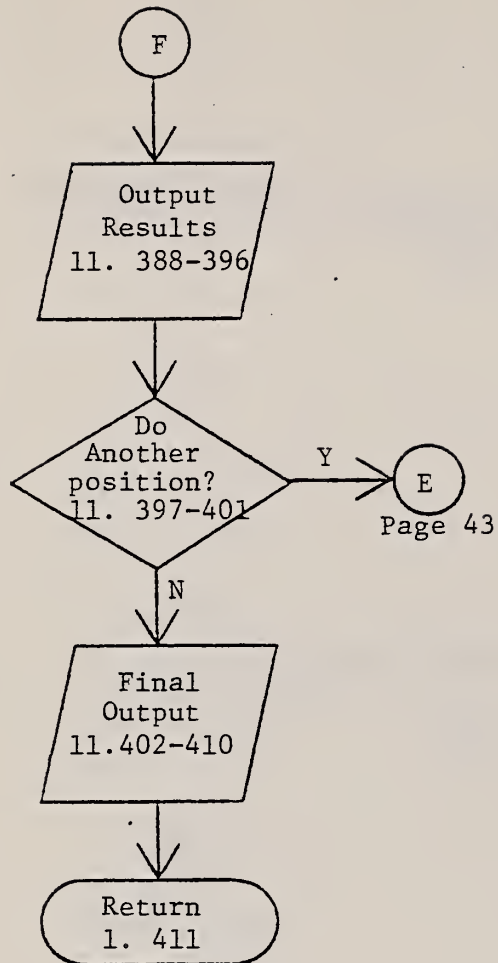
File manipulation and instructions.



Each Step profile will be measured three times.

Subroutine ADCIO is called.

The subroutine STPHGT is called again. This time, the results are used to calculate the height of the unknown step under test. The results are displayed on the console.



The average step height is calculated. At this point the operator may type in a label, which is then stored in the data file along with the third profile for each position.

- The measured step height (and other relevant information) is output to the data file and to the print file.

The final number of measured profiles is written on the data file, and the print file is output to the printer.

4.5 Example of STEPHGHT Printout

NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY, STEP HEIGHT MEASUREMENTS
SD8: C27SEP82.DAT
TALYSTEP SYSTEM CHECK USING THE 2.993UM CALIBRATION STEP
TALYSTEP 10,000X VERT. 50X HORIZ.
THE POINT-TO-POINT SPACING IS 2.0000 UM.

WE NOW START THE RECALIBRATION PROCEDURE.

TRACE 1
THE EXTREMA ARE 32 AND 54 MM.
H1 H3 H5 H7
2.9875 2.9912 2.9948 2.9985
STEP HEIGHT = 2.9930 UM
KCAL = 0.127003E-02 UM/QUANTIZATION LEVEL
TRACE 2
THE EXTREMA ARE 37 AND 60 MM.
H1 H3 H5 H7
2.9864 2.9908 2.9952 2.9996
STEP HEIGHT = 2.9930 UM
KCAL = 0.126280E-02 UM/QUANTIZATION LEVEL
TRACE 3
THE EXTREMA ARE 37 AND 61 MM.
H1 H3 H5 H7
2.9846 2.9902 2.9958 3.0014
STEP HEIGHT = 2.9930 UM
KCAL = 0.126439E-02 UM/QUANTIZATION LEVEL

THE FIRST SIX RECORDS IN TODAY'S FILE ARE:

SD8: C27SEP82.DAT
TALYSTEP SYSTEM CHECK USING THE 2.993UM CALIBRATION STEP
TALYSTEP 10,000X VERT. 50X HORIZ.
2.0000 UM/PT(SPACING)
2.9930 UM = THE HEIGHT OF THE CALIBRATING STEP.
0.126574E-02 UM/QUANTIZATION LEVEL = KCAL

POS. 1 AVE. STEP HEIGHT = 3.0002 UM; REMEASUREMENT OF 2.993 UM STEP
TRACE 1 STEP HEIGHT = 3.0000 UM
TRACE 2 STEP HEIGHT = 3.0085 UM
TRACE 3 STEP HEIGHT = 2.9922 UM

5. WHATSON

5.1 Summary

The operator can find out what kinds of data are stored on a disk by typing WHATSON followed by the name of the disk. The disk must first be in place in one of the drives and marked on by the operating system. The program reads the disk directory into a file named SRF:WHATSON.DAT and reads each line in this file for the names of the data files. Then it reads and prints the first ten logical records of each data file so that the operator can scan the information and learn the important parameters of each data file.

5.2 Operating System Commands

```
1  ***** WHATSON *****
2  *
3  XDE SRF:WHATSON.DAT
4  AL SRF:WHATSON.DAT, IN, 80
5  D T, SRF:WHATSON.DAT
6  D F, @1, SRF:WHATSON.DAT
7  LD .BG, SRF:WHATSON.TSK
8  T .BG
9  AS 1, SRF:WHATSON.DAT
10 AS 3, L4:
11 AS 6, C:
12 ST
13 * RETURN TO "FASTMENU"
14 *
15 FASTMENU
16 $EXIT
```

Notes:

"D T" writes the time and date in the temporary file SRF:WHATSON.DAT.

"D F" writes the file directory from the chosen disk in SRF:WHATSON.DAT.

5.3 WHATSON FORTRAN Program

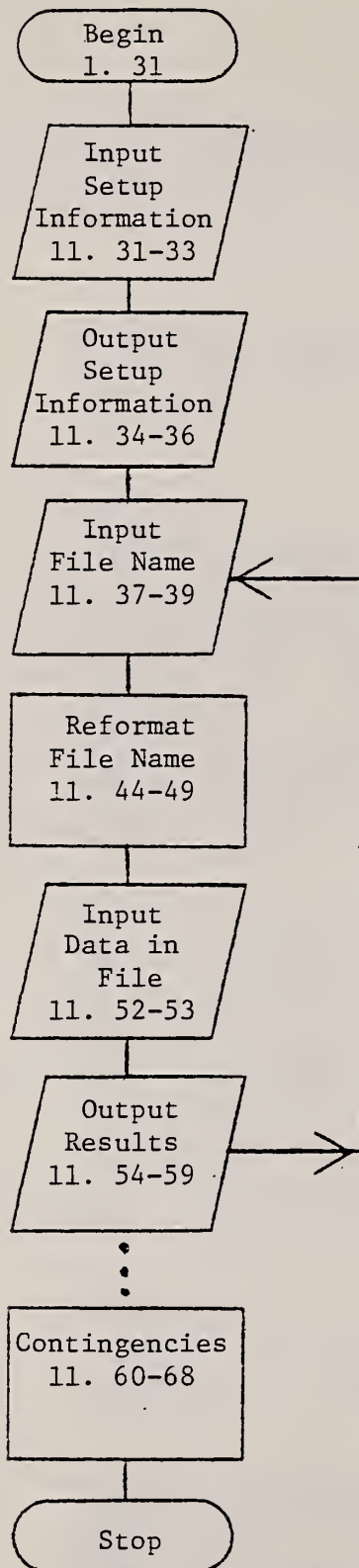
```

1      C ***** SRF:WHATSON.FTN CALLED BY "WHATSON" *****
2      C THIS PROGRAM IS USED TO FIND OUT WHAT KINDS OF
3      C DATA ARE STORED ON A DISK . THE DISK IS SPECIFIED
4      C BY THE OPERATOR IN THE "WHATSON" COMMAND THAT
5      C STARTS THIS PROGRAM.
6      C          T. VORBURGER (8/82)
7      C
8      C
9          INTEGER LABELS(20),STATUS,STAT2,WORDS(200),TIME(20)
10         CHARACTER*80 HEADER
11         CHARACTER*16 FILNAM
12         HEADER(5:5) = ':'
13     C
14     C
15     1      FORMAT(A75)
16     2      FORMAT(' ')
17     3      FORMAT(/)
18     4      FORMAT(/' AN ERROR WAS ENCOUNTERED WHILE READING'/
19     1' A LINE IN THE DIRECTORY.  I"LL TRY ANOTHER.')
20     5      FORMAT(' END OF FILE DIRECTORY'/' THAT"S ALL.')
21     6      FORMAT(8X,A4)
22     7      FORMAT(' HERE"S WHAT"S ON  DISK ',A5,' TODAY.')
23     8      FORMAT(1H1)
24     9      FORMAT(20A4)
25     10     FORMAT(/' I CAN" T OPEN THIS FILE:',A17/' I"LL TRY ANOTHER.')
26     11     FORMAT(/' I CAN" T READ THIS FILE:',A17/' I"LL TRY ANOTHER.')
27     13     FORMAT(1H ,20A4)
28     14     FORMAT(/1X,A75)
29     C
30     C
31         READ(1,9) TIME
32         READ(1,6) HEADER(1:4)
33         READ(1,9) LABELS
34         WRITE(3,8)
35         WRITE(3,9) TIME
36         WRITE(3,7) HEADER(1:5)
37     50     READ(1,1,END=102,ERR=101) HEADER(6:80)
38         IF (HEADER(15:17) .NE. 'DAT') GO TO 50
39         HEADER(14:14) = '.'
40     C
41     C THIS LOOP STRIPS OUT THE BLANKS FROM THE FILE
42     C NAME SO IT CAN BE ASSIGNED.
43     C
44         J1=0
45         DO 201 J=1,17
46             IF (HEADER(J:J) .EQ. ' ') GO TO 201
47             J1 = J1+1
48             FILNAM(J1:J1) = HEADER(J:J)
49     201     CONTINUE
50     C
51     C
52         OPEN(2,FILE=FILNAM(1:J1),STATUS='OLD',ERR=104)
53         READ(2,9,ERR=105,END=105)  WORDS
54         WRITE(3,3)
55         WRITE(3,14) HEADER(1:17)
56         WRITE(3,2)
57         WRITE(3,13) WORDS
58         CLOSE(2)

```

```
59      GO TO 50
60      101  WRITE(3,4)
61      GO TO 50
62      104  WRITE(3,10) FILNAM(1:J1)
63      CLOSE(2)
64      GO TO 50
65      105  WRITE(3,11) FILNAM(1:J1)
66      CLOSE(2)
67      GO TO 50
68      102  WRITE(6,5)
69      STOP
70      END
```

5.4 Flowchart for WHATSON



Prior to executing the FORTRAN program, the time of day and the directory of disk file names are read into a temporary file, SRF:WHATSON.DAT.

The time of day and the name of the disk are read from SRF:WHATSON.DAT

Read a file name from SRF:WHATSON.DAT and check to make sure that it is a data file.

Add punctuation and strip out the blanks so that the file name can be read correctly and opened as a logical unit.

Open the data file as a logical unit and read the first ten lines.

Print the file name and the first ten lines, then go back to line 37 to read the name of the next data file.

Statements for handling errors and the encounter with an end of file.

9/27/82 12:54:38
HERE'S WHAT'S ON DISK SDB : TODAY.

SD8: A07SEP82.DAT
MCDONNELL DOUGLAS ASTRONAUTICS CO.; S/N EI24548;
TALYSURF; 2,000X VERTICAL; 4X HORIZONTAL; 0.75 MM CUTOFF
0.9500
12.7300
0.648671E-02

SDB: A24AUG82.DAT
GAR ELECTROFORMING DIV.; SPECIMENS #8, 10, 12
TALYSURF; 10,000X VERT.; 4X HOR.; 7.5 MICRON RAD. STYLUS; 0.75 MM CUTOFF
0.9500
2.9930
0.131316E-02

```

SDB:A11MAY82.DAT
ULTRA SONIC REFERENCE BLOCKS; CS-2; URB-1
TALYSURF; 10,000X VERTICAL; 4X HORIZONTAL; 7.5 U-M RADIUS STYLUS
    0.9500
    2.9930
    0.132030E-02
14
  1    0.2679
    -82 -111 -145 -168 -169 -155 -138 -125 -115 -107 -99 -93 -93 -100 -114 -13
    -161 -198 -243 -292 -339 -395 -442 -518 -609 -706 -801 -871 -891 -823 -733 -64

```

6. AVRGRA

6.1 Summary

AVRGRA is used in calibrations to calculate the average values of the R_a and step height data gathered by ROUGHNES and STEPHGHT. The random uncertainty shown in the printout, sec. 6.5, represents three standard deviations but includes a statistical factor for finite samples. For a set of n values x_i the random uncertainty (RU) is given by

$$RU = [3/K(n)] \left(\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1} \right)^{1/2} \quad (6.1)$$

where $\bar{x}$ is the average value and $K(n)$ is the factor which depends on the number of values n . $K(n)$ has been tabulated by Natrella [7] for a range of n values.

The calibration uncertainty is calculated from one of ten formulas, the choice of which depends on 1) whether R_a or step height is being calculated and 2) the calibration step height. Each formula depends explicitly on the measured step height. Similar formulas have been given previously [8], but a few changes have been made on them to yield the present ones. These changes result from ongoing improvements and remeasurements of the system including the introduction of a new calibration step, the remeasurement of two others, remeasurement of the system noise, and further analysis of the uncertainties of roughness measurement for a highly smooth surface. The total uncertainty (sec. 6.5) is the sum of the random and calibration uncertainties.

6.2 Operating System Commands

```
1      ***** AVRGRA *****
2      *
3      LO .BG, SRF:AVRGRA.TSK
4      T .BG
5      CLOSE ALL
6      AS 3,L4:
7      AS 5,C:
8      AS 6,C:
9      ST
10     *
11     * RETURN TO "FASTMENU".
12     *
13     FASTMENU
14     $EXIT
```

6.3 AVRGRA FORTRAN Program

```

1      C ***** SRF:AVRGRA.FTN   CALLED BY 'AVRGRA' *****
2      C THIS PROGRAM DOES TWO THINGS:
3      C 1) IT CALCULATES THE AVERAGE, AND STANDARD DEVIATION
4      C FOR A SET OF M DATA POINTS, WHICH ARE ENTERED AT THE
5      C CONSOLE. THESE DATA POINTS ARE NORMALLY THE RESULTS
6      C FOR ROUGHNESS AVERAGE (RA) OR STEP HEIGHT MEASUREMENTS
7      C OBTAINED IN THE "PROFILE" PROGRAM. NOTE THAT THE FACTOR
8      C FOR SMALL STATISTICAL SAMPLES IS INCLUDED IN THE STANDARD
9      C DEVIATION (LINES 80-82).
10     C 2) THE PROGRAM ALSO CALCULATES THE CALIBRATION UNCERTAINTY
11     C AND TOTAL UNCERTAINTY FOR RA AND STEP HEIGHT MEASUREMENTS.
12     C THEREFORE, IT MAINLY IS USED AS A SUMMARY IN CALIBRATION REPORTS.
13     C T. VORBURGER, 1978 (LAST REVISION 7/82)
14     C
15     C
16     DIMENSION RA(25),STATK(25),DATE(10),SMPLID(20)
17     1   FORMAT(' SOMETHING'S WRONG. WHAT'S THE CALIBRATING STEP?')
18     11  FORMAT(10X,8(2X,F10.5))
19     12  FORMAT(F10.4)
20     13  FORMAT(20A4)
21     14  FORMAT(' NOW WE WILL CALCULATE THE AVERAGE ROUGHNESS',
22     1' OR STEP HEIGHT OF'/' THE SURFACE, THE RANDOM',
23     1' UNCERTAINTY, THE CALIBRATION'/' UNCERTAINTY, &',
24     1' THE TOTAL UNCERTAINTY.'/' WHAT IS THE DATE OF THE',
25     1' MEASUREMENTS?')
26     15  FORMAT(' STEP OR ROUGHNESS?')
27     16  FORMAT(' WHAT IS THE SAMPLE ID?')
28     17  FORMAT(' WHAT IS THE HEIGHT OF THE CALIBRATING '
29     1' STEP IN UM?')
30     18  FORMAT(' HOW MANY MEASUREMENTS? INCLUDE THE DECIMAL',
31     1' POINT EXPLICITLY.')
32     19  FORMAT(' NOW PUNCH IN THE',I3,' DATA POINTS IN',
33     1' UM.'/' ONE ON EACH LINE.')
34     20  FORMAT(10X,F7.4,' UM CALIBRATING STEP')
35     21  FORMAT(/10X,'THE MEASURED POINTS IN UM ARE:')
36     22  FORMAT(10X,5F10.4)
37     23  FORMAT(/10X,'THE FINAL RESULTS IN UM ARE:'/'
38     118X,'RA',8X,'RANDOM   CALIBRATION',4X,'TOTAL'/'
39     124X,3(' UNCERTAINTY'))
40     24  FORMAT(/10X,'THE FINAL RESULTS IN UM ARE:'/'
41     117X,'STEP',7X,'RANDOM   CALIBRATION',4X,'TOTAL'/'
42     116X,'HEIGHT',2X,,3(' UNCERTAINTY'))
43     25  FORMAT(1H1///10X,20A4)
44     26  FORMAT(10X,20A4)
45     27  FORMAT(1X,20A4)
46     28  FORMAT(///' DO ANOTHER CALCULATION?'/' Y OR N?')
47     29  FORMAT(A2)
48     30  FORMAT(///' ***** REMINDER! ***** /
49     1' THE CALIBRATION UNCERTAINTY IS GIVEN FOR THE TALYSTEP.' /
50     2' THE TALYSURF CU FOR A CALIBRATING STEP OF 0.301 UM' /
51     3' MUST BE CALCULATED BY HAND.' ///)
52     C
53     C
54     C FIRST, THE OPERATOR IS ASKED TO TYPE IN SEVERAL
55     C PARAMETERS NEEDED IN THE CALCULATIONS.
56     C
57     C
58     CALL CARCON (3,1)

```

```

59      100  WRITE(6,14)
60          READ(5,13) (DATE(I), I=1,10)
61          WRITE(6,15)
62          READ(5,13) HORR
63          WRITE(6,16)
64          READ(5,13) (SMPLID(I), I=1,20)
65          WRITE(6,17)
66          READ(5,12) CALSTP
67          WRITE(6,18)
68          READ(5,12) RM
69          M = RM
70          WRITE(6,19) M
71          READ(5,12) (RA(I), I=1,M)
72          WRITE(6,27) (DATE(I), I=1,10)
73          WRITE(6,27) (SMPLID(I), I=1,20)
74          WRITE(6,20) CALSTP
75          WRITE(6,21)
76          WRITE(6,22) (RA(I), I=1,M)
77          WRITE(3,25) (DATE(I), I=1,10)
78          WRITE(3,26) (SMPLID(I), I=1,20)
79          WRITE(3,20) CALSTP
80          WRITE(3,21)
81          WRITE(3,22) (RA(I), I=1,M)
82          DATA STATK/0...797,.886,.921,.940,.952,.959,.965,.969
83          1,.973,.975,.977,.979,.981,.982,.983,.984,.985,.986
84          2,.987,.987,.988,.988,.989,.989/
85      C
86      C
87      C THE CALCULATION OF THE AVERAGES AND STANDARD
88      C DEVIATIONS FOLLOWS NEXT.
89      C
90      C
91          SUM = 0.
92          DO 110 I=1,M
93      110  SUM = SUM+RA(I)
94          FINLRA = SUM/RM
95          SUMDEV = 0.
96          DO 112 I=1,M
97          DEVSQ = (RA(I)-FINLRA)**2
98      112  SUMDEV = SUMDEV+DEVSQ
99          VAR = SUMDEV/(RM-1.)
100         SD = SQRT(VAR)/STATK(M)
101         RU = 3.*SD
102         Q = FINLRA/CALSTP
103         IF (HORR .EQ. 'ROUG') GO TO 130
104     C
105     C
106     C THE CALIBRATION UNCERTAINTY FOR STEP HEIGHTS IS
107     C CALCULATED IN THIS SECTION.
108     C
109     C
110         WRITE (3,24)
111         WRITE (6,24)
112         IF (CALSTP .EQ. .025) GO TO 121
113         IF (CALSTP .EQ. .301) GO TO 122
114         IF (CALSTP .EQ. 2.993) GO TO 123
115         IF (CALSTP .EQ. 12.73) GO TO 123
116         IF (CALSTP .EQ. 22.9) GO TO 123

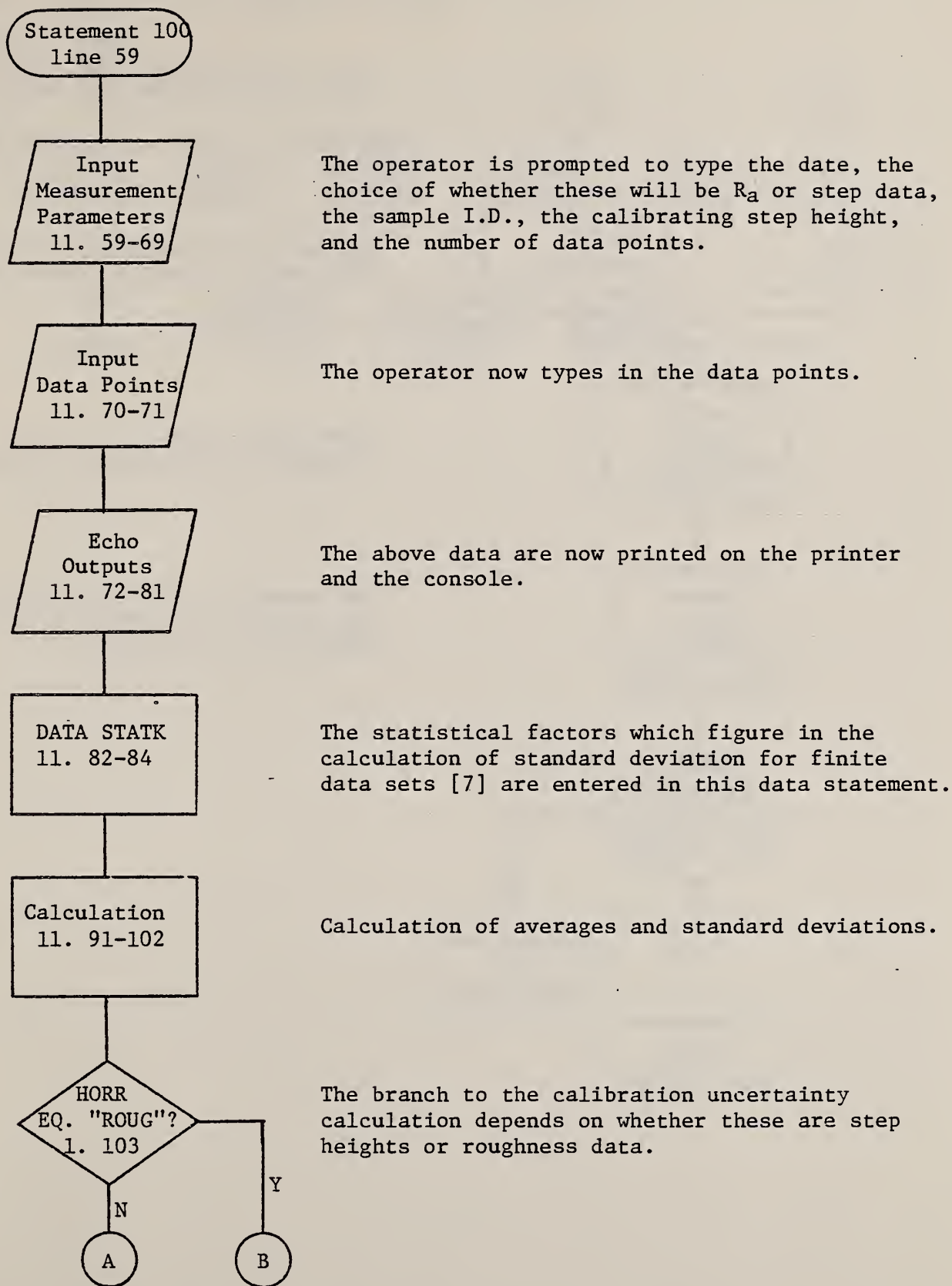
```

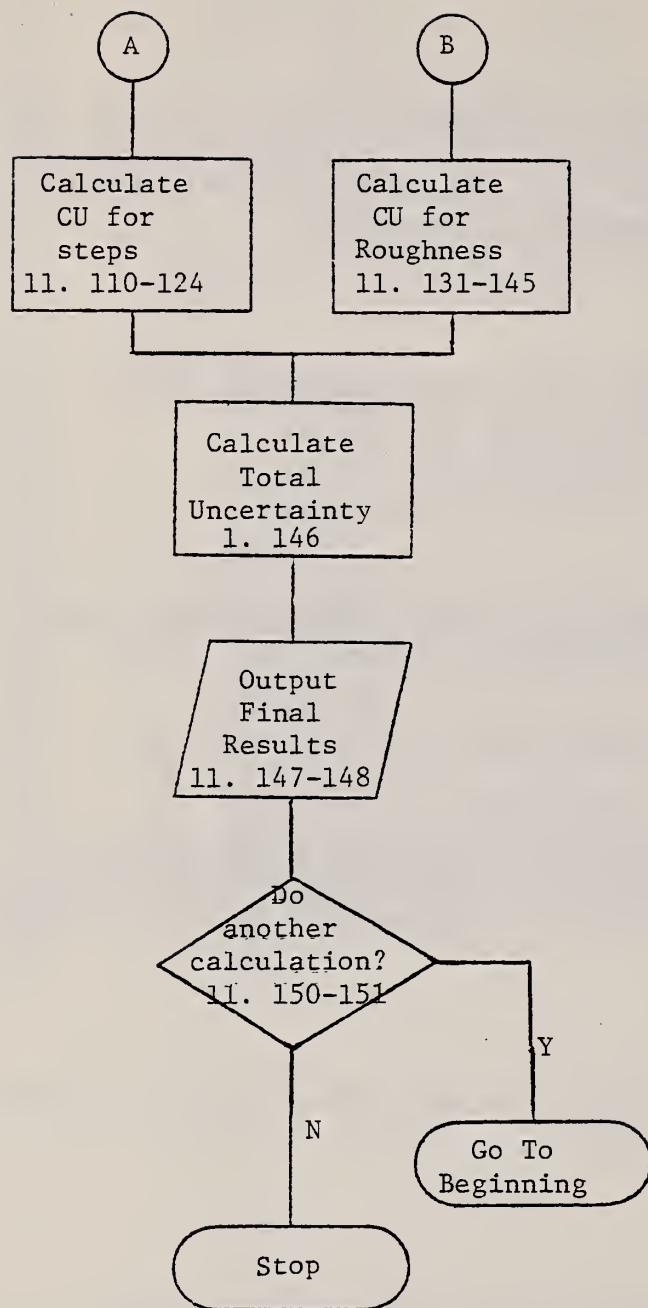
```

117         WRITE (6,1)
118         GO TO 200
119     121    CU = .001*(SQRT(6.1*Q**2 + 1.69) + 6.*Q)
120         GO TO 140
121     122    CU = .001*(SQRT(233.*Q**2 + 4.) + 12.*Q)
122         GO TO 140
123     123    CU = SQRT((.027*Q)**2 + (.01*FINLRA)**2 + (.01*CALSTP)**2) + .03*Q
124         GO TO 140
125     C
126     C
127     C THE CALIBRATION UNCERTAINTY FOR ROUGHNESS IS
128     C CALCULATED IN THIS SECTION, AND THE FINAL RESULTS ARE PRINTED.
129     C
130     C
131     130    WRITE(3,23)
132         WRITE(6,23)
133         IF (CALSTP .EQ. .025) GO TO 131
134         IF (CALSTP .EQ. .301) GO TO 132
135         IF (CALSTP .EQ. 2.993) GO TO 133
136         IF (CALSTP .EQ. 12.73) GO TO 133
137         IF (CALSTP .EQ. 22.9) GO TO 133
138         WRITE(6,1)
139         GO TO 200
140     131    CU = .001*(2.791*Q + SQRT(36.*Q*Q + .0881))
141         GO TO 140
142     132    CU=.001*(SQRT(233.*Q**2+4.)+SQRT(144.*Q**2+16.0256))
143         WRITE(6,30)
144         GO TO 140
145     133    CU=SQRT((.027*Q)**2+.000296*FINLRA**2)+SQRT((.03*Q)**2+3.6E-5)
146     140    TU = RU+CU
147         WRITE(3,11) FINLRA,RU,CU,TU
148         WRITE(6,11) FINLRA,RU,CU,TU
149     200    WRITE (6,28)
150         READ (6,29) QUERY
151         IF (QUERY .EQ. 'Y ') GO TO 100
152         STOP
153         END

```

6.4 Flowchart for AVRGRA





Calculation of Calibration Uncertainty.

Total Uncertainty
= Random + Calibration Uncertainties

The average R_a or step height and the uncertainties are printed on the console and the printer.

6.5 Example of AVRGRA Printout

JUNE 5, 1980

0.3 UM PROTOTYPE SRM

2.9930 UM CALIBRATING STEP

THE MEASURED POINTS IN UM ARE:

0.2953	0.2949	0.2949	0.2944	0.2946
0.2950	0.2942	0.2950	0.2946	0.2928

THE FINAL RESULTS IN UM ARE:

RA	RANDOM UNCERTAINTY	CALIBRATION UNCERTAINTY	TOTAL UNCERTAINTY
0.29457	0.00216	0.01241	0.01457

7. SMORGAS

7.1 Summary

This program calculates seven surface texture parameters from profile data. These are the roughness average - R_a , rms roughness - R_q , peak-to-valley roughness - R_{tm} , average slope - S_a , average wavelength - D_a , a new parameter we call the peak-count wavelength - D_{pc} , and skewness - Q . The parameters are calculated for each profile in a file as well as the averages and standard deviations of the mean for the set of profiles. The key elements in the program are the formulas used to calculate the various parameters. They are summarized as follows:

$$R_a = (1/N) \sum_{i=1}^N |y_i|, \quad (7.1)$$

where the y_i 's represent the set of digitized profile ordinates measured with respect to the mean line. The total number of points (N) in the profile is equal to 4000.

$$R_q = \left[(1/N) \sum_{i=1}^N y_i^2 \right]^{1/2}. \quad (7.2)$$

$$Q = (1/N) \sum_{i=1}^N y_i^3 / R_q^3. \quad (7.3)$$

$$R_{tm} = (1/J) \sum_{j=1}^J (y_{\max} - y_{\min})_j, \quad (7.4)$$

where $(y_{\max} - y_{\min})_j$ is the height difference between the highest peak and lowest valley in a given sampling length, a quantity chosen by the operator. The sampling length is divided into the total profile length to give the integral number of sampling lengths J .

$$S_a = (1/PkT) \sum_{j=1, 1+k, 1+2k \dots}^{1+Pk} |y_{j+k} - y_j|, \quad (7.5)$$

where T is the horizontal point spacing of the digitized profile and kT is the horizontal point spacing, chosen by the operator, that determines the resolution of the calculation. The quantity

kT is equal to an integral number of point spacings in the profile itself and P is the total number of these kT spacings in the profile.

The wavelength parameters, D_a and D_{pc} , attempt to quantify the average horizontal distance between adjacent irregularities of a surface profile. Since, for a random profile, this involves judgement as to the definition of the irregularities themselves, the parameters, D_a and D_{pc} , calculate the wavelength in two different ways.

The average wavelength, D_a , uses the ratio of the height parameter R_a to the average slope [9],

$$D_a = 2\pi R_a/S_a. \quad (7.6)$$

Alternatively, the peak-count wavelength D_{pc} relates the irregularities to a definition of the significant peaks and valleys of the profile.

$$D_{pc} = 2(N-1)T/X, \quad (7.7)$$

where $(N-1)T$ is the total profile length and X is the number of times that the profile crosses completely through a mean band with height equal to R_a , that is centered around the profile mean line [10]. If the profile were a perfect sine wave, D_{pc} and D_a would yield essentially the same value of wavelength.

There are three important options which the operator must provide SMORGAS:

1. the sampling length for the R_{tm} calculation,
2. the point to point spacing kT for the average slope calculation, and
3. the choice of calculating a mean line by taking the least squares straight line fit to the data or by taking the simple mean of the data points.

7.2 Operating System Commands

```
1      ***** SMORGAS *****
2      *
3      LD .BG, SRF:SMORGAS.TSK
4      T.BG;  CLOSE ALL
5      AS 3,L4:;  AS 6,C:
6      $IFNN @1
7      $IFNX @1
8      $W*;  $W*
9      $W* THE INPUT FILE YOU NAMED DOES NOT EXIST.
10     $W* TRY AGAIN .
11     $W*;  $W*
12     CA .BG
13     $EXIT
14     $ENDC
15     AS 5,@1
16     $ENDC
17     $IFNU @1
18     AS 5,C:
19     $ENDC
20     ST
21     *
22     * RETURN TO "FASTMENU".
23     *
24     FASTMENU
25     $EXIT
```

Notes:

The statements on lines 6-18 direct the computer to read the input parameters either from a data file or from the system console.

7.3 SMORGAS FORTRAN Program

```

1  C ***** SRF:SMORGAS.FTN CALLED BY THE COMMAND "SMORGAS" *****
2  C THIS PROGRAM CALCULATES A SMORGASBORD OF SURFACE
3  C ROUGHNESS PARAMETERS FOR SURFACE PROFILES. THESE ARE
4  C ROUGHNESS AVERAGE-RA, RMS ROUGHNESS-RQ, PEAK-VALLEY ROUGHNESS-RTM,
5  C AVERAGE SLOPE-SA, AVERAGE WAVELENGTH-DA,
6  C PEAK-COUNT WAVELENGTH-DPC, AND SKEWNESS.
7  C THE STANDARD DEVIATION CALCULATED AT THE END
8  C INCLUDES THE STATISTICAL K FACTOR FOR FINITE
9  C SAMPLES.
10 C   T.VORBURGER 3/79 (REVISED 2/26/82)
11 C
12 C
13 C   DIMENSION RA(75),RQ(75),SA(75),RTM(75),SKEW(75)
14 C   DIMENSION DPC(75),PC(75)
15 C   INTEGER SUM1,SPEC(20),NRUN(75),SUMRA,SUMSA,CUTOFF,UP2,LOW2
16 C   INTEGER*2 PROFIL(4000),DATFIL(9),A,B
17 C   REAL KCAL,NSUM,LAMBDA(75)
18 C   DATA P1,DATFIL(7),DATFIL(8),DATFIL(9)/
19 C   13.14159,2H.D,2HAT,2H /
20 C
21 C
22 1   FORMAT(F10.4)
23 2   FORMAT(40A2)
24 3   FORMAT(20A4)
25 4   FORMAT(' PRESENTING A SMORGASBORD OF SURFACE' /
26 1' PARAMETERS FOR YOUR ANALYTICAL PLEASURE!' /
27 2' WHAT IS THE FILE NAME? FORMAT SHOULD' /
28 3' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
29 5   FORMAT(I2)
30 6   FORMAT('/' POINT TO POINT SPACING =',F8.4,' UM')
31 7   FORMAT('/' KCAL =',E13.6,' UM/QUANTIZATION LEVEL')
32 8   FORMAT('/' THERE ARE ', I2,' PROFILES')
33 9   FORMAT(16I5)
34 10  FORMAT(1X,I2,3F10.4,E13.4,F9.3,F6.1,2F9.3)
35 11  FORMAT(' ERROR STATUS = ',I2,' ON OPENW ROUTINE' //
36 1' COME ON, TURKEY. DO IT OVER AND GET IT RIGHT THIS TIME!')
37 12  FORMAT(1X,I2,3F10.5,E13.4,F9.3,F6.1,2F9.3)
38 13  FORMAT(3X,3F10.5,E13.4,F9.3,6X,2F9.3)
39 14  FORMAT('/' DO YOU WISH TO CONTINUE?'/' Y OR N?')
40 15  FORMAT('/' 1X,' POS',4X,'RA',8X,'RQ',7X,'RTM',9X,'AVE',
41 17X,'AVE',6X,'PC',1X,'PEAK-COUNT',3X,'SKEW' /
42 239X,'SLOPE',2X,'WAVELENGTH',6X,'WAVELENGTH' /)
43 16  FORMAT('/10X,' AVERAGES WITH THREE STANDARD DEVIATIONS',
44 1' OF THE MEAN (UM)' /)
45 17  FORMAT(3X,3F10.4,E13.4,F9.3,6X,2F9.3)
46 18  FORMAT(E13.6)
47 19  FORMAT('/' THE DATA ARE BEING FITTED TO A LEAST SQUARES' /
48 1' STRAIGHT LINE FOR THIS CALCULATION.')
49 20  FORMAT(' WHAT IS THE POINT-TO-POINT SPACING' /
50 1' FOR THE SLOPE CALCULATION? A GOOD CHOICE' /
51 2' IS THE STYLUS RADIUS OR, IF THE STYLUS IS FLAT' /
52 3' ON TOP, THE WIDTH OF THE STYLUS TIP.')
53 21  FORMAT('/ ' EXPRESS THE VALUE IN UM & INCLUDE' /
54 5' THE DECIMAL POINT EXPLICITLY.')
55 22  FORMAT(' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE FIT' /
56 1' TO THE DATA? "Y" OR "N"?')
57 23  FORMAT(' I CAN'T READ THIS NAME. TRY AGAIN.')
58 24  FORMAT(1H ,40A2)

```

```

39 25  FORMAT(1H1)
60 26  FORMAT(' WE ROUNDED OFF THE VALUE YOU SUGGESTED' /
61      1' TO',F10.4,' SO THAT THE POINT-TO-POINT SPACING' /
62      2' FOR THE SLOPE CALCULATION IS EXACTLY ',I2 /
63      3' TIMES THE POINT-TO-POINT SPACING OF THE DATA.')
64 27  FORMAT(/F8.4,' UM = POINT-TO-POINT SPACING',
65      1' FOR THE SLOPE CALCULATION.' / ' THIS = ',I3,
66      2' TIMES THE DATA POINT SPACING.')
67 28  FORMAT(' WHAT IS THE SAMPLING LENGTH FOR THE RTM CALCULATION?' /
68      1' A GOOD CHOICE IS 800 UM.')
69 29  FORMAT(1X,20A4)
70 30  FORMAT(/F9.3,' UM = SAMPLING LENGTH FOR THE RTM CALCULATION.' /
71      1' THIS = ',I4,' TIMES THE DATA POINT SPACING.')
72 31  FORMAT(/ ' A MEAN VALUE IS BEING SUBTRACTED FROM THE DATA.')
73  C
74  C FIRST, THE OPERATOR IS ASKED TO TYPE THE NAME OF THE DATA
75  C FILE. THIS FILE IS THEN ASSIGNED TO LU 10 AND SUBSEQUENT
76  C DATA ARE READ FROM IT. THE OPERATOR IS ALSO PROMPTED TO
77  C TYPE THE INTERVAL SPACING FOR THE SLOPE CALCULATION
78  C AND THE SAMPLE INTERVAL FOR RTM.
79  C
80 95  ITRY = 0
81 92  WRITE(6,4)
82      READ(5,2,ERR=93,END=703) (DATFIL(J),J=1,6)
83      WRITE(6,24) DATFIL
84      GO TO 94
85 93  ITRY = ITRY + 1
86      IF (ITRY .GT. 2) GO TO 703
87      WRITE(6,23)
88      GO TO 92
89 94  CALL OPENW(10,DATFIL,0,0,0,ISTAT)
90      IF(ISTAT.LT.1) GO TO 90
91      WRITE(6,11) ISTAT
92      ITRY = ITRY + 1
93      IF (ITRY .GT. 2) GO TO 703
94      GO TO 92
95 90  READ(10,3,REC=2) SPEC
96      READ(10,1,REC=4) PTTOPT
97      READ(10,18,REC=6) KCAL
98      READ(10,5) NTOT
99      WRITE (3,25)
100     WRITE(3,24) DATFIL
101     WRITE(3,29) SPEC
102     WRITE(3,6) PTTOPT
103     WRITE(3,7) KCAL
104     WRITE(3,8) NTOT
105     WRITE (6,28)
106     WRITE (6,21)
107     READ (5,1) SAMLGT
108     RCUT = SAMLGT/PTTOPT
109     CUTOFF = NINT(RCUT)
110     IF (CUTOFF .GT. 4000) CUTOFF = 4000
111     SAMLGT = CUTOFF*PTTOPT
112     WRITE (6,30) SAMLGT,CUTOFF
113     WRITE (3,30) SAMLGT,CUTOFF
114     WRITE(6,20)
115     WRITE (6,21)
116     READ(5,1) RPT

```

```

117      RSPACE = RPT/PTTOPT
118      NSPACE = NINT(RSPACE)
119      RPT = NSPACE*PTTOPT
120      WRITE(6,26) RPT,NSPACE
121      WRITE(3,27) RPT,NSPACE
122      WRITE (6,22)
123      READ (5,2) QUERY1
124      IF (QUERY1 .EQ. 'Y ') WRITE(3,19)
125      IF (QUERY1 .EQ. 'N ') WRITE(3,31)
126      C
127      C NOW, WE START THE MAIN LOOP IN WHICH EACH SET OF
128      C PROFILE DATA IS READ AND ANALYZED TO FIND THE ABOVE QUANTITIES
129      C
130      DO 900 IRUN=1,NTOT
131          READ(10,5)NRUN(IRUN)
132          READ(10,9)PROFIL
133      C
134      C AT THIS POINT, WE SUBTRACT A MEAN VALUE
135      C FROM THE PROFILE DATA OR DO A LEAST SQUARES
136      C STRAIGHT LINE FIT.
137      C
138          IF (QUERY1 .EQ. 'Y') GO TO 210
139          CALL MEAN(PROFIL,4000)
140          GO TO 220
141      210      CALL LEASQ(PROFIL,4000)
142      C
143      C NOW, WE CALCULATE RA, RQ, AND SKEWNESS.
144      C
145      220      SUMRA = 0
146          SUMRQ = 0.
147          SUMSKW = 0.
148          DO 200 I = 1,4000
149              SUMRA = SUMRA + ABS(PROFIL(I))
150              SUMRQ = SUMRQ + PROFIL(I)*PROFIL(I)
151              RPRO = PROFIL(I)
152              SUMSKW = SUMSKW + RPRO*RPRO*RPRO
153      200      CONTINUE
154          RA(IRUN) = SUMRA*KCAL/4000.
155          RQ(IRUN) = SQRT(SUMRQ/4000.)*KCAL
156          SKEW(IRUN) = SUMSKW/SQRT(4000.)/(SUMRQ**1.5)
157      C
158      C AT THIS POINT, WE CALCULATE RTM.
159      C
160          PVSUM = 0.
161          NUMCUT = 4000/CUTOFF
162          DO 800 I = 1,NUMCUT
163              UP2 = -2048
164              LOW2 = 2048
165              DO 801 J=1,CUTOFF
166                  K = (I-1)*CUTOFF + J
167                  IF (PROFIL(K) .GT. UP2) UP2 = PROFIL(K)
168                  IF (PROFIL(K) .LT. LOW2) LOW2 = PROFIL(K)
169      801      CONTINUE
170          IWRITE = UP2-LOW2
171      800      PVSUM = UP2 - LOW2 + PVSUM
172          RCUT = NUMCUT
173          RTM(IRUN) = KCAL*PVSUM/RCUT
174      C

```

```

175 C NOW, WE CALCULATE THE AVERAGE SLOPE,
176 C AND THE AVERAGE WAVELENGTH.
177 C
178 SUMSA = 0
179 NSUM = 0.
180 DO 300 I=1,3999,NSPACE
181 IUP = I + NSPACE
182 IF (IUP .GT. 4000) GO TO 300
183 DIFF=PROFIL(IUP)-PROFIL(I)
184 SUMSA = SUMSA + ABS(DIFF)
185 NSUM = NSUM + 1.
186 300 CONTINUE
187 C
188 SA(IRUN) = SUMSA*KCAL/(NSUM*RPT)
189 LAMBDA(IRUN) = 2.*PI*RA(IRUN)/SA(IRUN)
190 C
191 C NOW WE COMPUTE THE PEAK-COUNT WAVELENGTH. IT IS DEFINED
192 C AS THE PROFILE LENGTH DIVIDED BY HALF THE NUMBER OF TIMES
193 C THAT THE PROFILE CROSSES A MEAN BAND CENTERED ABOUT THE
194 C MEAN LINE. THE WIDTH OF THE MEAN BAND IS EQUAL TO RA.
195 C
196 UP = SUMRA/8000.
197 RLOW = -SUMRA/8000.
198 C
199 C NOW, WE DETECT AND COUNT THE BANDWIDTH CROSSINGS.
200 C
201 A=0
202 B=0
203 E=0.0
204 DO 500 I=1,4000
205 IF (PROFIL(I).EQ.UP.OR.PROFIL(I).EQ.RLOW)GO TO 500
206 410 IF (PROFIL(I).LT.UP)GO TO 420
207 IF (A.EQ.1)GO TO 500
208 A=1
209 GO TO 450
210 420 IF (PROFIL(I).LT.RLOW)GO TO 430
211 GO TO 450
212 430 IF (B.EQ.1)GO TO 500
213 B=1
214 450 IF ((A+B).NE.2)GO TO 500
215 E=E+1.0
216 A=0
217 B=0
218 GO TO 410
219 500 CONTINUE
220 PC(IRUN) = E/2.
221 DPC(IRUN) = 7998.*PTTOPT/E
222 900 CONTINUE
223 C
224 C THE RESULTS FOR EACH PROFILE ARE NOW PRINTED
225 C
226 WRITE(3,15)
227 DO 600 K=1,NTOT
228 IF (RA(K).LT.0.001) GO TO 601
229 600 CONTINUE
230 GO TO 602
231 601 WRITE(3,12) (NRUN(K),RA(K),RQ(K),RTM(K),SA(K),
232 1LAMBDA(K),PC(K),DPC(K),SKEW(K),K=1,NTOT)

```



```

233      GO TO 603
234 602  WRITE(3,10) (NRUN(K),RA(K),RQ(K),RTM(K),SA(K),
235      1LAMBDA(K),PC(K),DPC(K),SKEW(K),K=1,NTOT)
236  C
237  C THE MAIN LOOP IS COMPLETE.
238  C FINALLY WE CALCULATE THE AVERAGES AND STANDARD
239  C DEVIATIONS OF ALL THE ABOVE QUANTITIES AND PRINT THEM.
240  C
241 603  CALL AVSD(NTOT,RA,FNLRA,SDRA)
242      CALL AVSD(NTOT,RQ,FNLRQ,SDRQ)
243      CALL AVSD(NTOT,RTM,FNLRTM,SDRTM)
244      CALL AVSD(NTOT,SA,FNLSA,SDSA)
245      CALL AVSD(NTOT,LAMBDA,FNLLAM,SDLAM)
246      CALL AVSD(NTOT,DPC,FNLDPC,SDDPC)
247      CALL AVSD(NTOT,SKEW,FNLSKW,SDSKW)
248      WRITE (3,16)
249      CALL CLOSE(10,ISTATE)
250      IF (FNLRA .LT. 0.001) GO TO 701
251      GO TO 702
252 701  WRITE(3,13) FNLRA,FNLRQ,FNLRTM,FNLSA,FNLLAM,
253      1FNLDPC,FNLSKW,SDRA,SDRQ,SDRTM,SDSA,
254      2SDLAM,SDDPC,SDSKW
255      GO TO 92
256 702  WRITE (3,17) FNLRA,FNLRQ,FNLRTM,FNLSA,FNLLAM,
257      1FNLDPC,FNLSKW,SDRA,SDRQ,SDRTM,SDSA,
258      2SDLAM,SDDPC,SDSKW
259      WRITE (6,14)
260      READ (5,2) QUERY2
261      IF (QUERY2 .EQ. 'Y ') GO TO 95
262 703  STOP
263      END
264  C
265  C
266  C THIS SUBROUTINE CALCULATES THE MEAN AND STAN-
267  C DARD DEVIATION OF THE MEAN OF A SET OF DATA VALUES.
268  C
269  C
270      SUBROUTINE AVSD(N,VALUE,FNL,SDM)
271      DIMENSION VALUE(75),STATK(25)
272      DATA STATK/0.,.797,.886,.921,.940,.952,.959,.965,.969
273      1,.973,.975,.977,.979,.981,.982,.983,.984,.985,.986
274      2,.987,.987,.988,.988,.989,.989/
275      SUM1 = 0.
276      RN = N
277      SUM2 = 0.
278      DO 100 I = 1,N
279          SUM1 = SUM1 + VALUE(I)
280 100  CONTINUE
281      FNL = SUM1/RN
282      DO 200 I = 1,N
283          DEV = VALUE(I) - FNL
284          SUM2 = SUM2 + DEV**2
285 200  CONTINUE
286      SDM = 3.*SQRT(SUM2/(RN*(RN-1.)))/STATK(N)
287      RETURN
288      END
289  C
290  C NEXT, WE CALCULATE THE MEAN OF THE DATA AND SUBTRACT

```

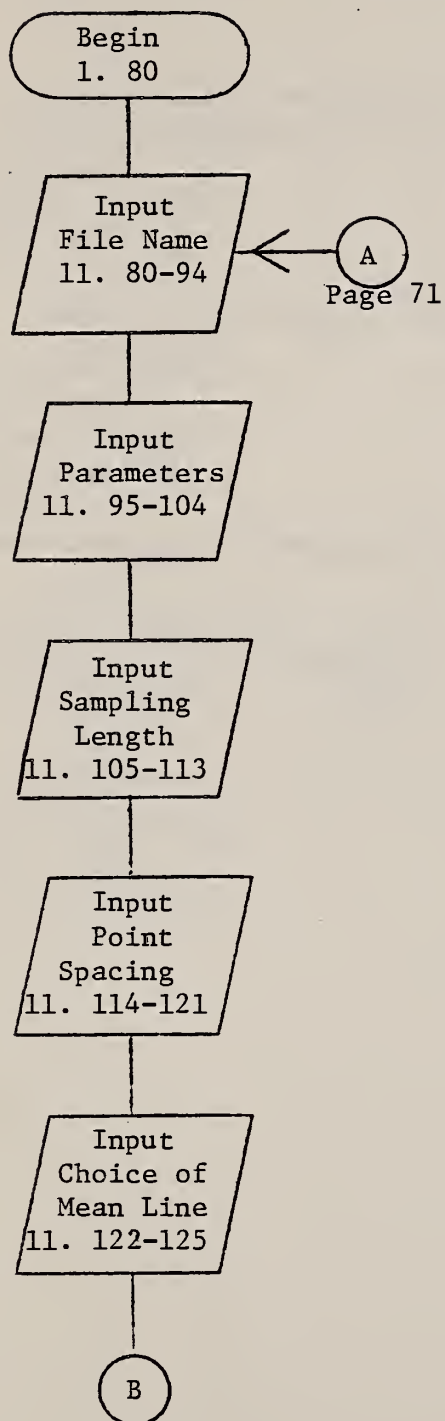


```

291 C IT FROM EACH POINT OF THE SURFACE PROFILE.
292 C
293 SUBROUTINE MEAN(PROFIL,N)
294 INTEGER*2 PROFIL(N)
295 INTEGER SUM1
296 SUM1=0
297 DO 50 I=1,4000
298 SUM1=SUM1+PROFIL(I)
299 50 CONTINUE
300 RMEAN = SUM1/4000.
301 MEEN = NINT(RMEAN)
302 DO 100 I=1,4000
303 PROFIL(I) = PROFIL(I) - MEEN
304 100 CONTINUE
305 RETURN
306 END
307 C
308 C
309 C THIS ROUTINE CALCULATES THE LEAST SQUARES STRAIGHT LINE FOR
310 C A SET OF 4000 EQUALLY SPACED DATA POINTS. IT CAN BE USED TO
311 C FILTER OUT ANY SLOPE IN A SET OF PROFILE DATA.
312 C
313 SUBROUTINE LEASQ(PROFIL,N)
314 INTEGER*2 PROFIL(N)
315 DOUBLE PRECISION SUMX2,SUMXY,DELTA,AZERO,AONE
316 RN = N
317 C
318 C IN THE LEAST SQUARES FIT, WE HAVE ALREADY
319 C CALCULATED THE SUM OF THE X'S AND X SQUARES,
320 C WHOSE VALUES NEVER CHANGE. THIS AVOIDS HAVING
321 C TO CALCULATE THEIR VALUES IN THE LOOP.
322 C
323 SUMX1 = 8002000.
324 SUMX2 = 2.1341334D10
325 SUMY1 = 0.
326 SUMXY = 0.
327 DO 101 I=1,N
328 SUMY1 = SUMY1 + PROFIL(I)
329 SUMXY = SUMXY + I*PROFIL(I)
330 101 CONTINUE
331 C
332 DELTA = RN*SUMX2 - SUMX1*SUMX1
333 AZERO = (SUMX2*SUMY1 - SUMX1*SUMXY)/DELTA
334 AONE = (RN*SUMXY - SUMX1*SUMY1)/DELTA
335 C
336 DO 102 I=1,N
337 RI = I
338 SUB = AZERO + AONE*RI
339 X = 0.5
340 IF (SUB .LT. 0.) X = -X
341 NSUB = SUB + X
342 PROFIL(I) = PROFIL(I) - NSUB
343 102 CONTINUE
344 RETURN
345 END

```

7.4 Flowchart for SMORGAS



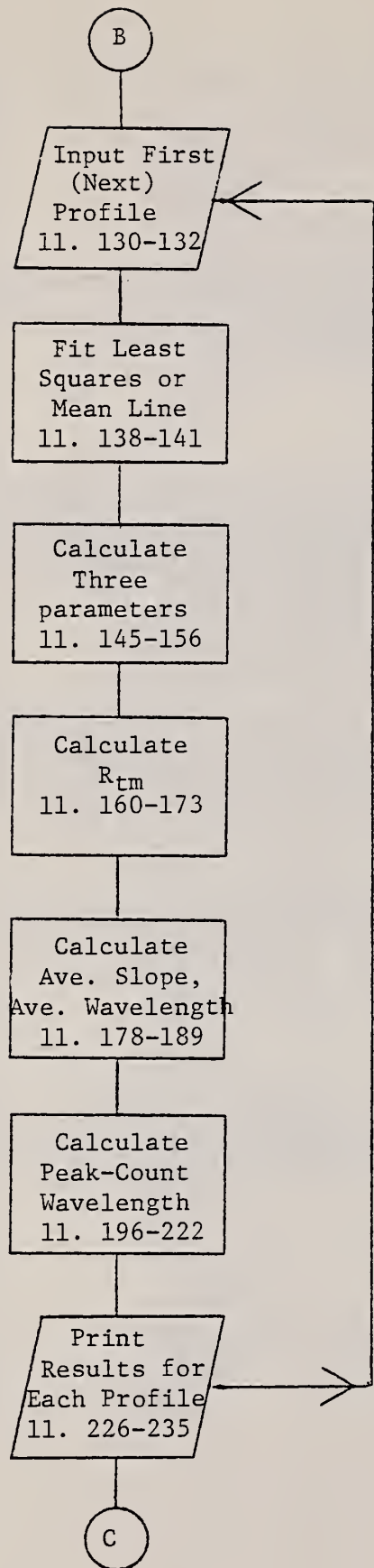
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The program reads the specimen I.D., the horizontal spacing of the data points, the instrument calibration constant, and the total number of profiles in the file. The information is then printed out.

The operator is prompted to input the sampling length for the R_{tm} calculation.

The operator is prompted to input the point-to-point spacing for the slope calculation.

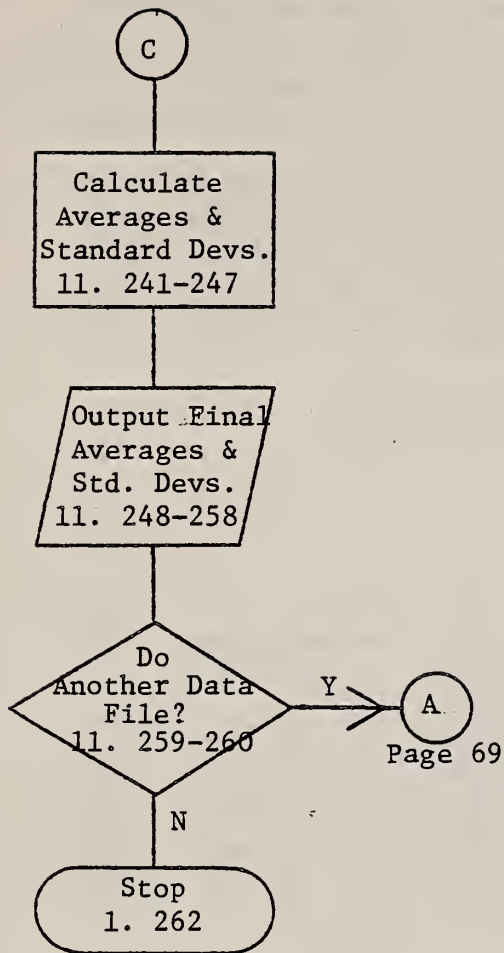
The operator is prompted to choose whether or not to fit a least squares line to the data to be used as the mean line.



The loop to calculate the parameters for each profile begins here.

A subroutine, either LEASQ or MEAN, is called.

R_a , R_q , and the skewness are calculated.



Call AVSD.

7.5 Example of SMORGAS Printout

The surface described here is a prototype calibration specimen that was specially machined to have a sinusoidal profile [11].

3:SRMS3MUM.DAT
SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78

POINT TO POINT SPACING = 0.9500 UM

KCAL = 0.672072E-02 UM/QUANTIZATION LEVEL

THERE ARE 8 PROFILES

799.900 UM = SAMPLING LENGTH FOR THE RTM CALCULATION.
THIS = 842 TIMES THE DATA POINT SPACING.

7.6000 UM = POINT-TO-POINT SPACING FOR THE SLOPE CALCULATION.
THIS = 8 TIMES THE DATA POINT SPACING.

THE DATA ARE BEING FITTED TO A LEAST SQUARES
STRAIGHT LINE FOR THIS CALCULATION.

POS	RA	RQ	RTM	AVE SLOPE	AVE WAVELENGTH	PC	PEAK-COUNT WAVELENGTH	SKEW
1	2.9611	3.3075	9.7921	0.1865E+00	99.749	37.5	101.308	-0.093
2	2.9737	3.3164	9.6644	0.1857E+00	100.623	37.0	102.677	-0.114
3	2.9786	3.3171	9.6896	0.1859E+00	100.676	37.5	101.308	-0.091
4	2.9675	3.3096	9.7417	0.1861E+00	100.174	37.5	101.308	-0.110
5	2.9388	3.2879	9.7299	0.1863E+00	99.115	37.5	101.308	-0.112
6	3.0074	3.3455	9.7098	0.1861E+00	101.539	37.5	101.308	-0.091
7	2.9701	3.3107	9.7047	0.1861E+00	100.289	37.5	101.308	-0.084
8	3.0121	3.3441	9.6476	0.1860E+00	101.757	37.5	101.308	-0.089

AVERAGES WITH THREE STANDARD DEVIATIONS OF THE MEAN (UM)

2.9762	3.3173	9.7100	0.1861E+00	100.490	101.479	-0.098
0.0263	0.0211	0.0501	0.2761E-03	0.960	0.532	0.013

8. PLOTSVIL

8.1 Summary

PLOTSVIL provides plots of surface profiles. The operator first chooses the name of the file that contains the desired data. Then the operator has the option of choosing,

1. the number of profiles to plot and which ones to plot,
2. the beginning and ending data points to be plotted (if the entire profile is to be plotted, the beginning point is 1 and the ending point is 4000), and
3. the length of the plot in cm.

8.2 Operating Systems Commands

```
1      ***** PLOTSVIL *****
2      *
3      VPHS1 SRF:PLOTSVIL,@1
4      VPHS2 SRF:PLOTSVIL,3
5      $W*
6      $W*
7      $W*
8      $W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
9      $W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
10     $W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI
11     "
12     $W*
13     $W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
14     $W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
15     $W* USE THE COMMAND "FASTMENU".
16     $EXIT
```

Notes:

"VPHS1" and "VPHS2" are sets of operating system commands used when programs that plot graphs are executed.

8.3 PLOTSVIL FORTRAN Program

```

1  C ***** SRF:PLOTSVIL.FTN  CALLED BY "PLOTSVIL" *****
2  C THIS ROUTINE IS USED TOGETHER WITH VERSATEC PLOTTING SOFTWARE
3  C TO PRODUCE PLOTS OF THE DIGITIZED SURFACE PROFILES.
4  C THE OPERATOR IS PROMPTED TO TYPE THE FILE NAME
5  C WHERE THE DATA CAN BE FOUND, THE NUMBERS OF THE
6  C PROFILES THAT ARE TO BE PLOTTED, THE NUMBER OF
7  C POINTS TO BE PLOTTED IN EACH ONE, AND THE
8  C LENGTH OF EACH PLOT ON THE OUTPUT PAGE.
9  C      T. VORBURGER, 1978 (LAST REVISION 3/82)
10 C
11 C
12     DIMENSION X(4002),Y(4002),SPEC(20)
13     INTEGER*2 DATFIL(9),PRONUM(50),START
14     REAL KCAL,LENGTH
15     DATA DATFIL(7),DATFIL(8),
16     1DATFIL(9)/'.D','.AT','. '/
17 C
18     1  FORMAT(F10.4)
19     2  FORMAT(16F5.0)
20     3  FORMAT(' WHAT IS THE DATA FILE? 12 CHARACTERS')
21     4  FORMAT(E13.6)
22     5  FORMAT(40A2)
23     6  FORMAT(I2)
24     7  FORMAT(' THERE ARE A TOTAL OF',1X,I2,1X,'PROFILES.'/
25     + ' HOW MANY PROFILES DO YOU WANT GRAPHED? I2 FORMAT')
26     8  FORMAT(' INPUT THE NUMBER OF THE NEXT PROFILE DESIRED.'/
27     1' USE I2 FORMAT. THEN HIT RETURN.')
28     9  FORMAT(1X,10F6.0)
29     10  FORMAT(' INPUT THE NUMBER OF THE FIRST PROFILE DESIRED.'/
30     1' USE I2 FORMAT. THEN HIT RETURN.')
31     11  FORMAT(' THE AVERAGE HEIGHT FOR PROFILE',1X,I2,1X,' IS',
32     +1X,F10.4,1X,'MICRONS')
33     12  FORMAT(20A4)
34     13  FORMAT(' AT WHAT POINT ON THE PROFILE WOULD YOU LIKE',
35     1' TO START THE PLOT?'/ ' ANSWER CAN BE BETWEEN 1 AND',
36     2' 4000 IN I4 FORMAT.')
37     14  FORMAT(I4)
38     15  FORMAT(' HOW MANY POINTS DO YOU WANT TO PLOT?',
39     1' I4 FORMAT AGAIN.')
40     16  FORMAT(' HOW LONG SHOULD THE PLOT BE IN CM?',
41     1' USE I3 FORMAT.')
42     17  FORMAT(I3)
43     18  FORMAT(' THERE ARE ONLY 4000 POINTS IN THE WHOLE',
44     1' PROFILE, DUMMY. TRY AGAIN.')
45 C
46 C FIRST, THE OPERATOR TYPES IN THE DATA FILE NAME
47 C AND THE NECESSARY PARAMETERS.
48 C
49     WRITE(6,3)
50     READ(5,5) (DATFIL(J),J=1,6)
51 C
52 C
53     CALL OPENW(9,DATFIL,0,0,0,STATUS)
54     READ(9,12,REC=2)SPEC
55     READ(9,1,REC=4)PTTOPT
56     WRITE(6,1)PTTOPT
57     READ(9,4,REC=6)KCAL
58     WRITE(6,1)KCAL

```

```

59      READ(9,6) ITOTAL
60      C
61      C
62      WRITE(6,7) ITOTAL
63      READ(5,6) ITOPRO
64      WRITE(6,10)
65      READ(5,6) PRONUM(1)
66      DO 50 M=2, ITOPRO
67          WRITE(6,8)
68      50  READ(5,6) PRONUM(M)
69      65  WRITE(6,13)
70          READ(5,14) N0
71          WRITE(6,15)
72          READ(5,14) N1
73          NCHECK = N1+N0
74          IF (NCHECK.LE.4001) GO TO 60
75          WRITE(6,18)
76          GO TO 65
77      60  WRITE(6,16)
78          READ(5,17) L
79          LENGTH = L/2
80          CALL PLOTS(0,0,0)
81          CALL FACTOR(.7874)
82          CALL PLOT(1.,3.,-3)
83      C
84      C THEN, THE PROGRAM READS THE RELEVANT PROFILE DATA.
85      C
86          DO 300 M=1, ITOPRO
87              RNUMBR=PRONUM(M)
88              START=(PRONUM(M)-1)*251+8
89              READ(9,6,REC=START) ITEST
90              READ(9,2)(Y(I), I=1,4000)
91              WRITE(6,9)(Y(I), I=1,10)
92      C
93      C THE AVERAGE MEAN HEIGHT OF THE PROFILE DATA IS
94      C CALCULATED HERE AS A CHECK.
95      C
96          SUM=0.0
97          DO 100 N=1,4000
98              SUM=SUM+Y(N)
99      100  CONTINUE
100         AVE=KCAL*SUM/4000.0
101         WRITE(3,11) PRONUM(M), AVE
102      C
103      C FINALLY, THE DATA ARE SCALED AND PLOTTED.
104      C
105          DO 110 I=1, N1
106              RI = I
107              X(I) = PTTOPT*RI
108              I1 = I+N0-1
109      110  Y(I) = KCAL*Y(I1)
110          CALL SCALE(Y,4.,N1,+1)
111          CALL SCALE(X,LENGTH,N1,+1)
112          CALL NEWPEN(-1)
113          CALL LINE(X,Y,N1,1,0,31)
114          CALL NEWPEN(1)
115          FIRST=Y(N1+1)-2.0*Y(N1+2)
116          CALL AXIS(0.,-2., 'DISTANCE (MICROMETERS)',-22,LENGTH,0.,

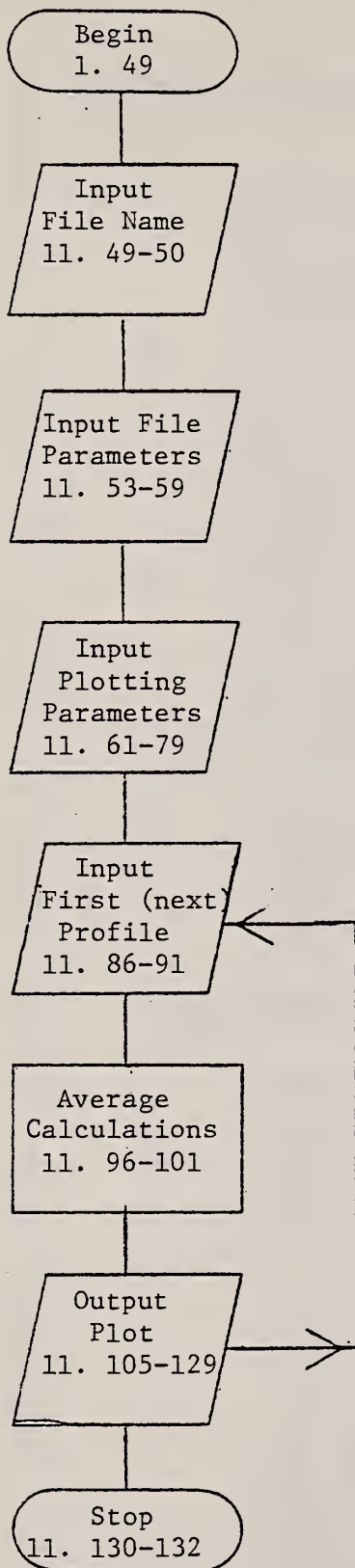
```

```

117      1 X(N1+1),X(N1+2))
118      CALL AXIS(0.,-2.,'VERTICAL DISPLACEMENT (MICROMETERS)',
119      1 35.8.,90.,FIRST,Y(N1+2))
120      CALL SYMBOL(1.,7.5.,2,SPEC,0.,+80)
121      CALL SYMBOL(1.,7.,2,'PROF NO = ',0.,+10)
122      CALL NUMBER(3.,7.,2,RNUMBR,0.,-1)
123      RN0 = N0
124      RN2 = N0+N1-1
125      CALL SYMBOL (4.,7.,2,'PTS      TO',0.,+11)
126      CALL NUMBER(4.8,7.,2,RN0,0.,-1)
127      CALL NUMBER(6.5,7.,2,RN2,0.,-1)
128      CALL PLOT(0.,0.,+23)
129      300 CONTINUE
130      CALL PLOT(0.,0.,999)
131      STOP
132      END

```


8.4 Flowchart for PLOTSVIL



The operator is prompted to type in the name of the data file to be examined.

The file is opened as a logical unit, and from it the program reads the specimen information, the point-to-point spacing, the calibration constant, and the total number of profiles.

The operator is prompted to type the total number of profiles to be plotted, the position of each one in the file and, for all the plots, the beginning data point, the number of data points, and the length of the plot output.

The Main Loop begins. Each profile is read.

The average height of the profile is calculated as information for the operator.

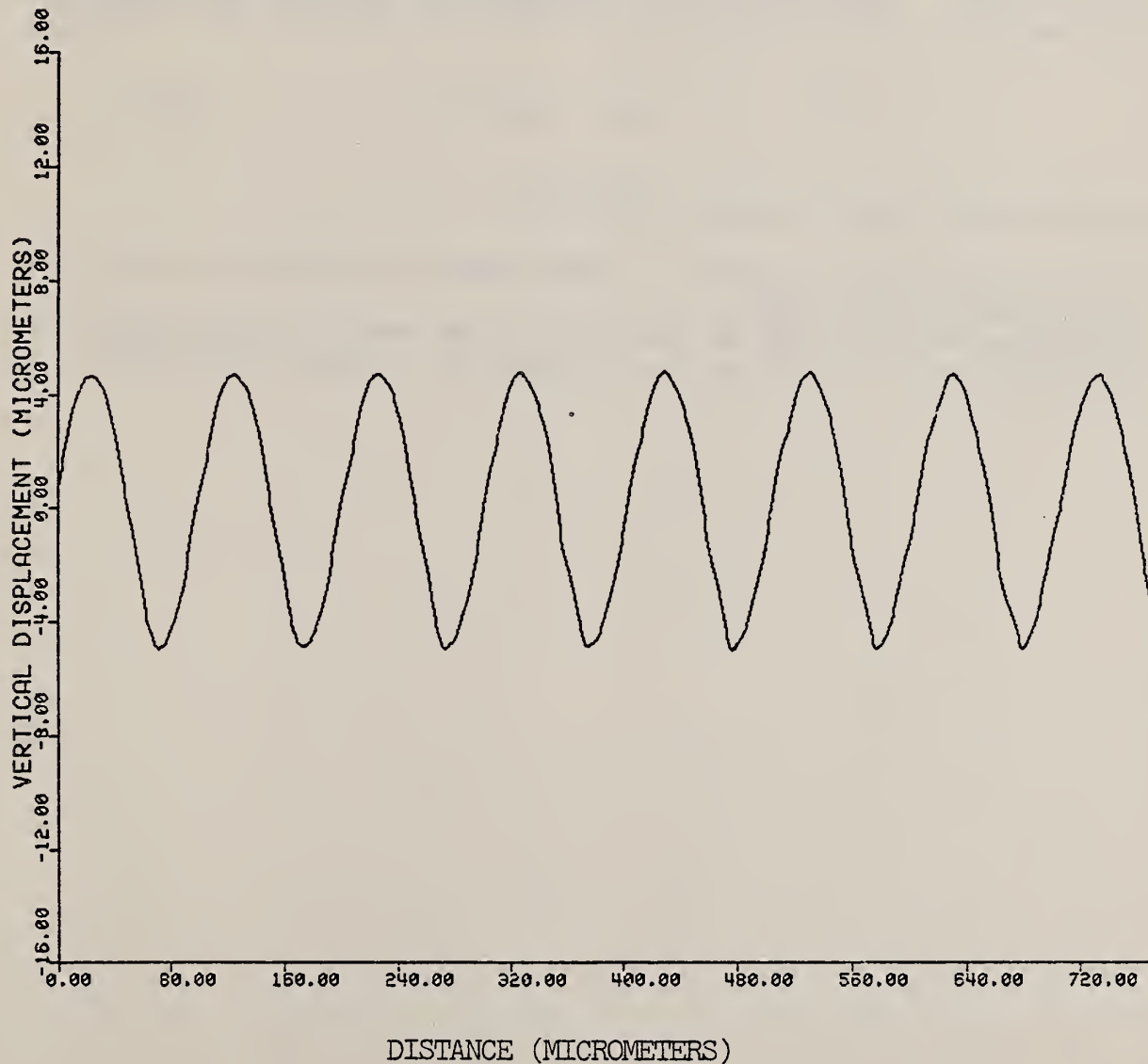
The data points are scaled and plotted.

Close plot sequence and stop.

8.5 Example of PLOTSVIL Plot

The sinusoidal surface depicted here was also discussed in sec. 7.5. The ADF, ACF, and PSD statistical functions for this profile are given in secs. 9.5, 10.5, and 11.5. Shown here is the first page of a graph which spans several pages.

SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78
PROF NO = 4 PTS 1 TO 2000



9. ADF

9.1 Summary

The ADF calculates and plots a histogram of the surface heights y_i for a profile. The 4000 data points, whose quantization levels range from -2048 to 2047, are sorted into 256 boxes containing 16 quantization levels each. The formula is

$$\text{ADF}(j) = N_j / (16 \times 4000 \times \text{KCAL}), \quad (9.1)$$

where N_j is the number of data points with height values falling in the j th box. The height of the j th box $y(j)$ is the abscissa of this function. It is given by

$$y(j) = 16 \text{ KCAL } (j - 257/2). \quad (9.2)$$

The operator has the following options:

1. choosing whether to fit a least squares straight line or a mean line to the data,
2. printing out the set of numbers which comprise the function,
3. choosing which profiles to analyze in a given data file.

9.2 Operating System Commands

```
1      ***** ADF *****
2      *
3      VPHS1 SRF:ADF,@1
4      VPHS2 SRF:ADF,3
5      $W*
6      $W*
7      $W*
8      $W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
9      $W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
10     $W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI
11     $W*
12     $W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
13     $W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
14     $W* USE THE COMMAND "FASTMENU".
15     $EXIT
```

9.3 ADF FORTRAN Program

```

1  C ***** SRF:ADF.FTN CALLED BY "ADF" *****
2  C THIS ROUTINE CALCULATES THE ADF FOR A SET OF N=4000 PROFILE
3  C DATA POINTS RANGING FROM -2048 TO +2047. THESE ARE SORTED
4  C INTO K=256 BOXES. THE NUMBER OF POINTS IN EACH BOX IS
5  C ACUMULATED INTO THE ARRAY ADF. THE ARRAY ORDNIT GIVES
6  C THE MIDPOOINT IN EACH BOX. THE PROCEDURE IS REPEATED FOR
7  C EACH SET OF PROFILE DATA. AN AVERAGE ADF, AVADF, IS ALSO
8  C CALCULATED AND PLOTTED.
9  C          T. VORBURGER (5/24/78)    REVISED (3/82)
10 C
11 C
12 C
13     DIMENSION ORDNIT(258),ADF(258),AVADF(258)
14     DIMENSION SPEC(15)
15     INTEGER START
16     REAL KCAL
17     INTEGER*2 PROFIL(4000),DATFIL(9)
18     INTEGER*2 QUERY1,QUERY2,QUERY3,PRONUM(50)
19     DATA DATFIL(7),DATFIL(8),DATFIL(9) / 'D','AT',' ' /
20 C
21     1  FORMAT(// ' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE' /
22     1' FIT TO THE DATA? "Y" OR "N"?')
23     2  FORMAT(40A2)
24     3  FORMAT(20A4)
25     4  FORMAT(E13.6)
26     5  FORMAT(E13.6,' UM/QUANTIZATION LEVEL = KCAL')
27     6  FORMAT(1H ,20A4)
28     7  FORMAT('***** ADF CALCULATIONS *****' /
29     1' WHAT IS THE FILE NAME? ' / THE FORMAT SHOULD',
30     2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
31     8  FORMAT(' THE FIRST & LAST PROFILE POINTS ARE ',I5,' & ',I5)
32     11  FORMAT(' POINT-TO-POINT SPACING (16 KCAL) = ',E13.6,' UM')
33     12  FORMAT(' TOTAL PROBABILITY = ',F8.4,' UM')
34     13  FORMAT(' RA CALCULATED FROM ADF = ',F9.5,' UM')
35     14  FORMAT(' RMS ROUGHNESS CALCULATED FROM ADF = ',
36     1F9.5,' UM')
37     16  FORMAT(8E12.4)
38     17  FORMAT(16I5)
39     18  FORMAT(12)
40     19  FORMAT(' I CAN" T READ THIS NAME. TRY AGAIN.')
41     21  FORMAT(1H / ' ADF DATA FOR POSITION ',I2)
42     22  FORMAT(1H / ' AVERAGE ADF FOR ALL ',I2,' POSITIONS')
43     23  FORMAT(' ERROR ON OPENW ROUTINE' /
44     1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
45     24  FORMAT(// ' THE DATA ARE BEING FITTED TO A' /
46     1' LEAST SQUARES STRAIGHT LINE FOR THIS' /
47     2' CALCULATION.')
48     25  FORMAT(// ' A MEAN VALUE IS BEING SUBTRACTED',
49     1' FROM THE DATA.')
50     27  FORMAT(1H ,40A2)
51     28  FORMAT(1H1,5X,' ***** ADF CALCULATIONS *****' /)
52     29  FORMAT(// ' DO YOU WANT A PRINTOUT OF THE ' /
53     1' ADF NUMBERS? "Y" OR "N"?')
54     31  FORMAT(1H1,' ***** WARNING! *****' / ' SOME OF THE'
55     1' PROFILE POINTS ARE OUTSIDE THE USUAL RANGE' /
56     2' OF -2047 TO 2048. YOU WILL GET SOME ERRORS' /
57     3' FOR THE ADF OF PROFILE', I3,'.')
58 C

```



```

59      C
60      C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
61      C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
62      C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
63      C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
64      C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
65      C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
66      C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
67      C
68      C
69      ITRY = 0
70      92  WRITE(6,7)
71          READ(5,2,ERR=93,END=502)(DATFIL(J),J=1,6)
72          CALL CARCON(3,1)
73          GO TO 94
74      93  ITRY = ITRY + 1
75          IF (ITRY .GT. 2) GO TO 502
76          WRITE(6,19)
77          GO TO 92
78      94  CALL OPENW(10,DATFIL,0,0,0,ISTAT)
79          IF(ISTAT.LT.1)GO TO 90
80          WRITE(6,23)
81          ITRY = ITRY + 1
82          IF (ITRY .GT. 2) GO TO 502
83          GO TO 92
84      C
85      C
86      90  READ(10,3,REC=2)SPEC
87          READ(10,4,REC=6)KCAL
88          K = 256
89          KPLUS = K+2
90          N = 4000
91          CONST = 16.*KCAL
92          IAVRG = 0
93          WRITE (6,1)
94          READ (5,2) QUERY1
95          WRITE(3,28)
96          WRITE(3,27) DATFIL
97          WRITE(3,6) SPEC
98          WRITE(3,5) KCAL
99          IF (QUERY1 .EQ. 'Y ') WRITE (3,24)
100         IF (QUERY1 .EQ. 'N ') WRITE (3,25)
101         WRITE(6,29)
102         READ(5,2) QUERY3
103      C
104         DO 110 J=1,K
105             ORDNIT(J) = CONST*(J-(K+1)/2.)
106      110     AVADF(J) =0.
107      C
108         CALL PLOTS(0,0,0)
109         CALL FACTOR(.7874)
110         CALL PLOT(.5,.5,-3)
111         READ(10,18,REC=7) NTOT
112      C
113      C
114      C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
115      C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
116      C

```

```

117 C
118 CALL CHOICE(NTOT,ITOPRO,PRONUM)
119 C
120 C
121 C THE MAIN LOOP TO CALCULATE AND PLOT THE ADF FOR EACH
122 C PROFILE BEGINS HERE
123 C
124 C
125 DO 300 M = 1,ITOPRO
126 RNUMBR = PRONUM(M)
127 START = (PRONUM(M) - 1)*251 + 8
128 READ (10,18,REC=START) NRUN
129 READ (10,17) PROFIL
130 WRITE(6,8) PROFIL(1),PROFIL(4000)
131 C
132 IF (QUERY1 .EQ. 'Y ') CALL LEASQ (PROFIL,4000)
133 IF (QUERY1 .EQ. 'N ') CALL MEAN (PROFIL,4000)
134 C
135 C
136 C NOW, WE CALCULATE THE NORMALIZED ADF AND THE CORRESPOND-
137 C ING ARRAY OF ORDINATE VALUES (IN UM). THE RESULTS ARE
138 C CHECKED BY CALCULATING THE RA AND THE RMS ROUGHNESS
139 C FROM THE ADF RESULTS.
140 C
141 C
142 MESSG = 0
143 DO 101 J=1,K
144 101 ADF(J) = 0.
145 C
146 DO 102 I= 1,N
147 BOX = .5*K*(1.+(PROFIL(I)/2048.))
148 IBOX = BOX+1.
149 IF (IBOX .LE. K .AND. IBOX .GE. 1) GO TO 102
150 IF (MESSG .GT. 0) GO TO 105
151 MESSG = 1
152 WRITE(3,31) NRUN
153 105 IF (IBOX .GT. K) IBOX=K
154 IF (IBOX .LT. 1) IBOX=1
155 102 ADF(IBOX) = ADF(IBOX) + 1.
156 C
157 DO 103 J=1,K
158 ADF(J) = ADF(J)/(CONST*4000.)
159 103 AVADF(J) = AVADF(J) + ADF(J)/ITOPRO
160 C
161 SUM0 = 0.
162 SUM1 = 0.
163 SUM2 = 0.
164 DO 104 J=1,K
165 SUM0 = SUM0 + ADF(J)
166 SUM1 = SUM1 +ADF(J)*ABS(ORDNIT(J))
167 104 SUM2 = SUM2+ADF(J)*(ORDNIT(J)**2)
168 C
169 UNITY = CONST*SUM0
170 RACHK = CONST*SUM1
171 RMSCHK = SQRT(CONST*SUM2)
172 C
173 C
174 C NOW, WE WRITE THE RESULTS ON THE PRINTER

```

```

175      C IF CALLED FOR.
176      C
177      C
178      C
179      IF (QUERY3 .EQ. 'N ') GO TO 601
180      CALL CARCON(3,0)
181      CALL CARCON(3,1)
182      WRITE(3,21) NRUN
183      WRITE(3,12) UNITY
184      WRITE(3,13) RACHK
185      WRITE(3,14) RMSCHK
186      WRITE(3,11) CONST
187      WRITE(3,16) (ADF(I), I=1,256)
188      C
189      C
190      C NOW, WE PLOT THE ADF.
191      C
192      C
193      601  CALL PLOTME(ORDNIT,ADF,KPLUS,'ADF (INVERSE UM)',16,
194            1'HEIGHT FROM MEAN LINE (UM)',26,SPEC,RMSCHK,
195            INRUN,IAVRG)
196      300  CONTINUE
197            CALL CLOSE (10,STATUS)
198      C
199      C
200      C THE MAIN LOOP IS COMPLETED. NOW, WE PRINT AND PLOT THE
201      C RESULTS FOR THE AVERAGE OF ALL RUNS.
202      C
203      C
204      IF (ITOPRO .EQ. 1) GO TO 501
205      IAVRG = 1
206      SUMAV0 = 0.
207      SUMAV1 = 0.
208      SUMAV2 = 0.
209      DO 301 J=1,K
210      SUMAV0 = SUMAV0+AVADF(J)
211      SUMAV1 = SUMAV1+AVADF(J)*ABS(ORDNIT(J))
212      301  SUMAV2 = SUMAV2+AVADF(J)*(ORDNIT(J)**2)
213      C
214      AVONE = CONST*SUMAV0
215      AVRA = CONST*SUMAV1
216      AVRMS = SQRT(CONST*SUMAV2)
217      CALL CARCON(3,0)
218      CALL CARCON(3,1)
219      WRITE(3,22) ITOPRO
220      WRITE(3,12) AVONE
221      WRITE(3,13) AVRA
222      WRITE(3,14) AVRMS
223      WRITE(3,11) CONST
224      WRITE(3,16) (AVADF(J),J=1,256)
225      CALL PLOTME(ORDNIT,AVADF,KPLUS,'ADF (INVERSE UM)',16,
226            1'HEIGHT FROM MEAN LINE (UM)',26,SPEC,AVRMS,ITOPRO,IAVRG)
227      501  CALL PLOT(0.,0.,999)
228      502  STOP
229      END
230      C
231      C
232      C SUBROUTINE FOR PLOTTING

```

```

233      C
234      C
235      SUBROUTINE PLOTME(X,Y,KPLUS,YLABEL,NYCHAR,XLABEL,
236      1NXCHAR,SPEC,RMS,NRUN,IAVRG)
237      C
238      C THE INPUT VARIABLES IN THIS SUBROUTINE ARE:
239      C X - THE ARRAY OF X VALUES,
240      C Y - THE ARRAY OF Y VALUES,
241      C KPLUS - THE SIZE OF THE ARRAYS PLUS 2 FOR STORAGE OF
242      C THE SCALE FACTOR AND INTIAL POINT IN THE PLOT,
243      C YLABEL, XLABEL - THE LABELS FOR THE X AND Y AXES,
244      C NXCHAR,NYCHAR - THE NUMBER OF CHARACTERS IN EACH LABEL,
245      C SPEC - A TITLE LABEL FOR THE PLOT,
246      C RMS - THE VALUE FOR THE RMS ROUGHNESS CALCULATED FROM
247      C THE FUNCTION,
248      C NRUN - THE PROFILE'S NUMBER IN THE INPUT DATA FILE,
249      C IAVRG - A FLAG TO TELL THE PROGRAM WHETHER AN
250      C INDIVIDUAL FUNCTION OR THE AVERAGE IS BEING
251      C CALCULATED.
252      C
253      DIMENSION X(KPLUS),Y(KPLUS),XLABEL(20),YLABEL(20)
254      DIMENSION SPEC(15)
255      RUN = NRUN
256      K = KPLUS-2
257      CALL SCALE(Y,8.,K,+1)
258      CALL SCALE(X,12.,K,+1)
259      CALL NEWPEN(-1)
260      Y(K+2) = -Y(K+2)
261      Y(K+1) = Y(K+1) - 8.*Y(K+2)
262      CALL LINE(Y,X,K,1,0,31)
263      CALL AXIS(0.,0.,YLABEL,-NYCHAR,9.,0.,Y(K+1),Y(K+2))
264      CALL AXIS(9.,0.,XLABEL,-NXCHAR,12.,90.,X(K+1),X(K+2))
265      CALL SYMBOL(0.,0.2,.2,SPEC,90.,60)
266      CALL SYMBOL(0.5,.2,.2,'RMS ROUGHNESS = ',90.,16)
267      IF ( RMS .LT. 0.01) GO TO 501
268      CALL NUMBER(0.5,3.4,.2,RMS,90.,+3)
269      CALL SYMBOL (0.4,4.6,0.2,7,90.,-1)
270      CALL SYMBOL (0.4,4.8,0.2,109,90.,-1)
271      GO TO 502
272      501 CALL NUMBER (0.5,3.4,0.2,RMS,90.,+5)
273      CALL SYMBOL (0.4,5.0,0.2,7,90.,-1)
274      CALL SYMBOL(0.4,5.2,0.2,109,90.,-1)
275      502 IF (IAVRG.EQ.1) GO TO 400
276      CALL SYMBOL(1.,2,.2,'ADF FOR POSITION',90.,16)
277      CALL NUMBER(1.,3.7,.2,RUN,90.,-1)
278      GO TO 401
279      400 CALL SYMBOL(1.,2,.2,'AVERAGE ADF FOR ALL ',90.,20)
280      CALL NUMBER(1.,4.3,.2,RUN,90.,-1)
281      CALL SYMBOL(1.,4.7,.2,' POSITIONS',90.,10)
282      401 CALL PLOT(0.,0.,+23)
283      RETURN
284      END
285      C
286      C
287      C SUBROUTINE FOR DETERMINING WHICH PROFILES IN
288      C THE FILE ARE ANALYZED
289      C
290      C

```

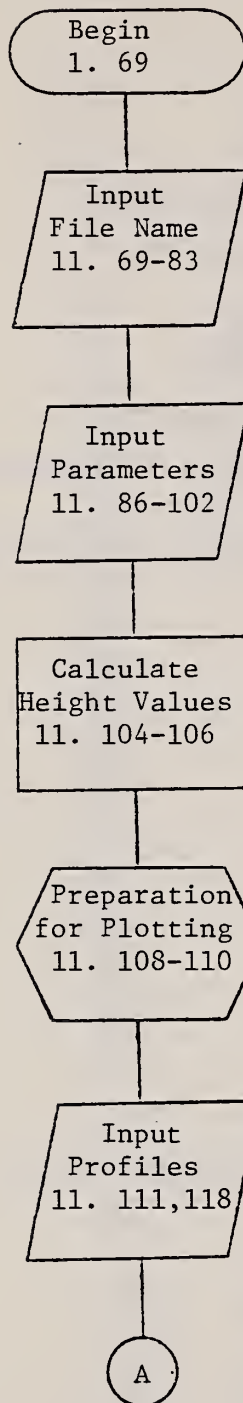
```

291          SUBROUTINE CHOICE(NTOT,ITOPRO,PRONUM)
292          INTEGER*2 QUERY2,PRONUM(50)
293      C
294      C
295      C INPUT VARIABLES:
296      C NTOT - THE TOTAL NUMBER OF PROFILES IN THE FILE.
297      C OUTPUT VARIABLES:
298      C ITOPRO - THE NUMBER OF PROFILES THAT THE OPERATOR CHOOSES
299      C           TO ANALYZE.
300      C PRONUM - THE ARRAY WHICH CONTAINS THE POSITIONS OF THE
301      C           CHOSEN PROFILES IN THE DATA FILE.
302      C
303      1      FORMAT(' THERE ARE',I3,' PROFILES'/// DO YOU',
304      1' WANT TO ANALYZE ALL OF THEM?'// "Y" OR "N"?')
305      2      FORMAT(I2)
306      3      FORMAT(/// HOW MANY PROFILES DO YOU WANT TO ANALYZE?'/
307      1' USE I2 FORMAT.')
308      4      FORMAT(/// TYPE THE NUMBER OF THE FIRST PROFILE TO BE ANALYZED.'/
309      1' USE I2 FORMAT.')
310      5      FORMAT(A2)
311      6      FORMAT(/// TYPE THE NUMBER OF THE NEXT PROFILE TO BE ANALYZED.'/
312      1' USE I2 FORMAT.')
313      C
314          WRITE (6,1) NTOT
315          READ (5,5) QUERY2
316          IF ( QUERY2 .EQ. 'N ') GO TO 131
317      C
318          ITOPRO = NTOT
319          DO 132 I = 1,ITOPRO
320      132      PRONUM(I) = I
321          RETURN
322      C
323      131      WRITE (6,3)
324          READ (5,2) ITOPRO
325          WRITE (6,4)
326          READ (5,2) PRONUM(1)
327          DO 133 I=2,ITOPRO
328          WRITE (6,6)
329      133      READ (5,2) PRONUM(I)
330          RETURN
331          END

```

The subroutines, MEAN and LEASQ, have already been listed in the program SMORGAS.

9.4 Flowchart for ADF



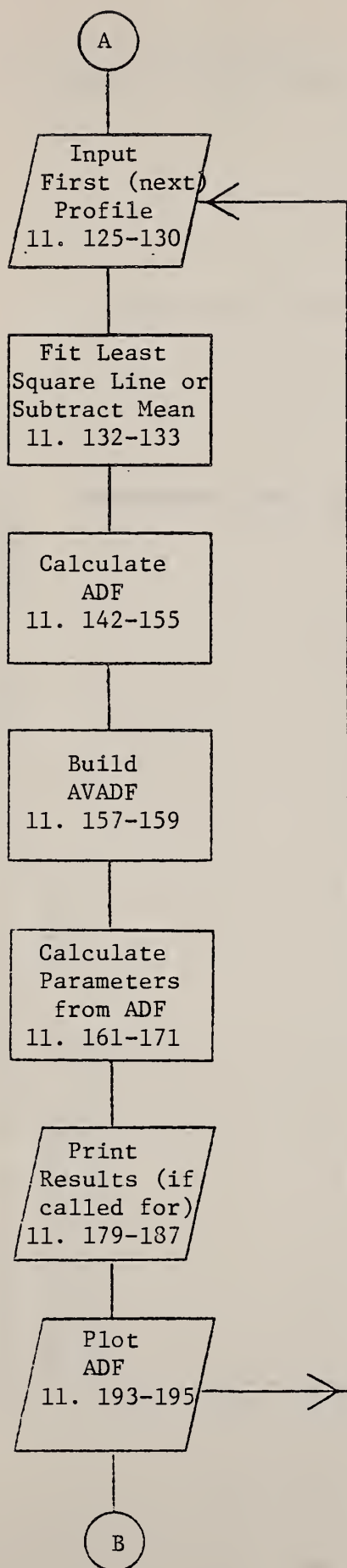
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The program reads the data file for the specimen I.D. and the instrument calibration constant and prompts the operator to choose whether or not to use a least squares fitted line as the mean line. If not the mean line is simply the mean of the data values. The operator must also choose whether or not to print out the set of numbers which comprise the ADF.

The set of height values, the independent variable in the ADF, are calculated here.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is called for this task.



The main loop to calculate and plot the ADF begins.

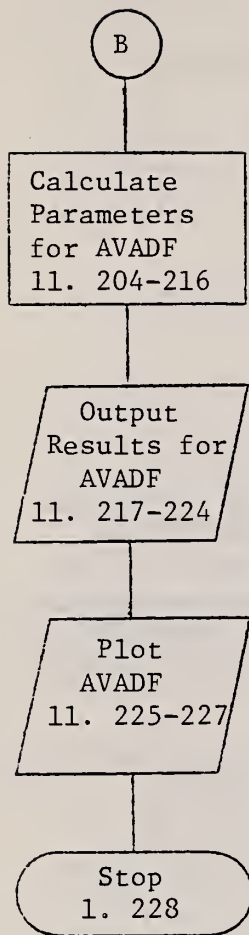
A subroutine, either LEASQ or MEAN, is called.

This section includes a set of statements (11. 150-154) to warn the operator and message the data when data points are out of range. This can happen when using the least squares fit with large amplitude data.

An average ADF is being calculated by adding each function to a summing array.

Three quantities are calculated to check the accuracy of the ADF: R_a , R_q , and the sum of the ADF values, which should equal unity.

The subroutine PLOTME is called here.



R_a , R_q , and the sum of the ADF values are calculated for AVADF.

The AVADF function and the three parameters are now printed.

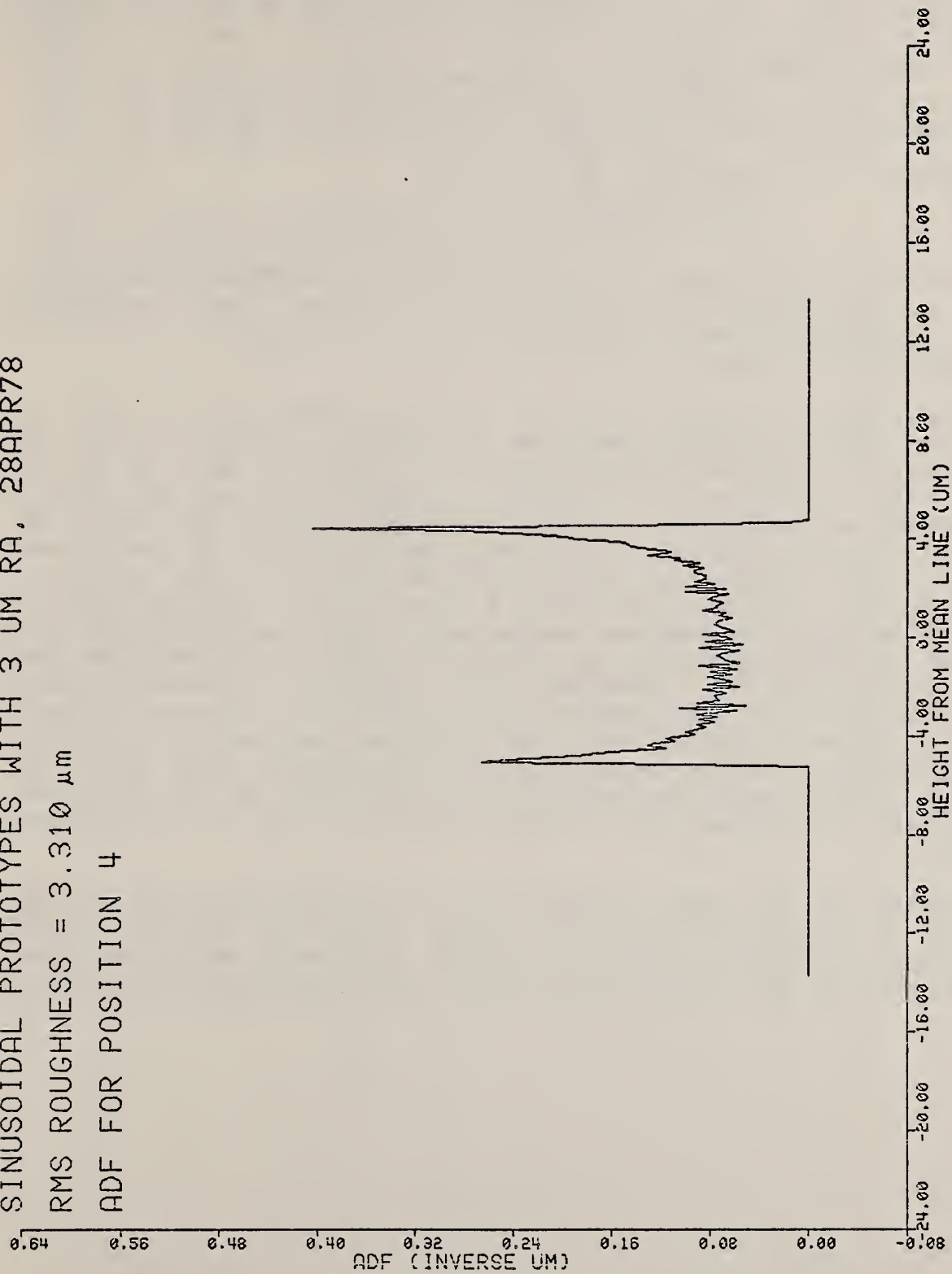
PLOTME is called again.

9.5 Example of ADF Plot for Sinusoidal Surface

SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78

RMS ROUGHNESS = 3.310 μm

ADF FOR POSITION 4



10. ACF

10.1 Summary

The ACF calculates and plots the autocorrelation function from the profile data. It is a quantitative measure of the similarity between a laterally shifted and an unshifted version of the profile and is given by

$$\text{ACF}(kT) = 1/[(N-k)R_q^2] \sum_{i=1}^{N-k} y_i y_{i+k}, \quad (10.1)$$

where T is the point-to-point spacing of the profile, k equals 0, 2, 4 . . . , and hence kT is the shift distance. The factor R_q^2 in the denominator normalizes the value of the ACF to unity at a shift distance of zero.

The above formula is known as the unbiased estimator [12] of the autocorrelation function because it divides the sum by $N-k$ to compensate for the decreasing number of terms in the sum as the shift index k increases. This estimator is not as widely recommended as the biased estimator which divides the sum by N , because it can be shown that the mean square error of the biased estimator is generally smaller than that of the unbiased estimator [12]. These errors should be small for random surfaces typically studied in surface metrology where the characteristic spacings of surface features should be much smaller than the length of the profile. Moreover, for calibration specimens with highly periodic profiles, such as the sinusoidal specimen already discussed in secs. 8.5 and 9.5, the unbiased estimator (eq. 10.1) is significantly better over the shift distance range shown in fig. 10.5. The biased estimator incorrectly yields an ACF whose oscillations would damp out significantly as the shift distance increases.

The operator has the same options here as in the ADF program. The function is calculated only for shift distances that are even multiples of the point spacing in order to save calculation time. With 501 points the calculation itself takes approximately 200 s.

10.2 Operating System Commands

```
1      ***** ACF *****
2      *
3      VPHS1 SRF:ACF,@1
4      VPHS2 SRF:ACF,3
5      $W*
6      $W*
7      $W*
8      $W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
9      $W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
10     $W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI
11     $W*
12     $W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
13     $W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
14     $W* USE THE COMMAND "FASTMENU".
15     $EXIT
```

10.3 ACF FORTRAN Program

```

1      C ***** SRF:ACF.FTN CALLED BY "ACF" *****
2      C THIS ROUTINE CALCULATES THE ACF FOR A SET OF N=4000 PROFILE
3      C DATA POINTS . THE PROCEDURE IS REPEATED FOR EACH SET OF PROFILE
4      C DATA CHOSEN TO BE ANALYZED. AN AVERAGE ACF IS ALSO CALCULATED
5      C AND PLOTTED.
6      C          T. VORBURGER (5/24/78)    REVISED (7/82)
7      C
8      C
9      C
10     DIMENSION ACF(503),AVACF(503)
11     DIMENSION SPEC(15),X(503)
12     INTEGER START
13     REAL KCAL
14     INTEGER*2 PROFIL(4000),DATFIL(9),N,ISHIFT,JPLUS
15     INTEGER*2 QUERY1,QUERY2,QUERY3,PRONUM(50)
16     DATA DATFIL(7),DATFIL(8),DATFIL(9)/'.D','AT','.'/
17     C
18     1      FORMAT(// ' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE' /
19     1' FIT TO THE DATA? "Y" OR "N"?')
20     2      FORMAT(40A2)
21     3      FORMAT(20A4)
22     4      FORMAT(E13.6)
23     5      FORMAT(E13.6,' UM/QUANTIZATION LEVEL = KCAL')
24     6      FORMAT(1H ,20A4)
25     7      FORMAT(5X,' ***** ACF CALCULATIONS *****' /
26     1' WHAT IS THE FILE NAME? ' / ' THE FORMAT SHOULD',
27     2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
28     8      FORMAT(I2,F10.4)
29     9      FORMAT(' APPROX RMS SLOPE (FROM ACF) = ',F6.4,'-',F6.4)
30     10     FORMAT(F10.4)
31     11     FORMAT(' POINT-TO-POINT SPACING = ',F9.4,' UM')
32     12     FORMAT(' POINT-TO-POINT SPACING OF ACF = ',F10.4,' UM')
33     13     FORMAT(' RA CALCULATED FROM ORIGINAL DATA = ',F9.5,' UM')
34     14     FORMAT(' RMS ROUGHNESS CALCULATED FROM ACF = ',
35     1F9.5,' UM')
36     15     FORMAT(I3)
37     17     FORMAT(16I5)
38     18     FORMAT(I2)
39     19     FORMAT(' I CAN" T READ THIS NAME. TRY AGAIN.')
40     20     FORMAT(' START LOOP')
41     21     FORMAT(1H /' ACF DATA FOR POSITION ',I2)
42     22     FORMAT(1H /' AVERAGE ACF FOR ALL ',I2,' POSITIONS')
43     23     FORMAT(' ERROR ON OPENW ROUTINE' /
44     1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
45     24     FORMAT(// ' THE DATA ARE BEING FITTED TO A' /
46     1' LEAST SQUARES STRAIGHT LINE FOR THIS' /
47     2' CALCULATION.')
48     25     FORMAT(// ' A MEAN VALUE IS BEING SUBTRACTED',
49     1' FROM THE DATA.')
50     26     FORMAT(' COMPLETE LOOP')
51     27     FORMAT(1H ,40A2)
52     28     FORMAT(1H1,5X,' ***** ACF CALCULATIONS *****' ///)
53     29     FORMAT(// ' DO YOU WANT A PRINTOUT OF THE ' /
54     1' ACF NUMBERS? "Y" OR "N"?')
55     30     FORMAT(' HOW MANY LAG POINTS DO YOU WANT TO CALCULATE?' /
56     1' THE UPPER LIMIT IS 501.')
57     31     FORMAT(' THE FIRST AND LAST PROFILE POINTS ARE',I5,' & ',I5)
58     32     FORMAT(// ' THE RMS SLOPE CANNOT BE CALCULATED FROM' /

```

```

59          1' THE ACF FOR THIS PROFILE BECAUSE SMALL ERRORS'/
60          2' IN THE CALCULATION LEAD TO A NEGATIVE SQUARE ROOT.')
61      33      FORMAT(10F8.4)
62      C
63      C
64      C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
65      C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
66      C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
67      C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
68      C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
69      C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
70      C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
71      C
72      C
73          ITRY = 0
74      92      WRITE(6,7)
75          READ(5,2,ERR=93,END=502)(DATFIL(J),J=1,6)
76          GO TO 94
77      93      ITRY = ITRY + 1
78          IF (ITRY .GT. 2) GO TO 502
79          WRITE(6,19)
80          GO TO 92
81      94      CALL OPENW(10,DATFIL,0,0,0,ISTAT)
82          IF(ISTAT.LT.1)GO TO 90
83          WRITE(6,23)
84          ITRY = ITRY + 1
85          IF (ITRY .GT. 2) GO TO 502
86          GO TO 92
87      C
88      C
89      90      WRITE (6,30)
90          READ (5,15) K
91          READ(10,3,REC=2)SPEC
92          READ(10,10,REC=4) PTTOPT
93          READ(10,4,REC=6)KCAL
94          KPLUS = K+2
95          N = 4000
96          AVACF0=0.
97          IAVRG = 0
98          WRITE (6,1)
99          READ (5,2) QUERY1
100         WRITE(3,28)
101         WRITE(3,27) DATFIL
102         WRITE(3,6) SPEC
103         WRITE (6,11) PTTOPT
104         DUBLPT = 2.*PTTOPT
105         WRITE(3,12) DUBLPT
106         WRITE(3,5) KCAL
107         IF (QUERY1 .EQ. 'Y ') WRITE (3,24)
108         IF (QUERY1 .EQ. 'N ') WRITE (3,25)
109         WRITE(6,29)
110         READ(5,2) QUERY3
111      C
112          DO 113 I=1,K
113          ISHIFT = 2*I - 2
114          X(I) = PTTOPT*ISHIFT
115          AVACF(I) = 0.
116      113      CONTINUE

```

```

117 C
118 CALL PLOTS(0,0,0)
119 CALL FACTOR(.7874)
120 CALL PLOT(.5,.5,-3)
121 READ(10,18,REC=7) NTOT
122 C
123 C
124 C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
125 C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
126 C
127 C
128 CALL CHOICE(NTOT,ITOPRO,PRONUM)
129 C
130 C
131 C THE MAIN LOOP TO CALCULATE AND PLOT THE ACF FOR EACH
132 C PROFILE BEGINS HERE
133 C
134 C
135 DO 300 M = 1,ITOPRO
136 RNUMBR = PRONUM(M)
137 START = (PRONUM(M) - 1)*251 + 8
138 READ (10,8,REC=START) NRUN,RA
139 READ (10,17) PROFIL
140 WRITE(6,31) PROFIL(1),PROFIL(4000)
141 C
142 IF (QUERY1 .EQ. 'Y ') CALL LEASQ (PROFIL,4000)
143 IF (QUERY1 .EQ. 'N ') CALL MEAN (PROFIL,4000)
144 C
145 C
146 C NOW WE DO THE DOUBLE LOOP IN THE AUTOCORRELATION CALCULATION.
147 C
148 C
149 WRITE (6,20)
150 DO 115 I=1,K
151 ISHIFT = 2*I - 2
152 SMPROD = 0.
153 MTOP = N-ISHIFT
154 RM= MTOP
155 DO 112 J=1,MTOP
156 JPLUS = J+ISHIFT
157 112 SMPROD = SMPROD+PROFIL(J)*PROFIL(JPLUS)
158 SMTEMP = SMPROD/RM
159 IF (I.GT.1) GO TO 111
160 ACF0 = SMTEMP
161 AVACF0 = AVACF0 + ACF0/ITOPRO
162 111 ACF(I) = SMTEMP/ACF0
163 115 AVACF(I) = AVACF(I) + ACF(I)/ITOPRO
164 WRITE(6,26)
165 RMS = SQRT(ACF0)*KCAL
166 IF (ACF(2).GT.1 .OR. ACF(3).GT.1) GO TO 117
167 SLOPE1 = RMS*SQRT(2.*(1.-ACF(2)))/DUBLPT
168 SLOPE2 = 0.5*RMS*SQRT(2.*(1.-ACF(3)))/DUBLPT
169 C
170 C
171 C NOW, WE WRITE THE ACF VALUES ONTO THE PRINTER IF CALLED FOR.
172 C
173 C
174 117 WRITE(3,21) NRUN

```

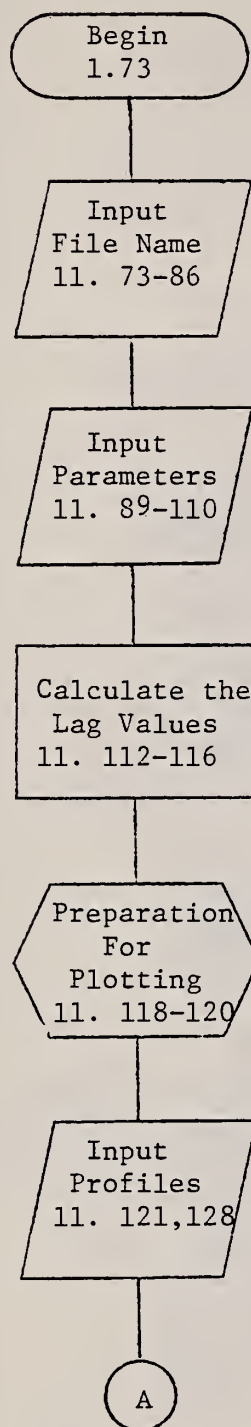
```

175          WRITE(3,13)RA
176          WRITE(3,14)RMS
177          IF (ACF(2).GT.1 .OR. ACF(3).GT.1) GO TO 116
178          WRITE(3,9)SLOPE1,SLOPE2
179          GO TO 118
180      116    WRITE(3,32)
181      118    IF (QUERY3 .EQ. 'N') GO TO 601
182          WRITE(3,33)(ACF(I),I=1,K)
183      C
184      C
185      C NOW, WE PLOT THE ACF.
186      C
187      C
188      601    CALL PLOTME(X,ACF,KPLUS,'NORMALIZED ACF',14,
189          1'SHIFT DISTANCE (UM)',19,SPEC,RMS,
190          1NRUN,IAVRG)
191      300    CONTINUE
192          CALL CLOSE (10,STATUS)
193      C
194      C
195      C THE MAIN LOOP IS COMPLETED. NOW, WE PRINT AND PLOT THE
196      C RESULTS FOR THE AVERAGE OF ALL RUNS.
197      C
198      C
199          IF(ITOPRO .EQ. 1) GO TO 501
200          IAVRG = 1.
201          AVRMS = SQRT(AVACF0)*KCAL
202          AVSLP1 = AVRMS*SQRT(2.*(1.-AVACF(2)))/DUBLPT
203          AVSLP2 = 0.5*AVRMS*SQRT(2.*(1.-AVACF(3)))/DUBLPT
204          CALL CLOSE(9,STATUS)
205          WRITE(3,22) ITOPRO
206          WRITE(3,14) AVRMS
207          WRITE(3,9) AVSLP1,AVSLP2
208          WRITE(3,33) (AVACF(I),I=1,K)
209          CALL PLOTME(X,AVACF,KPLUS,'NORMALIZED ACF',14,
210          1'SHIFT DISTANCE (UM)',19,SPEC,AVRMS,ITOPRO,IAVRG)
211      501    CALL PLOT(0.,0.,999)
212      502    STOP
213          END

```

The subroutines, MEAN and LEASQ, have been listed already in the SMORGAS program. CHOICE and PLOTME have been listed in the ADF program. The PLOTME subroutine in ACF differs from the other one by a single row of text in an output label.

10.4 Flowchart for ACF



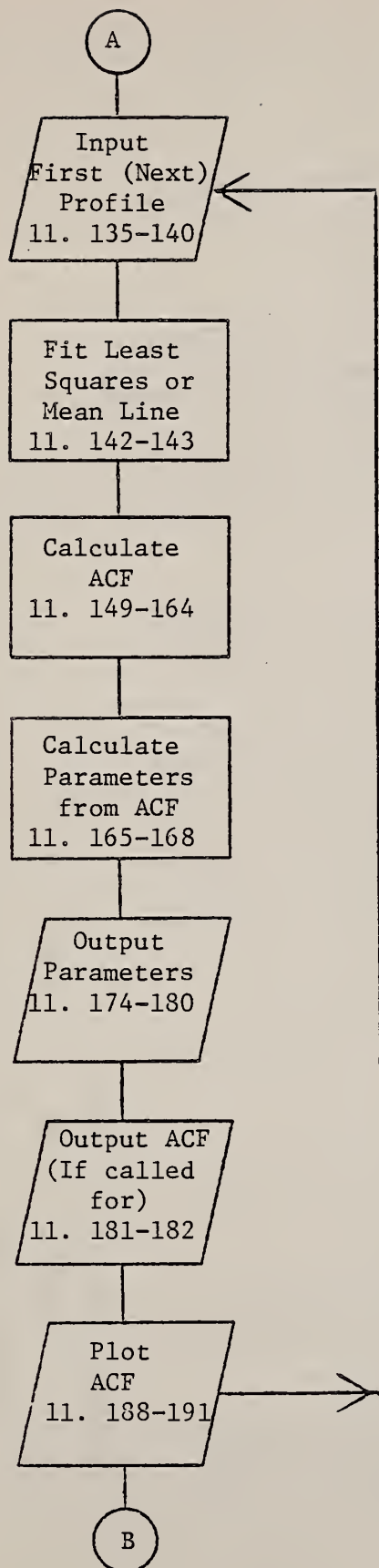
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The operator is prompted to type the number of points in the ACF. This value must range from 1 to 501. Then the program reads the specimen I.D., the horizontal spacing of the data points, and the instrument calibration constant and prompts the operator to choose whether or not to fit a least squares line to the data and whether to print out the values of the ACF function.

The lag values comprise the independent variable of the ACF.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the data file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is used for this task.



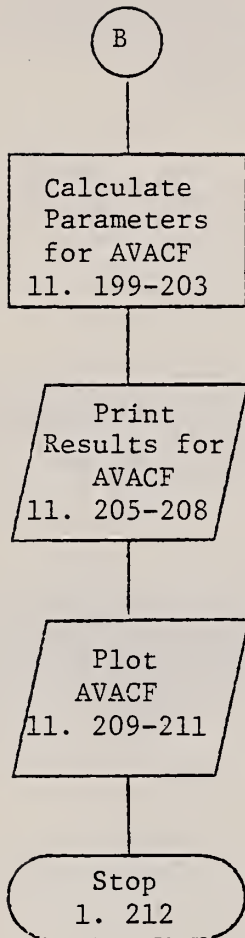
The main loop to calculate and plot the ACF begins. In addition to the profile data, the number of the profile and the stored value of R_a are also read from the file.

A subroutine, either LEASQ or MEAN, is called.

The ACF is calculated and normalized and the running average AVACF is built.

R_q and the rms slope are calculated from the ACF. In certain cases, a square root of a negative number may occur. If so, the calculation of rms slope is skipped.

The subroutine PLOTME is called here.



R_q and rms slope are calculated from AVACF.

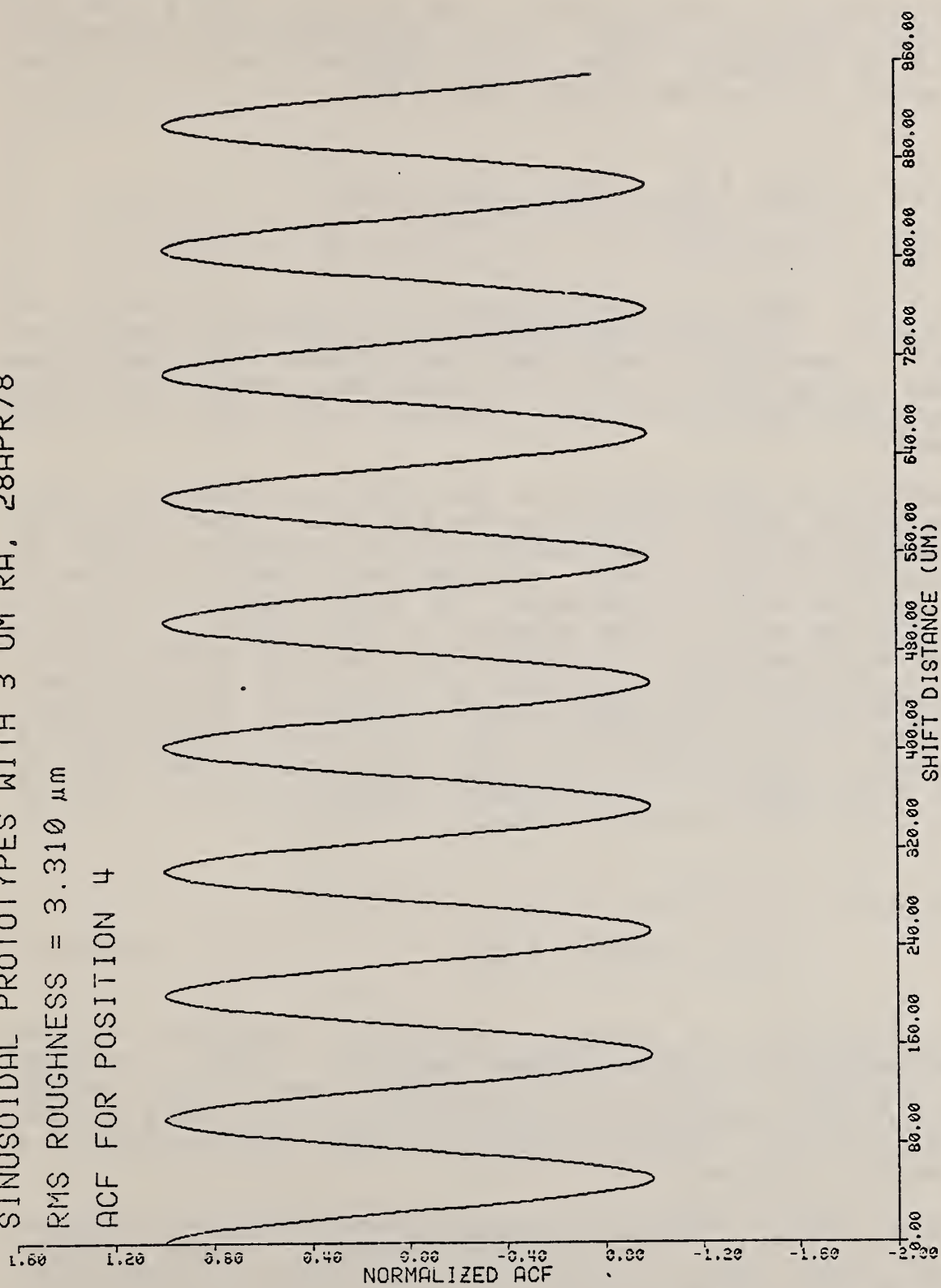
• PLOTME is called again. •

10.5 Example of ACF Plot

SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78

RMS ROUGHNESS = 3.310 μm

ACF FOR POSITION 4



11. PSD

11.1 Summary

The power spectral density resolves the surface profile into its spatial frequency components. It is calculated by taking the square of the absolute value of the Fourier transform of the surface profile. The basic algorithm is given by

$$\text{PSD}(f) = (2T/N) \left| \sum_{i=1}^N y_i \exp[-j2\pi f(i-1)T] \right|^2 \quad (11.1)$$

where $j = \sqrt{-1}$ and f is the spatial frequency with units of inverse length. The initial factor of 2 is included because we calculate a one-sided power spectrum ($0 < f < \infty$) and ignore the components having $-\infty < f \leq 0$. However, it is important to note that there are other modifications of the basic algorithm due to our requirements.

The transform is calculated by a Fast Fourier Transform (FFT) algorithm called EZFFT supplied by the NBS GAMS library [13]. In the present application, the use of this FFT algorithm speeds up the calculation of the Fourier transform by at least a factor of 20 over the conventional approach. The EZFFT algorithm requires a large storage array which depends on the number of profile points to be transformed. In order to save user memory space, the 4000 point profile is divided into two 2000 point halves. The least squares fitted mean line is found for each section, then the PSD is calculated for each. The 2000 point profile yields a digitized PSD with 1000 digitized points, and this is smoothed to 500 points in the present calculation with a Hanning procedure [14]. The smoothing formula is given by

$$\begin{aligned} \text{PSD}(f=I\Delta) = & 1/4 \text{ PSD}([I-1/2]\Delta) + 1/2 \text{ PSD}(I\Delta) \\ & + 1/4 \text{ PSD}([I+1/2]\Delta), \end{aligned} \quad (11.2)$$

where $\Delta (=2/NT)$ is the spacing in spatial frequency of the digitized points and I is an integer between 1 and 499. The last point in the PSD ($I = 500$) is left unsmoothed and is not plotted. The final PSD is calculated as the average of the PSD's for the two profile halves. The PSD calculation takes approximately 8 s on the present computer.

The operator has the option of choosing which profiles to analyze in a given data file. The PSD is plotted for each, and the operator has the option to print out the numbers for each PSD. The average PSD is automatically calculated, plotted, and printed as well.

11.2 Operating System Commands

```
1 ***** PSD *****
2 *
3 SE SYS 14.5
4 VPHS1 SRF:PSD,@1
5 SE SYS 20
6 VPHS2 SRF:PSD,3
7 $W*
8 $W*
9 $W*
10 $W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
11 $W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
12 $W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI".
13 $W*
14 $W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
15 $W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
16 $W* USE THE COMMAND "FASTMENU".
17 $EXIT
```

11.3 PSD FORTRAN Program

```

1  C ***** SRF:PSD.FTN CALLED BY "PSD" *****
2  C THIS ROUTINE CALCULATES THE PSD FOR A SET OF 4000 PROFILE
3  C DATA POINTS. THIS IS DONE BY BREAKING THE PROFILE INTO
4  C 2000 POINT SECTIONS, FITTING A LEAST SQUARES STRAIGHT
5  C LINE TO EACH SECTION, AND CRANKING THROUGH AN FFT WHICH
6  C YIELDS A 1000 POINT TRANSFORM. THIS ARRAY IS SMOOTHED
7  C INTO A 500 POINT POWER SPECTRUM FOR EACH 2000 POINT
8  C PROFILE SECTION. THE TWO HALVES ARE THEN AVERAGED TO
9  C YIELD THE FINAL PSD.
10 C THE PROCEDURE IS REPEATED FOR EACH SET OF PROFILE
11 C DATA CHOSEN TO BE ANALYZED. AN AVERAGE PSD IS ALSO CALCULATED
12 C AND PLOTTED.
13 C THE FFT ROUTINES CAME VIA SALLY HOWE IN THE
14 C CENTER FOR APPLIED MATHEMATICS.
15 C      T. VORBURGER (1/8/83)
16 C
17 C
18     DIMENSION PSD(500),PSD2(500),AVPSD(500),WSAVE(6015)
19     DIMENSION SPEC(15),F(501),PSDLOG(501)
20     INTEGER START
21     REAL KCAL,LAMBDA
22     INTEGER*2 PROFIL(2000),QUERY3,PRONUM(50)
23     CHARACTER*16 DATFIL
24     DATFIL(13:16) = '.DAT'
25
26     1  FORMAT(' THE DC LEVEL IS',E12.4,' um')
27     2  FORMAT(40A2)
28     3  FORMAT(20A4)
29     4  FORMAT(E13.6)
30     5  FORMAT(E13.6,' um/QUANTIZATION LEVEL = KCAL')
31     6  FORMAT(1H ,20A4)
32     7  FORMAT(5X,'***** PSD CALCULATIONS ***'/
33     1' WHAT IS THE FILE NAME?'// THE FORMAT SHOULD',
34     2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
35     8  FORMAT(I2,F10.4)
36     9  FORMAT(' HARMONIC PURITY IS',F10.4)
37     10  FORMAT(F10.4)
38     11  FORMAT(' POINT-TO-POINT SPACING =',F9.4,' um')
39     12  FORMAT(' POINT SPACING OF PSD =',F10.4,' INVERSE um')
40     13  FORMAT('// PROFILE',I3)
41     14  FORMAT(5E12.4)
42     15  FORMAT(A12)
43     16  FORMAT(' FREQUENCY RANGE IS FROM',F10.4,' TO',F10.4,
44     1' INVERSE um')
45     17  FORMAT(16I5)
46     18  FORMAT(I2)
47     19  FORMAT(' I CAN"T READ THIS NAME. TRY AGAIN.')
48     20  FORMAT('// AVERAGE PSD FOR ALL ',I2,' POSITIONS')
49     21  FORMAT(' RMS ROUGHNESS CALCULATED FROM THE INTEGRAL ',
50     1' OF THE PSD = ',F10.5,' um')
51     23  FORMAT(' I CAN"T OPEN THIS FILE'/
52     1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
53     24  FORMAT(' PSD (um)***3')
54     26  FORMAT(' AVE WAVELENGTH CALCULATED FROM PSD IS',F10.4,' um')
55     27  FORMAT(1H ,A16)
56     28  FORMAT(1H1,5X,' ***** PSD CALCULATIONS *****'///)
57     29  FORMAT('// DO YOU WANT A PRINTOUT OF THE '/
58     1' PSD NUMBERS? "Y" OR "N"?')

```

```

59      31      FORMAT(' THE FIRST AND LAST PROFILE POINTS ARE',15,' & ',15)
60      33      FORMAT(' BEGIN FFT')
61      34      FORMAT(' END FFT')
62      C
63      C
64      C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
65      C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
66      C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
67      C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
68      C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
69      C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
70      C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
71      C
72      C
73      ITRY = 0
74      92      WRITE(6,7)
75              READ(5,15,ERR=93,END=502)DATFIL(1:12)
76              GO TO 94
77      93      ITRY = ITRY + 1
78              IF (ITRY .GT. 2) GO TO 502
79              WRITE(6,19)
80              GO TO 92
81      94      OPEN(UNIT=10,ERR=91,FILE=DATFIL(1:16),STATUS='OLD',
82              1ACCESS='DIRECT',FORM='FORMATTED',RECL=80,BLANK='ZERO')
83              GO TO 90
84      91      WRITE(6,23)
85              ITRY = ITRY + 1
86              IF (ITRY .GT. 2) GO TO 502
87              GO TO 92
88      C
89      C
90      90      N = 2000
91              RN = FLOAT(N)
92              K = 500
93              KPLUS = K+1
94              IAVRG = 0
95      C
96      CALL EZFFT1(N,WSAVE)
97      C
98      READ(10,3,REC=2)SPEC
99      READ(10,10,REC=4) DELTAX
100     READ(10,4,REC=6)KCAL
101     WRITE(3,28)
102     WRITE(3,27) DATFIL(1:16)
103     WRITE(3,6) SPEC
104     WRITE (6,11) DELTAX
105     WRITE(3,11) DELTAX
106     WRITE(3,5) KCAL
107     WRITE(6,29)
108     READ(5,2) QUERY3
109     C
110     DELTAF = 2./(RN*DELTAX)
111     WRITE(3,12) DELTAF
112     DO 113 I=1,K-1
113     F(I) = I*DELTAF
114     AVPSD(I) = 0.
115     113 CONTINUE
116     C

```

```

117      CALL PLOTS(0,0,0)
118      CALL FACTOR(.7874)
119      CALL PLOT(.5,.5,-3)
120      READ(10,18,REC=7) NTOT
121      C
122      C
123      C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
124      C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
125      C
126      C
127      CALL CHOICE(NTOT,ITOPRO,PRONUM)
128      C
129      C
130      C THE MAIN LOOP TO CALCULATE AND PLOT THE PSD FOR EACH
131      C PROFILE BEGINS HERE
132      C
133      C
134      RTOPRO = ITOPRO
135      DO 300 M = 1,ITOPRO
136          RNUMBR = PRONUM(M)
137          START = (PRONUM(M) - 1)*251 + 8
138          READ (10,8, REC=START) NRUN,RA
139          WRITE(3,13) NRUN
140          DO 301 M1=1,2
141              READ (10,17) PROFIL
142              WRITE(6,31) PROFIL(1),PROFIL(2000)
143          C
144              CALL LEASQ(PROFIL,2000)
145          C
146          C
147              WRITE(6,33)
148              DO 310 I=1,N
149                  WSAVE(I) = KCAL*PROFIL(I)
150          310      CONTINUE
151              IF (M1 .EQ. 1) CALL EZPSDF(DELTA,N,DCLEV1,PSD,WSAVE)
152              IF (M1 .EQ. 2) CALL EZPSDF(DELTA,N,DCLEV2,PSD2,WSAVE)
153              WRITE(6,34)
154          301      CONTINUE
155              SQCHK = 0.
156              FCHK = 0.
157              DO 110 L=1,K
158                  PSD(L) = (PSD(L) + PSD2(L))/2.
159                  FCHK = FCHK + F(L)*PSD(L)
160                  SQCHK = SQCHK + PSD(L)
161                  PSDLOG(L) = ALOG10(PSD(L))
162                  AVPSD(L) = AVPSD(L) + PSD(L)/RTOPRO
163          110      CONTINUE
164              RMSCHK = SQRT(SQCHK*DELTAF)
165              LAMBDA = SQCHK/FCHK
166              FUNDAM = PSD(16) + PSD(17) + PSD(18) + PSD(19) + PSD(20)
167              PURITY = FUNDAM/SQCHK
168              DCLEVL = (DCLEV1 + DCLEV2)/2.
169          C
170          C
171          C NOW WE WRITE ALL THE RESULTS ON THE PRINTER.
172          C
173          C
174              WRITE(3,1) DCLEVL

```

```

175          WRITE(3,21) RMSCHK
176          WRITE(3,9) PURITY
177          WRITE(3,26) LAMBDA
178          IF (QUERY3 .EQ. 'N') GO TO 202
179          WRITE(3,24)
180          WRITE(3,14) (PSD(I),I=1,K)
181      202    WRITE(3,16) F(1),F(K-1)
182      C
183      C
184      C THE PLOT ROUTINES ARE NOW CALLED AND THE LOG OF THE
185      C PSD IS PLOTTED.
186      C
187      C
188          CALL PLOTME(F,PSDLOG,KPLUS,
189          1'LOG OF [PSD (um CUBED)]',23,
190          2'FREQUENCY (INVERSE um)',22,SPEC,RMSCHK,NRUN,IAVRG)
191      300    CONTINUE
192      C
193      C
194      C THE MAIN LOOP IS COMPLETE. NOW, WE PLOT AND PRINT THE
195      C RESULTS FOR THE AVERAGE PSD OF ALL THE SETS IN THE
196      C FILE.
197      C
198      C
199          IAVRG = 1
200          AVSQCK = 0.
201          AVFCHK = 0.
202          DO 302 L=1,K
203              AVSQCK = AVSQCK+AVPSD(L)
204              AVFCHK = AVFCHK+F(L)*AVPSD(L)
205              PSDLOG(L) = ALOG10(AVPSD(L))
206      302    CONTINUE
207          AVRMS = SQRT(AVSQCK*DEL TAF)
208          AVLMDA = AVSQCK/AVFCHK
209          WRITE(3,20) ITOPRO
210          WRITE(3,21) AVRMS
211          WRITE(3,26) AVLMDA
212          WRITE(3,24)
213          WRITE(3,14) (AVPSD(I),I=1,K)
214          WRITE(3,16) F(1),F(K-1)
215          CALL PLOTME(F,PSDLOG,KPLUS,
216          1'LOG OF [PSD (um CUBED)]',23,
217          2'FREQUENCY (INVERSE um)',22,SPEC,AVRMS,ITOPRO,IAVRG)
218          CALL PLOT(0.,0.,999)
219      502    STOP
220          END
221      C
222      C
223      C
224          SUBROUTINE PLOTME(X,Y,KPLUS,YLABEL,NYCHAR,XLABEL,
225          1 NXCHAR,SPEC,RMS,NRUN,IAVRG)
226          DIMENSION X(KPLUS),Y(KPLUS),XLABEL(20),YLABEL(20),
227          1 SPEC(15)
228          RUN = NRUN
229          K = KPLUS-2
230          CALL SCALE(Y,8.,K,+1)
231          CALL SCALE(X,12.,K,+1)
232          CALL NEWPEN(-1)

```



```

233      Y(K+2) = -Y(K+2)
234      Y(K+1) = Y(K+1) - 8.*Y(K+2)
235      CALL LINE(Y,X,K,1,0,31)
236      CALL AXIS(0.,0.,YLABEL,-NYCHAR,9.,0.,Y(K+1),Y(K+2))
237      CALL AXIS(9.,0.,XLABEL,-NXCHAR,12.,90.,X(K+1),X(K+2))
238      CALL SYMBOL(0.,2.,2,SPEC,90.,60)
239      CALL SYMBOL(.5.,2.,2,'RMS ROUGHNESS = ',90.,16)
240      IF (RMS .LT. 0.01) GO TO 501
241      CALL NUMBER(.5,3,4.,2,RMS,90.,+3)
242      CALL SYMBOL ( 0.4,4,6,0.2,7,90.,-1)
243      CALL SYMBOL(.4,4,8.,2,109,90.,-1)
244      GO TO 502
245 501    CALL NUMBER (0.5,3,4,0.2,RMS,90.,+5)
246      CALL SYMBOL (0.4,5,0,0.2,7,90.,-1)
247      CALL SYMBOL (0.4,5,2,0.2,109,90.,-1)
248 502    IF (IAVRG.EQ.1) GO TO 400
249      CALL SYMBOL(1.,2.,2,'PSD FOR POSITION',90.,16)
250      CALL NUMBER(1.,3,7.,2,RUN,90.,-1)
251      GO TO 401
252 400    CALL SYMBOL(1.,2.,2,'AVERAGE PSD FOR ALL ',90.,20)
253      CALL NUMBER(1.,4,3.,2,RUN,90.,-1)
254      CALL SYMBOL(1.,4,7.,2,'POSITIONS',90.,10)
255 401    CALL PLOT(0.,0.,+23)
256      RETURN
257      END
258  C
259  C
260  C THIS ROUTINE CALCULATES THE LEAST SQUARES STRAIGHT LINE FOR
261  C A SET OF 2000 EQUALLY SPACED DATA POINTS. IT CAN BE USED TO
262  C FILTER OUT ANY SLOPE IN A SET OF PROFILE DATA.
263  C
264  C
265      SUBROUTINE LEASQ(PROFIL,N)
266      INTEGER*2 PROFIL(N)
267 1      FORMAT(' AZERO',11X,'AONE'/2E15.7)
268      RN = N
269  C
270  C IN THE LEAST SQUARES FIT, WE HAVE ALREADY
271  C CALCULATED THE SUM OF THE X'S AND X SQUARES,
272  C WHOSE VALUES NEVER CHANGE. THIS AVOIDS HAVING
273  C TO CALCULATE THEIR VALUES IN THE LOOP.
274  C
275      SUMX1 = 2001000.
276      SUMX2 = 2.668667E9
277      SUMY1 = 0.
278      SUMXY = 0.
279      DO 101 I=1,N
280          SUMY1 = SUMY1 + PROFIL(I)
281          SUMXY = SUMXY + I*PROFIL(I)
282 101    CONTINUE
283  C
284      DELTA = RN*SUX2 - SUMX1*SUMX1
285      AZERO = (SUMX2*SUMY1 - SUMX1*SUMXY)/DELTA
286      AONE = (RN*SUMXY - SUMX1*SUMY1)/DELTA
287      WRITE(3,1) AZERO,AONE
288  C
289      DO 102 I=1,N
290          RI = I

```

```

291          SUB = AZERO + AONE*RI
292          X = 0.5
293          IF (SUB .LT. 0.) X = -X
294          NSUB = SUB + X
295          PROFIL(I) = PROFIL(I) - NSUB
296      102  CONTINUE
297          RETURN
298          END
299      C
300      C
301      C SUBROUTINE FOR DETERMINING WHICH PROFILES IN
302      C THE FILE ARE ANALYZED
303      C
304      C
305          SUBROUTINE CHOICE(NTOT,ITOPRO,PRONUM)
306          INTEGER*2 QUERY2,PRONUM(50)
307      C
308      1      FORMAT(' THERE ARE',I3,' PROFILES'/// DO YOU',
309      1' WANT TO ANALYZE ALL OF THEM?'// "Y" OR "N"?')
310      2      FORMAT(I2)
311      3      FORMAT(/// HOW MANY PROFILES DO YOU WANT TO ANALYZE?'/
312      1' USE I2 FORMAT.')
313      4      FORMAT(/// TYPE THE NUMBER OF THE FIRST PROFILE TO BE ANALYZED.'/
314      1' USE I2 FORMAT.')
315      5      FORMAT(A2)
316      6      FORMAT(/// TYPE THE NUMBER OF THE NEXT PROFILE TO BE ANALYZED.'/
317      1' USE I2 FORMAT.')
318      C
319          WRITE (6,1) NTOT
320          READ (5,5) QUERY2
321          IF ( QUERY2 .EQ. 'N ') GO TO 131
322      C
323          ITOPRO = NTOT
324          DO 132 I = 1,ITOPRO
325      132  PRONUM(I) = I
326          RETURN
327      C
328      131  WRITE (6,3)
329          READ (5,2) ITOPRO
330          WRITE (6,4)
331          READ (5,2) PRONUM(1)
332          DO 133 I=2,ITOPRO
333          WRITE (6,6)
334      133  READ (5,2) PRONUM(I)
335          RETURN
336          END
337      C
338      C
339      C
340      C
341          SUBROUTINE RFFTF (N,R,WSAVE)
342          DIMENSION      R(1)      ,WSAVE(1)
343          CALL RFFTF1 (N,R,WSAVE,WSAVE(N+1),WSAVE(2*N+1))
344          RETURN
345          END
346      C
347      C
348      C

```

```

349      C
350      C
351      SUBROUTINE RADF5 (IDO,L1,CC,CH,WA1,WA2,WA3,WA4)
352      DIMENSION      CC(IDO,L1,5)      ,CH(IDO,5,L1)
353      1      WA1(1)      ,WA2(1)      ,WA3(1)      ,WA4(1)
354      DATA TR11,TI11,TR12,TI12 / .309016994374947, .951056516295154,
355      1- .809016994374947, .587785252292473/
356      DO 101 K=1,L1
357          CR2 = CC(1,K,5)+CC(1,K,2)
358          CI5 = CC(1,K,5)-CC(1,K,2)
359          CR3 = CC(1,K,4)+CC(1,K,3)
360          CI4 = CC(1,K,4)-CC(1,K,3)
361          CH(1,1,K) = CC(1,K,1)+CR2+CR3
362          CH(IDO,2,K) = CC(1,K,1)+TR11*CR2+TR12*CR3
363          CH(1,3,K) = TI11*CI5+TI12*CI4
364          CH(IDO,4,K) = CC(1,K,1)+TR12*CR2+TR11*CR3
365          CH(1,5,K) = TI12*CI5-TI11*CI4
366      101 CONTINUE
367          IF (IDO .EQ. 1) RETURN
368          IDP2 = IDO+2
369          DO 103 K=1,L1
370              DO 102 I=3,IDO,2
371                  IC = IDP2-I
372                  DR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
373                  DI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
374                  DR3 = WA2(I-2)*CC(I-1,K,3)+WA2(I-1)*CC(I,K,3)
375                  DI3 = WA2(I-2)*CC(I,K,3)-WA2(I-1)*CC(I-1,K,3)
376                  DR4 = WA3(I-2)*CC(I-1,K,4)+WA3(I-1)*CC(I,K,4)
377                  DI4 = WA3(I-2)*CC(I,K,4)-WA3(I-1)*CC(I-1,K,4)
378                  DR5 = WA4(I-2)*CC(I-1,K,5)+WA4(I-1)*CC(I,K,5)
379                  DI5 = WA4(I-2)*CC(I,K,5)-WA4(I-1)*CC(I-1,K,5)
380                  CR2 = DR2+DR5
381                  CI5 = DR5-DR2
382                  CR5 = DI2-DI5
383                  CI2 = DI2+DI5
384                  CR3 = DR3+DR4
385                  CI4 = DR4-DR3
386                  CR4 = DI3-DI4
387                  CI3 = DI3+DI4
388                  CH(I-1,1,K) = CC(I-1,K,1)+CR2+CR3
389                  CH(I,1,K) = CC(I,K,1)+CI2+CI3
390                  TR2 = CC(I-1,K,1)+TR11*CR2+TR12*CR3
391                  TI2 = CC(I,K,1)+TR11*CI2+TR12*CI3
392                  TR3 = CC(I-1,K,1)+TR12*CR2+TR11*CR3
393                  TI3 = CC(I,K,1)+TR12*CI2+TR11*CI3
394                  TR5 = TI11*CR5+TI12*CR4
395                  TI5 = TI11*CI5+TI12*CI4
396                  TR4 = TI12*CR5-TI11*CR4
397                  TI4 = TI12*CI5-TI11*CI4
398                  CH(I-1,3,K) = TR2+TR5
399                  CH(IC-1,2,K) = TR2-TR5
400                  CH(I,3,K) = TI2+TI5
401                  CH(IC,2,K) = TI5-TI2
402                  CH(I-1,5,K) = TR3+TR4
403                  CH(IC-1,4,K) = TR3-TR4
404                  CH(I,5,K) = TI3+TI4
405                  CH(IC,4,K) = TI4-TI3
406      102 CONTINUE
102

```

```

407      103 CONTINUE
408      RETURN
409      END
410      C
411      C
412      C
413      C
414      C
415      SUBROUTINE EZFFT1 (N,WA,IFAC)
416      DIMENSION      WA(1)      ,IFAC(1)      ,NTRYH(4)
417      DATA NTRYH(1),NTRYH(2),NTRYH(3),NTRYH(4)/4,2,3,5/
418      1      ,TPI/6.28318530717959/
419      NL = N
420      NF = 0
421      J = 0
422      101 J = J+1
423      IF (J-4) 102,102,103
424      102 NTRY = NTRYH(J)
425      GO TO 104
426      103 NTRY = NTRY+2
427      104 NQ = NL/NTRY
428      NR = NL-NTRY*NQ
429      IF (NR) 101,105,101
430      105 NF = NF+1
431      IFAC(NF+2) = NTRY
432      NL = NQ
433      IF (NTRY .NE. 2) GO TO 107
434      IF (NF .EQ. 1) GO TO 107
435      DO 106 I=2,NF
436      IB = NF-I+2
437      IFAC(IB+2) = IFAC(IB+1)
438      106 CONTINUE
439      IFAC(3) = 2
440      107 IF (NL .NE. 1) GO TO 104
441      IFAC(1) = N
442      IFAC(2) = NF
443      ARGH = TPI/FLOAT(N)
444      IS = 0
445      NFM1 = NF-1
446      L1 = 1
447      IF (NFM1 .EQ. 0) RETURN
448      DO 111 K1=1,NFM1
449      IP = IFAC(K1+2)
450      L2 = L1*IP
451      IDO = N/L2
452      IPM = IP-1
453      ARG1 = FLOAT(L1)*ARGH
454      CH1 = 1.
455      SH1 = 0.
456      DCH1 = COS(ARG1)
457      DSH1 = SIN(ARG1)
458      DO 110 J=1,IPM
459      CH1H = DCH1*CH1-DSH1*SH1
460      SH1 = DCH1*SH1+DSH1*CH1
461      CH1 = CH1H
462      I = IS+2
463      WA(I-1) = CH1
464      WA(I) = SH1

```

```

465             IF (IDO .LT. 5) GO TO 109
466             DO 108 II=5,IDO,2
467                 I = I+2
468                 WA(I-1) = CH1*WA(I-3)-SH1*WA(I-2)
469                 WA(I) = CH1*WA(I-2)+SH1*WA(I-3)
470     108         CONTINUE
471     109         IS = IS+IDO
472     110         CONTINUE
473                 L1 = L2
474     111 CONTINUE
475             RETURN
476             END
477     C
478     C
479     C
480     C
481     C
482             SUBROUTINE EZFFT1 (N,WSAVE)
483             DIMENSION WSAVE(1)
484             CALL EZFFT1 (N,WSAVE(2*N+1),WSAVE(3*N+1))
485             RETURN
486             END
487     C
488     C
489     C
490     C
491     C
492     C THE FOLLOWING SUBROUTINE IS MODIFIED FROM THE
493     C EZFFT1 SUBROUTINE SUPPLIED TO US. THIS ONE CAL-
494     C CULATES THE POWER DIRECTLY, NOT JUST THE FOURIER
495     C AMPLITUDES.
496     C
497             SUBROUTINE EZPSDF (DELTAX,N,DCLEVL,PSD,WSAVE)
498     C
499     C             VERSION 3   JUNE 1979
500     C
501             DIMENSION PSD(1) ,WSAVE(1)
502     C
503             CALL RFFT1 (N,WSAVE,WSAVE(N+1))
504             CF = 1./FLOAT(N)
505             CFM = -CF
506             DCLEVL = CF*WSAVE(1)
507             NS2 = (N+1)/2
508             NS2M = NS2-1
509             DO 105 I=2,NS2M,2
510                 REAL = WSAVE(2*I)
511                 RIMAG = WSAVE(2*I+1)
512                 DNREAL = WSAVE(2*I-2)
513                 DNIMAG = WSAVE(2*I-1)
514                 UPREAL = WSAVE(2*I+2)
515                 UPIMAG = WSAVE(2*I+3)
516                 RMID = REAL*REAL + RIMAG*RIMAG
517                 RDN = DNREAL*DNREAL + DNIMAG*DNIMAG
518                 RUP = UPREAL*UPREAL + UPIMAG*UPIMAG
519     C
520     C THE FACTOR OF 2 IS NEEDED BECAUSE WE ARE COMPUTING A
521     C ONE-SIDED PSD. WHEN WE INTEGRATE TO GET POWER, WE SHOULD
522     C INCLUDE THE NEGATIVE FREQUENCY CONTRIBUTION. INSTEAD,

```



```

523 C WE MULTIPLY THE POSITIVE SIDE BY A FACTOR OF 2.
524 C
525 C
526 C IN ADDITION, INSTEAD OF CALCULATING A 1000 POINT POWER
527 C SPECTRUM FROM THE 2000 POINT PROFILE, WE CALCULATE A SMOOTHED
528 C 500 POINT POWER SPECTRUM.
529 C
530 C
531 PSD(I/2) = 2.*DELTA X*CF*(.25*RDN + .5*RMID + .25*RUP)
532 105 CONTINUE
533 PSD(NS2/2) = 2.*DELTA X*CF*WSAVE(N)*WSAVE(N)
534 RETURN
535 END
536 C
537 C
538 C
539 C
540 C
541 SUBROUTINE RADF2 (IDO,L1,CC,CH,WA1)
542 DIMENSION CH(IDO,2,L1) ,CC(IDO,L1,2)
543 1 WA1(1)
544 DO 101 K=1,L1
545 CH(1,1,K) = CC(1,K,1)+CC(1,K,2)
546 CH(IDO,2,K) = CC(1,K,1)-CC(1,K,2)
547 101 CONTINUE
548 IF (IDO-2) 107,105,102
549 102 IDP2 = IDO+2
550 DO 104 K=1,L1
551 DO 103 I=3,IDO,2
552 IC = IDP2-I
553 TR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
554 TI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
555 CH(I,1,K) = CC(I,K,1)+TI2
556 CH(IC,2,K) = TI2-CC(I,K,1)
557 CH(I-1,1,K) = CC(I-1,K,1)+TR2
558 CH(IC-1,2,K) = CC(I-1,K,1)-TR2
559 103 CONTINUE
560 104 CONTINUE
561 IF (MOD(IDO,2) .EQ. 1) RETURN
562 105 DO 106 K=1,L1
563 CH(1,2,K) = -CC(IDO,K,2)
564 CH(IDO,1,K) = CC(IDO,K,1)
565 106 CONTINUE
566 107 RETURN
567 END
568 C
569 C
570 C
571 C
572 C
573 SUBROUTINE RADF4 (IDO,L1,CC,CH,WA1,WA2,WA3)
574 DIMENSION CC(IDO,L1,4) ,CH(IDO,4,L1)
575 1 WA1(1) ,WA2(1) ,WA3(1)
576 DATA HSQT2 / .7071067811865475/
577 DO 101 K=1,L1
578 TR1 = CC(1,K,2)+CC(1,K,4)
579 TR2 = CC(1,K,1)+CC(1,K,3)
580 CH(1,1,K) = TR1+TR2

```

```

581      CH(IDO,4,K) = TR2-TR1
582      CH(IDO,2,K) = CC(1,K,1)-CC(1,K,3)
583      CH(1,3,K) = CC(1,K,4)-CC(1,K,2)
584 101 CONTINUE
585      IF (IDO-2) 107,105,102
586 102 IDP2 = IDO+2
587      DO 104 K=1,L1
588          DO 103 I=3,IDO,2
589              IC = IDP2-I
590              CR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
591              CI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
592              CR3 = WA2(I-2)*CC(I-1,K,3)+WA2(I-1)*CC(I,K,3)
593              CI3 = WA2(I-2)*CC(I,K,3)-WA2(I-1)*CC(I-1,K,3)
594              CR4 = WA3(I-2)*CC(I-1,K,4)+WA3(I-1)*CC(I,K,4)
595              CI4 = WA3(I-2)*CC(I,K,4)-WA3(I-1)*CC(I-1,K,4)
596              TR1 = CR2+CR4
597              TR4 = CR4-CR2
598              TI1 = CI2+CI4
599              TI4 = CI2-CI4
600              TI2 = CC(1,K,1)+CI3
601              TI3 = CC(1,K,1)-CI3
602              TR2 = CC(I-1,K,1)+CR3
603              TR3 = CC(I-1,K,1)-CR3
604              CH(I-1,1,K) = TR1+TR2
605              CH(IC-1,4,K) = TR2-TR1
606              CH(I,1,K) = TI1+TI2
607              CH(IC,4,K) = TI1-TI2
608              CH(I-1,3,K) = TI4+TR3
609              CH(IC-1,2,K) = TR3-TI4
610              CH(I,3,K) = TR4+TI3
611              CH(IC,2,K) = TR4-TI3
612 103 CONTINUE
613 104 CONTINUE
614      IF (MOD(IDO,2) .EQ. 1) RETURN
615 105 CONTINUE
616      DO 106 K=1,L1
617          TI1 = -HSQT2*(CC(IDO,K,2)+CC(IDO,K,4))
618          TR1 = HSQT2*(CC(IDO,K,2)-CC(IDO,K,4))
619          CH(IDO,1,K) = TR1+CC(IDO,K,1)
620          CH(IDO,3,K) = CC(IDO,K,1)-TR1
621          CH(1,2,K) = TI1-CC(IDO,K,3)
622          CH(1,4,K) = TI1+CC(IDO,K,3)
623 106 CONTINUE
624 107 RETURN
625      END
626 C
627 C
628 C
629 C
630 C
631 SUBROUTINE RFFTF1 (N,C,CH,WA,IFAC)
632 DIMENSION CH(1) ,C(1) ,WA(1) ,IFAC(1)
633 NF = IFAC(2)
634 NA = 1
635 L2 = N
636 IW = N
637 DO 111 K1=1,NF
638     KH = NF-K1

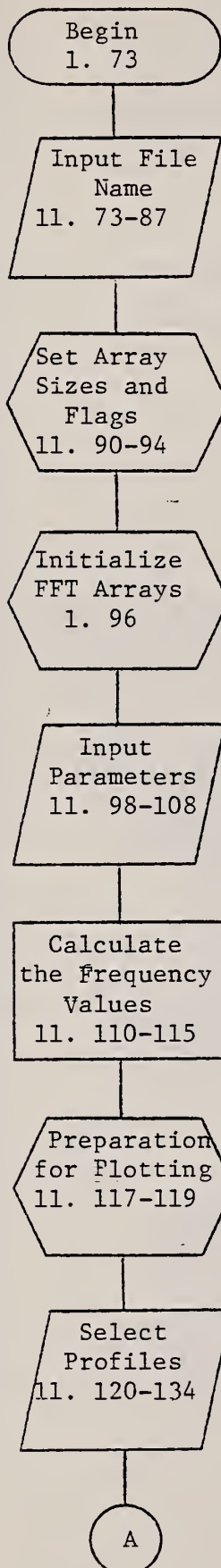
```

```

639      IP = IFAC(KH+3)
640      L1 = L2/IP
641      IDO = N/L2
642      IDL1 = IDO*L1
643      IW = IW-(IP-1)*IDO
644      NA = 1-NA
645      IF (IP .NE. 4) GO TO 102
646      IX2 = IW+IDO
647      IX3 = IX2+IDO
648      IF (NA .NE. 0) GO TO 101
649      CALL RADF4 (IDO,L1,C,CH,WA(IW),WA(IX2),WA(IX3))
650      GO TO 110
651      101  CALL RADF4 (IDO,L1,CH,C,WA(IW),WA(IX2),WA(IX3))
652      GO TO 110
653      102  IF (IP .NE. 2) GO TO 104
654      IF (NA .NE. 0) GO TO 103
655      CALL RADF2 (IDO,L1,C,CH,WA(IW))
656      GO TO 110
657      103  CALL RADF2 (IDO,L1,CH,C,WA(IW))
658      GO TO 110
659      104  IX2 = IW+IDO
660      IX3 = IX2+IDO
661      IX4 = IX3+IDO
662      IF (NA .NE. 0) GO TO 107
663      CALL RADF5 (IDO,L1,C,CH,WA(IW),WA(IX2),WA(IX3),WA(IX4))
664      GO TO 110
665      107  CALL RADF5 (IDO,L1,CH,C,WA(IW),WA(IX2),WA(IX3),WA(IX4))
666      110  L2 = L1
667      111  CONTINUE
668      IF (NA .EQ. 1) RETURN
669      DO 112 I=1,N
670      C(I) = CH(I)
671      112  CONTINUE
672      RETURN
673      END

```

11.4 Flowchart for PSD



The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

There are 2000 points to be analyzed for each calculation of a 500 point PSD.

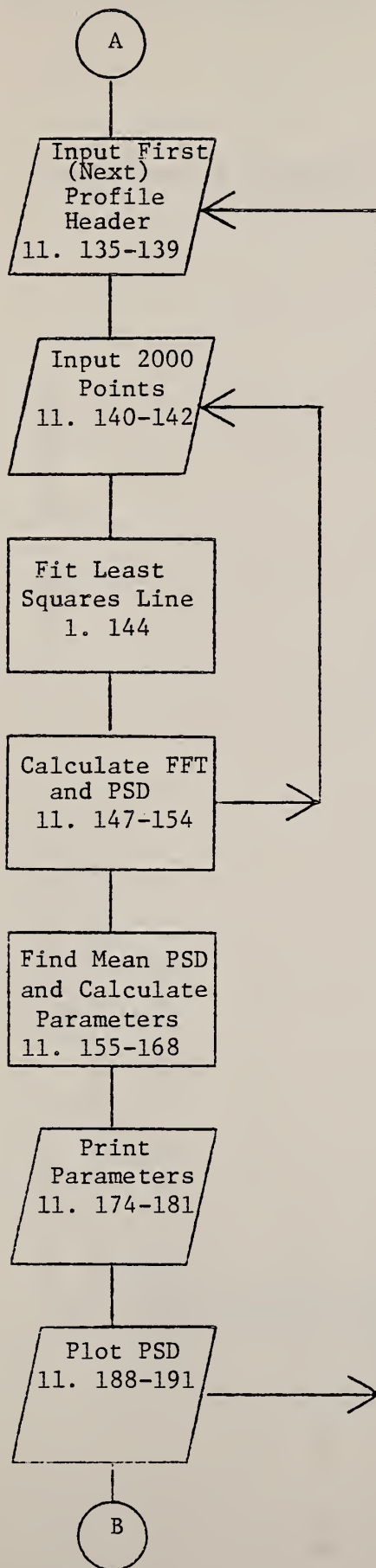
The main program calls EZFFTI, which in turn calls EZFFT1.

The program reads the specimen I.D., the horizontal spacing of the data points, and the instrument calibration constant and prompts the operator to choose whether or not to print the values of the PSD.

These frequency values comprise the independent variable of the PSD.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the data file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is used for this task.



The main loop to calculate and plot the PSD begins.

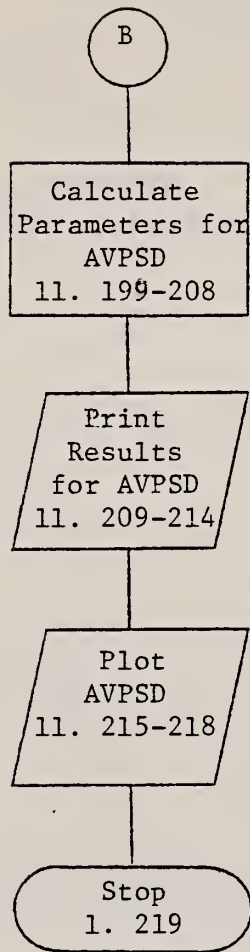
The inner loop to calculate the PSD for each half of the profile begins.

The subroutine EZPSDF is called and in turn calls several others.

The mean PSD for the two halves is averaged in this section. The PSD is used to calculate values for R_q and a mean wavelength parameter.

The PSD values are also printed if called for.

PLOTME is called here.



R_q and the mean wavelength parameter are calculated again.

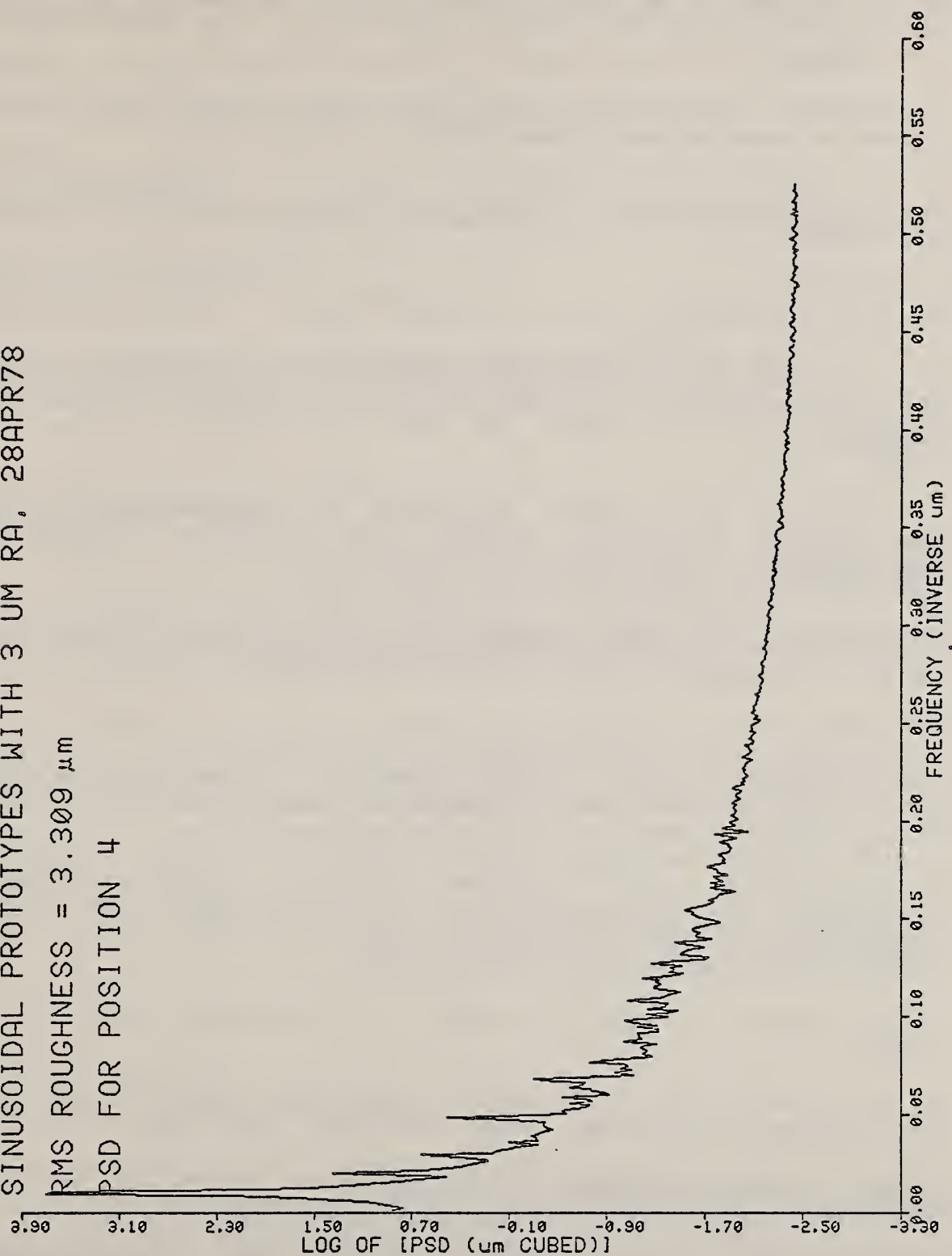
PLOTME is called again.

11.5 Example of PSD Plot

SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78

RMS ROUGHNESS = 3.309 μm

PSD FOR POSITION 4



12. References

1. D. Byrne, P. A. Fitzsimmons, A. K. Brook, Paper E-13 in Proc. Symposium on Ship Costs and Energy '82 (Society of Naval Architects and Marine Engineers, New York, To be published).
2. B. Scheffer, Comparison de Differentes Normes Nationales (Regie Nationale des Usines Renault, Paris, 1969).
3. E. L. Church, M. R. Howells, and T. V. Vorburger in Reflecting Optics for Synchrotron Radiation, Proc. SPIE 315, 202 (1981).
4. T. V. Vorburger, E. C. Teague, and F. E. Scire, Dimensions/NBS 62, No. 11, 18 (Nov. 1978).
5. E. C. Teague, Evaluation, Revision, and Application of the NBS Stylus/Computer System for the Measurement of Surface Roughness, NBS Tech. Note 902 (Dept. of Commerce, Wash., DC, 1976).
6. E. C. Teague, Evaluation, Revision, and Application of the NBS Stylus Computer System for Surface Roughness Measurement: Minicomputer Software, NBS Internal Report 75-924 (Dept. of Commerce, Wash., DC, 1975).
7. M. G. Natrella, Experimental Statistics, NBS Handbook 91 (Dept. of Commerce, Wash., DC, 1963) p. 1-10.
8. Ref. 5, p. 15.
9. R. C. Spragg and D. J. Whitehouse, Proceedings of the Institution of Mechanical Engineers, 185, (47/71) 697 (1970-71).
10. T. V. Vorburger, F. E. Scire, and E. C. Teague, Surface Roughness Measurements of Circular Disks and Their Correlation with Hydrodynamic Drag, NBS Tech. Note 1151 (Dept. of Commerce, Wash., DC, 1982).
11. E. C. Teague, F. E. Scire, and T. V. Vorburger, Wear 83, 61 (1982).
12. G. M. Jenkins and D. G. Watts, Spectral Analysis and Its Applications (Holden-Day, San Francisco, 1968) pp. 171 ff.
13. Guide to Available Mathematical Software, Center for Applied Mathematics (Nat. Bur. Stds., Wash. DC, 1981).
14. J. S. Bendat and A. G. Piersol, Random Data: Analysis and Measurement Procedures (Wiley-Interscience, New York, 1971) p. 318.

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 83-2703	2. Performing Organ. Report No.	3. Publication Date July 1983
---	---	---------------------------------	----------------------------------

4. TITLE AND SUBTITLE

FASTMENU: A Set of FORTRAN Programs for Analyzing Surface Texture

5. AUTHOR(S)

Theodore V. Vorburger

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A set of FORTRAN programs for surface texture analysis is described. These programs were developed for use with a minicomputer that is interfaced to stylus type instruments. The programs 1) perform data acquisition from the stylus instruments, 2) store the data on magnetic disk, and 3) perform statistical analyses for parameters such as the roughness average R_a , rms roughness R_q , and for the autocorrelation function and amplitude density function.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

amplitude density function; autocorrelation function; calibration; digitization; metrology; minicomputer; roughness; statistics; stylus; surface metrology; surface profile; surface texture

13. AVAILABILITY

- ☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

123

15. Price

\$13.00

