LBL-20677
UC-32

c.2

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

RECEIVED
LAWRENCE
BERKELEY LABORATORY

JUN 1 8 1986

LIBRARY AND
DOCUMENTS SECTION

Materials & Molecular Research Division

DATA ACQUISITION AND REDUCTION SOFTWARE
FOR A SELF-NULLING ELLIPSOMETER

S.T. Mayer and R.H. Muller

March 1986

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks.

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

LBL-20677 c.2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-20677

DATA ACQUISITION AND REDUCTION SOFTWARE FOR A SELF-NULLING ELLIPSOMETER

S. T. Mayer and R. H. Muller

Lawrence Berkeley Laboratory
Materials and Molecular Research Division
and
Department of Chemical Engineering
University of California, Berkeley
Berkeley, California 94720

March, 1986

This work was supported by the Assistant Secretary of Conservation and Renewable Energy, Office of Energy Systems Research, Energy Storage Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098

LBL-20677

DATA ACQUISITION AND REDUCTION SOFTWARE FOR A SELF-NULLING ELLIPSOMETER

S. T. Mayer and R. H. Muller

Lawrence Berkeley Laboratory
Materials and Molecular Research Division
and
Department of Chemical Engineering
University of California, Berkeley
Berkeley, California 94720

ABSTRACT

An automatic, magneto-optic, spectroscopic ellipsometer interfaced with an LSI-11 series computer has been reported previously (1). Two software changes and one instrumental change have been introduced which enable the ellipsometer to acquire time dependent spectra. The storage of multiple, time resolved spectra has given the ellipsometer the ability to study time dependent phenomena. Computer adjustment of the mechanical polarizer and analyzer azimuths (when the dynamic range of the Faraday cell is approached) has yielded a nearly unlimited range for this instrument. Introduction of Faraday cores with greater transmittance (possible because of the use of computer adjusted azimuths) has significantly increased the signal to noise ratio of the instrument.

Due to a large increase in the amount of data to be processed, new FORTRAN software for the reduction of these data was developed. Both the transformation of the raw data to

time dependent optical properties of the surface, and the reduction of the time dependent optical data by modeling to obtain physical properties of a surface, are presented. Software for the plotting of the kinetic spectra in three dimensions is also given.

TABLE OF CONTENTS

PART 1 --PROGRAM DESCRIPTIONS--

I. CONVER 1
II. CURDIS 2
III. DATFIL 3
IIII. DERIV 3
V. DYNSPE 4
VI. INTREF 6
VII. MDONOP 6
VIII. MDQOUT 9
XI. NUCNUM 9
X. OPTCOM 10
XI. PLTCUR 10
XII. PLTDAT 10
XIII. SMTH 11
XIV. STDDEV 11
XV. STEMOD 11
XVI. TIMER 11

PART 2 --PROGRAM LISTINGS, DEMONSTRATION AND EQUATIONS--

I. CONVER 14
II. CURDIS 20
III. DATFIL 24
IIII. DERIV 30
V. DYNSPE 35
VI. INTREF 44
VII. MDONOP 47

VIII. MDQOUT 60
XI. NUCNUM 69
X. OPTCOM 73
XI. PLTCUR 78
XII. PLTDAT 84
XIII. SMTH 92
XIV. STDDEV 96
XV. STEMOD 101
XVI. TIMER 106

PART 3 --SUBROUTINE LISTINGS, DEMONSTRATION AND EQUATIONS--

I. CSM2 112
II. DELAY 115
III. DELPSI 116
IIII. EDATA 121
IV. EMA 125
V. FCREAD 128
VI. FILM2 129
VII. FINISH 133
VIII. IN 134
IX. INPUT 136
X. NEWSET 138
XI. OUT 140
XII. OUTPUT 143
XIII. PLAC 145
XIV. PLAC2 147

XV. PSIDEL 149

XVI. P1INFO 150

XVII. P1TIME 151

XVIII. RATE 153

XIX. READ 154

XX. REFIND 155

XXI. RESET 157

XXII. STEPER 160

XXIII. T1INFO 161

XXIV. TITIME 162

XXV. WLCALC 164

--BIBLIOGRAPHY--

PROGRAM DESCRIPTIONS

I. CONVER

This program is a modification of the program "DSCONV" developed earlier by Farmer [1]. CONVER recalls data files (containing digitalized current levels as a function of encoder count) stored by DYNSPE and converts the contents to delta and psi values. Therefore, files created by the data acquisition routine DYNSPE are required for the execution of this program. These files can be identified by noting that the first symbol in the names is the number 0. As with the program DSCONV, the program requires the user to pick the type of conversion formula used. The "rotated azimuth formula option" (vertical plane coincides with 0 degrees) should be selected by the user. Each file which makes up the experimental file class is called, and raw data is converted by the program to delta and psi values. The conventions established by Muller [2] have been used. To convert the digitalized current levels (ADCU) to polarizer and analyzer azimuths (delta and psi), proportionality constants (effective Verdet coefficents) must be known. These constants are independent of the current in the solenoid but are a function of wavelength. Data which relate the specific rotation of the plane of polarization as a function of wavelength are stored in the file "FARADY.DAT", which is created by the routine CALFC1 developed by Farmer [1]. The basic operating equations are

$$P = P_o + K_p(j)I_p(j) \qquad 1$$

$$A = A_o + K_a(j)I_a(j). \qquad 2$$

In the above equations P_o and A_o are the manual polarizer and analyzer settings, j is the encoder position, $I_p(j)$ and $I_a(j)$ are the digitalized current levels of the polarizer and analyzer Faraday cells, and $K_p(j)$ and $K_a(j)$ are the Faraday cell rotation constants at the wavelength which corresponds to the encoder position j. During the execution of this program a new data file is created for each file of the file class. These new files contain the converted data (delta and psi values) corresponding to the digitalized current levels of the unconverted files. A file class is a group of individual files which contain data taken sequentially during the experiment. Like the unconverted data files, all files in a converted data file class have the second through the fifth symbol in their file name the same (see reference [3], and the example of the program's execution). The converted data files created by this program all contain the letter "C" (for converted) as the first symbol in their names (eg. CGOOD0.002 is the second convert data file of the experiment "GOOD").

II. CURDIS

Program CURDIS is a program which finds the solution to Laplace's equation (primary current distribution) of the elec-

trochemical cell discussed in reference [3]. This calculation employs a simple relaxation technique to find the solution to this equation. The boundary conditions applied were

1. Constant potential at the working-electrode surface.
2. Constant potential at the counter electrode surface.
3. No flux at the insulating boundaries (image points were used).

III. DATFIL

DATFIL creates formatted spectral data files from unformatted data for transfer to the L.B.L. DEC 8600 computers (via the data transfer routine KERMIT). Converted (but unformatted) data files created by CONVER are called up and are placed in a new data file suitable for data retrieval on the DEC 8600. The formatted data have been used in conjunction with the program PLTDAT to create three dimensional plots of the spectral data.

IIII. DERIV

By taking the difference in the value of delta and psi between two sequential spectral scans, a numerical estimate of the derivative of delta and psi with time is determined. The operating equations are

$$\frac{\partial \Delta}{\partial t} \cong \frac{\Delta(I,t_2) - \Delta(I,t_1)}{t_2 - t_1} \qquad 3$$

$$\frac{\partial \Psi}{\partial t} \cong \frac{\Psi(I,t_2) - \Psi(I,t_1)}{t_2 - t_1} \qquad 4$$

Converted spectral data generated by program CONVER are therefore required as input for the execution of this program. The spectral data files created by DERIV all contain the letter "D" (for derivative) as the first symbol in their names. The file class name of the output data files is unchanged (eg. DGOOD0.002 is the second computed derivitive file of experiment "GOOD").

V. DYNSPE

Program DYNSPE is a modification of the program SEV002 developed by Farmer [1]. DYNSPE allows multifile time resolved ellipsometric spectra to be obtained, in addition to computer feedback controlled azimuth settings. The use of computer feedback to establish "manual" azimuth settings of the ellipsometer has greatly increased both the performance and range of the instrument. Data are stored as a series of files, each corresponding to a single scan or signal averaged series of scans. In each file the time at which the spectral scan was started and finished is stored. This allows the reproduction of the average value of time associated with each data file. A value of the digitalized current level, I(j) of the channel associated with the analyzer or polarizer solenoids at each value of the encoder position, is stored. The present manual setting of the Glan Thompson polarizer and analyzer is also stored for each file (P_o, A_o). Finally, information pertaining to

the present instrument settings is also stored for each file.

The names of the data files created by DYNSPE all start with the number "0", and this symbol can be used to recognize the contents of the data file as being unconverted. The next four symbols in the file name are the name of the experiment (the "file class" name). The last four symbols are the number of the file for a given experiment ("file number"). The file number corresponds to the sequential time when the file was obtained. For example, file 0GOOD0.002 is the second unconverted data file of the experiment named "GOOD".

A value of the maximum A to D current levels (ADCU) of the analyzer and polarizer of the previous scan is determined in the data aquisition subroutine EDATA. When the current level in either the polarizer or analyzer approaches its maximum value (indicating that the rotation of the plane of polarization in the Faraday cores is approaching its maximum value), the computer program invokes a modification of the subroutine MOVE written by Armstrong [4] (called STEPER), to change the "manual" settings of the analyzer or polarizer. Subroutine STEPER sends a signal to the stepping motors of the analyzer or polarizer, making the encoder rotate the Glan-Thompson prism a set amount in the direction necessary to reduce the current levels in the Faraday core. The required direction of rotation is determined in the main data acquisition routine, while the amount of rotation of the manual azimuth setting is required as input information upon execution of the main routine.

VI. INTREF

INTREF calculates the phase change associated with the total internal reflection of light from a three reflection Fresnel Rhomb compensator. Only the refractive index of the glass is required as input data. The value of the phase change for various angles of incidence is placed in a data file named "INTREF.DAT". The angle used for the calculations is the angle between the first inclined interface and the optic axis. This calculated specular response of the phase change can be used as a guide for the design of a Fresnel Rhomb compensator, or to determine the error associated with the manufacture of such a device.

VII. MDONOP

Program MDONOP is a simplex data optimization routine for spectroscopic ellipsometric data. The simplex algorithm of minimizing a mathematical function of several variables is based on sampling the response surface of an error function in multidimensional space. The error function used to optimize the experimentally determined spectra with respect to a mathematical model spectrum has been discussed previously [3]. Simplex points are the error values associated with different parameter combinations. The algorithm of the simplex optimization procedure is as follows. After initial guesses of the best fit model parameters are initially made, the simplex points

with the highest and lowest value of the error function are determined. Next, a central average value for each model parameter is determined. This point is referred to as the "centrex". A new point (known as the "reflected" point) is calculated at a point which lies on a line in multidimensional space from the worst simplex point, through a known (calculated) distance to the centrex, and "reflecting" it the same distance to a new point. The error value associated with the new "reflected" point is then determined. If the error value of the reflected point is between the values of the worst and best simplex verticies, the worst simplex point is replaced by the "reflected" point, and the whole process is then repeated. When the "reflected" error value is better than the previously best simplex point, a new "double reflected" point is determined along the line between the worst simplex vertex and the centrex vertex. If this new value is superior to the "reflected" vertex, it is used to replace the worst simplex vertex. Otherwise, the worst simplex vertex is simply replaced by the reflected point. On occasion the error of the reflected point is worse than the worst simplex vertex. When this happens, a "contracted point" is determined by moving the worst simplex vertex towards the centrex point. If the new contracted point is still not superior to the worst simplex value, all the simplex vertices are contracted toward the best simplex value. This is referred to as a "simplex contraction".

By repeated application of the algorithm described above,

the program minimizes the difference between the model predicted and observed ellipsometric spectra. MD0NOP is a simplex optimization which minimizes difference between optical model 0 ("MD0") predictions (see ref. [3]) and the observed spectra. No CRT output of the intermediate values of the simplex are displayed on the screen ("NOP"). Other simplex routines which also do not display intermediate results on the CRT also have their last three letters of their names ending in "NOP" (eg. MD9NOP is a simplex routine which is the same as MD0NOP, except that optimization is made with respect to model 9 calculations).

These simplex routines require the operator to furnish initial guesses of the model parameters. Values of the optical constants of the film(s), encoder cutoff points, number of iterations, file numbers of the first and last sequential file to be optimized, and the name of the file class which is to be optimized are required as input data in addition to the initial guesses for model parameters. After each file number is optimized for a given model, initial guesses of the next (sequential) file number are made inside the program based on the best fit values of the previous spectral data file.

The optimization routine allows one to optimize around every point in the spectrum, or to use a smaller number of evenly spaced points in order to increase the speed of the calculation. Lastly, the file by file optimized model parameters are placed in a new file at the completion of the execution of this routine, and a name for this output file is required prior to the program

execution.

VIII. MDQOUT

MDQOUT is a program used to create a data file of the model predicted ellipsometric spectra corresponding to model Q (see ref [3]). The calculated spectra are placed in an unformatted data file (whose name is chosen by the operator) similar to those which contain raw and converted data. The calculated spectra generated by this routine can be called for plotting or CRT display using software described previously [1]. A number of other programs which create ellipsometric spectra corresponding to other models presented in reference [3] are also available. These can be recognized by the last three letters in the program name being "OUT" (eg. MD0OUT, model 0 output).

IX. NUCNUM

Program NUCNUM is used for the calculation of the number density of surface nuclei predicted by model 9 calculations. NUCNUM opens a reduced data file generated by program MD9NOP and calculates the number density of surface nuclei corresponding to the optically optimized parameters. The operative equation is

$$N_c = \frac{\Omega \times 10^{16}}{\pi r^2} \qquad 5$$

where is the surface coverage of crystals, and is the thickness of the secondary crystals (see ref. [3]). A new data file is created which contains both the values of the MD9NOP opt-

imized optical parameters and the nuclei number density results.

X. OPTCOM

Software for collecting four zone measurements on the ellipsometer has been presented before [4]. OPTCOM reads the values of the four zone measurements into the computer, and determines the standard deviation between the four zone measurements and the four zone average measurements. This program has been used to minimize four zone errors associated with the incorrect azimuth position of the Fresel Rhomb compensator as discussed in reference [3].

XI. PLTCUR

Program PLTCUR is a routine for making 3-D plots of current distributions. Data for the current distribution is obtained from the results of program CURDIS. Many of the subroutines called in this program are from ISSCO product plotting routines DISSPLA.

XII. PLTDAT

PLTDAT is a 3-D plotting routine for spectral ellipsometry data. The data which are plotted must be converted and formatted (See program DATFIL). By linking the PLTDAT source code with the appropriate LBL emulation subroutine found on the VMS 8600 computers, PLTDAT can be used to create 3-D plots on a number of different plotting devices.

XIII. SMTH

Digital averaging can be used to reduce the random noise and increase the accuracy of the observed ellipsometric spectra. In order to minimize systematic errors, multiple scans of a clean surface can be taken at the same scan rate as the experiment. These scans can be averaged using program SMTH.

XIV. STDDEV

Program STDDEV computes the standard deviation of a number of converted (delta and psi) spectral data files. This information can be used to estimate the level of uncertainty associated with the measurements at various scan rates. The results of the computations versus wavelength are placed in a data file.

XV. STEMOD

Program STEMOD is used to calculate the current/time response to a potential step as given by the dissolution/percipitation model presented in the thesis which accompanies this report [ ]. The calculated theoretical current transients are placed in a data file, which can later be called for plotting.

XVI. TIMER

Program TIMER calculates the average time at which each converted spectral data file was taken with respect to the start of an experiment. This allows one to fix values of time to each

spectrum in time dependent spectroscopic studies.

PROGRAM LISTINGS AND DEMONSTRATIONS

DEMONSTRATION OF THE EXECUTION OF PROGRAM CONVER

```
TSX>DIR LD0:OCARA*.*
29-Jan-86
OCARA0.004      5   12-Jun-85      OCARA0.005      5   12-Jun-85
OCARA0.007      5   12-Jun-85      OCARA0.008      5   12-Jun-85
OCARA0.009      5   12-Jun-85      OCARA0.001      5   12-Jun-85
OCARA0.006      5   12-Jun-85      OCARA0.010      5   12-Jun-85
OCARA0.011      5   12-Jun-85      OCARA0.012      5   12-Jun-85
OCARA0.013      5   12-Jun-85      OCARA0.014      5   12-Jun-85
OCARA0.002      5   12-Jun-85      OCARA0.015      5   12-Jun-85
OCARA0.016      5   12-Jun-85      OCARA0.017      5   12-Jun-85
OCARA0.018      5   12-Jun-85      OCARA0.019      5   12-Jun-85
OCARA0.020      5   12-Jun-85      OCARA0.003      5   12-Jun-85
OCARA0.021      5   12-Jun-85      OCARA0.022      5   12-Jun-85
OCARA0.023      5   12-Jun-85      OCARA0.024      5   12-Jun-85
OCARA0.025      5   12-Jun-85      OCARA0.026      5   12-Jun-85
OCARA0.027      5   12-Jun-85      OCARA0.028      5   12-Jun-85
OCARA0.029      5   12-Jun-85      OCARA0.030      5   12-Jun-85
OCARA0.031      5   12-Jun-85      OCARA0.032      5   12-Jun-85
 32 Files, 160 Blocks
 2642 Free blocks                  {DIRECTORY CONTAINS THE RAW
                                   {DATA OF THE FILE CLASS "CARA"

TSX>DIR LD0:WLCALC.DAT
 29-Jan-86
WLCALC.DAT      1   15-Nov-84     {DIRECTORY CONTAINS THE WAVE-
 1 Files, 1 Blocks                 {LENGTH CALIBRATION FILE
 2642 Free blocks

TSX>ASS LD0: DY1:                  {ASSIGN THE DEVICE WHERE THE DATA IS
                                   {     LOCATED TO DY1

TSX>RUN LD1:CONVER {RUN THE PROGRAM

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

   ELLIPSOMETER RAW DATA CONVERSION ROUTINE


 FOR USE WITH MULTIPLE FILES OF ONE FILE CLASS

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

 INPUT THE BASE FILE NAME (A4) CARA

 HOW MANY FILES ARE TO BE CONVERTED(I2)?32

 ARE STANDARD(1) OR ROTATED(2) AZIMUTH CONVERSION FORMULAS TO
 BE USED(I1)? 2

 #  1CONVERSION IN PROGRESS......
```

```
 #  2CONVERSION IN PROGRESS......
 #  3CONVERSION IN PROGRESS......
 #  4CONVERSION IN PROGRESS......
 #  5CONVERSION IN PROGRESS......
 #  6CONVERSION IN PROGRESS......
                    .
                    .
                    .
                    .
                    .
 # 30CONVERSION IN PROGRESS......
 # 31CONVERSION IN PROGRESS......
 # 32CONVERSION IN PROGRESS......
 DO YOU WANT TO CONVERT ANOTHER FILE CLASS(Y/N)? N
STOP --
```

PROGRAM CONVER VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
A		-Analyzer azymuth (deg.x100)
ANA(j)		-Digital current level of the analyzer at encoder count I (ACDU) -Psi (x100) at encoder count I (deg.)
ASLOPE(j)		-Verdat rotation constant at encoder count I of the analyzer (deg./ADCU)
C		-Compensator azimuth (deg.)
D		-Delta (deg.)
IMAX		-File number of the last data file to be converted
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NANA		-Manual setting of the analyzer (deg. x100)
NANA		-Manual setting of the compensator (deg. x100)
NPOL		-Manual setting of the polarizer (deg. x100)
NPASS		-File number loop counter
P		-Polarizer azimuth (deg. x100)
POL(j)		-Digital current level of the polarizer at encoder count j -Delta (x100) at encoder count I (deg.)
PSLOPE(j)		-Verdat rotation constant at encoder count I of the polarizer (deg./ADCU)
S		-Psi (deg.)

SUBROUTINES CALLED

DELPSI
FCREAD
IN
OUT
WLCALC

EQUATIONS USED FOR CALCULATIONS

$$P = P_o + K_p(j) I_p(j)$$

$$A = A_o + K_a(j) I_a(j)$$

```
      PROGRAM CONVER
C*********************************************************************
C     OBJECTIVE: TO CONVERT RAW DATA FROM ELLIPSOMETER SPECTRAL SCAN
C                INTO PSI, DELTA, AND WAVELENGTH.
C
C     PROGRAM WRITTEN BY JOSEPH C. FARMER IN SPRING 1981 AT LBL-MMRD
C                        UPDATED FOR MULTIPLE FILE CONVERTION BY
C                        S. MAYER FALL 1984.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      INTEGER*2 ANGOPT
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      TYPE 100
      TYPE 101
      TYPE 102
      TYPE 100
      CALL WLCALC
C*****THE FARADAY CELL CALIBRATION DATA IS OBTAINED.******************
      CALL FCREAD
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1001
      ACCEPT 2000,NAME2
      TYPE 1002
      ACCEPT 1004,IMAX
C*****THE OPERATOR SELECTS THE AZIMUTH CONVERION FORMULAS TO BE USED.**
    2 TYPE 1013
      ACCEPT 1014, ANGOPT
      IF(ANGOPT.NE.1.AND.ANGOPT.NE.2)GOTO 2
      DO 4 NPASS=1,IMAX
      IPASS=NPASS
      NAME1='DY10'
```

```
      CALL IN
      TYPE 1015,NPASS
      DO 3 I=1,400
      P=NPOL/100.+POL(I)*PSLOPE(I)
      A=NANA/100.+ANA(I)*ASLOPE(I)
      C=NCMP/100.
      CALL DELPSI(P,C,A,D,S,ANGOPT)
      POL(I)=D*100.
      ANA(I)=S*100.
    3 CONTINUE
      NAME1='DY1C'
      CALL OUT
    4 CONTINUE
      TYPE 1016
      ACCEPT 1003,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 1
      IF (FLAG2.NE.'N') GOTO 4
  100 FORMAT (/,'****************************************')
  101 FORMAT (/,'    ELLIPSOMETER RAW DATA CONVERSION ROUTINE')
  102 FORMAT (//,2X,'FOR USE WITH MULTIPLE FILES OF ONE FILE CLASS')
 1001 FORMAT(/,'$ INPUT THE BASE FILE NAME (A4) ')
 1002 FORMAT(/,'$ HOW MANY FILES ARE TO BE CONVERTED(I2)?')
 1003 FORMAT(A1)
 1004 FORMAT(I2)
 1013 FORMAT(/,'$ ARE STANDARD(1) OR ROTATED(2) AZIMUTH
     C CONVERSION FORMULAS TO BE USED(I1)? ')
 1014 FORMAT(I1)
 1015 FORMAT(/,2X,'# ',I2, 'CONVERSION IN PROGRESS......')
 1016 FORMAT(/,'$ DO YOU WANT TO CONVERT ANOTHER FILE CLASS(Y/N)? ')
 2000 FORMAT(A4)
      STOP
      END
```

PROGRAM CURDIS VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
CIR(I,J)		-Local current at the x,y position i,j on the electrode surface
IEMAX		-Maximum array value in the x direction which corresponds to a point on the electrode's surface
IMAX		-Maximum array value in the x direction which corresponds to the containers x boundry.
ISCAL		-Scale used in the calculation of the current distribution.
ITER		-Present iteration number.
ITMAX		-Maximum number of iteration to be made.
JEMAX		-Maximum array value in the y direction which corresponds to a point on the electrode's surface
JMAX		-Maximum array value in the y direction which corresponds to the containers y boundry.
KMAX		-Maximum array value in the z direction which corresponds to the containers z boundry, and the counter electrode location.
POT(I,J,K)		-Present estimate of the potential at the 3 dimensional array point i,j,k.

SUBROUTINES CALLED

LOOP (specific for this routine).

```
      PROGRAM CURDIS
C*****************************************************
C     PROGRAM FOR CALCULATION OF CURRENT DISTRIBUTION
C     IN ELLIPSOMETER ELLECTROCHEMICAL CELL.
C     WRITTEN BY S. MAYER JULY 1985
C*****************************************************
      REAL POT(129,150,200),CUR(59,24)
      CHARACTER FILENM(12)
      COMMON/LOOP/ CAR(30),JEMAX
      TYPE *,'INITALIZATION PHASE 1'
      TYPE 80
      ACCEPT 110,ISCAL
      TYPE 81
      ACCEPT 110,ITMAX
      KMAX=20*ISCAL
      JMAX=15.*ISCAL
      IMAX=12.81*ISCAL
      IEMAX=5.94*ISCAL
      JEMAX=2.38*ISCAL
      DO 1 K=1,KMAX
      DO 2 J=1,JMAX
      DO 3 I=1,IMAX
      POT(I,J,K)=((KMAX-K+1)/KMAX)*10.
 3    CONTINUE
 2    CONTINUE
 1    CONTINUE
      TYPE *,'INITALIZATION PHASE 2'
      DO 5 I=1,IEMAX
      DO 6 J=1,JEMAX
      POT(I,J,2)=10.
      POT(I,J,1)=10.
 6    CONTINUE
 5    CONTINUE
      ITER=0
      TYPE *,'START COMPUTATION'
 25   CONTINUE
      ITER=ITER+1
      IPASS=ITER+1
      IF (IPASS.GE.KMAX) IPASS=KMAX
      TYPE *,'ITERATION #',ITER
C*******************************************************************
C     SYMETRIC NO FLUX BOUNDRY CONDICTION
C     AT BACK PLAN WHERE ELECTRODE ISN'T
C*******************************************************************
      DO 7 I=IEMAX+1,IMAX
      DO 8 J=JEMAX+1,JMAX
      POT(I,J,1)=POT(I,J,3)
 8    CONTINUE
 7    CONTINUE
```

```
      DO 9 K=1,IPASS
C*****************************************
C     SYMETRIC AXIS AROUND EDGE OF X=1
C     AND X=IMAX
C*****************************************
      DO 10 J=1,JMAX
      POT(1,J,K)=POT(3,J,K)
      POT(IMAX,J,K)=POT(IMAX-2,J,K)
 10   CONTINUE
C*****************************************
C     SYMETRIC AXES AROUND EDGE OF Y=1
C     AND Y=JMAX
C*****************************************
      DO 11 I=1,IMAX
      POT(I,1,K)=POT(I,3,K)
      POT(I,JMAX,K)=POT(I,JMAX-2,K)
 11   CONTINUE
 9    CONTINUE
      DO 12 K=3,IPASS
      DO 13 J=2,JMAX
      DO 14 I=2,IMAX
      POT(I,J,K)=(POT(I+1,J,K)+POT(I-1,J,K)+POT(I,J+1,K)+POT(I,J-1,K)+
     *POT(I,J,K+1)+POT(I,J,K-1))/6.
 14   CONTINUE
 13   CONTINUE
 12   CONTINUE
      IF(ITER.GT.ITMAX) GOTO 200
      DO 15 II=2,IEMAX
      DO 16 JJ=2,JEMAX
      CUR(II,JJ)=POT(II,JJ,3)-POT(II,JJ,2)
 16   CONTINUE
 15   CONTINUE
      GOTO 25
200   CONTINUE
      DO 21 II=2,IEMAX
      DO 22 JJ=2,JEMAX
      CUR(II,JJ)=POT(II,JJ,3)-POT(II,JJ,2)
 22   CONTINUE
 21   CONTINUE
      REWIND (1)
      OPEN(UNIT=1,FILE='CURDIS.DAT',TYPE='NEW',DISP='SAVE',FORM='
     *FORMATTED')
      DO 23 II=2,IEMAX
      TYPE 100,(CUR(II,J),J=2,JEMAX)
      DO 30 JJ=2,JEMAX
      CAR(JJ)=-CUR(II,JJ)
 30   CONTINUE
      CALL LOOP(II)
 23   CONTINUE
```

```
 80     FORMAT(/,'$WHAT IS THE SCALE YOU WANT(I2)?')
 81     FORMAT(/,'$HOW MANY ITERATIONS(I2)?')
 82     FORMAT(/,'$WHAT SHOULD THE OUTPUT FILE NAME BE(A12)? ',1X)
100     FORMAT(10(2X,F6.3))
110     FORMAT(I3)
105     FORMAT(1X,12A1)
150     FORMAT(/,2X,'ITER #',1X,I3,1X,'ACUR=',1X,F7.4,
     *'FACUR=',1X,F7.4)
        STOP
        END

        SUBROUTINE LOOP(J)
        COMMON/LOOP/ CAR(30),JEMAX
        WRITE (1,2) J,(CAR(L),L=2,9)
        WRITE (1,3) (CAR(L),L=10,17)
 2      FORMAT (I5,8(2X,F7.3))
 3      FORMAT (5X,8(2X,F7.3))
        RETURN
        END
```

DEMONSTRATION OF THE EXECUTION OF PROGRAM DATFIL

```
TSX>DIR LD0:CCARA*.*
 29-Jan-86
CCARA0.001      5  29-Jan-86      CCARA0.002      5  29-Jan-86
CCARA0.003      5  29-Jan-86      CCARA0.004      5  29-Jan-86
CCARA0.005      5  29-Jan-86      CCARA0.006      5  29-Jan-86
CCARA0.007      5  29-Jan-86      CCARA0.008      5  29-Jan-86
CCARA0.009      5  29-Jan-86      CCARA0.010      5  29-Jan-86
CCARA0.011      5  29-Jan-86      CCARA0.012      5  29-Jan-86
CCARA0.013      5  29-Jan-86      CCARA0.014      5  29-Jan-86
CCARA0.015      5  29-Jan-86      CCARA0.016      5  29-Jan-86
CCARA0.017      5  29-Jan-86      CCARA0.018      5  29-Jan-86
CCARA0.019      5  29-Jan-86      CCARA0.020      5  29-Jan-86
CCARA0.021      5  29-Jan-86      CCARA0.022      5  29-Jan-86
CCARA0.023      5  29-Jan-86      CCARA0.024      5  29-Jan-86
CCARA0.025      5  29-Jan-86      CCARA0.026      5  29-Jan-86
CCARA0.027      5  29-Jan-86      CCARA0.028      5  29-Jan-86
CCARA0.029      5  29-Jan-86      CCARA0.030      5  29-Jan-86
CCARA0.031      5  29-Jan-86      CCARA0.032      5  29-Jan-86
 32 Files, 160 Blocks
 2632 Free blocks
```

{AFTER THE EXECUTION OF "CONVER" CONVERTED DATA WILL BE LOCATED ON THE DISC

```
TSX>ASS LD0: DY1:
```

{ASSIGN THE DEVICE WHERE THE DATA IS LOCATED TO DY1

```
TSX>RUN LD1:DATFIL
```

{RUN THE PROGRAM

```
 WHAT IS THE BASE 4 LETTER FILE NAME (A4)  CARA

 HOW MANY FILES ARE IN THE FILE CLASS(I2)?32

 STEP SIZE FOR CONVERSION(I2)1

 #  1CONVERSION IN PROGRESS......

 #  2CONVERSION IN PROGRESS......

 #  3CONVERSION IN PROGRESS......

 #  4CONVERSION IN PROGRESS......

                    .
                    .
                    .
                    .
                    .

 # 31CONVERSION IN PROGRESS......

 # 32CONVERSION IN PROGRESS......
```

TRANSFORM ANOTHER DATA FILE CLASS(Y/N)?N

STOP --

TSX>ED LD0:CARA00.1 {OBSERVE ONE OF THE OUTPUT DATA FILES

100	90.500	90.140	90.090	90.160	89.580
105	89.270	89.070	88.890	88.910	88.590
110	88.430	88.230	88.050	87.600	87.380
115	87.450	86.960	86.700	86.420	86.410
120	86.060	85.690	85.650	85.250	84.840
125	84.600	84.490	84.240	84.090	83.730
130	83.540	83.330	82.980	82.790	82.510
135	82.430	82.070	81.660	81.540	81.160
140	81.020	80.770	80.390	80.130	79.960
145	79.870	79.590	79.270	78.890	78.720
150	78.410	78.140	77.780	77.710	77.420
155	77.130	76.730	76.560	76.260	76.030
160	75.700	75.470	75.110	74.900	74.510
165	74.340	74.110	73.740	73.510	73.130
170	72.860	72.560	72.400	72.110	71.770
175	71.390	71.200	70.870	70.680	70.300
180	69.800	69.660	69.380	69.030	68.740
185	68.510	68.030	67.880	67.420	67.150
190	66.870	66.500	66.180	65.810	65.580
195	65.190	64.810	64.590	64.250	63.860
200	63.440	63.080	62.760	62.620	62.090
205	61.700	61.500	61.080	60.780	60.430
210	60.030	59.700	59.420	58.980	58.670
215	58.200	57.870	57.610	57.150	56.840
220	56.540	56.070	55.720	55.320	55.060
225	54.660	54.270	53.980	53.560	53.180
230	52.770	52.410	51.940	51.510	51.290
235	50.860	50.230	49.800	49.550	49.100
240	48.460	48.360	48.750	47.880	46.960
245	46.460	46.110	45.740	45.250	44.700
250	44.130	43.820	43.330	42.780	42.320
255	42.020	41.590	40.970	40.730	40.060
260	39.660	38.980	38.890	38.370	37.590
265	37.440	36.900	36.240	35.990	35.470
100	45.060	45.010	44.890	44.820	44.720
105	44.850	44.830	44.830	44.820	44.730
110	44.830	44.810	44.730	44.770	44.870
115	44.790	44.710	44.760	44.720	44.790
120	44.810	44.760	44.780	44.770	44.680
125	44.740	44.690	44.710	44.720	44.720
130	44.650	44.720	44.670	44.640	44.650
135	44.650	44.720	44.730	44.660	44.720

140	44.710	44.690	44.640	44.670	44.660
145	44.690	44.670	44.580	44.670	44.640
150	44.670	44.690	44.640	44.680	44.610
155	44.610	44.670	44.650	44.630	44.620
160	44.600	44.660	44.630	44.620	44.630
165	44.640	44.640	44.600	44.640	44.610
170	44.640	44.640	44.620	44.600	44.600
175	44.640	44.610	44.610	44.610	44.590
180	44.590	44.600	44.580	44.550	44.550
185	44.560	44.560	44.560	44.560	44.580
190	44.590	44.560	44.590	44.560	44.550
195	44.560	44.610	44.540	44.570	44.570
200	44.530	44.560	44.570	44.580	44.550
205	44.560	44.580	44.520	44.550	44.520
210	44.540	44.530	44.550	44.540	44.540
215	44.510	44.550	44.520	44.540	44.540
220	44.540	44.540	44.550	44.520	44.520
225	44.530	44.520	44.550	44.530	44.540
230	44.530	44.520	44.510	44.520	44.510
235	44.520	44.540	44.530	44.500	44.570
240	44.490	44.570	44.540	44.530	44.560
245	44.530	44.550	44.570	44.520	44.560
250	44.550	44.560	44.550	44.590	44.580
255	44.580	44.540	44.600	44.590	44.590
260	44.590	44.580	44.610	44.610	44.630
265	44.620	44.630	44.670	44.650	44.660

DELTA OR PSI VALUE

ENDCODER COUNT

PROGRAM DATFIL VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
ANA(I)		-Psi (x100) at encoder count I (deg.)
IMAX		-File number of the last data file to be converted
ISTP		-File step size of the conversion
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NANA		-Manual setting of the analyzer (deg. x100)
NCMP		-Manual setting of the compensator (deg. x100)
NPOL		-Manual setting of the polarizer (deg. x100)
NPASS		-File number loop counter
POL(I)		-Delta (x100) at encoder count I (deg.)

SUBROUTINES CALLED

IN
PLAC2
WLCALC

```
      PROGRAM DATFIL
C*********************************************************************
C     OBJECTIVE: TO TAKE UNFORMATED SPECTRAL DATA FILES AND CREATE A
C                FORMATED DATA FILE FOR TRANSFER TO THE L.B.L. MAIN
C                FRAME COMPUTERS.
C
C     PROGRAM WRITTEN S. MAYER SPRING 1985.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAM(20),NAME1,NAME2,NAME3
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      CALL WLCALC
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1001
      ACCEPT 2000,NAME2
      TYPE 1002
      ACCEPT 1004,IMAX
      TYPE 1003
      ACCEPT 1004,ISTP
      DO 8000 NPASS=1,IMAX,ISTP
      IPASS=NPASS
      TYPE 1015,IPASS
      NAME1='DY1C'
      CALL IN
      NAME(1)='DY1:'
      NAME(2)=NAME2
      CALL PLAC2(IPASS,NAME3)
      NAME(3)=NAME3
      OPEN (UNIT=2,NAME=NAME,TYPE='NEW',DISP='SAVE')
      DO 7000,N=100,265,5
      WRITE (2,10) N,POL(N)/100.,POL(N+1)/100.,POL(N+2)/100.
     *,POL(N+3)/100.,POL(N+4)/100.
```

```
7000  CONTINUE
      DO 7500 N=100,265,5
      WRITE (2,10) N,ANA(N)/100.,ANA(N+1)/100.,ANA(N+2)/100.
     *,ANA(N+3)/100.,ANA(N+4)/100.
7500  CONTINUE
      CLOSE (UNIT=2)
8000  CONTINUE
    6 TYPE 1018
      ACCEPT 100,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 1
      IF (FLAG2.NE.'N') GOTO 6
   10 FORMAT(I5,5(2X,F7.3))
  100 FORMAT(A1)
 1001 FORMAT(/,'$ WHAT IS THE BASE 4 LETTER FILE NAME (A4)   ')
 1002 FORMAT(/,'$ HOW MANY FILES ARE IN THE FILE CLASS(I2)?')
 1003 FORMAT(/,'$ STEP SIZE FOR CONVERSION(I2)')
 1004 FORMAT(I2)
 1015 FORMAT(/,2X,'# ',I2, 'CONVERSION IN PROGRESS......')
 1018 FORMAT(/,'$CONVERT ANOTHER FILE CLASS(Y/N)?')
 2000 FORMAT(A4)
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF PROGRAM DERIV

```
TSX>ASS LD0: DY1:          {ASSIGN THE DEVICE WHERE
                            {   THE DATA IS LOCATED TO DY1

TSX>RUN LD1:DERIV          {RUN THE PROGRAM

 WHAT IS THE BASE 4 LETTER FILE NAME OF THE EXPERIMENT(A4) CARA

 HOW MANY FILES WERE IN THE EXPERIMENT(I2)?32

CALCULATIONS FOR FILE #  2 COMPLETED

CALCULATIONS FOR FILE #  3 COMPLETED

CALCULATIONS FOR FILE #  4 COMPLETED
                  .
                  .
                  .
                  .

CALCULATIONS FOR FILE # 30 COMPLETED

CALCULATIONS FOR FILE # 31 COMPLETED

CALCULATIONS FOR FILE # 32 COMPLETED

STOP --

TSX>DIR LD0:DCARA*.*                {DIRECTORY OF THE DISC
                                     {   AFTER THE EXECUTION OF
                                     {   OF THE PROGRAM

29-Jan-86
DCARA0.002     5  29-Jan-86     DCARA0.003     5  29-Jan-86
DCARA0.004     5  29-Jan-86     DCARA0.005     5  29-Jan-86
DCARA0.006     5  29-Jan-86     DCARA0.007     5  29-Jan-86
DCARA0.008     5  29-Jan-86     DCARA0.009     5  29-Jan-86
DCARA0.010     5  29-Jan-86     DCARA0.011     5  29-Jan-86
DCARA0.012     5  29-Jan-86     DCARA0.013     5  29-Jan-86
DCARA0.014     5  29-Jan-86     DCARA0.015     5  29-Jan-86
DCARA0.016     5  29-Jan-86     DCARA0.017     5  29-Jan-86
DCARA0.018     5  29-Jan-86     DCARA0.019     5  29-Jan-86
DCARA0.020     5  29-Jan-86     DCARA0.021     5  29-Jan-86
DCARA0.022     5  29-Jan-86     DCARA0.023     5  29-Jan-86
DCARA0.024     5  29-Jan-86     DCARA0.025     5  29-Jan-86
DCARA0.026     5  29-Jan-86     DCARA0.027     5  29-Jan-86
DCARA0.028     5  29-Jan-86     DCARA0.029     5  29-Jan-86
DCARA0.030     5  29-Jan-86     DCARA0.031     5  29-Jan-86
DCARA0.032     5  29-Jan-86
 31 Files, 155 Blocks
 2228 Free blocks
```

PROGRAM DERIV VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
ADIR(I)	$\frac{\partial \Psi}{\partial t}$	-Calculated derivative of psi at the encoder count I
ANA(I)	$\Psi(I, t_1)$	-Psi (x100) at encoder count I (deg.)
AOLD(I)	$\Psi(I, t_2)$	-Psi (x100) of the previous scan at encoder count I (deg.)
DT	$t_2 - t_1$	-Calculated elapsed time between two sequential spectral scans
IHRS0		-Hours passed since the computer time clock was reset
IMAX		-File number of the last data file to be converted
IMIN0		-Minutes passed since the computer time clock was reset
ISEC0		-Seconds passed since the computer time clock was reset
ITIC0		-Tics (1/60 sec) since the computer time clock was reset
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC2")
NPASS		-File number loop counter
PDIR(I)	$\frac{\partial \Delta}{\partial t}$	-Calculated derivative of delta
POL(I)	$\Delta(I, t_1)$	-Delta (x100) at encoder count I (deg.)
POLD(I)	$\Delta(I, t_2)$	-Delta (x100) of the previous scan at encoder count I (deg.)
TNEW	t_1	-The average time that the present spectra was taken (sec), relative to the start of the experiment

TOLD	t_2	-The average time that the previous spectra was taken (sec), relative to the start of the experiment

SUBROUTINES CALLED

IN
PLAC2
WLCALC

EQUATIONS USED FOR CALCULATIONS

$$\frac{\partial \Delta}{\partial t} \cong \frac{\Delta(I, t_2) - \Delta(I, t_1)}{t_2 - t_1}$$

$$\frac{\partial \Psi}{\partial t} \cong \frac{\Psi(I, t_2) - \Psi(I, t_1)}{t_2 - t_1}$$

```
      PROGRAM DERIV
C*********************************************************************
C     OBJECTIVE: TO COMPUTE A NUMERICAL ESTIMATE OF THE TIME DERIVATE
C                OF  PSI AND DELTA VERSUS WAVELENGTH FOR AN EXPERIMENT
C                MADE BY "DYNSPE"
C
C     PROGRAM WRITTEN S. MAYER FALL 1985.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      REAL POLD(400),AOLD(400),PDIR(400),ADIR(400)
      INTEGER*2 ANGOPT
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      CALL WLCALC
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1000
      ACCEPT 2000,NAME2
      TYPE 1001
      ACCEPT 2001,IMAX
      DO 3 IWV=1,399
      POLD(IWV)=0
      AOLD(IWV)=0
    3 CONTINUE
      IPASS=1
      NAME1='DY1C'
      CALL IN
      DO 4 IWV=1,399
      POLD(IWV)=POL(IWV)
      AOLD(IWV)=ANA(IWV)
    4 CONTINUE
      TOLD=3600.*IHRSO+60*IMINO+ISECO+ITICO/60.
      DO 9 NPASS=2,IMAX
```

```
      IPASS=NPASS
      CALL IN
      TNEW=3600.*IHRSO+60*IMINO+ISECO+ITICO/60.
      DT=TNEW-TOLD
      DO 6 IWV=1,399
      PDIR(IWV)=-(POLD(IWV)-POL(IWV))/DT
      ADIR(IWV)=-(AOLD(IWV)-ANA(IWV))/DT
    6 CONTINUE
      DO 7 IWV=1,399
      POLD(IWV)=POL(IWV)
      AOLD(IWV)=ANA(IWV)
    7 CONTINUE
      DO 8 IWV=1,399
      POL(IWV)=PDIR(IWV)
      ANA(IWV)=ADIR(IWV)
    8 CONTINUE
      TOLD=TNEW
      TYPE 1002,NPASS
      NAME1='DY1D'
      CALL OUT
    9 CONTINUE
 1000 FORMAT(/,'$ WHAT IS THE BASE 4 LETTER FILE NAME OF THE
     *EXPERIMENT(A4)   ')
 1001 FORMAT(/,'$ HOW MANY FILES WERE IN THE EXPERIMENT(I2)?')
 1002 FORMAT(/,' CALCULATIONS FOR FILE # ',I2,' COMPLETED')
 2000 FORMAT(A4)
 2001 FORMAT(I2)
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM DYNSPE

```
.RUN DY0:DYNSPE

DO YOU WANT TO REVIEW ENCODER WAVELENGTH CALIBRATION PARAMETERS(Y/N)? Y

        DIGITAL ENCODER PRESETS:

        CALIBRATION CURVE SEGMENT NUMBER 1:
             SLOPE =     18.0210
             INTERCEPT =     5921.19
             CORRELATION COEFFICIENT =      0.9996

        CALIBRATION CURVE SEGMENT NUMBER 2:
             SLOPE =    -18.1178
             INTERCEPT =     8810.10
             CORRELATION COEFFICIENT =      0.9999

        LIMITS OF APPLICATION:
             UPPER =    281
             LOWER =     83

DO YOU WANT TO CHANGE THESE(Y/N)? N

INSERT DATA STORAGE DISK, THEN "RETURN".

DO YOU WANT TO ADJUST THE SCAN RATE(Y/N)? N

STEADY SCAN RATE ESTABLISHED.  SCAN RATE =   786

IS SETUP INFO TO BE (A)ENTERED FROM KEYBOARD OR (B)READ FROM OLD DATA FILE? B

NAME OF SPECTRA FILE TO BE RETRIEVED = DY10BDYS0001

NAME(A40) = STEVEN T. MAYER

DATE(I2-A3-I2/DAY-MONTH-YEAR) = 28-JAN-86

LAMP OPERATIONAL LIFE(I5/HOURS) = 850

EXPERIMENT IDENTIFICATION(A40) = DEMONSTRATION OF DYNSPE

HAVE MANUAL AZIMUTHS BEEN CHANGED(Y/N)? Y

MANUAL AZIMUTH SETTINGS(DEG):
 POLARIZER(F7.2)    = 80.
 ANALYZER(F7.2)     = 135.
 COMPENSATOR(F7.2) = 45.
 MONOCHROMATOR SETTING AT NULL(F7.2) =
```

```
ANALOG/DIGITAL CONVERTOR PRESET PARAMETERS:
 POL. INPUT CH. = 0
 ANA. INPUT CH. = 1
 ADC GAIN       = 3

 DO YOU WANT TO CHANGE THESE(Y/N)? N

HOW MANY SPECTRAL SCANS ARE TO BE AVERAGED PER FILE(I5)?1
 FOUR LETTERS IN FILE NAME (EX. ABCD)= TEST
 TIME DURATION FOR EXPERIMENT IN MINUTES (A2)= 2

 HOW MANY DEG. SHOULD THE STEP MOTER MOVE WHEN THE "I" GT. 1000 ADCU"S(I2)?1

 IS SETUP INFORMATION TO BE OUTPUT (A)TO CRT, (B)TO PRINTER, OR (C)NOT AT ALL? A

        28-JAN-86
        STEVEN T. MAYER

        DEMONSTRATION OF DYNSPE

        MANUAL AZIMUTHS(DEG):

          P =   80.00
          A =  135.00
          C =   45.00
          WL=    0.00

        LAMP:

          TYPE        = KX
          SERIAL NO. = 75-2002
          LAMP VOLTAGE(VOLTS) =   15
          LAMP CURRENT(AMPS)  =   6
          LAMP OPERATIONAL LIFE(HOURS) =   850

        PHOTOMULTIPLIER:

          MODEL = HAMAMATSU
          DYNODE VOLTAGE(VOLTS) =  1100

        FARADAY CELL CONTROLLER:

          POLARIZER

          PHASE              =  700
          GAIN               =  999
          AMPLITUDE          =  800
          TIME CONSTANT(MS) =   10

          ANALYZER
```

```
          PHASE              =  968
          GAIN               =  999
          AMPLITUDE          =  800
          TIME CONSTANT(MS)  =   10

        GALVANOMETER AMPLIFIER:

          POLARIZER

          GAIN               =     10
          TIME CONSTANT(MS)  =      1

          ANALYZER

          GAIN               =     10
          TIME CONSTANT(MS)  =      1

        ANALOG/DIGITAL CONVERTOR:

          POL. INPUT CH. = 0
          ANA. INPUT CH. = 1
          ADC GAIN       = 3

ENTER "G" AND "RETURN" FOR SPECTRA COLLECTION. G

SPECTRA BEING COLLECTED...........

EACH POINT AN AVERAGE OF:   11. CHANNEL SCANS

        FILE # 1

        DATA COLLECTION INITIATED AT:  0: 5:40:37

        DATA COLLECTION FINISHED AT:  0: 5:48:46
    THE MAXIMUM POLARIZER CURRENT=   -11.00000     AT AN ENCODER COUNT=      14
    THE MAXIMUM ANALYZER CURRENT=   -9.300000     AT AN ENCODER COUNT=     337

EACH POINT AN AVERAGE OF:   11. CHANNEL SCANS

        FILE # 2

        DATA COLLECTION INITIATED AT:  0: 5:58:22

        DATA COLLECTION FINISHED AT:  0: 6: 3:46
    THE MAXIMUM POLARIZER CURRENT=   -10.90909     AT AN ENCODER COUNT=     372
    THE MAXIMUM ANALYZER CURRENT=   -9.400000     AT AN ENCODER COUNT=     115

EACH POINT AN AVERAGE OF:   11. CHANNEL SCANS
```

PROGRAM DYNSPE VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
ANA(I)		-Digital current level of the analyzer at encoder count I (ACDU)
DEVICE		-Device name send to subroutine "STEPER"
DURAT		-Assigned duration of the experiment (min)
IHRSO		-Hours associated with the start of the spectral scan (hrs)
IISTEP		-Number of degrees that the step motor moves the polarizer or analyzer when signal saturation is approached
IMAX		-File number of the last data file to be converted
IMINO		-Minutes associated with the start of the spectral scan (min)
ISECO		-Seconds associated with the start of the spectral scan (sec)
ITICO		-Tics (1/60 sec) associated with the start of the spectral scan (tics)
JTIME		-Maximum clock reading (reading at the end of the experiment) (min)
KHRS		-Clock setting (Hrs) at the beginning of the experiment
KMIN		-Clock setting (Min) at the beginning of the experiment
KSEC		-Clock setting (Sec) at the beginning of the experiment
KTIC		-Clock setting (Tic) at the beginning of the experiment

NAME1	-File type identifier
NAME2	-File class name
NAME3	-File number (see subroutine "PLAC2")
NPASS	-File number loop counter
POL(I)	-Digital current level of the polarizer at encoder count I (ACDU)
RTIME	-Running time of the experiment (min)
STIME	-Time (min) of the start of the experiment

SUBROUTINES CALLED

CVTTIM (SYSTEM SUBROUTINE)
EDATA
GTIM (SYSTEM SUBROUTINE)
INPUT
NEWSET
OUT
PLAC2
P1INFO
RATE
RESET
STEPER
T1INFO
WLCALC

```
      PROGRAM DYNSPE
C*********************************************************************
C     OBJECTIVE: TO COORDINATE THE ACTUAL DATA ACQUISITION DURING A
C                TIME DEPENDENT ELLIPSOMETRIC SPECTRAL SCAN WITH
C                OPERATOR REVIEW OFTHE RAW DATA.  THIS IS THE MAIN DATA
C                COLLECTION ROUTINE.
C
C     PROGRAM "SEV002" WRITTEN BY JOSEPH C. FARMER IN SPRING 1981
C     ADAPTION TO CONTINUOUS SCANING DONE BY STEVE T. MAYER IN FALL 1984
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/Z/NAME1,NAME2,NPASS,DURAT,RTIME
      COMMON/STATUS/ISTAT(16)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/STEP/DEVICE,ISTEP,IPC,IAC
      REAL*4 NAME1,NAME2,NAME3
      INTEGER*4 LTIME,KTIME
      INTEGER RTIME,NPASS,JTIME,DURAT
      NPASS=0
C*****THE ENCODER COUNT IS CONVERTED TO WAVELENGTH.*******************
      CALL WLCALC
      TYPE 10
   10 FORMAT(/,'$ INSERT DATA STORAGE DISK, THEN "RETURN".')
      ACCEPT 20,NWAIT
   20 FORMAT(A1)
C*****THE FOURTH CHANNEL OF THE ADAC MODEL 1604/OPI COUNTER CARD IS ***
C     READ AT VARIOUS TIMES TO ASSURE THAT THE RATE OF ROTATION OF
C     THE FILTER DISC(SPECTRAL SCANNING RATE) IS CONSTANT.  PROGRAM
C     EXECUTION IS DELAYED UNTIL A SPECIFIED STABILITY CRITERIA IS
C     MET.  CHANNEL 4 OF THIS COUNTER CARD IS WIRE WRAPPED TO
C     OPERATE IN THE FREQUENCY MODE; THE FREQUENCY OF THE ENCODER
C*****GENERATED PULSES IS MEASURED BY THIS CHANNEL.*******************
   30 CALL RATE
      NSAVE=NRATE
C*****SETUP INFORMATION IS EITHER READ FROM A PREVIOUS EXPERIMENTAL ****
C     DATA FILE OR INPUT FROM THE VT55 KEYBOARD BY THE OPERATOR,
C     DEPENDING UPON THE OPERATOR'S RESPONSE AT THIS POINT.
```

```
  100 TYPE 110
  110 FORMAT(/,'$ IS SETUP INFO TO BE (A)ENTERED FROM KEYBOARD
     C OR (B)READ FROM OLD DATA FILE? ')
      ACCEPT 20,IF1
      IF(IF1.NE.1HA.AND.IF1.NE.1HB)GOTO 100
      IF(IF1.NE.1HA)GOTO 120
C*****THIS SUBROUTINE ALLOWS THE OPERATOR TO INPUT THE SETUP **********
C     INFORMATION FROM THE KEYBOARD.
      CALL NEWSET
      GOTO 130
  120 CONTINUE
C*****THIS SUBROUTINE ACCESSES AN OLD DATA FILE SPECIFIED BY THE ******
C     OPERATOR TO REASSIGN OLD SETUP VALUES TO THE CURRENT
C     EXPERIMENT.
      CALL INPUT
      NRATE=NSAVE
  130 CONTINUE
C*****SOME PARAMETERS MUST BE ENTERED FROM THE KEYBOARD SINCE *********
C     THEY WILL PROBABLY BE CHANGED FROM ONE SPECTRAL SCAN TO
C     ANOTHER(FOR EXAMPLE MANUAL AZIMUTH SETTINGS).  SUBROUTINE
C     "RESET" ALLOWS FOR THIS.
      CALL RESET
C*****THE OPERATOR CAN SPECIFY THE TYPE OF OUTPUT DESIRED FOR *********
C     THE SETUP INFORMATION.
      TYPE 190
  190 FORMAT(/,'$ HOW MANY DEG. SHOULD THE STEP MOTER MOVE
     * WHEN THE "I" GT. 1000 ADCU"S(I2)?')
      ACCEPT 191,IISTEP
      IISTEP=IISTEP*50
  191 FORMAT(I2)
  200 TYPE 210
  210 FORMAT(/,'$ IS SETUP INFORMATION TO BE OUTPUT (A)TO CRT,
     C (B)TO PRINTER, OR (C)NOT AT ALL? ')
      ACCEPT 20,IF2
      IF(IF2.NE.1HA.AND.IF2.NE.1HB.AND.IF2.NE.1HC)GOTO 200
      IF(IF2.EQ.1HC)GOTO 230
      IF(IF2.EQ.1HC)GOTO 230
      IF(IF2.NE.1HA)GOTO 220
C*****SETUP INFORMATION IS DISPLAYED ON THE CRT.**********************
      CALL T1INFO
      GOTO 230
  220 CONTINUE
C*****SETUP INFORMATION IS SENT TO THE LINE PRINTER.******************
      CALL P1INFO
  230 CONTINUE
  400 TYPE 410
  410 FORMAT(/,'$ ENTER "G" AND "RETURN" FOR SPECTRA COLLECTION. ')
      ACCEPT 20,IF3
      IF(IF3.NE.1HG)GOTO 400
```

```
      TYPE 411
  411 FORMAT(/,'  SPECTRA BEING COLLECTED..........')
  412 IF (NPASS.EQ.0) CALL GTIM(KTIME)
      NPASS=NPASS+1
      CALL CVTTIM(KTIME,KHRS,KMIN,KSEC,KTIC)
      STIME=KHRS*60+KMIN+KSEC/60+KTIC/3600
      JTIME=DURAT+STIME
      CALL EDATA
      TYPE 430,NPASS
  430 FORMAT (/10X,'FILE #',I2)
      TYPE 440,IHRSO,IMINO,ISECO,ITICO
  440 FORMAT(/10X,'DATA COLLECTION INITIATED AT: ',3(I2,':'),I2)
      TYPE 450,IHRSF,IMINF,ISECF,ITICF
  450 FORMAT (/10X,'DATA COLLECTION FINISHED AT: ',3(I2,':'),I2)
      NAME1='DY10'
      CALL OUT
C*********************************************************************
C       STEP MOTER CHANGE IN POLARIZER AND ANALYSER SETTING
C       WHEN CURRENTS ARE GETTING TO LARGE
C*********************************************************************
      TYPE *,'     THE MAXIMUM POLARIZER CURRENT= ',POL(IPC),
     C'AT AN ENCODER COUNT= ',IPC
      IF (ABS(POL(IPC)).LT.1500.) GOTO 460
      DEVICE='P'
      IF ((POL(IPC)).GT.0) ISTEP=-IISTEP
      IF ((POL(IPC)).GT.0) GOTO 451
      ISTEP=IISTEP
  451 NPOL=NPOL-ISTEP*2
      CALL STEPER
  460 TYPE *,'     THE MAXIMUM ANALYZER CURRENT= ',ANA(IAC),
     C'AT AN ENCODER COUNT= ',IAC
      IF (ABS(ANA(IAC)).LT.1500) GOTO 470
      DEVICE='A'
      IF ((ANA(IAC)).GT.0) ISTEP=IISTEP
      IF ((ANA(IAC)).GT.0) GOTO 461
      ISTEP=-IISTEP
  461 NANA=NANA+ISTEP*2
      CALL STEPER
  470 CONTINUE
      CALL GTIM(LTIME)
      CALL CVTTIM(LTIME,IHRSL,IMINL,ISECL,ITICL)
      RTIME=IHRSL*60+IMINL+(ISECL/60)+(ITICL/3600)
      RTIME=RTIME
      IF (RTIME.GT.JTIME) GOTO 820
      GOTO 412
  820 CONTINUE
C*****THE OPERATOR HAS THE OPTION OF REPEATING PROGRAM EXECUTION.******
 1000 TYPE 1010
 1010 FORMAT(/,'$ IS THE PROGRAM TO BE EXECUTED AGAIN(Y/N)? ')
```

```
ACCEPT 20,IF7
IF(IF7.NE.1HY.AND.IF7.NE.1HN)GOTO 1000
IF(IF7.EQ.1HY)GOTO 30
STOP
END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM INTREF

```
TSX>RUN LD1:INTREF              {RUN THE PROGRAM

WHAT IS THE GLASSES REFRACTIVE INDEX? 1.5
 THETA=      0.100  PHASE CHANGE       0.596
 THETA=      0.200  PHASE CHANGE       1.193
 THETA=      0.300  PHASE CHANGE       1.789
 THETA=      0.400  PHASE CHANGE       2.385
 THETA=      0.500  PHASE CHANGE       2.981
                 .
                 .
                 .
                 .
 THETA=     20.500  PHASE CHANGE     103.200
 THETA=     20.600  PHASE CHANGE     103.304
 THETA=     20.700  PHASE CHANGE     103.393
STOP --

TSX>ED INTREF.DAT               {LOOK AT THE OUTPUT FILE CREATED

 0.100    0.596
 0.200    1.193
 0.300    1.789
 0.400    2.385
 0.500    2.981
 0.600    3.577
 0.700    4.174
 0.800    4.770
 0.900    5.366
 1.000    5.962
          .
          .
          .
          .
20.600  103.304
20.700  103.393
```

PROGRAM INTREF VARIABLES EXPLANATION

Variable Name	Symbol	EXPLANATION
A		-Angle of incidence of the first reflection in the Fresnel Rhomb (deg. --> rad.)
B		-Phase change associated with the sum of first and third reflections in the Fresnel Rhomb (deg.)
D		-Phase change associated with the second reflection in the Fresnel Rhomb (deg.)
N		-Refractive index of the Fresnel Rhomb interface
Q1		-Total phase change
PH1		-Common block "FLM" variable. Angle of incidence (rad.)

SUBROUTINES CALLED

FILM2
LOOP

```
      PROGRAM INTREF
C*****************************************************************
C     PROGRAM FOR CALCULATION OF THE PHASE CHANGE ASSOCIATED
C     WITH THE INTERNAL REFLECTION FROM THREE REFLECTION FROM
C     THE FRENEL RHOMB
C*****************************************************************
      REAL N,Q2(500),A3(500)
      COMMON/FLM/DELC,PSIC,PHI
      PI=3.1415926
 1    CONTINUE
      TYPE 100
      ACCEPT 200,N
      DO 500 IA=1,450
      A=IA/10.
      A1=A
      A=2*PI*A/360.
      B=PI/2.-A
      D=PI/2.-2*A
      PHI=B
      CALL FILM2(N,0,N,0,1.0,0,4000.,0)
      P1=2*DELC
      PHI=D
      CALL FILM2(N,0,N,0,1.0,0,4000.,0)
      P2=DELC
      Q1=P1+P2
      Q1=-Q1
      TYPE 300,A1,Q1
      Q2(IA)=Q1
      A3(IA)=A1
C     PRINT 300,A1,Q1
 500  CONTINUE
1000  CONTINUE
      OPEN (UNIT=2,NAME='INTREF.DAT',TYPE='NEW',DISP='SAVE')
      DO 23 II=1,450
      CALL LOOP(A3(II),Q2(II))
 23   CONTINUE
 100  FORMAT(/,'$WHAT IS THE GLASSES REFRACTIVE INDEX? ')
 200  FORMAT(F8.4)
 300  FORMAT(2X,'THETA= ',2X,F7.3,2X,'PHASE CHANGE ',2X,F8.3)
      STOP
      END

      SUBROUTINE LOOP(AI,THETA)
      WRITE (2,2) AI,THETA
 2    FORMAT (F7.3,1X,F7.3)
      RETURN
      END
```

```
DEMONSTRATION OF THE EXECUTION OF MDONOP
TSX>RUN [1,1]:MDONOP {RUN THE PROGRAM
INPUT SUBSTRATE RI FILE NAME(A12)DY1SILVERRI
FIRST ENCODER COUNT IN SPECTRUM SWEEP(I3)=100
FINAL ENCODER COUNT IN SPECTRUM SWEEP(I3)=265
STEP SIZE FOR NUMBER OF POINTS TO BE USED= 20
 HOW MANY PARAMETERS ARE IN THE MODEL(I1)?3
```

{DEFINES THE RANGE OF THE DATA TO BE PROCESSED

```
VARIABLE #1MAXIMUM LIMIT= 3.
VARIABLE #1MINIMUM LIMIT= 5.
VARIABLE #2MAXIMUM LIMIT= 2.6
VARIABLE #2MINIMUM LIMIT= 1.5
VARIABLE #3MAXIMUM LIMIT= .7
VARIABLE #3MINIMUM LIMIT= .1
HOW MANY ITERATIONS(I3)?100
```

{INPUT INITIAL GUESSES OF THE MODEL PARAMETERS

```
WHAT IS THE  EXPERIMENT DELTA/PSI DATA FILE CLASS
NAME(A4)?GIRL

WHAT IS THE NUMBER OF THE FILE YOU WANT TO START
WITH(A2)?1

WHAT IS THE FINAL DATA FILE NUMBER?(I3)5

WHAT SHOULD THE REDUCED DATA FILE NAME BE(EG.
PULL.DAT)?GIRL.TST

STOP --
```

{OUTPUT FILE WILL CONTAIN THE REDUCED DATA

```
TSX>ED GIRL.TST                INSPECT THE OUTPUT FILE

    1.     6.80     3.54   1.2062   0.1056   0.0399
    2.     7.99     3.11   1.2964   0.0899   0.0457
    3.     8.85     3.26   1.2669   0.0860   0.0433
    4.     9.57     3.31   1.2573   0.0825   0.0450
    5.    10.07     3.38   1.2450   0.1130   0.0460
```

FILE NUMBER	PRIMARY LAYER THICK.	REFRAC. INDEX	EXTINCT. COEFF.	DELTA ERROR	PSI ERROR

PROGRAM MDONOP VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
ANA(I)		-Psi (x100) at encoder count I (deg.) -substrate extinction coefficient (x1000)
AVD		-Average delta error of the model's best fit
AVP		-Average psi error of the model's best fit
CENTX(J)		-Center value of the jth variable of the simplex
DELC		-model calculation of delta (deg)
DELM(I)		-Measured value of delta (deg) at the ith encoder count
HOLD(NPASS,J)		-Jth parameter model best fit value of file number NPASS
IMAX		-File number of the last data file to be converted
IPASS		-Iteration counter of the overall optimization routine (iteration of the nth file optimization)
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NDAT1		-Number of data points being used in the optimization
NPASS		-File number loop counter
NSTART		-Encoder count where calculations should start at (this defines the first encoder count where spectra data should be used in the calculations)
NSTEP		-Spacing of encoder counts to be considered in the calculation

NSTOP	-Encoder count where calculations should stop
MAX(I)	-Maximum initial guess of the value of the ith variable
MIN(I)	-Minimum initial guess of the value of the ith variable
POL(I)	-Delta (x100) at encoder count I (deg.) -substrate refractive index (x1000)
POW	-Maximum number of iterations per file
PSIC	-model calculation of psi (deg)
PSIM(I)	-The measured value of psi (deg) at the ith encoder count
REF(J)	-Reflected value of the jth variable
REF2(J)	-Double reflected value of the jth variable
TEMP(J)	-Jth variable value of the model sent to the model/measured error calculation
TN1	-Ambient refractive index (non-absorbing)
TNF	-Refractive index (real component) of the film
TNKF	-Extinction coefficient of the film
TNKS(J)	-Extinction coefficient of the substrate
TNS(J)	-Refractive index (real component) of the substrate
TRAC(J)	-contracted value of the jth variable
SIMB	-Simplex point number with the lowest error value
SIMP(I,J)	-Value of the jth variable of the ith simplex point
SIMW	-Simplex point number with the highest error value

WL(I) -Wavelength associated with the ith encoder count

SUBROUTINES CALLED

CSM2
IN
INPUT
LOOP
PSIDEL
WLCALC

EQUATIONS USED FOR CALCULATIONS

see SUBROUTINE CSM2

```
      PROGRAM MDONOP
C*********************************************************************
C     PURPOSE: TO OPTIMIZE SPECTROSCOPIC ELLIPSOMETRIC DATA
C              USING A SIMPLEX ALGORITHM. NO INTERMEDIATE VALUES
C              OF THE SIMPLEX RESPONSE SURFACE IS DISPLAYED ON THE
C              CRT.
C
C     PROGRAM WRITTEN BY STEVE T. MAYER AT L.B.L. SUMMER 1984
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/FLM/DELC,PSIC,PHI
      COMMON/Z/NAME1,NAME2,NPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2,NAME3,FILENM(3)
      REAL TN1,TNF,TNKF,TNS(400)
      REAL TNKS(400),WL(400),MAX(7),MIN(7),REF(7),REF2(7),TEMP(7)
      REAL TRAC(7),SIMP(7,7),DELM(400),PSIM(400),CENTX(7),DEL(7)
      REAL THICK
      REAL HOLD(50,7),ARRAY(7)
      INTEGER NVAR,NSTEP,POW,NFILE,NPASS,NSTART,NSTOP
      COMPLEX RSS,RPS,RSF,RPF
C*******************************************************************
C     DATA INPUT
C*******************************************************************
      CALL WLCALC
      DO 1 I=1,400
      WL(I)=IWAVE(I)
    1 CONTINUE
      TN1=1.338
      TNF=1.81
      TNKF=0.281
    3 CONTINUE
      NPASS=0
      TYPE 2019
      CALL INPUT
      DO 4 I=1,400
      TNS(I)=ABS(POL(I)/1000.)
      TNKS(I)=ABS(ANA(I)/1000.)
    4 CONTINUE
```

```
      TYPE 8888
      ACCEPT 400,NSTART
      TYPE 8889
      ACCEPT 400,NSTOP
      TYPE 8010
      ACCEPT 400,NSTEP
      NSTP=NSTEP
      NDAT1=(NSTOP-NSTART)/NSTEP
      TYPE 5900
      ACCEPT 400,NVAR
      DO 6000,I=1,NVAR
      TYPE 6010,I
      ACCEPT 100,MAX(I)
      TYPE 6020,I
      ACCEPT 100,MIN(I)
 6000 CONTINUE
      TYPE 6030
      ACCEPT 400,POW
      TYPE 2027
      ACCEPT 111,NAME2
      NAME1='DY1C'
      TYPE 2028
      ACCEPT 400,NPASS
      TYPE 2025
      ACCEPT 400,NFILE
C*************************
      NSLOP=NFILE
C*************************
      TYPE 2500
      ACCEPT 111,FILENM(1),FILENM(2),FILENM(3)
C*********************************************************************
C
C     SIMPLEX
C
C**********************************************************************
C
C*********************************************************************
C
C     CREATION OF NVAR+1 VERTEX POINTS
C
C***********************************************************************
      SIMP(1,1)=MAX(1)
      SIMP(1,2)=MAX(2)
      SIMP(1,3)=MAX(3)
      SIMP(2,1)=MIN(1)
      SIMP(2,2)=MIN(2)
      SIMP(2,3)=MIN(3)
      SIMP(3,1)=(MAX(1)+MIN(1))/2.
      SIMP(3,2)=(MAX(2)+MIN(2))/2.
```

```
      SIMP(3,3)=(MAX(3)+MIN(3))/2.
      SIMP(3,4)=(MAX(4)+MIN(4))/2.
      SIMP(4,1)=MAX(1)
      SIMP(4,2)=MIN(2)
      SIMP(4,3)=MAX(3)
  5   CONTINUE
      NAME1='DY1C'
      CALL IN
      DO 6 I=1,400
      DELM(I)=POL(I)/100.
      PSIM(I)=ANA(I)/100.
  6   CONTINUE
      TNK1=0.
      IPASS=0.
 150  CONTINUE
C*****************************************************************
C     COMPUTE SIGMA OF ALL VERTICES
C*****************************************************************
      DO 7 I1=1,NVAR+1
      DO 8 I=1,NVAR
      TEMP(I)=SIMP(I1,I)
  8   CONTINUE
      NFLAG=1
      GOTO 5000
 300  CONTINUE
      SIMP(I1,NVAR+1)=TEMP(NVAR+1)
  7   CONTINUE
      DO 99 I=1,NVAR
      IF (SIMP(I,1).LT.0.0) SIMP(I,1)=0.0
 99   CONTINUE
      IPASS=IPASS+1
C*********************************************************************
C     PRINT OUT PRESENT VALUES OF THE SIMPLEX VERTICIES
C*********************************************************************
C     TYPE 8211
C     TYPE 9001,IPASS,NPASS
      DO 46 I=1,NVAR+1
C*****************************************************************
C     CHANGE THIS LINE WITH CHANGE IN MODEL
C*****************************************************************
C     TYPE 9000,I,SIMP(I,1),SIMP(I,2),SIMP(I,3),SIMP(I,4)
 46   CONTINUE
C**********************************************************************
C     FIND THE HIGHEST AND LOWEST SIGMA VERTICIES
C**********************************************************************
 15   CONTINUE
      SIGW=SIMP(1,NVAR+1)
      SIGB=SIGW
      IW=1
```

```
      IB=1
      DO 9 I1=2,NVAR+1
      IF (SIMP(I1,NVAR+1).GT.SIGW) IW=I1
      IF (SIMP(I1,NVAR+1).GT.SIGW) SIGW=SIMP(I1,NVAR+1)
      IF (SIMP(I1,NVAR+1).LT.SIGB) IB=I1
      IF (SIMP(I1,NVAR+1).LT.SIGB) SIGB=SIMP(I1,NVAR+1)
   9  CONTINUE
C*************************************************************************
C     CHECK IF PRESET NUMBER OF ITERATIONS HAS BEEN ACHIEVED
C*************************************************************************
      IF (IPASS.GE.POW) GOTO 664
C*************************************************************************
C     COMPUTE CENTREX
C*************************************************************************
      DO 10 K2=1,NVAR
      CENTX(K2)=0.0
  10  CONTINUE
      DO 11 J=1,NVAR
      DO 12 K=1,NVAR+1
      IF (K.EQ.IW) GOTO 12
      CENTX(J)=CENTX(J)+SIMP(K,J)
 12   CONTINUE
      CENTX(J)=CENTX(J)/NVAR
 11   CONTINUE
C*************************************************************************
C     FIND RELECTED VERTEX
C*************************************************************************
      DO 13 KK=1,NVAR
      DEL(KK)=0.
 13   CONTINUE
      DO 14 L=1,NVAR
      DEL(L)=-(SIMP(IW,L)-CENTX(L))
 14   CONTINUE
      DO 105 LL=1,NVAR
      REF(LL)=CENTX(LL)+DEL(LL)
 105  CONTINUE
C*************************************************************************
C     FIND SIGMA AT REFLECTED POINT
C*************************************************************************
      DO 16 I=1,NVAR
      TEMP(I)=REF(I)
  16  CONTINUE
      NFLAG=2
      GOTO 5000
 310  CONTINUE
      REF(NVAR+1)=TEMP(NVAR+1)
      IF (REF(NVAR+1).LT.SIMP(IB,NVAR+1)) GOTO 50
      IF (REF(NVAR+1).LT.SIMP(IW,NVAR+1)) GOTO 70
      GOTO 60
```

```
C*********************************************************************
C       FIND AN EXPANDED REFLECTED POINT AND SEE IF IT IS BETTER THAN THE
C       PREVIOUS REFLECTED POINT
C*********************************************************************
  50    DO 51 K1=1,NVAR
        REF2(K1)=REF(K1)+DEL(K1)
  51    CONTINUE
        DO 52 I=1,NVAR
        TEMP(I)=REF2(I)
  52    CONTINUE
        NFLAG=3
        GOTO 5000
  320   CONTINUE
        REF2(NVAR+1)=TEMP(NVAR+1)
        IF (REF2(NVAR+1).GT.REF(NVAR+1)) GOTO 70
C*********************ACCEPT DOUBLE RELECTION POINT*******************
        DO 54 LK=1,NVAR+1
  54    SIMP(IW,LK)=REF2(LK)
C       TYPE 8211
C       TYPE 9020
C       TYPE 9050
C*****************************************************************
C       CHANGE THIS LINE WITH A CHANGE IN MODEL
C*****************************************************************
C       TYPE 9000,IW,SIMP(IW,1),SIMP(IW,2),SIMP(IW,3),SIMP(IW,4)
C       TYPE 8211
        GOTO 90
C*************************************************************
C       FIND CONTRACTED POINT
C*************************************************************
  60    CONTINUE
        DO 61 I2=1,NVAR
        TRAC(I2)=CENTX(I2)-.75*DEL(I2)
  61    CONTINUE
        DO 62 I=1,NVAR
        TEMP(I)=TRAC(I)
  62    CONTINUE
        NFLAG=4
        GOTO 5000
  330   CONTINUE
        TRAC(NVAR+1)=TEMP(NVAR+1)
        IF(TRAC(NVAR+1).GT.SIMP(IW,NVAR+1)) GOTO 80
C*********************ACCEPT POINT CONTRACTION*******************************
        DO 63 LK=1,NVAR+1
        SIMP(IW,LK)=TRAC(LK)
  63    CONTINUE
C       TYPE 8211
C       TYPE 9030
C       TYPE 9050
```

```
C*********************************************************************
C       CHANGE THIS LINE WITH A CHANGE IN MODEL
C*********************************************************************
C       TYPE 9000,IW,SIMP(IW,1),SIMP(IW,2),SIMP(IW,3),SIMP(IW,4)
C       TYPE 8211
        GOTO 90
C*********************ACCEPT REFLECTED POINT*******************
  70    DO 71 I=1,NVAR+1
        SIMP(IW,I)=REF(I)
  71    CONTINUE
C       TYPE 8211
C       TYPE 9010
C       TYPE 9050
C*******************************************************************
C       CHANGE THIS LINE WITH A CHANGE IN MODEL
C*******************************************************************
C       TYPE 9000,IW,SIMP(IW,1),SIMP(IW,2),SIMP(IW,3),SIMP(IW,4)
C       TYPE 8211
        GOTO 90
C*********************************************************************
C       ACCEPT A COMPLETE SIMPLEX CONTRACTION
C*********************************************************************
  80    DO 81 I=1,NVAR+1
        DO 82 J=1,NVAR
        SIMP(I,J)=SIMP(IB,J)+(SIMP(I,J)-SIMP(IB,J))/2.
  82    CONTINUE
  81    CONTINUE
C       TYPE 8211
C       TYPE 9040
C       TYPE 8211
        GOTO 150
  90    GOTO 7
C*********************************************************************
 5000   CONTINUE
        SIGMA=0.0
        PHI=75.*2*3.1415926/360.
        AVD=0.0
        AVP=0.0
        DO 1001 J=NSTART,NSTOP,NSTEP
C*******************************************************************
C
C       INSERT MODEL ROUTINE HERE FOR COMPUTING SIGMA
C
C       USE TEMP(PARAMTER#) IN AS THE PARAMETER PRESENT VALUE
C       USE VARIABLE(J) FOR THE ENCODER COUNT DEP. PARAMETER
C*******************************************************************
        CALL CSM2(TN1,TEMP(2),TEMP(3),TNS(J),TNKS(J),WL(J),TEMP(1),
     *RSS,RPS)
        CALL PSIDEL(RPS/RSS)
```

```
      AVD=AVD+ABS(DELC-DELM(J))
      AVP=AVP+ABS(PSIC-PSIM(J))
 1001 CONTINUE
 1000 CONTINUE
      AVD=AVD/(NSTOP-NSTART)*NSTEP
      AVP=AVP/(NSTOP-NSTART)*NSTEP
      TEMP(NVAR+1)=AVD+2*AVP
C*******************************************************************
C     RETURN TO CORRECT LOCATIONS (REFLECTION,DOUBLE REF.,ETC.)
C*******************************************************************
      IF (NFLAG.EQ.1) GOTO 300
      IF (NFLAG.EQ.2) GOTO 310
      IF (NFLAG.EQ.3) GOTO 320
      IF (NFLAG.EQ.4) GOTO 330
      IF (NFLAG.EQ.5) GOTO 200
C*************************************************************
C    FINAL PRINT OUT
C*************************************************************
  664 CONTINUE
  95  CONTINUE
C*************************************************************************
C     CHANGE THIS LINE WITH A CHANGE IN MODEL
C*************************************************************************
  96  CONTINUE
      DO 4070 I=1,NVAR
      TEMP(I)=SIMP(IB,I)
 4070 CONTINUE
C     TYPE *,' '
C     TYPE *,'****FINAL COMPUTATION FOR FILE ',NPASS,' IN PROGRESS****'
C     TYPE *,' '
      NFLAG=5
      NSTEP=1
      GOTO 5000
 200  CONTINUE
C     TYPE *,'THE AVERAGE ABS. ERROR IN DELTA=  ',AVD
C     TYPE *,'THE AVERAGE ABS. ERROR IN PSI= ',AVP
C     TYPE 9000,IB,SIMP(IB,1),SIMP(IB,2),SIMP(IB,3),SIMP(IB,4)
      HOLD(NPASS,1)=SIMP(IB,1)
      HOLD(NPASS,2)=SIMP(IB,2)
      HOLD(NPASS,3)=SIMP(IB,3)
      HOLD(NPASS,4)=AVD
      HOLD(NPASS,5)=AVP
C*****************************************
      IF (NPASS.GE.NFILE) GOTO 211
      NPASS=NPASS+1
C*****************************************
      NSTEP=NSTP
      SIMP(1,1)=SIMP(IB,1)
      SIMP(1,2)=SIMP(IB,2)
```

```
      SIMP(1,3)=SIMP(IB,3)
      SIMP(2,1)=SIMP(IB,1)-SIMP(IB,1)/3.
      SIMP(2,2)=SIMP(IB,2)+SIMP(IB,2)/5.
      SIMP(2,3)=SIMP(IB,3)-SIMP(IB,3)/10.
      SIMP(3,1)=SIMP(IB,1)+SIMP(IB,1)/3.
      SIMP(3,2)=SIMP(IB,2)-SIMP(IB,2)/5.
      SIMP(3,3)=SIMP(IB,3)+SIMP(IB,3)/10.
      SIMP(4,1)=SIMP(IB,1)+SIMP(IB,1)/3.
      SIMP(4,2)=SIMP(IB,2)+SIMP(IB,2)/5.
      SIMP(4,3)=SIMP(IB,3)-SIMP(IB,3)/10.
  207 CONTINUE
      DO 208 I=1,NVAR+1
      IF (SIMP(I,1).LT.0.0) SIMP(I,1)=0.0
  208 CONTINUE
      GOTO 5
  211 CONTINUE
      OPEN(UNIT=2,NAME=FILENM,TYPE='NEW',disp='SAVE')
      DO 212 J=1,NSLOP
      ARRAY(1)=HOLD(J,1)
      ARRAY(2)=HOLD(J,2)
      ARRAY(3)=HOLD(J,3)
      ARRAY(4)=HOLD(J,4)
      ARRAY(5)=HOLD(J,5)
      CALL LOOP(J,ARRAY(1),ARRAY(2),ARRAY(3),ARRAY(4),ARRAY(5))
212   CONTINUE
  100 FORMAT(F10.4)
  110 FORMAT (A1)
  111 FORMAT (3A4)
  400 FORMAT (I3)
 2019 FORMAT(/,'$INPUT SUBSTRATE RI FILE NAME(A12)')
 2025 FORMAT (/,'$WHAT IS THE FINAL DATA FILE NUMBER?(I3)')
 2027 FORMAT (/,'$WHAT IS THE  EXPERIMENT DELTA/PSI DATA FILE CLASS
     C NAME(A4)?')
 2028 FORMAT (/,'$WHAT IS THE NUMBER OF THE FILE YOU WANT TO START
     C WITH(A2)?')
 2500 FORMAT (/,'$WHAT SHOULD THE REDUCED DATA FILE NAME BE
     C(EG. PULL.DAT)?')
 3000 FORMAT (/,'$WHAT IS THE FILM REFRACTIVE INDEX(F10.4)? ')
 3010 FORMAT (/,'$WHAT IS THE FILM EXTINCTION COEFF.(F10.4)? ')
 5900 FORMAT (,'$ HOW MANY PARAMETERS ARE IN THE MODEL(I1)?')
 6010 FORMAT (,'$VARIABLE #',I1,'MAXIMUM LIMIT= ')
 6020 FORMAT (,'$VARIABLE #',I1,'MINIMUM LIMIT= ')
 6030 FORMAT (,'$HOW MANY ITERATIONS(I3)?')
 8010 FORMAT (,'$STEP SIZE FOR NUMBER OF POINTS TO BE USED= ')
 8211 FORMAT (/,'*****************************************************')
C*********************************************************************
C       CHANGE THIS FORMAT STATEMENT AS NEEDED
C*********************************************************************
 8888 FORMAT(/,'$FIRST ENCODER COUNT IN SPECTRUM SWEEP(I3)=')
```

```
8889 FORMAT(/,'$FINAL ENCODER COUNT IN SPECTRUM SWEEP(I3)=')
9000 FORMAT (/,2X,'#',I2,1X,'T1=',F8.4,1X,'NF=',F8.2,1X,'KF=',
    *F8.2,1X,'SIGMA=',F8.5)
9001 FORMAT(/,14X,'PASS NUMBER=',I3,3X,'FILE NUMBER=',I3)
9010 FORMAT (/,5X,'--REFLECTION ACCEPTED--')
9020 FORMAT (/,5X,'--DOUBLE REFLECTION ACCEPTED--')
9030 FORMAT (/,5X,'--CONTRACTION OF WORST POINT ACCEPTED--')
9040 FORMAT (/,5X,'--CONTRACTION OF SIMPLEX ACCEPTED--')
9050 FORMAT (/,5X,'POINT # ',I1,'HAVE BEEN REPLACED BY')
     STOP
1060 END

       SUBROUTINE LOOP(J,ARR1,ARR2,ARR3,ARR4,ARR5)
       REAL AD
       AD=J*1.0
       WRITE(2,2) AD,ARR1,ARR2,ARR3,ARR4,ARR5
2      FORMAT(F7.0,2X,F7.2,1X,F7.2,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4)
       RETURN
       END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM MDQOUT

```
TSX>RUN LD1:MDQOUT              {RUN THE PROGRAM
WHAT IS THE FILM REFRACTIVE INDEX(F10.4)? 1.81
WHAT IS THE FILM EXTINCTION COEF.(F10.4)? 0.281
NAME OF SPECTRA FILE TO BE RETRIEVED? DY1SILVERRI
FIRST ENCODER COUNT IN SPECTRUM SWEEP(I3)=100
FINAL ENCODER COUNT IN SPECTRUM SWEEP(I3)=265
VARIABLE #1VALUE = 50.
VARIABLE #2VALUE = 1000.          {INPUT OPTIMIZED DATA
VARIABLE #3VALUE = 100.           {FROM MDQNOP
VARIABLE #4VALUE = .2
VARIABLE #5VALUE = .1
WHAT SHOULD THE THEORETICAL SPECTRAL FILE NAME BE?
 FILE NAME FOR THE NEW SPECTRAL SCAN (A12) = DY1MDQNOPDEM
DO YOU WANT TO CREATE ANOTHER THEORETICAL SPECTA(Y/N)?N
STOP --
```

PROGRAM MDQOUT VARIABLE EXPLANATIONS

Name	Variable Symbol	EXPLANATION
ANA(I)		-Substrate extinction coefficient (x1000)
DELC		-Model calculation of delta (deg)
MAX(I)		-Value of the model's ith variable for the spectra to be created
NSTART		-Encoder count where calculations should start at (this defines the first encoder count where spectra data should be used in the calculations)
NSTEP		-Spacing of encoder counts to be considered in the calculation
NSTOP		-Encoder count where calculations should stop POL(I)
POL(I)		-Substrate refractive index (x1000)
PSIC		-Model calculation of psi (deg)
PHI	φ_o	-Angle of incidence of light (rad)
RPF	$\int_0^\infty \zeta(T_2, T_{AVG})\, r_{b,p}\, dT_2$	-Resultant p- component reflection coefficient of the distribution of crystals
RPF1	$r_{b,p}$	-P- component of the reflection coefficient of crystals of size THICK
RPS	R_p	-Total resultant p- component reflection coefficient of the model Q surface
	$r_o{}^{2/p}$	-P- reflection coefficient of the primary layer
RSF	$\int_0^\infty \zeta(T_2, T_{AVG})\, r_{b,s}\, dT_2$	-Resultant s- component reflection coefficient of the distribution of crystals

RSF1	$r_{b.s}$	-S- component of the reflection coefficient of crystals of size THICK
RSS	R_s	-Total resultant s- component reflection coefficient of the model Q surface
	$r_{02,s}$	-S- reflection coefficient of the primary layer
TEMP(J)		-Jth variable value of the model sent to the spectra calculation section of the program
TNE1		-Effective refractive index of the surface with a primary layer on top of the substrate
TNKE1		-Effective extinction coefficient of the surface with a primary layer on top of the substrate
TN1	n_o	-Ambient refractive index (non-absorbing)
TNF	n_{f_1}	-Refractive index (real component) of the film
TNFE	n_{f_2}	-Effective refractive index of the hydrated crystals
TNKFE	K_{f_2}	-Effective extinction coefficient of the hydrated crystals
TNKF	K_{f_1}	-Extinction coefficient of the film
TNKS(J)	K_s	-Extinction coefficient of the substrate
TNS(J)	n	-Refractive index (real component) of the substrate
WL(I)	λ	-Wavelength associated with the ith encoder count

SUBROUTINES CALLED

CSM2
IN
INPUT
LOOP
PSIDEL
WLCALC

EQUATIONS USED FOR CALCULATIONS

$$\zeta(T_2, T_{AVG}) = \frac{\exp\left[\frac{1}{2}\left(\frac{T_{AVG}-T_2}{\sigma}\right)^2\right]}{\int_0^\infty \exp\left[\frac{1}{2}\left(\frac{T_{AVG}-T_2}{\sigma}\right)^2\right] dT_2}$$

$$R_j = r_{aj}(1-\Omega) + \Omega \int_0^\infty \zeta(T_2, T_{AVG})\, r_{b,j}\, dT_2$$

$$r_{bij} = \frac{(r_{01,j}+r_{12,j}e^{-iD_1})+(r_{01}r_{12}+e^{-iD_1})r_{23}e^{-iD_2}}{(1+r_{01,j}r_{12,j}e^{-iD_2})+(r_{12,j}+r_{01,j}e^{-iD_1})r_{23}e^{-iD_2}}$$

with

$$D_1 = 4\pi \left(\frac{\pi}{\lambda}\right) \hat{n}_{f_1} \cos(\hat{\phi}_{f_1})$$

$$D_2 = 4\pi \left(\frac{T_2}{\lambda}\right) \hat{n}_{f_2} \cos(\hat{\phi}_{f_2})$$

$$r_{01,p} = \frac{\tan(\hat{\phi}_o-\hat{\phi}_{f_1})}{\tan(\phi_o+\hat{\phi}_{f_1})}$$

$$r_{01,s} = -\frac{\sin(\phi_o-\hat{\phi}_{f_1})}{\sin(\phi_o+\hat{\phi}_{f_1})}$$

$$r_{12,p} = \frac{\tan(\phi_{f_1}-\hat{\phi}_{f_2})}{\tan(\phi_{f_1}+\hat{\phi}_{f_2})}$$

$$r_{12,s} = -\frac{\sin(\phi_f-\hat{\phi}_{f_2})}{\sin(\phi_f+\hat{\phi}_{f_2})}$$

$$r_{23,p} = \frac{\tan(\hat{\phi}_{f_2}-\hat{\phi}_s)}{\tan(\hat{\phi}_{f_2}+\hat{\phi}_s)}$$

$$r_{23,s} = -\frac{\sin(\hat{\phi}_{f_2}-\hat{\phi}_s)}{\sin(\phi_{f_2}+\phi_s)}$$

$$r_{02,s} = -\frac{\sin(\phi_o - \hat{\phi}_{f_2})}{\sin(\phi_o + \hat{\phi}_{f_2})}$$

$$r_{02,p} = \frac{\tan(\hat{\phi}_o - \hat{\phi}_{f_2})}{\tan(\phi_o + \hat{\phi}_{f_2})}$$

$$r_{a,j} = \frac{r_{02,j} + r_{23,j}\, e^{-iD_1}}{1 + r_{02,j}\, r_{23,j}\, e^{-iD_1}}$$

$$n_o \sin(\phi_o) = \hat{n}_{f_1} \sin(\hat{\phi}_{f_1}) = \hat{n}_s \sin(\hat{\phi}_s) = \hat{n}_{f_2} \sin(\hat{\phi}_{f_2})$$

$\hat{n}_{f_2}$ is the complex refractive index of the crystals calculated from the Bruggeman equation.

```
      PROGRAM MDQOUT
C*********************************************************************
C     PURPOSE: TO CREATE A ELLIPSOMETRIC SPECTRA FILE CORRESPONDING
C              TO A SET OF PARAMETERS FOR MODEL "Q"
C
C     PROGRAM WRITTEN BY STEVE T. MAYER AT L.B.L. WINTER 1984
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/FLM/DELC,PSIC,PHI
      REAL*4 DBLK(2),NAME(20)
      REAL TN1,TNF,TNKF,TNS(400),TNKS(400),WL(400),MAX(7),TEMP(7)
      REAL THICK,TFRACT(200)
      INTEGER NVAR,NSTEP,POW,NFILE,NPASS,NSTART,NSTOP
      COMPLEX RSS,RPS,RSF,RPF,RSF1,RPF1
C*******************************************************************
C     DATA INPUT
C*******************************************************************
      CALL WLCALC
      DO 1 I=1,400
      WL(I)=IWAVE(I)
    1 CONTINUE
      TYPE 101
      ACCEPT 100,TNF
      TYPE 102
      ACCEPT 100,TNKF
      TN1=1.338
    3 CONTINUE
      TYPE 2027
      CALL INPUT
      DO 4 I=1,400
      TNS(I)=ABS(POL(I)/1000.)
      TNKS(I)=ABS(ANA(I)/1000.)
    4 CONTINUE
      TYPE 8888
      ACCEPT 400,NSTART
      TYPE 8889
      ACCEPT 400,NSTOP
      NSTEP=1
```

```
      NDAT1=(NSTOP-NSTART)/NSTEP
      NVAR=5
    5 CONTINUE
      DO 6000,I=1,NVAR
      TYPE 6010,I
      ACCEPT 100,MAX(I)
 6000 CONTINUE
    6 CONTINUE
      TNK1=0.
      IPASS=0.
  150 CONTINUE
      DO 8 I=1,NVAR
      TEMP(I)=MAX(I)
    8 CONTINUE
      NFLAG=1
      SIGMA=0.0
C***********************************************************************
 5000 CONTINUE
      WFRAC=0.0
      PHI=75.*3.1415*2./360.
      NTH1=(TEMP(2)-2.5*TEMP(3))/20
      NTH2=(TEMP(2)+2.5*TEMP(3))/20
      DO 4050 ITHICK=NTH1,NTH2
      THICK=20*ITHICK
      TFRACT(ITHICK)=1/(SQRT(2*3.1415))*EXP(-.5*((TEMP(2)-THICK)
     * /TEMP(3))**2)
      WFRAC=WFRAC+TFRACT(ITHICK)
 4050 CONTINUE
      DO 4060 I=NTH1,NTH2
      TFRACT(I)=TFRACT(I)/WFRAC
 4060 CONTINUE
      DO 1001 J=NSTART,NSTOP,NSTEP
      CALL CSM2(TN1,TNF,TNKF,TNS(J),TNKS(J),WL(J),TEMP(1),RSS,RPS)
      CALL PSIDEL(RPS/RSS)
      CALL REFIND(TN1,TNE1,TNKE1)
      TNE1=ABS(TNE1)
      TNKE1=ABS(TNKE1)
      CALL EMA(TN1,0.0,TNF,TNKF,TEMP(5),TNFE,TNKFE)
      DO 1002, ITHICK=NTH1,NTH2
      THICK=20*ITHICK
      CALL CSM2(TN1,TNFE,TNKFE,TNE1,TNKE1,WL(J),THICK,RSF1,RPF1)
      RSF=RSF+TFRACT(ITHICK)*RSF1
      RPF=RPF+TFRACT(ITHICK)*RPF1
 1002 CONTINUE
      RSS=RSS*(1-TEMP(4))+RSF*TEMP(4)
      RPS=RPS*(1-TEMP(4))+RPF*TEMP(4)
      RSF=0.0
      RPF=0.0
      RPS=RPS/RSS
```

```
      CALL PSIDEL(RPS)
      IF (DELC.GT.180.) DELC=DELC-360.
      POL(J)=DELC*100
      ANA(J)=PSIC*100.
1001  CONTINUE
1000  CONTINUE
      TYPE 9000
      CALL OUTPUT
      TYPE 9050
      ACCEPT 110,IFLAG1
      IF (IFLAG1.EQ.'Y') GOTO 5
 100 FORMAT(F10.4)
 101 FORMAT(/,'$WHAT IS THE FILM REFRACTIVE INDEX(F10.4)? ')
 102 FORMAT(/,'$WHAT IS THE FILM EXTINCTION COEF.(F10.4)? ')
 110 FORMAT (A1)
 111 FORMAT (A4)
 400 FORMAT (I3)
2027 FORMAT (/,'WHAT IS THE SUBSTATE RI DATA FILE NAME?')
6010 FORMAT (,'$VARIABLE #',I1,'VALUE = ')
8888 FORMAT(/,'$FIRST ENCODER COUNT IN SPECTRUM SWEEP(I3)=')
8889 FORMAT(/,'$FINAL ENCODER COUNT IN SPECTRUM SWEEP(I3)=')
9000 FORMAT(/,'WHAT THEORETICAL SPECTRAL FILE NAME BE?')
9050 FORMAT(/,'$DO YOU WANT TO CREATE ANOTHER THEORETICAL
    *SPECTA(Y/N)?')
     STOP
1060 END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM NUCNUM

```
TSX>COPY BMUD.DAT PLTDAT.DAT          {NUCNUM READ DATA FROM
                                       THE FILE PLTDAT.DAT

TSX>ED PLTDAT.DAT                     {PLTDAT.DAT IS NOW A
                                       REDUCED DATA FILE FROM
                                       MD9NOP

        BMUD.DAT
   1.     0.59     0.00  1.0000  0.0000  0.3197  0.1185
   2.    43.21     0.00  1.0000  0.0000  0.4756  0.3846
   3.    64.67     0.00  1.0000  0.0000  0.3193  0.4894
   4.     5.01   133.48  0.8090  0.7920  0.6736  0.4873
                  :
  45.    31.78  1697.98  0.4280  0.1310  2.9155  1.2688
  46.    31.35  1675.03  0.4270  0.1307  2.9537  1.2886
  47.    31.81  1699.31  0.4221  0.1292  2.7700  1.2499
  48.    31.96  1707.82  0.4241  0.1298  2.6135  1.1797
  49.    31.77  1697.53  0.4194  0.1284  2.7132  1.2060
  50.    31.36  1675.59  0.4085  0.1251  2.6939  1.3521
```

FILE NUMBER	PRIMARY LAYER THICK. (A°)	SECOND. LAYER THICK. (A°)	SURF. COV.	FRACT. SOLID IN CRYSTALS	DELTA ERROR	PSI ERROR

```
TSX>RUN LD1:NUCNUM                    RUN THE PROGRAM

HOW MANY DATA FILES WERE REDUCED?  50

STOP --

TSX>ED PLTDAT.DAT                     {INSPECT THE NEW PLTDAT.DAT FILE

     .DAT
   1.     0.59     0.00  1.0000  0.0000  0.3197  0.1185    0.00
   2.    43.21     0.00  1.0000  0.0000  0.4756  0.3846    0.00
   3.    64.67     0.00  1.0000  0.0000  0.3193  0.4894    0.00
   4.     5.01   133.48  0.8090  0.7920  0.6736  0.48731443.92
                  :
  45.    31.78  1697.98  0.4280  0.1310  2.9155  1.2688    4.72
  46.    31.35  1675.03  0.4270  0.1307  2.9537  1.2886    4.84
  47.    31.81  1699.31  0.4221  0.1292  2.7700  1.2499    4.65
  48.    31.96  1707.82  0.4241  0.1298  2.6135  1.1797    4.62
  49.    31.77  1697.53  0.4194  0.1284  2.7132  1.2060    4.63
  50.    31.36  1675.59  0.4085  0.1251  2.6939  1.3521    4.63
  50.    31.36  1675.59  0.4085  0.1251  2.6939  1.3521    4.63
```

FILE NUMBER	PRIMARY LAYER THICK. (A°)	SECOND. LAYER THICK. (A°)	SURF. COV.	FRACT. SOLID IN CRYSTALS	DELTA ERROR	PSI ERROR	NUMBER DENSITY OF CRYSTALS (#X10^{6})

PROGRAM NUCNUM VARIABLE EXPLANATIONS-

Variables of interest

Variable Name	Symbol	EXPLANATION
ARRAY(L,J)		value of the Lth file numbers Jth optimized parameter

SUBROUTINES CALLED

LOOP

EQUATIONS USED FOR CALCULATIONS

$$N_c = \frac{\Omega \times 10^{16}}{\pi r^2}$$

```
      PROGRAM NUCNUM
C*****************************************************************
C
C     PURPOSE: TO USE THE OPTIMIZED MODEL9 DATA FROM THEIR DATA
C              FILES AND OUTPUT A DATA NEW FILE CONTAINING BOTH
C              THE OPTIMIZED MODEL9 VALUES AND THE CALCULATED
C              NUMBER DENSITY OF SURFACE NUCLII.
C
C     WRITTEN BY STEVEN T. MAYER SUMMER 1985.
C*****************************************************************
      COMMON ARRAY(50,9),ID(20)
      TYPE 1
      ACCEPT 2,NFILE
 1000 OPEN(UNIT=1,NAME='DY1:PLTDAT.DAT',TYPE='OLD')
      READ(1,5)(ID(I),I=1,20)
      TYPE 5,(ID(I),I=1,20)
      DO 4000 L=1,NFILE
      READ(1,4)(ARRAY(L,J),J=1,7)
      TYPE *, ARRAY(L,3)
 4000 CONTINUE
      CLOSE(UNIT=1,DISPOSE='SAVE')
      DO 5000 I=1,NFILE
      IF (ARRAY(I,4).GE.1.0) GOTO 4500
      ARRAY(I,8)=ARRAY(I,4)*3.18E15/(ARRAY(I,3)**2)
      ARRAY(I,8)=ARRAY(I,8)/1E08
      GOTO 5000
 4500 ARRAY(I,8)=0.0
 5000 CONTINUE
      OPEN(UNIT=1,NAME='DY1:PLTDAT.DAT',TYPE='NEW')
      WRITE(1,5)(ID(I),I=1,20)
      DO 6000 J=1,NFILE
      CALL LOOP(J)
 6000 CONTINUE
      CLOSE(UNIT=1,DISPOSE='SAVE')
    1 FORMAT(/,'$HOW MANY DATA FILES WERE REDUCED?  ')
    2 FORMAT(I2)
    3 FORMAT(A1)
    4 FORMAT(F7.0,1X,F7.2,1X,F7.2,4(1X,F7.4))
    5 FORMAT(/,5X,20A2)
    6 FORMAT(I2)
      STOP
      END


      SUBROUTINE LOOP(J)
      COMMON ARRAY(50,9),ID(20)
      REAL AD
      AD=J*1.0
      WRITE(1,2) AD,ARRAY(J,2),ARRAY(J,3),(ARRAY(J,K),K=4,8)
```

```
      TYPE *,ARRAY(J,3)
2     FORMAT(F7.0,2X,F7.2,1X,F7.2,4(1X,F7.4),F7.2)
      RETURN
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM OPTCOM

```
TSX>RUN LD1:OPTCOM                    {RUN THE PROGRAM
                                                        {FILE CLASS NAME
WHAT IS THE BASE 4 LETTER FILE NAME (A4)   PR51         {FROM 4ZONE MEASUREMENT
ABS. DELTA ERR=    2.287412      ABS PSI ERR=    0.9549717
S.D IN DELTA=      2.332691      S.D. IN PSI=    1.199025

ANOTHER 4 ZONE FILE CLASS(Y/N)?N
STOP --
```

PROGRAM OPTCOM VARIABLE EXPLANATIONS-

Variables of interest

Variable Name	Symbol	EXPLANATION
ANA(I)		-Psi (x100) at encoder count I (deg.)
AAVG(I)		-Average psi value of the 4 zone measurement
SIGM1		-Average delta error of the 4 zone measurement (deg)
SIGM2		-Average psi error of the 4 zone measurement (deg)
SIGM3		-Standard deviation of delta (deg)
SIGM4		-Stardard deviation of psi (deg)
IMAX		-File number of the last data file to be read
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NPASS		-File number loop counter
NSTART		-Encoder count where calculations should start at (this defines the first encoder count where spectra data should be used in the calculations
NSTOP		-Encoder count were calculations should stop
POL(I)		-Delta (x100) at encoder count I (deg.)
PAVG(I)		-Average delta value of the 4 zone measurement

SUBROUTINES CALLED

IN
WLCALC

EQUATIONS USED FOR CALCULATIONS

$$\sigma = \frac{1}{4}\sqrt{[\ \beta_j(I) - \ \beta_{avg}(I)]^2}$$

where $\beta_j(I)$ is the value of delta or psi in the jth zone at encoder count I, and $\beta_{avg}(I)$ is the 4 zone average value of delta or psi at encoder count I.

```
      PROGRAM OPTCOM
C*********************************************************************
C     OBJECTIVE: TO FINE THE STANDARD DEVIATION BETWEEN 4ZONE MEASUREMENTS
C                AND THEIR 4ZONE AVERAGE
C
C     PROGRAM WRITTEN S. MAYER SUMMER 1985.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2,PAVG(400),AAVG(400)
      INTEGER*2 ANGOPT
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      CALL WLCALC
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1001
      ACCEPT 2000,NAME2
      NSTART=100
      NSTOP=265
      IMAX=4
      POINT=4.*(NSTOP-NSTART)
      DO 3 IWV=1,399
      PAVG(IWV)=0
      AAVG(IWV)=0
   3  CONTINUE
      DO 8000 NPASS=1,IMAX
      IPASS=NPASS
      NAME1='DY1C'
      CALL IN
      DO 4 IWV=NSTART,NSTOP
      PAVG(IWV)=PAVG(IWV)+(POL(IWV)/100.)
      AAVG(IWV)=AAVG(IWV)+(ANA(IWV)/100.)
    4 CONTINUE
8000  CONTINUE
```

```
      DO 5 IWV=NSTART,NSTOP
      PAVG(IWV)=PAVG(IWV)/IMAX
      AAVG(IWV)=AAVG(IWV)/IMAX
    5 CONTINUE
      SIGM1=0.
      SIGM2=0.
      SIGM3=0.
      SIGM4=0.
      DO 9000 NPASS=1,IMAX
      IPASS=NPASS
      NAME1='DY1C'
      CALL IN
      DO 6 IWV=NSTART,NSTOP
      SIGM1=SIGM1+ABS(PAVG(IWV)-POL(IWV)/100.)
      SIGM2=SIGM2+ABS(AAVG(IWV)-ANA(IWV)/100.)
      SIGM3=SIGM3+(POL(IWV)/100.-PAVG(IWV))**2
      SIGM4=SIGM4+(ANA(IWV)/100.-AAVG(IWV))**2
  6   CONTINUE
9000  CONTINUE
      SIGM1=(SIGM1)/(POINT)
      SIGM2=(SIGM2)/(POINT)
      SIGM3=SQRT(SIGM3)/SQRT(POINT)
      SIGM4=SQRT(SIGM4)/SQRT(POINT)
      TYPE *,'ABS. DELTA ERR=',SIGM1,'ABS PSI ERR= ',SIGM2
      TYPE *,'S.D IN DELTA=  ',SIGM3,'S.D. IN PSI=',SIGM4
   10 TYPE 1018
      ACCEPT 100,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 1
      IF (FLAG2.NE.'N') GOTO 10
  100 FORMAT(A1)
 1001 FORMAT(/,'$WHAT IS THE BASE 4 LETTER FILE NAME (A4)   ')
 1004 FORMAT(I2)
 1018 FORMAT(/,'$ANOTHER 4 ZONE FILE CLASS(Y/N)?')
 2000 FORMAT(A4)
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF PLTCUR

```
Csa2>RUN  PLTCUR    {RUN THIS PROGRAM ON THE VMS
                    {COMPUTER WITH GRAPHIC CAPABILITIES
WHAT IS THE SCALE OF THE PLOT(<= 1.0)?1.0
ENTER MODEL (1=VT125, 2=VK100, 3=VT240, 4=350PC)
3
ENTER MONITOR TYPE (0=MONOCHROME, 1=COLOR)
0
WHAT IS THE SCALE FOR THE DATA FILE?5
WHAT IS VIEW PHI(DEG.)?270.0
WHAT IS VIEW THETA(DEG.)?0.0
WHAT IS VIEW RADIUS(DEG.)?20.
MORE PLOTS(Y/N)?N
FORTRAN STOP
```

PROGRAM PLTCUR VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
CIR(I,J)		-Local current at the x,y position i,j on the electrode surface as calculated from "CURDIS"
IEMAX		-Maximum array value in the x direction which corresponds to a point on the electrode's surface
ISCAL		-Scale used in the calculation of the current distribution from routine "CURDIS"
JEMAX		-Maximum array value in the y direction which corresponds to a point on the electrode's surface
PHI		-Angle of view of the 3-D plot from the xy plane
RADIUS		-Radial distance from the origin where the view is placed
SCALE		-Scale of the 3-D plot
THETA		-Angle of view of the 3-D plot from the yz plane
X3AXIS		-X scale of the plot
Y3AXIS		-Y scale of the plot
Z3AXIS		-Z scale of the plot

SUBROUTINES CALLED

AREA2D	ISSCO PRODUCT
ENDPL	ISSCO PRODUCT
GRAF3D	ISSCO PRODUCT
NOHIDE	ISSCO PRODUCT
PAGE	ISSCO PRODUCT
SURMAT	(LBL EMULATION SUBROUTINE)
VOLM3D	ISSCO PRODUCT
VUANGL	ISSCO PRODUCT
X3NAME	ISSCO PRODUCT
Y3NAME	ISSCO PRODUCT
Z3NAME	ISSCO PRODUCT

```
      PROGRAM PLTCUR
C*****************************************************************
C
C     PURPOSE: TO PLOT THE THREE DIMENSIONAL CURRENT DISTRIBUTION
C              GIVEN IN FILE "CURDIS.DAT"
C
C     WRITTEN BY STEVEN T. MAYER SUMMER 1985.
C
C     NOTE:   MANY OF THE SUBROUTINES CALLED IN THIS PROGRAM ARE
C             ISSCO PRODUCT ROUTINES, FOR 3D PLOTING WITH "DISSPLA".
C
C*****************************************************************
      COMMON/LOOP/ CIR(30,30),J
      DIMENSION ZBOT(28,10),CUR(28,10)
      REAL X,Y
      TYPE 50
      ACCEPT 100,SCALE
      X3AXIS=SCALE*7.62
      Y3AXIS=SCALE*7.62
      Z3AXIS=Y3AXIS
      XPAGE=SCALE*7.62
      YPAGE=SCALE*7.62
C*********************************************************************
C     SET THE OUTPUT FORMAT FOR A FILE WHICH CAN BE PUT OUTPUT
C     TO THE TALARS PRINTER/PLOTTER.  OTHER PLOTTERS CAN BE USED.
C     SEE LBL DOCUMENTATION "DISSPLA.DOC".
C*********************************************************************
      CALL TALARS
      TYPE 81
      ACCEPT 110,ISCAL
      IEMAX=5.94*ISCAL
      JEMAX=2.38*ISCAL
      J=JEMAX
      Z3MAX=0.
      Z3ORIG=0.
      REWIND (1)
      OPEN(UNIT=1,FILE='CURDIS.DAT',TYPE='OLD',DISP='SAVE',FORM='
     *FORMATTED')
      DO 23 II=1,IEMAX-1
      CALL LOOP(II)
      DO 22 L=1,JEMAX-1
      CUR(II,L)=CIR(II,L)
22    CONTINUE
23    CONTINUE
      DO 24 I=1,IEMAX-1
      DO 25 J=1,JEMAX-1
      IF (CUR(I,J).GT.Z3MAX) Z3MAX=CUR(I,J)
      ZBOT(I,J)=0.
25    CONTINUE
```

```
 24     CONTINUE
        Z3STP=(Z3MAX-Z3ORIG)/2.
1       CONTINUE
        TYPE 60
        ACCEPT 100,PHI
        TYPE 70
        ACCEPT 100,THETA
        TYPE 80
        ACCEPT 100,RADIUS
        X3ORIG=1.
        X3MAX=1.*(IEMAX-1)
        X3STP=X3MAX/2.
        Y3ORIG=1.
        Y3MAX=1.*(JEMAX-1.)
        Y3STP=Y3MAX/2.
        CALL PAGE(XPAGE,YPAGE)
        CALL AREA2D(XPAGE-1.,YPAGE-1.)
        CALL VUANGL(PHI,THETA,RADIUS)
        CALL NOHIDE
        CALL VOLM3D(X3AXIS,Y3AXIS,Z3AXIS)
        CALL X3NAME('XAXIS$',100)
        CALL Y3NAME('YAXIS$',100)
        CALL Z3NAME('ZAXIS$',100)
        CALL GRAF3D(X3ORIG,X3STP,X3MAX,Y3ORIG,Y3STP,Y3MAX
     *,Z3ORIG,Z3STP,Z3MAX)
        CALL SURMAT(CUR,1,28,1,10,0)
        CALL SURMAT(ZBOT,2,28,2,10,0)
20      CONTINUE
        CALL ENDPL(0)
        TYPE 4
        ACCEPT 5,WAIT
        IF (WAIT.EQ.'N') GOTO 6
        GOTO 1
  6     CONTINUE
  4     FORMAT(/,'$ MORE PLOTS(Y/N)?')
  5     FORMAT(A1)
 50     FORMAT(/,'$WHAT IS THE SCALE OF THE PLOT(<= 1.0)?')
 60     FORMAT(/,'$WHAT IS VIEW PHI(DEG.)?')
 70     FORMAT(/,'$WHAT IS VIEW THETA(DEG.)?')
 80     FORMAT(/,'$WHAT IS VIEW RADIUS(DEG.)?')
 81     FORMAT(/,'$WHAT IS THE SCALE FOR THE DATA FILE?')
110     FORMAT(I3)
105     FORMAT(1X,12A1)
100     FORMAT(F8.4)
        STOP
        END

        SUBROUTINE LOOP(II)
        COMMON/LOOP/ CIR(30,30),JEMAX
```

```
      READ (1,2) K,(CIR(II,L),L=1,8)
      READ (1,3) (CIR(II,L),L=9,16)
2     FORMAT (I5,8(2X,F7.3))
3     FORMAT (5X,8(2X,F7.3))
      RETURN
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM PLTDAT

```
Csa2>RUN PLTDAT

 WHAT IS THE BASE 4 LETTER FILE NAME (A4)  NUTS

 HOW MANY FILES ARE IN THE FILE CLASS(I2)?20

 STEP SIZE FOR CONVERSION(I2)1

 INPUT FROM FILE #   1  IN PROGRESS
 INPUT FROM FILE #   2  IN PROGRESS
 INPUT FROM FILE #   3  IN PROGRESS
 INPUT FROM FILE #   4  IN PROGRESS
 INPUT FROM FILE #   5  IN PROGRESS
 INPUT FROM FILE #   6  IN PROGRESS
 INPUT FROM FILE #   7  IN PROGRESS
 INPUT FROM FILE #   8  IN PROGRESS
 INPUT FROM FILE #   9  IN PROGRESS
 INPUT FROM FILE #  10  IN PROGRESS
 INPUT FROM FILE #  11  IN PROGRESS
 INPUT FROM FILE #  12  IN PROGRESS
 INPUT FROM FILE #  13  IN PROGRESS
 INPUT FROM FILE #  14  IN PROGRESS
 INPUT FROM FILE #  15  IN PROGRESS
 INPUT FROM FILE #  16  IN PROGRESS
 INPUT FROM FILE #  17  IN PROGRESS
 INPUT FROM FILE #  18  IN PROGRESS
 INPUT FROM FILE #  19  IN PROGRESS
 INPUT FROM FILE #  20  IN PROGRESS

ENTER MODEL (1=VT125, 2=VK100, 3=VT240, 4=350PC)
3

ENTER MONITOR TYPE (0=MONOCHROME, 1=COLOR)
0

INPUT "D" FOR A DELTA PLOT,"P" FOR A PSI PLOT P

WHAT IS VIEW ANGLE PHI OF THE PLOT(DEG.)?180.

WHAT IS VIEW ANGLE THETA OF THE PLOT(DEG.)?0.0

WHAT IS VIEW RADIUS(MULTIPLE OF X AXIS LENGHT)?20.

WHAT IS THE MAXIMUM DELTA VALUE OF THE PLOT(DEG.)?60.

WHAT IS MINIMUM PSI VALUE OF THE PLOT(DEG.)?30.

DIVISIONS SHOULD BE PLACES EVERY HOW MANY DEGREES?10.

WHAT IS THE WAVELENGTH GRAPH STEP SIZE(I2)?20
```

```
WHAT IS THE FILE GRAPH STEP SIZE(I2)?2

DATA INPUT COMPLETE.  PUSH RETURN FOR PLOTING
WOULD YOU LIKE TO PLOT THE SAME DATA AGAIN(Y/N)?N

FORTRAN STOP
Csa2>
```

PROGRAM PLTDAT VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
ANA(IPASS,N)		-Value of psi (deg) of file IPASS at encoder count N
NAME(I)		-Name of the location and file name of the data to be retrieved.
NAME2		-Experimental file class name
NAME3		-File number of experiment file class NAME2
POL(IPASS,N)		-Value of DELTA (deg) of file IPASS at encoder count N
PHI		-Angle of view of the 3-D plot from the xy plane
RADIUS		-Radial distance from the origin where the view is placed
SCALE		-Scale of the 3-D plot
THETA		-Angle of view of the 3-D plot from the yz plane
X3AXIS		-X scale of the plot
Y3AXIS		-Y scale of the plot
Z3AXIS		-Z scale of the plot

SUBROUTINES CALLED

AREA2D	ISSCO PRODUCTS
ENDPL	ISSCO PRODUCTS
GRAF3D	ISSCO PRODUCTS
NOHIDE	ISSCO PRODUCTS
PAGE	ISSCO PRODUCTS
SURMAT	ISSCO PRODUCTS

TALARS	(LBL EMULATION SUBROUTINE)
VOLM3D	ISSCO PRODUCTS
VUANGL	ISSCO PRODUCTS
X3NAME	ISSCO PRODUCTS
Y3NAME	ISSCO PRODUCTS
Z3NAME	ISSCO PRODUCTS

```
      PROGRAM PLTDAT
C*********************************************************************
C
C       PURPOSE: TO GENERATE THREE DIMENSIONAL PLOTS OF SPECTRAL
C                ELLIPSOMETRY DATA.  DATA HAS TO BE CONVERTED TO
C                FORMATED DATA VIA PROGRAM "DATFIL" ON THE DEC RT11/73
C                AND THEN TRANSFERED TO THE LBL CSA COMPUTER VIA
C                "KERMIT".
C
C       WRITTEN BY STEVE T. MAYER  SPRING 1985
C
C       NOTE: A NUMBER OF SUBROUTINES CALL IN THIS PROGRAM ARE ISSCO
C             PRODUCT ROUTINES FROM THEIR PLOTING PACKAGE "DISSPLA".
C*********************************************************************
      REAL*4 NAME(20),NAME1,NAME2,NAME3
      REAL POL(20,170),ANA(20,170),ZBOT(20,170)
      INTEGER WSSS,FSSS
    1 TYPE 1000
      ACCEPT 1001,NAME2
      TYPE 1002
      ACCEPT 1004,IMAX
      TYPE 1003
      ACCEPT 1004,ISTP
      DO 2 NPASS=1,IMAX,ISTP
      IPASS=NPASS
      NAME(1)='[S10'
      NAME(2)='754.'
      NAME(3)='DAT]'
      NAME(4)=NAME2
      CALL PLAC2(IPASS,NAME3)
      NAME(5)=NAME3
      TYPE 1006,IPASS
      OPEN (UNIT=2,FILE=NAME,TYPE='OLD',READONLY)
      DO 3,N=1,170,5
      READ(2,1007),I,POL(IPASS,N),POL(IPASS,N+1),POL(IPASS,N+2)
     *,POL(IPASS,N+3),POL(IPASS,N+4)
    3 CONTINUE
      DO 4 N=1,170,5
      READ (2,1007),I,ANA(IPASS,N),ANA(IPASS,N+1),ANA(IPASS,N+2)
     *,ANA(IPASS,N+3),ANA(IPASS,N+4)
    4 CONTINUE
      CLOSE (UNIT=2)
    2 CONTINUE
    5 CONTINUE
      SCALE=1.
      X3AXIS=SCALE*7.62
      Y3AXIS=SCALE*7.62
      Z3AXIS=Y3AXIS
      XPAGE=SCALE*7.62
```

```
      YPAGE=SCALE*7.62
      X3MAX=IMAX
      X3MIN=1.
      X3STP=5.
      Y3ORIG=1.
      Y3MAX=170.
      Y3STP=33.
C***********************************************************************
C       PLACE PLOTING DATA INTO A FILE WHICH IS SUITABLE FOR PLOTING
C       ON THE TALARIS PRINTER/PLOTER.  OTHER PLOTERS CAN BE USED.
C       SEE LBL DOCUMENTATION "DISSPLA.DOC" ON THE 8600 COMPUTERS.
C***********************************************************************
      CALL TALARS
    6 CONTINUE
      TYPE 1008
      ACCEPT 100,ANS
      IF (ANS.EQ.'D') GOTO 2000
      IF (ANS.EQ.'P') GOTO 2500
      GOTO 5
2000  CONTINUE
      TYPE 1009
      ACCEPT 1012,PHI
      TYPE 1010
      ACCEPT 1012,THETA
      TYPE 1011
      ACCEPT 1012,RADIUS
      TYPE 1013
      ACCEPT 1012,Z3MAX
      TYPE 1014
      ACCEPT 1012,Z3ORIG
      DO 2100 IF=1,20
      DO 2200 IWL=1,170
      ZBOT(IF,IWL)=Z3ORIG
2200  CONTINUE
2100  CONTINUE
      TYPE 1015
      ACCEPT 1012,Z3STP
      TYPE 2001
      ACCEPT 60,WSSS
      TYPE 2002
      ACCEPT 60,FSSS
      TYPE 1019
      ACCEPT 100,WAIT
      CALL PAGE(XPAGE,YPAGE)
      CALL AREA2D(XPAGE-1.,YPAGE-1.)
      CALL VUANGL(PHI,THETA,RADIUS)
      CALL NOHIDE
      CALL VOLM3D(X3AXIS,Y3AXIS,Z3AXIS)
      CALL X3NAME('FILE NUMBER$',100)
```

```
      CALL Y3NAME('WAVELENGTH$',100)
      CALL Z3NAME('DELTA(DEG)$',100)
      CALL GRAF3D(X3ORIG,X3STP,X3MAX,Y3ORIG,Y3STP,Y3MAX
     *,Z3ORIG,Z3STP,Z3MAX)
      CALL SURMAT(POL,FSSS,20,WSSS,170,0)
      CALL SURMAT(ZBOT,FSSS,20,WSSS,170,0)
      CALL ENDPL(0)
      GOTO 3000
 2500 CONTINUE
      TYPE 1009
      ACCEPT 1012,PHI
      TYPE 1010
      ACCEPT 1012,THETA
      TYPE 1011
      ACCEPT 1012,RADIUS
      TYPE 1013
      ACCEPT 1012,Z3MAX
      TYPE 1020
      ACCEPT 1012,Z3ORIG
      DO 2600 IF=1,20
      DO 2700 IWL=1,170
      ZBOT(IF,IWL)=Z3ORIG
2700  CONTINUE
2600  CONTINUE
      TYPE 1018
      ACCEPT 1012,Z3STP
      TYPE 2001
      ACCEPT 60,WSSS
      TYPE 2002
      ACCEPT 60,FSSS
      TYPE 1019
      ACCEPT 100,WAIT
      CALL PAGE(XPAGE,YPAGE)
      CALL AREA2D(XPAGE-1.,YPAGE-1.)
      CALL VUANGL(PHI,THETA,RADIUS)
      CALL NOHIDE
      CALL VOLM3D(X3AXIS,Y3AXIS,Z3AXIS)
      CALL X3NAME('FILE NUMBER$',100)
      CALL Y3NAME('WAVELENGTH$',100)
      CALL Z3NAME('PSI(DEG)$',100)
      CALL GRAF3D(X3ORIG,X3STP,X3MAX,Y3ORIG,Y3STP,Y3MAX
     *,Z3ORIG,Z3STP,Z3MAX)
      CALL SURMAT(ANA,FSSS,20,WSSS,170,0)
      CALL SURMAT(ZBOT,FSSS,20,WSSS,170,0)
      CALL ENDPL(0)
 3000 CONTINUE
      TYPE 3001
      ACCEPT 100,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 6
```

```
      IF (FLAG2.EQ.'N') GOTO 3500
      GOTO 3000
 3500 CONTINUE
   50 FORMAT(10(A4))
   60 FORMAT(I3)
  100 FORMAT(A1)
 1000 FORMAT(/,'$ WHAT IS THE BASE 4 LETTER FILE NAME (A4)   ')
 1001 FORMAT(A4)
 1002 FORMAT(/,'$ HOW MANY FILES ARE IN THE FILE CLASS(I2)?')
 1003 FORMAT(/,'$ STEP SIZE FOR CONVERSION(I2)')
 1004 FORMAT(I2)
 1006 FORMAT(/,2X,'INPUT FROM FILE #',I4,'  IN PROGRESS')
 1007 FORMAT(I5,5(2X,F7.3))
 1008 FORMAT(/,'$INPUT "D" FOR A DELTA PLOT,"P" FOR A PSI PLOT ')
 1009 FORMAT(/,'$WHAT IS VIEW ANGLE PHI OF THE PLOT(DEG.)?')
 1010 FORMAT(/,'$WHAT IS VIEW ANGLE THETA OF THE PLOT(DEG.)?')
 1011 FORMAT(/,'$WHAT IS VIEW RADIUS(MULTIPLE OF X AXIS LENGHT)?')
 1012 FORMAT(F7.3)
 1013 FORMAT(/,'$WHAT IS THE MAXIMUM DELTA VALUE OF THE PLOT(DEG.)?')
 1014 FORMAT(/,'$WHAT IS THE MINIMUM DELTA VALUE OF THE PLOT(DEG.)?')
 1015 FORMAT(/,'$DIVISIONS SHOULD BE PLACE EVERY HOW MANY DEGREES ?')
 1016 FORMAT(/,'$WHAT IS THE MAXIMUM PSI VALUE OF THE PLOT(DEG.)?')
 1018 FORMAT(/,'$DIVISIONS SHOULD BE PLACES EVERY HOW MANY DEGREES?')
 1019 FORMAT(/,'$DATA INPUT COMPLETE.  PUSH RETURN FOR PLOTING')
 1020 FORMAT(/,'$WHAT IS MINIMUM PSI VALUE OF THE PLOT(DEG.)?')
 2001 FORMAT(/,'$WHAT IS THE WAVELENGTH GRAPH STEP SIZE(I2)?')
 2002 FORMAT(/,'$WHAT IS THE FILE GRAPH STEP SIZE(I2)?')
 3001 FORMAT(/,'$WOULD YOU LIKE TO PLOT THE SAME DATA AGAIN(Y/N)?')
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM SMTH

```
TSX>RUN LD1:SMTH
 WHAT IS THE BASE 4 LETTER FILE NAME (A4)  PR51
 HOW MANY FILES ARE TO BE AVERAGED(I2)?4
 FILE NAME FOR THE NEW SPECTRAL SCAN (A12) = DY1TESTERDAT
SMOOTH ANOTHER FILE CLASS(Y/N)?N
STOP --
```

OUTPUT FILE TESTER.DAT IS
IS THE AVERAGE OF THE FOUR
FILES WHICH COMPRISE THE FILE
CLASS "PR51"

PROGRAM SMTH VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
ANA(I)		-Psi (x100) at encoder count I (deg.)
AAVG(I)		-Average psi value of the IMAX files
IMAX		-File number of the last data file to be read
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NPASS		-File number loop counter
POL(I)		-Delta (x100) at encoder count I (deg.)
PAVG(I)		-Average delta value of the IMAX files

SUBROUTINES CALLED

IN
OUTPUT
WLCALC

```
      PROGRAM SMTH
C*********************************************************************
C     OBJECTIVE: TO AVERAGE CONVERTED DATA FROM ELLIPSOMETER SPECTRAL SCAN
C                INTO AVERAGE PSI, DELTA, AND WAVELENGTH, MADE FROM MANY
C                FILES.
C
C     PROGRAM WRITTEN S. MAYER SPRING 1985.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2,PAVG(400),AAVG(400)
      INTEGER*2 ANGOPT
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      CALL WLCALC
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1001
      ACCEPT 2000,NAME2
      TYPE 1002
      ACCEPT 1004,IMAX
      DO 3 IWV=1,399
      PAVG(IWV)=0
      AAVG(IWV)=0
   3  CONTINUE
      DO 8000 NPASS=1,IMAX
      IPASS=NPASS
      NAME1='DY1C'
      CALL IN
      DO 4 IWV=1,399
      PAVG(IWV)=PAVG(IWV)+POL(IWV)
      AAVG(IWV)=AAVG(IWV)+ANA(IWV)
    4 CONTINUE
8000  CONTINUE
```

```
      DO 5 IWV=1,399
      POL(IWV)=PAVG(IWV)/IMAX
      ANA(IWV)=AAVG(IWV)/IMAX
    5 CONTINUE
      CALL OUTPUT
    6 TYPE 1018
      ACCEPT 100,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 1
      IF (FLAG2.NE.'N') GOTO 6
  100 FORMAT(A1)
 1001 FORMAT(/,'$ WHAT IS THE BASE 4 LETTER FILE NAME (A4)   ')
 1002 FORMAT(/,'$ HOW MANY FILES ARE TO BE AVERAGED(I2)?')
 1004 FORMAT(I2)
 1018 FORMAT(/,'$SMOOTH ANOTHER FILE CLASS(Y/N)?')
 2000 FORMAT(A4)
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM STDDEV

```
TSX>DIR DY1:CGIR1*.*                MULTIPLE SCANS OF THE SAME SURFACE
 29-Jan-86
CGIR10.001      5  24-Jan-85        CGIR10.002      5  24-Jan-85
CGIR10.003      5  24-Jan-85        CGIR10.004      5  24-Jan-85
CGIR10.005      5  24-Jan-85        CGIR10.006      5  24-Jan-85
CGIR10.007      5  24-Jan-85        CGIR10.008      5  24-Jan-85
CGIR10.009      5  24-Jan-85        CGIR10.010      5  24-Jan-85
 10 Files, 50 Blocks
 149 Free blocks

TSX>RUN LD1:STDDEV

 WHAT IS THE BASE 4 LETTER FILE NAME (A4)  GIR1

 HOW MANY FILES ARE TO BE USED IN THE CALCULATION(I2)?10

 FILE NAME FOR THE NEW SPECTRAL SCAN (A12) = DY1TESTERDAT

 DO YOU WANT TO PERFOM CALCULATIONS FOR ANOTHER FILE
CLASS(Y/N)?N

STOP --
```

OUTPUT FILE TESTER.DAT IS A SPECTRAL FILE OF THE STANDARD DEVIATION OF THE MEASUREMENTS

PROGRAM STDDEV VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
ANA(I)		-Psi (x100) at encoder count I (deg.) -Standard deviation of psi of the IMAX files
AAVG(I)		-Average psi value of the IMAX files
IMAX		-File number of the last data file to be read
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC")
NPASS		-File number loop counter
NSTOP		-Encoder count where calculations should stop
POL(I)		-Delta (x100) at encoder count I (deg.) -Standard deviation of delta of the IMAX files
PAVG(I)		-Average delta value of the 4 zone measurement

SUBROUTINES CALLED

IN
OUTPUT
WLCALC

EQUATIONS USED FOR CALCULATIONS

$$\sigma = \frac{1}{N}\sqrt{[\ \beta_j(I) - \ \beta_{avg}(I)]^2}$$

where $\beta_j(I)$ is the value of delta or psi in the jth zone at encoder count I

```
      PROGRAM STDDEV
C*********************************************************************
C        OBJECTIVE: TO COMPUTE THE STANDARD DEVIATION OF A NUMBER OF
C                   OF SPECTRAL FILE.
C
C     PROGRAM WRITTEN S. MAYER SPRING 1985.
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/O/NPASS
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      REAL PAVG(400),AAVG(400),PAVG2(400),AAVG2(400)
      INTEGER*2 ANGOPT
C*****THE ENCODER COUNT IS CONVERTED TO TO WAVELENGTH BY "WLCALC".*****
      CALL WLCALC
C*****RAW SPECTRAL SCAN DATA IS OBTAINED FROM THE FLOPPY DISK.*********
    1 TYPE 1001
      ACCEPT 2000,NAME2
      TYPE 1002
      ACCEPT 1004,IMAX
      DO 3 IWV=1,399
      PAVG(IWV)=0
      AAVG(IWV)=0
      PAVG2(IWV)=0
      AAVG2(IWV)=0
    3 CONTINUE
      DO 8000 NPASS=1,IMAX
      IPASS=NPASS
      NAME1='DY1C'
      CALL IN
      DO 4 IWV=1,399
      PAVG(IWV)=PAVG(IWV)+POL(IWV)
      AAVG(IWV)=AAVG(IWV)+ANA(IWV)
    4 CONTINUE
```

```
8000  CONTINUE
      DO 5 IWV=1,399
      PAVG(IWV)=PAVG(IWV)/IMAX
      AAVG(IWV)=AAVG(IWV)/IMAX
    5 CONTINUE
      DO 110 NPASS=1,IMAX
      IPASS=NPASS
      CALL IN
      DO 111 IWV=1,399
      PAVG2(IWV)=PAVG2(IWV)+(POL(IWV)-PAVG(IWV))**2
      AAVG2(IWV)=AAVG2(IWV)+(ANA(IWV)-AAVG(IWV))**2
 111  CONTINUE
 110  CONTINUE
      DO 112 IWV=1,399
      POL(IWV)=10*SQRT(PAVG2(IWV))/IMAX
      ANA(IWV)=10*SQRT(AAVG2(IWV))/IMAX
 112  CONTINUE
      CALL OUTPUT
    6 TYPE 1018
      ACCEPT 100,FLAG2
      IF (FLAG2.EQ.'Y') GOTO 1
      IF (FLAG2.NE.'N') GOTO 6
  100 FORMAT(A1)
 1001 FORMAT(/,'$ WHAT IS THE BASE 4 LETTER FILE NAME (A4)  ')
 1002 FORMAT(/,'$ HOW MANY FILES ARE TO BE USED IN THE CALCULATION
     *(I2)?')
 1004 FORMAT(I2)
 1018 FORMAT(/,'$DO YOU WANT TO PERFOM CALCULATIONS FOR
     *ANOTHER FILE CLASS(Y/N)?')
 2000 FORMAT(A4)
      STOP
      END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM STEMOD

```
TSX>RUN LD1:STEMOD
WHAT IS THE CRYSTAL FLUX CURRENT DENSITY (MA/CM2)?.2
WHAT IS THE SURFACE CONCENTRATION (MOLE/CM3X10-5)?100.
WHAT IS THE CRYSTAL NUMBER DENSITY(#/CM2X10+8)?3.
WHAT IS THE TOTAL AREA OF THE ELECTRODE(CM**2)?1.
STOP --
TSX>ED LD1:STEMOD.DAT
```

INPUT MODEL PARAMETERS

INSPECT THE OUTPUT DATA FILE "STEMOD.DAT"

```
 0.100    1.000    0.172
 0.200    1.414    0.122
 0.300    1.732    0.099
 0.400    2.000    0.086
 0.500    2.236    0.077
 0.600    2.449    0.070
 0.700    2.646    0.065
 0.800    2.828    0.061
 0.900    3.000    0.058
   .
   .
   .
   .
   .
```

PROGRAM STEMOD VARIABLE EXPLANATIONS

Variable Name	Symbol	EXPLANATION
CS	C_s	-Activity of the solid oxide of infinite radius (moles/cm^3)
CUR(J)	I_t	-Theoretical current at time J (mA)
D	D	-Diffusion coefficient of the soluble oxide species (cm^2/sec)
F	F	-Faraday's constant (96487 C/mole)
IC	I_c	-Crystal oxide incorporation rate (mA/cm^2)
J	t	-Time elapsed from the potential step (sec)
N	n	-Number of electrons transferred in the charge transfer process
NC	N_c	-Number of growing crystals on the surface (#/cm^2)
MW	M	-Molecular weight of the oxide
RHO	ρ	-Density of the oxide (g/cm^3)

SUBROUTINES CALLED

LOOP

EQUATIONS USED FOR CALCULATIONS

$$I_t = I_d + I_c$$

$$I_d = (1-\Omega)\ nF\ C_s \sqrt{\frac{D}{\pi t}}$$

$$\frac{dr}{dt} = \frac{J_{mass}}{\rho} = \frac{Mi_i(t)}{nF\rho}$$

$$i_c(t) = \bar{i}_c\ (1-\Omega)$$

$$I_c(t) = N_c \bar{i}_c\ (1-\Omega)\ 2\pi \left[r(t)^2\right]$$

$$\Omega(t) = N_c\ \pi \left[r(t)^2\right] .$$

$$\frac{\rho nF}{mi_c}\ dr = \left[1 - N_c\ \pi\ (r(t)^2)\right]\ dt$$

$$r = \frac{C(e^{Kt} - 1)}{1+e^{kt}}$$

$$K = \frac{2N_c\ \pi\ CM\bar{i}_c}{nFp}$$

$$C = \frac{1}{\sqrt{N_c \pi}}$$

```
CC      PROGRAM STEMOD
C*****************************************************************
C       PURPOSE: TO CALCULATE THE CURRENT/TIME REPOSNSE TO A POTENTIAL
C                STEP AS GIVEN BY THE A DISSOLUTION/PERCIPITATION MODEL,
C                AND TO PUT THE CALCULATED THEORETICAL CURRENT TRANSIENT
C                INTO A DATA FILE FOR PLOTING.
C
C       WRITTEN BY STEVE T. MAYER JULY 1985
C*******************************************************************
        REAL CUR(400)
        REAL IC,NC,K,N
        MW=232.
        N=1.
        RHO=7.14
        F=96500.
        D=1.0E-5
        TYPE 10
        ACCEPT 100,IC
        IC=IC/1000.
        TYPE 20
        ACCEPT 100,CS
        CS=CS*1.E-8
        TYPE 30
        ACCEPT 100,NC
        type 40
40      FORMAT(/,'$WHAT IS THE TOTAL AREA OF THE ELECTRODE(CM**2)?')
        accept 100,area
        NC=NC*1.E8
        C=1./SQRT(3.1415*NC)
        K=2*3.1415*NC*C*MW*IC/N/F/RHO
        DO 1000,J=1,400
        T=J
        R=C*(EXP(K*T)-1.0)/(1.0+EXP(K*T))
        THETA=NC*3.1415*R**2
        CCUR=NC*IC*(1.0-THETA)*2*3.1415*R**2
        DCUR=(1.-THETA)*N*F*CS*SQRT(D/3.1415/T)
        TCUR=DCUR+CCUR
        A=TCUR/DCUR
        CUR(J)=TCUR*1000*area
1000    CONTINUE
        OPEN (UNIT=2,NAME='STEMOD.DAT',TYPE='NEW',DISP='SAVE')
        DO 23 II=1,400
        TR=1.*II/10.
        TRR=SQRT(TR*10)
        CALL LOOP(TR,TRR,CUR(II))
 23     CONTINUE
 10     FORMAT(/,'$WHAT IS THE CRYSTAL FLUX CURRENT DENSITY (MA/CM2)?')
 20     FORMAT(/,'$WHAT IS THE SURFACE CONCENTRATION (MOLE/CM3X10-5)?')
 30     FORMAT(/,'$WHAT IS THE CRYSTAL NUMBER DENSITY(#/CM2X10+8)?')
```

```
100     FORMAT(F8.3)
        STOP
        END

        SUBROUTINE LOOP(T,SQT,CUR)
        WRITE (2,2) T,SQT,CUR
 2      FORMAT (F7.3,1X,F7.3,1X,F7.3)
        RETURN
        END
```

DEMONSTRATION OF THE EXECUTION OF THE PROGRAM TIMER

```
TSX>RUN LD1:TIMER
 WHAT IS THE FOUR LETTER DATA FILE CODE NAME(A4)?GIR1
 HOW MANY FILE TIMES ARE TO BE FOUND(A2)?10
 DO YOU WANT A HARD COPY(Y/N)?N
 TIMES OF FILES OF EXPERIMENT  GIR1

    FILE#=  1  EXP. ELAPSED TIME=     4.06  SECONDS
    FILE#=  2  EXP. ELAPSED TIME=    15.43  SECONDS
    FILE#=  3  EXP. ELAPSED TIME=    26.93  SECONDS
    FILE#=  4  EXP. ELAPSED TIME=    38.43  SECONDS
    FILE#=  5  EXP. ELAPSED TIME=    49.93  SECONDS
    FILE#=  6  EXP. ELAPSED TIME=    61.52  SECONDS
    FILE#=  7  EXP. ELAPSED TIME=    73.00  SECONDS
    FILE#=  8  EXP. ELAPSED TIME=    84.43  SECONDS
    FILE#=  9  EXP. ELAPSED TIME=    96.00  SECONDS
    FILE#= 10  EXP. ELAPSED TIME=   107.40  SECONDS
 IS THE PROGRAM TO BE EXECUTED AGAIN(Y/N)? N
STOP --
```

PROGRAM TIMER VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
IHRS0		-Hours associated with the start of the first spectral scan (hrs)
IMAX		-File number of the last data file to be converted
IMIN0		-Minutes associated with the start of the first spectral scan (min)
IPASS		-File number loop counter
ISEC0		-Seconds associated with the start of the first spectral scan (sec)
ITIC0		-Tics (1/60 sec) associated with the start of the first spectral scan (tics)
NAME1		-File type identifier
NAME2		-File class name
NAME3		-File number (see subroutine "PLAC2")
T0		-Time (min) of the start of the experiment
T1		-Time (min) of the start of data acquisition of file number IPASS
T2		-Time (min) of the end of data acquisition of file number IPASS
TAVG		-Average time (min) at which file number IPASS was taken
TDIF		-Absolute time difference between the average time when file IPASS was taken, and the start of the experiment

SUBROUTINES CALLED

IN

```
      PROGRAM TIMER
C*********************************************************************
C     PURPOSE: TO CALCULATE THE AVERAGE TIME AT WHICH EACH CONVERTED
C              SPECTRAL FILE WAS TAKEN WITH RESPECT TO THE START OF THE
C              OF THE EXPERIMENT.
C
C     WRITTEN BY STEVEN T. MAYER SUMMER 1984
C
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      COMMON/TITLE/NTITLE(5,30),NWORD(2,21)
      COMMON/MODE/MODEO,MODE1,FACTOR
      DIMENSION ALPHA(35),PWORD(5),AWORD(5),HOLD(30)
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      REAL*4 BDLK(2)
      NPASS=1
    3 TYPE 1009
      ACCEPT 1002,NAME2
      TYPE 1005
      ACCEPT 1003,IMAX
      IPASS=1
      NAME1='DY1C'
      CALL IN
      TO=3600.*IHRSO+IMINO*60.+ISECO+ITICO/60.
      IPASS=0
    4 IPASS=1+IPASS
      CALL CINPUT
      T2=3600.*IHRSF+IMINF*60.+ISECF+ITICF/60.
      T1=3600.*IHRSO+IMINO*60.+ISECO+ITICO/60.
      TAVG=(T1+T2)/2.
      TDIF=TAVG-TO
      HOLD(IPASS)=TDIF
      IF (IPASS.GE.IMAX) GOTO 80
      GOTO 4
   80 CONTINUE
      TYPE 1020
```

```
      ACCEPT 1001,FLAG3
      TYPE 1400,NAME2
      IF (FLAG3.NE.'Y') GOTO 83
      PRINT 1400,NAME2
      PRINT *,' '
      PRINT *,' '
 83   CONTINUE
      DO 5 L=1,IPASS
      TYPE 1500,L,HOLD(L)
      IF (FLAG3.NE.'Y') GOTO 5
      PRINT 1500,L,HOLD(L)
  5   CONTINUE
  6   CONTINUE
      TYPE 1019
      ACCEPT 1001,FLAG2
      IF (FLAG2.EQ.'Y') I=0
      IF(FLAG2.EQ.'Y') GOTO 3
1001  FORMAT(A1)
1002  FORMAT(A4)
1003  FORMAT(I2)
1004  FORMAT(I1)
1005  FORMAT(/,'$ HOW MANY FILE TIMES ARE TO BE FOUND(A2)?')
1009  FORMAT(/,'$ WHAT IS THE FOUR LETTER DATA FILE CODE NAME(A4)?')
1010  FORMAT (/,'$ INSERT THE DISK WITH THE AVERAGE ELAPSED TIME
    C TO BE DETERMINED, THEN "RETURN".')
1019  FORMAT(/,'$ IS THE PROGRAM TO BE EXECUTED AGAIN(Y/N)? ')
1020  FORMAT(/,'$ DO YOU WANT A HARD COPY(Y/N)?')
1400  FORMAT(/,2X,'TIMES OF FILES OF EXPERIMENT',2X,A4)
1500  FORMAT(5X,'FILE#= ',I2,2X,'EXP. ELAPSED TIME=',1X,F8.2,
    C 2X,'SECONDS')
      STOP
      END
```

SUBROUTINE LISTINGS

SUBROUTINE CSM2 VARIABLES EXPLANATION

Variable Name	SYMBOL	EXPLANATION
CPHI2	$\hat{\varphi}_f$	-Complex angle of propagation in the film
CPHI3	$\hat{\varphi}_s$	-Complex angle of propagation in the substrate
PHI	φ_o	-Angle of light propagation in the ambient media (angle of incidence relative to the surface normal)
RP	R_p	-Resultant p polarization reflection coefficient of a film covered surface calculated by the Drude equation
R1P	r_{1p}	-P polarization reflection coefficient of the electrolyte/film interface.
R1S	r_{1p}	-S polarization reflection coefficient of the electrolyte/film interface.
R2P	r_{2p}	-P polarization reflection coefficient of the film/substrate interface.
R2S	r_{2s}	-S polarization reflection coefficient of the film/substrate interface.
SP		sine of the variable PHI
T	T	-Film thickness (A)
TNF	n_f	-Refractive index (real component) of the film
TNKF	K_f	-Extinction coefficient (imaginary component of the complex refractive index) of the film
TNS	n_s	-Refractive index (real component) of the substrate
TNKS	K_s	-Extinction coefficient (imaginary component of the complex refractive index) of the substrate
TN1	n_o	-Refractive index of the ambient

TN2	$\hat{n}_f$	-Complex refractive index of the film
TN3	$\hat{n}_s$	-Complex refractive index of the substrate
WL	λ_o	-Wavelength of light
Z	Z	-Complex optical film path length

CALLED SUBROUTINES

NONE

EQUATIONS USED FOR CALCULATIONS

$$\hat{n}_j = n_j - i\,k_j \qquad j = s,f$$

$$n_o \sin\varphi_o = \hat{n}_f \sin\hat{\varphi}_f = \hat{n}_s \sin\varphi_1$$

$$r_{1s} = \frac{n_o \cos\varphi_o - \hat{n}_f \cos\hat{\varphi}_f}{n_o \cos\varphi_o + \hat{n}_f \cos\hat{\varphi}_f}$$

$$r_{1p} = \frac{-(n_f \cos\hat{\varphi}_f - \hat{n}_f \cos\hat{\varphi}_o)}{n_o \cos\varphi + \hat{n}_f \cos\varphi_o}$$

$$r_{2p} = \frac{\hat{n}_f \cos\hat{\varphi}_f - \hat{n}_s \cos\varphi_s}{\hat{n}_f \cos\hat{\varphi}_f + \hat{n}_s \cos\varphi_s}$$

$$r_{2s} = \frac{-\left(\hat{n}_f \cos\hat{\varphi}_s - \hat{n}_s \cos\varphi_f\right)}{\hat{n}_f \cos\varphi_s + \hat{n}_s \cos\varphi_f}$$

$$Z = \frac{4\pi T}{\lambda_o} \hat{n}_f \cos\hat{\varphi}_f$$

$$R_s = \frac{r_{1s} + r_{2s} \exp^{-iZ}}{1 + r_{1s} r_{2s} \exp^{-iZ}}$$

$$R_p = \frac{r_{1p} + r_{2p} \exp^{-iZ}}{1 + r_{1p} r_{2p} \exp^{-iZ}}$$

```
      SUBROUTINE CSM2(TN1,TNF,TNKF,TNS,TNKS,WL,T,RS,RP)
C*********************************************************************
C
C     PURPOSE: TO USE THE DRUDE EXACTION TO FIND THE REFLECTION
C              COEFFICENTS FROM A COMPONENT ON A SUBRTRATE FOR
C              USE IN AN ISLAND FILM ON A PREDEFINDED SUBSTRATE,
C              WHICH MAY ITSELF BE A COMPLEX FILM.
C
C      SUBROUTINE DEVELOPED BY S. MAYER SUMMER 1984
C
C*********************************************************************
      COMMON/FLM/DELC,PSIC,PHI
      COMPLEX TN2,TN3,CPHI2,CPHI3,R1S,R1P,R2S,R2P,Z,RS,RP,RHO
      CP=COS(PHI)
      SP=SIN(PHI)
      TN3=CMPLX(TNS,-TNKS)
      CPHI3=CSQRT(1.0-TN1**2*SP**2/(TN3**2))
      TN2=CMPLX(TNF,-TNKF)
      CPHI2=CSQRT(1.0-TN1**2*SP**2/(TN2**2))
      R1S=(TN1*CP-TN2*CPHI2)/(TN1*CP+TN2*CPHI2)
      R1P=-(TN1*CPHI2-TN2*CP)/(TN1*CPHI2+TN2*CP)
      R2S=(TN2*CPHI2-TN3*CPHI3)/(TN2*CPHI2+TN3*CPHI3)

      R2P=-(TN2*CPHI3-TN3*CPHI2)/(TN2*CPHI3+TN3*CPHI2)
      Z=(0.0,1.0)*(4.0*3.1415927*T/WL)*TN2*CPHI2
      RS=(R1S+R2S*CEXP(-Z))/(1.0+R1S*R2S*CEXP(-Z))
      RP=(R1P+R2P*CEXP(-Z))/(1.0+R1P*R2P*CEXP(-Z))
      RETURN
      END
```

```
      SUBROUTINE DELAY
C*********************************************************************
C     OBJECTIVE: TO GENERATE A 15 SECOND TIME DELAY FOR SUBROUTINE
C                "RATE".
C*********************************************************************
      INTEGER*4 ITIME
   10 CALL GTIM(ITIME)
      CALL CVTTIM(ITIME,IHRS,IMIN,ISEC,ITIC)
      ISECO=ISEC
      IF(ISECO.GT.44)GOTO 10
   20 CALL GTIM(ITIME)
      CALL CVTTIM(ITIME,IHRS,IMIN,ISEC,ITIC)
      ISECF=ISEC
      IF(ISECF.EQ.ISECO+15)RETURN
      GOTO 20
      END
```

SUBROUTINE DELPSI VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
A		-Calculated analyzer azimuth (deg)
D		-Delta (deg)
C		-Compensator position (deg)
P		-Calculated polarizer azimuth (deg)

SUBROUTINES CALLED

NONE

EQUATIONS USED

I

Derivation of ψ and Δ from ellipsometer azimuths of polarizer, quarter wave plate, and analyzer (all angles in degrees).
($\psi = 0°$ to $90°$, $\Delta = 0$ to $360°$)

Zone	Range of polarizer transmission azimuth p	Compensator fast axis azimuth q	Range of analyzer transmission azimuth a	ψ	Δ
A-1	0–45	45	90–180	180 − a	90 − 2p
A-2	45–90	135	90–180	180 − a	2p − 90
A-3	90–135	45	0–90	a	270 − 2p
A-4	135–180	135	0–90	a	2p − 270
B-1	0–45	135	0–90	a	90 + 2p
B-2	45–90	45	0–90	a	270 − 2p
B-3	90–135	135	90–180	180 − a	2p − 90
B-4	135–180	45	90–180	180 − a	450 − 2p
C-1	0–45	45	0–90	a	270 − 2p
C-2	45–90	135	0–90	a	90 + 2p
C-3	90–135	45	90–180	180 − a	450 − 2p
C-4	135–180	135	90–180	180 − a	2p − 90
D-1	0–45	135	90–180	180 − a	270 + 2p
D-2	45–90	45	90–180	180 − a	450 − 2p
D-3	90–135	135	0–90	a	90 + 2p
D-4	135–180	45	0–90	a	630 − 2p

```
      SUBROUTINE DELPSI(P,C,A,D,S,ANGOPT)
C*********************************************************************
C     OBJECTIVE: THIS SUBROUTINE, GIVEN THE POLARIZER, COMPENSATOR, AND
C                ANALYZER AZIMUTH COMPUTES THE CORRESPONDING PSI AND
C                DELTA ELLIPSOMETER ANGLES USING EITHER STANDARD OR
C                ROTATED CONVERSION FORMULAS(DEPENDING UPON THE
C                VALUE OF "ANGOPT".  IF ANGOPT=1 STANDARD FORMULAS
C                ARE USED; IF ANGOPT=2 ROTATED FORMULAS ARE USED.
C*********************************************************************
      INTEGER*2 PFLAG,AFLAG,CFLAG,FLAG,ANGOPT
      D=0.
      S=0.
      PFLAG=0
      GOTO 5
    4 P=P+180.
    5 CONTINUE
      IF(P.GE.180.) P=P-180.
      IF(P.LE.180.)PFLAG=400
      IF(P.LE.135.)PFLAG=300
      IF(P.LE. 90.)PFLAG=200
      IF(P.LE. 45.)PFLAG=100
      IF(P.LE.  0.)GOTO 4
    6 CONTINUE
      AFLAG=0
      GOTO 8
    7 A=A+180.
    8 CONTINUE
      IF(A.GT.180.) A=A-180.
      IF(A.LE.180.)AFLAG=20
      IF(A.LE. 90.)AFLAG=10
      IF(A.LT.  0.)GOTO 7
      CFLAG=0
      IF(C.EQ.45.)CFLAG=1
      IF(C.EQ.135.)CFLAG=2
      FLAG=PFLAG+AFLAG+CFLAG
      IF(ANGOPT.NE.2)GOTO 180
      IF(FLAG.NE.312)GOTO 10
      D=270.-2.*P
      S=90.-A
   10 CONTINUE
      IF(FLAG.NE.411)GOTO 20
      D=2.*P-270.
      S=90.-A
   20 CONTINUE
      IF(FLAG.NE.122)GOTO 30
      D=90.-2.*P
      S=A-90.
   30 CONTINUE
      IF(FLAG.NE.221)GOTO 40
```

```
      D=2.*P-90.
      S=A-90.
   40 CONTINUE
      IF(FLAG.NE.321)GOTO 50
      D=2.*P-90.
      S=A-90.
   50 CONTINUE
      IF(FLAG.NE.422)GOTO 60
      D=450.-2.*P
      S=A-90.
   60 CONTINUE
      IF(FLAG.NE.111)GOTO 70
      D=2.*P-90.
      S=90.-A
   70 CONTINUE
      IF(FLAG.NE.212)GOTO 80
      D=270.-2.*P
      S=90.-A
   80 CONTINUE
      IF(FLAG.NE.322)GOTO 90
      D=450.-2.*P
      S=A-90.
   90 CONTINUE
      IF(FLAG.NE.421)GOTO 100
      D=2.*P-90.
      S=A-90.
  100 CONTINUE
      IF(FLAG.NE.112)GOTO 110
      D=270.-2.*P
      S=90.-A
  110 CONTINUE
      IF(FLAG.NE.211)GOTO 120
      D=2.*P+90.
      S=90.-A
  120 CONTINUE
      IF(FLAG.NE.311)GOTO 130
      D=2.*P+90.
      S=90.-A
  130 CONTINUE
      IF(FLAG.NE.412)GOTO 140
      D=630.-2.*P
      S=90.-A
  140 CONTINUE
      IF(FLAG.NE.121)GOTO 150
      D=2.*P+270.
      S=A-90.
  150 CONTINUE
      IF(FLAG.NE.222)GOTO 160
      D=450.-2.*P
```

```
      S=A-90.
  160 CONTINUE
      IF(D.EQ.0.0.AND.S.EQ.0.0)TYPE 170
  170 FORMAT(/,1X,'*PROBABLE ELLIPSOMETER ANGLE CONVERSION ERROR IN
     C SUBROUTINE "PSIDEL" DETECTED;',/,' EXECUTION TERMINATED*')
      GOTO 1180
  180 CONTINUE
      IF(ANGOPT.NE.1)GOTO 1180
      IF(FLAG.NE.121)GOTO 1010
      D=90.-2.*P
      S=180.-A
 1010 CONTINUE
      IF(FLAG.NE.222)GOTO 1020
      D=2.*P-90.
      S=180.-A
 1020 CONTINUE
      IF(FLAG.NE.311)GOTO 1030
      D=270.-2.*P
      S=A
 1030 CONTINUE
      IF(FLAG.NE.412)GOTO 1040
      D=2.*P-270.
      S=A
 1040 CONTINUE
      IF(FLAG.NE.112)GOTO 1050
      D=90.+2.*P
      S=A
 1050 CONTINUE
      IF(FLAG.NE.211)GOTO 1060
      D=270.-2.*P
      S=A
 1060 CONTINUE
      IF(FLAG.NE.322)GOTO 1070
      D=2.*P-90.
      S=180.-A
 1070 CONTINUE
      IF(FLAG.NE.421)GOTO 1080
      D=450.-2.*P
      S=180.-A
 1080 CONTINUE
      IF(FLAG.NE.111)GOTO 1090
      D=270.-2.*P
      S=A
 1090 CONTINUE
      IF(FLAG.NE.212)GOTO 1100
      D=2.*P+90.
      S=A
 1100 CONTINUE
      IF(FLAG.NE.321)GOTO 1110
```

```
      D=450.-2.*P
      S=180.-A
 1110 CONTINUE
      IF(FLAG.NE.422)GOTO 1120
      D=2.*P-90.
      S=180.-A
 1120 CONTINUE
      IF(FLAG.NE.122)GOTO 1130
      D=2.*P+270.
      S=180.-A
 1130 CONTINUE
      IF(FLAG.NE.221)GOTO 1140
      D=450.-2.*P
      S=180.-A
 1140 CONTINUE
      IF(FLAG.NE.312)GOTO 1150
      D=2.*P+90.
      S=A
 1150 CONTINUE
      IF(FLAG.NE.411)GOTO 1160
      D=630.-2.*P
      S=A
 1160 CONTINUE
      IF(D.EQ.0.0.AND.S.EQ.0.0)TYPE 170
      IF(D.EQ.0.0.AND.S.EQ.0.0)STOP
 1180 CONTINUE
      IF(ANGOPT.EQ.1.OR.ANGOPT.EQ.2)GOTO 2010
      TYPE 2000
 2000 FORMAT(/,1X,'*INCORRECT OPTION CODE SPECIFIED IN SUBROUTINE
     C "PSIDEL" SO EXECUTION WAS TERMINATED*')
 2010 CONTINUE
      IF (D.LE.0) D=D+180.
      IF (A.LE.0) A=A+180.
      RETURN
      END
```

PROGRAM EDATA VARIABLE EXPLANATION

Variable Name	Symbol	EXPLANATION
ANA(I)		-Digital current level of the analyzer at encoder count I (ACDU)
ANM		-Maximum value of the analyser current for a given scan
DEVICE		-Device name send to subroutine "STEPER"
IHRSF		-Hours associated with the end of the spectral scan (hrs)
IHRSO		-Hours associated with the start of the spectral scan (hrs)
IMAX		-File number of the last data file to be converted
IMINF		-Minutes associated with the end of the spectral scan (min)
IMINO		-Minutes associated with the start of the spectral scan (min)
ISECF		-Seconds associated with the end of the spectral scan (sec)
ISECO		-Seconds associated with the start of the spectral scan (sec)
ITICF		-Tics (1/60 sec) associated with the end of the spectral scan (tics)
ITICO		-Tics (1/60 sec) associated with the start of the spectral scan (tics)
PLM		-Maximum value of the polarizer current for a given scan
POL(I)		-Digital current level of the polarizer at encoder count I (ACDU)

SUBROUTINES CALLED

CVTTIM (SYSTEM SUBROUTINE)
GTIM (SYSTEM SUBROUTINE)

```
      SUBROUTINE EDATA
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/Z/NAME1,NAME2,NPASS,DURAT,RTIME
      COMMON/STEP/DEVICE,ISTEP,IPC,IAC
      INTEGER*4 ITIMEO,ITIMEF,LTIME
      INTEGER NPASS
      CALL IPOKE("170400,0)
      IGAIN=3
      JGAIN=IGAIN*64
      N=1
      DO 100 I=1,400
      POL(I)=0.
      ANA(I)=0.
      NAVG(I)=0
  100 CONTINUE
      CALL GTIM(TIMEO)
      CALL IPOKE("164400,0)
 1000 IF(IPEEK("164400).LE.0)GOTO 1000
      CALL IPOKE("164404,1)
 2000 IO=IPEEK("164404)
 2500 CONTINUE
      CALL IPOKE("170402,IPOL.OR.(JGAIN))
 3000 IF(.NOT.(IPEEK("170400).AND."100000))GOTO 3000
      POL(N)=POL(N)+IPEEK("170402)
      CALL IPOKE("170402,IANA.OR.(JGAIN))
 4000 IF(.NOT.(IPEEK("170400).AND."100000))GOTO 4000
      ANA(N)=ANA(N)+IPEEK("170402)
      NAVG(N)=NAVG(N)+1
      IF(IPEEK("164404).EQ.IO)GOTO 2500
      N=N+1
      IF(N.LE.399)GOTO 2000
      N=1
 4500 IF(IPEEK("164400).EQ.I1)GOTO 4500
      I1=IPEEK("164400)
      IF(I1.LT.NSCAN)GOTO 2000
      CALL GTIM(TIMEF)
      CALL CVTTIM(TIMEO,IHRSO,IMINO,ISECO,ITICO)
      CALL CVTTIM(TIMEF,IHRSF,IMINF,ISECF,ITICF)
      DO 6000 I=1,399
      IF(NAVG(I).EQ.0)GOTO 5000
      POL(I)=POL(I)/NAVG(I)
      ANA(I)=ANA(I)/NAVG(I)
      GOTO 6000
 5000 POL(I)=0.
      ANA(I)=0.
 6000 CONTINUE
      SUM=0.
```

```
      IPC=1
      IAC=1
      PLM=ABS(POL(1))
      ANM=ABS(ANA(1))
      DO 7000 I=1,399
      IF(PLM.GT.ABS(POL(I))) GOTO 6500
      PLM=ABS(POL(I))
      IPC=I
 6500 CONTINUE
      IF (ANM.GT.ABS(ANA(I))) GOTO 7000
      ANM=ABS(ANA(I))
      IAC=I
 7000 SUM=SUM+NAVG(I)
      SUM=(SUM/399.)
      TYPE 8000,SUM
 8000 FORMAT(/,'  EACH POINT AN AVERAGE OF: ',F5.0,' CHANNEL SCANS')
      RETURN
      END
```

SUBROUTINE EMA VARIABLE EXPLANATIONS

Variable Name	SYMBOL	EXPLANATION
E	ε	-Complex dielectric constant of the effective media
E1	ε_i	-Complex dielectric constant of the first composite film material (the volume fraction is based on this component)
E2	ε_j	-Complex dielectric constant of the second composite film material
THETA	Θ	-Volume fraction of component 1 in the composite film
TN1	n_i	-Refractive index (real component) of the first composite film material
TNK1	K_i	-Extinction coefficient (imaginary component of the first composite film material
TN2	n_j	-Refractive index (real component) of the second composite film material
TNK2	K_j	-Extinction coefficient (imaginary component of the second composite film material
TNFE	n_ε	-Effective real component of the composite media refractive index
TNKFE	K_ε	-Effective extinction coefficient of the composite media

CALLED SUBROUTINES

NONE

EQUATIONS USED FOR CALCULATIONS

$$\frac{\varepsilon - \varepsilon_h}{\varepsilon - \varepsilon_h} = \Theta_i \frac{\varepsilon_i - \varepsilon_h}{\varepsilon_i + 2\varepsilon_h} + (1-\Theta_i) \frac{\varepsilon_j - \varepsilon_h}{\varepsilon_i + 2\varepsilon_h}$$

For the Bruggeman Model

$$\varepsilon_h = \varepsilon.$$

And so this reduces to

$$\varepsilon = \tfrac{1}{2} \left(-A + \sqrt{A^2 - 4B}\right)$$

with

$$A = \tfrac{1}{2} \; \varepsilon_i(1 - 3\Theta) + \varepsilon_j \; (3\Theta - 2)$$

$$B = \tfrac{1}{2} \; \varepsilon_j \varepsilon_i .$$

Also $\hat{n}_\varepsilon = \sqrt{\varepsilon} \quad = n_\varepsilon - iK_\varepsilon .$

And $\hat{n}_K = n_K - iK_K \qquad K = i,j$

$$\varepsilon_K = \hat{n}_K^{\,2} .$$

```
      SUBROUTINE EMA (TN2,TNK2,TN1,TNK1,THETA,TNFE,TNKFE)
C*********************************************************************
C     COMPUTE THE EFFECTIVE MEDIA REFRACTIVE INDEX OF A FILM
C     FOR A POROUS FILM BY THE EFFECTIVE MEDIA APPROXIMATION
C     (EMA) GIVEN THE REF. INDEX OF THE INCIDENT AND SOLUTION
C     REF. INDEX, AND THE VOLUME FRACTION OF THE FILM
C
C     SUBROUTINE WRITTEN BY JOE FARMER
C*********************************************************************
      COMPLEX E,E1,E2,A,B,TN4
      E2=CMPLX(TN2,-TNK2)
      E1=CMPLX(TN1,-TNK1)
      E1=E1*E1
      E2=E2*E2
      X1=THETA
      X2=1.-THETA
      A=0.5*(E1*(X2-2.*X1)+E2*(X1-2.*X2))
      B=-0.5*(E1*E2)
      E=0.5*(-A+CSQRT(A*A-4.*B))
      TN4=CSQRT(E)
      TNFE=REAL(TN4)
      TNKFE=ABS(AIMAG(TN4))
      RETURN
      END
```

```
      SUBROUTINE FCREAD
C*********************************************************************
C     OBJECTIVE: TO RETRIEVE ONLY THE VERDET COEFFICIENTS STORED BY
C                THE FARADAY CELL CALIBRATION ROUTINE "CALFC1",
C                DISREGARDING OTHER INFORMATION STORED ON THE
C                DISK WITH THE VERDET COEFFICIENTS.  THIS
C                INFORMATION IS USED BY "DSCONV" TO CONVERT
C                A/D CONVERTOR READINGS TO POLARIZER AND
C                ANALYZER AZIMUTHS PRIOR TO CALLING
C                SUBROUTINE "DELPSI" FOR AZIMUTH
C                CONVERSION TO PSI AND DELTA.
C*********************************************************************
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/QBRIEF/ASLOPE(400),PSLOPE(400)
      REAL*4 FCBDLK(2),NAME(20)
      IBLOCK=0
      FACTO1=1.0E05
      NAME(1)='DY1F'
      NAME(2)='ARAD'
      NAME(3)='YDAT'
    2 FORMAT(3A4)
      N=IRAD50(12,NAME,FCBDLK)
      ICHAN=IGETC
      IF(ICHAN.LT.0)STOP 'CANNOT ALLOCATE CHANNEL'
      IF(IFETCH(ICHAN).LT.0)STOP 'FETCH FAIL'
      IF(LOOKUP(ICHAN,FCBDLK).LT.0)STOP 'BAD LOOKUP'
      CALL READ
      CALL READ
      DO 40 I=1,200
   40 PSLOPE(I)=IBUFF(I)/FACTO1
      CALL READ
      DO 50 I=1,200
   50 ASLOPE(I)=IBUFF(I)/FACTO1
      CALL READ
      DO 60 I=1,200
   60 PSLOPE(I+200)=IBUFF(I)/FACTO1
      CALL READ
      DO 70 I=1,200
   70 ASLOPE(I+200)=IBUFF(I)/FACTO1
      CALL CLOSEC(ICHAN)
      CALL IFREEC(ICHAN)
      RETURN
      END
```

SUBROUTINE FILM2 VARIABLES EXPLANATION

Variable Name	SYMBOL	EXPLANATION
CPHI2	$\hat{\varphi}_f$	-Complex angle of propagation in the film
CPHI3	$\hat{\varphi}_s$	-Complex angle of propagation in the substrate
DELC	Δ	-Calculated value of delta for the film covered surface (deg)
PHI	φ_o	-Angle of light propagation in the ambient media (angle of incidence relative to the surface normal)
PSIC	ψ	-Calculated value of psi for the film covered surface (deg)
RHO	ρ	-Reflection coefficient of the film covered surface
RP	R_p	-Resultant p polarization reflection coefficent of a film covered surface calculated by the Drude equation
R1P	r_{1p}	-P polarization reflection coefficient of the electrolyte/film interface.
R1S	r_{1s}	-S polarization reflection coefficient of the electrolyte/film interface.
R2P	r_{2p}	-P polarization reflection coefficient of the film/substrate interface.
R2S	r_{2s}	-S polarization reflection coefficient of the film/substrate interface.
SP		-Sine of the variable PHI
T	T	-Film thickness (Å)
TNF	n_f	-Refractive index (real component) of the film
TNKF	K_f	-Extinction coefficient (imaginary component of the complex refractive index) of

		the film
TNS	n_s	-Refractive index (real component) of the substrate
TNKS	K_s	-Extinction coefficient (imaginary component of the complex refractive index) of the substrate
TN1	n_o	-Refractive index of the ambient
TN2	$\hat{n}_f$	-Complex refractive index of the film
TN3	$\hat{n}_s$	-Complex refractive index of the substrate
WL	λ_o	-Wavelength of light
Z	Z	-Complex optical film path length

CALLED SUBROUTINES

NONE

EQUATIONS USED FOR CALCULATIONS

$$\hat{n}_j = n_j - iK_j \qquad j = s_1 f$$

$$n_o \sin\varphi_o = \hat{n}_f \sin\hat{\varphi}_f = \hat{n}_s \sin\varphi_s$$

$$r_{1s} = \frac{n_o \cos\varphi_o - \hat{n}_f \cos\hat{\varphi}_f}{n_o \cos\hat{\varphi}_o + \hat{n}_f \hat{\varphi}_f}$$

$$r_{1p} = -\frac{n_o \cos\hat{\varphi}_f - \hat{n}_f \cos\varphi_o}{n_o \cos\hat{\varphi}_f + \hat{n}_f \cos\varphi_o}$$

$$r_{2p} = \frac{\hat{n}_f \cos\hat{\varphi}_f - \hat{n}_s \cos\hat{\varphi}_s}{\hat{n}_f \cos\hat{\varphi}_f + \hat{n}_s \cos\hat{\varphi}_s}$$

$$r_{2s} = -\frac{\hat{n}_f \cos\hat{\varphi}_s - \hat{n}_s \cos\hat{\varphi}_f}{\hat{n}_f \cos\hat{\varphi}_s + \hat{n}_s \cos\hat{\varphi}_f}$$

$$Z = \frac{4\pi T}{\lambda_o} \hat{n}_f \cos\hat{\varphi}_f$$

$$R_s = \frac{r_{1s} + r_{2s} \exp^{-iZ}}{1 + r_{1s} r_{2s} \exp^{-iZ}}$$

$$R_p + \frac{r_{1p} + r_{2p} \exp^{-iZ}}{1 + r_{1p} r_{2p} \exp^{-iZ}}$$

$$\rho = \psi \exp(-i\Delta)$$

```
      SUBROUTINE FILM2(N1,K1,TNF,TNKF,TNS,TNKS,WL,T)
C*********************************************************************
C     THIS SUBROUTINE COMPUTES "PSI" AND "DELTA" FOR A SIMPLE HOMO-
C     GENEOUS FILM ON A SUBSTRATE GIVEN (1) THE REFRACTIVE INDICES OF
C     THE SUBSTRATE, FILM, AND INCIDENT MEDIUM; (2) THE FILM THICKNESS;
C     (3) THE WAVELENGTH OF THE SOURCE; AND (4) THE ANGLE OF INCIDENCE.
C*********************************************************************
      COMMON/FLM/DELC,PSIC,PHI
      COMPLEX TN2,TN3,CPHI2,CPHI3,R1S,R1P,R2S,R2P,Z,RS,RP,RHO
      COMPLEX CP,SP
      ANG=360.00000
      ANG2=180.00000
      CP=COS(PHI)
      SP=SIN(PHI)
      TN1=CMPLX(N1,-K1)
      TN3=CMPLX(TNS,-TNKS)
      CPHI3=CSQRT(1.0-TN1**2*SP**2/(TN3**2))
      TN2=CMPLX(TNF,-TNKF)
      CPHI2=CSQRT(1.0-TN1**2*SP**2/(TN2**2))
      R1S=(TN1*CP-TN2*CPHI2)/(TN1*CP+TN2*CPHI2)
      R1P=-(TN1*CPHI2-TN2*CP)/(TN1*CPHI2+TN2*CP)
      R2S=(TN2*CPHI2-TN3*CPHI3)/(TN2*CPHI2+TN3*CPHI3)
      R2P=-(TN2*CPHI3-TN3*CPHI2)/(TN2*CPHI3+TN3*CPHI2)
      Z=(0.0,1.0)*(4.0*3.1415927*T/WL)*TN2*CPHI2
      RS=(R1S+R2S*CEXP(-Z))/(1.0+R1S*R2S*CEXP(-Z))
      RP=(R1P+R2P*CEXP(-Z))/(1.0+R1P*R2P*CEXP(-Z))
      RHO=RP/RS
      PSIC=DATAN(CABS(RHO))/0.01745329252
      DELC=DATAN2(AIMAG(RHO),REAL(RHO))/0.01745329252
      IF (DELC.LT.0.0) DELC=ANG+DELC
      IF (PSIC.LT.0.0) PSIC=ANG2+PSIC
      RETURN
      END
```

```
SUBROUTINE FINISH
COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
IFLAG=IFLAG-1
RETURN
END
```

```
      SUBROUTINE IN
C*********************************************************************
C     OBJECTIVE: TO READ THE SETUP INFORMATION AND SPECTRAL SCAN DATA
C                AS FROM THE DEVICE AND OF THE NAME DICTATED BY THE
C                PROGRAM.
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      REAL*4 BDLK(2)
      IBLOCK=0
      NAME(1)=NAME1
      NAME(2)=NAME2
      CALL PLAC(IPASS,NAME(3))
  200 FORMAT(3A4)
      N=IRAD50(12,NAME,BDLK)
      ICHAN=IGETC(I)
      IF(ICHAN.LT.0)STOP 'CANNOT ALLOCATE CHANNEL'
      IF(IFETCH(ICHAN).LT.0)STOP 'FETCH FAIL'
      IF(LOOKUP(ICHAN,BDLK).LT.0)STOP 'BAD LOOKUP'
      DO 10 I=1,256
   10 IBUFF(I)=0
      CALL READ
      DO 20 I=1,20
      NAMEOP(I)=IBUFF(I)
   20 ID(I)=IBUFF(20+I)
      LMPTYP=IBUFF(41)
      DO 30 I=1,9
   30 LMPSER(I)=IBUFF(41+I)
      LMPVLT=IBUFF(51)
      LMPAMP=IBUFF(52)
      IPHASP=IBUFF(54)
      IGAINP=IBUFF(55)
      IAMPLP=IBUFF(56)
      ITIMEP=IBUFF(57)
      IPHASA=IBUFF(58)
      IGAINA=IBUFF(59)
      IAMPLA=IBUFF(60)
```

```
      ITIMEA=IBUFF(61)
      IGAGP =IBUFF(62)
      IGATP =IBUFF(63)
      IGAGA =IBUFF(64)
      IGATA =IBUFF(65)
      NRATE =IBUFF(66)
      NSCAN =IBUFF(67)
      IPOL  =IBUFF(68)
      IANA  =IBUFF(69)
      NWL   =IBUFF(70)
      NPOL  =IBUFF(71)
      NCMP  =IBUFF(72)
      NANA  =IBUFF(73)
      IHRSO =IBUFF(74)
      IHRSF =IBUFF(75)
      IMINO =IBUFF(76)
      IMINF =IBUFF(77)
      ISECO =IBUFF(78)
      ISECF =IBUFF(79)
      ITICO =IBUFF(80)
      ITICF =IBUFF(81)
      IDAY  =IBUFF(82)
      IYRS  =IBUFF(83)
      LMPHRS=IBUFF(84)
      IMON(1)=IBUFF(85)
      IMON(2)=IBUFF(86)
      IMON(3)=IBUFF(87)
      DO 35 I=1,10
   35 IPMTYP(I)=IBUFF(87+I)
      IPMTDV=IBUFF(98)
      CALL READ
      DO 40 I=1,200
   40 POL(I)=IBUFF(I)
      CALL READ
      DO 50 I=1,200
   50 ANA(I)=IBUFF(I)
      CALL READ
      DO 60 I=1,200
   60 POL(I+200)=IBUFF(I)
      CALL READ
      DO 70 I=1,200
   70 ANA(I+200)=IBUFF(I)
      CALL CLOSEC(ICHAN)
      CALL IFREEC(ICHAN)
      RETURN
      END
```

```
      SUBROUTINE INPUT
C*********************************************************************
C     OBJECTIVE: TO READ THE SETUP INFORMATION AND SPECTRAL SCAN DATA
C                FROM ANY FLOPPY DISK FILE SPECIFIED BY THE OPERATOR
C                AND CREATED BY SUBROUTINE "OUTPUT".
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/WER/NAME
      REAL*4 BDLK(2),NAME(20)
      IBLOCK=0
      READ(1,200)(NAME(I),I=1,3)
  200 FORMAT(3A4)
      IF(NAME(1).EQ.4HSTOP)RETURN
      N=IRAD50(12,NAME,BDLK)
      ICHAN=IGETC(I)
      IF(ICHAN.LT.0)STOP 'CANNOT ALLOCATE CHANNEL'
      IF(IFETCH(ICHAN).LT.0)STOP 'FETCH FAIL'
      IF(LOOKUP(ICHAN,BDLK).LT.0)STOP 'BAD LOOKUP'
      DO 10 I=1,256
   10 IBUFF(I)=0
      CALL READ
      DO 20 I=1,20
      NAMEOP(I)=IBUFF(I)
   20 ID(I)=IBUFF(20+I)
      LMPTYP=IBUFF(41)
      DO 30 I=1,9
   30 LMPSER(I)=IBUFF(41+I)
      LMPVLT=IBUFF(51)
      LMPAMP=IBUFF(52)
      IPHASP=IBUFF(54)
      IGAINP=IBUFF(55)
      IAMPLP=IBUFF(56)
      ITIMEP=IBUFF(57)
      IPHASA=IBUFF(58)
      IGAINA=IBUFF(59)
      IAMPLA=IBUFF(60)
      ITIMEA=IBUFF(61)
      IGAGP =IBUFF(62)
```

```
      IGATP =IBUFF(63)
      IGAGA =IBUFF(64)
      IGATA =IBUFF(65)
      NRATE =IBUFF(66)
      NSCAN =IBUFF(67)
      IPOL  =IBUFF(68)
      IANA  =IBUFF(69)
      NWL   =IBUFF(70)
      NPOL  =IBUFF(71)
      NCMP  =IBUFF(72)
      NANA  =IBUFF(73)
      IHRSO =IBUFF(74)
      IHRSF =IBUFF(75)
      IMINO =IBUFF(76)
      IMINF =IBUFF(77)
      ISECO =IBUFF(78)
      ISECF =IBUFF(79)
      ITICO =IBUFF(80)
      ITICF =IBUFF(81)
      IDAY  =IBUFF(82)
      IYRS  =IBUFF(83)
      LMPHRS=IBUFF(84)
      IMON(1)=IBUFF(85)
      IMON(2)=IBUFF(86)
      IMON(3)=IBUFF(87)
      DO 35 I=1,10
   35 IPMTYP(I)=IBUFF(87+I)
      IPMTDV=IBUFF(98)
      CALL READ
      DO 40 I=1,200
   40 POL(I)=IBUFF(I)
      CALL READ
      DO 50 I=1,200
   50 ANA(I)=IBUFF(I)
      CALL READ
      DO 60 I=1,200
   60 POL(I+200)=IBUFF(I)
      CALL READ
      DO 70 I=1,200
   70 ANA(I+200)=IBUFF(I)
      CALL CLOSEC(ICHAN)
      CALL IFREEC(ICHAN)
      RETURN
      END
```

```
      SUBROUTINE NEWSET
C*********************************************************************
C     OBJECTIVE: TO ALLOW THE OPERATOR TO COMPLETELY INITIALIZE ALL
C                THE SETUP PARAMETERS FROM THE VT55 KEYBOARD.
C*********************************************************************
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      TYPE 140
  140 FORMAT(/,' LAMP PARAMETERS:')
      TYPE 141
  141 FORMAT('$ TYPE(A2/XE,ETC.) = ')
      ACCEPT 142,LMPTYP
  142 FORMAT(A2)
      TYPE 143
  143 FORMAT('$ SERIAL NO.(A18) = ')
      ACCEPT 144,(LMPSER(I),I=1,9)
  144 FORMAT(9A2)
      TYPE 145
  145 FORMAT('$ LAMP VOLTAGE(I3/VOLTS) = ')
      ACCEPT 146,LMPVLT
  146 FORMAT(I4)
      TYPE 147
  147 FORMAT('$ LAMP CURRENT(I2/AMPS) = ')
      ACCEPT 148,LMPAMP
  148 FORMAT(I3)
      TYPE 150
  150 FORMAT(/,' PHOTOMULTIPLIER PARAMETERS:')
      TYPE 151
  151 FORMAT('$ MODEL(A20) = ')
      ACCEPT 152,(IPMTYP(I),I=1,10)
  152 FORMAT(10A2)
      TYPE 153
  153 FORMAT('$ DYNODE VOLTAGE(I4/VOLTS) = ')
      ACCEPT 154,IPMTDV
  154 FORMAT(I5)
      TYPE 160
  160 FORMAT(/,' FARADAY CELL CONTROLLER PARAMETERS:')
      TYPE 161
  161 FORMAT(/,' POLARIZER CHANNEL')
      TYPE 162
  162 FORMAT('$ PHASE(I3/POT SETTING) = ')
      ACCEPT 163,IPHASP
  163 FORMAT(I4)
      TYPE 164
  164 FORMAT('$ GAIN(I3/POT SETTING) = ')
      ACCEPT 163,IGAINP
      TYPE 165
```

```
165 FORMAT('$ AMPLITUDE(I3/POT SETTING) = ')
    ACCEPT 163,IAMPLP
    TYPE 166
166 FORMAT('$ TIME CONSTANT(I3/MILLISECONDS) = ')
    ACCEPT 163,ITIMEP
    TYPE 167
167 FORMAT(/,' ANALYZER CHANNEL')
    TYPE 162
    ACCEPT 163,IPHASA
    TYPE 164
    ACCEPT 163,IGAINA
    TYPE 165
    ACCEPT 163,IAMPLA
    TYPE 166
    ACCEPT 163,ITIMEA
    TYPE 170
170 FORMAT(/,' GALVANOMETER AMPLIFIER PARAMETERS:')
    TYPE 171
171 FORMAT(/,' POLARIZER CHANNEL')
    TYPE 172
172 FORMAT('$ GAIN(I5) = ')
    ACCEPT 173,IGAGP
173 FORMAT(I6)
    TYPE 174
174 FORMAT('$ TIME CONSTANT(I5/MILLISECONDS) = ')
    ACCEPT 173,IGATP
    TYPE 175
175 FORMAT(/,' ANALYZER CHANNEL')
    TYPE 172
    ACCEPT 173,IGAGA
    TYPE 174
    ACCEPT 173,IGATA
    RETURN
    END
```

```
      SUBROUTINE OUT
C*********************************************************************
C     OBJECTIVE: TO WRITE THE SETUP INFORMATION AND DATA FROM THE
C                SPECTRAL SCAN ONTO THE DEVICE "DY1".  THE OUTPUT
C                FILE FIRST SYMBOL IS "C" INDICATING THAT THE VALUES
C                IN THE FILE ARE CONVERTED TO VALUES OF DELTA
C                AND PSI VERSUS WAVELENGTH.
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      COMMON/Z/NAME1,NAME2,IPASS,DURAT,RTIME
      REAL*4 DBLK(2),NAME(20),NAME1,NAME2
      INTEGER NPASS
      IBLOCK=0
      NAME(1)=NAME1
      NAME(2)=NAME2
      CALL PLAC(IPASS,NAME(3))
      N=IRAD50(12,NAME,DBLK)
      ICHAN=IGETC(0)
      IF(ICHAN.LT.0)STOP 'CANNOT ALLOCATE CHANNEL'
      IERROR=IENTER(ICHAN,DBLK,0)
      IF(IERROR.LT.0)STOP 'ENTER FAILURE'
      DO 10 I=1,256
   10 IBUFF(I)=0
      DO 20 I=1,20
      IBUFF(I)=NAMEOP(I)
   20 IBUFF(20+I)=ID(I)
      IBUFF(41)=LMPTYP
      DO 30 I=1,9
   30 IBUFF(41+I)=LMPSER(I)
      IBUFF(51)=LMPVLT
      IBUFF(52)=LMPAMP
      IBUFF(54)=IPHASP
      IBUFF(55)=IGAINP
      IBUFF(56)=IAMPLP
      IBUFF(57)=ITIMEP
      IBUFF(58)=IPHASA
      IBUFF(59)=IGAINA
      IBUFF(60)=IAMPLA
```

```
      IBUFF(61)=ITIMEA
      IBUFF(62)=IGAGP
      IBUFF(63)=IGATP
      IBUFF(64)=IGAGA
      IBUFF(65)=IGATA
      IBUFF(66)=NRATE
      IBUFF(67)=NSCAN
      IBUFF(68)=IPOL
      IBUFF(69)=IANA
      IBUFF(70)=NWL
      IBUFF(71)=NPOL
      IBUFF(72)=NCMP
      IBUFF(73)=NANA
      IBUFF(74)=IHRSO
      IBUFF(75)=IHRSF
      IBUFF(76)=IMINO
      IBUFF(77)=IMINF
      IBUFF(78)=ISECO
      IBUFF(79)=ISECF
      IBUFF(80)=ITICO
      IBUFF(81)=ITICF
      IBUFF(82)=IDAY
      IBUFF(83)=IYRS
      IBUFF(84)=LMPHRS
      IBUFF(85)=IMON(1)
      IBUFF(86)=IMON(2)
      IBUFF(87)=IMON(3)
      DO 35 I=1,10
   35 IBUFF(87+I)=IPMTYP(I)
      IBUFF(98)=IPMTDV
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 40 I=1,200
   40 IBUFF(I)=POL(I)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 50 I=1,200
   50 IBUFF(I)=ANA(I)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 60 I=1,200
   60 IBUFF(I)=POL(I+200)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 70 I=1,200
   70 IBUFF(I)=ANA(I+200)
      CALL WRITE
      CALL IWAIT(ICHAN)
      CALL CLOSEC(ICHAN)
```

```
CALL IFREEC(ICHAN)
RETURN
END
```

```
      SUBROUTINE OUTPUT
C*********************************************************************
C     OBJECTIVE: TO WRITE THE SETUP INFORMATION AND DATA FROM THE
C                SPECTRAL SCAN INTO ANY FLOPPY DISK FILE SPECIFIED
C                BY THE OPERATOR.
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/M/IWAVE(400)
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      REAL*4 DBLK(2),NAME(20)
      IBLOCK=0
      TYPE 100
  100 FORMAT(/,'$ FILE NAME FOR THE NEW SPECTRAL SCAN (A12) = ')
      ACCEPT 200,(NAME(I),I=1,3)
  200 FORMAT(3A4)
      N=IRAD50(12,NAME,DBLK)
      ICHAN=IGETC(0)
      IF(ICHAN.LT.0)STOP 'CANNOT ALLOCATE CHANNEL'
      IERROR=IENTER(ICHAN,DBLK,0)
      IF(IERROR.LT.0)STOP 'ENTER FAILURE'
      DO 10 I=1,256
   10 IBUFF(I)=0
      DO 20 I=1,20
      IBUFF(I)=NAMEOP(I)
   20 IBUFF(20+I)=ID(I)
      IBUFF(41)=LMPTYP
      DO 30 I=1,9
   30 IBUFF(41+I)=LMPSER(I)
      IBUFF(51)=LMPVLT
      IBUFF(52)=LMPAMP
      IBUFF(54)=IPHASP
      IBUFF(55)=IGAINP
      IBUFF(56)=IAMPLP
      IBUFF(57)=ITIMEP
      IBUFF(58)=IPHASA
      IBUFF(59)=IGAINA
      IBUFF(60)=IAMPLA
      IBUFF(61)=ITIMEA
      IBUFF(62)=IGAGP
      IBUFF(63)=IGATP
```

```
      IBUFF(64)=IGAGA
      IBUFF(65)=IGATA
      IBUFF(66)=NRATE
      IBUFF(67)=NSCAN
      IBUFF(68)=IPOL
      IBUFF(69)=IANA
      IBUFF(70)=NWL
      IBUFF(71)=NPOL
      IBUFF(72)=NCMP
      IBUFF(73)=NANA
      IBUFF(74)=IHRSO
      IBUFF(75)=IHRSF
      IBUFF(76)=IMINO
      IBUFF(77)=IMINF
      IBUFF(78)=ISECO
      IBUFF(79)=ISECF
      IBUFF(80)=ITICO
      IBUFF(81)=ITICF
      IBUFF(82)=IDAY
      IBUFF(83)=IYRS
      IBUFF(84)=LMPHRS
      IBUFF(85)=IMON(1)
      IBUFF(86)=IMON(2)
      IBUFF(87)=IMON(3)
      DO 35 I=1,10
   35 IBUFF(87+I)=IPMTYP(I)
      IBUFF(98)=IPMTDV
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 40 I=1,200
   40 IBUFF(I)=POL(I)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 50 I=1,200
   50 IBUFF(I)=ANA(I)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 60 I=1,200
   60 IBUFF(I)=POL(I+200)
      CALL WRITE
      CALL IWAIT(ICHAN)
      DO 70 I=1,200
   70 IBUFF(I)=ANA(I+200)
      CALL WRITE
      CALL IWAIT(ICHAN)
      CALL CLOSEC(ICHAN)
      CALL IFREEC(ICHAN)
      RETURN
      END
```

```
SUBROUTINE PLAC(NPASS,PNAME)
  REAL*4 PNAME
  IF (NPASS.EQ.1) PNAME='0001'
  IF (NPASS.EQ.2) PNAME='0002'
  IF (NPASS.EQ.3) PNAME='0003'
  IF (NPASS.EQ.4) PNAME='0004'
  IF (NPASS.EQ.5) PNAME='0005'
  IF (NPASS.EQ.6) PNAME='0006'
  IF (NPASS.EQ.7) PNAME='0007'
  IF (NPASS.EQ.8) PNAME='0008'
  IF (NPASS.EQ.9) PNAME='0009'
  IF (NPASS.EQ.10) PNAME='0010'
  IF (NPASS.EQ.11) PNAME='0011'
  IF (NPASS.EQ.12) PNAME='0012'
  IF (NPASS.EQ.13) PNAME='0013'
  IF (NPASS.EQ.14) PNAME='0014'
  IF (NPASS.EQ.15) PNAME='0015'
  IF (NPASS.EQ.16) PNAME='0016'
  IF (NPASS.EQ.17) PNAME='0017'
  IF (NPASS.EQ.18) PNAME='0018'
  IF (NPASS.EQ.19) PNAME='0019'
  IF (NPASS.EQ.20) PNAME='0020'
  IF (NPASS.EQ.21) PNAME='0021'
  IF (NPASS.EQ.22) PNAME='0022'
  IF (NPASS.EQ.23) PNAME='0023'
  IF (NPASS.EQ.24) PNAME='0024'
  IF (NPASS.EQ.25) PNAME='0025'
  IF (NPASS.EQ.26) PNAME='0026'
  IF (NPASS.EQ.27) PNAME='0027'
  IF (NPASS.EQ.28) PNAME='0028'
  IF (NPASS.EQ.29) PNAME='0029'
  IF (NPASS.EQ.30) PNAME='0030'
  IF (NPASS.EQ.31) PNAME='0031'
  IF (NPASS.EQ.32) PNAME='0032'
  IF (NPASS.EQ.33) PNAME='0033'
  IF (NPASS.EQ.34) PNAME='0034'
  IF (NPASS.EQ.35) PNAME='0035'
  IF (NPASS.EQ.36) PNAME='0036'
  IF (NPASS.EQ.37) PNAME='0037'
  IF (NPASS.EQ.38) PNAME='0038'
  IF (NPASS.EQ.39) PNAME='0039'
  IF (NPASS.EQ.40) PNAME='0040'
  IF (NPASS.EQ.41) PNAME='0041'
  IF (NPASS.EQ.42) PNAME='0042'
  IF (NPASS.EQ.43) PNAME='0043'
  IF (NPASS.EQ.44) PNAME='0044'
  IF (NPASS.EQ.45) PNAME='0045'
  IF (NPASS.EQ.46) PNAME='0046'
  IF (NPASS.EQ.47) PNAME='0047'
```

```
IF (NPASS.EQ.48) PNAME='0048'
IF (NPASS.EQ.49) PNAME='0049'
IF (NPASS.EQ.50) PNAME='0050'
RETURN
END
```

```
SUBROUTINE PLAC2(NPASS,PNAME)
  REAL*4 PNAME
  IF (NPASS.EQ.1) PNAME='00.1'
  IF (NPASS.EQ.2) PNAME='00.2'
  IF (NPASS.EQ.3) PNAME='00.3'
  IF (NPASS.EQ.4) PNAME='00.4'
  IF (NPASS.EQ.5) PNAME='00.5'
  IF (NPASS.EQ.6) PNAME='00.6'
  IF (NPASS.EQ.7) PNAME='00.7'
  IF (NPASS.EQ.8) PNAME='00.8'
  IF (NPASS.EQ.9) PNAME='00.9'
  IF (NPASS.EQ.10) PNAME='0.10'
  IF (NPASS.EQ.11) PNAME='0.11'
  IF (NPASS.EQ.12) PNAME='0.12'
  IF (NPASS.EQ.13) PNAME='0.13'
  IF (NPASS.EQ.14) PNAME='0.14'
  IF (NPASS.EQ.15) PNAME='0.15'
  IF (NPASS.EQ.16) PNAME='0.16'
  IF (NPASS.EQ.17) PNAME='0.17'
  IF (NPASS.EQ.18) PNAME='0.18'
  IF (NPASS.EQ.19) PNAME='0.19'
  IF (NPASS.EQ.20) PNAME='0.20'
  IF (NPASS.EQ.21) PNAME='0.21'
  IF (NPASS.EQ.22) PNAME='0.22'
  IF (NPASS.EQ.23) PNAME='0.23'
  IF (NPASS.EQ.24) PNAME='0.24'
  IF (NPASS.EQ.25) PNAME='0.25'
  IF (NPASS.EQ.26) PNAME='0.26'
  IF (NPASS.EQ.27) PNAME='0.27'
  IF (NPASS.EQ.28) PNAME='0.28'
  IF (NPASS.EQ.29) PNAME='0.29'
  IF (NPASS.EQ.30) PNAME='0.30'
  IF (NPASS.EQ.31) PNAME='0.31'
  IF (NPASS.EQ.32) PNAME='0.32'
  IF (NPASS.EQ.33) PNAME='0.33'
  IF (NPASS.EQ.34) PNAME='0.34'
  IF (NPASS.EQ.35) PNAME='0.35'
  IF (NPASS.EQ.36) PNAME='0.36'
  IF (NPASS.EQ.37) PNAME='0.37'
  IF (NPASS.EQ.38) PNAME='0.38'
  IF (NPASS.EQ.39) PNAME='0.39'
  IF (NPASS.EQ.40) PNAME='0.40'
  IF (NPASS.EQ.41) PNAME='0.41'
  IF (NPASS.EQ.42) PNAME='0.42'
  IF (NPASS.EQ.43) PNAME='0.43'
  IF (NPASS.EQ.44) PNAME='0.44'
  IF (NPASS.EQ.45) PNAME='0.45'
  IF (NPASS.EQ.46) PNAME='0.46'
  IF (NPASS.EQ.47) PNAME='0.47'
```

```
IF (NPASS.EQ.48) PNAME='0.48'
IF (NPASS.EQ.49) PNAME='0.49'
IF (NPASS.EQ.50) PNAME='0.50'
RETURN
END
```

```
      SUBROUTINE PSIDEL(RHO)
C*************************************************************
C
C     PURPOSE: TO CALCULATE DELTA AND PSI GIVEN THE COMPLEX
C              REFLECTION COEFFICENT
C
C*************************************************************
      COMMON/FLM/DELC,PSIC,PHI
      COMPLEX RHO
      PSIC=DATAN(CABS(RHO))/0.01745329252
      DELC=DATAN2(AIMAG(RHO),REAL(RHO))/0.01745329252
      IF (DELC.LT.0.0) DELC=DELC+360.0000
      IF (PSIC.LT.0.0) PSIC=PSIC+180.0000
      RETURN
      END
```

```
      SUBROUTINE P1INFO
C*********************************************************************
C     OBJECTIVE: OUTPUTS ALL THE SETUP INFORMATION FOR A SPECTRAL SCAN
C                TO THE LINE PRINTER, EXCEPT FOR THE DATA ACQUISITION
C                TIME.
C*********************************************************************
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      PRINT 100,IDAY,(IMON(I),I=1,3),IYRS
  100 FORMAT(/10X,I2,'-',3A1,'-',I2)
      PRINT 101,(NAMEOP(I),I=1,20)
  101 FORMAT(10X,20A2)
      PRINT 102,(ID(I),I=1,20)
  102 FORMAT(10X,20A2)
      PRINT 110
  110 FORMAT(/10X,'MANUAL AZIMUTHS(DEG):')
      P=NPOL/100.
      A=NANA/100.
      C=NCMP/100.
      WL=NWL/100.
      PRINT 111,P,A,C,WL
  111 FORMAT(/12X,'P = ',F7.2,
     C       /12X,'A = ',F7.2,
     C       /12X,'C = ',F7.2,
     C       /12X,'WL= ',F7.2)
      PRINT 120
  120 FORMAT(/10X,'LAMP:')
      PRINT 121,LMPTYP
  121 FORMAT(/12X,'TYPE       = ',A2)
      PRINT 122,(LMPSER(I),I=1,9)
  122 FORMAT(12X,'SERIAL NO. = ',9A2)
      PRINT 123,LMPVLT
  123 FORMAT(12X,'LAMP VOLTAGE(VOLTS) = ',I4)
      PRINT 124,LMPAMP
  124 FORMAT(12X,'LAMP CURRENT(AMPS)  = ',I3)
      PRINT 125,LMPHRS
  125 FORMAT(12X,'LAMP OPERATIONAL LIFE(HOURS) = ',I5)
      PRINT 130
  130 FORMAT(/10X,'PHOTOMULTIPLIER:')
      PRINT 131,(IPMTYP(I),I=1,10)
  131 FORMAT(/12X,'MODEL = ',10A2)
      PRINT 132,IPMTDV
```

```
  132 FORMAT(12X,'DYNODE VOLTAGE(VOLTS) = ',I5)
      PRINT 140
  140 FORMAT(/10X,'FARADAY CELL CONTROLLER:')
      PRINT 141
  141 FORMAT(/12X,'POLARIZER')
      PRINT 142,IPHASP,IGAINP,IAMPLP,ITIMEP
  142 FORMAT(/12X,'PHASE              = ',I4,
     C       /12X,'GAIN               = ',I4,
     C       /12X,'AMPLITUDE          = ',I4,
     C       /12X,'TIME CONSTANT(MS) = ',I4)
      PRINT 143
  143 FORMAT(/12X,'ANALYZER')
      PRINT 142,IPHASA,IGAINA,IAMPLA,ITIMEA
      PRINT 150
  150 FORMAT(/10X,'GALVANOMETER AMPLIFIER:')
      PRINT 141
      PRINT 151,IGAGP,IGATP
  151 FORMAT(/12X,'GAIN               = ',I6,
     C       /12X,'TIME CONSTANT(MS) = ',I6)
      PRINT 143
      PRINT 151,IGAGA,IGATA
      IGAIN=IPOL/64
      NCHP=IPOL-IGAIN*64
      NCHA=IANA-IGAIN*64
      PRINT 160
  160 FORMAT(/10X,'ANALOG/DIGITAL CONVERTOR:')
      PRINT 161,NCHP,NCHA,IGAIN
  161 FORMAT(/12X,'POL. INPUT CH. = ',I1,
     C       /12X,'ANA. INPUT CH. = ',I1,
     C       /12X,'ADC GAIN       = ',I1)
      RETURN
      END

      SUBROUTINE P1TIME
C*********************************************************************
C     OBJECTIVE: OUTPUTS THE DATA ACQUISITION FOR THE GIVEN SPECTRAL
C                SCAN TO THE LINE PRINTER.
C*********************************************************************
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      PRINT 100,NSCAN
  100 FORMAT(/,10X,'THE NUMBER OF SPECTRA AVERAGED = ',I5)
      PRINT 110,NRATE
  110 FORMAT(10X,'MEASURED SCAN RATE VIA COUNTER = ',I5)
      PRINT 120,IHRSO,IMINO,ISECO,ITICO
  120 FORMAT(/10X,'DATA COLLECTION INITIATED AT: ',3(I2,':'),I2)
      PRINT 130,IHRSF,IMINF,ISECF,ITICF
  130 FORMAT(10X,'DATA COLLECTION FINISHED  AT: ',3(I2,':'),I2)
      RETURN
```

END

```
      SUBROUTINE RATE
C*********************************************************************
C     OBJECTIVE: MAKE TWO "AVERAGED" DETERMINATIONS OF THE ROTATION
C                RATE OF THE FILTER DISK(SPECTRAL SCAN RATE) VIA
C                CHANNEL 4 OF THE ADAC 1604/OPI COUNTER CARD WHICH
C                IS WIRE WRAPPED IN THE FREQUENCY MODE.  THESE TWO
C                RATES OF ROTATION ARE MADE AT A TIME INTERVAL
C                DETERMINED BY THE SUBROUTINE "DELAY"(15 SECONDS).
C                IF THE TWO RATES ARE IDENTICAL, THE SCAN RATE IS
C                DETERMINED TO BE STABLE(FREE OF "CHATTER", ETC.)
C                AND PROGRAM CONTROL IS RETURNED TO THE MAIN
C                PROGRAM "SEV001".
C*********************************************************************
      COMMON/B/NRATE,NSCAN,IPOL,IANA
   10 TYPE 20
   20 FORMAT(/,'$ DO YOU WANT TO ADJUST THE SCAN RATE(Y/N)? ')
      ACCEPT 30,MRATE
   30 FORMAT(A1)
      IF(MRATE.NE.1HY.AND.MRATE.NE.1HN)GOTO 10
      IF(MRATE.NE.1HY)GOTO 85
      TYPE 40
   40 FORMAT(/,'$ ADJUST THE SPECTRAL SCANNER SPEED, THEN "RETURN".')
      ACCEPT 50,IWAIT
   50 FORMAT(A1)
   60 ISUM0=0
      DO 70 I=1,3
   70 ISUM0=ISUM0+IPEEK("164406)
      CALL DELAY
      ISUM1=0
      DO 80 I=1,3
   80 ISUM1=ISUM1+IPEEK("164406)
      IF(ISUM0.NE.ISUM1)GO TO 60
   85 NRATE=IPEEK("164406)
      TYPE 90,NRATE
   90 FORMAT(/,'  STEADY SCAN RATE ESTABLISHED.  SCAN RATE = ',I5)
      RETURN
      END
```

```
      SUBROUTINE READ
C*********************************************************************
C     OBJECTIVE: TO ACTUALLY READ THE DATA STORED ON THE FLOPPY DISK
C                INTO BUFFER MEMORY.
C*********************************************************************
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      IERROR=IREADW(256,IBUFF,IBLOCK,ICHAN)
      IF(IERROR.LT.0)STOP 'FATAL READ'
      IBLOCK=IBLOCK+1
      RETURN
      END
```

SUBROUTINE REFIND VARIABLES EXPLANATION

Variable Name	SYMBOL	EXPLANATION
DELC	Δ	-Value of delta for a given reflection (either theoretical or measured) (deg)
PHI	φ_o	-Angle of light propagation in the ambient media (angle of incidence relative to the surface normal)
PSIC	ψ	-Value of Psi for a given reflection (either theoretical or measured) (deg)
RHO	ρ	-Reflection coefficent of the reflection
TNC	$\hat{n}_\varepsilon$	-Complex refractive index of the surface
TN	n_ε	-Refractive index of the surface (real component)
TK	K_e	-Extinction coefficent of the surface

SUBROUTINES CALLED

NONE

EQUATIONS USED FOR CALCULATIONS

$$\rho = \psi \exp(-i\Delta)$$

$$\hat{n}_\varepsilon = \frac{n_o \sin\varphi_o \sqrt{1+a^2}}{a}$$

with

$$a = \frac{1 + \rho_r}{1 - \rho} \cot\varphi_o$$

```
      SUBROUTINE REFIND(TNO,TN,TK)
C*********************************************************************
C      OBJECTIVE: TO COMPUTE THE COMPLEX REFRACTIVE INDEX OF A BARE
C                 SURFACE FOR A SINGLE SET OF VALUES OF DELTA, PSI,
C                 WAVELENGTH, AND INCIDENT MEDIUM REFRACTIVE INDEX.
C*********************************************************************
      COMMON/FLM/DELC,PSIC,PHI
      COMPLEX RHO,T2,S2,TNC,I,S1,T1,PHI
      TNOC=CMPLX(TNO,-0.0)
      ALPHA=0.01745329252
      PSI=PSIC*ALPHA
      DEL=DELC*ALPHA
      I=CMPLX(0.0,1.0)
      RHO=(SIN(PSI)/COS(PSI))*CEXP(I*DEL)
      S1=SIN(PHI)
      T1=SIN(PHI)/COS(PHI)
      T2=(1.0+RHO)/((1.0-RHO)*T1)
      S2=T2/CSQRT(1.0+T2*T2)
      TNC=(TNOC*S1)/S2
      TN=ABS(REAL(TNC))
      TK=ABS(AIMAG(TNC))
      RETURN
      END
```

```
      SUBROUTINE RESET
C*********************************************************************
C     OBJECTIVE: TO ALLOW THE OPERATOR TO CHANGE ONLY THOSE SETUP
C                PARAMETERS WHICH ARE TYPICALLY CHANGED FROM ONE
C                SPECTRAL SCAN TO ANOTHER.  THIS MODIFYS SETUP
C                INFORMATION READ FROM OLD DATA FILES TO MAKE IT
C                APPLICABLE TO THE CURRENT SPECTRAL SCAN OR
C                COMPLETES THE INITIALIZATION OF SETUP
C                INFORMATION STARTED WITH "NEWSET".
C*********************************************************************
      COMMON/A/NAVG(400),POL(400),ANA(400)
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      COMMON/Z/NAME1,NAME2,NPASS,DURAT,RTIME
      REAL*4 NAME1,NAME2
      INTEGER NPASS,DURAT
      TYPE 100
  100 FORMAT(/,'$ NAME(A40) = ')
      ACCEPT 101,(NAMEOP(I),I=1,20)
  101 FORMAT(20A2)
      TYPE 102
  102 FORMAT(/,'$ DATE(I2-A3-I2/DAY-MONTH-YEAR) = ')
      ACCEPT 103,IDAY,(IMON(I),I=1,3),IYRS
  103 FORMAT(I2,1X,3A1,1X,I2)
      TYPE 104
  104 FORMAT(/,'$ LAMP OPERATIONAL LIFE(I5/HOURS) = ')
      ACCEPT 105,LMPHRS
  105 FORMAT(I5)
  106 CONTINUE
      TYPE 107
  107 FORMAT(/,'$ EXPERIMENT IDENTIFICATION(A40) = ')
      ACCEPT 101,(ID(I),I=1,20)
  108 TYPE 109
  109 FORMAT(/,'$ HAVE MANUAL AZIMUTHS BEEN CHANGED(Y/N)? ')
      ACCEPT 110,MANAZ
  110 FORMAT(A1)
      IF(MANAZ.NE.1HY.AND.MANAZ.NE.1HN)GOTO 108
      IF(MANAZ.NE.1HY)GOTO 119
      TYPE 112
  112 FORMAT(/,' MANUAL AZIMUTH SETTINGS(DEG):')
      TYPE 113
  113 FORMAT('$ POLARIZER(F7.2)    = ')
      ACCEPT 114,P
  114 FORMAT(F7.2)
      NPOL=P*100.
      TYPE 115
```

```
115 FORMAT('$ ANALYZER(F7.2)     = ')
    ACCEPT 114,A
    NANA=A*100.
116 TYPE 117
117 FORMAT('$ COMPENSATOR(F7.2) = ')
    ACCEPT 114,C
    IF(C.NE.135.0.AND.C.NE.45.0)GOTO 116
    NCMP=C*100.
    TYPE 118
118 FORMAT('$ MONOCHROMATOR SETTING AT NULL(F7.2) = ')
    ACCEPT 114,WL
    NWL=WL*100.
119 CONTINUE
    IGAIN=3
    NCHP=0
    NCHA=1
120 TYPE 121
121 FORMAT(/,' ANALOG/DIGITAL CONVERTOR PRESET PARAMETERS:')
122 TYPE 123,NCHP
123 FORMAT('  POL. INPUT CH. = ',I1)
    TYPE 124,NCHA
124 FORMAT('  ANA. INPUT CH. = ',I1)
    TYPE 125,IGAIN
125 FORMAT('  ADC GAIN       = ',I1)
    TYPE 126
126 FORMAT(/,'$ DO YOU WANT TO CHANGE THESE(Y/N)? ')
    ACCEPT 110,NADC
    IF(NADC.NE.1HY.AND.NADC.NE.1HN)GOTO 120
    IF(NADC.EQ.1HN)GOTO 131
    TYPE 127
127 FORMAT('$ POL. INPUT CH.(I1) = ')
    ACCEPT 128,NCHP
128 FORMAT(I1)
    TYPE 129
129 FORMAT('$ ANA. INPUT CH.(I1) = ')
    ACCEPT 128,NCHA
    TYPE 130
130 FORMAT('$ ADC GAIN(I1)       = ')
    ACCEPT 128,IGAIN
    GOTO 120
131 CONTINUE
    IPOL=IGAIN*64+NCHP
    IANA=IGAIN*64+NCHA
    TYPE 132
132 FORMAT(/,'$HOW MANY SPECTRAL SCANS ARE TO BE AVERAGED PER
   CFILE(I5)?')
    ACCEPT 133,NSCAN
133 FORMAT(I5)
    DO 140 I=1,400
```

```
      POL(I)=0.
      ANA(I)=0.
  140 NAVG(I)=0
      TYPE 150
      NAME1='DY10'
  150 FORMAT('$ FOUR LETTERS IN FILE NAME (EX. ABCD)= ')
      ACCEPT 190,NAME2
      TYPE 170
      ACCEPT 180,DURAT
  170 FORMAT ('$ TIME DURATION FOR EXPERIMENT IN MINUTES (A2)= ')
  180 FORMAT (I2)
  190 FORMAT (A4)
      RETURN
      END
```

```
      SUBROUTINE STEPER
C*****************************************************************
C     OBJECTIVE:  TO STEP THE ANALYSER OR POLARIZER STEP
C                 MOTOR CONTROLERS SO AS TO MINIMIZE THE
C                 NECESSARY CONTROLER CURRENT TO THE FARADAY
C                 CORES.
C
C     PROGRAM WRITTEN BY MICHAEL ARMSTRONG IN FALL 1983
C*****************************************************************
      COMMON/STEP/DEVICE,ISTEP,IPC,IAC
      I=ISTEP
      A=DEVICE
      IDIR=0
      IF(A.EQ.'A')GOTO 30
      IF(A.EQ.'P')GOTO 40
30    IF(I.LT.0)IDIR=2048
      I=IABS(I)
32    IF(IPEEK("164502).GT.0)GOTO 32
      IF(I.GT.2047)GOTO 31
      I=I+IDIR
      CALL IPOKE("164502,I)
      GOTO 100
31    I=I-2047
      IT=2047+IDIR
      CALL IPOKE("164502,IT)
      GOTO 32
40    IF(I.LT.0)IDIR=2048
      I=IABS(I)
42    IF(IPEEK("164506).GT.0) GOTO 42
      IF(I.GT.2047)GOTO 41
      I=I+IDIR
      CALL IPOKE("164506,I)
      GOTO 100
41    I=I-2047
      IT=IDIR+2047
      CALL IPOKE("164506,IT)
      GOTO 42
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE T1INFO
C*********************************************************************
C     OBJECTIVE: DISPLAYS ALL THE SETUP INFORMATION FOR A GIVEN
C                SPECTRAL SCAN ON THE VT55 GRAPHICS TERMINAL SCREEN,
C                EXCEPT FOR THE DATA ACQUISTION TIME.
C*********************************************************************
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/E/NAMEOP(20),ID(20)
      COMMON/F/LMPTYP,LMPSER(10),LMPVLT,LMPAMP
      COMMON/G/IPMTYP(10),IPMTDV
      COMMON/H/IPHASP,IPHASA,IGAINP,IGAINA,IAMPLP,IAMPLA,ITIMEP,ITIMEA
      COMMON/I/IGAGP,IGAGA,IGATP,IGATA
      COMMON/J/NWL,NPOL,NCMP,NANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      COMMON/L/IDAY,IMON(3),IYRS,LMPHRS
      TYPE 100,IDAY,(IMON(I),I=1,3),IYRS
  100 FORMAT(/10X,I2,'-',3A1,'-',I2)
      TYPE 101,(NAMEOP(I),I=1,20)
  101 FORMAT(10X,10A2)
      TYPE 102,(ID(I),I=1,20)
  102 FORMAT(10X,20A2)
      TYPE 110
  110 FORMAT(/10X,'MANUAL AZIMUTHS(DEG):')
      P=NPOL/100.
      A=NANA/100.
      C=NCMP/100.
      WL=NWL/100.
      TYPE 111,P,A,C,WL
  111 FORMAT(/12X,'P = ',F7.2,
     C       /12X,'A = ',F7.2,
     C       /12X,'C = ',F7.2,
     C       /12X,'WL= ',F7.2)
      TYPE 120
  120 FORMAT(/10X,'LAMP:')
      TYPE 121,LMPTYP
  121 FORMAT(/12X,'TYPE        = ',A2)
      TYPE 122,(LMPSER(I),I=1,9)
  122 FORMAT(12X,'SERIAL NO. = ',9A2)
      TYPE 123,LMPVLT
  123 FORMAT(12X,'LAMP VOLTAGE(VOLTS) = ',I4)
      TYPE 124,LMPAMP
  124 FORMAT(12X,'LAMP CURRENT(AMPS)  = ',I3)
      TYPE 125,LMPHRS
  125 FORMAT(12X,'LAMP OPERATIONAL LIFE(HOURS) = ',I5)
      TYPE 130
  130 FORMAT(/10X,'PHOTOMULTIPLIER:')
      TYPE 131,(IPMTYP(I),I=1,10)
  131 FORMAT(/12X,'MODEL = ',10A2)
```

```
      TYPE 132,IPMTDV
  132 FORMAT(12X,'DYNODE VOLTAGE(VOLTS) = ',I5)
      TYPE 140
  140 FORMAT(/10X,'FARADAY CELL CONTROLLER:')
      TYPE 141
  141 FORMAT(/12X,'POLARIZER')
      TYPE 142,IPHASP,IGAINP,IAMPLP,ITIMEP
  142 FORMAT(/12X,'PHASE              = ',I4,
     C       /12X,'GAIN               = ',I4,
     C       /12X,'AMPLITUDE          = ',I4,
     C       /12X,'TIME CONSTANT(MS) = ',I4)
      TYPE 143
  143 FORMAT(/12X,'ANALYZER')
      TYPE 142,IPHASA,IGAINA,IAMPLA,ITIMEA
      TYPE 150
  150 FORMAT(/10X,'GALVANOMETER AMPLIFIER:')
      TYPE 141
      TYPE 151,IGAGP,IGATP
  151 FORMAT(/12X,'GAIN               = ',I6,
     C       /12X,'TIME CONSTANT(MS) = ',I6)
      TYPE 143
      TYPE 151,IGAGA,IGATA
      IGAIN=IPOL/64
      NCHP=IPOL-IGAIN*64
      NCHA=IANA-IGAIN*64
      TYPE 160
  160 FORMAT(/10X,'ANALOG/DIGITAL CONVERTOR:')
      TYPE 161,NCHP,NCHA,IGAIN
  161 FORMAT(/12X,'POL. INPUT CH. = ',I1,
     C       /12X,'ANA. INPUT CH. = ',I1,
     C       /12X,'ADC GAIN       = ',I1)
      RETURN
      END

      SUBROUTINE T1TIME
C*********************************************************************
C     OBJECTIVE: DISPLAYS THE DATA ACQUISITION TIME FOR THE GIVEN
C                SPECTRAL SCAN ON THE CRT SCREEN.
C*********************************************************************
      COMMON/B/NRATE,NSCAN,IPOL,IANA
      COMMON/K/IHRSO,IMINO,ISECO,ITICO,IHRSF,IMINF,ISECF,ITICF
      TYPE 100,NSCAN
  100 FORMAT(/,10X,'THE NUMBER OF SPECTRA AVERAGED = ',I5)
      TYPE 110,NRATE
  110 FORMAT(10X,'MEASURED SCAN RATE VIA COUNTER = ',I5)
      TYPE 120,IHRSO,IMINO,ISECO,ITICO
  120 FORMAT(/10X,'DATA COLLECTION INITIATED AT: ',3(I2,':'),I2)
      TYPE 130,IHRSF,IMINF,ISECF,ITICF
  130 FORMAT(10X,'DATA COLLECTION FINISHED  AT: ',3(I2,':'),I2)
```

```
RETURN
END
```

```
      SUBROUTINE WLCALC
C*********************************************************************
C     OBJECTIVE: TO CONVERT ENCODER COUNTS INTO WAVELENGTH USING
C                PREDETERMINED CALIBRATION CURVE PARAMETERS.
C*********************************************************************
      COMMON/M/IWAVE(400)
      OPEN(UNIT=2,NAME='DY1:WLCALC.DAT',TYPE='OLD')
      READ(2,996)IU,IL,A1,B1,R1,A2,B2,R2
  996 FORMAT(2(1X,I3),2(1X,F8.4,1X,F8.2,1X,F6.4))
      CLOSE(UNIT=2,DISPOSE='SAVE')
      DO 40 I=1,400
      IF(I.LE.IL.AND.I.GE.IU)GOTO 10
      IF(I.GT.IL)GOTO 20
      COUNT=I
      WL=A1*COUNT+B1
      GOTO 30
   10 COUNT=I
      WL=A2*COUNT+B2
      GOTO 30
   20 COUNT=I-400
      WL=A1*COUNT+B1
   30 CONTINUE
   40 IWAVE(I)=WL
      RETURN
      END
```

```
      SUBROUTINE WRITE
C*********************************************************************
C     OBJECTIVE: THIS SUBROUTINE ACTUALLY WRITES THE DATA STORED IN THE
C                BUFFER MEMORY TO THE SPECIFIED DISK FILE.
C*********************************************************************
      COMMON/N/ICHAN,IBLOCK,IFLAG,IBUFF(256)
      EXTERNAL FINISH
      IERROR=IWRITF(256,IBUFF,IBLOCK,ICHAN,IAREA,FINISH)
      IF(IERROR.LT.0)STOP 'FATAL WRITE'
      IBLOCK=IBLOCK+1
      RETURN
      END
```

BIBLIOGRAPHY

1) Farmer, Joseph C. Ph. D. Thesis, Department of Chemical Engineering, University of California, Berkeley, March 1983. Appendix. LBL-15525.

2) Muller, Rolf H. "Principles of Ellipsometry." in Advances in Electrochemistry and Electrochemical Engineering. Vol. 9. P. Delahay and C. W. Tobias, ed. John Wiley and Sons, New York, 1973.

3) Mayer, Steven T. M.S. Thesis, Department of Chemical Engineering, University of California, Berkeley, Dec. 1985. LBL-20503.

4) Armstrong, M. J. M.S. Thesis, Department of Chemical Engineering, University of California, Berkeley, Dec. 1984. LBL-18599.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

LAWRENCE BERKELEY LABORATORY
TECHNICAL INFORMATION DEPARTMENT
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720