ELLIPSE User's Manual and Program Reference

SAND--89-0008

DE89 014575

Herbert L. Tardy
Photovoltaic Cell Research Division
Sandia National Laboratories
Albuquerque, New Mexico 87185

Abstract

A new program, ELLIPSE, for the analysis of ellipsometric data is described. The program is interactive and includes on-line help, performs explicit error analysis for single-angle-of-incidence measurements, allows the input of four-zone null data, and performs least-squares analysis of multiple-angle-of-incidence data. Solutions for the transparent-film-on-substrate model are obtained using decoupled equations for film thickness and index. No initial guess is required for thickness, and the algorithm is insensitive to the initial guess for film index.

This document combines the user's manual and program description for ELLIPSE, and includes several examples of its use.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or 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 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. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

*This work was supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or 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 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. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Disclaimers

This computer software has been developed under sponsorship of the Department of Energy. Any further distribution by any holder of this software package or other data therein outside of DOE offices or other DOE contractors, unless otherwise specifically provided for, is prohibited without the approval of the National Energy Software Center. Requests from outside the Department for DOE-developed computer software shall be directed to the Director, National Energy Software Center, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.

This material was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or 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.

Contents

I. Introduction

For nearly twenty years, the NBS McCrackin program[1] has been the standard code for the analysis of ellipsometric data. This paper describes ELLIPSE, a new program which takes advantage of advances that have occurred since the NBS code was written. Although it lacks some of the features of the McCrackin code, ELLIPSE has several advantages: it is interactive and includes on-line help, it performs explicit error analysis for single angle-of-incidence measurements[2], it uses the decoupled form of the ellipsometric equations[3] to invert the transparent-film-on-substrate (TFS) model, it allows the input of four-zone data, and it performs least-squares analysis of multiple-angle-of-incidence (MAI) data.

The program is written in DEC VAX FORTRAN-77 running under VMS Version 4.0 and higher. All calculations are performed in double precision, using REAL*8 and COMPLEX*16 data types. ELLIPSE is also available for IBM PC's and compatibles, as described in Appendix E. The PC version has the additional capability of reading data directly from Rudolph AutoEl II ellipsometers.

Standard notation for the ellipsometric angles ψ and Δ is used throughout, based on the definition of ρ as the ratio of the complex amplitude reflection coefficients for p- and s-polarization, R_p and R_s:

$$\rho = R_p/R_s = \tan\psi \, e^{i\Delta} \, . \qquad 1$$

This manual assumes that the reader is familiar with the fundamentals of ellipsometric measurement. Introductory material can be found in instrument manuals, or in Ref. 4.

II. User's Manual

ELLIPSE is intended as a comprehensive system for analysis of single-wavelength ellipsometric data. As such, it supports data input in the form of ellipsometric angles or prism azimuths, and allows analysis of the data in a variety of modes. Uncertainties are carried with the data and fixed model parameters at all times, and no result is reported without an error estimate.

The operation of ELLIPSE proceeds in a two-step loop, as shown in Fig. 1. The user first enters data and experimental parameters, then executes the appropriate solution process, and returns to the entry phase. Data and parameters can be entered in any order.

II.1. ELLIPSE concepts

II.1.1. Commands

ELLIPSE's user interface consists of three menus: data input, parameter input, and mode selection. Commands are entered as single characters. ELLIPSE is insensitive to the case of the instructions. If further input, such as a parameter value, is required ELLIPSE issues the

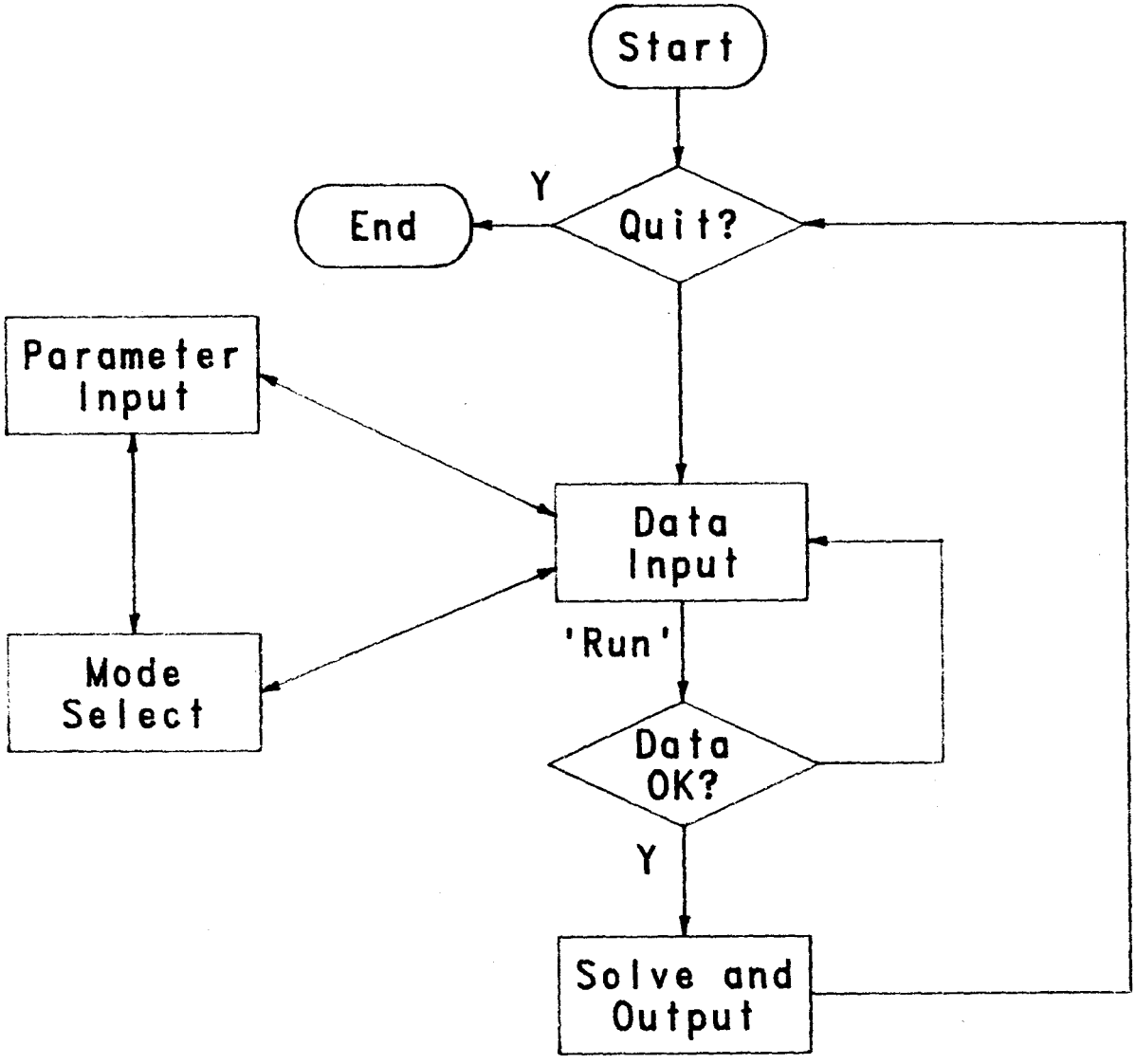

Figure 1. General structure of ELLIPSE.

necessary prompts. Except at the start of a session, the menus are displayed only when help is requested. This makes the menu structure transparent to the experienced user, and conserves screen space for the display of data and solutions. Each menu sends a different command prompt, so it is not necessary to call the help routine just to learn which menu is active. The menus are described in detail in a later section.

II.1.2. The stack

The stack is a user-defined model of the sample surface in terms of film refractive indices and thicknesses and substrate index. ELLIPSE can store up to twenty films, transparent or absorbing (ellipsometry is unlikely to be useful on larger stacks). You have complete control of the model. At any point in a session you can examine the current stack, add or remove films, change the parameters of any layer, change a single parameter, or build an entire new stack.

At the start of program execution, default values for the stack and some other system parameters such as the wavelength and the azimuth uncertainties are read from a file (ELLIPSE.DEF). At any time, the default parameter set can be restored. Conversely, the current model values can be written to the default file so that the program can be restarted without the need for tedious re-entry of a long list of model parameters.

II.1.3. Analysis mode

Several options exist for analyzing data given to ELLIPSE: 1) For a single measurement, solve for any two parameters of a stack of up to twenty films. 2) For multiple measurements at a single angle of incidence, provide a least-squares solution to an arbitrary two-unknown problem. 3) For measurements at multiple angles of incidence, provide a least-squares solution for up to five unknown parameters.

Two modes are special cases for analyzing single measurements: the 'BARE' mode yields an analytical solution for the complex refractive index of a bulk material, and the 'SINGLE' mode gives a numerical solution for the index and thickness of a transparent film on top of an arbitrary stack of films. Any two stack parameters can be obtained from a single measurement using a Levenberg-Marquardt optimization (e.g., top-film index and an interior-film thickness). The same optimization routine can be used to obtain a least-squares solution for any two stack parameters given multiple measurements at the same incident angle. In this way some signal-averaging capability is included in ELLIPSE.

Up to five unknown parameters can be obtained using the nonlinear optimization on data acquired at multiple incident angles. Multiple-incident-angle ellipsometry is often plagued by correlations between the parameters[5,6], so the correlation matrix is output with each MAI solution.

II.1.4. Data

Measurement data can be entered as two- or four-zone averages or as ψ, Δ pairs. As data are entered, the incident angle, azimuth settings, and processed ellipsometric angles are stored in buffers, and a count is kept of the number of measurements in the buffers. The buffers can be cleared at any time. The bare-substrate and transparent-film analysis modes operate on only the last data entered; all other modes use the full buffer contents.

All four-zone averages in the session contribute to the calculation of correction factors that are used to correct two-zone averages for systematic errors. The corrections[4,7] for polarizer ellipticity, t_{2P}, and for polarizer/compensator azimuth error, $\eta_{P,C}$, are stored as weighted averages over all four-zone measurements in the session, and are updated each time a new four-zone average is entered. Before the data are analyzed, the data buffers are scanned, and any two-zone data are corrected according to the latest estimates of t_{2P} and $\eta_{P,C}$. The correction factors can be cleared at any time, or their default values can be restored from the default file.

II.2. Starting

Two files, ELLIPSE.EXE and ELLIPSE.DEF, are required to run ELLIPSE. ELLIPSE.DEF contains the default values for the stack and the basic program operation, and must reside in the same subdirectory from which ELLIPSE is run. ELLIPSE.EXE is the VMS executable code; the program is started by typing 'R ELLIPSE' in response to the '$' VMS prompt.

All routines necessary for the operation of ELLIPSE are in the source files ELLIPSE.FOR, ELLSUBS.FOR, and ELLSLAT.FOR. ELLIPSE.FOR holds the main program unit and most of the I/O routines. ELLSUBS.FOR contains the thin-film analysis subroutines, function-evaluation subroutines for the equation solvers, and stack manipulation subroutines. The subroutines in ELLSUBS.FOR have general applicability for optical modelling, so it is suggested that they be kept separate and stored in an object-module library. The third file, ELLSLAT.FOR, contains the SLATEC mathematical library[8] subroutines used by ELLIPSE. To create the executable file, use the following sequence of VMS commands:

```
$ FOR ELLIPSE, ELLSUBS, ELLSLAT
$ LINK ELLIPSE, ELLSUBS, ELLSLAT
```

If you have access to the SLATEC library on your VAX, omit the compilation of ELLSLAT, and replace 'ELLSLAT' in the LINK string with the appropriate symbols for linking to the SLATEC library.

II.3. Menu structure

Three instructions have the same meaning in all menus: 'H' displays the current menu (Figs. 2-4), 'R' initiates a solution, and 'Q' exits the program. In addition, the characters 'D', 'P', and 'M' are used for

Figure 2. Parameter menu

```
Enter:
   1 - to change 1 parameter
   C - to Close log file
   D - to go to Data menu
   E - to Enter a complete stack
   G - to Get default parameters
   H - for Help
   K - to Keep default parameters
   L - to change one Layer
   O - to Open a log file
   M - to change Mode (Mode = SINGLE)
   Q - to Quit
   R - to Run data
   S - to Show stack
   U - to enter Uncertainties (dPhi = 0.020, dP = 0.040, dA = 0.040)
   W - to change Wavelength (wl = 6328.0 Angstroms)
   + - to add one layer
   - - to subtract one layer
```

Figure 3. Mode menu

```
Current mode is SINGLE
Enter:
   0 - for Multiple-angle ns, ks mode
   2 - for Multiple-angle nf, df mode
   3 - for Multiple-angle nf, kf, df mode
   4 - for Multiple-angle nf, df, ns, ks mode
   5 - for Multiple-angle nf, kf, df, ns, ks mode
   A - for Arbitrary n-parameter solution
   B - for Bare-substrate model
   D - to go to Data menu
   H - for Help
   P - to go to Parameter menu
   Q - to Quit
   R - to Run data
   S - for Single-angle nf, df model
```

changing menus so that each menu is accessible directly from the other two.

II.3.1. Parameter input menu

This section of ELLIPSE allows you to enter or modify the various quantities that are relatively constant within a measurement session. These are: the stack parameters and their uncertainties (1, E, L, +, and - instructions, Fig. 2), the uncertainties in the incident angle and azimuth settings (U), and the wavelength (W). The menu choices shown in Fig. 2 are brought to the screen by entering an 'H'. The default file can be read or written from this menu (G and K instructions). In addition, this section can open and close an ASCII disk file to which all output is sent (O and C).

You can examine the stack on the screen with the S instruction; unknowns are marked on the left with asterisks in this display. An example of the stack display is given in the sample ELLIPSE session of Sec. II.4.1.

It is important that you become comfortable with manipulating the stack. The menu and the prompts are self-explanatory -- experiment with them, keeping in mind that it is impossible to cause any real damage because you can always restore the default stack.

II.3.2. Mode menu

Here you select the type of analysis desired, as shown in Fig. 3. The mathematical details of the analyses are described in Sec. IV. The numbered modes are preset MAI modes for convenience and the S and B selections are the single-incident-angle transparent-film and bare-substrate modes mentioned in Sec. II.1. The A instruction is for requesting an arbitrary n-parameter solution; you are prompted for the number of unknowns and their locations in the stack. All of the numbered modes can be duplicated via the A instruction.

The complex index of refraction of a bulk material, or the 'pseudo-index' of an arbitrary surface, is calculated when the mode is set using 'B'. The index and thickness of a film can be obtained with either the S or the 2 command. If the film is transparent ($k = 0$ for layer 1 in the screen display), both commands work but S is recommended. If $k \neq 0$, the 2 mode must be used.

When an initial guess is required to obtain an iterative solution, the values in the stack are used. The initial stack values are preserved unless you instruct ELLIPSE to replace them with the final values.

II.3.3. Data input menu

Entering an 'H' in the data menu results in the output shown in Fig. 4, a list of the available data-entry choices. With the D instruction, you are prompted first for the incident angle and then for a ψ, Δ pair. Input of a '2' results in a prompt for incident angle followed by a

Figure 4. Data menu

```
Enter:
   2 - to enter Two-zone data (*)
   4 - to enter Four-zone data (*)
   C - to enter Calibration data
   D - to enter psi, delta Directly (*)
   G - to Get default parameters from ELLIPSE.DEF
   H - for Help
   M - to change Mode (Mode = SINGLE)
   N - to enter a Name string for data
   P - to go to Parameter menu
   Q - to Quit
   R - to Run data
   V - to View current raw data arrays
   X - to clear data
   Z - to clear corrections

* = One required
```

request for two sets of polarizer and analyzer azimuths that give a null, along with the associated ±45° compensator azimuth. '4' is similar to '2', except that two-zone nulls at both compensator azimuths must be entered. You are allowed to mix data input forms.

The X and Z commands allow you to clear the data buffers and correction factors, respectively. A screen dump of the data arrays is also available (V). The calibrate (C) function prompts you to obtain a four-zone null with the ellipsometer set at 90° incident angle. This measurement serves only to set the values of t_{2P} and $\eta_{P,C}$. The name function (N) allows you to enter a description of the output that follows.

II.4. Basic ELLIPSE operation.

A typical ELLIPSE session follows this pattern:

1. '$ R ELLIPSE'
2. In the parameter menu:.
   a. Open a log file (O).
   b. Adjust the stack parameters to model the sample (E, L, +, -, 1).
   c. Save the stack (K).
3. In the mode menu, choose the desired analytical method.
4. In the data menu:
   a. Identify the sample (N).
   b. Enter data (2, 4, D).
5. Enter 'R'.
6. Repeat from step 2, 3, or 4.

The operations in steps 2-4 can be performed in any order. It is often desirable to save the stack (2c) after setting the mode because the mode information is written to the ELLIPSE.DEF file along with the stack parameters. If the default stack and mode are correct on entry, and no log file is desired, you can omit steps 2 and 3.

To analyze MAI data, or to obtain a least-squares fit to several single-angle measurements made at different locations on one sample, repeat step 4b as many times as necessary. Remember to clear the data buffers (choice X in the data menu) before entering the next set of MAI data.

II.4.1. Example session

Here is a sample terminal session in which measurements pertaining to the SNOS (Silicon/Nitride/Oxide/Semiconductor) memory stack in the SA2999-2 circuit are analyzed. On this occasion, data in the form of ψ, Δ pairs were obtained earlier using a Rudolph AutoEl II ellipsometer. One set of angles was measured on the wafer immediately following a 15-minute oxidation in dry O_2 at 600°C. Another measurement was made on the same wafer following low-pressure chemical vapor deposition of silicon oxynitride. The desired output is the thickness of the oxide layer and the thickness and index of the oxynitride.

In the following, the operator responses are set in boldface, and descriptive material is set off in square brackets. To conserve space, no help requests are shown in the example; refer to Figs. 2-4 for the available command choices.

```
$ R ELLIPSE

ELLIPSE:  Integrated Ellipsometry System, Version 2
Input and Analysis of Ellipsometric Null Data, using
Single or Multiple Incident Angles, and
Two- or Four-Zone Averaging.
H.L. Tardy, SNLA 6224
September, 1988

Enter:                                    [Data menu.  This is the
   2 - to enter Two-zone data (*)          only unrequested help.]
   4 - to enter Four-zone data (*)
   C - to enter Calibration data
   D - to enter psi, delta Directly (*)
   G - to Get default parameters from ELLIPSE.DEF
   H - for Help
   M - to change Mode (Mode = SINGLE)
   N - to enter a Name string for data
   P - to go to Parameter menu
   Q - to Quit
   R - to Run data
   V - to View current raw data arrays
   X - to clear data
   Z - to clear corrections

* = One required

DATIN> p                                  [Go to parameter menu, Fig. 2.]

PARIN> o                                  [Open a log file.]

Enter file name (<= 12 chars. + ext):  L191x.dat

Enter header
Lot DE191A; example session               [Up to 80 characters allowed.]

Enter operator name
HLT

PARIN> s                                  [Examine the stack.]

Stack model:  * - Unknown, starting value
Layer           n                    k                      d
  1    * 1.4640 +/- 0.0050     0.0000 +/- 0.0005   *   10.00 +/-  0.50
  2      2.8000 +/- 0.1000     0.0000 +/- 0.0020       6.00 +/-  1.00
  S      3.8830 +/- 0.0050     0.0180 +/- 0.0020
```

[This is the stack as stored in the ELLIPSE.DEF file. Layer parameters are listed in top-down order. The 'S' indicates the substrate, which is actually layer 3. Values are correct for thin SiO_2 on Si, including a 6 Å interface layer, but the film is too thin to get an accurate solution for top index and thickness (*'s). The following sequence of commands tells ELLIPSE to solve for the oxide thickness and the real part of the substrate index, with the film index fixed at the value shown above.]

```
PARIN> m                                    [Go to mode menu, Fig. 3.]

MODIN> a                                    [Select arbitrary mode.]
Enter number of unknown parameters: 2

Enter 2 solution parameters from the list below:

  1: n( 1)     2: k( 1)     3: d( 1)              [Parameters available
  4: n( 2)     5: k( 2)     6: d( 2)               in the current stack.]
  7: n(sub)    8: k(sub)

3, 7                                        [Solve for SiO2 thickness and
                                             real part of Si index.]

MODIN> d                                    [Back to data menu, Fig. 4.]

DATIN> d                                    [Enter data as ψ, Δ pairs.]

Enter incident angle:  70.

Enter Psi, Delta
10.58, 174.67

DATIN> n                                    [Name the output.]
Enter sample name
Wafer c2, post oxide

DATIN> r                                    [Run.]

Wafer c2, post oxide

Stack model:  * - Unknown, starting value
Layer          n                      k                        d
  1     1.4640 +/- 0.0050     0.0000 +/- 0.0005   *   10.00 +/-  0.50
  2     2.8000 +/- 0.1000     0.0000 +/- 0.0020        6.00 +/-  1.00
  S   * 3.8830 +/- 0.0050     0.0180 +/- 0.0020

Wavelength =  6328.00                       [Thickness units must match
                                             wavelength units.]

Input Uncertainties
dPhi = 0.020, dP = 0.040, dA = 0.040
```

```
Ellipsometric Angles and Residuals
   Phi        Psi    Resid.      Del    Resid.
  70.00     10.580    0.000  174.670    0.000

t2P = 0.019  EtaPC = 0.000

Fit results
  x          Final        Uncert.
d( 1)      10.4678         1.78
n( S)      3.87966        0.754E-02

SSQ = 0.463E-06

INFO =   2:  XTOL criterion met.

Correlation Coefficients
        d( 1)   n( S)
d( 1)   1.000   0.273
n( S)           1.000

Elapsed time =  0.25 sec.
```

[Note that the n(S) result is very close to the literature value[9] of 3.883. Also, the correlation coefficient is low at 0.273. This analysis is similar to fixing the film index and using Δ to obtain the film thickness. The advantage to solving for d(1) and n(S) is that both ψ and Δ are used; the measured substrate index helps to account for grading of the interlayer and roughness at the interface.]

```
Overwrite stack? [y/n] y
```

[The stack is not altered without the user's permission. A positive response results in replacement of d(1) and n(S) in the stack. Because this structure is the substrate for the next measurement, the response is affirmative.]

```
DATIN> p                                  [Back to the parameter menu.]

PARIN> +                                              [Add a layer.]
Insert new layer above which layer? 1                 [Top of the stack.]
Enter n, dn, k, dk, d, dd:
1.92, 0., 0., .0005, 230., 0.
```

[There is no significance to the zero values for the index and thickness uncertainties. Since n and d are the unknown quantities, their uncertainties will be calculated. A zero could be entered here for the thickness because the transparent-film mode does not require an initial guess for thickness.]

```
PARIN> m                                  [To the mode menu.]

MODIN> s                                  [Select transparent-film mode.]
```

```
MODIN> d                                    [To the data menu.]

DATIN> n
Enter sample name
Wafer c2, post oxynitride

DATIN> d

Enter incident angle:  70.

Enter Psi, Delta
15.38, 111.96

DATIN> r

Wafer c2, post oxynitride

Stack model:  * - Unknown, starting value
Layer            n                      k                        d
  1    * 1.9200 +/- 0.0000     0.0000 +/- 0.0005    *  230.00 +/-  0.00
  2      1.4640 +/- 0.0050     0.0000 +/- 0.0005       10.47 +/-  1.78
  3      2.8000 +/- 0.1000     0.0000 +/- 0.0020        6.00 +/-  1.00
  S      3.8797 +/- 0.0075     0.0180 +/- 0.0020

Wavelength =  6328.00

Input Uncertainties
dPhi = 0.020, dP = 0.040, dA = 0.040

Ellipsometric Angles
   Phi        Psi        Delta        t2P        EtaPC
 70.00     15.380     111.960       0.019       0.000
         +/- 0.020  +/- 0.040  +/- 0.119  +/- 0.000

Film-on-Substrate Results
nf =  1.9275 +/- 0.0263
D =  226.99 +/-   3.68
Phase thickness = 1880.02 +/-  25.69

Diagnostics
Im[D] =  0.00  Im[Dphase] =  0.00  F =-0.44E-06  IER =   0


Overwrite stack? [y/n] n

DATIN> q                                    [Close file and exit.]
```

The log file, L191x.dat, that results from this session is not a verbatim log of keystrokes. It contains the title/header block that follows '$ R ELLIPSE', a block containing the file name, header line, operator name,

date, and time, and all ELLIPSE output which occurs between a Run instruction and the next 'DATIN>' prompt.

Additional sample output is given in Appendix A. As an exercise, try to reproduce the examples.

III. Data input and processing; Error correction

ELLIPSE accepts keyboard input of ellipsometric data in three forms: direct input of the ellipsometric angles (ψ, Δ), and input of polarizer and analyzer azimuths in either two or four zones. Data are accumulated in arrays for least-squares analysis. The arrays can contain any combination of (ψ, Δ), two-zone, and four-zone data. In addition, a special four-zone calibration mode is provided for measuring corrections for systematic errors. All zone averaging in ELLIPSE assumes the polarizer-compensator-sample-analyzer (PCSA) configuration[4].

III.1. Four-zone input

Four-zone measurements are converted to (ψ, Δ) pairs through the usual zone relations[4]:

$$\psi = (A_1 - A_3 + A_2 - A_4)/4 \ , \qquad 2$$

$$\Delta = (P_1 + P_3 - P_2 - P_4)/2 \ , \qquad 3$$

where P_i and A_i represent the polarizer and analyzer azimuths, respectively, in zone i. In the absence of cell windows, all systematic errors due to azimuth errors and component imperfections cancel out to first order in Eqs. (2) and (3)[4,7], and the resulting uncertainties in (ψ, Δ) are purely random. Given the uncertainties (δP, δA) in (P, A) readings, uncertainties in ψ and Δ are computed by the standard propagation-of-errors method;

$$\delta\psi = \delta A/2, \ \delta\Delta = \delta P. \qquad 4$$

In addition to the ellipsometric angles, four-zone data contain information about the systematic errors in the system. The polarizer ellipticity, t_{2P}, is obtained through the analyzer azimuths[4,7] via

$$A_1 - A_3 - A_2 + A_4 = 4t_{2P} \sin 2\psi \ . \qquad 5$$

A single parameter, $\eta_{P,C}$, describes the polarizer and compensator divided-circle errors ΔP and ΔC and the relative attenuation in the polarizer t_{1P}. $\eta_{P,C}$ is obtained from the polarizer azimuths[4,7] as

$$\eta_{P,C} = -(P_1 + P_2 + P_3 + P_4)/2 = 2(t_{1P} + \Delta P - \Delta C) \ . \qquad 6$$

Because $\eta_{P,C}$ and t_{2P} are used to correct two-zone data for systematic errors, uncertainty in the measurement of t_{2P} and $\eta_{P,C}$ contributes to

the uncertainty of ψ and Δ obtained from two-zone measurements. The uncertainties δt_{2P} and $\delta \eta_{P,C}$ are given by

$$\delta t_{2P} = \delta A/(2\sin 2\psi) \ , \qquad 7$$

and

$$\delta \eta_{P,C} = \delta P. \qquad 8$$

The individual nulls can be entered in any order, because ELLIPSE uses the analyzer and compensator values to deduce the zone. For example, if $C = -45°$ and $0° \leq A \leq 90°$, the null is from zone 1. The only restriction is that the even- and odd-zone data must be entered in pairs.

III.2. Two-zone input

ψ and Δ are obtained from two-zone azimuths through[4,7]

$$\psi = (A_1 - A_3)/2 - t_{2P} \sin 2\psi = (A_2 - A_4)/2 + t_{2P} \sin 2\psi \qquad 9$$

and

$$\Delta = P_1 + P_3 + \eta_{P,C} = - P_2 - P_4 - \eta_{P,C} \ . \qquad 10$$

ψ and Δ thus represent two-zone averages corrected to first order for all systematic errors except those due to cell windows. ELLIPSE calculates uncertainties ($\delta\psi$, $\delta\Delta$) using

$$\delta\psi^2 = \delta A^2/2 + (\delta t_{2P} \sin 2\psi)^2, \qquad 11$$

and

$$\delta\Delta^2 = 2\delta P^2 + \delta \eta_{P,C}^2 \ . \qquad 12$$

III.3. Calibration

From Eq. (7), it is clear that the most accurate measurement of t_{2P} is made when $\psi = \pi/4$. For this reason, ELLIPSE contains a special acquisition mode in which a four-zone measurement is made with the instrument in the straight-through configuration ($\phi = 90°$, no sample in the beam). The disadvantage of this method is that $\eta_{P,C}$ contains no information about azimuth errors induced by sample misalignment[7].

Once values of t_{2P} and $\eta_{P,C}$ are made available through a four-zone measurement, all subsequent two-zone data are corrected accordingly. As more four-zone data become available, the program updates t_{2P} and $\eta_{P,C}$ by maintaining running weighted averages and standard deviations of the two parameters. If several two-zone measurements have been stored for least-squares analysis, they are re-corrected using the latest values of t_{2P} and $\eta_{P,C}$.

IV. Analysis

ELLIPSE analyzes ellipsometric data in terms of either bare-substrate or film-on-substrate models. For MAI data, a Levenberg-Marquardt solution[6] can be obtained for any combinations of model parameters. All solutions are returned with uncertainties.

IV.1. Bare substrate

Given the ambient index n_o, the incident angle ϕ, and a measurement of $\rho = \tan\psi\, e^{i\Delta}$, the complex index, $\tilde{n}_s = n_s - i\, k_s$, of a bare substrate is given by[4]

$$\tilde{n}_s = n_o \tan\phi \left[1 - \frac{4\rho}{(1+\rho)^2} \sin^2\phi \right]^{1/2} . \qquad 13$$

The solution is obtained in ELLIPSE by a call to subroutine SUBSTRATE. This mode is unique in that it cannot operate on a stack with a nonzero number of films. Invoking the bare-substrate mode resets the stack to contain only a substrate. The user is first warned that the stack will be overwritten and is given the opportunity to stop the overwrite. Because this is an analytical solution, SUBSTRATE ignores the initial values of n_s and k_s.

IV.2. Transparent film on substrate

ELLIPSE differs radically from McCrackin[1] in its treatment of the TFS problem. Charlot and Maruani[3] decoupled the ellipsometric equations for film index, n_f, and thickness, d_f, through the argument which follows. In using a variation of their technique, ELLIPSE obtains the solution more reliably and with much less dependence on a good initial guess than does the NBS code.

Charlot and Maruani began with the quadratic form of the ellipsomety equation for TFS:

$$a(n) + b(n)\, X + c(n)\, X^2 = 0, \qquad 14$$

where

$$X = e^{-2\pi i d/D} , \qquad 15$$

and D is the phase thickness given by

$$D = \frac{\lambda}{2n \cos\phi_n} . \qquad 16$$

The coefficients of Eq. (14) are

$$a(n) = f_{op} - \rho\, f_{os} , \qquad 17$$

$$b(n) = r_{1p} - \rho\, r_{1s} + f_{op}\, f_{os}\, (r_{1s} - \rho\, r_{1p}) , \qquad 18$$

and

$$c(n) = r_{1p}\, r_{1s}\, (f_{os} - \rho\, f_{op}) \quad . \qquad 19$$

In the above expressions, f_{ox} is the ambient-film Fresnel coefficient for x polarization, r_{1x} is the film-substrate Fresnel reflectance for x polarization, and ϕ_n is the incident angle within the film as given by Snell's law. All of these quantities are calculated assuming that the film index is n; Eq. (14) is satisfied when $n = n_f$ and $d = d_f$.

Noting that the complex conjugate of Eq. (14) is a second equation for X, and that $|X| = 1$ for transparent films, Charlot and Maruani derived an equation with only n unknown:

$$G(n) = (|a|^2 - |c|^2)^2 - |a^*\, b - c\, b^*|^2 = 0 \quad . \qquad 20$$

Only the single nonlinear real equation for n_f need be solved iteratively, and then d_f is obtained from Eq. (15). The form of Eq. (14) does not restrict this analysis to a single film on a substrate. The 'substrate' can consist of a stack of films, as long as the appropriate Fresnel reflectances are inserted in Eqs. (18) and (19).

The Charlot-Maruani equation as stated in Ref. 3 (Eq. (20), above) has solutions at n_o and n_f, and an additional minimum may exist near $n = n_s$ (Fig. 5). The second minimum is explained by noting that if the film is very thin, then $a = 0$ is almost satisfied at $n = n_s$. Also at $n = n_s$, $b = c = 0$ because there is no reflection at the film/substrate interface. G(n) does not have a solution at $n = n_s$ (unless there is actually no film present), but the minimum near n_s can create problems with numerical A poor initial guess can result in convergence to the false $n = n_o$ solution or failure to converge at all. Fortunately, the spurious solution can be factored out of G(n). Substituting the definitions for the Fresnel coefficients in Eqs. (17) and (19) yields

$$a(n) = \frac{(n_o^2 - n^2)\, [n_o \sin^2\phi\, (1 - \rho) - n \cos\phi\, \cos\phi_n (1 + \rho)]}{n\, (n_o \cos\phi + n \cos\phi_n)\, (n \cos\phi + n_o \cos\phi_n)} \qquad 21$$

and

$$c(n) = r_{1p}\, r_{1s}\, \frac{(n_o^2 - n^2)\, [n_o \sin^2\phi\, (1 - \rho) + n \cos\phi\, \cos\phi_n (1 + \rho)]}{n\, (n_o \cos\phi + n \cos\phi_n)\, (n \cos\phi + n_o \cos\phi_n)} \quad . \qquad 22$$

Defining

$$a'(n) = a(n)/(n_o^2 - n^2) \qquad 23$$

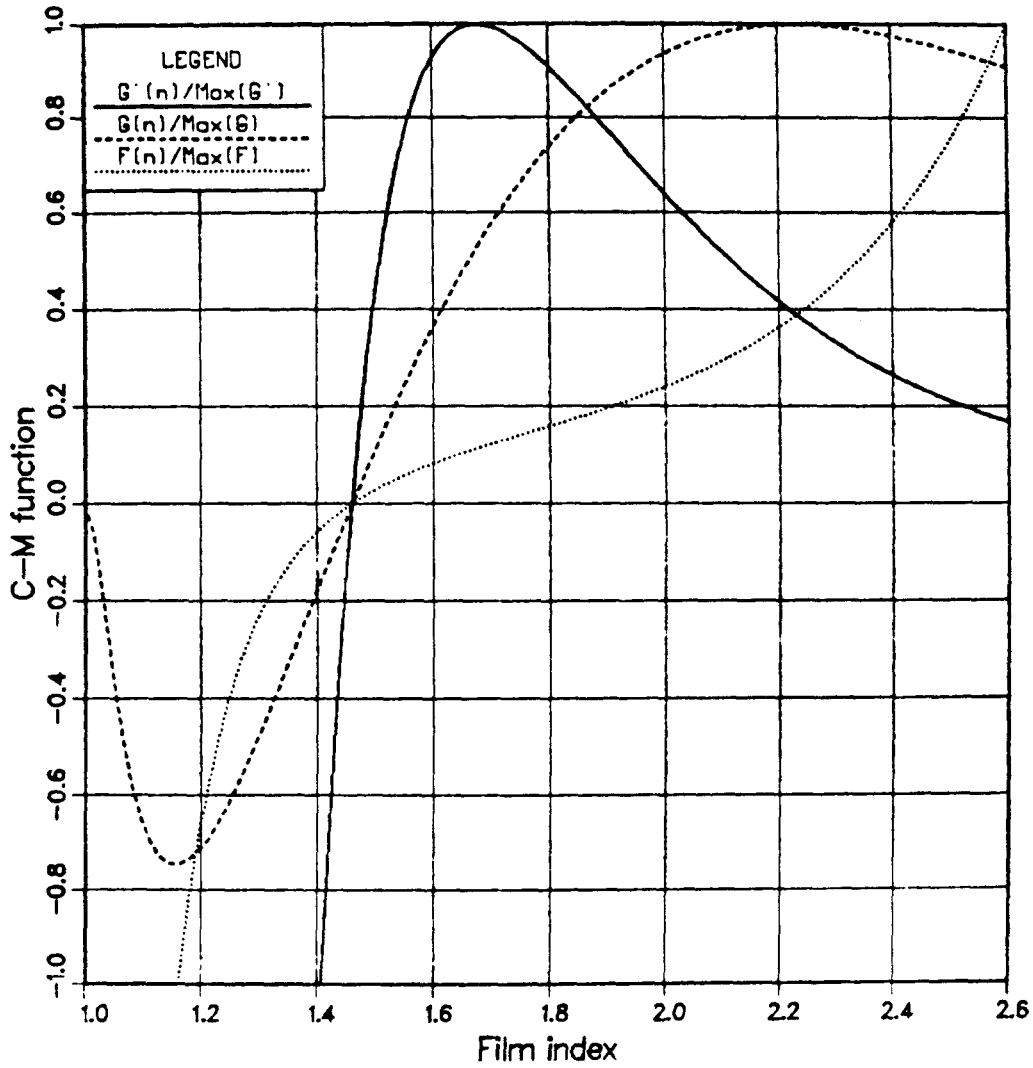

Figure 5. Variations of the Charlot-Maruani function for 200 Å of SiO_2 on Si with HeNe light (λ = 6328 Å) at 70° incident angle. Each curve is normalized to its highest value in the interval $1.46 < n < 2.6$; the normalization values are Max(G') = 1.07×10^{-4}, Max(G) = 6.57×10^{-4}, and Max(F) = 5.72×10^{-3}.

and

$$c'(n) = c(n)/(n_o^2 - n^2) \ , \qquad 24$$

we obtain a new equation for n_f:

$$G'(n) = (n_o^2 - n^2)^2 \ (|a'|^2 - |c'|^2)^2 - |a'^* \ b - c' \ b^*|^2 = 0 \ . \qquad 25$$

G'(n) does not have $n = n_o$ as a solution, but does retain the additional minimum near $n = n_s$ for all film thicknesses (Fig. 5)

Problems due to the possible presence of a minimum at $n = n_s$ can be avoided by multiplying G'(n) by a function that increases strongly enough to mask the minimum. Thus, ELLIPSE solves

$$F(n) = G'(n) \ e^{(n^2 - n_o^2)} = 0 \qquad 26$$

to find the film index. For all practical cases, this function is monotonic over a very wide region around the solution (Fig. 5). The program can start with an arbitrarily-chosen initial guess (e.g., n = 1.5) and convergence is virtually guaranteed. Only the most simple-minded Newton's method equation solver is required. This is implemented in subroutine NEWT.

After solving Eq. (26) for n_f, ELLIPSE calls subroutine THICKN to obtain d_f from X through[3]

$$X = \frac{(n_o^2 - n_f^2) \ (|a'|^2 - |c'|^2)}{c' \ b^* - a'^* \ b} \qquad 27$$

and

$$d_f = \frac{i \ D}{2\pi} \ \ln X \qquad 28$$

NEWT takes the value in the stack as the initial guess for n_f and iterates until two successive trial solutions are equal to within six significant digits. Because d_f is obtained analytically, the stack value of d_f is ignored. Note that the Charlot-Maruani algorithm does not guarantee $\mathrm{Im}(d_f) = 0$. $\mathrm{Im}(d_f)$ and $\mathrm{Im}(D)$ are output by ELLIPSE, along with the value of F(n), for diagnostic purposes. Failure to arrive at the correct index and thickness usually indicates that the film is very thin, or that its thickness is close to a full phase.

The transparent-film mode is demonstrated in Sec. II.4.1. Appendix A.1 is a sample output demonstrating the same mode on a low-index film. The sample was a Si wafer coated with a Støber-process sol gel film. An uncoated area of the wafer was measured first to estimate the thickness of the native oxide on the Si wafer.

IV.3. Multiple-angle-of-incidence analysis

A full discussion of multiple-incident-angle (MAI) ellipsometry is beyond the scope of this report. Detailed discussions of the technique can be found in Refs. 4-6. Briefly, MAI exploits the variation of ψ and Δ with incident angle to obtain enough information about the optical stack to allow simultaneous determination of more than two stack parameters, or to improve the ellipsometer's ability to measure both the index and thickness of ultrathin films. Frequently MAI analysis is hindered by correlations between parameters: if the ellipsometric sensitivities to two of the unknowns do not vary much with incident angle, then little new information is gained by making measurements at several angles and the two unknowns cannot be determined simultaneously with precision.

One situation where MAI can be helpful is in measuring thicker films when the number of periods is unknown. The thickness ambiguity inherent in single-angle measurements is completely resolved by measuring ψ and Δ at two or more incident angles. The technique is effective unless the film thickness is within about 100 Å of the phase thickness.

MAI data can be analyzed in ELLIPSE for up to five unknown parameters. Solutions are obtained with an unweighted Levenberg-Marquardt algorithm[8], which minimizes the function

$$SSQ = \sum_{i=1}^{N} [(\psi(\phi_i) - \psi_i)^2 + (\Delta(\phi_i) - \Delta_i)^2] , \qquad 29$$

where ψ_i and Δ_i are the i'th measurement, and $\psi(\phi_i)$ and $\Delta(\phi_i)$ are the model values at incident angle ϕ_i. The reported uncertainties are 68% confidence limits derived from the inverse-Hessian matrix[5], and hence represent only the random uncertainty in the results.

The same numerical technique is used to solve two-parameter problems at a single incident angle. This is useful when the sample fits neither the bare-substrate nor the transparent-film model (See Sec. II.4.1. It is also useful for obtaining least-squares fits to ψ and Δ obtained at several locations on a sample at a single incident angle.

The ELLIPSE output from a multiple-angle measurement of a sol gel-coated silvered Si wafer is shown in Appendix A.2. This result illustrates the parameter-correlation problem inherent to MAI measurements. A near-unit off-diagonal element in the correlation matrix means that the measurement cannot simultaneously determine the two indicated parameters with high precision.

The SLATEC Levenberg-Marquardt routine, DNLS1, terminates execution on any of four conditions. The FTOL parameter sets the relative precision of the final sum of squares. The XTOL parameter sets the relative precision of the solution vector. The GTOL parameter tests for orthogonality between the residual vector and the columns of the Jacobian matrix. Termination also occurs when the residual vector is evaluated

more than MAXFEV times. In ELLIPSE, all termination parameters are set to the values recommended in the SLATEC documentation: FTOL and XTOL are set to the square root of the machine precision, GTOL is set to zero, and MAXFEV is set to 100*(NPAR+1). The partial derivatives (Jacobian matrix) required by DNLS1 are calculated analytically[10] in MATRD.

V. Error analysis

ELLIPSE performs explicit error analysis on all single-angle measurements using a generalization of the technique described in Ref. 2. The ellipsometric angles are described by the column vector **e**:

$$\mathbf{e} = \begin{pmatrix} \psi \\ \Delta \end{pmatrix} . \qquad 30$$

A column vector **x** contains the two unknown parameters, and another column vector $\delta\mathbf{f}$ is used to denote the uncertainties in all of the fixed model parameters;

$$\delta\mathbf{f}^T = (\delta k_1,\ \delta d_1,\ \ldots,\ \delta k_s,\ \delta\phi,\ \delta\lambda) . \qquad 31$$

We also define the Jacobian matrix, J, where

$$J_{ij} = \frac{\partial e_i}{\partial x_j} , \qquad 32$$

and the sensitivity matrix F, where

$$F_{ij} = \frac{\partial e_i}{\partial f_j} . \qquad 33$$

Note that there is mathematically no difference between J and F; both are matrices of partials with respect to model parameters. The distinction is made here only to facilitate error analysis. Using these definitions, we can write

$$\delta\mathbf{e} = \mathbf{J}\bullet\delta\mathbf{x} + \mathbf{F}\bullet\delta\mathbf{f} . \qquad 34$$

Finding the uncertainties in the unknowns is then a matter of inverting the Jacobian:

$$\delta\mathbf{x} = \mathbf{J}^{-1} \bullet (\delta\mathbf{e} - \mathbf{F}\bullet\delta\mathbf{f}) . \qquad 35$$

In practice, the terms in Eq. (35) are added either in magnitude or in quadrature, depending on whether a particular model uncertainty is systematic or random.

Eqs. (30-35) represent standard error analysis methodology; the difficulty in applying this technique to ellipsometry comes in evaluating the derivatives. In Ref. 2, the partial derivatives were taken by hand for the bare-substrate and single-film-on-substrate models. Humlicek[10] showed how all partials of a multilayer system can be calculated simply during the course of the reflectance calculation when the matrix representation of the stack is used. In subroutine UNIERR, ELLIPSE uses subroutine MATRD to obtain the s- and p-reflectance of the stack and the

reflectance partial derivatives, then calls DVRHO to convert the reflectance derivatives to ψ and Δ derivatives. Subroutine DVSOL computes $J^{-1} \bullet F$ and J^{-1}. ELLIPSE treats the ψ and Δ uncertainties as random, and all others as systematic.

The user should be aware that the uncertainties reported for single-angle measurements do not reflect the possibility that the data were analyzed under false assumptions (isotropic materials, planar interfaces, etc.). For example, an analysis made with the assumption of an interlayer between the film and the substrate could yield film parameters outside the uncertainty envelope obtained in the absence of the interlayer.

Acknowledgment

The author is grateful to S.C. Everist for providing the data used in the sample session.

References

1. F.L. McCrackin, 'A FORTRAN Program for Analysis of Ellipsometer Measurements,' NBS Tech. Note 479 (U.S. Gov't. Printing Office, Washington, DC, 1969).
2. D. Chandler-Horowitz, 'Analytical Analysis of Ellipsometric Errors,' NBS Spec. Pub. 400-78 (U.S. Dept. of Commerce, Washington, DC, 1986).
3. D. Charlot and A. Maruani, 'Ellipsometric Data Processing: An Efficient Method and an Analysis of the Relative Errors,' Appl. Opt. **24**, 3368 (1985).
4. R.M.A. Azzam and N.M. Bashara, Ellipsometry and Polarized Light (North-Holland, New York, 1977).
5. J. Humlicek, 'Sensitivity Extrema in Multiple-Angle Ellipsometry,' JOSA A **2**, 713 (1985).
6. M.M. Ibrahim and N.M. Bashara, 'Parameter-Correlation and Computational Considerations in Multiple-Angle Ellipsometry,' JOSA **61**, 1622 (1971).
7. R.M.A. Azzam and N.M. Bashara, 'Unified Analysis of Ellipsometry Errors Due to Imperfect Components, Cell-Window Birefringence, and Incorrect Azimuth Angles,' JOSA **61**, 600 (1971).
8. K.H. Haskell, W.H. Vandevender, and E.L. Walton, 'The SLATEC Common Mathematical Subprogram Library: SNLA Implementation,' Sandia Natinal Laboratories Report SAND80-2792 (1980). Available from National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161, (703) 487-4650.
9. D.E. Aspnes and A.A. Studna, 'Dielectric functions and optical parameters of Si, Ge, GaP, GaAs, GaSb, InP, InAs, and InSb from 1.5 to 6.0 eV,' Phys. Rev. B **27**, 985 (1983).
10. J. Humlicek, 'Evaluation of Derivatives of Reflectance and Transmittance by Stratified Structures and Solution of the Reverse Problem of Ellipsometry,' Opt. Acta **30**, 97 (1983).
11. Digital Equipment Corporation, Programming in VAX FORTRAN, Software Version V4.0 (Digital Equipment Corporation, Maynard, MA, 1984).

Appendix A. Sample output

A.1. Sample transparent-film calculation.

Sample '17-10', coated area.

Stack model: * - Unknown, starting value

Layer	n	k	d
1	* 1.2000 +/- 0.0000	0.0000 +/- 0.0005	* 1400.00 +/- 0.00
2	1.4640 +/- 0.0050	0.0000 +/- 0.0005	19.07 +/- 1.86
3	2.8000 +/- 0.1000	0.0000 +/- 0.0020	6.00 +/- 1.00
S	3.9079 +/- 0.0078	0.0180 +/- 0.0020	

Wavelength = 6328.00

Raw Data

Phi	P	A	P'	A'	C
75.00	93.58	42.14	3.74	320.69	45.00
75.00	86.20	317.33	355.50	38.93	-45.00

Input Uncertainties
dPhi = 0.020, dP = 0.040, dA = 0.040

Ellipsometric Angles

Phi	Psi	Delta	t2P	EtaPC
75.00	40.763	82.190	0.038	0.090
	+/- 0.020	+/- 0.040	+/- 0.020	+/- 0.028

Film-on-Substrate Results
nf = 1.2380 +/- 0.0024
D = 1402.89 +/- 8.01
Phase thickness = 4085.67 +/- 8.09

Diagnostics
Im[D] = 0.00 Im[Dphase] = 0.00 F =-0.11E-09 IER = 0

A.2. Sample multiple-incident-angle calculation.

Sol gel film on Ag.

Stack model: * - Unknown, starting value

Layer	n	k	d
1	* 1.4000 +/- 0.0000	0.0000 +/- 0.0005	* 2000.00 +/- 0.00
S	* 0.1350 +/- 0.0000	* 4.0000 +/- 0.0000	

Wavelength = 6328.00

Raw Data

Phi	P	A	P'	A'	C
80.00	351.97	42.98	81.78	313.88	45.00
75.00	96.48	313.88	186.65	43.90	45.00
70.00	198.29	43.20	108.18	314.17	45.00
65.00	117.03	314.97	207.10	43.58	45.00
60.00	213.79	44.04	123.73	314.78	45.00
55.00	128.43	315.04	218.49	224.44	45.00
50.00	221.67	224.63	131.64	315.12	45.00

Input Uncertainties
dPhi = 0.020, dP = 0.040, dA = 0.040

Ellipsometric Angles and Residuals

Phi	Psi	Resid.	Del	Resid.
80.00	44.550	0.154	-73.750	-0.004
75.00	45.010	-0.397	-103.130	0.001
70.00	44.515	0.058	-126.470	0.032
65.00	44.305	0.273	-144.130	-0.112
60.00	44.630	-0.018	-157.520	0.099
55.00	44.700	-0.039	-166.920	0.051
50.00	44.755	-0.040	-173.310	-0.072

t2P = 0.019 EtaPC = 0.000

Fit results

x	Final	Uncert.
n(S)	0.826724E-01	0.178E-01
k(S)	3.38150	0.208
n(1)	1.38355	0.206E-01
d(1)	1963.94	70.0

SSQ = 0.172E+00

INFO = 1: FTOL criterion met.

Correlation Coefficients

	n(S)	k(S)	n(1)	d(1)
n(S)	1.000			
k(S)	0.561	1.000		
n(1)	-0.573	-0.977	1.000	
d(1)	0.569	0.998	-0.987	1.000

Elapsed time = 37.43 sec.

Appendix B. ELLIPSE symbolic names

B.1. ELLIPSE variables

ELLIPSE does not assume a specific length unit. All thicknesses and wavelengths must be entered in the same units; if the wavelength is entered in Ångstrom units, then all stack thicknesses must be entered as Ångstroms and all thickness results will be reported in Ångstroms. All angles must be entered in degrees.

B.1.1. Array sizes.

NDMAX: Maximum number of measurements that can be stored in the data buffers. NDMAX = 20.
NPMAX: Multiple-incident-angle measurements can be solved for up to NPMAX = 5 parameters.
NFMAX: The optical stack can consist of between 0 and NFMAX = 20 films.

NDMAX, NPMAX, and NFMAX are set throughout the program using FORTRAN PARAMETER statements, so that the size of the arrays is fairly easy to adjust.

B.1.2. Optical model variables

NFLM: INTEGER
The number of layers in the stack. NFLM ≤ NFMAX.
STACK(I): REAL*8
Array of stack parameters. If J is the layer index (J = 1 for the top layer), then STACK(3J-2) = Re(N(J)), STACK(3*J-1) = Im(N(J)), STACK(3J) = D(J). STACK(3NFLM+1) = n_s, STACK(3NFLM+2) = k_s, STACK(3NFLM+3) = n_o. The ambient index is not placed in order with the other parameters so that the order of indices follows that used for partial derivatives.
USTACK(I): REAL*8
Array of stack uncertainties. The order is the same as STACK, except USTACK(3NFLM+3) = incident angle uncertainty, USTACK(3NFLM+4) = wavelength uncertainty.
N(I): COMPLEX*16
Array of complex refractive indices, as used by MATRT and MATRD. N(1) = n_o, N(NFLM+2) = n_s- ik_s.
D(I): REAL*8
Array of film thicknesses, as used by MATRT and MATRD. D(1) = top layer thickness. No unit is explicitly attached to the thicknesses. ELLIPSE assumes that the same unit is used for thicknesses and wavelengths.
R(M): COMPLEX*16
Fresnel reflectance of optical stack at polarization M.
DR(M, I): COMPLEX*16
Partial derivative of reflectance at polarization M with respect to

STACK element I. Mapping of index I to physical parameters corresponds to USTACK.

DEA(L, I): REAL*8
Partial derivatives of ellipsometric angles with respect to STACK element I. L = 1 corresponds to ψ, L = 2 corresponds to Δ.

IS(K): INTEGER
Array of unknown parameter indices. K ≤ NPMAX. STACK(IS(K)) contains the starting value of the K'th unkown.

X(K), DX(K): REAL*8
Arrays of unknowns and uncertainties.

WL: REAL*8
Wavelength. No unit is explicitly attached to the wavelength: the only requirement is that the same unit be used for all film thicknesses and the wavelength.

B.1.3. Data variables

NDAT: INTEGER
Number of measurements in current data buffers. NDAT ≤ NDMAX.

PHI(I): REAL*8
Array of incident angles.

DPHI: REAL*8
Instrumental incident angle uncertainty.

P(J, I), A(J, I): REAL*8
Arrays of polarizer, analyzer azimuths. J is a zone index: J = 1, 2 correspond to one two-zone pair and J = 3, 4 correspond to the other two-zone pair. I is the data index. For data input directly as ψ, Δ pairs, P(1, I) = ψ and P(2, I) = Δ.

C(J, I): REAL*8
Array of compensator azimuths. C(1, I) is the azimuth for the first two-zone pair (J = 1, 2 in the P and A arrays) and C(2, I) is the azimuth for the second pair (J = 3, 4 in the P and A arrays). C is used to convey information about the form of data input. C(1, I) = C(2, I) = 0 when I > NDAT; i.e. there is no data for the I'th measurement. C(2, I) = 0 indicates a two-zone measurement; C(1, I) = ±45° is required. C(2, I) = 999 when data are entered as ψ and Δ. For four-zone input, C(1, I) and C(2, I) are required to have 45° magnitude and opposite signs.

DP, DA: REAL*8
Instrumental polarizer and analyzer uncertainties.

EANG(L, I), UEA(L, I): REAL*8
Arrays of ψ and Δ values and their uncertainties: L = 1 indicates ψ and L = 2 indicates Δ. The uncertainties are stored in an array because two- and four-zone measurements have potentially different uncertainties.

T2P, ETAPC: REAL*8
Correction factors t_{2P} and $\eta_{P,C}$, obtained from four-zone nulls and used to correct two-zone nulls for systematic errors due to polarizer ellipticity (T2P) and polarizer and compensator azimuth errors (ETAPC). T2P and ETAPC are weighted averages from all four-zone data entered (not just current data buffers), and are updated whenever a new four-zone null is input.

B.2. ELLIPSE Subroutines

B.2.1. Low-level routines

These subroutines make no calls.

ADDLAYER: Adds one layer to the stack, at a user-input location. First ADDLAYER requests the position in the stack of the new layer, then requests layer parameters and uncertainties. The film count is incremented and the STACK and USTACK arrays are changed to include the new layer.

AZCON: Converts all two-zone data in the current data buffers to ψ and Δ, correcting for systematic errors using the current values of T2P and ETAPC.

BUFLOAD: Loads a single set of azimuth measurements into to the P, A, and C buffers.

CALOUT: Output of four-zone calibration results.

CHGLAYER: Changes the STACK and USTACK arrays for one layer at a user-input stack position.

DATHELP: Help for the data input menu (DATIN).

DCHECK: Checks data and parameters before allowing return to ELLIPSE. Requires NDAT > 0, STACK(2) = 0 if the model is TFS, and that NDAT is large enough to allow unique solution of MAI models.

DVRHO: Accepts reflectance derivatives (computed in MATRD) and returns partials of ψ and Δ with respect to all stack parameters, ϕ, and λ.

DVSOL: Inverts the two-unknown Jacobian to calculate the partial derivatives of the unknown parameters with respect to ψ, Δ, and the known parameters. Requires partial derivatives of ψ and Δ as calculated by DVRHO.

ELOUT: Output of last ψ and Δ (EANG(L, NDAT)) along with TFS results.

FOURZONE: Processes four-zone azimuths to obtain ψ and Δ along with their uncertainties. Also updates correction factors t_{2p} and $\eta_{P,C}$.

GETAZ: Accepts keyboard input of polarizer, analyzer, and compensator azimuths for a two- or four-zone null.

GET_DEF: Reads file ELLIPSE.DEF to get default model parameters.

GET_IS: Accepts keyboard input of the STACK indices of the two unknown parameters for which the optical model is to be solved.

HEADOUT: Output of header boilerplate to specified logical unit.

LOGCL: Writes the system time and closes the log file.

LOGOP: Opens a log file and writes header information.

MAOUT: Output of EANG array and residuals, along with MAI results.

MATRD: Given all necessary optical model information, MATRD calculates R_p, R_s and their partial derivatives with respect to all model parameters, using the Humlicek matrix formulation.

MATRT: Similar to MATRD, but does not compute the derivatives.

MODEHELP: Help for the mode menu (MODIN).

MODLOAD: Separates STACK into a complex array of indices of refraction and a real array of thicknesses for use in MATRD or MATRT.

PARHELP: Help for the parameter menu (PARIN).

PUT_DEF: Writes current model parameters to file ELLIPSE.DEF to be used as defaults.

RAWOUT: Output of raw data to specified logical unit. Prints PHI, P, A, and Carrays along with WL, DPHI, DP, and DA.

REARR: Rearranges an upper-triangular matrix according to a given pivot vector.

SSTACKIN: Accepts keyboard input of a full set of stack parameters and stores them in STACK.

STACKIN: Similar to SSTACKIN, but also gets USTACK.

STDUMP: Outputs STACK and USTACK to a specified logical unit.

STRING: Given a vector of stack indices, STRING creates a vector of character labels. For example, a value of 3 in the input vector results in an output of 'd(1)' if NFLM $\geq$ 1.

SUBLAYER: Removes a user-input layer from the stack, modifying STACK, USTACK, and NFLM.

SUBSTRATE: Analytical solution of the bare-substrate model. SUBSTRATE operates only on the last data input (PHI(NDAT), EANG(L, NDAT)).

SUBOUT: Output of EANG(L, NDAT) along with bare-substrate results.

THICKN: Calculates the film thickness in the TFS model following solution of the Charlot-Maruani equation for the film index.

TWOZONE: Processes azimuths to obtain ψ and Δ for a two-zone null, correcting for systematic errors using the current values of T2P and ETAPC.

XTXUP: Computes matrix product $\mathbf{X}^T\mathbf{X}$ for an upper-triangular matrix **X**.

B.2.2. Intermediate-level routines.

These subroutines call only the low-level subroutines.

ANGS: Calls MODLOAD and MATRT to calculate ψ and Δ from ϕ, λ, and STACK.

ANGSD: Calls MODLOAD, MATRD, and DVRHO to calculate ψ, Δ and their partial derivatives from ϕ, λ, and STACK.

CMNEWT: Function subroutine that evaluates the modified Charlot-Maruani function. Calls MATRT. Uses only the last data input (PHI(NDAT), EANG(L, NDAT)).

CORRLS1: Computes correlation matrix given DNLS1 output. Calls XTXUP and REARR, and SLATEC library routines DSPCO and DSPDI.

MODELIN: Calls STACKIN to get STACK and USTACK arrays, an prompts for input of ϕ and λ.

PARCHG: Changes one element in STACK and USTACK arrays. Calls GET_IS and STRING.

SMODELIN: Same as MODELIN, but calls SSTACKIN and so does not get the USTACK array.

B.2.3. High-level routines

These subroutines call low- and intermediate-level routines.

FCNG: Computes a vector of residuals ψ(PHI(I)) - EANG(1, I), Δ(PHI(I)) - EANG(2, I) for use in multiple-incident-angle solutions. Also computes the Jacobian for use in SLATEC routine DNLS1. Calls ANGS, ANGSD.

UNIERR: Calls ANGSD and DVSOL, and does the required matrix multiplication with USTACK and UEA to produce the total uncertainties, DX(K), in the two unknown parameters, X(K), defined by the IS vector.

B.2.4. Menu subroutines

These call low- and intermediate-level routines, and other menu routines.

DATIN: Main input menu routine. Gets all keyboard input of data variables, and calls the other two menu routines (MODIN and PARIN). All data passes through DATIN before being returned to ELLIPSE. Calls BUFLOAD, CALOUT, DATHELP, DCHECK, FOURZONE, GETAZ, LOGCL, MODIN, PARIN, and RAWOUT.

In addition to its straightforward data-entry function, DATIN serves as a filter for all input acquired by the program. Although a 'Run' instruction can be issued from any of the three menus, the data passes through DATIN before it is submitted for a solution. At any time in the session, entry of an 'R' initiates the following sequence of events: 1) Program control returns to DATIN if it is not already there. 2) The data are checked to see that a sufficient number of measurements were made to allow a unique solution for the requested number of parameters, and if a transparent-film solution is requested, that STACK contains a purely-real index of refraction for the top film. If either test fails, an error message is written to the screen and the user is prompted to correct the error. 3) Control returns to the main program. 4) The contents of STACK and USTACK are written to the screen and to the log file, if open. 5) The contents of the azimuth data buffers are written to the screen and the log file. 6) The azimuths are processed and all two-zone nulls are corrected. 7) Control passes to the appropriate code for the requested analysis mode. 8) The analysis results are written to the screen and log file. 9) Control is sent back to DATIN.

MODIN: Analysis mode input menu. Called only from DATIN. Calls MODEHELP and GET_IS.

PARIN: Parameter input menu. Obtains all keyboard and file input of model parameter variables. Called only from DATIN. Calls ADDLAYER, CHGLAYER, GET_DEF, LOGCL, LOGOP, PARHELP, PARCHG, PUT_DEF, STACKIN, STDUMP, and SUBLAYER.

B.2.5. Library routines

ELLIPSE uses several SLATEC library routines. Documentation for these can be found in Ref. 8.

DNLS1: Solves the nonlinear least-squares problem.

DSPCO and DSPDI: Called in CORRLS1 to invert the Hessian matrix, which results in the table of correlation coefficients reported with MAI output.

UERSET: Sets the severity level at which SLATEC begins reporting errors.

XSETF: Controls reporting of SLATEC error messages.

Appendix C. Deviations from ANSI FORTRAN-77

ELLIPSE uses the following VAX extensions to FORTRAN-77[11], which will generate compile-time errors under ANSI FORTRAN:

1. Some symbolic names are longer than six characters (MODELIN), and some include the underscore character (GET_IS).
2. The exclamation point character is used liberally to indicate trailing comments.
3. Data types are specified using the notation 'REAL*8' rather than 'DOUBLE PRECISION'.
4. The COMPLEX*16 data type is used throughout.
5. The DO WHILE statement is used, and all DO loops terminate with the END DO statement rather than a CONTINUE statement.
6. Variable format expressions (e.g., <I>F8.2) are included.

In addition, input prompts which require short responses (e.g. the menu prompts) make use of the '$' carriage-control character. This suppresses the carriage return after the prompt and allows the response to appear on the same line as the prompt.

All trigonometric and inverse-trigonometric functions in ELLIPSE require (or return) angles in degrees. In other words the VAX intrinsic functions COSD, SIND, ATAND, and ATAN2D are used in place of COS, SIN, ATAN, and ATAN2. The VAX intrinsic functions DREAL and DIMAG are used to select parts of COMPLEX*16 variables because VAX FORTRAN does not include generic REAL and IMAG functions. All other intrinsic functions used by ELLIPSE are called by their generic names.

ELLIPSE uses the VAX functions DATE, TIME, and SECNDS to get the system date and time, and to measure the time required to arrive at an MAI solution.

Appendix D. ELLIPSE diagnostics

Message	Meaning
DATIN: Too much data. Use 'X' to clear.	An attempt was made to overfill the data buffers. The buffers hold up to NDMAX = 20 measurements.
aaaIN: Invalid input. Type H for help.	The choice entered was not on the menu.
DCHECK: No data entered. Use '2', '4', or 'D'.	
DCHECK: SINGLE mode requires k(1) = 0. Use '1'.	
DCHECK: aaaaaa mode requires at least ii data sets. Use '2', '4', or 'D'.	Too few measurements were entered to allow solution in the requested mode. No check is made to ensure that the buffers contain enough unique incident angles to give a solution.
INFO = ii: Improper input parameters.	Information output from DNLS1.
INFO = ii: FTOL criterion met.	
INFO = ii: XTOL criterion met.	
INFO = ii: FTOL and XTOL criteria both met.	
INFO = ii: GTOL criterion met.	
INFO = ii: Too many function calls.	
INFO = ii: FTOL too small; can't reduce SSQ.	
INFO = ii: XTOL too small; can't improve solution.	
INFO = ii: GTOL too small; F orthogonal to FJAC.	
ADDLAYER: Too many layers.	The stack can contain at most NFMAX = 20 layers.
ADDLAYER: Invalid layer index.	Layer cannot be inserted below the substrate.
AZCON: Bad data at I = ii.	The data buffers indicate that the data from measurement ii is neither 2-zone, 4-zone, nor ψ, Δ pairs.
CHGLAYER: Invalid layer index.	Attempt to modify a layer not currently in the stack.
GETAZ: Invalid data. Reenter.	Angle entered was greater than 360°.
GETAZ: Set compensator to +/- 45 and reenter.	Only ±45° compensator settings are allowed.
GETAZ: Set compensator to xxx.	Second pair of zones must be measured at the opposite compensator setting.
GET_IS: Too many parameters for model.	The current model has fewer parameters than the requested solution (e.g., requesting a 3-parameter solution for a film-free substrate).

SUBLAYER: Invalid layer index.	Attempt to remove the substrate.
TWOZONE: Invalid compensator setting.	Attempt to correct 2-zone data taken with the compensator not set at ±45°.
CORRLS1: Condition = 1/xxxxxxxxx. Hessian is singular.	No correlation matrix could be obtained because the Hessian matrix was too ill-conditioned.

Appendix E. IBM PC version of ELLIPSE

A slightly modified version of ELLIPSE is available for the IBM PC. This version was compiled with Lahey Computer Systems' F77L compiler (version 3.00) and runs on machines equipped with MS DOS 2.0 and higher. It requires 240 kbytes of RAM. The PC version of ELLIPSE has all of the capabilities of the VAX version and differs from VAX ELLIPSE in the following particulars:

1. The source code for the PC contains fewer extensions to FORTRAN. There are no DO WHILE structures and all DO loops terminate in CONTINUE statements rather than END DO's. No variable FORMAT expressions are used. COMPLEX*16 data types are used just as in VAX ELLIPSE. If you intend to compile and link your own ELLIPSE.EXE, your compiler must support the COMPLEX*16 data type; otherwise you must edit all of the source code to the eliminate double-precision complex calculations.

2. The Lahey DATE and TIME functions are used in place of the VAX functions, and the VAX SECNDS function is replaced by Lahey's TIMER function.

3. The PC version is capable of reading ψ, Δ data directly from a Rudolph AutoEl II ellipsometer through its RS-232 port. The ellipsometer must be configured for data transmission to passive terminals (see Sec. 8 of the AutoEl II manual). The cable conecting the PC to the AutoEl must not cross-connect pins 2 and 3. Communication with the AutoEl is accomplished through MEF Software's NOLIMIT utilities version 4.0. The port number, baud rate, parity (none), number of stop bits (one), and word length (seven bits) are read from the file AEII.DAT at the start of program execution, along with a flag to allow a screen dump of the character string received from the AutoEl. Three subroutines are added to ELLIPSE to effect communications: INITAE reads the AEII.DAT file and calls a NOLIMIT routine to initialize the comm port, GETAE calls a NOLIMIT routine to read the AutoEl II data block, and ELDEC decodes the ASCII data block and returns REAL*8 ψ and Δ values. You need not own a copy of NOLIMIT if you do not intend to link your own ELLIPSE.EXE. You can link without the NOLIMIT library if you do not need to read data from an AutoEl II; ignore the unresolved globals OPNCOM, INOCOM, and BEEPIT.

To use the program, copy ELLIPSE.EXE, ELLIPSE.DEF, and AEII.DAT to your ellipsometry subdirectory. From DOS, type 'ELLIPSE' followed by a carriage return to run the program.

To read data from the AutoEl, enter 'A' in response to the data menu prompt 'DATIN>'. ELLIPSE issues a prompt for incident angle and then waits for characters to arrive from the ellipsometer. Striking any key will bring back the prompt. ψ and Δ data from an AutoEl are processed in ELLIPSE exactly like keyboard-input ψ and Δ. If the AEII.DAT file is not present, an error message appears immediately on program execution, and of course the 'A' data-input command waits indefinitely for input.

Appendix F. ELLIPSE source code

```
      PROGRAM ELLIPSE                                  !VERSION 2, 10/25/88
C
C     INTEGRATED ELLIPSOMETRY SYSTEM
C
C     INTEGRATION OF CMELPR AND MAILM PROGRAMS TO OBTAIN SINGLE- OR
C     MULTIPLE-INCIDENT ANGLE DATA AND PROCESS ACCORDINGLY
C
C     SINGLE-INCIDENT ANGLE DATA:
C       ANALYTICAL SOLUTION FOR BARE SUBSTRATES. (SUBR. SUBSTRATE)
C
C       FOR FILM-ON-SUBSTRATE, USES CHARLOT-MARUANI ALGORITHM TO
C       DECOUPLE N, D EQUATIONS.  SEE D. CHARLOT AND A. MARUANI,
C       APPL. OPT. 24, 3368 (1985).
C
C       SPECIFICALLY DESIGNED FOR PURELY-REAL FILM INDEX: USES A
C       MODIFIED VERSION OF THE CHARLOT-MARUANI EQUATION WHICH DOES
C       NOT HAVE NF = NO AS A SOLUTION.  (FUNCTION CMNEWT, SUBR. THICKN)
C
C
C     MULTIPLE-INCIDENT ANGLE DATA:
C       BEST FIT OF MAI DATA TO ELLIPSOMETRY EQUATION FOR SUBSTRATE
C       NS, KS, AND FILM NF, DF.  USES LEVENBERG-MARQUARDT
C       ALGORITHM PER SLATEC ROUTINE DNLS1. (SUBR. FCNG)
C
C     USE OF COMMON IS DICTATED BY FORMAT OF SLATEC ROUTINES.
C
C     LINK ELLIPSE, ELLSUBS/LIB, 'SLATEC'
C
C     H.L. TARDY, SNLA 6224
C
C ************************* DISCLAIMERS ****************************
C
C       This computer software has been developed under sponsorship of
C       the Department of Energy.  Any further distribution by any holder
C       of this software package or other data therein outside of DOE
C       offices or other DOE contractors, unless otherwise specifically
C       provided for, is prohibited without the approval of the National
C       Energy Software Center.  Requests from outside the Department for
C       DOE-developed computer software shall be directed to the
C       Director, National Energy Software Center, Argonne National
C       Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.
C
C       This material was prepared as an account of work sponsored by an
C       agency of the United States Government.  Neither the United
C       States nor the United States Department of Energy, nor any of
C       their employees, makes any warranty, express or implied, or
C       assumes any legal liability or responsibility for the accuracy,
C       completeness, or usefulness of any information, apparatus,
C       product, or process disclosed, or represents that its use would
C       not infringe privately owned rights.
C
C ----------------------------------------------------------------
```

```
C
      IMPLICIT REAL*8 (A-H, O-Z)
C
C ----- USED IN ALL MODES
C
      PARAMETER(NDMAX = 20, NFMAX = 20, NPAR = 5)

C ----- DATA STORAGE

      REAL*8 P(4, NDMAX), A(4, NDMAX), C(2, NDMAX)
      REAL*8 PHI(NDMAX), EANG(2, NDMAX), DEA(2, NDMAX)
      REAL*8 EAI(2), DEAI(2)
      COMMON /MAIN_FCNS/ PHI, EANG
      COMMON /WAVE/ WL

C ----- OPTICAL STACK AND MODEL UNKNOWNS

      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      REAL*8 STKT(3*NFMAX+3)
      REAL*8 X(NPAR), DX(NPAR)
      INTEGER IS(NPAR)
      COMMON /MODEL/ NFLM, STACK, IS
C
      REAL*8 N0
      REAL*4 T0
      CHARACTER*1 ANS
      CHARACTER*6 MODE
      CHARACTER*16 FILE, FILO
      CHARACTER*80 HEAD
      LOGICAL DONE
      PARAMETER(PI = 3.141592654D0, N0 = 1.000D0)
      PARAMETER(NUNIT = 16, NSCREEN = 6)

C ----- SINGLE-ANGLE MODE ONLY

      EXTERNAL CMNEWT
      COMPLEX*16 D, DPHS, XSOL
      REAL*8 PARNT(3*NFMAX+9)
      LOGICAL RANDOM(3*NFMAX+4)

C ----- MULTI-ANGLE MODE ONLY

      EXTERNAL FCNG

C ----- USED IN DNLS1

      INTEGER IPVT(NPAR)
      REAL*8 FJAC(2*NDMAX, NPAR), F(2*NDMAX)
      REAL*8 DIAG(NPAR), QTF(NPAR), WA1(NPAR),
     1         WA2(NPAR), WA3(NPAR), WA4(2*NDMAX)

C ----- USED TO INVERT XJTJ
```

```
      REAL*8 CORR(NPAR*(NPAR+1)/2)
C
C-----------------------------------------------------------------
C
C ----- DEFAULT VALUES
C
      CALL GET_DEF(WL, DP, DA, NFLM, STACK, USTACK, NPARM,
     2               IS, T2P, ETAPC, DT2P, DETA, MODE)
      STACK(3*NFLM+3) = NO
      DO I = 1, 3*NFMAX+4
        RANDOM(I) = .FALSE.  ! TREAT ALL MODEL PARAMS AS SYSTEMATIC
      END DO
C
      FILO = '****************'  ! LOG FILE NAME
      FILE = FILO
      HEAD = ' '           ! HEADER
      NDAT = 0             ! NUMBER OF DATA POINTS
C
      DO J = 1, NDMAX
        C(1, J) = 0.    ! CONDITION FOR NO DATA GIVEN
        C(2, J) = 0.    ! CONDITION FOR TWO-ZONE DATA GIVEN
      END DO
C
      DONE = .FALSE.
C
      CALL HEADOUT(NSCREEN)
      CALL DATHELP(MODE)
C
C ----- LOOP TO HERE FOR REPEATED MEASUREMENTS
C
      DO WHILE (.NOT. DONE)
C
      CALL DATIN(FILE, HEAD, WL, PHI, P, A, C, DP, DA,
     1             EANG, DEA, NDAT, NFLM, STACK, USTACK, NPARM, IS,
     2             T2P, ETAPC, DT2P, DETA, MODE)
C
C
      IF (MODE .EQ. 'QUIT') THEN
        DONE = .TRUE.
      ELSE
C
        WRITE (NSCREEN, 5) HEAD
        IF (FILE .NE. FILO) THEN
          WRITE (NUNIT, 5) HEAD
        END IF
 5      FORMAT(/, X, A80)
C
C ----- SAVE STACK
C
        DPHI = USTACK(3*NFLM+3)
        DO I = 1, 3*NFLM+3
          STKT(I) = STACK(I)
        END DO
```

```
        IF (MODE .NE. 'BARE') THEN
          IF (MODE .NE. 'MULTO') THEN
C
C ----- DUMP STACK MODEL
C
            CALL STDUMP(NFLM, STACK, USTACK, NPARM, IS, NSCREEN)
            IF (FILE .NE. FILO) THEN
              CALL STDUMP(NFLM, STACK, USTACK, NPARM, IS, NUNIT)
            END IF
          END IF
        END IF
C
C ----- DUMP RAW DATA
C
        CALL RAWOUT(WL, PHI, DPHI, P, A, C, DP, DA,
     1              NDAT, MODE, NSCREEN)
        IF (FILE .NE. FILO) THEN
          CALL RAWOUT(WL, PHI, DPHI, P, A, C, DP, DA,
     1                NDAT, MODE, NUNIT)
        END IF
C
C ----- ZONE-AVERAGE AZIMUTHS
C
        CALL AZCON(P, A, C, DP, DA, EANG, DEA,
     1             T2P, ETAPC, DT2P, DETA, NDAT)
C
      END IF
C
      IF (MODE .EQ. 'BARE') THEN
C
C ----- BARE-SUBSTRATE MODE
C
        DO I = 1, 3
          STACK(I) = STACK(3*NFLM+I)
          STKT(I) = STACK(I)
          USTACK(I) = USTACK(3*NFLM+I)
        END DO
        NFLM = 0
C
        PHI1 = PHI(NDAT)
        DO K = 1, 2
          EAI(K) = EANG(K, NDAT)
          DEAI(K) = DEA(K, NDAT)
        END DO
C
        CALL SUBSTRATE(PHI1, EAI, X)
        DO I = 1, 2
          STKT(I) = X(I)
        END DO
C
        CALL UNIERR(0, WL, PHI1, STKT, USTACK, DEAI, IS,
     1              RANDOM, DX)
C
```

```
         CALL SUBOUT(PHI1, EAI, DEAI, T2P, ETAPC,
     2                DT2P, DETA, X, DX, NSCREEN)
         IF (FILE .NE. FILO) THEN
           CALL SUBOUT(PHI1, EAI, DEAI, T2P, ETAPC,
     2                 DT2P, DETA, X, DX, NUNIT)
         END IF
C
       ELSE IF (MODE .EQ. 'SINGLE') THEN  ! SINGLE INCIDENT ANGLE MODE
C
C ----- SET UP FOR SOLUTION;  SINGLE-ANGLE DATA
C
C
C        RANDOM(3*NFLM+3) = .TRUE.    ! UNC. IN PHI
         X(1) = STACK(1)
         PHI1 = PHI(NDAT)
         DO K = 1, 2
           EAI(K) = EANG(K, NDAT)
           DEAI(K) = DEA(K, NDAT)
         END DO
C
         NSIG = 6
         ITMAX = 100
C
         DO I = 1, 3*NFLM+3
           PARNT(I) = STACK(I)
         END DO
         PARNT(3*NFLM+4) = WL
         PARNT(3*NFLM+5) = PHI1
         PARNT(3*NFLM+6) = EANG(1, NDAT)
         PARNT(3*NFLM+7) = EANG(2, NDAT)
         NNEWT = 3*NFLM+9
C
         CALL NEWT(CMNEWT, X(1), F, NNEWT, PARNT, NSIG, ITMAX,
     1              IER)
         XSOL = CMPLX(PARNT(NNEWT-1), PARNT(NNEWT))
C
         STKT(1) = X(1)
C
         CALL THICKN(WL, PHI1, X(1), XSOL, D, DPHS)
         X(2) = DREAL(D)
         STKT(3) = X(2)
C
         CALL UNIERR(NFLM, WL, PHI1, STKT, USTACK, DEAI, IS,
     1               RANDOM, DX)
C
C ----- UNCERTAINTY IN PHASE THICKNESS
C
         SP = NO*SIND(PHI1)/STKT(1)
         CP = SQRT(1.DO-SF**2)
         DDPH = DX(1)**2 +
     1          (DPHI*NO*SP*COSD(PHI1)/CP/CP*PI/180.DO)**2
         DDPH = SQRT(DDPH) * DPHS/STKT(1)
C
```

```
        CALL ELOUT(PHI1, EAI, DEAI,
     1              T2P, ETAPC, DT2P, DETA, STKT(1), D, DPHS,
     2              DX, DDPH, F, IER, NSCREEN)
        IF (FILE .NE. FILO) THEN
          CALL ELOUT(PHI1, EAI, DEAI,
     1               T2P, ETAPC, DT2P, DETA, STKT(1), D, DPHS,
     2               DX, DDPH, F, IER, NUNIT)
        END IF
        PRINT *
C
      ELSE IF (MODE(1:1) .EQ. 'M') THEN
C
C ----- MULTIPLE INCIDENT ANGLE MODES;
C
C     SET UP PARAMETERS FOR DNLS1 ROUTINE
C
        IOPT = 2                  !  WE SUPPLY THE JACOBIAN
        FTOL = 1.D-8              !  RECOMMENDED VALUE
        XTOL = FTOL               !  RECOMMENDED VALUE
        GTOL = 0.D0               !  RECOMMENDED VALUE
        IMODE = 1                 !  DNLS1 SCALES VARIABLES INTERNALLY
        FACTOR = 100.D0           !  RECOMMENDED VALUE
        NPRINT = 0                !  PRINT NO ITERATES OF SSQ
        KONTRL = 0                !  SUPPRESS SLATEC ERROR MESSAGES
        LEVEL = 1                 !  PRINT ONLY FATAL IMSL ERRORS
C
        CALL XSETF(KONTRL)
C
        T0 = SECNDS(0.)
C
        DO I = 1, NPARM
          X(I) = STACK(IS(I))
        END DO
C
C ----- SOLVE LEAST-SQUARES MINIMUM
C
          MAXFN = 100*(NPARM + 1)    !  RECOMMENDED VALUE
          CALL DNLS1(FCNG, IOPT, 2*NDAT, NPARM, X, F, FJAC, 2*NDMAX,
     1                    FTOL, XTOL, GTOL, MAXFN, EPSFCN, DIAG, IMODE,
     2               FACTOR, NPRINT, INFO, NFEV, NJEV, IPVT, QTF,
     3               WA1, WA2, WA3, WA4)
C
C ----- DEFINE SSQ TO BE STANDARD DEV. PER UNIT WEIGHT
C
        SSQ = 0.
        DO I = 1, 2*NDAT
          SSQ = SSQ + F(I)**2
        END DO
        SSQ = SQRT(SSQ)
        IF (NPARM .NE. 2*NDAT) THEN
          SSQ = SSQ/SQRT(2.*NDAT - NPARM)
        END IF
C
```

```
C ----- INVERT HESSIAN TO OBTAIN CORRELATION
C
        CALL CORRLS1(SSQ, FJAC, 2*NDMAX, NPARM, IPVT, DX, CORR)
C
        IF (NPARM .EQ. 2) THEN
          IF (NDAT .EQ. 1) THEN
            PHI1 = PHI(NDAT)
            DO I = 1, 2
              DEAI(I) = DEA(I, 1)
            END DO
            CALL UNIERR(NFLM, WL, PHI1, STKT, USTACK, DEAI, IS,
     1                                                    RANDOM, DX)
          END IF
        END IF
C
        TO = SECNDS(TO)
C
        DO I = 1, NPARM
          STKT(IS(I)) = X(I)
        END DO
C
C ----- OUTPUT RESULTS
C
        CALL MAOUT(NDAT, PHI, EANG, T2P, ETAPC,
     1               X, DX, SSQ, INFO, CORR, F,
     2               MODE, NFLM, NPARM, IS, TO, NSCREEN)
        IF (FILE .NE. FILO) THEN
          CALL MAOUT(NDAT, PHI, EANG, T2P, ETAPC,
     1                X, DX, SSQ, INFO, CORR, F,
     2                MODE, NFLM, NPARM, IS, TO, NUNIT)
        END IF
C
      END IF
C
      IF (MODE .NE. 'QUIT') THEN
        PRINT 899
 899    FORMAT(/,'$Overwrite stack? [y/n] ')
        READ 100, ANS
 100    FORMAT(A)
        IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
          DO I = 1, NPARM
            STACK(IS(I)) = X(I)
            USTACK(IS(I)) = DX(I)
          END DO
        END IF
      END IF
C
      END DO   ! END OF LOOP
C
 9000 END
C
C
```

```
      SUBROUTINE DATIN(FILE, HEAD, WL, PHI, P, A, C, DP, DA,
     1            EANG, DEA, NDAT, NL, STACK, USTACK, NP, IS,
     2            T2P, ETAPC, DT2P, DETA, MODE)              !10/10/88
C
C     H.L. TARDY, 1824
C     MODIFIED FROM ELIN ROUTINE IN MAILM.FOR, 7/16/87
C     ADDED MENU DRIVER, 7/16/87
C     CHANGED TO MAIN MENU MODULE AND SEPARATED INTO THREE PARTS
C     (SUBR.'S ELIN, DATIN, MODIN):  8/31/87
C
C     CHANGED TO STACK ARRAY STORAGE:  9/26/88
C
C     INPUT MODULE FOR CMELP
C
C     FILE                                                          :
      FILE NAME
C     HEAD                                                          :
      HEADER STRING
C     WL                                                            :
      WAVELENGTH
C     PHI                                                           :
      INCIDENT ANGLE (DEGREES)
C     P                                                             :
      POLARIZER AZIMUTHS OR PSI, DEL
C     A                                                             :
      ANALYZER AZIMUTHS
C     C                                                             :
      COMPENSATOR AZIMUTHS
C     DP          :   UNC. IN P
C     DA          :   UNC IN A
C     EANG(I)     :   ELLIPSOMETRIC ANGLE (1, 2 -> PSI, DELTA)
C     DEA         :   UNC. IN EANG
C     NDAT                                                          :
      NUMBER OF DATA POINTS
C     NL          :   # OF LAYERS
C     STACK(I)    :      STACK MODEL (SEE DOCUMENTATION IN SUBR. GET_DEF)
C     USTACK(I)   :                                  STACK UNCERTAINTIES
C     NP          :   # OF UNKNOWN PARAMETERS
C     IS(I)       :   INDICES OF UNKNOWNS
C     T2P         :   POLARIZER ELLIPTICITY
C     ETAPC       :   COMBINED ERRORS, P, C, POL. IMPERFECTION
C     DT2P        :   UNC. IN T2P
C     DETA        :   UNC. IN ETAPC
C     MODE                                                          :
      BARE, SINGLE OR MULTI INCIDENT ANGLE
C
C
C     DATA IS STORED IN P, A, C ARRAYS IN SINGLE-ANGLE MODES FOR USE
C     IN MULTI-ANGLE MODES.  THIS ALLOWS ANALYSIS OF MULTIPLE-ANGLE
C     DATA AS IT IS INPUT.
C
C
```

```
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NDMAX = 20, NFMAX = 20, NPMAX = 5)
C
      REAL*8 NO, STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      REAL*8 P(4, NDMAX), A(4, NDMAX), C(2, NDMAX)
      REAL*8 PJ(4), AJ(4), CJ(2)
      REAL*8 PHI(NDMAX), EANG(2, NDMAX), DEA(2, NDMAX)
      REAL*8 EAI(2), DEAI(2)
      INTEGER IS(NPMAX)
      CHARACTER*1 NOCHOICE, CHOICE, ANS
      CHARACTER*6 MODE
      CHARACTER*7 PROMPT
      CHARACTER*8 CTIME
      CHARACTER*16 FILE, FILO
      CHARACTER*80 HEAD
      LOGICAL DONE
      PARAMETER(NO = 1.000D0)
      PARAMETER(NUNIT = 16, NSCRN = 6, INU = 5)
C
      FILO = '****************'
      PROMPT = 'DATIN> '
      DONE = .FALSE.
      NOCHOICE = ' '
      CHOICE = NOCHOICE
C
C ----- MENU DRIVER
C
      DO WHILE (.NOT. DONE)
        IF (CHOICE .EQ. NOCHOICE) THEN
          PRINT 10, PROMPT
 10       FORMAT(/, '$', A7)
          READ 15, CHOICE
 15       FORMAT(A)
C
        ELSE IF (CHOICE .EQ. '2') THEN
C
C ----- ENTER TWO-ZONE DATA:  TWO- OR FOUR-ZONE INPUT REQUIRED
C
          IF (NDAT .LT. NDMAX) THEN
            NDAT = NDAT + 1
 800        PRINT 18
 18         FORMAT(/, '$Enter incident angle:  ')
            READ (INU, *, ERR = 800) PHI(NDAT)
C
            CALL GETAZ(PJ, AJ, CJ, 2)  !  OBTAIN 2-ZONE AZIMUTHS
C
            CALL BUFLOAD(NDAT, P, A, C, PJ, AJ, CJ)
C
          ELSE
            PRINT 82
          END IF
 82       FORMAT(/, ' DATIN:  Too much data.  Use ''X''
     1 to clear.')
```

```
C
C ----- NOTE: 2-ZONE AZIMUTHS ARE CONVERTED TO PSI, DEL IN SUBR. AZCON
C     ON RETURN TO MAIN
C
          CHOICE = NOCHOICE
C
        ELSE IF (CHOICE .EQ. '4') THEN
C
C ----- ENTER FOUR-ZONE DATA:  TWO- OR FOUR-ZONE INPUT REQUIRED
C
          IF (NDAT .LT. NDMAX) THEN
            NDAT = NDAT + 1
 810        PRINT 18
            READ (INU, *, ERR = 810) PHI(NDAT)
C
            CALL GETAZ(PJ, AJ, CJ, 4)  ! OBTAIN 4-ZONE AZIMUTHS
C
            CALL BUFLOAD(NDAT, P, A, C, PJ, AJ, CJ)
          ELSE
            PRINT 82
          END IF
C
C ----- 4-ZONE AVERAGE
C
          CALL FOURZONE(PJ, AJ, CJ, DP, DA, EAI,
     1                  DEAI, T2P, ETAPC, DT2P,
     2                  DETA)

C
          DO K = 1, 2
            EANG(K, NDAT) = EAI(K)
            DEA(K, NDAT) = DEAI(K)
          END DO
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'C') .OR. (CHOICE .EQ. 'c')) THEN
C
C ----- CALIBRATE
C
          PRINT 95
 95       FORMAT(/, ' Place Analyzer in straight-through position')
C
          CALL GETAZ(PJ, AJ, CJ, 4)
C
          CALL FOURZONE(PJ, AJ, CJ, DP, DA, EAI,
     1                  DEAI, T2P, ETAPC, DT2P, DETA)
C
          CALL CALOUT(PJ, AJ, CJ, DP, DA, EAI, DEAI,
     1                T2P, ETAPC, DT2P, DETA, NSCRN)
          IF (FILE .NE. FILO) THEN
            CALL CALOUT(PJ, AJ, CJ, DP, DA, EAI, DEAI,
     1                  T2P, ETAPC, DT2P, DETA, NUNIT)
          END IF
```

```
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'D') .OR. (CHOICE .EQ. 'd')) THEN
C
C ----- DIRECT ENTRY OF PSI, DELTA
C
          IF (NDAT .LT. NDMAX) THEN
            NDAT = NDAT + 1
 880        PRINT 18
            READ (INU, *, ERR = 880) PHI(NDAT)
 890        PRINT 100
            READ (INU, *, ERR = 890) P(1, NDAT), P(2, NDAT)
            C(2, NDAT) = 999.
 100        FORMAT(/, ' Enter Psi, Delta')
C
C ----- CONVERT DIRECT PSI, DEL DATA
C
            DO K = 1, 2
              EANG(K, NDAT) = P(K, NDAT)
            END DO
C
C ----- ASSUME MINIMUM UNCERTAINTIES FOR DIRECT DATA
C
            DEA(1, NDAT) = DA/2.
            DEA(2, NDAT) = DP
          ELSE
            PRINT 82
          END IF
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
C
C ----- HELP!
C
          CALL DATHELP(MODE)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
C
C ----- GET MODE
C
          CALL MODIN(MODE, NL, NP, IS, CHOICE)
C
        ELSE IF ((CHOICE .EQ. 'N') .OR. (CHOICE .EQ. 'n')) THEN
C
C ----- NAME DATA:  OPTIONAL
C
          PRINT *, 'Enter sample name'
          READ 38, HEAD
          CHOICE = NOCHOICE
 38       FORMAT(A80)
```

```
C
        ELSE IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
C
C ----- GOTO PARAMETER MENU
C
          CALL PARIN(FILE, HEAD, WL, DP, DA,
     1                    NL, STACK, USTACK, NP, IS,
     2                    T2P, ETAPC, DT2P, DETA, MODE, CHOICE)
C
        ELSE IF ((CHOICE .EQ. 'Q') .OR. (CHOICE .EQ. 'q')) THEN
C
C ----- QUIT
C
          IF (FILE .NE. FILO) THEN
            CALL LOGCL(NUNIT)
          END IF
          MODE = 'QUIT'
          DONE = .TRUE.
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'R') .OR. (CHOICE .EQ. 'r')) THEN
C
C ----- RETURN TO MAIN (IF SUFFICIENT INPUT)
C
          CALL DCHECK(WL, PHI, NDAT, STACK, NL, NP,
     2                   DONE, CHOICE, MODE)
C
        ELSE IF ((CHOICE .EQ. 'V') .OR. (CHOICE .EQ. 'v')) THEN
C
C ----- DUMP RAW DATA
C
          CALL RAWOUT(WL, PHI, DPHI, P, A, C, DP, DA,
     1                NDAT, 'DUMP', NSCRN)
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'X') .OR. (CHOICE .EQ. 'x')) THEN
C
C ----- CLEAR DATA
C
          DO J = 1, 2
            DO L = 1, NDAT
              C(J, L) = 0.
            END DO
          END DO
          NDAT = 0
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'Z') .OR. (CHOICE .EQ. 'z')) THEN
C
C ----- CLEAR CORRECTIONS
C
```

```
            T2P = 0.
            ETAPC = 0.
            DT2P = 0.
            DETA = 0.
C
            CHOICE = NOCHOICE
C
          ELSE         ! NONE OF THE ABOVE
C
            PRINT *
            PRINT *, 'DATIN:  Invalid input.  Type ''H'' for help.'
            CHOICE = NOCHOICE
C
          END IF
C
        END DO
C
        RETURN
        END
C
C
```

```
      SUBROUTINE PARIN(FILE, HEAD, WL, DP, DA,
     1                 NL, STACK, USTACK, NP, IS,
     2                 T2P, ETAPC, DT2P, DETA, MODE, CHOICE)
!10/24/88
C
C     H.L. TARDY, 1824
C
C     CREATED OUT OF ELIN ROUTINE, INCLUDES ONLY DATA INPUT
C     FUNCTIONS:  8/31/87
C     MODIFIED TO PERFORM INITIAL PROCESSING OF DATA:  9/1/87
C     ADDED ERROR CHECKING ON DATA INPUTS:  9/2/87
C
C     DATA IS STORED IN P, A, C ARRAYS IN SINGLE-ANGLE MODES FOR USE
C     IN MULTI-ANGLE MODES.  THIS ALLOWS ANALYSIS OF MULTIPLE-ANGLE
C     DATA AS IT IS INPUT.
C
C     RETURNS 4-ZONE PSI, DEL, ETC. AND UNCORRECTED 2-ZONE PSI, DEL.
C     THIS ALLOWS UPDATE OF CORRECTIONS ON PREVIOUS 2-ZONE
C     MEASUREMENTS AS NEW 4-ZONE MEASUREMENTS ARE MADE.
C     (SEE SUBR. AZCON)
C
C     CHAR. VBLE. CHOICE IS RETURNED TO ALLOW 'RUN', 'MODE' AND
C     'QUIT' COMMANDS TO EXECUTE WITHOUT MULTIPLE ENTRIES.  THESE
C     COMMANDS ARE EXECUTED IN SUBR. DATIN VIA RETURN STATEMENTS.
C     THUS ISSUING 'MODE' COMMAND IN DATIN LEAVES USER IN MODIN.
C
C
C     CHANGED TO STACK ARRAY STORAGE:  9/26/88
C     REMOVED LOG CLOSE, LOG OPEN, ADD LAYER, REMOVE LAYER, AND
C        CHANGE LAYER FUNCTIONS TO SUBR.'S:  10/10/88
C     ADDED 1-PARAM. CHANGE (SUBR. PARCHG):  10/24/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NDMAX = 20, NFMAX = 20, NPMAX = 5)
C
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4), NO
      INTEGER IS(NPMAX)
      CHARACTER*1 NOCHOICE, CHOICE, ANS
      CHARACTER*6 MODE
      CHARACTER*7 PROMPT
      CHARACTER*16 FILE, FILO
      CHARACTER*9 CDATE
      CHARACTER*80 HEAD
      LOGICAL DONE
      PARAMETER(NO = 1.000D0)
      PARAMETER(NUNIT = 16, NSCRN = 6, INU = 5)
C
      FILO = '****************'
      PROMPT = 'PARIN> '
      DONE = .FALSE.
      NOCHOICE = ' '
      CHOICE = NOCHOICE
C
```

```
C ----- MENU DRIVER
C
      DO WHILE (.NOT. DONE)
        IF (CHOICE .EQ. NOCHOICE) THEN
          PRINT 10, PROMPT
 10       FORMAT(/, '$', A7)
          READ 15, CHOICE
 15       FORMAT(A)
C
        ELSE IF (CHOICE .EQ. '1') THEN
C
C ----- CHANGE ONE PARAMETER
C
          CALL PARCHG(NL, STACK, USTACK)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'C') .OR. (CHOICE .EQ. 'c')) THEN
C
C ----- CLOSE LOG
C
          IF (FILE .NE. FILO) THEN
            CALL LOGCL(NUNIT)
            FILE = FILO
          END IF
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'D') .OR. (CHOICE .EQ. 'd')) THEN
C
C ----- RETURN TO DATA MENU
C
          CHOICE = NOCHOICE
          RETURN
C
        ELSE IF ((CHOICE .EQ. 'E') .OR. (CHOICE .EQ. 'e')) THEN
C
C ----- ENTER FULL STACK
C
          CALL STACKIN(NL, STACK, USTACK)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'G') .OR. (CHOICE .EQ. 'g')) THEN
C
C ----- GET DEFAULTS
C
          CALL GET_DEF(WL, DP, DA, NL, STACK, USTACK, NP,
     1                                  IS, T2P, ETAPC, DT2P, DETA, MODE)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
C
C ----- HELP!
C
```

```
          CALL PARHELP(WL, DA, DP, NL, STACK, USTACK, MODE)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'K') .OR. (CHOICE .EQ. 'k')) THEN
C
C ----- KEEP DEFAULT PARAM'S.
C
          CALL PUT_DEF(WL, DP, DA, NL, STACK, USTACK,
     2                 NP, IS, T2P, ETAPC, DT2P, DETA, MODE)
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'L') .OR. (CHOICE .EQ. 'l')) THEN
C
C ----- CHANGE ONE LAYER
C
          CALL CHGLAYER(NL, STACK, USTACK)
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'O') .OR. (CHOICE .EQ. 'o')) THEN
C
C ----- OPEN LOG FILE:  OPTIONAL
C
          IF (FILE .NE. FILO) THEN
            CALL LOGCL(NUNIT)
          END IF
          CALL LOGOP(FILE, HEAD, NUNIT)
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'M') .OR. (CHOICE .EQ. 'm')) THEN
C
C ----- MODE
C
          RETURN
C
        ELSE IF ((CHOICE .EQ. 'Q') .OR. (CHOICE .EQ. 'q')) THEN
C
C ----- QUIT
C
          GOTO 999           ! RETURN
C
        ELSE IF ((CHOICE .EQ. 'R') .OR. (CHOICE .EQ. 'r')) THEN
C
C ----- RUN
C
          RETURN
C
        ELSE IF ((CHOICE .EQ. 'S') .OR. (CHOICE .EQ. 's')) THEN
C
C ----- SHOW STACK
C
          CALL STDUMP(NL, STACK, USTACK, NP, IS, NSCRN)
```

```
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'U') .OR. (CHOICE .EQ. 'u')) THEN
C
C ----- UNCERTAINTIES
C
          PRINT *, 'Enter uncertainties in Phi, P, and A
     1 in degrees'
          READ *, USTACK(3*NL+3), DP, DA
C
          CHOICE = NOCHOICE
C
        ELSE IF ((CHOICE .EQ. 'W') .OR. (CHOICE .EQ. 'w')) THEN
C
C ----- SET WAVELENGTH:  OPTIONAL
C
 930      PRINT 50
          READ (INU, *, ERR = 930, IOSTAT = IOS) WL
 50       FORMAT(/, '$Enter wavelength:  ')
C
          CHOICE = NOCHOICE
C
        ELSE IF (CHOICE .EQ. '+') THEN
C
C ----- ADD ONE LAYER
C
          CALL ADDLAYER(NL, STACK, USTACK)
C
          CHOICE = NOCHOICE
C
        ELSE IF (CHOICE .EQ. '-') THEN
C
C ----- SUBTRACT ONE LAYER
C
          CALL SUBLAYER(NL, STACK, USTACK)
C
          CHOICE = NOCHOICE
C
        ELSE
          PRINT *
          PRINT *, 'PARIN:  Invalid input.  Type ''H'' for help.'
          CHOICE = NOCHOICE
        END IF
C
      END DO
C
 999  RETURN
      END
C
C
```

```
      SUBROUTINE MODIN(MODE, NFLM, NPARM, IS, CHOICE)       !10/24/88
C
C     OBTAINS MODE DESIGNATION FROM KEYBOARD
C
C     CHAR. VBLE. CHOICE IS TRANSFERRED TO ALLOW 'RUN', 'MODE' AND
C     'QUIT' COMMANDS TO EXECUTE WITHOUT MULTIPLE ENTRIES.  THESE
C     COMMANDS ARE EXECUTED IN SUBR. DATIN VIA RETURN STATEMENTS.
C     THUS ISSUING 'DATA' COMMAND IN MODIN LEAVES USER IN DATIN.
C
C     CHANGED TO STACK ARRAY STORAGE:  9/26/88
C     REMOVED IS(I) INPUT TO SUBR.:  10/10/88
C
      PARAMETER (NFMAX = 20, NPMAX = 5)
      INTEGER IS(NPMAX)
      CHARACTER*6 MODE
      CHARACTER*1 CHOICE, NOCHOICE, ANS
C
      NOCHOICE = ' '
C
 1    PRINT 10
      READ 20, CHOICE
C
 10   FORMAT(/, '$MODIN> ')
 20   FORMAT(A)
C
      IF ((CHOICE .EQ. 'A') .OR. (CHOICE .EQ. 'a')) THEN
C
C ----- ARBITRARY N-PARAMETER MODE
C
        MODE = 'MAR2'
        PRINT 22
 22     FORMAT('$Enter number of unknown parameters: ')
        READ *, NPARM
        CALL GET_IS(NFLM, NPARM, IS)
C
      ELSE IF ((CHOICE .EQ. 'B') .OR. (CHOICE .EQ. 'b')) THEN
C
C ----- BARE-SUBSTRATE MODE
C
        PRINT 30
 30     FORMAT('$This mode overwrites the stack.  Do you wish to
     1 continue? [y/n] ')
        READ 20, ANS
        IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
          MODE = 'BARE'
          NPARM = 2
          DO I = 1, 2
            IS(I) = I
          END DO
        END IF
C
      ELSE IF ((CHOICE .EQ. 'D') .OR. (CHOICE .EQ. 'd')) THEN
C
C ----- EXIT TO DATA MENU
```

```
C
        CHOICE = NOCHOICE
        RETURN
C
      ELSE IF ((CHOICE .EQ. 'H') .OR. (CHOICE .EQ. 'h')) THEN
        CALL MODEHELP(MODE)
        GOTO 1
      ELSE IF (CHOICE .EQ. '0') THEN
        PRINT 30
        READ 20, ANS
        IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
          MODE = 'MULT0'
          NFLM = 0
          NPARM = 2
          DO I = 1, NPARM
            IS(I) = I
          END DO
        END IF
      ELSE IF (CHOICE .EQ. '2') THEN
        MODE = 'MULT2'
        NPARM = 2
        IS(1) = 1
        IS(2) = 3
      ELSE IF (CHOICE .EQ. '3') THEN
        MODE = 'MULT3'
        NPARM = 3
        DO I = 1, NPARM
          IS(I) = I
        END DO
      ELSE IF (CHOICE .EQ. '4') THEN
        MODE = 'MULT4'
        NPARM = 4
        IS(1) = 1
        IS(2) = 3
        IS(3) = 4
        IS(4) = 5
      ELSE IF (CHOICE .EQ. '5') THEN
        MODE = 'MULT5'
        NPARM = 5
        DO I = 1, NPARM
          IS(I) = I
        END DO
      ELSE IF ((CHOICE .EQ. 'P') .OR. (CHOICE .EQ. 'p')) THEN
C
C ----- TO PARAMETER MENU
C
        RETURN
C
      ELSE IF ((CHOICE .EQ. 'Q') .OR. (CHOICE .EQ. 'q')) THEN
C
C ----- QUIT
C
        GOTO 999            ! RETURN
```

```
C
      ELSE IF ((CHOICE .EQ. 'R') .OR. (CHOICE .EQ. 'r')) THEN
C
C ----- RUN
C
        RETURN
C
      ELSE IF ((CHOICE .EQ. 'S') .OR. (CHOICE .EQ. 's')) THEN
        MODE = 'SINGLE'
        NPARM = 2
        IS(1) = 1
        IS(2) = 3
C
      ELSE
        PRINT *, 'MODIN:  Invalid input.  Type ''H'' for help.'
        GO TO 1
      END IF
C
      GOTO 1  ! LOOP
C
 999  RETURN
      END
C
C
```

```
      SUBROUTINE MODEHELP(mode)                    !9/27/88
C
C     HELP FOR CMELPR MENU
C
C     H.L. TARDY, 1824
C     7/16/87
C
      IMPLICIT REAL*8 (A-H, O-Z)
      CHARACTER*6 MODE
C
      print 10, mode
 10   format(/, ' Current mode is ', a6)
      print *, 'Enter:'
      print *, '   0 - for Multiple-angle ns, ks mode'
      print *, '   2 - for Multiple-angle nf, df mode'
      print *, '   3 - for Multiple-angle nf, kf, df mode'
      print *, '   4 - for Multiple-angle nf, df, ns, ks mode'
      print *, '   5 - for Multiple-angle nf, kf, df, ns, ks mode'
      print *, '   A - for Arbitrary n-parameter solution'
      print *, '   B - for Bare-substrate model'
      print *, '   D - to go to Data menu'
c     print *, '   F - for Fixed-film index mode'
      print *, '   H - for Help'
      print *, '   P - to go to Parameter menu'
      print *, '   Q - to Quit'
      print *, '   R - to Run data'
      print *, '   S - for Single-angle nf, df model'
c
      return
      end
c
c
```

```
      SUBROUTINE PARHELP(WL, DA, DP, NL, STACK, USTACK,
     1                   MODE)                                  !9/27/88
C
C     HELP FOR CMELPR MENU
C
C     H.L. TARDY, 1824
C     7/16/87
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20, NPMAX = 5)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      CHARACTER*6 MODE
C
      print *
      print *, 'Enter:'
      print *, '   1 - to change 1 parameter'
      print *, '   C - to Close log file'
      print *, '   D - to go to Data menu'
      print *, '   E - to Enter a complete stack'
      print *, '   G - to Get default parameters'
      print *, '   H - for Help'
      print *, '   K - to Keep default parameters'
      print *, '   L - to change one Layer'
      print *, '   O - to Open a log file'
      print 25, mode
 25   format(4x,  'M - to change Mode (Mode = ', a6, ')')
      print *, '   Q - to Quit'
      print *, '   R - to Run data'
      print *, '   S - to Show stack'
      print 35, USTACK(3*NL+3), DP, DA
 35   format(4x,  'U - to enter Uncertainties (dPhi =', f6.3,
     1         ', dP =', f6.3, ', dA =', f6.3, ')')
      print 40, wl
 40   format(4x,  'W - to change Wavelength
     1 (wl =', f7.1, ' Angstroms)')
      print *, '   + - to add one layer'
      print *, '   - - to subtract one layer'
c
      return
      end
c
c
```

```
      SUBROUTINE DATHELP(MODE)                    !9/27/88
C
C     HELP FOR CMELPR MENU
C
C     H.L. TARDY, 1824
C     7/16/87
C
      IMPLICIT REAL*8 (A-H, O-Z)
      CHARACTER*6 MODE
C
      print *
      print *, 'Enter:'
      print *, '   2 - to enter Two-zone data (*)'
      print *, '   4 - to enter Four-zone data (*)'
      print *, '   C - to enter Calibration data'
      print *, '   D - to enter psi, delta Directly (*)'
      print *, '   H - for Help'
      print 25, mode
 25   format(4x,  'M - to change Mode (Mode = ', a6, ')')
      print *, '   N - to enter a Name string for data'
      print *, '   P - to go to Parameter menu'
      print *, '   Q - to Quit'
      print *, '   R - to Run data'
      print *, '   V - to View current raw data arrays'
      print *, '   X - to clear data'
      print *, '   Z - to clear corrections'
c
      print *
      print *, '* = One required'
      return
      end
c
c
```

```
      SUBROUTINE DCHECK(WL, PHI, NDAT, STACK, NL, NP,
     2                  DONE, CHOICE, MODE)          !9/26/88
C
C     H.L. TARDY, 1824
C     10/16/87
C
C     DATA AND PARAMETER CHECKER FOR ELLIPSE.  MAKES SURE ALL
C     NECESSARY INFO IS AVAILABLE BEFORE ALLOWING PROGRAM TO
C     PROCEED TO SOLUTION PHASE.  NO CHECKING IS DONE FOR VALIDITY
C     OF P, A, C VALUES, AS THAT IS DONE AS THEY ARE COLLECTED.
C
C     CHANGED TO STACK ARRAY STORAGE:  9/26/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER(NDMAX = 20, NFMAX = 20)
      REAL*8 PHI(NDMAX)
      REAL*8 STACK(3*NFMAX+3)
      CHARACTER*6 MODE
      LOGICAL DONE
      CHARACTER*1 CHOICE, NOCHOICE
C
      NOCHOICE = ' '
      DONE = .TRUE.
      CHOICE = NOCHOICE
C
      IF (NDAT .EQ. 0) THEN  ! NO DATA ENTERED
        PRINT *
        PRINT *, 'DCHECK:  No data entered.  Use ''2'', ''4'',
     1 or ''D''.'
        DONE = .FALSE.
      END IF
C
      IF (MODE .EQ. 'SINGLE') THEN
        IF (STACK(2) .NE. 0.) THEN
          PRINT *, 'DCHECK:  SINGLE mode requires k(1) = 0.
     1  Use ''1''.'
C
          DONE = .FALSE.
          CHOICE = 'P'  ! SEND USER TO SUBR. PARIN
        END IF
      END IF
C
      IF (MODE .EQ. 'MULT3') THEN
        NDMIN = 2
        IF (NDAT .LT. NDMIN) THEN
          PRINT 100, MODE, NDMIN
          DONE = .FALSE.
        END IF
      ELSE IF (MODE .EQ. 'MULT4') THEN
        NDMIN = 2
        IF (NDAT .LT. NDMIN) THEN
          PRINT 100, MODE, NDMIN
          DONE = .FALSE.
        END IF
```

```
      ELSE IF (MODE .EQ. 'MULT5') THEN
        NDMIN = 3
        IF (NDAT .LT. NDMIN) THEN
          PRINT 100, MODE, NDMIN
          DONE = .FALSE.
        END IF
      END IF
 100  FORMAT(/, ' DCHECK:  ', A6, ' mode requires at least ', I2,
     1 ' data sets.  Use ''2'', ''4'', or ''D''.')
C
      RETURN
      END
C
C
```

```
      SUBROUTINE HEADOUT(NUNIT)                              !9/28/88
C
C     H.L. TARDY, 1824
C     7/17/87
C
C     HEADER OUTPUT MODULE FOR CMELPR
C
C
      WRITE (NUNIT, 10)
      WRITE (NUNIT, 15)
      WRITE (NUNIT, 20)
      WRITE (NUNIT, 30)
      WRITE (NUNIT, 100)
 10   FORMAT(/, ' ELLIPSE:  Integrated Ellipsometry System, Version 2')
 15   FORMAT(' Input and Analysis of Ellipsometric Null Data, using')
 20   FORMAT(' Single or Multiple Incident Angles, and')
 30   FORMAT(' Two- or Four-Zone Averaging.')
 100  FORMAT(' H.L. Tardy, SNLA 6224', /, ' September, 1988')
C
      RETURN
      END
C
C
```

```
      SUBROUTINE RAWOUT(WL, PHI, DPHI, P, A, C,
     1             DP, DA, NDAT, MODE, NUNIT)          !9/22/88
C
C     H.L. TARDY, 1824
C     7/17/87
C
C     OUTPUT MODULE FOR CMELPR;  DUMPS RAW DATA TO FILE AND DISK
C     DATA OUTPUT IS SEPARATED FROM RESULT OUTPUT SO THAT DATA
C     IS NOT LOST IF SOLUTION BOMBS
C
C     FILE ON NUNIT MUST HAVE BEEN OPENED IN DATIN
C     USING 'L' INSTRUCTION
C
C     MODIFIED FOR USE WITH STDUMP ROUTINE; DOES NOT REPORT
C        STACK PARAMS:  9/22/88
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NDMAX = 20)
      REAL*8 P(4, NDMAX), A(4, NDMAX), C(2, NDMAX)
      REAL*8 PHI(NDMAX)
      INTEGER K(NDMAX)
      CHARACTER*6 MODE
      LOGICAL PAFLAG
C
      PAFLAG = .TRUE.
C
      IF (MODE(1:4) .NE. 'DUMP') THEN
C
C ----- DETERMINE TYPE OF EACH DATA SET
C
        DO L = 1, NDAT
          IF (C(2, L) .EQ. 999.) THEN  ! PSI, DEL ONLY
            K(L) = 0
            PAFLAG = .FALSE.
          ELSE IF (C(2, L) .EQ. 0.) THEN  ! TWO-ZONE DATA
            K(L) = 1
            PAFLAG = .TRUE.    ! AT LEAST ONE 2-ZONE SET
          ELSE
            K(L) = 2
            PAFLAG = .TRUE.    ! AT LEAST ONE 4-ZONE SET
          END IF
        END DO
      ELSE
        PAFLAG = .TRUE.
        DO L = 1, NDAT
          K(L) = 2
        END DO
      END IF
C
      IF (MODE(1:4) .NE. 'DUMP') THEN
        WRITE (NUNIT, 10) WL
 10     FORMAT(/, ' Wavelength = ', F8.2)
      END IF
```

```
C
C ----- DUMP P, A, C SETS (IF ANY)
C
      IF (PAFLAG) THEN
        WRITE (NUNIT, 80)
 80     FORMAT(/, ' Raw Data')
        WRITE (NUNIT, 90)
 90     FORMAT(4X, 'Phi', 7X, 'P', 8X, 'A', 8X,
     1           'P'''', 7X, 'A'''', 6X, 'C')
C
        IF ((MODE .EQ. 'BARE') .OR. (MODE .EQ. 'SINGLE')) THEN
          L0 = NDAT
        ELSE
          L0 = 1
        END IF
C
        DO L = L0, NDAT
          DO I = 1, K(L)
            IDX = 2*(I - 1)
            WRITE (NUNIT, 100, IOSTAT = IOS) PHI(L),
     1        (P((IDX + J), L), A((IDX + J), L), J = 1, 2),
     2        C(I, L)
          END DO
 100      FORMAT(X, F7.2, 4F9.2, F8.2)
        END DO
      END IF
C
C ----- UNCERTAINTIES
C
      WRITE (NUNIT, 124)
      WRITE (NUNIT, 126, IOSTAT = IOS) DPHI, DP, DA
C
 124  FORMAT(/, ' Input Uncertainties')
 126  FORMAT(' dPhi =', F6.3, ', dP =', F6.3, ', dA =',
     1           F6.3)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE ELOUT(PHI, EANG, DEA, T2P, ETAPC, DT2P, DETA,
     1                 NF, D, DPHS, DRES, DDPH,
     2                 F, IER, NUNIT)   !9/28/88
C
C     H.L. TARDY, 1824
C     7/17/87
C
C     OUTPUT MODULE FOR CMELPR;  DUMPS RESULTS TO FILE AND SCREEN
C
C     FILE ON NUNIT MUST HAVE BEEN OPENED IN DATIN
C     USING 'L' INSTRUCTION
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 DRES(2), EANG(2), DEA(2), NF
      REAL*4 TIME
      COMPLEX*16 D, DPHS
      CHARACTER*16 FILE, FILO
C
C
C ----- DUMP PSI, DEL RESULTS.  DATA WHICH APPEARS HERE BUT NOT IN
C     P, A, C DUMP (SUBR. RAWOUT) WAS INPUT DIRECTLY AS PSI, DELTA
C
        WRITE (NUNIT, 110)
 110    FORMAT(/, ' Ellipsometric Angles', /,
     1            4X, 'Phi', 7X, 'Psi', 7X, 'Delta', 7X,
     2            't2P', 7X, 'EtaPC')
C
        WRITE (NUNIT, 120) PHI, (EANG(K), K = 1, 2), T2P, ETAPC
 120    FORMAT(X, F7.2, 4F11.3)
        IF ((DEA(1).NE. 0.) .AND. (DEA(2).NE. 0.)) THEN
          WRITE (NUNIT, 125, IOSTAT = IOS) (DEA(K), K = 1, 2),
     1                                       DT2P, DETA
        END IF
 125    FORMAT(10X, '+/-', F6.3, '  +/-', F6.3, '  +/-', F6.3,
     1          '  +/-', F6.3)
C
        WRITE (NUNIT, 130, IOSTAT = IOS) NF, DRES(1), DREAL(D),
     1       DRES(2), DREAL(DPHS), DDPH
 130    FORMAT(/, ' Film-on-Substrate Results', /,
     1          ' nf =', F8.4, ' +/-', F7.4, /,
     2          ' D =', F8.2, ' +/-', F7.2, /,
     3                        ' Phase thickness =', F8.2, ' +/-', F7.2)
        WRITE (NUNIT, 140, IOSTAT = IOS) DIMAG(D), DIMAG(DPHS),
     1       F, IER
 140    FORMAT(/, ' Diagnostics', /,
     1          ' Im[D] =', F6.2, '  Im[Dphase] =', F6.2,
     2          '  F =', E9.2, '  IER = ', I3)
C
C       WRITE (NUNIT, 150) TIME
C 150   FORMAT(/, ' Elapsed time =', f6.2, ' sec.')
C
C
```

```
      RETURN
      END
C
C
```

```
      SUBROUTINE SUBOUT(PHI, EANG, DEA,
     1                  T2P, ETAPC, DT2P, DETA, NS, DNS, NUNIT)
!9/28/88
C
C     H.L. TARDY, 1824
C     8/7/87
C
C     OUTPUT MODULE FOR ELLIPSE;  DUMPS BARE-SUBST.
C     RESULTS TO FILE AND SCREEN
C
C     FILE ON NUNIT MUST HAVE BEEN OPENED IN DATIN
C     USING 'L' INSTRUCTION
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 NS(2), DNS(*), EANG(2), DEA(2)
C

C ----- DUMP PSI, DEL RESULTS.  DATA WHICH APPEARS HERE BUT NOT IN
C     P, A, C DUMP (SUBR. RAWOUT) WAS INPUT DIRECTLY AS PSI, DELTA
C
        WRITE (NUNIT, 110)
 110    FORMAT(/, ' Ellipsometric Angles', /,
     1            4X, 'Phi', 7X, 'Psi', 7X, 'Delta', 7X,
     2            't2P', 7X, 'EtaPC')
C
        WRITE (NUNIT, 120) PHI, (EANG(K), K = 1, 2), T2P, ETAPC
 120    FORMAT(X, F7.2, 4F11.3)
        IF ((DEA(1).NE. 0.) .AND. (DEA(2).NE. 0.)) THEN
          WRITE (NUNIT, 125) (DEA(K), K = 1, 2), DT2P, DETA
        END IF
 125    FORMAT(10X, '+/-', F6.3, '  +/-', F6.3, '  +/-', F6.3,
     1          '  +/-', F6.3)
C
        WRITE (NUNIT, 130, IOSTAT = IOS), (NS(I), DNS(I),
     1       I = 1, 2)
 130    FORMAT(/, ' Bare Substrate Results', /, ' ns =',
     1          F8.4, ' +/-', F7.4, '  ks =', F8.4, ' +/-',
     2          F7.4)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MAOUT(NDAT, PHI, EANG, T2P, ETAPC,
     1                 NS, DNS, SSQ, INFO, CORR, F,
     2                 MODE, NFLM, NPARM, IS, TIME, UNIT)     !9/28/88
C
C     OUTPUT MODULE FOR MAILM
C
C     H.L. TARDY, 1824
C     7/2/87
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER(NDMAX = 20, NPAR = 5)
C
      REAL*8 PHI(NDMAX), EANG(2, NDMAX)
      REAL*8 NS(NPAR), DNS(NPAR)
      REAL*8 CORR(NPAR*(NPAR+1)/2), F(NDMAX)
      REAL*4 TIME
      INTEGER UNIT, IS(NPAR)
      BYTE ANS
      CHARACTER*16 FILE, FILO
      CHARACTER*5 STR(NPAR)
      CHARACTER*6 MODE
      PARAMETER(NUNIT = 16, NSCRN = 6)
C
      CALL STRING(NFLM, NPARM, IS, STR)
C
      WRITE (UNIT, 175)
 175  FORMAT(/, ' Ellipsometric Angles and Residuals', /,
     1       4X, 'Phi', 7X, 'Psi', 3X,
     3             'Resid.', 5X, 'Del', 3X, 'Resid.')
C
      DO J = 1, NDAT
        WRITE (UNIT, 177) PHI(J), EANG(1, J),
     1                    F(J), EANG(2, J), F(J + NDAT)
      END DO
 177  FORMAT(X, F7.2, F10.3, F8.3, F9.3, F8.3)
C
      WRITE (UNIT, 170) T2P, ETAPC
 170  FORMAT(/, ' t2P =', f6.3, '  EtaPC =', f6.3)
C
      WRITE (UNIT, 179)
 179  FORMAT(/, ' Fit results', /, 3X, 'x', 9X,
     1       'Final', 7X, 'Uncert.')
      DO I = 1, NPARM
        WRITE (UNIT, 180, IOSTAT = IOS) STR(I),
     1       NS(I), DNS(I)
      END DO
 180  FORMAT(X, A5, 3X, G13.6, 3X, G9.3)
C
      WRITE (UNIT, 181) SSQ
 181  FORMAT(/, ' SSQ =', E10.3)
C
C ----- DECODE DNLS1 ERROR OUTPUT
C
```

```
      PRINT *
      IF (INFO.EQ. 0) THEN
        WRITE (UNIT, 130) INFO
      ELSE IF (INFO .EQ. 1) THEN
        WRITE (UNIT, 131) INFO
      ELSE IF (INFO .EQ. 2) THEN
        WRITE (UNIT, 132) INFO
      ELSE IF (INFO .EQ. 3) THEN
        WRITE (UNIT, 133) INFO
      ELSE IF (INFO .EQ. 4) THEN
        WRITE (UNIT, 134) INFO
      ELSE IF (INFO .EQ. 5) THEN
        WRITE (UNIT, 135) INFO
      ELSE IF (INFO .EQ. 6) THEN
        WRITE (UNIT, 136) INFO
      ELSE IF (INFO .EQ. 7) THEN
        WRITE (UNIT, 137) INFO
      ELSE IF (INFO .EQ. 8) THEN
        WRITE (UNIT, 138) INFO
      END IF
C
 130  FORMAT(' INFO = ', I3,':  Improper input parameters.')
 131  FORMAT(' INFO = ', I3,':  FTOL criterion met.')
 132  FORMAT(' INFO = ', I3,':  XTOL criterion met.')
 133  FORMAT(' INFO = ', I3,':  FTOL and XTOL criteria both met.')
 134  FORMAT(' INFO = ', I3,':  GTOL criterion met.')
 135  FORMAT(' INFO = ', I3,':  Too many function calls.')
 136  FORMAT(' INFO = ', I3,':  FTOL too small; can''t reduce SSQ.')
 137  FORMAT(' INFO = ', I3,':  XTOL too small; can''t improve
     1 solution.')
 138  FORMAT(' INFO = ', I3,':  GTOL too small; F orthogonal
     1 to FJAC.')
C
 192  FORMAT(/, ' Correlation Coefficients')
      WRITE (UNIT, 192)
      WRITE (UNIT, 191) (STR(K), K = 1, NPARM)
 191  FORMAT(6X, 5(3X, A5))
C
      K = 0
      DO I = 1, NPARM
        WRITE (UNIT, 190) STR(I), (CORR(K+J), J = 1, I)
        K = K + I
 190    FORMAT(X, A5, 10F8.3)
      END DO
C
      WRITE (UNIT, 150) TIME
 150  FORMAT(/, ' Elapsed time =', f6.2, ' sec.')
C
      RETURN
      END
C
C
```

```
      SUBROUTINE CALOUT(PJ, AJ, CJ, DP, DA, EANG,
     1                  DEA, T2P, ETAPC, DT2P, DETA, NUNIT)    !9/28/88
C
C     H.L. TARDY, 1824
C     8/31/87
C
C     OUTPUT MODULE FOR CALIBRATION DATA
C
C     FILE ON NUNIT MUST HAVE BEEN OPENED IN DATIN
C     USING 'L' INSTRUCTION
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 PJ(4), AJ(4), CJ(2), EANG(2), DEA(2)
C
C ----- DUMP P, A, C
C
        WRITE (NUNIT, 80)
 80     FORMAT(/, ' Calibration Data')
        WRITE (NUNIT, 90)
 90     FORMAT(4X, 'P', 8X, 'A', 8X,
     1            'P''', 7X, 'A''', 7X, 'C')
C
          DO I = 1, 2
            IDX = 2*(I - 1)
            WRITE (NUNIT, 100)
     1        (PJ((IDX + J)), AJ((IDX + J)), J = 1, 2),
     2        CJ(I)
          END DO
 100      FORMAT(X, F7.2, 3F9.2, F8.2)
C
C ----- UNCERTAINTIES
C
          WRITE (NUNIT, 124)
          WRITE (NUNIT, 126) DP, DA
C
 124    FORMAT(/, ' Input Uncertainties')
 126    FORMAT(' dP =', F6.3, ', dA =',
     1            F6.3)
C
C ----- RESULTS
C
        WRITE (NUNIT, 110)
 110    FORMAT(/, ' Ellipsometric Angles', /,
     1            5X, 'Psi', 7X, 'Delta', 7X,
     2            't2P', 7X, 'EtaPC')
C
        WRITE (NUNIT, 120) (EANG(K), K = 1, 2), T2P, ETAPC
 120    FORMAT(X, F9.3, 3F11.3)
        IF ((DEA(1) .NE. 0.) .AND. (DEA(2) .NE. 0.)) THEN
          WRITE (NUNIT, 125, IOSTAT = IOS) (DEA(K), K = 1, 2),
     1         DT2P, DETA
        END IF
```

```
 125     FORMAT(X, '+/-', F6.3, '  +/-', F6.3, '  +/-', F6.3,
     1            '  +/-', F6.3)
C
       RETURN
       END
C
C
```

```
      SUBROUTINE GET_DEF(WL, DP, DA, NFLM, STACK, USTACK, NP,
     2                   IS, T2P, ETAPC, DT2P, DETA, MODE)
!10/14/88
C
C     RETRIEVE DEFAULT VALUES OF ELLIPSOMETRIC PARAMETERS.
C     FILE IS 'HARDWIRED' TO [DEFAULT DIR]ELLIPSE.DEF.
C
C     WL         = VACUUM WAVELENGTH
C     DP, DA     = POLARIZER, ANALYZER AZIMUTH UNCERTAINTIES
C     NFLM       = # OF FILMS IN STACK
C     STACK(I)   = STACK PARAMETERS:  J = J'TH FILM (TOP = 1)
C                  I = 3*J-2; N(J):  I = 3*J-1; K(J):  I = 3*J; D(J)
C                  I = 3*NFLM+1; N(SUBST):  I = 3*NFLM+2; K(SUBST)
C
       I = 3*NFLM+3; N(AMBIENT)
C     USTACK(I)  = STACK UNCERTAINTIES.  SAME CONVENTION, EXCEPT:
C                  I = 3*NFLM+3; INC. ANGLE UNC.:
C                  I = 3*NFLM+4; WAVELENGTH UNC.
C     NP         = # OF UNKNOWN PARAMETERS
C     IS(I)      = INDEX OF UNKNOWNS IN STACK
C     T2P        = POLARIZER ELLIPTICITY CORRECTION
C     ETAPC      = POL. AND COMPENSATOR AZIMUTH CORRECTION
C     DT2P, DETA = UNC.'S IN T2P, ETAPC
C
C     MODIFIED FOR MODEL STORAGE IN STACK, USTACK, IS ARRAYS:
C        9/22/88
C     ADDED ERROR HANDLER (STATEMENT 900):  2/2/89
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NFMAX = 20, NPMAX = 5)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      INTEGER IS(NPMAX)
      CHARACTER MODE*6, ANS
      INU = 15
C
      OPEN(UNIT = INU, FILE = 'ELLIPSE.DEF', STATUS = 'OLD',
     1       ERR = 900)
C
      READ(INU, 10) WL    ! WAVELENGTH
      READ(INU, 10) DP, DA    ! POLARIZER, ANALYZER UNC., DEGREES
      READ(INU, 20) NFLM, NP  ! # OF FILMS IN STACK, # UNKNOWNS
      DO I = 1, NFLM
        READ(INU, 10) (STACK(3*(I-1) + J), J = 1, 3)
C                     (N, K, D FOR FILM # I)
      END DO
      READ(INU, 10) STACK(3*NFLM+1), STACK(3*NFLM+2)
C                   (N(SUBST)         K(SUBST))
      DO I = 1, NFLM
        READ(INU, 10) (USTACK(3*(I-1) + J), J = 1, 3)
C                     (UNC. IN N, K, D FOR FILM # I)
      END DO
      READ(INU, 10) USTACK(3*NFLM+1), USTACK(3*NFLM+2)
C                   (UNC. IN N, K (SUBST))
```

```
      READ(INU, 10) USTACK(3*NFLM+3), USTACK(3*NFLM+4)
C                   (UNC. IN PHI,       LAMBDA)
      READ(INU, 20) (IS(I), I = 1, NP)  ! INDICES OF UNKNOWNS
      READ(INU, 10) T2P, ETAPC    ! POL. AND COMP. AZIMUTH CORRECTIONS
      READ(INU, 10) DT2P, DETA    ! UNC. IN T2P, ETAPC
      READ(INU, 30) MODE
C
 10   FORMAT(13X, 3G)
 20   FORMAT(13X, 5I)
 30   FORMAT(13X, A8)
C
      CLOSE(INU)
C
      RETURN
C
 900  PRINT 910
 910  FORMAT('$File ELLIPSE.DEF does not exist.  Create one? [Y/N] ')
      READ 920, ANS
 920  FORMAT(A1)
      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
        WL = 6328.
        DP = .04
        DA = .04
        NFLM = 1
        STACK(1) = 1.462
        STACK(2) = 0.
        STACK(3) = 500.
        STACK(4) = 3.883
        STACK(5) = .018
        USTACK(1) = 0.
        USTACK(2) = 0.0005
        USTACK(3) = 0.
        USTACK(4) = 0.005
        USTACK(5) = .002
        NP = 2
        IS(1) = 1
        IS(2) = 3
        T2P = 0.
        ETAPC = 0.
        DT2P = 0.
        DETA = 0.
        MODE = 'SINGLE'
        CALL PUT_DEF(WL, DP, DA, NFLM, STACK, USTACK, NP,
     1                  IS, T2P, ETAPC, DT2P, DETA, MODE)
      ELSE
        STOP
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE PUT_DEF(WL, DP, DA, NFLM, STACK, USTACK, NP,
     2                   IS, T2P, ETAPC, DT2P, DETA, MODE)  !10/14/88
C
C     SAVE DEFAULT VALUES OF ELLIPSOMETRIC PARAMETERS
C
C     MODIFIED FOR MODEL STORAGE IN STACK, USTACK, IS ARRAYS:
C        9/22/88
C
      PARAMETER(INU = 15, NDEF = 12, NFMAX = 20, NPMAX = 5)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      INTEGER IS(NPMAX)
      CHARACTER*6 MODE
      CHARACTER*12 LABEL(NDEF)
C
      DATA LABEL /'WL         =', 'DP, DA     =', 'NFLM, NP   =',
     1            'N, K, D    =', 'NS, KS     =', 'DN, DK, DD =',
     2            'DNS, DKS   =', 'DPHI, DLAM =', 'IS         =',
     3            'T2P, ETAPC =', 'DT2P, DETA =', 'MODE       ='/
C
C
      OPEN(UNIT = INU, FILE = 'ELLIPSE.DEF', STATUS = 'UNKNOWN')
C
      WRITE(INU, 10) LABEL(1), WL
      WRITE(INU, 10) LABEL(2), DP, DA
      WRITE(INU, 20) LABEL(3), NFLM, NP
      DO I = 1, NFLM
        WRITE(INU, 10) LABEL(4), (STACK(3*(I-1) + J), J = 1, 3)
      END DO
      WRITE(INU, 10) LABEL(5), STACK(3*NFLM+1), STACK(3*NFLM+2)
      DO I = 1, NFLM
        WRITE(INU, 10) LABEL(6), (USTACK(3*(I-1) + J), J = 1, 3)
      END DO
      WRITE(INU, 10) LABEL(7), USTACK(3*NFLM+1), USTACK(3*NFLM+2)
      WRITE(INU, 10) LABEL(8), USTACK(3*NFLM+3), USTACK(3*NFLM+4)
      WRITE(INU, 20) LABEL(9), (IS(J), J = 1, NP)
      WRITE(INU, 10) LABEL(10), T2P, ETAPC
      WRITE(INU, 10) LABEL(11), DT2P, DETA
      WRITE(INU, 30) LABEL(12), MODE
C
 10   FORMAT(X, A12, G14.5, ', ', G14.5, ', ', G14.5)
 20   FORMAT(X, A12, 4(I2, ', '), I2)
 30   FORMAT(X, A12, A)
C
      CLOSE(INU)
C
      RETURN
      END
C
C
```

```
C
C     ELLSUBS.FOR
C
C     SOURCE FILE FOR LIBRARY OF ELLIPSOMETRY SUBR.'S
C     ELLSUBS.OLB.
C
C     CUT FROM ELLIPSE.FOR: 7/12/88
C
C ********************** DISCLAIMERS *************************
C
C       This computer software has been developed under sponsorship of
C       the Department of Energy.  Any further distribution by any holder
C       of this software package or other data therein outside of DOE
C       offices or other DOE contractors, unless otherwise specifically
C       provided for, is prohibited without the approval of the National
C       Energy Software Center.  Requests from outside the Department for
C       DOE-developed computer software shall be directed to the
C       Director, National Energy Software Center, Argonne National
C       Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439.
C
C       This material was prepared as an account of work sponsored by an
C       agency of the United States Government.  Neither the United
C       States nor the United States Department of Energy, nor any of
C       their employees, makes any warranty, express or implied, or
C       assumes any legal liability or responsibility for the accuracy,
C       completeness, or usefulness of any information, apparatus,
C       product, or process disclosed, or represents that its use would
C       not infringe privately owned rights.
C
C ************************************************************************
C
C
      SUBROUTINE ADDLAYER(NL, STACK, USTACK)      !10/24/88
C
C     ADD ONE LAYER TO OPTICAL STACK
C
C     H.L. TARDY, 6224
C     10/10/88
C
      PARAMETER(NFMAX = 20, INU = 5)
      REAL*8 STACK(3*NL+6), USTACK(3*NL+7)
C
      IF (NL .GE. NFMAX) THEN
        PRINT *, 'ADDLAYER:  Too many layers.'
      ELSE
        IF (NL .EQ. 0) THEN
          IL = 1
        ELSE
 120      PRINT 122
 122      FORMAT('$Insert new layer above which layer? ')
          READ (INU, *, ERR = 120) IL
          IF (IL .GT. NL+1) THEN
```

```
            PRINT *, 'ADDLAYER:  Invalid layer index.'
            GOTO 120
          END IF
        END IF
        II = 3*IL - 2
        DO I = 3*NL+3, II, -1
          STACK(I+3) = STACK(I)
        END DO
        DO I = 3*NL+4, II, -1
          USTACK(I+3) = USTACK(I)
        END DO
 150    PRINT *, 'Enter n, dn, k, dk, d, dd:'
        READ (INU, *, ERR = 150) (STACK(II+J), USTACK(II+J), J = 0, 2)
C
        NL = NL+1
C
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE ANGS(NFLM, WL, PHI, STACK, EANG)     !9/26/88
C
C     COMPUTES ELLIPSOMETRIC ANGLES OF STACK
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20)
      REAL*8 STACK(3*NFLM+3), EANG(2), D(NFMAX)
      COMPLEX*16 RHO, R(2), N(NFMAX+2)
      INTEGER S, P
      DATA S, P /1, 2/
C
      CALL MODLOAD(NFLM, STACK, N, D)
C
      CALL MATRT(NFLM, WL, PHI, N, D, R)
      RHO = R(P)/R(S)
      EANG(1) = ATAND(ABS(RHO))
      EANG(2) = ATAN2D(DIMAG(RHO), DREAL(RHO))
C
      RETURN
      END
C
C
```

```
      SUBROUTINE ANGSD(NFLM, WL, PHI, STACK, EANG, DEA)      ! 10/4/88
C
C     COMPUTES ELLIPSOMETRIC ANGLES OF STACK, W/ DERIVS.
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20)
      REAL*8 STACK(3*NFLM+3), EANG(2), D(NFMAX), DEA(2, 3*NFLM+4)
      COMPLEX*16 RHO, R(2), N(NFMAX+2), DR(2, 2*NFMAX+3)
      INTEGER S, P
      DATA S, P /1, 2/
C
      CALL MODLOAD(NFLM, STACK, N, D)
C
      CALL MATRD(NFLM, WL, PHI, N, D, R, DR)
      RHO = R(P)/R(S)
      EANG(1) = ATAND(ABS(RHO))
      EANG(2) = ATAN2D(DIMAG(RHO), DREAL(RHO))
C
      CALL DVRHO(NFLM, R, DR, DEA)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE AZCON(P, A, C, DP, DA, EANG, DEA,
     1                 T2P, ETAPC, DT2P, DETA, NDAT)
C
C     H.L. TARDY, 1824
C     7/27/87
C     RE WRITE:  9/1/87
C
C     THIS VERSION PROCESSES ONLY THE 2-ZONE DATA SETS.  2-ZONE
C     DATA IS REPROCESSED ON EACH 'RUN' COMMAND TO UPDATE
C     CORRECTIONS FROM NEW 4-ZONE MEASUREMENTS.
C
C     EANG(1, I) = PSI(I),  DEA(1, I) = DPSI(I)
C     EANG(2, I) = DEL(I),  DEA(2, I) = DDEL(I)
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NDMAX = 20)
      REAL*8 P(4, NDMAX), A(4, NDMAX), C(2, NDMAX)
      REAL*8 PJ(4), AJ(4), CJ(2), EAJ(2), DEAJ(2)
      REAL*8 EANG(2, NDAT), DEA(2, NDAT)
      INTEGER IND2(NDMAX), IND4(NDMAX), INDPD(NDMAX)
C
      K2 = 0
      K4 = 0
      KDP = 0
C
C ----- SORT INDICES OF DATA, CONVERT FOUR-ZONE DATA,
C     COMPUTE AVERAGE T2P AND ETAPC
C
      DO I = 1, NDAT
        IF (C(2, I) .EQ. 0.) THEN  ! TWO-ZONE DATA
          K2 = K2 + 1
          IND2(K2) = I
        ELSE IF (C(2, I) .EQ. 999.) THEN  ! PSI, DEL DATA
          KDP = KDP + 1
          INDPD(KDP) = I
        ELSE IF (ABS(C(2, I)) .EQ. 45.) THEN  ! FOUR-ZONE DATA
          K4 = K4 + 1
          IND4(K4) = I
        ELSE                                  ! BAD DATA
          PRINT *, 'AZCON:  Bad data at I =', I, '.'
        END IF
      END DO
C
C ----- CONVERT AND CORRECT TWO-ZONE DATA, USING T2P
C     AND ETAPC
C
      DO I = 1, K2
        DO J = 1, 2
          PJ(J) = P(J, IND2(I))
          AJ(J) = A(J, IND2(I))
        END DO
```

```
         CJ(1) = C(1, IND2(I))
         CALL TWOZONE(PJ, AJ, CJ, DP, DA, EAJ, DEAJ,
     2                T2P, ETAPC, DT2P, DETA)
C
         DO II = 1, 2
           EANG(II, IND2(I)) = EAJ(II)
           DEA(II, IND2(I)) = DEAJ(II)
         END DO
C
       END DO
C
       RETURN
       END
C
C
```

```
      SUBROUTINE BUFLOAD(NDAT, P, A, C, PJ, AJ, CJ)
C
C     LOADS SINGLE AZIMUTHS INTO BUFFER
C
C     H.L. TARDY, 6224
C     7/12/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 P(4, *), A(4, *), C(2, *)
      REAL*8 PJ(4), AJ(4), CJ(2)
C
      DO J = 1, 4
        P(J, NDAT) = PJ(J)
        A(J, NDAT) = AJ(J)
      END DO
C
      C(1, NDAT) = CJ(1)
      C(2, NDAT) = CJ(2)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE CHGLAYER(NL, STACK, USTACK)          !10/19/88
C
C     CHANGE ONE LAYER OF OPTICAL STACK
C
C     H.L. TARDY, 6224
C     10/10/88
C
      PARAMETER (NFMAX = 20, INU = 5)
      REAL*8 STACK(3*NL+3), USTACK(3*NL+4)
C
 110  IF (NL .GT. 0) THEN
        PRINT 112
        READ (INU, *, ERR = 110) IL
      ELSE
        IL = 1
      END IF
 112  FORMAT('$Enter number of layer to change: ')
C
      II = 3*(IL-1)
      IF (IL .LE. NL) THEN
 120    PRINT *, 'Enter new n, dn, k, dk, d, dd:'
        READ (INU, *, ERR = 120) (STACK(II+J), USTACK(II+J), J = 1, 3)
      ELSE IF (IL .EQ. NL+1) THEN
 130    PRINT *, 'Enter new ns, dns, ks, dks:'
        READ (INU, *, ERR = 130) (STACK(II+J), USTACK(II+J), J = 1, 2)
      ELSE
        PRINT *, 'CHGLAYER:  Invalid layer index.'
        GOTO 110
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE CORRLS1(SSQ, FJAC, LDFJ, NPAR, IPVT,
     1                   DPAR, CORR)                    !11/2/88
C
C
C     OBTAINS CORRELATION MATRIX, UNCERTAINTIES FROM
C     DNLS1 OUTPUT
C
C     H.L. TARDY, 6224
C     7/27/88
C
C
      PARAMETER(NPMAX = 5)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 FJAC(LDFJ, NPAR), DPAR(NPAR), CORR(NPAR*(NPAR+1)/2)
      REAL*8 XJTJR(NPMAX*(NPMAX+1)/2)
      REAL*8 WORK(NPMAX), DET(2)
      INTEGER INERT(NPMAX), IPVT(NPAR), IPVI(NPMAX)
C
C ----- CHECK FJAC OUTPUT
C     PRINT *
C     PRINT *, 'FJAC output'
C     DO I = 1, NPAR
C       PRINT 190, (FJAC(I, J), J = 1, NPAR)
C 190 FORMAT(X, <12*(I-1)>X, <NPAR-I+1>G12.5)
C     END DO
C
C ----- CREATE R MATRIX FROM FJAC (SYMMETRIC STORAGE MODE)
C
      K = 0
      DO J= 1, NPAR
        DO I = 1, J
          K = K + 1
          CORR(K) = FJAC(I, J)
        END DO
      END DO
C
C ----- CHECK XJTJ OUTPUT
D     PRINT *
D     PRINT *, 'Original R:'
D     CALL MDUMP(CORR, NPAR)
C
C ----- CALC TRANS(R)*R MATRIX CORR
C
      CALL XTXUP(CORR, NPAR, XJTJR)
C
C ----- CHECK XJTJ OUTPUT
D     PRINT *
D     PRINT *, 'Trans(R)*R:'
D     CALL MDUMP(XJTJR, NPAR)
C
C ----- INVERT HESSIAN TO OBTAIN CORRELATION
C
```

```
      CALL DSPCO(XJTJR, NPAR, IPVI, COND, WORK)
D     PRINT *, 'Condition = 1/', COND
C
      IF (COND .GT. 0.) THEN    ! INVERSE SHOULD BE OK
C
        CALL DSPDI(XJTJR, NPAR, IPVI, DET, INERT, WORK, 1)
C
C ----- CHECK R OUTPUT
D     PRINT *
D     PRINT *, 'Inverse:'
D     CALL MDUMP(XJTJR, NPAR)
        CALL REARR(XJTJR, NPAR, IPVT, CORR)
C
C ----- CALC. UNCERTAINTIES
C
      K = 0
      DO I = 1, NPAR
        K = K + I
        DPAR(I) = SQRT(CORR(K))
      END DO
C
C ----- CONVERT INVERSE HESSIAN TO CORRELATION COEFF'S.
C
        K = 0
        DO J = 1, NPAR
          DO I = 1, J
            K = K + 1
            ZZ = DPAR(I)*DPAR(J)
            CORR(K) = CORR(K)/ZZ
C           PRINT *, I, J, CORR(K)
          END DO
        END DO
C
        DO I = 1, NPAR
          DPAR(I) = SSQ*DPAR(I)
        END DO
C
      ELSE
        PRINT 100
        WRITE(NUNIT, 100) COND
 100    FORMAT(/, ' CORRLS1:  Condition = 1/',
     1           e9.2, '.  Hessian is singular.')
      END IF
C
      RETURN
      END
C
C
```

```
      FUNCTION CMNEWT(NF, NPARAM, PARAM)              !10/21/88
C
C     CHARLOT-MARUANI FUNCTION FOR SUBR. NEWT SOLUTION
C
C     H.L. TARDY, 6224
C     10/21/88
C
C
C     USES CHARLOT-MARUANI ALGORITHM FOR N EQUATION.
C     SEE D. CHARLOT AND A. MARUANI, APPL. OPT. 24, 3368 (1985).
C
C     (NF**2 - N0**2) IS FACTORED OUT OF THE C-M EQUATION TO AVOID
C     FALSE SOLUTIONS.
C
C     ASSUMED MEDIUM IS AIR (N0 = 1.000)
C     ARRAY PARAM HAS 3*NFLM+9 ELEMENTS:  FIRST 3*NFLM+3 ARE ARRAY STACK.
C        PARAM(3*NFLM+3+J) = WL, PHI, PSI, DEL, RE(CAM/CBAB),
IM(CAM/CBAB)
C
C     ROUTINE ASSUMES CONSISTENT SIGN CONVENTION FOR n - i k HAS BEEN
C     APPLIED.
C
C     CHANGED TO STACK STORAGE:  9/22/88
C     ADDED EXPONENTIAL TERM TO FORCE MONOTONICITY:  9/30/88
C
C
C     DECLARATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER(NFMAX = 20, NPMAX = 5)
      REAL*8 CMNEWT, NF, DM(NFMAX-1), N0
      REAL*8 PARAM(NPARAM)
      COMPLEX*16 IM, RHO, R(2), AP, B, CP
      COMPLEX*16 NAR(NFMAX+1), CF
      INTEGER S, P
      PARAMETER (IM = (0., 1.), N0 = 1.000, PI = 3.1415926D0)
C
C ----- SHARED WITH MAIN
C
      COMPLEX*16 CBAB
C
C ***** DEFINITIONS
C
      DATA S, P /1, 2/
      NFLM = (NPARAM - 9)/3
      WL = PARAM(NPARAM-5)
      PHI1 = PARAM(NPARAM-4)
      PSI1 = PARAM(NPARAM-3)
      DEL1 = PARAM(NPARAM-2)
C
      RHO = TAND(PSI1)*EXP(IM*DEL1*PI/180.)
C
```

```
      S0 = SIND(PHI1)
      C0 = COSD(PHI1)
      SF = N0*S0/NF
      CF = (1. - SF**2)
      CF = SQRT(CF)    ! REAL FOR TRANSPARENT FILMS IN AIR
      IF (DIMAG(CF) .GT. 0.) THEN
        CF = -CF
      END IF
C
C ----- SET UP SUBSTRATE STACK
C
      NAR(1) = NF
      DO I = 1, NFLM-1
        II = 3*I
        NAR(I+1) = CMPLX(PARAM(II+1), -ABS(PARAM(II+2)))
        DM(I) = PARAM(II+3)
      END DO
      NAR(NFLM+1) = CMPLX(PARAM(3*NFLM+1), -ABS(PARAM(3*NFLM+2)))
      IF (ABS(SF) .LT. 1.) THEN
        PHIS = ASIND(SF)
      ELSE
        PHIS = 90.
      END IF
C
      CALL MATRT(NFLM-1, WL, PHIS, NAR, DM, R)
C
C ---- DEFINE QUAD COEFF.'S:  0 = A*X**2 + B*X + C
C
C     (N0 - NF) DOES NOT FACTOR OUT OF B
C
      R1S = (N0*C0 - NF*CF)/(N0*C0 + NF*CF)
      R1P = (NF*C0 - N0*CF)/(NF*C0 + N0*CF)
C
      B = RHO*(R(S) + R1S*R1P*R(P)) - R(P) -
     1      R1P*R1S*R(S)
C
      AP = N0*S0**2 * (1. - RHO) + C0*CF*NF*(1. + RHO)
      AP = AP/NF/(CF*N0 + C0*NF)/(C0*N0 + CF*NF)
      AP = AP*R(S)*R(P)
C
      CP = N0*S0**2 * (1. - RHO) - C0*CF*NF*(1. + RHO)
      CP = CP/NF/(CF*N0 + C0*NF)/(C0*N0 + CF*NF)
C
C ---- DEFINE EQUATION FOR NF
C
      Z = (NF**2 - N0**2)
      CBAB = AP*CONJG(B) - B*CONJG(CP)
      CAM = Z * (ABS(CP)**2 - ABS(AP)**2)
C
      CMNEWT = (CAM**2 - ABS(CBAB)**2)
C
C ----- FORCE MONOTONICITY.  ORIGINAL CMFCN = CMFCN*Z**2
```

```
C
      CMNEWT = CMNEWT*EXP(Z)
C
      PARAM(NPARAM-1) = DREAL(CAM/CBAB)
      PARAM(NPARAM) = DIMAG(CAM/CBAB)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE DVRHO(NFILM, R, DR, DVEA)
C
C     CALCULATES DERIVATIVES OF RHO = R(P)/R(S) WRT INDICES,
C     THICKNESSES, AND ANGLE OF INCIDENCE.  R AND DR MUST BE
C     OBTAINED THROUGH A CALL TO MATRD.
C
C     INPUTS:
C     NFILM = # OF FILMS IN STACK (1 = TOP FILM)
C     R(POL) = COMPLEX REFLECTANCE FOR POL.
C     DR(POL, J) = DERIVATIVE OF REFLECTANCES WRT PARAM'S.:
C        J = 2*IFILM-1, N(IFILM+1);  J = 2*IFILM, D(IFILM);
C        J = 2*NFILM+1, N(SUBSTR);  J = 2*NFILM+2, PHI;
C        J = 2*NFILM+3, LAM.
C        (N(IFILM+1) BECAUSE N(1) IS FOR AMBIENT)
C
C     OUTPUTS:
C     DVEA(I, J) = DERIV OF PSI (I = 1), DEL (I = 2) WRT TO PARAM'S:
C        J = 3*IFILM-2, N(IFILM+1);  J = 3*IFILM-1, K(IFILM+1);
C        J = 3*IFILM, D(IFILM);  J = 3*NFILM+1, N(SUBSTR);
C        J = 3*NFILM+2, K(SUBSTR);  J = 3*NFILM+3, PHI;
C        J = 3*NFILM+4, LAM.
C
C
C     H.L. TARDY,, SNLA 1824
C     4/4/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      COMPLEX*16 R(2), DR(2, 2*NFILM+3)
      COMPLEX*16 LRP, IM
      REAL*8 DVEA(2, 3*NFILM+4)
      INTEGER S, P
      PARAMETER(PI = 3.141592654D0, DTOR = PI/180.D0)
C
      S = 1
      P = 2
      NLAST = 3*NFILM + 4
C
      TPS = ABS(R(P)/R(S))
      TPS = TPS/(1.D0 + TPS**2)
C
      DO I = 1, NFILM
C
C ----- WRT FILM N'S AND K'S
C
        LRP = DR(P, 2*I-1)/R(P) - DR(S, 2*I-1)/R(S)
        DVEA(1, 3*I-2) = DREAL(LRP) * TPS
        DVEA(2, 3*I-2) = DIMAG(LRP)
C
C       LRP = -IM*LRP
        DVEA(1, 3*I-1) = DIMAG(LRP) * TPS
        DVEA(2, 3*I-1) = -DREAL(LRP)
C
```

```
C ----- WRT FILM D'S
C
        LRP = DR(P, 2*I)/R(P) - DR(S, 2*I)/R(S)
        DVEA(1, 3*I) = DREAL(LRP) * TPS
        DVEA(2, 3*I) = DIMAG(LRP)
C
      END DO
C
C ----- WRT SUBSTRÀTE N AND K
C
      LRP = DR(P, 2*NFILM+1)/R(P) - DR(S, 2*NFILM+1)/R(S)
      DVEA(1, 3*NFILM+1) = DREAL(LRP) * TPS
      DVEA(2, 3*NFILM+1) = DIMAG(LRP)
C
C     LRP = -IM*LRP
      DVEA(1, 3*NFILM+2) = DIMAG(LRP) * TPS
      DVEA(2, 3*NFILM+2) = -DREAL(LRP)
C
C ----- WRT PHI
C
      LRP = DR(P, 2*NFILM+2)/R(P) - DR(S, 2*NFILM+2)/R(S)
      DVEA(1, 3*NFILM+3) = DREAL(LRP) * TPS
      DVEA(2, 3*NFILM+3) = DIMAG(LRP)
C
C ----- WRT WAVELENGTH
C
      LRP = DR(P, 2*NFILM+3)/R(P) - DR(S, 2*NFILM+3)/R(S)
      DVEA(1, 3*NFILM+4) = DREAL(LRP) * TPS
      DVEA(2, 3*NFILM+4) = DIMAG(LRP)
C
C ----- CHANGE DPSI, DDEL TO DEGREES
C
      DO I = 1, NLAST
        DVEA(1, I) = DVEA(1, I)/DTOR
        DVEA(2, I) = DVEA(2, I)/DTOR
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE DVSOL(NL, DEA, IS, JACO, DFIX)        !10/25/88
C
C
C     COMPUTE INVERSE JACOBIAN AND DERIVS WRT FIXED PARAMS
C     OF AN NL-LAYER SYSTEM IN WHICH PSI, DELTA ARE SOLVED
C     FOR PARAMETERS IS(1) AND IS(2)
C
C     H.L. TARDY, 6224
C     7/26/88
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 DEA(2, 3*NL+4), JACO(2, 2), DFIX(2, 3*NL+4)
      INTEGER IS(2)
C
      NLAST = 3*NL + 4
C
C ----- DEFINE JACOBIAN
C
      DO J = 1, 2
        DO K = 1, 2
          JACO(K, J) = DEA(K, IS(J))
        END DO
      END DO
C
C ----- INVERT
C
      DET = JACO(1, 1)*JACO(2, 2) - JACO(1, 2)*JACO(2, 1)
      TEM = JACO(1, 1)
      JACO(1, 1) = JACO(2, 2)/DET
      JACO(1, 2) = -JACO(1, 2)/DET
      JACO(2, 1) = -JACO(2, 1)/DET
      JACO(2, 2) = TEM/DET
C
C ----- COMPUTE INTERMEDIATE MATRIX
C     (DER. OF SOL'N PARAMETERS WRT FIXED ONES =
C      INV. JACO * DEA)
C
      DO K = 1, 2
        DO I = 1, NLAST
          DFIX(K, I) = 0.
          DO J = 1, 2
            DFIX(K, I) = DFIX(K, I) - JACO(K, J)*DEA(J, I)
          END DO
        END DO
C       PRINT *, (DFIX(K, I), I = 1, NLAST)
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE FCNG(IFLG, NDAT, NPAR, NS, F, FJAC, LDFJ)  !11/10/88
C
C     CALCULATES RESIDUAL FUNCTION FOR DNLS1.
C
C     NDAT HERE IS 2*ACTUAL NUMBER OF (PSI, DEL) PAIRS
C
C     CONVERTED TO STORE MODEL IN ARRAY STACK, USING STANDARD
C        INDICES:  9/22/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 NO, NS(NPAR), F(NDAT), EANGC(2)
      REAL*8 FJAC(LDFJ, NPAR)
      PARAMETER(NO = 1.000D0)
C
C ----- SHARED WITH MAIN
C
      PARAMETER(NDMAX = 20)
      REAL*8 PHI(NDMAX), EANG(2, NDMAX)
      COMMON /MAIN_FCNS/ PHI, EANG
C
      PARAMETER (NFMAX = 20, NPMAX = 5)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
      INTEGER IS(NPMAX)
      COMMON /MODEL/ NFLM, STACK, IS
      COMMON /WAVE/ WL
C
      REAL*8 DEA(2, 3*NFMAX+4), STK(3*NFMAX+3)
C
C ----- SAVE STACK
C
      DO I = 1, 3*NFLM+3
        STK(I) = STACK(I)
      END DO
C
C ----- SET UP MODEL
C
      DO I = 1, NPAR
        STK(IS(I)) = NS(I)
      END DO
C
      IF (IFLG .EQ. 0) THEN
        PRINT *
        SSQ = 0.
        DO I = 1, NDAT/2
          SSQ = SSQ + F(I)**2 + F(I+NDAT/2)**2
          PRINT *
          PRINT *, F(I), F(I+NDAT/2)
          PRINT *
C         DO J = 1, NPAR
C           PRINT *,  FJAC(I, J), FJAC(I+NDAT/2, J)
C         END DO
        END DO
```

```
        PRINT *, 'SSQ =', SSQ
      ELSE IF (IFLG .EQ. 1) THEN
C
C ----- COMPUTE VALUES OF F
C
        DO I = 1, NDAT/2
          CALL ANGS(NFLM, WL, PHI(I), STK, EANGC)
          F(I) = EANGC(1) - EANG(1, I)
          F(I+NDAT/2) = EANGC(2) - EANG(2, I)
        END DO
C
      ELSE IF (IFLG .EQ. 2) THEN
C
C ----- COMPUTE JACOBIAN
C
        DO I = 1, NDAT/2
          CALL ANGSD(NFLM, WL, PHI(I), STK, EANGC, DEA)
          DO J = 1, NPAR
            FJAC(I, J) = DEA(1, IS(J))
            FJAC(I+NDAT/2, J) = DEA(2, IS(J))
          END DO
        END DO
C
      ELSE IF (IFLG .EQ. 3) THEN
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE FOURZONE(P, A, C, DP, DA, EANG, DEA,
     1                    T2P, ETAPC, DT2P, DETA)
C
C     H.L. TARDY, 1824
C     MODIFIED FROM MAILM.FOR VERSION, 7/17/87
C
C     PERFORMS FOUR-ZONE AVERAGES
C     See Azzam and Bashara, JOSA 61, 600 (1971)
C
C     ASSUMPTION IS THAT COMP. ANGLES ARE +/- 45.00.
C     ROUTINE DOES NOT CHECK FOR THIS CONDITION.
C
C     OUTPUT T2P, ETAPC, AND UNCERTAINTIES REPRESENT WEIGHTED
C     AVERAGES.  THIS INSURES THAT NO INFO ON CORRECTIONS IS LOST
C     WHEN TAKING SEVERAL MEASUREMENTS IN A SESSION, AND ALLOWS A
C     RUNNING AVERAGE TO BE KEPT WITHOUT UNDERWEIGHTING THE
C     PREVIOUSLY-ACCUMULATED AVERAGE.
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 EANG(2), DEA(2), P(1), A(1), C(1)
      REAL*8 PSIZ(2), DELZ(2), PZ(2), AZ(2), EAZ(2), X(2)
C
C ----- INITIALIZE WEIGHTED AVERAGES
C
      IF (DT2P .NE. 0.) THEN
        DT2P = 1./DT2P**2
        T2P = T2P*DT2P
      ELSE
        T2P = 0.
      END IF
C
      IF (DETA .NE. 0.) THEN
        DETA = 1./DETA**2
        ETAPC = ETAPC*DETA
      ELSE
        ETAPC = 0.
      END IF
C
C ----- PERFORM FOUR-ZONE AVERAGE
C
      DO I = 1, 2
        DO K = 1, 2
          PZ(K) = P(2*(I-1) + K)
          AZ(K) = A(2*(I-1) + K)
        END DO
        CALL TWOZONE(PZ, AZ, C(I), DP, DA, EAZ,
     1                 X, 0., 0., 0., 0.)
        IF (C(I) .LT. 0.) THEN  ! I IS ODD-ZONE DATA
          PODD = EAZ(1)
          DODD = EAZ(2)
        ELSE                    ! I IS EVEN-ZONE DATA
```

```
            PEVEN = EAZ(1)
            DEVEN = EAZ(2)
          END IF
        END DO
        EANG(1) = (PODD + PEVEN)/2.
        EANG(2) = (DODD + DEVEN)/2.
        DEA(1) = DA/2.
        DEA(2) = DP
C
C ----- CORRECTION FACTORS
C
        RES1A = PODD - PEVEN
        T2PN = RES1A/2./SIND(2.*EANG(1))
        DT2PN = DA/2./SIND(2.*EANG(1))
C
        RES1P = (DEVEN - DODD)/2.
        ETAPCN = RES1P
        DETAN = DP
C
C ----- WEIGHTED AVERAGES
C
        T2P = T2P + T2PN/DT2PN**2
        DT2P = DT2P + 1./DT2PN**2
        T2P = T2P/DT2P
        DT2P = SQRT(1./DT2P)
C
        ETAPC = ETAPC + ETAPCN/DETAN**2
        DETA = DETA + 1./DETAN**2
        ETAPC = ETAPC/DETA
        DETA = SQRT(1./DETA)
C
        RETURN
        END
C
C
```

```
      SUBROUTINE GETAZ(P, A, C, NZ)
C
C     KEYBOARD INPUT OF TWO- OR FOUR-ZONE AZIMUTHS
C
C     H.L. TARDY, 6224
C     7/12/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (INU = 5)
      REAL*8 P(4), A(4), C(2)
C
C ----- INITIALIZE
C
      DO I = 1, 4
        P(I) = 0.
        A(I) = 0.
      END DO
      C(1) = 0.
      C(2) = 0.
C
 840  PRINT 90
 90   FORMAT(/, ' Enter P, A, P'', A'', C')
      READ (INU, *, ERR = 840) (P(K), A(K),
     1                          K = 1, 2), C(1)
C     PRINT *, (P(K), A(K), K = 1, 2), C(1)
C
      IF (NZ .EQ. 4) THEN
 850    PRINT 94
 94     FORMAT(/, ' Enter P, A, P'', A'', C, for
     1 second two zones')
        READ (INU, *, ERR = 850) (P(K), A(K),
     1                          K = 3, 4), C(2)
C       PRINT *, (P(K), A(K), K = 3, 4), C(2)
C
      END IF
C
C ----- PROCESS COMPENSATOR AZIMUTHS
C
      DO I = 1, 2
        DO WHILE (C(I) .GT. 45.)
          C(I) = C(I) - 180.
        END DO
        DO WHILE (C(I) .LT. -45.)
          C(I) = C(I) + 180.
        END DO
      END DO
C
C ----- CHECK FOR VALID DATA
C
      DO K = 1, 4
        IF (P(K) .GT. 360.) THEN
          PRINT 83
```

```
          GOTO 840
        ELSE IF (A(K) .GT. 360.) THEN
          PRINT 83
          GOTO 840
        END IF
      END DO
C
      IF (ABS(C(1)) .NE. 45.) THEN
        PRINT 84
        GOTO 840
      END IF
      IF (NZ .EQ. 4) THEN
        IF (C(2) .NE. -C(1)) THEN
          PRINT 85, -C(1)
          GOTO 850
        END IF
      END IF
C
 83   FORMAT(/, ' GETAZ:  Invalid data.  Reenter.')
 84   FORMAT(/, ' GETAZ:  Set compensator to +/-45 and
     1 reenter.')
 85   FORMAT(/, ' GETAZ:  Set compensator to ', f6.2, '.')
C
      RETURN
      END
C
C
```

```
      SUBROUTINE GET_IS(NFLM, NP, IS)                    !10/24/88
C
C     INPUT OF SOLUTION INDICES FROM KEYBOARD
C
C     H.L. TARDY, 6224
C     10/10/88
C
      INTEGER IS(NP)
      PARAMETER (INU = 5)
C
      IF (NP .GT. 3*NFLM+2) THEN
        PRINT *, 'GET_IS:  Too many parameters for model.'
      ELSE
C
 29     IF (NP .EQ. 1) THEN
          PRINT 30, NP
 30       FORMAT(/, ' Enter ', I1, ' parameter from the list
     1 below:', /)
        ELSE
          PRINT 31, NP
 31       FORMAT(/, ' Enter ', I1, ' parameters from the list
     1 below:', /)
        END IF
C
        DO I = 1, NFLM
          PRINT 40, ((3*I-J, I), J = 2, 0, -1)
 40       FORMAT(X, I3, ': n(', I2, ')', I6, ': k(', I2, ')',
     1           I6, ': d(', I2, ')')
        END DO
        PRINT 50, (3*NFLM+J, J = 1, 2)
 50     FORMAT(X, I3, ': n(sub)', I5, ': k(sub)', /)
C
        READ (INU, *, ERR = 29) (IS(J), J = 1, NP)
C
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE LOGCL(NUNIT)                    !10/10/88
C
C     CLOSE LOG FILE AND WRITE TIME
C
C     H.L. TARDY, 6224
C     10/10/88
C
      CHARACTER*8 CTIME
      CALL TIME(CTIME)
      WRITE(NUNIT, 17) CTIME
 17   FORMAT(/, X, A8)
      CLOSE(NUNIT)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE LOGOP(FILE, HEAD, NUNIT)                    !10/25/88
C
C     OPEN LOG FILE AND WRITE HEADER LINES
C
C     H.L. TARDY, 6224
C     10/10/88
C
C
      CHARACTER*8 CTIME
      CHARACTER*9 CDATE
      CHARACTER*16 FILE
      CHARACTER*20 WHO
      CHARACTER*80 HEAD
C
      CALL DATE(CDATE)
      CALL TIME(CTIME)
17    FORMAT(/, X, A8)
C
      PRINT 20
 20   FORMAT(/, '$Enter file name (<= 12 chars. + ext):  ')
      READ 25, FILE
 25   FORMAT(A16)
      OPEN(UNIT = NUNIT, FILE = FILE, STATUS = 'NEW',
     1       FORM = 'FORMATTED')
C
      PRINT *
      PRINT *, 'Enter header'
      READ 35, HEAD
 35      FORMAT(A80)
C
      WHO = '?'  ! ENSURE AT LEAST ONE CHAR. IN WHO
      PRINT *
      PRINT *, 'Enter operator name'
      READ 26, WHO
 26   FORMAT(A20)
C
      ILEN = 20
      DO WHILE (WHO(ILEN:ILEN) .EQ. ' ')
        ILEN = ILEN - 1
      END DO
C
      CALL HEADOUT(NUNIT)
      WRITE(NUNIT, 36) FILE
      WRITE (NUNIT, 37) HEAD
      WRITE (NUNIT, 38) WHO, CDATE, CTIME
 36   FORMAT(/, X, A12)
 37   FORMAT(X, A80)
 38   FORMAT(' Run by:  ', A<ILEN>, ', ', A9, ', ', A8)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MATRD(NFILM, LAM, PHI, N, D, R, DR)      !, T,
C
C     CALCULATE COMPLEX REFLECTANCE AND TRANSMITTANCE  OF
C     STACK OF FILMS USING MATRIX FORMULATION.  ALSO COMPUTES
C     REFLECTANCE DERIVATIVES WRT INCIDENT ANGLE, INDICES,
C     AND THICKNESS.
C     FORMULAS ARE IN
C        P.H. Berning, 'Theory and Calculations of
C        Optical Thin Films,' in 'PHYSICS OF THIN FILMS',
C        ADV. IN R&D, G. HASS (ED.), V.1, P.69 (Academic Press, 1963)
C        J. Humlicek, 'Evaluation of derivatives of reflectance and
C        transmittance by stratified planar structures and ...,'
C        Opt. Acta., v.30, p. 97, (1983).
C
C     NB:  COORDINATES SYSTEM CONVENTIONS DIFFER BETWEEN BERNING
C     AND HUMLICEK.  BERNING'S IS MORE STANDARD AND IS USED HERE.
C
C     SIGN OF P-REFLECTANCE IS REVERSED TO CONFORM TO ELLIPSOMETRIC
C     CONVENTION.
C
C     INPUT:
C     NFILM = # OF FILMS
C     LAM = WAVELENGTH
C     PHI = INCIDENT ANGLE, DEGREES
C     N(I) = COMPLEX INDICES:  N(1) = MEDIUM
C     D(I) = THICKNESSES:  D(1) = TOP LAYER
C     OUTPUT:
C     R(POL) = COMPLEX REFLECTANCE OF STACK:  R(1) = S-POL.
C     T(POL) = COMPLEX  TRANS.:  T(1) = S-POL.
C     DR(POL, I) = REFLECTANCE DERIVATIVE WRT:
C        I=2*J-1, N(J+1);  I=2*J, D(J);  I=2*NFILM+1, N(SUBSTR);
C        I=2*NFILM+2, PHI; I=2*NFILM+3, LAM.  NO DERIV. WRT N(1)
C
C     DERIV. WRT LAM IGNORES DN(I)/DLAM DISPERSION TERMS; CONSIDERS
C     ONLY EFFECT ON PHASE
C
C     MODIFIED FOR ELLIPSOMETRY USE:  LINES PERTAINING TO T(POL)
C     ARE COMMENTED OUT.  (HLT, 3/31/88)
C     ADDED DERIV. WRT LAM:  7/18/88
C
C     H.L. TARDY, SNLA 1824
C     3/31/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20)
      COMPLEX*16 N(NFILM+2), NHAT(2, NFMAX+2), NH
      COMPLEX*16 IM, E, H, ENEW, HNEW, R(2), T(2)
      COMPLEX*16 DR(2, 2*NFILM+3), ZZ1, ZZ2
      COMPLEX*16 DBDN(NFMAX), DBDPH(NFMAX), DBDD(NFMAX)
      COMPLEX*16 DBDLAM(NFMAX)
      COMPLEX*16 DNHDN(2, NFMAX+1), DNHDPH(2, NFMAX+2)
      COMPLEX*16 SF, CF, CB(NFMAX), SB(NFMAX), BETA, SNO
```

```
      REAL*8 D(*), LAM
      INTEGER S, P
      PARAMETER(IM = (0., 1.), PI = 3.141592654D0,
     1          DTOR = PI/180.D0)
C
      S = 1
      P = 2
C
      BET0 = 2.D0*PI/LAM
      S0 = SIND(PHI)
      SN0 = N(1)*S0
      C0 = COSD(PHI)
      NSUB = 2*NFILM + 1
      NPHI = 2*NFILM + 2
      NLAST = 2*NFILM + 3
C
C ----- CALCULATE EFFECTIVE INDICES, ETC.
C     DOING THIS IN SEPARATE LOOP AVOIDS UNNECESSARY RECALC'S.
C
C ----- TOP/AMBIENT
      NHAT(1, 1) = N(1)*C0
      NHAT(2, 1) = N(1)/C0
      DNHDPH(1, 1) = -SN0
      DNHDPH(2,1) = SN0/C0/C0
C ----- NFILM FILMS
      DO I = 1, NFILM
        SF = SN0/N(I+1)
        CF = SQRT(N(I+1)**2 - SN0**2)
        IF (DIMAG(CF) .GT. 0.) THEN
          CF = -CF    ! N*CF MUST BE IN L.H.P. FOR EXPONENTIAL
        END IF
        CF = CF/N(I+1)
C
        NHAT(1, I+1) = N(I+1)*CF  ! S-POL.
        NHAT(2, I+1) = N(I+1)/CF  ! P-POL.
        BETA = BET0*D(I)*NHAT(1, I+1) ! PHASE IN RADIANS
        SB(I) = SIN(BETA)
        CB(I) = COS(BETA)
C
        DBDD(I) = BET0*NHAT(1, I+1)
        DNHDN(1, I) = 1.D0/CF
        DNHDN(2, I) = (2.D0 - 1.D0/CF**2)/CF
        DNHDPH(1, I+1) = -N(1)*SF*C0/CF
        DNHDPH(2, I+1) = N(1)*SF*C0/CF**3
        DBDN(I) = DNHDN(1, I)*BET0*D(I)
        DBDPH(I) = DNHDPH(1, I+1)*BET0*D(I)
        DBDLAM(I) = -BETA/LAM
      END DO
C ----- BOTTOM/SUBSTRATE
      SF = SN0/N(NFILM+2)
      CF = SQRT(N(NFILM+2)**2 - SN0**2)
      IF (DIMAG(CF) .GT. 0.) THEN
```

```
        CF = -CF
      END IF
      CF = CF/N(NFILM+2)
      NHAT(1, NFILM+2) = N(NFILM+2)*CF
      NHAT(2, NFILM+2) = N(NFILM+2)/CF
      DNHDN(1, NFILM+1) = 1.D0/CF
      DNHDN(2, NFILM+1) = (2.D0 - 1.D0/CF**2)/CF
      DNHDPH(1, NFILM+2) = -N(1)*SF*C0/CF
      DNHDPH(2, NFILM+2) = N(1)*SF*C0/CF**3
C
C ----- FOR EACH POLARIZATION...
      DO IP = S, P
C
C ----- INITIALIZE AT SUBSTRATE
C
        E = 1.D0
        H = NHAT(IP, NFILM+2)
        DR(IP, NSUB) = DNHDN(IP, NFILM+1)
        DR(IP, NPHI) = DNHDPH(IP, NFILM+2)
        DR(IP, NLAST) = 0.
C
        DO IFC = NFILM, 1, -1
C
C ------- PERFORM MATRIX CALCULATION
C       (USE N(IFC+1) BECAUSE N(1) = MEDIUM INDEX)
C
          NH = NHAT(IP, IFC+1)
          ENEW = CB(IFC)*E + IM*SB(IFC)/NH*H
          HNEW = NH*IM*SB(IFC)*E + CB(IFC)*H
C
C ----- DERIVATIVE TERMS
C
          ZZ1 = IM*SB(IFC)*(E*ENEW + (H/NH)*(HNEW/NH)) !OK
          ZZ2 = IM*(ENEW*ENEW*NH - HNEW*HNEW/NH)       !OK
C
C ----- WRT N(IFC+1)
C
          DR(IP, 2*IFC-1) = ZZ1*DNHDN(IP, IFC)
     1                      + ZZ2*DBDN(IFC)         !OK
C
C ----- WRT D(IFC)
C
          DR(IP, 2*IFC) = ZZ2*DBDD(IFC)           !OK
C
C ----- WRT PHI
C
          DR(IP, NPHI) = DR(IP, NPHI) + ZZ1*DNHDPH(IP, IFC+1)
     1                   + ZZ2*DBDPH(IFC)                  !OK
C
C ----- WRT LAM
C
          DR(IP, NLAST) = DR(IP, NLAST) + ZZ2*DBDLAM(IFC)
```

```
C
          E = ENEW
          H = HNEW
C
        END DO
C
C ----- AT TOP INTERFACE
C
        NH = NHAT(IP, 1)
C
        ENEW = (E - H/NH)/2.D0  ! REFLECTED FIELD
        HNEW = (E + H/NH)/2.D0  ! INCIDENT FIELD
C
        R(IP) = ENEW/HNEW
        T(IP) = 1.D0/HNEW
C
C ----- DERIVATIVE WRT PHI
C
        DR(IP, NPHI) = DR(IP, NPHI) - DNHDPH(IP, 1)/NH*E*H
C
C ----- FINISH ALL DERIVATIVES
C
        ZZ1 = -2.D0*NH*HNEW**2                          !OK
        DO J = 1, NLAST
          DR(IP, J) = DR(IP, J)/ZZ1                       !OK
        END DO
C
C ----- CHANGE PHI DERIV TO (DEG)** -1
C
        DR(IP, NPHI) = DR(IP, NPHI)*DTOR
C
      END DO
C
C ----- CHANGE SIGN OF R(P) FOR ELLIPSOMETRIC CONVENTION
C
      R(P) = -R(P)
      DO J = 1, NLAST
        DR(P, J) = -DR(P, J)
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MATRT(NFILM, LAM, PHI, N, D, R)  !, T)
C
C     CALCULATE COMPLEX REFLECTANCE AND TRANSMITTANCE  OF
C     STACK OF FILMS USING MATRIX FORMULATION.
C     FORMULAS ARE IN P.H. Berning, 'Theory and Calculations of
C        Optical Thin Films,' in 'PHYSICS OF THIN FILMS',
C        ADV. IN R&D, G. HASS (ED.), V.1, P.69 (Academic Press, 1963)
C
C     SIGN OF P-REFLECTANCE IS REVERSED TO CONFORM TO ELLIPSOMETRIC
C     CONVENTION.
C
C     INPUT:
C     NFILM = # OF FILMS
C     LAM = WAVELENGTH
C     PHI = INCIDENT ANGLE, DEGREES
C     N(I) = COMPLEX INDECES:  N(1) = MEDIUM
C     D(I) = THICKNESSES:  D(1) = TOP LAYER
C     OUTPUT:
C     R(POL) = COMPLEX REFLECTANCE OF STACK:  R(1) = S-POL.
C     T(POL) = COMPLEX  TRANS.:  T(1) = S-POL.
C
C     MODIFIED FOR ELLIPSOMETRY USE:  LINES PERTAINING TO T(POL)
C     ARE COMMENTED OUT.  (HLT, 7/28/87)
C
C     H.L. TARDY, SNLA 1824
C     3/31/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      COMPLEX*16 N(NFILM+2), NHAT
      COMPLEX*16 IM, E, H, ENEW, HNEW, R(2), T(2)
      COMPLEX*16 CF, CB, SB, BETA
      REAL*8 D(*), LAM
      INTEGER S, P
      PARAMETER(IM = (0., 1.), PI = 3.1415926D0)
C
      S = 1
      P = 2
C
      IF (PHI .EQ. 90.) THEN
        R(P) = (-1., 0.)
        R(S) = -R(P)
        T(P) = 0.
        T(S) = T(P)
        RETURN
      END IF
C
      BET0 = 2.D0*PI/LAM
      SN0 = N(1)*SIND(PHI)
      C0 = COSD(PHI)
C
C ----- FOR EACH POLARIZATION...
      DO IP = S, P
```

```
C
C ----- INITIALIZE AT SUBSTRATE
C
        CF = N(NFILM+2)**2 - SN0**2
        CF = SQRT(CF)
        IF (DIMAG(CF) .GT. 0.) THEN
          CF = -CF
        END IF
        CF = CF/N(NFILM+2)
C
        E = 1.D0
        IF (IP .EQ. 1) THEN
          H = N(NFILM+2)*CF
        ELSE
          H = N(NFILM+2)/CF
        END IF
C
        DO IFC = NFILM, 1, -1
C
C ------- PERFORM MATRIX CALCULATION
C       (USE N(IFC+1) BECAUSE N(1) = MEDIUM INDEX)
C
          CF = N(IFC+1)**2 - SN0**2
          CF = SQRT(CF)
          IF (DIMAG(CF) .GT. 0.) THEN
            CF = -CF
          END IF
          CF = CF/N(IFC+1)
C
          BETA = BET0*CF*N(IFC+1)*D(IFC) ! PHASE
          SB = SIN(BETA)*IM
          CB = COS(BETA)
C
          IF (IP .EQ. 1) THEN
            NHAT = N(IFC+1)*CF
          ELSE
            NHAT = N(IFC+1)/CF
          END IF
C
          ENEW = CB*E + SB/NHAT*H
          HNEW = NHAT*SB*E + CB*H
          E = ENEW
          H = HNEW
C
        END DO
C
        IF (IP .EQ. 1) THEN
          NHAT = N(1)*C0
        ELSE
          NHAT = N(1)/C0
        END IF
C
```

```
        ENEW = E - H/NHAT  ! REFLECTED FIELD
        HNEW = E + H/NHAT  ! INCIDENT FIELD
C
        R(IP) = ENEW/HNEW
        T(IP) = 2.DO/HNEW
C
      END DO
C
C ----- CHANGE SIGN OF R(P) FOR ELLIPSOMETRIC CONVENTION
C
      R(P) = -R(P)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MDUMP(MTX, N)
C
C     FORMATTED DUMP OF SYMMETRIC PACKED MATRIX
C     MATRIX MTX IS ASSUMED TO BE STORED SEQUENTIALLY
C     IN COLUMNS
C
      REAL*8 MTX(N*(N+1)/2)
C
      K = 0
      DO I = 1, N
        PRINT 191, (MTX(K+J), J = 1, I)
        K = K + I
 191  FORMAT(X, 6E13.5)
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MODELIN(NFLM, WL, PHI, STACK, USTACK)     !9/26/88
C
C     GETS MODEL PARAMETERS FOR NFLM-LAYER SYSTEM
C
C     CONVERTED TO STACK ARRAY STORAGE:  9/26/88
C
C     H.L. TARDY, 6224
C     7/15/88
C
C
      PARAMETER(NFMAX = 20, INU = 5)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4), NO
C
C
      CALL STACKIN(NFLM, STACK, USTACK)
C
 10   PRINT *, 'Enter incident angle, uncertainty:'
      READ (INU, *, ERR = 10) PHI, USTACK(3*NFLM+3)
C
 20   PRINT *, 'Enter wavelength:'
      READ (INU, *, ERR = 20) WL
C
      RETURN
      END
C
C
```

```
      SUBROUTINE MODLOAD(NFLM, STACK, N, D)          !10/1/88
C
C     CONVERTS ARRAY STACK INTO ARRAYS N, D
C
C     H.L. TARDY, 6224
C     10/4/88
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 STACK(3*NFLM+3), D(NFLM)
      COMPLEX*16 N(NFLM+2)
C
      N(1) = STACK(3*NFLM+3)      ! AMBIENT INDEX
C
      DO I = 1, NFLM
        II = 3*(I-1)
        N(I+1) = CMPLX(STACK(II+1), -ABS(STACK(II+2)))   ! I'TH INDEX
        D(I) = STACK(II+3)                               ! I'TH THICKNESS
      END DO
C
      N(NFLM+2) = CMPLX(STACK(3*NFLM+1),
     1                          -ABS(STACK(3*NFLM+2)))     ! SUBST. INDEX
C
      RETURN
      END
C
C
```

```
      SUBROUTINE NEWT(FCN, X, VAL, NPARAM, PARAM, NSIG,
     1                ITMAX, IER)                          !10/21/88
C
C     NEWTON'S METHOD FOR SINGLE SOLUTION OF MONOTONIC NONLINEAR
C        FUNCTION.
C     ARRAY PARAM IS USED TO PASS NPARAM PARAMETERS BETWEEN FUNCTION
C        CALLER.
C
C     H.L. TARDY, 6224
C     10/21/88
C
C     FCN          :   NAME OF FUNCTION SUBR.
C                      USAGE:  VAL = FCN(X, NPARAM, PARAM)
C                      FCN MUST BE DECLARED EXTERNAL IN MAIN
C     X            :   ARGUMENT OF FCN
C     VAL          :   VALUE OF FCN
C     NPARAM       :   NUMBER OF PARAMETERS NEEDED TO EVALUATE FCN
C     PARAM                :   ARRAY OF PARAMETERS NEEDED TO EVALUATE FCN
C     NSIG         :   NUMBER OF SIG. DIG.'S DESIRED IN SOLUTION
C     ITMAX        :   MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     IER          :   ERROR CONDITION:  0 -> NSIG SATISFIED,
C                      999 -> TOO MANY ITERATIONS
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      EXTERNAL FCN
      REAL*8 PARAM(NPARAM)
      LOGICAL DONE
C
      ALPHA = 0.01
      IEND = 0
      DONE = .FALSE.
      GOOD = 10.**(-NSIG)
C
      VAL = FCN(X, NPARAM, PARAM)
      IF (X .EQ. 0.) THEN
        DXNU = 0.001
      ELSE
        DXNU = ALPHA*X
      END IF
      X = X+DXNU
C
      DO WHILE (.NOT. DONE)
        IEND = IEND + 1
C ----- ESTIMATE NEW X
        VALNU = FCN(X, NPARAM, PARAM)
        SLOPE = (VALNU-VAL)/DXNU
        DXNU = -VALNU/SLOPE
D       PRINT *
D       PRINT *, 'X =', X, '  VALNU =', VALNU
D       PRINT *, 'SLOPE =', SLOPE, '  DXNU =', DXNU
        TEST = ABS(DXNU/X)
```

```
            X = X + DXNU
            VAL = VALNU
C ----- TEST
            IF (TEST .LT. GOOD) THEN
              IER = 0
              ITMAX = IEND
              DONE = .TRUE.
              VAL = FCN(X, NPARAM, PARAM)
            ELSE IF (IEND .GE. ITMAX) THEN
              IER = 999
              DONE = .TRUE.
            END IF
C
          END DO
C
          RETURN
          END
C
C
```

```
      SUBROUTINE PARCHG(NL, STACK, USTACK)          !10/24/88
C
C     CHANGE ONE PARAMETER FROM STACK
C
C     H.L. TARDY, 6224
C     10/21/88
C
      REAL*8 STACK(3*NL+3), USTACK(3*NL+4)
      INTEGER ICHG(1)
      CHARACTER*5 STR(1)
      PARAMETER (INU = 5)
C
      CALL GET_IS(NL, 1, ICHG)
      CALL STRING(NL, 1, ICHG, STR)
C
 10   PRINT 20, STR(1), STR(1)
 20   FORMAT('$Enter new ', A5, ', d', A5, ': ')
      READ (INU, *, ERR = 10) STACK(ICHG(1)), USTACK(ICHG(1))
C
      RETURN
      END
C
C
```

```
      SUBROUTINE REARR(AIN, N, IPVT, AOUT)          !11/2/88
C
C     REARRANGES UPPER-TRIANGULAR PACKED MATRIX AIN
C     ACCORDING TO IPVT.  AOUT = TRANS(P) . AIN . P,
C     WHERE P IS A PERMUTATION MATRIX CONSTRUCTED FROM
C     THE PIVOT VECTOR IPVT.
C     FOR USE WITH SLATEC MATRIX INVERSION ROUTINES.
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 AIN(N*(N+1)/2), AOUT(N*(N+1)/2)
      INTEGER IPVT(N)
C
C ----- REARRANGE AIN TO OBTAIN AOUT (SYMMETRIC STORAGE)
C     STORE IN AOUT
C
C     PRINT *
      K = 0
      DO J = 1, N
        JP = IPVT(J)
        DO I = 1, J
          K = K + 1
          IP = IPVT(I)
          IF (IP .LE. JP) THEN
            KP = JP*(JP-1)/2 + IP
          ELSE
            KP = IP*(IP-1)/2 + JP
          END IF
          AOUT(K) = AIN(KP)
        END DO
      END DO
C
D     PRINT 180, IPVT
D180  FORMAT(/, ' IPVT = ', 10I3)
D     PRINT *, 'Pivoted matrix:'
D     CALL MDUMP(AOUT, N)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE SMODELIN(NFLM, WL, PHI, STACK)    !9/26/88
C
C     GETS MODEL PARAMETERS FOR NFLM-LAYER SYSTEM
C
C     CONVERTED TO STACK ARRAY STORAGE:  9/26/88
C
C     H.L. TARDY, 6224
C     7/15/88
C
C
      PARAMETER(NFMAX = 20, INU = 5)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 STACK(3*NFMAX+3), NO
C
C
      CALL SSTACKIN(NFLM, STACK)
C
 10   PRINT *, 'Enter incident angle:'
      READ (INU, *, ERR = 10) PHI
C
 20   PRINT *, 'Enter wavelength:'
      READ (INU, *, ERR = 20) WL
C
      RETURN
      END
C
C
```

```
      SUBROUTINE SSTACKIN(NFLM, STACK)    !9/26/88
C
C     GETS MODEL PARAMETERS FOR NFLM-LAYER SYSTEM
C
C     H.L. TARDY, 6224
C     9/26/88
C
C
      PARAMETER(NFMAX = 20, INU = 5)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 STACK(3*NFMAX+3)
C
 10   PRINT *, 'Enter number of films:'
      READ (INU, *, ERR = 10) NFLM
C
 20   PRINT *, 'Enter ambient index:'
      READ (INU, *, ERR = 20) STACK(3*NFLM+3)
C
      PRINT *, 'Enter properties from top film down:'
C
      DO I = 1, NFLM
 50     PRINT 40, I
        II = 3*(I-1)
        READ (INU, *, ERR = 50) (STACK(II+J), J = 1, 3)
      END DO
 40   FORMAT(' Enter n, k, d for film #', i3)
C
 60   PRINT *, 'Enter substrate n, k:'
      READ (INU, *, ERR = 60) (STACK(3*NFLM+J), J = 1, 2)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE STACKIN(NFLM, STACK, USTACK)     !9/26/88
C
C     GETS MODEL PARAMETERS FOR NFLM-LAYER SYSTEM
C
C     H.L. TARDY, 6224
C     9/26/88
C
C
      PARAMETER(NFMAX = 20, INU = 5)
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 STACK(3*NFMAX+3), USTACK(3*NFMAX+4)
C
 100  PRINT *, 'Enter number of films:'
      READ (INU, *, ERR = 100) NFLM
C
 110  PRINT *, 'Enter ambient index:'
      READ (INU, *, ERR = 110) STACK(3*NFLM+3)
C
      PRINT *, 'Enter properties from top film down:'
C
      DO I = 1, NFLM
 120    PRINT 10, I
        II = 3*(I-1)
        READ (INU, *, ERR = 120) (STACK(II+J),
     1                            USTACK(II+J), J = 1, 3)
      END DO
 10   FORMAT(' Enter n, dn, k, dk, d, dd for film #', i3)
C
 130  PRINT *, 'Enter substrate n, dn, k, dk:'
      READ (INU, *, ERR = 130) (STACK(3*NFLM+J),
     1                          USTACK(3*NFLM+J), J = 1, 2)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE STDUMP(NFLM, STACK, USTACK, NP, IS, UNIT)   !9/22/88
C
C
C FUNCTIONAL DESCRIPTION:
C
C    OUTPUT OF OPTICAL STACK MODEL W/UNCERTAINTIES, UNKNOWNS LABELLED.
C
C DUMMY ARGUMENTS:
C
C    NFLM    = # FILMS
C    STACK(I) = MODEL PARAMETERS:
C     I = 3*J-2; STACK(I) = N(J).  I = 3*J-1; STACK(I) = K(J).
C     I = 3*J; STACK(I) = D(J).  I = 3*NFLM+1; STACK(I) = N(SUBST).
C     I = 3*NFLM+2; STACK(I) = K(SUBST).  I = 3*NFLM+3; STACK(I) = NO
C
C    USTACK(I) = MODEL UNCERTAINTIES.  SAME CONVENTION, EXCEPT NO NOT
C     REPRESENTED
C    NP = # UNKNOWNS
C    IS(I) = INDEX OF I'TH UNKNOWN
C    UNIT  = OUTPUT LOGICAL UNIT
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20)
      REAL*8 STACK(3*NFLM+3), USTACK(3*NFLM+4)
      INTEGER IS(NP), UNIT
      CHARACTER*1 MARK(3*NFMAX+3)
C
      DO I = 1, 3*NFLM+2
        MARK(I) = ' '
      END DO
      DO I = 1, NP
        MARK(IS(I)) = '*'
      END DO
C
      WRITE (UNIT, 10)
 10   FORMAT(/, ' Stack model:  * - Unknown, starting value', /,
     1         ' Layer', 11x, 'n', 21x, 'k', 22x, 'd')
      DO I = 1, NFLM
        II = 3*I-3
        WRITE(UNIT, 100, IOSTAT = IOS) I, (MARK(II+J), STACK(II+J),
     1                      USTACK(II+J), J = 1, 3)
      END DO
      WRITE (UNIT, 110, IOSTAT = IOS) (MARK(J), STACK(J), USTACK(J),
     1                     J = 3*NFLM+1, 3*NFLM+2)
C
 100  FORMAT(2X, I2, A4, F7.4, ' +/-', F7.4,
     1         A4, F7.4, ' +/-', F7.4,
     2         A4, F8.2, ' +/-', F6.2)
 110  FORMAT(3X, 'S', A4, F7.4, ' +/-', F7.4,
     1         A4, F7.4, ' +/-', F7.4)
C
```

```
C
      RETURN
      END
C
C
```

```
      SUBROUTINE STRING(NL, NP, IS, STR)                !9/26/88
C
C     CONVERTS SOLUTION-INDEX ARRAY IS(I) INTO STRING VECTOR
C
C     H.L. TARDY, 6224
C     9/26/88
C
C
      INTEGER IS(NP)
      CHARACTER*5 STR(NP)
C
      DO I = 1, NP
        STR(I)(2:5) = '(  )'
        IFLM = (IS(I)+2)/3
        IF (IFLM .EQ. NL+1) THEN
          STR(I)(4:4) = 'S'
        ELSE
          IF (IFLM/10 .GT. 0) THEN
            STR(I)(3:3) = CHAR(IFLM/10 + 48)
          END IF
          STR(I)(4:4) = CHAR(MOD(IFLM,10) + 48)
        END IF
C
        IND = MOD(IS(I), 3)
        IF (IND .EQ. 1) THEN
          STR(I)(1:1) = 'n'
        ELSE IF (IND .EQ. 2) THEN
          STR(I)(1:1) = 'k'
        ELSE
          STR(I)(1:1) = 'd'
        END IF
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE SUBLAYER(NL, STACK, USTACK)        !10/10/88
C
C     REMOVE ONE LAYER FROM OPTICAL STACK
C
C     H.L. TARDY, 6224
C     10/10/88
C
C
      PARAMETER(NFMAX = 20, INU = 5)
      REAL*8 STACK(3*NL+3), USTACK(3*NL+4)
C
 130  PRINT 132
 132  FORMAT('$Remove which layer? ')
      READ (INU, *, ERR = 130) IL
      IF (IL .GT. NL) THEN
        PRINT *, 'SUBLAYER:  Invalid layer index.'
        GOTO 130
      END IF
      II = 3*IL-2
      DO I = II, 3*NL
        STACK(I) = STACK(I+3)
      END DO
      DO I = II, 3*NL+1
        USTACK(I) = USTACK(I+3)
      END DO
C
      NL = NL - 1
C
      RETURN
      END
C
C
```

```
      SUBROUTINE SUBSTRATE(PHI, EANG, NS)
C
C     H.L. TARDY, 1824
C     8/7/87
C
C     CALCULATES BARE-SUBSTRATE NS FROM PHI, PSI, AND DEL
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 NO, EANG(2)
      REAL*8 NS(2)
      COMPLEX*16 RHO, IM
      PARAMETER(NO = 1.000, IM = (0., 1.), PI = 3.141592654D0)
C
      PSI = EANG(1)
      DEL = EANG(2)
C
      RHO = TAND(PSI)*EXP(IM*DEL*PI/180.DO)
      SO = SIND(PHI)
      TO = TAND(PHI)
      RHO = 1 - 4. * RHO * (SO/(1+RHO))**2
      RHO = SQRT(RHO)
      NS(1) = DREAL(RHO) * NO * TO
      NS(2) = DIMAG(RHO) * NO * TO
      NS(2) = ABS(NS(2))
C
      RETURN
      END
C
C
```

```
      SUBROUTINE THICKN(WL, PHI, NF, XSOL, D, DPHI)      !10/21/88
C
C     H.L. TARDY, 1824
C     10/21/88
C
C
C     CALCULATES FILM THICKNESS, PHASE THICKNESS FROM
C     FILM INDEX AND INCIDENT ANGLE.  ELLIPSOMETRIC DATA
C     IS OBTAINED THROUGH REFERENCE TO  CHARLOT-MARUANI
C     FUNCTION CMNEWT.
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 NF, N0
      COMPLEX*16 XSOL, CJ, D, DPHI, IM
      PARAMETER (N0 = 1.000, PI = 3.141592654D0,
     1             IM = (0., 1.))
C
C ----- CALCULATE THICKNESS
C
      S0 = SIND(PHI)
      CJ = 1. - (N0*S0/NF)**2
      CJ = SQRT(CJ)
      IF (DIMAG(CJ) .GT. 0.) THEN
        CJ = -CJ
      END IF
      DPHI = WL/(2.*CJ*NF)          !  PHASE THICKNESS
      D = IM*DPHI*LOG(XSOL)/2./PI
C
C ----- ADJUST D TO SMALLEST VALUE .GE. 0.
C
      IF (REAL(DPHI) .GT. 0.) THEN
        DO WHILE (REAL(D) .GE. REAL(DPHI))
          D = D - DPHI
        END DO
        DO WHILE (REAL(D) .LT. 0.)
          D = D + DPHI
        END DO
      END IF
C
      RETURN
      END
C
C
```

```
      SUBROUTINE TWOZONE(P, A, C, DP, DA, EANG, DEA,
     1                   T2P, ETAPC, DT2P, DETA)
C
C     H.L. TARDY, 1824
C     MODIFIED FROM MAILM.FOR VERSION, 7/17/87
C
C     CONVERTS TWO POLARIZER, ANALYZER AZIMUTHS TO PSI AND
C     DELTA.
C
C     CORRECTS FOR POLARIZER DEFECTS, AZIMUTH ERRORS WITH T2P
C     AND ETAPC.  IF NO FOUR-ZONE DATA HAS BEEN TAKEN, T2P =
C     ETAPC = 0.
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER(NDMAX = 20)
      REAL*8 P(1), A(1), C(1), PZ(2), AZ(2)
      REAL*8 EANG(2), DEA(2)
      COMPLEX*16 RHO, IM
      PARAMETER( IM = (0., 1.), PI = 3.141592654)
C
C ----- PUT C, A, P IN (-90., 90.)
C
      COMPZ = C(1)
      DO WHILE (COMPZ .GT. 90.)
        COMPZ = COMPZ - 180.
      END DO
      IF (ABS(COMPZ) .NE. 45.) THEN
        PRINT *, 'TWOZONE:  Invalid compensator setting.'
        RETURN
      END IF
      DO I = 1, 2
        PZ(I) = P(I)
        AZ(I) = A(I)
C
C ----- PUT A IN (-90, 90)
C
        DO WHILE (AZ(I) .GT. 90.)
          AZ(I) = AZ(I) - 180.
        END DO
        DO WHILE (AZ(I) .LT. -90.)
          AZ(I) = AZ(I) + 180.
        END DO
C
C ----- SET PZ(I) = DEL, DEPENDS ON ZONE
C
        IF (COMPZ .LT. 0.) THEN     ! ODD ZONES
          IF (AZ(I) .GT. 0.) THEN   ! ZONE 1
            PZ(I) = 2.*(PZ(I) + 45.)
          ELSE                      ! ZONE 3
            PZ(I) = 2.*(PZ(I) - 45.)
          END IF
        ELSE                        ! EVEN ZONES
```

```
          IF (AZ(I) .GT. 0.) THEN  ! ZONE 2
            PZ(I) = -2.*(PZ(I) + 45.)
          ELSE                     ! ZONE 4
            PZ(I) = -2.*(PZ(I) - 45.)
          END IF
        END IF
C
C ----- PUT PZ(I) IN (-180, 180)
C
        DO WHILE (PZ(I) .LT. -180.)
          PZ(I) = PZ(I) + 360.
        END DO
        DO WHILE (PZ(I) .GT. 180.)
          PZ(I) = PZ(I) - 360.
        END DO
C
      END DO
C
C ----- PERFORM ZONE AVERAGE
C
      EANG(1) = ABS(AZ(1) - AZ(2))/2.
      EANG(2) = (PZ(1) + PZ(2))/2.
C
C ----- CORRECT FOR DEFECTS, AZIMUTH ERRORS
C
      XP = T2P*SIND(2.*PSI)
      IF (COMPZ .LT. 0.) THEN    ! ODD ZONE DATA
        EANG(1) = EANG(1) - XP
        EANG(2) = EANG(2) + ETAPC
      ELSE                       ! EVEN ZONE DATA
        EANG(1) = EANG(1) + XP
        EANG(2) = EANG(2) - ETAPC
      END IF
C
C ----- CALCULATE UNCERTAINTIES:  SYSTEMATIC ERROR NOT INCLUDED IF
C     T2P, ETAPC = 0.
C
      DEA(1) = SQRT(DA**2/2. + (DT2P*SIND(2.*EANG(1)))**2)
      DEA(2) = SQRT(2.*DP**2 + DETA**2)
C
      RETURN
      END
C
C
```

```
      SUBROUTINE UNIERR(NL, WL, PHI, STACK, USTACK, UEA,
     1                  IS, RANDOM, DELT)                !11/10/88
C
C     COMPUTES PARAMETER UNCERTAINTIES OF GENERAL NL-LAYER SYSTEM
C     SOLVED FOR TWO SYSTEM PARAMETERS
C
C     H.L. TARDY, SNLA 6224
C     7/19/88
C
C
C     INPUT:
C
C        NL       = # OF FILMS (.GE. 0)
C        WL       = VAC. WAVELENGTH
C        PHI      = INCIDENT ANGLE (DEG.)
C        STACK(I)= STACK MODEL PARAMETERS
C        USTACK(I) = STACK UNCERTAINTIES
C        UEA(I)   = UNC. IN (PSI, DEL) (ASSUMED RANDOM)
C        IS(I)    = INDEX OF SOLUTION PARAMETERS (I = 1, 2,
C                   IS(I) .LE. 3*NL+2)
C        RANDOM(I) = LOGICAL TYPE OF I'TH UNCERTAINTY
C                   (.TRUE. => RANDOM; USUALLY RANDOM = .FALSE.
C                     EXCEPT FOR PHI AND WL UNCERTAINTIES)
C
C     OUTPUT:
C
C        DELT(I) = RESULT UNCERTAINTY IN SOLUTION I
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (NFMAX = 20)
      REAL*8 STACK(3*NL+3), USTACK(3*NL+4), DELT(2), UEA(2)
      REAL*8 EANG(2), DEA(2, 3*NFMAX+4)
      REAL*8 JACO(2, 2), DFIX(2, 3*NFMAX+4)
      INTEGER IS(2)
      LOGICAL RANDOM(3*NL+4)
C
      NLAST = 3*NL + 4
C
C ----- GET PSI, DEL DERIV'S WRT PARAMETERS
C
      CALL ANGSD(NL, WL, PHI, STACK, EANG, DEA)
C
C ----- GET INVERSE JACOBIAN, DERIVS OF SOLN WRT FIXED PARAMS
C
      CALL DVSOL(NL, DEA, IS, JACO, DFIX)
C
C ----- COMPUTE TOTAL RANDOM UNCERTAINTY
C
      DO K = 1, 2
        DELT(K) = 0.
        DO I = 1, 2
```

```
          DELT(K) = DELT(K) + (JACO(K, I)*UEA(I))**2
        END DO
      END DO
C
      DO K = 1, 2
        DO I = 1, NLAST
          IF ((I .NE. IS(1)) .AND. (I .NE. IS(2))) THEN
            IF (RANDOM(I)) THEN
              DELT(K) = DELT(K) + (DFIX(K, I) * USTACK(I))**2
            END IF
          END IF
        END DO
      END DO
C
      DO K = 1, 2
        DELT(K) = SQRT(DELT(K))
      END DO
C
C ----- ADD SYSTEMATIC UNCERTAINTY
C
      DO K = 1, 2
        DO I = 1, NLAST
          IF ((I .NE. IS(1)) .AND. (I .NE. IS(2))) THEN
            IF (.NOT. RANDOM(I)) THEN
              DELT(K) = DELT(K) + ABS(DFIX(K, I) * USTACK(I))
            END IF
          END IF
        END DO
      END DO
C
      RETURN
      END
C
C
```

```
      SUBROUTINE XTXUP(MTX, NPAR, XTX)            !11/2/88
C
C     CALCULATES THE PRODUCT TRANS(MTX)*MTX FOR AN
C     UPPER-TRIANGULAR MATRIX MTX.  RESULT IS SYMMETRIC
C     MATRIX XTX (UPPER-TRIANGULAR PART).  SYMMETRIC-MODE STORAGE,
C     WITH MTX AND XTX STORED SEQUENTIALLY BY COLUMNS.
C
C
      IMPLICIT REAL*8 (A-H, O-Z)
      REAL*8 MTX(NPAR*(NPAR+1)/2), XTX(NPAR*(NPAR+1)/2)
C
      IND = 0
      DO J = 1, NPAR
        K = 0
        DO I = 1, J
          IND = IND + 1
          XTX(IND) = 0.DO
          DO II = IND - I + 1, IND
            K = K + 1
            XTX(IND) = XTX(IND) + MTX(II)*MTX(K)
          END DO
        END DO
      END DO
C
      RETURN
      END
C
C
```

(this page intentionally left blank)

Distribution:

National Energy Software Center
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439 (15)

1151	A.J. Hurd
1512	R.D. Skocypec
1823	D.M. Haaland
1841	T.M. Christensen
1846	C.S. Ashley
1846	S.L. Martinez
2146	S.C. Everist
2146	S.L. Miller
2146	W.M. Miller
6224	D.G. Schueler, actg.
6224	H.L. Tardy (10)
6224	A.R. Mahoney
7241	P.D. Thacher
7471	S.T. Reed
3141	S.A. Landenberger (5)
3151	W.I. Klein
3154-1	C.L. Ward For DOE/OSTI (Unlimited Release)
8524	J.R. Wackerly