A MICROCOMPUTER PROGRAM FOR OPTICAL MULTILAYER THIN FILMS

by

KEVIN HOWARD BETTS

B.A.Sc., The University of British Columbia, 1983

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE STUDIES
(Department of Physics)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

APRIL 1985

© Kevin Howard Betts, 1985
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Physics

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date April 22, 1985
ABSTRACT

A microcomputer software package was written to calculate the transmission and reflection characteristics of multilayer thin films on a substrate. The program was written to be as "user-friendly", versatile and modular as possible. To test the program, a transparent oxide-type heat-mirror film capped with an antireflection coating was studied for greenhouse applications. SiO$_2$ and ZnO were considered as representative antireflection and heat-mirror materials respectively. The results of calculations of heat transfer coefficient and transmittance of the glazing construction polyethylene/air gap/SiO$_2$/ZnO/polyethylene are presented. The resulting structure is shown to give a visible light transmission spectrum which closely matches the plant sensitivity curve for plant growth and has higher thermal insulation compared to uncoated polyethylene.
TABLE OF CONTENTS

ABSTRACT ................................................................. ii
LIST OF TABLES .......................................................... v
LIST OF FIGURES .......................................................... vi
ACKNOWLEDGMENT .......................................................... vii
INTRODUCTION ............................................................. 1

CHAPTER TWO

2.1 Theoretical Background: Single Layer Films .......... 6
   Transmission and Reflection Coefficients .......... 6
   Generalizing the Snell Law ......................... 8
   Extinction Coefficient .............................. 9

2.2 Theoretical Background: Multilayer Thin Films .... 10
   Transmission and Reflection Coefficients .... 10
   Maximizing the T Coefficient .................... 15

CHAPTER THREE

3.1 Features of Hetero .............................................. 17
3.2 Executing Hetero ................................................ 21

CHAPTER FOUR

4.1 Hetero Compilation Units ................................. 23

CHAPTER FIVE

5.1 Background ........................................................ 29
5.2 Design Criteria for Multilayer Structure ........ 30
5.3 Choice of Materials ........................................... 32
5.4 Design of Heat Mirror Heterostructure ........ 33
5.5 Performance of Glazing Heterostructure .......... 35

CONCLUSION ............................................................... 44
REFERENCES .................................................. 45

APPENDIX A

Optical Constants for ZnO and Polyethylene .............. 47

APPENDIX B

Listing of Program Hetero .............................. 49
LIST OF TABLES

TABLE I
Optical Constants ........................................41

TABLE II
Cost Effectiveness ........................................42
LIST OF FIGURES

FIGURE 1
Structure of Hetero Compilation Units .......................4

FIGURE 2
Transmission/Reflection Coefficients for a Single Film .7

FIGURE 3
Transmission and Reflection Coefficients for Thin Films on One Side of Substrate .........................11

FIGURE 4
Transmission and Reflection Coefficients for Thin Films on Both Sides of Substrate .....................14

FIGURE 5
Transmission Through Polyethylene/air gap/SiO$_2$/ZnO/Polyethylene ..................31

FIGURE 6
Refractive Index Data for SiO$_2$, ZnO, Polyethylene ......34

FIGURE 7
Reflection of IR Radiation from ZnO/Polyethylene ........36

FIGURE 8
Cross Section of AR/Heat-mirror Film for Greenhouses ...37

FIGURE 9
Angular T Variation for Parallel Polarization ..........39

FIGURE 10
Angular T Variation for Perpendicular Polarization ....40
ACKNOWLEDGMENT

I would like to thank Dr. Robert Parsons, my thesis supervisor, for his continued support, constructive criticism, and assistance.

Thanks is also due to a number of fellow students. In particular, Mike Brett who was always willing to give assistance when asked, John Affinito for his thorough testing of all aspects of the program and for his constructive criticism on ways to make Hetero "user-friendly", and, Brian Sullivan for his invaluable help when I was learning Pascal.
INTRODUCTION

Optical multilayer thin film theory for antireflecting and reflecting systems was first formulated around 1937 [1-2]. Thin film applications in optics started around the same time with the introduction of the vacuum evaporation technique for depositing thin films [2-3]. Since then many thin film deposition technologies have been used in depositing multilayer thin films on a substrate (e.g. planar magnetron sputtering) [3]. With multilayer thin films being successfully used in transparent window insulation, solar-collector, electro-optical, industrial heat shielding and aircraft applications [3].

In 1978 a research group in the Physics Department at U.B.C. began to study the optical properties of multilayer thin films that were deposited by a planar magnetron sputtering system. The optical properties that were of main interest were the transmission and reflection (T/R) properties of multilayer thin films on a plastic substrate. Unfortunately, analytical work in studying even two or three film layers can become very labourious. As a result, it was decided to use a microcomputer program to compute the T/R curves. Since there were no precedents in the literature of a program which calculated the T/R curves of any number of thin film layers, it was necessary to write our own program. Consequently, in 1982 a simple computer program was written which could calculate the T/R curves of one, two or three thin film layers; but the theory used did not readily allow for easy expansion of this program.
Also, this program was not "user-friendly"; in other words, the user had to enter the program to make any changes (this required that the user know how the program worked).

In May of 1983 work began on a "user-friendly" program, called Hetero, which would handle m thin film layers on either or both sides of the substrate using the general multilayer thin film theory of reference 4. A decision was made to limit the program to calculate the T/R values for up to twenty thin film layers because: (i) more than twenty layers required too much memory and (ii) it was felt there would be no immediate need for more than twenty thin films on a substrate. It was also decided that the program should be written to allow for easy future expansion of the maximum number of thin films; this will be discussed in a later section.

By the time work had begun on Hetero the lab had already written a microcomputer software package to control and monitor the planar magnetron sputtering process and to display and record data from the process equipment [5]. The system called Microsput, was written in Pascal and operated on the UCSD Pascal Version IV.0 operating system. Consequently, Hetero was also written in Pascal, using the same operating system, to make the two software packages complimentary. A secondary reason for using Pascal over other programming languages was based on the fact that Pascal is a structured language which can handle data and number crunching as well as character and string handling efficiently.

The explanation and documentation of Hetero is the primary
topic of this thesis. To test the program the problem of calculating the T/R curves of a multilayer thin films structure on greenhouse polyethylene was considered. Restated, the main objective of this thesis was to write a "user-friendly" microcomputer program for studying multilayer thin films T/R characteristics as a function of wavelength, where the user interactively enters a minimal amount of important heterostructure data while still maintaining maximum variance in the heterostructure composition. It was also desired that this program be written in pure pascal code which allowed it to used on any operating system that supported UCSD Pascal; this allows for easy adaptability of Hetero to new microcomputer systems. Finally, as a program to be used in research one must be able to extend and alter the software easily as research necessitates the addition of, say, new theories for calculating the refractive indices of thin films (e.g. sermites). Hetero achieves this adaptability by providing seperately compiled unit routines with each unit preforming a very specific task (see Figure 1).

To verify that this program would work as described above a secondary research effort was done in this report on improving the T/R characteristics of polyethylene on greenhouses. In particular, the T/R characteristics of the glazing construction polyethylene/air gap/SiO$_2$/ZnO/polyethylene were calculated. This result was related to greenhouse applications.

The structure of this report is as follows. Chapter Two
Figure 1. The structure of Hetero compilation units, with an indication of how future compilation units can be added. Units at top of the figure have access to those below. For simplicity all global variable declarations are in unit ENTER DATA.
describes the theory of single layer and multilayer thin films as how they are applied to the Hetero. Chapter Three is an overview of the features of the program and how to use it. Chapter Four is an overview of the program; an explanation of each compilation unit. Chapter Five discusses the application of this program for improving the T/R characteristics of polyethylene on greenhouses. Chapter Six concludes the thesis. Appendices include the determination of optical constants for ZnO and polyethylene; and the entire listing of Hetero.
CHAPTER TWO

2.1 THEORETICAL BACKGROUND: SINGLE LAYER FILMS

TRANSMISSION AND REFLECTION COEFFICIENTS

The following theory has applications in calculation of (i) the total transmission and reflection coefficients \((T_s/R_s)\) of a single parallel plate and (ii) the absorption \(a\) and extinction \(k\) coefficients (defined below) of this plate. Both calculations are now proven and shall be used in a later section.

Consider a parallel transparent plate of refractive index \(n\), surrounded by a medium of refractive index \(n_0\). Now suppose a plane wave of light is incident upon the medium at an angle \(\theta\), represented by ray SB₁. At the first surface this wave is divided into two plane waves, one reflected in the direction B₁C₁ and the other transmitted into the plate in the direction B₁D₁. The latter wave is incident on the second surface at an angle \(\theta\), and is divided there into two plane waves, one reflected into the plate in the direction D₁B₂ and one transmitted in the direction D₁E₁. This process of division of the wave remaining inside the plate continues as indicated in Figure 2. Let \(I_0\) be the intensity of the incoming wave. By summing all the rays that are transmitted through the plate (i.e. rays D₁E₁, for \(i,j = 1,2,\ldots\)) the transmitted intensity is found

\[ I = I_0(1-r)^2\exp(-at/cos\theta_1) + I_0(1-r)^2r^2\exp(-3at/cos\theta_1) + \ldots \tag{1} \]

Simplifying
Figure 2. Cross section of a single film. Transmission and reflection values solved from an energy viewpoint. 
\[ d = \frac{t}{\cos \theta} \]
\[ I = T_s = \frac{(1-r)^2 \exp(-at/\cos \theta_1)}{1 - r^2 \exp(-2at/\cos \theta_1)} \]  

similarly for the total intensity reflected

\[ I = I_0 r + I_0 r(1-r)^2 \exp(-2at/\cos \theta_1) + I_0 r^3(1-r)^2 \exp(-4at/\cos \theta_1) + \]  

simplifying

\[ I = R_s = \frac{r(1 + (1-2r) \exp(-2at/\cos \theta_1))}{1 - r^2 \exp(-2at/\cos \theta_1)} \]  

where  
\[ r = \text{Fresnel coefficient for reflection at a single surface.} \]  
\[ t = \text{thickness of substrate} = d \cos \theta_1 \]  
\[ a = \text{absorption coefficient (defined below)} \]  
\[ \theta_1 = \text{complex angle of refraction; defined below.} \]

The determination of the complex \( \cos(\theta_1) \) is now discussed briefly; then the determination of \( a \) is shown.

**GENERALIZING THE SNELL LAW**

The refractive index, of the plate, is in general complex as defined by

\[ \hat{n}_1 = n_1 - ik_1 \]  

where \( k_1 \) is the extinction coefficient of the layer. It is related to the absorption coefficient \( a \) by the relation

\[ k_1 = \frac{\alpha \lambda}{4\pi} \]  

The angle \( \theta_1 \), is related to the angle of incidence of \( \theta \) by Snell's law.

\[ n_0 \sin(\theta) = \hat{n}_1 \sin(\theta_1) \]  

Since \( \hat{n}_1 \) is complex, inevitably one must assume a complex \( \theta_1 = \hat{\theta}_1 \), and correspondingly a complex \( \sin(\hat{\theta}_1) \). Solving for a complex \( \sin(\hat{\theta}_1) \) and \( \cos(\hat{\theta}_1) \) following reference 2; let us write

\[ \sin(\hat{\theta}_1) = S' + iS'' \]  

we find from Snell's law
\[
S' = \frac{n_0 \sin(\theta) \cdot n_1}{n_1^2 + k_1^2}
\]
\[
S'' = \frac{n_0 \sin(\theta) \cdot k_1}{n_1^2 + k_1^2}
\]

Now solving for the complex cosine
\[
\cos(\theta_1) = C' + iC'' = \sqrt{1 - \sin^2(\theta_1)}
\]

The components of this complex cosine are found to be
\[
C' = \sqrt{\frac{p + \sqrt{p^2 + q^2}}{2}}
\]
\[
C'' = \sqrt{\frac{-p + \sqrt{p^2 + q^2}}{2}}
\]

where
\[
p = 1 - S'^2 + S''^2 = C'^2 - C''^2
\]
\[
q = -2S'S'' = 2C'C''
\]

EXTINCTION COEFFICIENT

From equations (1)-(6) it is now possible to calculate the absorption coefficient. Since \( R > r \) it is a good approximation in general to take \( r << 1 \); thus from equation (2) and (4) we get an expression for the Fresnel coefficient
\[
r = \frac{R_s}{1 + T_s^2}
\]

Solving equation 2 for \( a \) gives
\[
a = -\ln((-1-r)^2 + \sqrt{(1-r)^4 + 4T_s^2r^2}) + \ln(2T_s r^2)
\]

Determining \( R_s \) and \( T_s \) experimentally, say, with a spectrophotometer and then by using equations (6), (16) and (17) the extinction coefficient of a single transparent parallel plate can be found.
2.2 THEORETICAL BACKGROUND: OPTICAL MULTILAYER THIN FILMS

TOTAL TRANSMISSION AND REFLECTION COEFFICIENTS

If a beam of electromagnetic radiation falls onto a structure consisting of thin films of several different materials, multiple reflections take place and interference occurs [4]. In order to calculate the reflection and transmission coefficient at any angle of incidence (and any number of thin film layers) it is convenient to use a matrix formulation [4,6]. Consider a thin film system consisting of 1 layers as shown in Figure 3. The construction parameters comprise not only the refractive index \( n_i \) and thickness \( t_i \) of the layers but also the refractive indices \( n_s \) and \( n_0 \) of the substrate and incident medium, respectively. The wavelength \( \lambda \), the angle of incidence \( \theta \), and the plane of polarization of the incident radiation are external variables of the system. It can be shown [4,6] that the jth film can be represented by the so called "characteristic matrix".

\[
M = \begin{pmatrix}
\cos(\delta_i) & \frac{i\sin(\delta_i)}{U_i} \\
iU_i\sin(\delta_i) & \cos(\delta_i)
\end{pmatrix}
\]  

\[
\delta_i = (2\pi/\lambda)(n_i t_i \cos(\phi_i))
\]

\[
U_i = \begin{cases}
n_i & \text{polarization parallel to plane of incidence} \\
\cos(\phi_i) & \text{polarization perpendicular to plane of incidence}
\end{cases}
\]

The quantity \( \delta_i \) is called the "effective phase difference" of
Figure 3. Construction parameters for the calculation of transmission and reflection characteristics for multilayer thin films on one side of the substrate.
the jth layer for an angle of refraction $\phi_i$. The refractive index is in general complex as defined by

$$\hat{n} = n - ik$$  \hspace{1cm} (21)$$

where $k$ is the extinction coefficient of the layer. It is related to the absorption coefficient $a$ defined in equation 6.

$$k = \frac{a\lambda}{4\pi}$$  \hspace{1cm} (22)$$

The angle $\phi_i$ is related to the angle of incidence $\theta$ by Snell's law.

$$n_0 \sin(\theta) = \hat{n}_i \sin(\phi_i).$$  \hspace{1cm} (23)$$

The complete multilayer structure of Figure 3 is represented by the product of the individual characteristic matrices,

$$M = M_1M_2M_3 \ldots M_{l-1}M_l$$  \hspace{1cm} (24)$$

which can be expressed as

$$M = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix}$$  \hspace{1cm} (25)$$

The electric field amplitude transmittance and reflectance coefficients are given by

$$r_{a,s} = \frac{n_0 \cos(\theta)(m_{11} + m_{12} \hat{n}_s \cos(\phi_s)) - m_{21} - m_{22} \hat{n}_s \cos(\phi_s)}{n_0 \cos(\theta)(m_{11} + m_{12} \hat{n}_s \cos(\phi_s)) + m_{21} + m_{22} \hat{n}_s \cos(\phi_s)}$$  \hspace{1cm} (26)$$

$$t_{a,s} = \frac{2n_0 \cos(\theta)}{n_0 \cos(\theta)(m_{11} + m_{12} \hat{n}_s \cos(\phi_s)) + m_{21} + m_{22} \hat{n}_s \cos(\phi_s)}$$  \hspace{1cm} (27)$$

The transmission and reflection coefficients, for energy, are

$$T_{a,s} = \frac{\text{Re}(\hat{n}_s \cos(\phi_s) t_{a,s})}{\text{Re}(n_0 \cos(\theta))}$$  \hspace{1cm} (28)$$

$$R_{a,s} = |r_{a,s}|^2$$  \hspace{1cm} (29)$$

where $\text{Re}(x)$ denotes the real part of the complex quantity, $x$. 
The absorption of the multilayer stack is calculated from

\[ A = 1 - T - R \]  \hspace{1cm} (30)

In general we want to be able to calculate the transmission, reflection and absorption characteristics of a heterostructure that has thin films on both sides of the substrate, see Figure 4. To calculate the transmission and reflection values from the substrate/thin films/air side of the heterostructure (labelled \( T_{s.a} \) and \( R_{s.a} \) on Figure 4) equations (26)-(29) become

\[ r_{s.a} = \frac{n_s \cos(\phi_s)(m'_{11} + m'_{12} n_0 \cos(\gamma)) - m'_{21} - m'_{22} n_0 \cos(\gamma)}{n_s \cos(\phi_s)(m'_{11} + m'_{12} n_0 \cos(\gamma)) + m'_{21} + m'_{22} n_0 \cos(\gamma)} \]  \hspace{1cm} (31)

\[ t_{s.a} = \frac{2n_s \cos(\phi_s)}{n_s \cos(\phi_s)(m'_{11} + m'_{12} n_0 \cos(\gamma)) + m'_{21} + m'_{22} n_0 \cos(\gamma)} \]  \hspace{1cm} (32)

\[ T_{s.a} = \frac{\text{Re}(n_0 \cos(\gamma)) |t_{s.a}|^2}{\text{Re}(n_s \cos(\phi_s))} \]  \hspace{1cm} (33)

\[ R_{s.a} = |r_{s.a}|^2 \]  \hspace{1cm} (34)

with \( m'_{ij} \) representing the coefficients of the product matrix on the substrate side which does not have incident electromagnetic radiation.

The above transmission, reflection and absorption values equations (26)-(34) are for the electromagnetic waves propagating into or out of the substrate; but, it is desired that the total transmission, reflection and absorption curves \((T, R, A\ \text{respectively})\) be of the entire heterostructure. Consequently, a correction term is added to the equations (26)-(34) to take into account absorption in the substrate. Because the substrate is, in general, much thicker than a thin film the above theory could not be used directly. Thus the single film
Figure 4. Construction parameters for the calculation of transmission and reflection characteristics for multilayer thin films on both sides of the substrate.
theory of the previous section is applied [6].

More specifically, $T_{s-a}$ and $T_{a-s}$ have already accounted for the reflection at the substrate/thin-film interface (and vice versa); thus equation (2) simplifies to

$$T_s = \exp(-at_s/cos\theta_s)$$

(35)

Also, in general the Fresnel coefficient is small and the attenuation, in the substrate, large thus the major contribution to the overall reflectance of the heterostructure is from the thin film layers on the substrate side which has the incident electromagnetic radiation, $R_{o-s}$. A minor contribution to $R$ comes from reflection from the substrate/thin films interface $R_o$. $R_o$ is small because the wave $R$ must (i) propagate through the substrate and thin films (i.e. films which are on substrate side which has electromagnetic radiation) which leads to large attenuation and (ii) undergoes multiple reflections from the substrate/thin films interface.

$$R_0 = R_{s-a}T_{s-a}'\exp(-at_s/cos\theta_s)$$

(36)

Using the above equations (28), (29), (33), (34), (35), and (36) the total transmission and reflection coefficients of the heterostructure of Figure 4 are found to be

$$T = T_{s-a}T_{a-s}\exp(-at_s/cos\theta_s)$$

(37)

$$R = R_{a-s}+ R_{s-a}T_{s-a}'\exp(-at_s/cos\theta_s)$$

(38)

where $T_{a-s}$, $T_{s-a}$, $T_{s-a}'$, $R_{a-s}$, $R_{s-a}$ are labelled and illustrated in Figure 4.
MAXIMIZING THE T COEFFICIENT

If equations (18)-(29) are solved it can be shown that for maximum transmission of light at a wavelength \( \lambda_0 \) the thickness \( t_1 \) of a single layer thin film should be

\[
t_1 = \lambda_0 \tan^{-1}\left[\frac{2n_1 k_s}{(n_1^2 - n_s^2 - k_s^2)}\right],
\]

(39)

assuming light is falling normally from air onto the film and the substrate has a complex index of refraction \( n_s \). In the case of a weakly or non absorbing substrate, Eq. (39) reduces to

\[
t_1 = \frac{\lambda_0}{4n_1}
\]

(40)

which is the well known quarter wavelength design. For a two layer antireflecting (AR) coating the optimum thicknesses are given by the quarter wavelength and half wavelength design equations, Eqs. (40) and (41) respectively [6,8]. With the layer closest to the substrate being governed by Eq. (41) and the layer furthest from the substrate being governed by Eq. (40).

\[
t_2 = \frac{\lambda_0}{4n_2}
\]

(41)

Hetero's use of this theory will be discussed in chapters Four and Five and Appendix B.
3.1 FEATURES OF HETERO

Aside from Hetero calculating the T/R characteristics as a function of wavelength of a thin film structure it was also desired that the program be very "user-friendly" (i.e. easy to use with a minimal amount of knowledge required to use the program) and versatile; these features of Hetero will now be discussed.

The first and most basic feature of Hetero which makes it user friendly is that the user is prompted, by questions displayed on the terminal, to interactively enter all the important heterostructure data (e.g. thicknesses of the thin films, angle of incident radiation, etc..). In each prompt line the user is, in general, given a choice of picking one variable out of a set of variables to enter. Then the user is shown how to enter the chosen variable. If the user does not enter one of the values listed in the prompt line then the prompt line will be redisplayed giving the user another chance to enter the right response. By checking the entered data, the program prevents user entered mistakes from crashing it because of wrong data types.

The second basic feature of Hetero is that it interactively allows the user to change any of the entered data (i) before running the program to make any last minute changes; and (ii) after running the program to make minor changes in the heterostructure composition. This feature saves time if the
user only wants one or two variables in the heterostructure altered because in these cases the user does not have to reenter all the data and reexecute the program.

To use the multilayer thin film theory of the previous section one must know the refractive index as a function of wavelength for each thin film. Thus it was decided that all the refractive index data that could be obtained from the literature should be collected together and stored on one disc, called Data:. Then when running Hetero to calculate the T/R characteristics, the user simply need enter the filename to obtain the appropriate optical data rather than manually entering this data for each layer before execution. The refractive index data is stored on the floppy disc by chemical formula or most common trade name, which ever is shorter.

The refractive index data that are currently stored on Data: are for the materials: Al, AlN, Ag, Au, Bi, CeO₂, Cu, La₂O₃, Mg, mylar, polyethylene, SiO₂, TiO₂, and ZnO. The data for these materials span the visible to infrared regions (0.3μm - 30μm). The format of the data in each file on Data: is wavelength, real part of refractive index (n) and imaginary part of refractive index (k) with the following constraints:

(1) wavelengths must be in units of angstroms; and
(2) wavelengths, in the file, must have data in increasing order wavelengths starting with the smallest wavelength.

Any wavelength values can be used with variable spacing between wavelength values; and up to 45 data points may be in this
file. This maximum number of data points may be changed by changing the index values in arrays mat_type2 and mat_type3 which are declared in unit enter_data. This feature of allowing any wavelength values is allowed because: (i) hetero interpolates the data for each thin film layer to one set of consistent wavelengths; and (ii) the wavelength range for which the T/R curves are calculated is variable. The range is variable because the range is based on the user entered range and the range of wavelengths in each file; with the smallest range of wavelengths being used. Optical data may be entered on to the floppy by either using the editor and file handling commands of the UCSD p-system commands or by running program Hetero and letting Hetero store the data. This feature is now discussed.

Another feature of hetero is to allow the user to enter the refractive index data for any of the layers. In general this feature is used if the heterostructure of interest is made up of materials for which disc Data: does not currently have the refractive index data; then the program prompts the user obtain the data for these layers so that execution can continue. For the heterostructures where the optical data for two or more of layers is the same the user need only enter the data once simply by naming these layers with the same chemical formula. After all of the data is entered the user can scan through the refractive index data and make any changes. Finally , if the operator wants these data values for future use Hetero will store it on disc data: by the chemical formula the user
entered. For more information about this feature see unit enter_data in Appendix B.

Other features of Hetero include the ability to calculate the T/R characteristics as a function of wavelength of:

1. most thin film materials;
2. up to 10 thin film layers on both sides of the substrate;
3. for any incident angle of electromagnetic radiation (i.e. between 0 +/- 90 degrees with the normal to the plane of incidence);
4. for polarizations of radiation parallel and perpendicular to the plane of incidence; and
5. up to 45 T/R curve data points.

Currently Hetero can handle up to ten thin film layers and 45 T/R data points. These limits were set because the operating system used only had 64K of random access memory (RAM) and Hetero was too large even with memory overlays (discussed in the next section) to be fit on the memory space available. Consequently this program is currently being transferred to the SAGE and IBM Columbia microcomputer systems which have over twice this much memory space. With these systems the number of thin film layers and T/R data points can be increased. The number of T/R data points can be increased by changing arrays mat_type2 and mat_type3 declared in unit enter_data. Similarly, the expansion of the number of thin film layers is done by increasing the index in arrays mat_type1, mat_type3, and mat_type5.
The final feature of Hetero is that it can calculate the T/R curves as a function of wavelength for (i) one thickness of each thin film layer; or (ii) a range of thicknesses for each thin film layer. If the operator would like to obtain a certain percentage of transmission of light (or reflection of infrared radiation) for the multilayer structure of interest then he enters a range of thicknesses. Hetero then iteratively tries to obtain this value (= max_value) by systematically varying each thickness of each thin film a user specified number of times. After each new thickness the T or R curve is calculated and integrated between (i) lower wavelength range limit and 1\mu m for T; or (ii) 1\mu m and the upper wavelength range limit for R. 1\mu m was used as a bound in the integration because it divides the visible and infrared regions. The closest area to max_value and also the maximum and minimum area T/R curves are plotted; along with the appropriate layer thicknesses. Currently only 10 variable layers can be used as was previously discussed. To increase this value one must change arrays: mat_type1, mat_type3 and mat_type5 in unit enter_data. The programmer must also change the repeat-until loops in procedure calc_and_store_data appropriately; see this procedure for more information.

3.2 EXECUTING HETERO

To run program Hetero the user needs:

(1) the discs called (a) Heterost:, which contains the entire compiled program and (b) Data:, which contains all the index data as a function of wavelength.
(2) the refractive index data for layers which do not have optical data on disc Data:

(3) a basic picture of the heterostructure which contains the thicknesses of each layer, the composition of each layer, angle of incidence of radiation. With this picture the user can then just read off the values as the program requests it.

The operator then boots up disc Hetero and X)ecutes file Hetero. Hetero will first prompt the user to put the disc Data: into disc drive B and then it will prompt the user to enter all the pertinent heterostructure data. Upon completion of execution, Hetero will display a T/R curve plot. The user can then manipulate this plot before getting a hard copy or alter some of the heterostructure data and rerun Hetero. These computer graphics capabilities will not be available on all versions of Hetero as will be discussed in the next section.
CHAPTER FOUR

4.1 HETERO COMPILATION UNITS

Hetero is written in Pascal and runs on the UCSD p-system version IV.0 operating system. Because of the size of Hetero, two p-system extensions to standard Pascal were used; they are (i) segments and (ii) separately compiled subprogram units.

Segments are used in mini and microcomputer operating systems to allow the execution of programs which are too large to fit entirely in random access memory (RAM). Since only the section of program code which is actually running need be in RAM, it is advantageous in systems with severe memory limitations to first load only the segments of the program which are to be run immediately into RAM; then when one segment is finished and a new segment is needed the new segment is simply written over the used segment. This process continues until execution of the program is completed. More specifically, a segment is the minimum amount of program code which can be read into RAM or overwritten in RAM. While segments can vastly increase the size of programs that the operating system can handle, a programmer must not make too liberal a use of segments because the execution time of the program can become unreasonably slow. Functions, procedures and processes may be explicitly declared as code segments.

A UCSD subprogram module is called a compilation unit. A unit is a group of separately compiled types, constants, variables, functions procedures and processes which can be
linked into a program for use at run time [5]. Units provide a means of packaging programs in a modular fashion, so that application libraries can be built. Ideal uses for units are the grouping together of sections of program which form self consistent packages such as defining all complex number operations or hyperbolic trigonometric functions; these units can then be used by any other programs. A compilation unit consists of an interface part, an implementation part and an optional initialization/termination code. The later part was not used and hence shall not be mentioned; for more information see reference 9.

The interface part of a unit contains declarations (i.e. constant, type, and variable) and procedure and function headings. These may be used by the program or other units that use the unit [9]. The implementation part contains other declarations and the code for the procedures and functions that were declared in the interface part. Procedures and functions can also be declared in the implementation part. These routines are generally used to help in the realization of the routines declared in the interface part; however, only declarations made in the interface part are available to the calling unit or program.

The program Hetero was "broken up" into units with each unit performing one specific function in the task of calculating the T/R characteristics of a multilayer thin film structure. These compilation units (shown in Figure 1) are now discussed, beginning with the units at the bottom of Figure 1.
The units in the figure are grouped primarily in the order of execution; where the bottom compilation units are executed first.

The first unit executed is Initialize. The purpose of this unit is to obtain all of the pertinent heterostructure data for the calculation of the T/R (Eqs. (37) and (38)) characteristics of a user defined heterostructure. The important data needed is:

1. **total number of thin film layers on substrate**;
2. **The number of thin film layers on both sides of the substrate**.
3. **The incident angle of electromagnetic radiation**;
4. **The angle of polarization of the incident radiation**.
5. **The refractive index of each layer as a function of wavelength**. As mentioned in chapter three most of this data is already stored on a floppy disc named DATA: and to obtain it the user need simply specify the chemical formula. However if the refractive index data for some of the thin film layers is not currently available the user has the option to enter his own data (see unit Enter_Data).

6. **The thickness of each layer or range of thicknesses**. As mentioned in chapter three if the user enters only one thickness for each layer of the structure then only one T/R curve will be calculated. If however the user enters a range of thicknesses for each thin film, then the program will try a specified number of thicknesses...
for each layer and for each new thickness a new T/R curve will be calculated. The curves with maximum area, minimum area, and the user specified value (i.e. max_value) will then be saved and displayed; along with the corresponding thin film thicknesses.

This unit also allows the user to alter any of this pertinent heterostructure data before calculation of the T/R curves; this gives the user the chance to make last minute alterations.

The second unit executed is enter_data. The purpose of this unit is to enter the optical data for the layers of the heterostructure which do not have refractive index data currently present on floppy disc Data:. For the other layers which have optical data on the floppy the user need only enter the filename. The optical data, for each layer with no data on Data:, can be entered with any wavelength, n and k values as long as they are in increasing order of wavelengths. After entering the optical data, this unit allows the operator to then scan the refractive index data and make the any appropriate changes. Finally, the user is given the chance to store any of this optical data for future use on Data: by its chemical formula simply by entering its layer position in the heterostructure.

The third unit executed is Set_up_data. The purpose of this unit is to first generate a set of wavelengths from the number of T/R data points and the range of wavelengths for which the T/R curves are to be plotted. Then using this set of wavelengths interpolate the optical data for each layer to this
set of wavelengths so that the T/R curves can be calculated.

The fourth unit used is calc_value. The purpose of this unit is to take the optical data for all the layers and first find the characteristic matrices (Eq. (25)) for each layer; then find the product matrix for each side of the substrate (Eq. (24)) and finally calculate the transmission and reflection coefficients (Eqs. (37) and (38)) of the heterostructure for the number of data points specified.

Some of the procedures in Hetero are then executed. The primary purpose of this unit is to call all the other units. It's secondary purpose is to take the generated transmission and reflection data, from unit calc_value and either (1) store it for a plot routine; this case occurs when the user wants the T/R curves calculated for only one thickness of each thin film; or (2) finds the optimum T/R curves. The optimum T/R curves are the ones with maximum area, minimum area and max_value. These optimum curves are found by integrating the the T or R data as was explained in chapter three.

Computer graphics programming languages and packages are not yet standardized such that one graphics language can be used (by recompilation of the program) on any system. Since it was desired to have Hetero in a form where it could be universally used on any microcomputer that supported UCSD pascal, Hetero in general will simply display the T/R characteristics as tables on the screen. But for my research it was easier to see a visual picture of the results, consequently a unit Plot_result was added to Hetero. It is mentioned here
for completeness sake. The purpose of this unit is to display a plot of the T/R or absorption curves of the operator defined heterostructure. In addition this unit also performs elementary computer graphics which allows the user to zoom in on any part of the plot, scale the plot etc..

As is shown in Figure 1, to expand Hetero the programmer must first add the appropriate prompt lines to unit initialize so that the user can enter any needed data for this future unit. Then the programmer must add the new unit and a procedure call statement in the main program (in unit Hetero). Finally he must add a line of text in a file called Userlib.text (stored on disc Heterost:) of the form Future.Code; this line is needed for the "Uses" statement at the beginning of each unit, see reference 9.
CHAPTER FIVE

5.1 BACKGROUND

Polyethylene is currently used as glazing material for approximately half of the commercial greenhouses in North America because of its availability and low cost. Standard construction comprises two or three polyethylene layers, with each layer either 100 or 150\(\mu\)m thick. Air is usually forced between the layers to keep them apart. A typical quality greenhouse polyethylene sheet has a transmittance of 88\% and a reflectance of approximately 12\% for solar energy. Polyethylene is essentially opaque to middle infrared radiation(8-15\(\mu\)m) and has a high emissivity of about 0.88 for room temperature thermal radiation. The solar transmittance of polyethylene can be improved by applying antireflection (AR) coatings. The emissivity for room temperature infrared radiation may be decreased by applying a heat-mirror coating which strongly reflects this middle infrared radiation. The primary purpose of our research was to design an AR/heat-mirror coating which follows the plant sensitivity curve and optimizes both solar transmission and reflectance of room temperature thermal radiation. To achieve these results we have considered a two layer coating consisting of a ZnO heat-mirror [10] capped with a SiO\(_2\) antireflection layer. Our results, however, are applicable to other oxide-type heat-mirrors and other low refractive index materials.

In this chapter we describe our calculation of the
reflectance \( R \) and transmittance \( T \) of the two layer coating on a polyethylene film. We relate this result to greenhouse applications. Our design criteria are discussed first, followed by our method of obtaining the optical constants for \( \text{SiO}_2 \), \( \text{ZnO} \) and polyethylene. The results of our calculations give a final choice of layer thicknesses for optimal performance. The cost-effectiveness of the heterostructure for greenhouse applications is discussed.

5.2 DESIGN CRITERIA FOR THE MULTILAYER STRUCTURE

We are interested in an optical coating which satisfies the following criteria:

C.1 A transmission curve close to that of the plant sensitivity curve; given in Figure 5, [11].
C.2 Minimum radiative heat loss;
C.3 Improved solar gain above that of a standard two or three layer polyethylene sheet greenhouse; and
C.4 A cost-effective product.

To a first approximation, the first design criterion, C.1, is satisfied by maximizing the transmission of light at \( \lambda_0 = 0.566\mu\text{m} \); i.e. the geometric centre of the plant sensitivity curve (see Figure 5). Similarly, design criterion C.2, can be satisfied by maximizing the reflection of infrared radiation at 10\( \mu\text{m} \) (the maximum for room temperature blackbody radiation). More accurate figures of merit which enable better comparison of performance characteristics of various optical coatings are:

\[
\%\text{plant} = \frac{\int_{0}^{\infty} T(\lambda) S(\lambda) I(\lambda) d\lambda}{\int_{0}^{\infty} S(\lambda) I(\lambda) d\lambda} \times 100\% \quad (42)
\]

\[
\%\text{sol}ar = \frac{\int_{0}^{\infty} T(\lambda) I(\lambda) d\lambda}{\int_{0}^{\infty} I(\lambda) d\lambda} \times 100\% \quad (43)
\]
Fig. 5. 1, 2 and 3 are (%) Transmission through one, two and three layers of uncoated polyethylene, respectively. The coated double layer is the transmission through our heterostructure (Fig. 3) with $t(SiO_2) = 0.0969 \mu m$, $t(ZnO) = 0.5 \mu m$ and $t(polyethylene) = 150 \mu m$. Normalized to 100% is the plant sensitivity.
U-value = thermal conductance of heterostructure.

where

\[ \lambda \] = wavelength;
\[ T(\lambda) \] = the transmittance through the complete glazing unit;
\[ S(\lambda) \] = the normalized values of plant sensitivity; and
\[ I(\lambda) \] = the intensity of the solar spectrum.

The cost-effectiveness of the coating is estimated using the following information gathered from commercial greenhouse growers in Canada. A typical greenhouse crop yield is about $56.00/m^2 (Canadian funds) of plastic coating, with a 1% increase in %plant giving about a 1% increase in crop yield [12]. Typical heating costs are about $8.60/m^2 of plastic covering, with approximately half the cost attributed to temperature control and the other half to humidity control [12]. A 1% increase in the U-value is expected to decrease the temperature costs by 1% whereas the humidity costs are expected to be unchanged. Finally, the %solar should be as large as possible to maximize passive solar heating.

5.3 CHOICE OF MATERIALS

Polyethylene substrate was chosen for the purpose of this study over other plastics because:

(1) It is widely used in the industry;
(2) Greenhouse quality polyethylene has negligible absorptivity for visible light;
(3) Polyethylene is available in a large variety of sizes, in particular, large, wide rolls appropriate for greenhouses;
(4) It is inexpensive; and
(5) It has good mechanical properties and does not rip as easily as many plastics.

In the present study, we choose our materials primarily to
satisfy the first two design criteria, C.1 and C.2. To achieve maximum possible transmission around the plant sensitivity peak we require an antireflection (AR) coating. Unfortunately, all materials that can be used as single layer "heat-mirrors" [13] cannot also be used as an AR coating on polyethylene due to an insufficiently low refractive index. Consequently we are lead to adding an AR coating onto the heat-mirror layer. SiO₂ and ZnO were chosen for the AR and heat-mirror coating, respectively, because they have the necessary refractive indices for a two layer AR structure [6]. Also, SiO₂ is very stable and provides an excellent protective layer, and ZnO can be fabricated as a good heat-mirror onto a plastic substrate [10].

5.4 DESIGN OF HEAT MIRROR HETEROSTRUCTURE

In order to calculate the transmission-reflection characteristics of a multilayer coating [6,8], (Eqs. (37) and (38)) one must know the refractive index of each layer. Measurements of the refractive index of SiO₂ are found in the literature [14-21]. Figure 6 shows these experimental values of the real and imaginary parts of refractive index, n and k respectively, as a function of wavelength. In the case of ZnO and polyethylene we could not find complete experimental refractive index data in the literature for the range of interest, 0.3 to 40 \( \mu \text{m} \) wavelength. Consequently, theoretical analysis of our own spectral data was used to fill in these gaps. The derivation of the complex indices for ZnO and
Fig. 6. The refractive index $n$ and extinction coefficient $k$ curves for ZnO, SiO₂ and polyethylene.
polyethylene are outlined in the Appendix A.

Our initial choices of thicknesses for the SiO$_2$/ZnO coating were the quarter wavelength ($t_1 = \lambda_0/4n_1 = 0.0969\mu m$) and half wavelength ($t_2 = \lambda_0/2n_2 = 0.1419\mu m$) layer thicknesses for a two layer AR coating [6] (Eqs. (40) and (41)); where $\lambda_0 = 0.566\mu m$, and $n_1$ and $n_2$ are the refractive indices for SiO$_2$ and ZnO, respectively. With $\%_{\text{plant}}$ near maximum, we then systematically varied the thickness of SiO$_2$ and ZnO in order to maximize the reflectance of infrared radiation. Varying the thickness of SiO$_2$ did not change the infrared (IR) reflectance. Consequently, we systematically varied the thickness of ZnO while keeping the thickness of SiO$_2$ constant in order to maximize the reflectance of infrared radiation while maintaining $\%_{\text{plant}}$ near its maximum. We found that the infrared reflectance curve "saturated" with a reflectance of 74% at 10\mu m for thicknesses of ZnO greater than or equal to 0.5\mu m, see Figure 7. Increasing the ZnO thickness from a half-wavelength thickness of 0.1419\mu m to 0.5\mu m decreased $\%_{\text{plant}}$ by less than 2%. In summary, with this optically "untuned" structure, $t(\text{SiO}_2) = 0.0969\mu m$ and $t(\text{ZnO}) = 0.5\mu m$, we came very close to simultaneously maximizing the $\%_{\text{plant}}$ and infrared reflectivity.

5.5 PERFORMANCE OF THE GLAZING HETEROSTRUCTURE

A standard polyethylene glazed greenhouse is comprised of two or three polyethylene layers. AR/heat-mirror coatings can be applied on any of these film surfaces. However, as indicated in Figure 8 the preferred film surface is one in the air gap
Figure 7. 1, 2, 3, 4 are the reflectance curves off of ZnO/polyethylene with the thickness of ZnO being 0.5\(\mu\)m, 0.3\(\mu\)m, 0.1\(\mu\)m, and 0.05\(\mu\)m respectively. The thickness of polyethylene is 150\(\mu\)m.
Fig. 8. Cross section of our AR/heat-mirror film for greenhouses. Not to scale.
where the coating is protected against mechanical damage, condensation and other contaminants in the surroundings. Figure 5 compares the calculated transmission of one, two, and three layers of uncoated polyethylene (labeled 1, 2, 3 respectively on the diagram) with the glazing structure shown in Figure 8. From these curves we calculated %plant and %solar. We assumed normal incidence; however, further calculations showed that the transmission properties do not change greatly with angle of incidence, provided the angles are less than 45°. Figures 9 and 10 illustrate the variation of the transmission of light through SiO₂/ZnO/polyethylene with angle of incidence for parallel (P) and perpendicular (S) polarizations respectively.

Using our results of emissivity we calculated the U-value with standard heat transfer formalism [22] and the following assumptions:

1. Polyethylene is a perfect absorber in the range 7-12μm, where the room temperature blackbody spectrum is a maximum;
2. The layers of polyethylene are separated by a uniform air gap of 2 cm; and
3. The air between the layers is still.
4. The outside temperature is 0°C and the inside is 20°C.

A summary of our calculated %plant, %solar and the U-values are given in Table 1. In addition, the cost-effectiveness is compared in Table 2. It is evident that achieving maximum possible light transmission for plant growth is the most important factor (within the assumptions of this paper) in determining the cost effectiveness of the glazing heterostructure. Finally, it should be noted that AR/heat-mirror coatings should also be useful on glass covered
Figure 9. Angular dependence of transmittance through SiO$_2$/ZnO/polyethylene for an incoming wave polarized parallel (P) to the plane of incidence. 0, 30, 45, 60 are the angles of incidence in degrees of the incoming wave.
Figure 10. Angular dependence of transmittance through SiO$_2$/ZnO/polyethylene structure for an incoming wave polarized perpendicular (S) to the plane of incidence. 0, 30, 45, 60 are the angles of incidence in degrees of the incoming wave.
Table I. Optical Constants

<table>
<thead>
<tr>
<th>SAMPLE</th>
<th>%PLANT</th>
<th>%SOLAR</th>
<th>U-VALUE (W/m²K)</th>
<th>HEAT LOSS (W/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two layers of uncoated polyethylene separated by a 2.0cm air gap.</td>
<td>59</td>
<td>71</td>
<td>0.337</td>
<td>59.4</td>
</tr>
<tr>
<td>Three layers of uncoated polyethylene separated by a 2.0cm air gap.</td>
<td>47</td>
<td>60.3</td>
<td>0.486</td>
<td>41.1</td>
</tr>
<tr>
<td>Our heterostructure.*</td>
<td>70.5</td>
<td>78</td>
<td>0.438</td>
<td>40.0</td>
</tr>
</tbody>
</table>

* polyethylene/air gap/SiO₂/ZnO/polyethylene

Table I. Comparison of heterostructure and polyethylene using design criteria C.1 - C.3.
Table II. Cost Effectiveness

<table>
<thead>
<tr>
<th>SAMPLE</th>
<th>INCREASE IN CROP YIELD (1/m²)</th>
<th>DECREASE IN HEATING COSTS (1/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterostructure versus two layers of uncoated polyethylene.</td>
<td>$9.10</td>
<td>$1.00</td>
</tr>
<tr>
<td>Heterostructure versus three layers of uncoated polyethylene.</td>
<td>$18.60</td>
<td>-$0.5</td>
</tr>
</tbody>
</table>

Table II. Cost effectiveness of our film coating versus uncoated polyethylene.
greenhouses for increasing crop yields.

There are a wide variety of wide bandgap semiconductors that have been used as heat-mirrors, of which ZnO is a representative material. Other heat-mirrors, such as In$_2$O$_3$:Sn, SnO$_2$, Cd$_2$SnO$_4$, and In$_2$O$_3$:Sb appear to be equally suitable for greenhouse applications. We believe that the performance and manufacturing cost of these materials are comparable to ZnO. Similarly, there are other types of materials and constructions for AR coatings that could be substituted for SiO$_2$. 
CONCLUSION

The convenience of using Hetero to calculate the transmission, reflection and absorption characteristics of multilayer thin films on one or both sides of a substrate has been discussed. The program has already been used by some of the UBC Physics sputtering group to aid in their research.

To test Hetero, a ZnO heat mirror film capped with a SiO$_2$ coating was studied for greenhouse applications. The results of calculations given here for a polyethylene/air gap/SiO$_2$/ZnO/polyethylene glazing unit are encouraging. From Table 2 a "heat-mirror" coating combined with an AR coating appears to have a payback in one year. The major benefit of the coating is in increased crop yield. Heat conservation appears to be a minor benefit.
REFERENCES


12. Private communication with Alec Mackenzie, President of Dome Greenhouses; Surrey B.C., Canada.


APPENDIX A: Determination of Optical Constants for ZnO and Polyethylene

In order to calculate the reflectance and transmittance coefficients [6,8], we require knowledge of the index of refraction over the wavelength range of interest for each material in the heterostructure. For greenhouse applications this corresponds to wavelengths from the near UV to the far infrared (0.3 to 40μm).

Optical constants of most plastics, including polyethylene, are not well known because the thickness variations, additives, draw rates and other manufacturing processes which change from one manufacturer to another. However to a good approximation the real part of the index of refraction of polyethylene is constant [23,24]. The imaginary part of the refractive index varies substantially between manufacturing processes. We obtained extinction coefficient data for polyethylene by using a 150μm thick representative sample of greenhouse polyethylene obtained from reference 12. Using a dual beam Beckmann Spectrophotometer UV 5270 (with an integrating sphere for reflection measurements) and a Perkin Elmer infrared Spectrophotometer model #598, we obtained transmission-reflection plots between 0.35-40μm of a sample of greenhouse quality polyethylene, represented in Figure 5. In the 0.4 - 2.5μm range, polyethylene absorbs very little light and has a transmission of approximately 88% and reflection of 12%. In the 2.5 - 30μm range two interface thin film theory was used to find the extinction coefficient [7] (see Eqs. (16) and
(17)). Finally, in the 30-40µm range the values of reference 24 were used. The n and k values used for polyethylene are shown in Figure 6 for all wavelengths.

The optical constants of ZnO over a large range of wavelengths are not well known. By dividing the range of interest into two areas, the visible and infrared, and by using different models for each area we were able to obtain satisfactory data.

In the visible region the absorption of heat-mirror quality ZnO is negligible [10], thus the index of refraction was taken to be real in this range with value n = 2.0 [25].

Recent work by Brett and Parsons [26] has shown that the infrared reflection characteristics of ZnO are well modelled by Drude free electron theory. This theory predicts the optical constants to be given by:

\[ n^2 - k^2 = \frac{\varepsilon_g}{\varepsilon_0 m^*(\omega^2 + \gamma^2)} \]  
\[ 2nk = \frac{N\varepsilon_g\gamma}{\varepsilon_0 m^*\omega(\omega^2 + \gamma^2)} \]

where \( \gamma \) is a damping factor which depends on the mobility (\( \mu \)) of charge carriers \( \gamma = e/\mu m^* \), \( \varepsilon_g \) is the dielectric constant in the absence of free charges; \( N \) is the free carrier concentration and \( m^* \) is the electron effective mass. Films of best IR reflectivity had \( N = 1.8E20 \) cm\(^3\), \( \mu = 20 \) cm\(^2\)/Vs, and \( \sigma = 500 \) Ωcm. In addition we used \( m^* = 0.24m_e \) [27] and \( \varepsilon_g = 3.7 \) [28]. With these values we generated the optical constants for ZnO. Figure 6 shows the refractive indices of ZnO as a function of wavelength.
APPENDIX B

LISTING OF PROGRAM HETERO
UNIT HYP_TRIG;

{*********************************************************************************************}
{ UNIT HYPERBOLIC TRIGONOMETRIC FUNCTIONS }
{*********************************************************************************************}
{ PURPOSE: To define some of the more elementary hyperbolic functions that need to be used in the transmission and reflection calculations. }
{*********************************************************************************************}

INTERFACE
{ Cosh(x) finds the hyperbolic cosine of x.}
Function Cosh(x: real): real;
{ Sinh(x) finds the hyperbolic sine of x.}
Function Sinh(x: real): real;

IMPLEMENTATION

Function Cosh;
Begin
  cosh := (exp(x) + exp(-x))/2;
End; {cosh}

Function Sinh;
Begin
  sinh := (exp(x) - exp(-x))/2;
End; {sinh}

End.
UNIT CMPLX_MATH;

***************
{ UNIT COMPLEX MATH
***************

PURPOSE: To define the elementary complex arithmetic rules needed in the calculation of the transmission and reflection curves.

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

INTERFACE

type
c_type = record
    re: real;
    im: real;
end;
m_type = record
    a1,a2,a3,a4: c_type;
end;

{CMPLX_MULT takes two complex numbers, z1 & z2, and multiplies them together. The complex result is returned.}
Procedure Cmplx_Mult(z1,z2: c_type; var mult: c_type);

{CMPLX_DIVSN takes two complex numbers, divides one into the other, z1/z2, and returns the complex result.}
Procedure Cmplx_Divsn(z1,z2: c_type; var divsn: c_type);

{CMPLX_ADD takes two complex numbers, z1 & z2, adds them together and returns the result.}
Procedure Cmplx_Add(z1,z2: c_type; var add: c_type);

{CMPLX_SUB takes two complex numbers, z1 & z2, subtracts them from each other and returns the complex result.}
Procedure Cmplx_Sub(z1,z2: c_type; var sub: c_type);

{MULT_MATRIX multiplies together two complex 2x2 matrices and returns the result of the complex product matrix. [Note the determinant of the product matrix is still unity.]
Procedure Mult_Matrix(w1,w2: m_type; var w3: m_type);

IMPLEMENTATION
Procedure Cmplx_Mult; {z1,z2: c_type; var mult: c_type}
Begin
  with z1 do
  begin
    mult.re := (re * z2.re - im * z2.im);
    mult.im := (im * z2.re + re * z2.im);
  end; { with z1}
End; {cmplx_mult}

Procedure Cmplx_Divsn; {z1,z2: c_type; var divsn: c_type}
Begin
  with z1 do
  begin
    if (abs(z2.re) <= 1.0e16) and (abs(z2.im) <= 1.0e16)
    then
      begin
        divsn.re := (re * z2.re + im * z2.im)/(sqr(z2.re) +
                    + sqr(z2.im));
        divsn.im := (im * z2.re - re * z2.im)/(sqr(z2.re) +
                    + sqr(z2.im));
      end
    else
      begin
        divsn.re := (re/z2.im + im/z2.re)/(z2.re/z2.im +
                    + z2.im/z2.re);
        divsn.im := (im/z2.im - re/z2.re)/(z2.re/z2.im +
                    + z2.im/z2.re);
      end; {if then else}
  end; {with z1}
End; { cmplx_divsn}

Procedure Cmplx_Add; {z1,z2: c_type; var add: c_type}
Begin
  with z1 do
  begin
    add.re := re + z2.re;
    add.im := im + z2.im;
  end; {with z1}
End; {cmplx_add}

Procedure Cmplx_Sub; {z1,z2: c_type; var sub: c_type}
Begin
  with z1 do
  begin
    sub.re := re - z2.re;
    sub.im := im - z2.im;
  end; {with z1}
End; {cmplx_sub}

{Coeff_matrix multiplies together the appropriate complex }
53

{numbers to return a value which is a coefficient of the }
{product matrix (see procedure Mult-Matrix).}  

Procedure Coeff_Matrix(z1,z2,z3,z4: c_type; var coeff: 
c_type);

var
  temp1,temp2: c_type;
Begin
  cmplx_mult(z1,z2,temp1);
  cmplx_mult(z3,z4,temp2);
  cmplx_add(temp1,temp2,coeff);
End; {coeff_matrix}

Procedure Mult_Matrix; {w1,w2: m_type; var w3: m_type}

Begin
  With w3 do
    begin
      coeff_matrix(w1.a1,w2.a1,w1.a2,w2.a3,a1);
      coeff_matrix(w1.a1,w2.a2,w1.a2,w2.a4,a2);
      coeff_matrix(w1.a3,w2.a1,w1.a4,w2.a3,a3);
      coeff_matrix(w1.a3,w2.a2,w1.a4,w2.a4,a4);
    end; {with}
End; {mult_matrix} End.
UNIT INITIALIZE_SYSTEM;

{***********************************************************************************************

UNIT INITIALIZE_SYSTEM

{******************************************************************************

PURPOSE: Prompts the user to enter all the important data required (e.g. chemical composition of each layer, thicknesses etc..) for the thin film heterostructure of interest. This unit also allows the user a chance to alter any of this pertinent heterostructure data before calculation of the transmission and reflection curves. The data entered from this unit is then used by the remaining procedures of the program to calculate the transmission and reflection T/R characteristics [6,8].

************************************************************}

INTERFACE USES screen_ops,
{\$U enter.code} enter_data;

{INTRODUCTION is a brief statement, displayed at the beginning of the program, on what the program can do.}

Procedure Introduction;

{***********************************************************************************************

START

{******************************************************************************

PURPOSE: To get most and/or all of the important heterostructure variables from the user for the transmission and reflection curve calculations.

DESCRIPTION: This procedure obtains the thin film structure variables needed by prompt lines displayed on the terminal. The user is told what to enter as a response and given an example of how to enter it. Surrounding each "prompt-response" case is a pascal "repeat -until" statement which will continue displaying the same prompt until one of a list of correct responses is received. This prevents user entered mistakes from crashing the program.

INPUTS: The user enters:}
(1) the total number of thin film layers = tot_num_lay
(2) the number of thin film layers on both sides of the substrate.
(3) the incident angle of electromagnetic radiation = theata.
(4) the chemical name for each layer of the structure (=name[m]) so that the refractive index data can be obtained; the chemical name = filename on floppy disc DATA:. Note that for those layers with no optical data on the floppy disc DATA:, the user can enter his own refractive index (n and k) data values. See unit ENTER_DATA.
(5) the thicknesses (= thick[i]) or range of thicknesses (= thick_up and thick_low) for each layer. If a range of thicknesses for each thin film layer is entered then the program will systematically vary each thickness in order to maximize the T or R of electromagnetic radiation.
(6) wavelength range for which the user wants the transmission and reflection characteristics calculated. = low_lamda and up_lamda.
(7) number of data points for which the user wants the transmission and reflection curves calculated = vis_spacing + ir_spacing.

OUTPUTS: Prompt lines to terminal for user to enter data.

NOTES: (1) This procedure calls the procedures RANGE, ENTER_OWN_DATA and ALTER_DATA

Procedure Start;

ALTER_DATA

PURPOSE: To allow the user a chance to change any of the data that he has just entered in procedure START before calculation of the T/R characteristics. It is also used in the main program to allow the user to alter the currently existing data and re-run the program.
DESCRIPTION: This procedure allows all of the variables specified in section INPUTS of procedure START to be changed by means of a "case" statement. The user is prompted to enter what he wants changed; and then he is prompted to change this one variable appropriately. This continues until all variables to be changed are changed.

INPUTS: The user enters a letter [a..g] which represents one of the variables to be changed.

OUTPUTS: Prompt lines to terminal for user to pick a variable and change it appropriately.

NOTES: This procedure calls procedures START and RANGE.

********************************************
Procedure Alter_Data(var ans : char);

IMPLEMENTATION

Segment Procedure S_intro;
Begin
    sc_clr_screen;
    writeln;
    writeln('Put disc called DATA: into drive B ');
    gotoxy(0,8);
    writelnC *** EVALUATION OF TRANSMITTANCE & REFLECTANCE
    COEFFICIENTS ***');
    writeln;
    writeln(' This program calculates the transmission,';
    writeln(' reflection and ');
    writeln(' absorption characteristics for multilayer thin'
    writeln(' films on a');
    writeln(' substrate based on the characteristic matrix'
    writeln(' formulation');
    writeln;
    writeln;
End; {introduction}

Segment Procedure S_Start;
var
    i :integer; {dummy variable.}
    wavelength,n,k: real; {optical data variables:}
        {n = real part of refractive index}
        {k = imaginary part of refractive index}
Begin
repeat
  writeln('Input the number of layers that your heterostructure has,';)
  write('not counting the substrate, between 1 to 10 layers.';)
  readln(tot_num_lay);
until (tot_num_lay in [1..10]);
writeln;
writeln;
writeln('Input the number of layers that your heterostructure has,';)
write('not counting the substrate, between 1 to 10 layers.';)
readln(tot_num_lay);
writeln;
writeln;
writeln('How many of these layers bound the substrate on the side which');
write('has the incident electromagnetic radiation {ie. left side}?');
readln(num_lay_on_inc);
until (num_lay_on_inc in [0..10]) and (num_lay_on_inc <= tot_num_lay);
writeln;
writeln;
repeat
  writeln('Input the angle of incidence, between 0 - 90 degrees');
  readln(theata);
until (theata <= 90) and (theata >= 0);
{convert theata from degrees to radians}
theata := theata * 3.1416 / 180;
writeln;
angle_ans := 'S';
if theata <> 0 then
  begin
    repeat
      writeln('Is the incident electromagnetic radiation polarized?');
      writeln('parallel (P) or perpendicular (S) to the plane of?');
      writeln('incidence? {e.g. S <return> }');
      readln(angle_ans);
      until angle_ans in ['S','P'];
  end; {if theata <> 0}
writeln;
repeat
  writeln('Do you want to see a list of the files of n & k data currently');
  write('available {y/n} ? ');  
  readln(respon1);
until respon1 in ['y','n'];

{DATA:DISPLAY.TEXT displays the filenames of all the materials for which optical data (i.e. wavelength n k) is currently stored on the floppy disc named DATA:.

if respon1 = 'y' then
  begin
    reset(store_name,'DATA:Display.text');
    while not eof(store_name) do
      begin
        readln(store_name,str_name);
        writeln(str_name);
      end; {while}
    close(store_name,normal);
  end; {if respon1}
repeat
  writeln;
  writeln('Are all the data files you need available {y/n} ? n means that');
  write('you want to enter your own data for some/all of the files. '); 
  readln(respon2);
until respon2 in ['y','n'];

{Some of the layers do not have their own optical data on disc DATA:. User can enter his own data or stop program and get data needed.}
if respon2 = 'n' then
  begin
    repeat
      writeln;
      writeln('Do you want to enter your own values for these materials');
      writeln('so the program will continue {y/n} ,n will exit you '); 
      write('from the program. '); 
      readln(respon3);
    until respon3 in ['y','n'];
    if respon3 = 'y' then
      enter_own_data
    else
      exit(program);
  end; {if response2}

{This option is used when all the optical data for the}
if (respon2 = 'y') then
begin
  respon3 := 'n';
  num_lay_with_no_data := 0;

  {This value will be used in unit SET_UP_DATA to get all the data from DATA:.}
layer_with_own_data[1] := tot_num_lay + 2;
writeln;
writeln('Input chemical formula for the layers starting from layer #1');
writeln('If the chemical formula is greater than six characters in ');
writeln('length then enter the most common name using only the first');
writeln('11 characters in the name {eg. polyethylene = polyethylene}.');
writeln('The substrate name will be asked for last.' );
writeln(' {eg. layer 1 is sio2 <return> }');
for i:= 1 to tot_num_lay do
begin
  repeat
    write('layer ',i,' is ');
    readln(name[i]);
  until (length(name[i]) <= 11 );
end; {for i}
repeat
  write('Substrate is ');
  readln(name[tot_num_lay + 1]);
  until (length(name[i]) <= 11);
{call procedure RANGE to enter the wavelength range for which the T/R curves are to be calculated.}
range;
end; {if respon2 = 'y'}
repeat
writeln;
writeln('Do you want to enter one thickness for each layer {y} or a');
write('range of thicknesses {n} ? ');
readln(respon1);
until respon1 in ['y','n'];
{One thickness for each layer is entered. The T/R curve may not necessarily have the optimum values.}
if respon1 = 'y' then
begin
  writeln;

writeln('Enter the thickness of each layer, again starting from layer ');
writeln('#1, in units of angstroms. {eg. sio2 800 <return> } ');
writeln('Warning: thicknesses of the thin films greater than 1e6 ');
writeln('angstroms will cause the program to crash. Insert the ');
writeln('thickness of the substrate last,in mm.');
writeln;
for i := 1 to tot_num_lay do
  begin
    repeat
      write(name[i],' ');
      readln(thick[i]);
    until (thick[i] >= 0) and (thick[i] <= 100e4);
  end; {if i}
repeat
  write(name[tot_num_lay + 1],' ');
  readln(thick[tot_num_lay + 1]);
{Convert substrate thickness from millimeters to angstroms.}
{angstroms.}
  thick[tot_num_lay + 1] := thick[tot_num_lay + 1] * 1e7;
  until (thick[tot_num_lay + 1] >= 0) and (thick[tot_num_lay + 1] <= 6.5e7);
writeln;
end {if respon1}

{A range of thicknesses is entered. By varying the thicknesses the program tries to maximize the T/R curves.}
else
begin
  writeln;
  writeln;
  writeln('Enter the range of thicknesses for each layer starting from');
  writeln('layer #1, in angstroms.{eg. SiO2 300 800 <return> } ');
  writeln('The substrate thickness will be asked for last; enter only');
  writeln('one thickness for it, in mm. Warning: thicknesses of the ');
  writeln('thin films greater than 1e6 angstroms will cause the program');
  writeln('to crash.');
  writeln;
  for i := 1 to tot_num_lay do
    begin
      repeat
        write(name[i],' ');
        readln(thick[i]);
      until (thick[i] >= 0) and (thick[i] <= 100e4);
    end; {if i}
repeat
  write(name[tot_num_lay + 1],' ');
  readln(thick[tot_num_lay + 1]);
{Convert substrate thickness from millimeters to angstroms.}
{angstroms.}
  thick[tot_num_lay + 1] := thick[tot_num_lay + 1] * 1e7;
  until (thick[tot_num_lay + 1] >= 0) and (thick[tot_num_lay + 1] <= 6.5e7);
writeln;
end {if respon1}
repeat
  write(name[i], ' ');
  readln(thick_low[i], thick_up[i]);
  until (thick_low[i] >= 0) and (thick_low[i] <= 100e4) and (thick_up[i] >= 0) and (thick_up[i] <= 100e4);
end; {for i}
repeat
  write(name[tot_num_lay + 1], ' ');
  readln(thick[tot_num_lay + 1]);
  {Convert the thickness of the substrate to angstroms.}
  thick[tot_num_lay + 1] := thick[tot_num_lay + 1] * 1e7;
  until (thick[tot_num_lay + 1] >= 0) and (thick[tot_num_lay + 1] <= 6.5e7);
{Up to 50 different thicknesses can be tried for each}
{thin film layer. Note that to systematically vary}
{each thin film thickness 50 times will involve}
{(50 * # of thin film layers) iterations through this program; this could take up to 24 hours.}
repeat
  writeln;
  writeln('Enter the number of different thicknesses that you want ');
  write('tried, in this range{1..50}. ');
  readln(points);
  until points in [1..50];
if up_lamda <= 1.0e4 then
  begin
    writeln;
    writeln('The transmission of light will be maximzed');
    respon5 := 't';
  end; {if in vis}
if low_lamda >= 1.0e4 then
  begin
    writeln;
    writeln('The reflectance of IR will be maximized.');
    respon5 := 'r';
  end; {if in ir}
if (low_lamda <= 1.0e4) and (up_lamda >= 1.0e4) then
  begin
    repeat
      writeln;
      writeln('Do you want to maximize the transmission of visible light ');
      writeln('{enter t} or the reflection of IR {enter r} for the range ');
      break;
    until break;
  end; {if (low_lamda <= 1.0e4) and (up_lamda >= 1.0e4)}
write('of thicknesses entered above. ');
readln(respon5);
until respon5 in ['t','r'];
end; {if in both}
repeat
writeln;
writelnln('Enter the maximum transmission {or
reflection} that you ');
writeln('would like to obtain with this range of
thicknesses and ');
write('materials, between 0 - 1.0. ');
readln(max_val);
writeln;
until (max_val >= 0) and (max_val <= 1);
{Convert the maximum value (max_val) to the
appropriate wavelength range.}
if respon5 = 't' then
  max_val := max_val * (1e4 - low_lamda)
else
  max_val := max_val * (up_lamda - 1e4);
end; {if else respon5}
repeat
writelnln('Do you want to change any of the data that you
have just ');
write('entered [y/n] ? ');
readln(respon4);
until respon4 in ['y','n'];
if respon4 = 'y' then
begin
  repeat
    alter_data(ans);
    until ans = 'n';
  end; {if response 4}
sc_clr_screen;
gotoxy(0,0);
write('This program will take anywhere from 1min-1hr to
generate the results...');
{$1+}
End; {start}

Segment Procedure S_alter_data;
var
  m : integer; {m = array index representing the layer
  number.}
begin
  {§1-}
  repeat
    writeln;
    writelnln('What do you want changed? type in');
    writelnln('a layer name a');
    writelnln('a layer thickness b');
  end; {for various options in response 4}
  writelnln('Enter the new value for the layer ');
  readln(ans);
sc_clr_screen;
gotoxy(0,0);
write('The new value for the layer is ');
write(ans);
write(' ');
write('and the index is now ');
write(m);
write('...');
End; {S_alter_data}

{$1-}

writeln('total no. of layers');
writeln('total no. of layers on inc. side');
writeln('wavelength range and/or # of data pts');
writeln('angle of incidence & E.M.R. polarization');
writeln('max. T/R that you want to obtain');
write('');
readln(respond);
until respond in ['a'..'g'];
if respond = 'a' then begin
  start;
  ans := 'n';
end; {if a}
if respond = 'b' then begin
  writeln('Do you want to change the thickness of the substrate{y}
  or the thin films{n} ? ');
  readln(ans);
  if ans = 'y' then begin
    repeat
      write('Enter thickness of substrate, in mm{eg. 0.152 <return>}');
      readln(thick[tot_num_lay + 1]);
      thick[tot_num_lay + 1] := thick[tot_num_lay + 1] * 1e7;
      until thick[tot_num_lay + 1] <= 6.5e7;
  end {thick of substrate}
  else begin
    if respond = 'n' then begin
      repeat
        writeln('Enter the layer number and the new thickness range');
        write(' {eg. 3 800 1000 <return> } ');
        readln(m, thick_low[m], thick_up[m]);
        until (m in [1..10]) and (thick_low[m] >= 0)
        and (thick_low[m] <= 100e4) and
        (thick_up[m] >= 0) and
        (thick_up[m] <= 100e4);          
    end
    else begin
      repeat
        write('Input the layer number and thickness
        {eg. 3 100}');
        write(' <return> } ');
        readln(m, thick[m]);
        until (m in [1..10]) and (thick[m] >= 0) and
        (thick[m] <= 100e4);
    end; {if respond1}
  end; {if else substrate/thin film}
end;
{By changing the number of layers the program needs to}
{know the names and thicknesses of these new layers.}
{Consequently the program jumps back to the beginning of }
{the program where the user must enter "all" the data. }

if respond= 'c' then
begin
start;
ans := 'n';
end; { if c}
if respond = 'd' then
begin
start;
ans := 'n';
end; { if d}
if respond = 'e' then
range;
if respond = 'f' then
begin
repeat
write('Input the new angle of incidence, in 
degrees');
readln(theata);
until (theata>= 0) and (theata <= 90);
theata := theata * 3.1416/180;
if theata <> 0 then
begin
repeat
writeln('Is the incident E.M.R parallel (P) or ');
writeln('perpendicular (S) to the plane of 
incidence?');
writeln('{e.g. P <return>} ');
readln(angle_ans); until angle_ans in ['P','S']
end;
end; {if f}
if respond = 'g' then
begin
if respos1 = 'n' then
begin
writeln('Enter the maximum transmission 
{reflection} that you ');
write('would like to obtain '); readln(max_val);
if respond5 = 't' then
max_val := max_val * (1e4 - low_lamda)
else
max_val := max_val * (up_lamda - 1e4);
end {if respos1}
else
begin
writeln('This option is meant to be used only if 
thickness range');
writeln('for each layer was used.');
end; {if else respond1}
end; {g}
if respond in ['a','b','e','f','g'] then
begin
repeat
writeln;
write('Do you want to change any more data {y/n}?');
readln(ans);
until ans in ['y','n'];
end; {if respond is not c or d}
end; {alter data}

Procedure introduction;
begin
  S_intro;
end; {intro}

Procedure start;
begin
  S_start;
end; {start}

Procedure Alter_data; {var ans: char}
begin
  S_alter_data
end; {alter_data}

End.
UNIT Enter_Data;

PASTE { PURPOSE: To enter optical data for the layers (of the heterostructure) with no data currently present on the floppy disc DATA:. For the other layers (which have refractive index data available on DATA:) the user must enter the chemical formula = filename on DATA:. The optical data, for each layer, can be entered with any wavelength n and k values as long as they are in increasing order of wavelengths. Interpolation is done on n and k values of each layer to get all the layers to have a consistent set of wavelengths. This unit is entered from procedure START. }

INTERFACE

type  
    mat_type1 = packed array[1..10] of real;  {stores the thickness of each layer}  
    mat_type2 = packed array[1..45] of real;  {stores all the wavelengths}  
    mat_type3 = packed array[1..10,1..3,1..45] of real;  {stores all the layer #s, and either wavelength or n or values.}  
    mat_type5 = packed array[1..10] of integer;  {stores the layer #s; for layers which need data to be entered.}  

var  
    low_lamda : real;  {The smallest and largest wavelengths which bound the wavelength range for which T/R characteristics will be found.}  
    up_lamda : real;  {Input in procedure RANGE.}  

theata : real;  
  {Angle of incidence. Input in procedure START.

max_val : real;  
  {The max. T/R value the user hopes to obtain. This is only used when
  the user enters a range of thicknesses. Input in procedure START.

vis_spacing : integer;  
  {The # of points in the visible & IR respectively, that are used in
  calculating the T/R curves. Input in procedure RANGE.

ir_spacing : integer;  
  {The # of different thickness for each layer that the user wants
  tried in the thickness range. Input in procedure START.

tot_num_lay : integer;  
  {The total number of thin film layers. Input in procedure START.

num_lay_on_inc : integer;  
  {The total # of layers on the incident side of structure. Input
  in procedure START.

num_lay_with_no_data: integer;  
  {The # of layers with no data.

name : packed array[1..5] of string[11];  
  {Contains all the file names. Input in procedure START.

thick : mat_type1;  
  {Contains the thicknesses of all the layers. Input in
  procedure START.

thick_low,thick_up: mat_type1;  
  {Contains the lower & upper thickness bounds for each layer.
  These are used only if the user wants to find the optimum
  thickness that maximizes the T/R. Input in procedure START.

dat : mat_type3;  
  {Array which contains the wavelength n & k data from either
  the files on floppy DATA: or as entered manually by user, before
  interpolation is done. Values input either in procedures
  NOW_ENTER or GET_DATA_FROM_FILE.

data : mat_type3;  
  {Array which contains the refractive index data "after"
interpolation is done. The n and k data for the layer has an equivalent set of wavelengths. Set in procedure PICK_AND_STORE.

\[ \text{lamda : mat_type2; \{array holds NOT the file wavelength values but instead the wavelength values which satisfy the # data points wanted.}\] 

\[ \text{all_lyrs_dat : text; \{Pascal internal file used in opening and associating an external data file with it.}\] 

\[ \text{store_name : text; \{Pascal internal file used in conjunction with creating a new data file on DATA:. See procedure ENTER OWN_DATA.}\] 

\[ \text{layer_with_own_data : mat_type5; \{Array holds the layer # that required its own data to be entered.\}} \]

\[ \text{dots : mat_type5; \{Array which tells how many n & k values the ith layer has; only used when user enters his own data. Input in procedure ENTER OWN_DATA.}\] 

\[ \text{respond,ans : char; \{Responses to all the questions that are asked in procedures START and Enter DATA.\}} \]

\[ \text{respon1,respon2: char; \{that are asked in procedures START and Enter DATA.\}} \]

\[ \text{respon3,respon4: char; \{\}\] 

\[ \text{respon5 : char; \{\}\] 

\[ \text{str_name : string; \{Used in procedure START to read file DATA:DISPLAY,TEXT which contains all the data files currently stored on DATA:.\}} \]

\[ \text{angle_ans : char; \{Is equal to either P or S depending on whether the polarization of E.M.R. is parallel or perpendicular to the plane of incidence respectively.\}} \]

\[ \text{********************

ENTER OWN_DATA

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

PURPOSE: To enter the refractive index data for all of the layers with no data currently on the disc DATA:.} \]
DESCRIPTION: The number of layers which are going to have optical data entered is input. Then each layer (which will have data entered) is named by its chemical formula. This naming of layers allows optical data to be entered only once for heterostructures with two or more equivalent composition layers by specifying the same name for each equivalent layer. After all the optical data is entered each layers refractive index data can be scanned and any needed changes can be made. In addition these layers (with user entered optical data) can be saved on DATA:. Finally, the remaining layers, with refractive index data on DATA:, have their chemical formulas entered by the user.

INPUTS: The user enters:
(1) Num_lay_with_no_data = the number of layers with no optical data on disc DATA:.
(2) Wavelength, n and k values for each layer with no data on DATA:. See procedure Now_Enter.
(3) Chemical formula and layer number of each layer = name[m], m respectively
(4) the layers to have their optical data stored on floppy disc DATA:.

OUTPUTS: Prompt lines to terminal for user to enter a particular piece of data.

NOTES:
(1) This procedure obtains the thin film variables specified above by prompt lines. See the DESCRIPTION section of procedure START for more details.
(2) This procedure calls procedure NOW_ENTER and RANGE.
(3) This procedure is called by procedure START

***********

Procedure Enter_Own_Data;

***********

RANGE

***********

PURPOSE: To enter the wavelength range of interest and the number of points in this region for which the T/R curves will be calculated.

INPUTS: The user enters:
(1) Low_Lambda and Up_lambda = the wavelength
Procedure Range;

IMPLEMENTATION

NOW_ENTER

PURPOSE: To specify how many data values are going to be entered for the layer chosen. And then to enter the wavelength, n and k values for this layer.

DESCRIPTION: The user enters the wavelength, n and k values for the number of points he specified (for the layer of interest). Note that the wavelengths entered must be in increasing order. If the lowest wavelength entered is larger than the lower range limit (e.g. low_lamda) then low_lamda is reset to this wavelength value; thus the wavelength range shrinks. This same technique applies to the upper wavelength range limit as well. This technique of adjusting the wavelength range is also used in procedure GET_DATA_FROM_FILE. Finally, the entered data is put into array DAT; it stays in their until interpolation is done on the data (see unit SET_UP_DATA).

INPUTS: The user enters:
(1) dots[m] = the number of data points for layer m that he is going to enter.
(2) wavelength
(3) n and k = real and imaginary parts of refractive index.

OUTPUTS: Prompt lines to terminal to enter above data.

NOTES: (1) This procedure is called by procedure
Procedure Now_Enter(m :integer); forward;

Segment Procedure S_Enter_Own_Data;
var
  l,m,i,j,low,m0,k0 :integer; {Index variables used in For loops and arrays.}
  same : char; {SAME is yes/no depending on whether two layers with the same entered names have the same optical data.}
Begin
  {$I-}$
  repeat
    writeln;
    write('How many layers are there with no data {1..10}? ');
    readln(num_lay_with_no_data);
    until num_lay_with_no_data in [1..10];
    writeln;

    { Call procedure RANGE to enter the wavelength range for which the T/R curves are to be calculated. }
    range;
    writeln;
    writeln('Input the layer no. {1..10} and chemical name in order starting');
    writeln('from the left most layer which does not have a data file. Then ');
    writeln('enter data for layer. If substrate data is going to be entered,');
    writeln('enter it last. {eg. layer # and name are 2 ZnO <return>} ');
    for 1 := 1 to num_lay_with_no_data do begin
      repeat
        write('Layer # and name are '); writeln;
        readln(m,name[m]);
      until (m in [1..10]) and (length(name[m]) <= 11);
      layer_with_own_data[1] := m;
      same := 'd';
      {More than one layer is having data entered manually. }
      if 1 > 1 then begin
        j := 1;
        repeat
          writeln;
          writeln('How many layers are there with no data {1..10}? ');
          readln(num_lay_with_no_data);
          until num_lay_with_no_data in [1..10];
          writeln;
          writeln('Input the layer no. {1..10} and chemical name in order starting');
          writeln('from the left most layer which does not have a data file. Then ');
          writeln('enter data for layer. If substrate data is going to be entered,');
          writeln('enter it last. {eg. layer # and name are 2 ZnO <return>} ');
          for 1 := 1 to num_lay_with_no_data do begin
            repeat
              write('Layer # and name are '); writeln;
              readln(m,name[m]);
            until (m in [1..10]) and (length(name[m]) <= 11);
            layer_with_own_data[1] := m;
            same := 'd';
          end;
        end;
      end;
      {More than one layer is having data entered manually. }
    end;
{Compare the just entered layer name with all 
other entered layer names to see if they are 
the same. }

j := j - 1;
i := layer_with_own_data[j];

{The layer names are the same which probably 
means the refractive index data is the same 
for both layers. If this is so the user only 
has to enter optical data for one of these two 
layers. }

if ((pos(name[i],name[m]) <> 0) and 
(length(name[i]) = length(name[m]))) then 
begin 
repeat 
writeln;
write('Is the data for layers ',i,' and 
',m);
writeln(' the same {y/n} ? ');
readln(same);
until (same in ['y','n']);
if same = 'y' then 
begin 
dots[m] := dots[i];
for k0 := 1 to dots[i] do 
begin 
dat[m,1,k0] := dat[i,1,k0];
dat[m,2,k0] := dat[i,2,k0];
dat[m,3,k0] := dat[i,3,k0];
end; {for k0}
end {if same}

{The user entered the same name for two 
layers with different optical data, so 
he must change the name of one of these 
layers. }

else 
begin 
write('Enter new name ');
readln(name[m]);
now_enter(m); 
end; {if not the same}
end; {names are the same} 
until (j=1);
if same = 'd' then 
now_enter(m);
end {if 1 >1} 
else 
begin 
now_enter(m)
end; {if 1 = 1}
end; {for 1}
repeat
writeln;
write('Do you want to change any entered wavelength, n &
k data {y/n}?');
readln(respond);
if respond = 'y' then
begin
writeln;
writeln('Enter layer for which you would like to see
data');
writeln('{eg. 3 <return>}');
readln(m);
{Scroll out optical data for layer m on screen. }
writeln('WAVELENGTH N K ');
for j := 1 to dots[m] do
begin
writeln(dat[m,3,j],dat[m,1,j],dat[m,2,j]);
end; {for j}
repeat
repeat
writeln('Do you want to change any data in this
layer {y/n}?');
readln(ans);
until (ans in ['y','n']);
{Change typing mistakes of appropriate data }
{values. }
if ans = 'y' then
begin
writeln;
writeln('Input the data point # that you want
changed ');
writeln('and then the new values of
wavelength, n, k ');
writeln('that you want for this point. ');
writeln('{eg. 13 4500 1.45 0.3 <return>}' );
repeat
readln(j,dat[m,3,j],dat[m,1,j],dat[m,2,j]);
until (dat[m,3,j] >= 3000) and (dat[m,3,j]
<= 120e4) and (dat[m,1,j] >= 0) and
(dat[m,1,j] <= 400) and (dat[m,2,j] >=
0) and
(dat[m,2,j] <= 600);
end; {if ans = yes}
until ans = 'n';
end; {if respon = yes}
until respond = 'n';
writeln;
writeln('Do you want to save any of these data files
permanently on');
writeln('floppy disc DATA: [y/n] ? ');
repeat
readln(ans);
until (ans in ['y','n']);

{Save some of the layers entered optical data on floppy }
{DATA:.}

if ans = 'y' then.
begin
writeln;
writeln('Input the # of layers that you want saved ?');
readln(j);
writeln('Input layers that you want saved. ');
writeln('{eg. 2 <return> 5 <return> } ');
repeat
  j := j - 1;
  readln(m);
  rewrite(all_lyrs_dat,concat('data:',name[m],'.text'))
{The format of the data on the floppy is wavelength}
{n k. with the following constraints: wavelengths }
{must be in units of angstroms; the wavelength must}
{have data in increasing order of wavelengths; any}
{wavelength values can be used with any spacing }
{between the wavelengths; And any number of data }
{points may be in the file. Optical data may be }
{entered on to the floppy either using the standard}
{editor and file handling commands, or it may be}
{entered by entering data for this program and then}
{saving it (the next few steps illustrate this). }

for 1 := 1 to dots[m] do begin
  writeln(all_lyrs_dat, dat[m,3,l], dat[m,1,l],
  datTm,2,l]);
end; {for 1}
close(all_lyrs_dat,lock);
reset(store_name,'DATA:display.text');
while not eof(store_name) do begin
  readln(store_name);
end; {while}
writeln(store_name,name[m]);
close(store_name,normal);
until j = 0;
end; {if ans = y}
layer_with_own_data[num_layer_with_no_data + 1] :=
tot_num_lay + 2;
writeln;
writeln('Input chemical formula for the remaining layers needed, starting');
writeln('from layer #1, skipping the layers for which you
just entered ');
writeln('data. If chemical formula is greater than six
characters in ');
writeln('length then enter the most common name using only
the first 11');
writeln('characters in the name. (eg. polyethylene =
polyethyle) ');
writeln('The substrate will be asked for last. ');
writeln;
low := 1;
{The remaining layers have data stored on disc DATA:.}
{Just the names of these layers names are input by using}
{a combination of the two FOR loops below.}

for m := 1 to (num_lay_with_no_data + 1) do
begin
  for m0 := low to (layer_with_own_data[m] - 1) do
  begin
    if (m0 <> tot_num_lay + 1) then
      begin
        repeat
          write('layer ',m0,' is '); readln(name[m0]);
        until (length(name[m0]) <= 11);
      end {if m0}
    else
      begin
        if (layer_with_own_data[num_lay_with_no_data]
        <> tot_num_lay+1) then
          repeat
            write('substrate is '); readln(name[m0]);
          until (length(name[m0]) <= 11);
        end {if else m0}
      end; {for m0}
    low := layer_with_own_data[m] + 1;
  end; {for m}
{S1+}
End; {enter_your_own_data}

Procedure Range;
Begin
repeat
  writeln;
  writeln('Input wavelength range of interest, between
3000-1.2e6 angstroms');
  writeln(' {eg. 3000 10e4 <return> }');
  writeln;
  readln(low_lamda,Up_lamda);
until (low_lamda>=3000) and (Up_lamda<=120e4) and
(low_lamda <= up_lamda);
if (low_lamda <= 10000) then
begin
  repeat
    writeln;
    writeln('Input the number of data points \{1..40\} you want in the \');
    write('visible, between 3000 and 10,000 angstroms.');
    readln(vis_spacing);
    until vis_spacing in [1..40];
  end; {if low_lamda}
if (up_lamda > 1.0e4) then
begin
  repeat
    writeln;
    writeln('Input the number of data points \{40 - vis_spacing\} you want in the IR, between 10,000-1.2e4.');
    readln(ir_spacing);
    until (ir_spacing in [1..40]) and (ir_spacing <= 40 - vis_spacing);
  end; {if up_lamda}
if (low_lamda <= 1.0e4) and (up_lamda <= 1.0e4) then
ir_spacing := 0;
if (low_lamda > 1.0e4) and (up_lamda > 1.0e4) then
vis_spacing := 0;
End; {Procedure Range}

Procedure Enter_own_data;
Begin
  S_enter_own_data;
End; {enter own data}

Procedure Now_Enter;
var
  i,j :integer;
  wavelength,n,k:real;
Begin
  writeln;
  writeln('Enter the number of data points for this layer that you are going to enter. Note that this number does not have to equal');
  writeln('the number of data points in the visible + IR that you want');
  writeln('plotted (interpolation will be done). Also, you must have at \');
  writeln('least two data points in the visible and/or IR.\');
  repeat
    readln(dots[m]);
until (dots[m] >= 0) and (dots[m] <= 40);
writeln;
writeln('Now input wavelength n & k values for the #
of data points');
writeln('given above, start with the smallest wavelength
and work ');
writeln('up to the largest. The range of wavelengths must
be ');
writeln('between 3000 - 120e4 angstroms.
{e.g. 3500 1.5 0.1 <return>}}');
for j := 1 to dots[m] do
begin
repeat
write('Data point ',j,' is ');
readln(wavelength,n,k);
until (wavelength >= 3000) and (wavelength <= 120e4)
and (n >= 0) and (n <= 400) and (k >= 0) and (k <= 600);
dat[m,1,j] := n;
dat[m,2,j] := k;
dat[m,3,j] := wavelength;
{Check to see if the first wavelength entered is .}
{larger than low_lamda. If this is so then adjust}
{range.}
if ((j=1) and (wavelength > low_lamda)) then
low_lamda := wavelength;
end; {for j}
{Check to see if the last wavelength entered }
{Up_lamda is greater. If it is not, then adjust the}
{range.}
if (wavelength < up_lamda) then
up_lamda := wavelength;
writeln;
End; {procedure now_enter data}

End.
UNIT Set_Up_Data;

{*******************************************************************************}
{ UNIT SET_UP_DATA
{*******************************************************************************}

{ PURPOSE: To take the set of wavelengths (stored in array LAMDA; calculated from the wavelength range (= Up_Lamda - Low_Lamda) and the number of data points (= Vis_Spacing + IR_Spacing)) and then interpolate the optical data for each layer to this set of wavelengths. This is done by getting the optical data for each layer entered into array DAT. Array DAT gets the refractive index data from user entered data directly (see procedure Enter_Own_Data) and/or from the refractive index values already on floppy DATA: (see procedure Get_Data_From_File). Then interpolation is used to fit the data for each layer to this set of wavelengths. These refractive index values, consistent with the wavelengths stored in array LAMDA, are stored in DATA:.

{*******************************************************************************

INTERFACE
uses screen_ops,
   {$U enter .code} enter_data,
   {$U initialize.code} Initialize;

{*******************************************************************************
{ GET_DATA_FROM_FILE
{*******************************************************************************

{ PURPOSE: To load only the wavelength and refractive index data for the layers, which have data files on floppy DATA:, into array DAT. The other layers which had their optical data entered (in procedure Enter_Own_Data) have data already already in some regions of DAT.

{ DESCRIPTION: A layer (m0) with optical data on disc DATA is selected. Then the name the user assigned to this layer (= name[m0]) is used to open this file on the floppy (i.e. reset(all_lyr_dat DATA:name[m0].text)). With the file open all the wavelength, n and k data is read from this file. These steps are repeated for all the layers that have optical data on disc DATA:.
Note that if the lowest wavelength in this file is larger than the lower wavelength range limit (e.g. Low_Lamda) then Low_Lamda is reset to this higher value, thus shrinking the RANGE. This same technique applies to the upper wavelength range limit. This technique of adjusting the range is also used in procedure Now_Enter.

INPUTS: Optical data (i.e. wavelength, n and k) from files on disc DATA:.

OUTPUTS: Prompt lines to screen, only if the user calls calls for a file not on DATA:.

NOTES: (1) The selection of layers with optical data "only" on floppy DATA: is done by the combination of FOR loops.
(2) If one of the data files is not on the floppy (e.g. user misspelled the filename) then this routine lets the user respell the filename so the program will continue.
(3) This procedure calls procedure MODIFY_DATA.
(4) This procedure is called from MAIN.

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

Procedure Get_Data_From_File;

IMPLEMENTATION
var
  i,j,m :integer; {indices used to get at points in arrays data, dat and lamda.}
  {m = is used to represent the layer no. in array DAT & DATA.}
  {j = is used to index n or k or wavelength values from array DAT.}
  {i = is used to represent either n or k values from array DATA.}
  wave :real; {A variable used in Pick_and_store to aid in interpolation.}

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

INTERPOLATE FROM ARRAY DAT FOR n & k

PURPOSE: To use array DAT and a wavelength associated with a data point (called count_lamda) to interpolate between some n and k values for the n and k values associated with the wavelength count lamda.
DESCRIPTION: n values from DAT are interpolated to coincide with wavelengths, count_lamda, using the formula

\[
\text{slope} = \frac{\lambda_2 - \lambda_1}{n_2 - n_1} = \frac{\text{count}_\text{lamda} - \lambda_1}{n(\text{count}_\text{lamda}) - n_1}
\]

with

\(n_1, n_2\) = the real parts of refractive index (from array DAT) before interpolation

\(n(\text{count}_\text{lamda})\) = the refractive index that coincides with the wavelength value \(\text{count}_\text{lamda}\) (which is stored in array LAMDA). It will be stored in array DATA, see procedure PICK\_STORE.

\(\lambda_1, \lambda_2\) = wavelength values from DAT

\(\text{count}_\text{lamda}\) = a wavelength data point.

The above technique is also used to find \(k(\text{count}_\text{lamda})\).

INPUTS: (1) \(\text{count}_\text{lamda}\) = a data point wavelength which is defined in procedure MODIFY FILE.
(2) refractive index \((n \text{ &} k)\) and wavelength values form array DAT.

OUTPUTS The refractive index values that coincide with wavelengths in LAMDA.

NOTES: (1) This procedure is called by procedure PICK\_and\_STORE.

**Int\_From\_Table\_For\_n\_\_k(count\_lamda: real; var n & k: real); forward;**

**PICK\_AND\_STORE**

**PURPOSE:** To get all the layers \(n \text{ &} k\) values to coincide with the data points \(\text{count}_\text{lamda}\) (usually by interpolation); and to store these refractive index values in array DATA. Also to store the \(\text{count}_\text{lamda}\) values in array LAMDA.

**DESCRIPTION:** A wavelength value is obtained from array DAT and compared to the data point \(\text{count}_\text{lamda}\).

If the wavelengths are the same then these \(n\)
and k values and count_lamda are stored in new arrays called DATA and LAMDA; if the wavelengths are not the same then the procedure INT_FROM_TABLR_FOR_n & k is used to find the n and k values associated with count_lamda, and again, these new n & k and count_lamda are stored in arrays called DATA and LAMDA, respectively. This is repeated for all wavelengths in each layer and then all layers.

**INPUTS:**
1. count_lamda = a data point wavelength which is defined in procedure MODIFY_FILE.
2. Optical data from array DAT.

**OUTPUTS:** Interpolated refractive index values in array DATA. And data point wavelengths in array LAMDA

**NOTES:**
1. This procedure is called by MODIFY_FILE.
2. This procedure calls procedure INT_FROM_TABLE_FOR_n_and_k.

---

**Procedure Pick_and_Store(count_lamda : real); forward;**

**Procedure Modify_File(count_lamda, Up_lamda, delta: real);**

**PURPOSE:** To simply increment count_lamda from the input starting value up to the upper wavelength range limit (i.e. Up_lamda) and to call procedure PICK_&_STORE between each increment.

**INPUTS:**
1. count_lamda = the starting value which is set in procedure MODIFY_DATA
2. Up_Lamda = the maximum wavelength. Count_lamda will be incremented up to this value.
3. Delta = is the increment added to count_lamda to find the next data point. Delta is set in procedure MODIFY_DATA.

**OUTPUTS:** New count_lamda values for PICK_&_STORE.

**NOTES:**
1. This procedure is called by MODIFY_DATA.
2. This procedure calls PICK_AND_STORE.
forward;

MODIFY_DATA

PURPOSE: To calculate the appropriate data point increment (= delta) for count_lamda.

DESCRIPTION: Using the data input from procedure RANGE (i.e. Up_lamda and Low_lamda) it decides in what area the data points are in (e.g. visible or infrared or both) and then finds the appropriate data point increment (= delta). Finally procedure MODIFY_FILE is called for each layer.

INPUTS: Up_lamda, low_lamda, vis_spacing and ir_spacing from procedure RANGE.

OUTPUTS: Delta = the increment for count_lamda.

NOTES: (1) This program divides the range of interest into the visible (3000-10000 angstroms) and the infrared (10000-50000 angstroms) because the transmission curves of the visible vary quickly (i.e. interference fringes) where as the IR curves do not. This allows the user to see more T/R details on one plot.
(2) This procedure call MODIFY_FILE.
(3) This procedure is called by procedure GET_DATA_FROM_FILES.

Segment Procedure S_Modify_data;

var
delta, waves: real; {waves = a dummy variable used only if plot is entirely in IR, want to set pointer in dat to right position. delta = the increment of count_lamda.}

Begin

{All of the T/R points are in the visible.}

if (low_lamda < 1.0e4) and (up_lamda <= 1.0e4) then begin
  for m := 1 to (tot_num_lay + 1) do begin
    wave := 0.25e4;
    i := 0;
  end

end
j := 0;
delta := (up_lamda - low_lamda) / vis_spacing;
modify_file(low_lamda, up_lamda, delta);
end; {for m}
end; {if in visible}

{All of the T/R points to be calculated in the infrared. }
if (low_lamda >= 1.0e4) and (up_lamda > 1.0e4) then
begin
for m := 1 to (tot_num_lay + 1) do
begin
j := 0;
delta := (up_lamda - low_lamda) / ir_spacing;

{All the data in array DAT will usually have data }
in the visible region. The array pointer will
{start at the smallest wavelength for each layer. }
{this will in general not be the smallest IR }
{wavelength so the pointer must be advanced to }
{right spot in array. }
repeat
j := j + 1;
wave := dat[m, 3, j];
until (wave >= low_lamda);
wave := 0.25e4;
i := 0;
modify_file(low_lamda, up_lamda, delta);
end; {for m}
end; {if in IR}

{The T/R data points are to be calculated in both the }
{visible and infrared regions. Thus two Deltas must be }
{calculated, one for each region. }
if (low_lamda < 1.0e4) and (up_lamda > 1.0e4) then
begin
for m := 1 to (tot_num_lay + 1) do
begin
wave := 0.25e4;
i := 0;
j := 0;
delta := (1.0e4 - low_lamda) / vis_spacing;
modify_file(low_lamda, 1.0e4, delta);
delta := (up_lamda - 1.0e4) / ir_spacing;
modify_file(1.0e4 + delta, up_lamda, delta);
end; {for m}
end; {if in both IR and visible}
end; {modify data}
Segment Procedure S_Get_data_from_file;
var
  go_on : boolean; {Is true as long as end of file (eof) has not been reached.}
  n_val,k_val,wave_val : real; {values read from the floppy DATA:}
  m0,j0,low,m1 : integer; {array and FOR loop dummy index variables.}
  respond2,respond1 : char; {responses to a misspelled layer name.}
begin
  low := 1;
  {Obtain data from a new layer.}
  for m:= 1 to (num_lay_with_no_data + 1) do begin
    {Get all the data for this layer.}
    for m0 := low to (layer_with_own_data[m1] - 1) do begin
      {$1-}
      reset(all_lyrs_dat,concat('Data:', name[m0],'.text'));
      {Filename, name[m0], exists on floppy DATA: so enter optical data.}
      if ioresult = 0 then begin
        {$1+}
        j0:= 0;
        go_on := true;
        while go_on do begin
          readln(all_lyrs_dat,wave_val,n_val,k_val);
          j0:= j0 + 1;
          dat[m0,1,j0] := n_val;
          dat[m0,2,j0] := k_val;
          dat[m0,3,j0] := wave_val;
          {Check to see if first wavelength entered is greater than low_lamda. If this is true then adjust the range.}
          if ((j0=1) and (wave_val > low_lamda)) then low_lamda := wave_val;
          {Check to see if the last wavelength entered is less than Up_lamda. If this is true then adjust range.}
        end
      end
    end
  end
end
if (up_lamda <= wave_val) or (eof(all_lyrs_dat)) then
  go_on := false;
  if eof(all_lyrs_dat) = true then
    up_lamda := wave_val;
  end; {while not eof}
  close(all_lyrs_dat,normal);
end {if ioresul}

{The filename name[m0] is not on disc DATA:.}
{Perhaps the name is misspelled.}
else
  begin
    writeln;
    writeln(' No data exists for a file called ');
    writeln(name[m0]);
    writeln;
    writeln('Do you want to see a list of the files available {y/n} ? ');
    readln(respond1);
    if respond1 = 'y' then
      begin
        reset(store_name,'data:display.text');
        while not eof(store_name) do
          begin
            readln(store_name,str_name);
            writeln(str_name);
          end; {while}
        close(store_name,normal);
      end; {if respond1 = yes}
    writeln;
    writeln(' Do you want to respell this file name {y/n} ? ');
    write(' N will exit you from the program ');
    readln(respond2);
    if respond2 = 'y' then
      begin
        writeln;
        write('enter new name ');
        readln(name[m0]);
        m0 := m0 - 1;
      end {if respond2}
    else
      exit(program);
  end; {if else ioresult}
end; {for m0 loop}
low := layer_with_own_data[m] + 1;
end; {for m loop}
s_modify_data
end; {get_data}
Procedure Int_From_Table_For_n_k;
  var
    n1, n2, k1, k2, slope_n, slope_k, wave1, wave2 : real;
    val_n, val_k : real;
  begin
    n2 := dat[m, 1, j];
    n1 := dat[m, 1, j - 1];
    k1 := dat[m, 2, j - 1];
    k2 := dat[m, 2, j];
    wave1 := dat[m, 3, j - 1];
    wave2 := dat[m, 3, j];
    val_k := (1.0e4 * (k2 - k1));
    val_n := (1.0e4 * (n2 - n1));

    {n and k values are the same for two consecutive values }
    {in DAT. No interpolation is needed in this case. }
    if (abs(val_n) <= 0.01) and (abs(val_k) <= 0.01) then
      begin
        n := n1;
        k := k1;
      end {if }
    else
      begin
        {Two consecutive n values in DAT are the same. Thus }
        {we need only interpolate k values. }
        if (abs(val_n) <= 0.01) and (abs(val_k) >= 0.01) then
          begin
            slope_k := (wave2 - wave1)/(val_k);
            k := k1 + 1e-4 * (count_lambda - wave1)/slope_k;
            n := n1;
          end; {if only n2=n1}

        {Two consecutive k values in DAT are the same. Thus, }
        {only the n values need interpolation. }
        if (abs(val_k) <= 0.01) and (abs(val_n) >= 0.01) then
          begin
            slope_n := (wave2 - wave1)/(val_n);
            n := n1 + 1e-4 * (count_lambda - wave1)/slope_n;
            k := k1;
          end; {if only k2-k1 =0}

        {Interpolation must be done on both n and k values. }
        if (abs(val_k) >= 0.01) and (abs(val_n) >= 0.01) then
          begin
            slope_k := (wave2 - wave1)/(val_k);
            slope_n := (wave2 - wave1)/(val_n);
            k := k1 + 1e-4 * (count_lambda - wave1)/slope_k;
            n := n1 + 1e-4 * (count_lambda - wave1)/slope_n;
          end; {neither are 0}
end; {if else}
end; {interpolate from table for n and k}

Procedure Pick_and_Store;
var
  n, k : real;
begin
  {Step through the wavelengths of layer m until wave > count_lamda. This condition must occur for proper interpolation.}
  if count_lamda > wave then
    begin
      j := j + 1;
      wave := dat[m,3,j];
      until (wave >= count_lamda) or (abs(wave-count_lamda) <= 1e-3);

      {The wavelengths are different so interpolate.}
      if wave > count_lamda then
        begin
          int_from_table_for_n_k(count_lamda,n,k);
          i := i + 1;
          data[m,1,i] := n;
          data[m,2,i] := k;
          lamda[i] := count_lamda;
        end
      {The wavelengths are equal so we have a data point at}
      {at wavelength, wave.}
      else begin i := i + 1;
        data[m,1,i] := dat[m,1,j];
        data[m,2,i] := dat[m,2,j];
        lamda[i] := dat[m,3,j];
      end; {if else }
    end {if else }

  {Count_lamda < wave. So we already have the right conditions for interpolation.}
  else
    begin
      if wave > count_lamda then
        begin
          int_from_table_for_n_k(count_lamda,n,k);
          i := i + 1;
        end
      else begin i := i + 1;
        data[m,1,i] := dat[m,1,j];
        data[m,2,i] := dat[m,2,j];
        lamda[i] := dat[m,3,j];
      end; {if else }
    end {if else }
data[m,1,i] := n;
data[m,2,i] := k;
lamda[i] := count_lamda;
end
else
begin
i := i + 1;
data[m,1,i] := dat[m,1,j];
data[m,2,i] := dat[m,2,j];
lamda[i] := dat[m,3,j];
end; {if else}
end; {if else count_lamda is still < wave}
End; {pick and Store}

Procedure Modify_File;
var
diff : real; {diff is needed because of round off error}
Begin
  diff := up_lamda - count_lamda;
  while (count_lamda <= up_lamda) or (diff <= delta/1000.0) do
    begin
      pick_and_store(count_lamda);
      count_lamda := count_lamda + delta;
      diff := abs(up_lamda - count_lamda);
    end; {while}
End; {modify data file}

Procedure Get_Data_From_File;
Begin
  s_get_data_from_file;
End;{get data from files}
End.
UNIT_CALC_VALUE;
{*****************************************************************************}
{UNIT_CALCULATE_VALUE}
{*****************************************************************************}

PURPOSE: To take the optical data for all the layers, put neatly into array DATA by unit SET_UP_DATA, and then proceed to find all the matrix coefficients of each layer. These layer matrices are then multiplied together to form final characteristic matrices; one on either side of the substrate. From these final matrices the total transmission and reflection (T/R) of electromagnetic radiation (E.M.R.) is found for each data point [6,8].

*****************************************************************************
INTERFACE
{*****************************************************************************}
{TOTAL_TRANSMISSION_AND_REFLECTION}
{*****************************************************************************}

PURPOSE: To call all the procedures in this unit and thus find the total transmission and reflection coefficients for one data point (i.e. one wavelength value from array LAMDA).

DESCRIPTION: First, each layers characteristic matrix is multiplied together to obtain one characteristic matrix for each side of the substrate (M[on incident side] and M[on transmitted side]). Then the total T/R values are calculated based on the following figure.

\[
\begin{align*}
M(\text{inc}) & \quad \text{Substrate} \quad M(\text{trans}) \\
I & \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \\
T(\text{in sub}) & \quad T(\text{in sub})x\text{Decay} \quad x \quad T(\text{out sub}) = T \\
<--<< & \quad <--<< \quad <--<< \quad <--<< \quad <--<< \\
R_1 & = T(\text{out}) \quad x \quad T(\text{in sub})x\text{decay}^2xR_0 \\
R \quad = \quad R_1 \quad + \quad R_2 \\
<--<< & \quad <--<< \quad <--<< \quad <--<< \\
R_2 & \\
I \quad \rightarrow \quad \rightarrow \quad \rightarrow 
\end{align*}
\]
where

\[ I = \text{incident electromagnetic radiation.} \]
\[ T = \text{total transmission of E.M.R through the heterostructure.} \]
\[ R = \text{total reflection of E.M.R. off of the heterostructure.} \]
\[ R = R_1 + R_2 \]
\[ R_1 = \text{reflection off of the substate-"transmitted" thin films interface. Usually a small percentage of} \]
\[ R_2 = \text{reflection off of the incident layers-substrate interface. The major component of} \]
\[ \text{Decay} = \exp(-4\pi k t/\lambda) = \text{how much the wave attenuates as it propagates through the substrate.} \]
\[ \text{Remaining terms are self explanatory from the above picture.} \]

Inputs:
\[ j = \text{an integer index used in array DATA to obtain the jth data point.} \]

Outputs:
\[ \text{tot}_R = \text{the total reflectance of E.M.R., at one wavelength, off of the heterostructure.} \]
\[ \text{tot}_T = \text{the total transmission of E.M.R., at "one" wavelength, through the heterostructure.} \]

Notes:
(1) CALC_AND_STORE_DATA calls this procedure \((\text{vis_spacing} + \text{ir_spacing})\) times.
(2) This procedure calls procedures: SIMPLIFY_TO_1_MATRIX, GET_n_and_k_VALUES, FIND_THEATA, TRANSMITTANCE, REFLECTANCE.

---

Procedure Tot_trans_and_refl(j : integer; var tot_T, tot_R : real);

Implementation

Uses screen_ops,
\{SU cmpIx_math.code\} cmplx_math,
\{SU hyp_trig.code\} hyp_trig,
\{SU enter.code\} enter_data,
\{SU initialize.code\} Initialize_system,
\{SU get_data.code\} set_up_data;

const
\[ \text{nair} = 1.0; \]
kair = 0.0;

type
mat_type4 = packed array[1..5] of m_type; {Array that is used to store the characteristic layer matrices.}

***************
GET_n_and_k_VALUES
***************

PURPOSE: To obtain the wavelength, n and k values for the mth layer and the jth data point in that layer from arrays DATA and LAMDA.

INPUTS: m = an integer index which represents the layer number in array data.
        j = an integer index which represents the jth data point in array data.

OUTPUTS: The jth data point wavelength, n and k values for layer m.

NOTES: This procedure is called by procedures FIND_MATRIX_COEFF, and TOT_TRANS_AND_REFL.

Procedure Get_n_and_k_values(m,j :integer; var wavelength,n,k: real); forward;

***************
FIND_THEATA
***************

PURPOSE: To find the cosine of complex angle \( \hat{\theta} \) which is used in finding the coefficients of each layers characteristic matrix.

DESCRIPTION: Using the technique outlined in reference 2. Start with complex Snell's law:
\[
n_0 \sin(\theta) = \hat{\sin}(\phi) = (n - ik)\sin(\phi)
\]
Assume a complex \( \phi = \hat{\theta} \) and correspondingly a complex \( \sin(\hat{\theta}) \)
\[
\sin(\hat{\theta}) = S' + iS''
\]
From Snells law
\[
S' = \frac{n_0 \sin(\theta) x n}{n^2 + k^2}
\]
\[
S'' = \frac{n_0 \sin(\theta) x k}{n^2 + k^2}
\]
similarly
\[ \cos(\phi^n) = C' + iC'' \]
\[ C' = \frac{P + \sqrt{P^2 + Q^2}}{2} \]
\[ C'' = -\frac{-P + \sqrt{P^2 + Q^2}}{2} \]

where:
\[ P = 1 - S'^2 + S''^2 = C'^2 + C''^2 \]
\[ Q = -2S'S'' = 2C'C'' \]

INPUTS:
- \( m \) = an integer index representing the layer number of array data.
- \( i \) = an integer index representing the \( i \)th data point in array data.

OUTPUTS:
The \( \cos(\phi) \) where \( \phi \) is a complex angle.

NOTES
1. This procedure is called by procedures FIND_MATRIX_COEFF and TOT_TRANS_AND_REFL.

PROCEDURE

Procedure Find_Theata(i, m : integer; var c : c_type); forward;

FINDMATRIXCOEFFICIENTS

PURPOSE: To find the matrix coefficients of each thin film layer, for the \( j \)th data point, and store these values in array matrix[m].

DESCRIPTION: The following equations were used to calculate the matrix coefficients [10]:
The \( j \)th film can be represented by the characteristic matrix

\[
M_i = \begin{bmatrix}
\cos(\delta_i) & i\sin(\delta_i)/U_i \\
-iU_i\sin(\delta_i) & \cos(\delta_i)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
a_1 & a_2 \\
a_3 & a_4
\end{bmatrix}
\]

with
\[
\frac{\cos(\phi_i)}{n_i}, \text{ parallel polarization (P)}
\]
\[
U_i = \begin{cases} 
\frac{x_i \cos(\phi_i)}{S} \\
\frac{\delta_i}{\lambda} = nt - i(kt) = 2\pi(n-ik)t_i\cos(\phi_i)
\end{cases}
\]

Simplifying the above equations gives the coefficients of the \(m\)th matrix:

\[
a_1 = a_4 = \cos(nt)\cosh(kt) + i\sin(nt)\sinh(kt)
\]
\[
a_2 = \frac{(\cos(nt)\sinh(kt) + i\sin(nt)\cosh(kt))}{U}
\]
\[
a_3 = U(\cos(nt)\sinh(kt) + i\sin(nt)\cosh(kt))
\]

**INPUTS:**
- \(j\) = an integer index representing the \(j\)th data point in array data.

**OUTPUTS:**
- The matrix coefficients of each thin film layer for the \(j\)th data point (i.e. matrix\([m]\)).

**NOTES:**
1. This procedure is called by procedure `SIMPLIFY_TO_1_MATRIX`.
2. This procedure calls procedures `CMPLX_DIV`, `CMPLX_MULT`, `GET_N_AND_K`, and `FIND_THEATA`.

---

Segment Procedure `Find_Matrix_Coeffs(j : integer; var matrix: mat_type4)`;

const
- \(\pi = 3.141592654\); var
- \(w, nt, kt : \text{real}\); {\(w\) = variable representing part of eq. \(\delta\) which is defined above.}
- \(nt\) and \(kt\) = represent the real and imaginary parts of \(\delta\).
- \(\text{wavelength, n, k : \text{real}}\); {Optical data variables.}
- \(x : \text{real}\); {dummy variable}
- \(m, l : \text{integer}\); {index variables}
- \(n1, n2, c, temp, U : \text{c\_type}\); {\(C = \text{the cosine of complex } \phi\).}
- \(U = \text{depends on polarization of E.M.R.}; \) {defined above.}

Begin
for \(m := 1\) to \(\text{tot\_num\_lay}\) do begin
begin
get_n_and_k_values\((m, j, \text{wavelength, n, k})\);
With matrix\([m]\) Do
Begin
\(w := (2 * \pi * \text{thick}[m]) / \text{wavelength};\)
\(n1.\text{re} := n;\)
\(n1.\text{im} := -k;\)
find_theata\((j, m, c);\)
cmplx_mult\((c, n1, n2);\)
{Incident E.M.R. is polarized perpendicular to the}
if angle_ans = 's' then
    cmplx_mult(c,n1,U)

else
    cmplx_div(n1,c,U);
    nt := n2.re * w;

l := 0;
repeat
    l := l + 1;
    x := l * 2 * pi;
until x >= nt;
    nt := nt - (x - (2 * pi));
    kt := -n2.im * w;

a1.re := cos(nt) * cosh(kt);
a4.re := a1.re;
temp.re := cos(nt) * sinh(kt);
temp.im := sin(nt) * cosh(kt);
cmplx_divsn(temp,U,a2);
a1.im := sin(nt) * sinh(kt);
a4.im := a1.im;
cmplx_mult(U,temp,a3);
end; {with}
end; {for m}
End; {find_matrix_coeffs}

Procedure Get_n_and_k_values;
Begin
    n := data[m,1,j];
k := data[m,2,j];
wavelength := lamda[j];
End; {get_n & k_values}

Procedure Find_Theata;
var
    s : c_type; {variables defined above.}
p,q,denom,x,diff : real;
l : integer;
Begin

{The following six steps are needed because this operating system can not take the cos/sin/tan of any angle greater than \(2\pi\).

\[
l := 0;
\]
repeat
\[
l := l + 1;
x := 6.283 * l;
\]
until \(x \geq \text{theata}\);
\[
\text{theata} := \text{theata} - (x - 6.283);
\]

{Find the cosine of complex \(\phi\) as defined by the equations in section DESCRIPTION above.}

\[
\text{denom} := \text{sqr(data[m,1,i]) + sqr(data[m,2,i])};
\]
\[
\text{s.re} := (\text{nair} * \text{sin(theata)} * \text{data[m,1,i]})/\text{denom};
\]
\[
\text{s.im} := (\text{nair} * \text{sin(theata)} * \text{data[m,2,i]})/\text{denom};
\]
\[
\text{p} := 1 - \text{sqr(s.re)} + \text{sqr(s.im)};
\]
\[
\text{q} := -2 * \text{s.re} * \text{s.im};
\]
\[
\text{c.re} := \text{sqrt}((\text{p} + \text{sqr(s.re)} + \text{sqr(q)})/2);
\]

{These steps are required because round off error.}

\[
\text{diff} := ((-\text{p} + \text{sqr(s.re)} + \text{sqr(q)})/2);
\]
if \(\text{diff} \leq 0.0001\) then
\[
\text{diff} := 0.0;
\]
\[
\text{c.im} := -\text{sqrt(diff)};
\]
End; {find theata}

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

SIMPLIFY_TO_ONE_MATRIX

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

PURPOSE: To multiply all the complex layer matrices together on each side of the substrate; and return one matrix for each side of the substrate. If there are no thin films on one side of the substrate then the identity matrix is returned for that side.

DESCRIPTION: The complete multilayer thin film structure on "one" side of the substrate can be represented by the product matrix \(M\).

\[
M = M_1M_2...M_l
\]
e.g.

\[
M(inc) = \begin{bmatrix}
    a_{11}^1 & a_{12}^1 & a_{21}^2 & a_{22}^2 \\
    a_{31}^1 & a_{32}^1 & a_{41}^2 & a_{42}^2
\end{bmatrix}
\]
Procedure Smplfy_to_1_Mat(j : integer; var matrix_inc,
matrix_trans : m_type);

var
m : integer;
matrix : mat_type4;
Begin
find_matrix_coeffs(j,matrix);

{No thin films on the substrate side which has the}
{incident E.M.R.}
if num_lay_on_inc = 0 then
begin
with matrix_inc do
begin
a1.re := 1;
a1.im := 0;
a2.re := 0;
a2.im := 0;
a3.re := 0;
a3.im := 0;
a4.re := 1;
a4.im := 0;
end; {if no layers on inc side}
end {if all layers on other side}
else

{Some thin films are on the substrate side which has } 
{the E.M.R. Simplify to one matrix.}
begin
for m := 1 to (num_lay_on_inc - 1) do
mult_matrix(matrix[m],matrix[m+1],matrix[m+1]);
with matrix_inc do
begin
a1.re := matrix[num_lay_on_inc].a1.re;
a1.im := matrix[num_lay_on_inc].a1.im;
a2.re := matrix[num_lay_on_inc].a2.re;
a2.im := matrix[num_lay_on_inc].a2.im;
a3.re := matrix[num_lay_on_inc].a3.re;
a3.im := matrix[num_lay_on_inc].a3.im;
a4.re := matrix[num_lay_on_inc].a4.re;
a4.im := matrix[num_lay_on_inc].a4.im;
end; {with}
end; {if else}

{No thin films on the substrate side which does "not"
have the incident E.M.R.}

if num_lay_on_inc = tot_num_lay then
begin
with matrix_trans do
begin
a1.re := 1;
a1.im := 0;
a2.re := 0;
a2.im := 0;
a3.re := 0;
a3.im := 0;
a4.re := 1;
a4.im := 0;
end; {if all layers on inc side}
end {if}

{Some thin films are on both sides of the substrate.}
{Simplify the matrices on the substrate side which
does not have the incident E.M.R. to one matrix.}

else
begin
for m := (num_lay_on_inc + 1) to (tot_num_lay - 1) do
  mult_matrix(matrix[m],matrix[m+1],matrix[m+1]);
with matrix_trans do
begin
a1.re := matrix[tot_num_lay].a1.re;
a1.im := matrix[tot_num_lay].a1.im;
a2.re := matrix[tot_num_lay].a2.re;
a2.im := matrix[tot_num_lay].a2.im;
a3.re := matrix[tot_num_lay].a3.re;
a3.im := matrix[tot_num_lay].a3.im;
a4.re := matrix[tot_num_lay].a4.re;
a4.im := matrix[tot_num_lay].a4.im;
end; {with}
end; {if else}
end; {simplify to one matrix}
PURPOSE: To use the final characteristic product matrices and the reflectance formula below to calculate the total reflectance of E.M.R off of the heterostructure at one wavelength.

DESCRIPTION: The following formulas were used [1]

\[
\begin{align*}
  r &= \frac{n_0 \cos(\theta)(a_1 + a_2 n_0 \cos(\phi)) - a_3 - a_4 n_0 \cos(\phi)}{n_0 \cos(\theta)(a_1 + a_2 n_0 \cos(\phi)) + a_3 + a_4 n_0 \cos(\phi)} \\
  R &= |r|^2
\end{align*}
\]

INPUTS:
- \( W \): The final product matrix for one side of the substrate.
- \( pf = n_0 \cos(\theta) \): for the first layer (either air or substrate), if it's the substrate \( \theta \) will be complex.
- \( pl = n \cos(\phi) \): for the last layer (either air or substrate depending on the direction of the incident E.M.R. \( \phi \) may be complex.

OUTPUTS:
- \( \text{reflec} \): is the total reflectance of E.M.R. at one wavelength off of the heterostructure.

NOTES:
1. This procedure is called by TOT_TRAN&_REF
2. This procedure calls the basic complex arithmetic procedures.

```
Procedure Reflectance(w: m_type; pf, pl: c_type; var reflec: real);

var
  temp1, temp2, numerator, denom, refl: c_type;

Begin
  With w Do
    Begin
      cmplx_mult(a2, pl, temp1);
      cmplx_add(a1, temp1, temp1);
      cmplx_mult(pf, temp1, temp2);
      cmplx_sub(temp2, a3, numerator);
      cmplx_mult(a4, pl, temp1);
      cmplx_sub(numerator, temp1, numerator);
      cmplx_add(temp2, a3, denom);
      cmplx_add(denom, temp1, denom);
      cmplx_divsn(numerator, denom, refl);
      reflec := sqr(refl.re) + sqr(refl.im);
    End; {with}
End; {reflectance}
```
PURPOSE: To use the product layer matrices and the formulas listed below to calculate the total transmission of E.M.R. through the heterostructure for one wavelength.

DESCRIPTION: The following formulas were used in the calculation of $T$ [10]:

$$t = \frac{2n_0 \cos(\theta)}{n_0 \cos(\theta)} \left( a_1 + a_2 \frac{n_0 \cos(\phi)}{n_0 \cos(\phi)} \right) + a_3 + a_4 \frac{n_0 \cos(\phi)}{n_0 \cos(\phi)}$$

$$T = \frac{\cos(\phi)}{\cos(\phi)} |t|^2$$

INPUTS: $W, pf, pl$ = see procedure reflectance for a verbal description.

OUTPUTS: Transmitt = the total transmission through the layers of interest. E.g. air/product matrix/sub outputs the transmission INTO the substrate.

NOTES: This procedure is called by TOT_TRANS_AND_REF1

---

```
Procedure Transmittance(w: m_type; pf, pl: c_type; var transmit: real);
var
temp1, temp2, temp3, numerator, denom, tra: c_type;
Begin
With w Do
Begin
  Complex_add(pf, pf, numerator);
  Complex_mult(a2, pl, temp1);
  Complex_add(a1, temp1, temp2);
  Complex_mult(a4, pl, temp1);
  Complex_add(a3, temp1, denom);
  Complex_mult(pf, temp2, temp1);
  Complex_add(temp1, denom, denom);
  Complex_divsn(numerator, denom, tra);
  transmit := (sqrt(tra.re) + sqrt(tra.im))*(pl.re)/pf.re;
End; {with}
End; {transmittance}
```

Procedure Tot_Trans_and_Refl; const
pi = 3.141592654;

var
matrix_inc,matrix_trans : m_type; {The final product matrices for either side of the substrate.}
{See the figure in section DESCRIPTION above.}

n_sub,k_sub : real; {The refractive index of the substrate.}
wavelength : real; {The absorption coefficient defined by \[ \alpha = \frac{4\pi k}{\lambda} \]}
alpha : real; {The major component of \{R; defined as \[ R \] in the figure that is in section DESCRIPTION.}

major_ref : real; {Defined in section DESCRIPTION above.}
trans_out_sub,trans_in_sub,decay : real; {Defined in section DESCRIPTION above.}
p1,p3,c : c_type; {Defined in section DESCRIPTION above.}

Begin
simplify_to_1_mat(j,matrix_inc,matrix_trans);
get_n_and_k_values(tot_num_lay + 1,j,wavelength,n_sub,k_sub);
p1.re:= nair * cos(theata);
p1.im:= -kair * cos(theata);
find_theata(j,tot_num_lay + 1,c);
p3.re:= n_sub * c.re;
p3.im:= -k_sub * c.im;
transmittance(matrix_inc,p1,p3,trans_in_sub);
transmittance(matrix_trans,p3,p1,trans_out_sub);
alpha := 4 * pi * k_sub/wavelength;
alpha := -alpha * thick[tot_num_lay + 1]/c;
if alpha < -9 then
  decay := 0
else
  decay := exp(alpha);
trans_in_sub := trans_in_sub * decay;
tot_T := trans_in_sub * trans_out_sub;
reflectance(matrix_trans,p3,p1,tot_R);
tot_R := tot_R * decay * trans_in_sub;
transmittance(matrix_inc,p3,p1,trans_out_sub);
tot_R := tot_R * trans_out_sub;
reflectance(matrix_inc,p1,p3,major_Ref);
tot_R := tot_R + major_ref;
end; {total E.M.R transmitted & Reflected}

End.
PROGRAM HETEROSTRUCTURE;

PURPOSE: To calculate the transmission/reflection of electromagnetic radiation, for a range of wavelengths (i.e. visible and/or IR) through/off of some defined heterostructure. This is accomplished by the main program which calls procedures:

START = user enters all important optical data about a heterostructure.
GET_DATA_FROM_FILES = The refractive index data for each layer is loaded into the program.
CALC_AND_STORE = The T/R curves are calculated for a wavelength range and stored on disc DATA:

PLOT_RESULT = these T/R curves are plotted on the terminal.
The above heterostructure data can then be altered and the above procedures repeated.

uses screen_ops,
  [SU graphic.library] plot_graphics,
  [SU cmplx_math.code] cmplx_math,
  [SU hyp_trig.code] hyp_trig,
  [SU enter.code] enter_data,
  [SU initialize.code] Initialize,
  [SU plot_rslt.code] plot_the_result,
  [SU calc.code] calc_value,
  [SU get_data.code] set_up_data;

var
  a_opt_thick : mat_type1; {Stores the thicknesses of all the layers which give T/R curves closest to max_val.}
  b_opt_thick : mat_type1; {Stores all the layer thicknesses which maximize the T/R curves.}
  c_opt_thick : mat_type1; {Stores all the layer thicknesses which minimize the T/R curves.}
  que : integer; {Is defined in procedure PLOT_RESULT.}

{************************************************************
Procedure Integrate(low_lamda, up_lamda : real; spacing, low : integer; f : mat_type2; var area: real); forward;

**STORE_DATA**

Purpose: To store the T/R/absorption (A) data on the floppy DATA; this is later retrieved by PLOT_RESULT.
{ DESCRIPTION: The maximum, minimum and max_val T data is 
stored in DATA: by different filenames. } 
{ Depending on where this procedure is called in 
CALC_AND_STORE_DATA a different append 
letter is passed to this procedure (e.g. for 
maximum T append = a and the filename = 
DATA:hetroTa ). Similarly for R and A. These 
filenames are then created on disc DATA: 
(i.e. rewrite(file1,DATA:hetroTa)) and the data 
is entered. Then the file is closed, to save 
this data. }

{ INPUTS: (1) j = is used in char(j) to pick out a 
character which appends to a filename so 
that the max, min and max_val T/R/A curves 
are filed under different filenames. }
{ (2) count = the total number of data points, it 
is used as a counter. }
{ (3) T,R,A = are the total transmission, 
reflection and absorption values for one 
set of layer thicknesses. }

{ OUTPUTS: The T/R/A data onto disc DATA: }

{ NOTES: (1) This procedure is called by procedure 
CALC_AND_STORE_DATA. }

Purpose: To take the generated transmission and 
reflection data, from unit CALC_VALUE, and 
either (1) store it for a plot routine (this 
case occurs when the user wants the T/R curves 
calculated for only one thickness of each thin 
film) or (2) finds the optimum T/R curves. The 
optimum T or R curves are found by integrating 
the T or R data for one set of layer 
thicknesses and then systematically varying 
each layer thickness and again integrating 
these new T/R curves; This iterative technique 
is repeated until all the possible layer 
thicknesses are tried and the optimum T/R 
curves (i.e. maximum, minimum and max_val 
areas) are found.
DESCRIPTION: The user, in procedure START, either
specified a range of thicknesses for each
layer or only one thickness for each layer of
the heterostructure. If the user wanted the T/R
data calculated for only one thickness of each
layer, then the REPEAT-UNTIL loops of this
procedure are only gone through once and the
T/R data is stored immediately. If however, the
user wants the max. and min. T/R (and possibly
to see of a certain value of T/R can be
obtained = max_val) found for the
heterostructure then the range of thicknesses
entered in procedure START are systematically
tried. For each new thickness the T/R data is
calculated and integrated; it is then compared
to see if it is the max/min/max_val, if so it
is stored, along with the appropriate optimum
thicknesses on floppy DATA: for procedure
PLOT_RESULT.

INPUTS:  
(1) The transmission and reflection data,
labeled tot_T and tot_R respectively.

OUTPUTS:  The optimum set of layer thicknesses which
maximize & minimize & max_val the T/R curves

NOTES:  
(1) This procedure calls procedures STORE_DATA
and TOT_TRANS_AND_REFL.
(2) This procedure is called by MAIN program.

Segment Procedure Calc_and_Store_data;
var
A,Ref,Tra : mat_type2;  {Arrays which contain the complete}
{absorption, reflection and
{transmission data for the
{wavelength range of interest,
{for "only" one set of
{thicknesses.
{
i,j  : integer;  {Array and FOR loop index
{variables.
{
count : integer;  {count = number of data points
|= vis_spacing + ir_spacing + 1
{All integers from 1 to count
{represent a wavelength data
{point.
{
tot_T,tot_R : real;  {Are the total transmission and
{reflection respectively; for
{one data point. See procedure
{TOT_TRAN_AND_REFL.
area : real; {the area of arrays TRA or REF.}
max, min, diff : real; {max = the maximum area obtained
by integration of TRA or REF curves. Used as an indicator to
compare each successive
calculation of TRA or REF and
is set to the largest value.
min = the minimum area
obtained by integration of TRA
or REF curves. Uses same
technique as max above.
diff = is used to compare the
area of TRA or REF curves with
the user desired area (max_val)
and try to get the curve closest
to this desired value =
diff - area.
}
value : real; {tolerance used in deciding if
any of the TRA or REF curve
areas are close enough to
max_val; must be with in +/- 10%
}
s_space : real; { }
inc : mat_type1; {Array which contains the amount
each thin film layer thickness
is incremented. This array is
set to zero if only one
thickness for each thin film is
entered.
}

Begin
que := 1;
max := 0.0;
min := 120e4;

{Calculate the tolerance to which T/R curves areas must}
{be before these curves will be stored on disc DATA: .}
{Note that if none of TRA and REF areas are equal to }
{max_val +/- tolerance then a message is displayed on the }
{screen. See procedure PLOT_RESULT.}

value := (max_val + 0.1 * max_val) - (max_val - 0.1 *
max_val);

{A range of thicknesses has been entered; calculate the}
{thickness increment for each thin film.
}
if respon1 = 'n' then
begin
for i := 1 to tot_num_lay do
   inc[i] := (thick_up[i] - thick_low[i])/points;
end; {if respon1}
{Only one thickness for each thin film has been entered.}
{Set array increment = 0 so all of the following}
{REPEAT-UNTIL loops are skipped.}

if respon1 = 'y' then
begin
  for j := 1 to tot_num_lay do
  begin
    thick_low[j] := thick[j];
    thick_up[j] := thick[j];
    inc[j] := 0.0;
  end; {for j}
end; {if respon1}
thick_low[tot_num_lay + 1] := thick[tot_num_lay + 1];
thick_up[tot_num_lay + 1] := thick[tot_num_lay + 1];
inct[tot_num_lay + 1] := 0;
for j := (tot_num_lay + 2) to 10 do
begin
  thick_low[j] := 0;
  thick_up[j] := 0;
  inc[j] := 0;
end; {for j}
repeat
repeat
repeat
repeat
repeat
repeat
repeat
repeat
repeat
repeat
count := (vis_spacing+ir_spacing+1);
for i := 1 to count do
begin
  tot_trans_and_refl(i,tot_T,tot_R);
  Tra[i] := tot_T;
  Ref[i] := tot_R;
end; {for i}

{A range of thicknesses has been entered so find the max, min and max_val areas.}
if respon1 = 'n' then
begin
  if (respon5 = 't') and (up_lamda > 1.0e4) then
    integrate(low_lamda, 1e4, vis_spacing + 1, 1, Tra, area);
  if (respon5 = 't') and (up_lamda <= 1.0e4) then
    integrate(low_lamda, up_lamda, vis_spacing+1,1,tra,area);
  if (respon5 = 'r') and (low_lamda > 1e4) then
    integrate(low_lamda, up_lamda, ir_spacing,1,ref,area);
  if (respon5 = 'r') and (low_lamda <= 1e4) then
    integrate(1e4,up_lamda, ir_spacing, vis_spacing + 1, ref,area);
{Compare TRA or REF area with max_val area.}
if ((max_val-0.1*max_val) < area) and(area<(max_val+0.1*max_val)) then
begin
  diff := abs(max_val - area);
  if diff < value then
    begin
      que:= 0;
      value := diff;
      store_data(count, 65, Tra, Ref);
      for i := 1 to tot_num_lay do
        a_opt_thick[i]:=thick[i];
    end; {if area in range}
{Is this TRA or REF curve area the maximum area? if so store it on DATA: along with the corresponding layer thicknesses.}
if area > max then
begin
  max := area;
store_data(count, 66, Tra, Ref);
for i := 1 to tot_num_lay do
  b_opt_thick[i] := thick[i];
end; {if area > max}

{Is this TRA or REF curve area}
{the minimum area? If yes}
{then store it on DATA: along}
{with the corresponding}
{layer thicknesses}
if area < min then
  begin
    store_data(count, 67, Tra, Ref);
    min := area;
    for i := 1 to tot_num_lay do
      c_opt_thick[i] := thick[i];
  end; {if area < min}
end {if response = 'n'}
else
  begin
    store_data(count, 65, Tra, Ref);
  end; {if else response 1}
  until (abs(thick[1] - thick_up[1]) < 1e-3) or
        (thick[1] >= thick_up[1]) and (tot_num_lay - 1 >= 0);
  until (abs(thick[2] - thick_up[2]) < 1e-3) or
        (thick[2] >= thick_up[2]) and (tot_num_lay - 2 >= 0);
  until (abs(thick[3] - thick_up[3]) < 1e-3) or
        (thick[3] >= thick_up[3]) and
        (tot_num_lay - 3 >= 0);
  until (abs(thick[4] - thick_up[4]) < 1e-3) or
        (thick[4] >= thick_up[4]) and
        (tot_num_lay - 4 >= 0);
  until (abs(thick[5] - thick_up[5]) < 1e-3) or
        (thick[5] >= thick_up[5]) and
        (tot_num_lay - 5 >= 0);
  until (abs(thick[6] - thick_up[6]) < 1e-3) or
        (thick[6] >= thick_up[6]) and
        (tot_num_lay - 6 >= 0);
  until (abs(thick[7] - thick_up[7]) < 1e-3) or
        (thick[7] >= thick_up[7]) and
        (tot_num_lay - 7 >= 0);
  until (abs(thick[8] - thick_up[8]) < 1e-3) or
        (thick[8] >= thick_up[8]) and (tot_num_lay - 8 >= 0);
  until (abs(thick[9] - thick_up[9]) < 1e-3) or
        (thick[9] >= thick_up[9]) and (tot_num_lay - 9 >= 0);
  until (abs(thick[10] - thick_up[10]) < 1e-3) or
        (thick[10] >= thick_up[10]) and (tot_num_lay -
\[ 10 \geq 0; \]

end; \{calculate & store data\}

Procedure Integrate;
  var
    i : integer; \{Index for FOR loop.\}
    h : real; \{Width of trapezoid.\}
  Begin
    h := (up_lamda - low_lamda)/spacing;
    area := 0.5 * h * (f[low] + f[spacing]);
    for i := (low + 1) to (spacing - 1) do
      area := area + (h * f[i]);
  End; \{integrate\}

Procedure Store_Data;
  var
    append : string[9];
    A : mat_type2;
    i : integer;
    file1, file2, file3 : file of graph_data;
    r_plot_filename, t_plot_filename, a_plot_filename : string;
    curr_data : graph_data;
  Begin
    append := ' .data[2]';
    append[1] := chr(j);
    r_plot_filename := concat('data:hetroR', append);
    rewrite(file1, r_plot_filename);
    t_plot_filename := concat('data:hetroT', append);
    rewrite(file2, t_plot_filename);
    a_plot_filename := concat('data:hetroA', append);
    rewrite(file3, a_plot_filename);
    for i := 1 to count do
      begin
        curr_data[gr_x_data] := lamda[i] * 1e-4;
        curr_data[gr_y_data] := R[i];
        file1a := curr_data;
        put(file1);
        curr_data[gr_y_data] := T[i];
        file2a := curr_data;
        put(file2);
        A[i] := 1 - R[i] - T[i];
        curr_data[gr_y_data] := A[i];
        file3a := curr_data;
        put(file3);
      end; \{for i\}
    close(file1, lock);
    close(file2, lock);
    close(file3, lock);
  End; \{store data on floppy\}

\{MAIN PROGRAM uses all the procedures defined previously to\}
\{calculate the T/R data and plot it one the screen. It also\}
\{allows the user to change any of the existing data and \}
{rerun the program.}

Begin {main program}
  introduction;
  start;
  get_data_from_files;
  calc_and_store_data;
  plot_result(low_lamda, up_lamda, respon1, a_opt_thick, b_opt_thick, c_opt_thick, que);
  repeat
    sc_clr_screen;
    gotoxy(0,0);
    writeln('Do you want to alter any of the currently existing data and ');
    write('rerun the program {y/n} ? ; n will exit you from the program. ');
    readln(respond);
    if respond = 'y' then
      begin
        repeat
          alter_data(ans);
          until ans in ['n'];
        sc_clr_screen;
        writeln('Thinking ?! .... ');  
        get_data_from_files;
        calc_and_store_data;
        plot_result(low_lamda, up_lamda, respon1, a_opt_thick, b_opt_thick, c_opt_thick, que);
      end; {if respond}
    until respond = 'n';
End. {main}
UNIT PLOT_The_RESULT;

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

UNIT PLOT_RESULTS

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

PURPOSE: To plot the results of T/R/A data generated by the program heterostructure.

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

INTERFACE

uses screen_ops,
{SU enter.code} enter_data,
{SU initialize.code} Initialize;

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

PLOT_RESULT

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

PURPOSE: To display a plot, on the terminal, of the transmission and/or reflection and/or absorption characteristics of the user defined heterostructure. In addition, this procedure also performs elementary computer graphics and allows the user to zoom in on any part of the plot, scale the plot appropriately etc. before getting a hardcopy of the plot. Note that if a range of thicknesses for each layer had been entered in procedure START then the user can see a T or R plot of:

1) the curve which is closest to max_val. That is if any of the curves come to within +/- 10%.

2) the curve which has maximum T {or R} of E.M.R. through {off of} the heterostructure.

3) the curve which has minimum T {or R} of E.M.R. through {off of} the heterostructure. Also, the corresponding thickness of all layers be displayed for all three cases above.

DESCRIPTION: Using the plotting function written by Brian Sullivan of Department of Physics U.B.C. this plot procedure works as follows:

- the main program calls this procedure after all the T/R/A data has been generated and stored on disc DATA;

- If only one thickness for each layer of the structure was used on the calculations then
upon entering this procedure a T and R plot is displayed with the x-axis $\epsilon_{\text{low}_\text{lamda}, \text{up}_\text{lamda}}$ and y-axis $\epsilon\{0,1\}$. 

- If a range of thicknesses for each thin film had been entered; then upon entering this procedure a statement is displayed asking the user whether he wants to see a maximum, minimum or max_value T {or R} plot. The corresponding layer thicknesses to his response are displayed and then the appropriate curve plotted with the x-axis $\epsilon_{\text{low}_\text{lamda}, \text{up}_\text{lamda}}$ and y-axis $\epsilon\{0,1\}$.

- the following prompt line is then displayed

- The following prompt line is then displayed

Q)uit P)lot X)setting Y)setting Var thickness
with

Q)uit = exits the user from this procedure and goes back into the main program where he can alter and of the data and rerun this program.

P(lot = allows the user to select from a menu what he wants plotted. Then with the current x-axis and y-axis settings the appropriate curve[s] is plotted.

X(settings = displays a prompt line which allows the user to change the range and divisions on x-axis.

Y(settings = similar to x(settings.

V(ariable thicknesses = if a range of thicknesses was entered then the max, minimum, or max_val T{or R} plots can be viewed along with the corresponding thicknesses. But if the user enters this option and only one thickness for each layer was used in the calculations then a message is displayed telling him that this option was not meant for this case.

INPUTS:

(1) $\text{Low}_\text{lamda}$ and $\text{up}_\text{lamda}$ = are used to initially set the axis bounds on the display of data on the screen.

(2) a/b/c_opt_thick = are used only if a range are used.

(3) a_opt_thick= is the optimum thicknesses of the thin film layers which give a T {or R} curve close to max_val.

(4) b_opt_thick = is the thicknesses of the films for which the T/R curve is a maximum (a maximum to within the range of thicknesses give; this may not be the absolute maximum of the heterostructure)

(5) c_opt_thick = is the thicknesses of the films for which the T/R curve is a minimum.
Again this may not be the global minimum of this heterostructure.

(6) que = is set to zero of the value the user wanted to obtain for T/R can be found to within +/- 10%. If que is zero then a_opt_thick will be printed out. If que = 1 then a message is displayed informing the operator that for the range of thicknesses entered the T/R curve desired could not be achieved.

OUTPUTS: A plot of T and/or R and/or A curves for a particular multilayered thin film structure.

NOTES: (1)This procedure is called by the MAIN program

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

Procedure plot_result(low_lamda,up_lamda : real; respon1 : char; a_opt_thick,b_opt_thick,c_opt_thick: mat_type1;que:integer);

IMPLEMENTATION
USES {SU GRAPHIC.LIBRARY} PLOT_GRAPHICS;

Var
t_plot,r_plot,a_plot : string;
min_x,max_x,x_div : real; {Variables used in setting x-axis bounds and scale.}
min_y,max_y,y_div : real; {Variables used in setting y-axis bounds and scale.}

{Plot transmission curves points with a "cross" symbol.}

Procedure Plot_T; begin
  gph_plot(t_plot,gr_sym_cross);
  gph_plot(t_plot,gr_sym_vector);
end; {plot transmission}

{Plot reflection curves point with a "square" symbol.}

Procedure Plot_R; begin
  gph_plot(r_plot,gr_sym_square);
  gph_plot(r_plot,gr_sym_vector);
end; {plot reflection}

{Plot absorption curves points with a "circle" symbol.}

Procedure Plot_A; begin
  gph_plot(a_plot,gr_sym_circle);
  gph_plot(a_plot,gr_sym_vector);
end; {plot absorption}
{clear screen and draw axis with appropriate scaling, } 
{divisions and bounds. }

Procedure Set_up_graph;
begin
  gph_scale(min_x,max_x,min_y,max_y);
  scClr_screen;
  gph_x_axis(x_div,2,5,2,'Wavelength (microns)');
  gph_y_axis(y_div,2,4,2,' ');
end;

Procedure Plot_Result;
var
  i : integer;
  append : string[6];
  ch,respond ,ch1,ch2,ch3: char;
  ans : integer;
  chset1,chset2 : set of 'A'..'Z';
  quitting,stopping : boolean;
Begin
  up_lamda := up_lamda * 1e-4;
  low_lamda := low_lamda * 1e-4;
  quitting := false;
  append:=' .data';
  chset1 := ['P','X','Y','Q','V'];
  chset2 := ['S','M','L','D'];
  scClr_screen;
  gotoxy(0,0);

  {The next 80 lines of program are used to initially } 
  {display a plot; see section DESCRIPTION above. } 

  {$I-}
  if respond1 = 'y' then
    append[1] := chr(65)

  {A range of thicknesses has been used. What optimized } 
  {plot does the user want to see. }
else
  begin
    repeat
      writeln('Which plot do you want to see ? type in ');
      writeln(' max. T{or R} that requested a');
      writeln(' max. possible T{or R} in thickness range b');
      writeln(' min. possible T{or R} in thickness range c');
      readln(append[1]); if (que = 1) and (append[1] = 'a') then
        begin
          writeln;
          writeln('The max. T{ or R} that you wanted to achieve does not');
    end;
writeln('exist for the range of thicknesses entered; at least');
writeln('to within +/- 10%');
writeln;
end {if}
else
begin
  que := 0;
sclr_screen;
gotoxy(0,0);

  {Display the optimum thicknesses of each layer.}

  writeln('For this plot: ');
case append[1] of
    'a' : begin
      for i := 1 to tot_num_lay do
        begin
          write('The optimum thickness of ' ,name[i],' is ');
          writeln(a_opt_thick[i]);
        end; {for i}
    end; {a}
    'b' : begin
      for i := 1 to tot_num_lay do
        begin
          write('The optimum thickness of ' ,name[i],' is ');
          writeln(b_opt_thick[i]);
        end;
    end; {b}
    'c' : begin
      for i := 1 to tot_num_lay do
        begin
          write('The optimum thickness of ' ,name[i],' is ');
          writeln(c_opt_thick[i]);
        end; {for i}
    end; {c}
  end; {case}
end; {if else response 1}
t_plot := concat('Data:hetroT',append);
r_plot := concat('Data:hetroR',append);
a_plot := concat('Data:hetroA',append);
max_x := up_lamda;
if low_lamda <= 0.5 then
  min_x := 0.0
else
  min_x := trunc(low_lamda);
if up_lamda > 2.0 then
begin
    min_y := 0.0;
    max_y := 1.0;
    x_div := max_x - min_x;
    y_div := 10.0;
end
else
begin
    min_y := 0.5;
    max_y := 1.0;
    x_div := 10.0;
    y_div := 10.0;
end; {if else}
set_up_graph;
plot_tr;
plot_re;
read(ch);
gotoxy(0,0);
repeat
    write('Do you want to replot this data on a different scale {Y/N} ? ');
    readln(respond);
    until respond in [ 'y','n'];
if respond = 'y' then
begin
    while not quitting do
    begin
        ch1 := sc_prompt('Parameters: Q(uit) P(lot
          X(settings) Y(settings) V(ariation) thicknesses', -1,0,0,0,chset1,false,'
            );
        sc_clr_screen;
        case ch1 of
            'Q' : quitting := true;
            'P' : begin
            gotoxy(0,0);
            write('Available for plotting is the transmission(T),
                reflection(R),
                and absorption(A) of the heterostructure. ');
            writeln('combination type in ');
            writeln(' T  1 ');
            writeln(' R  2 ');
            writeln(' A  3 ');
            writeln(' T & R  4 ');
            writeln(' T & A  5 ');
            writeln(' R & A  6 ');
            writeln(' T & R & A  7 ');
            writeln;
            write('T - cross, R- square, A- circle ');
            readln(ans);
            until ans in [1..7];
case ans of
  1 : begin
    set_up_graph;
    plot_tr;
    end;
  2 : begin
    set_up_graph;
    plot_re;
    end;
  3 : begin
    set_up_graph;
    plot_ab;
    end;
  4 : begin
    set_up_graph;
    plot_tr;
    plot_re;
    end;
  5 : begin
    set_up_graph;
    plot_tr;
    plot_ab;
    end;
  6 : begin
    set_up_graph;
    plot_re;
    plot_ab;
    end;
  7 : begin
    set_up_graph;
    plot_tr;
    plot_re;
    plot_ab;
    end;
end; {case}
end; {plot}

'X' : begin
  stopping := false;
  while not stopping do
    begin
      ch2 := sc_prompt('X Parameters: S(top
        M(ax L[owest D(iv'), -1, 0, 0, 0,
        chset2, false,' ')
      case ch2 of
        'S' : stopping := true;
        'M' : begin
          writeln;
          write('Enter the maximum
            wavelength(in angstroms)' );
          readln(max_x);
          max_x := max_x * 1.0e-4;
          sc_clr_screen;
          end; {M}
        'L' : begin

writeln;
write('Enter the minimum wavelength (in angstroms)');
readln(min_x);
min_x := min_x * 1.0e-4;
sc_clr_screen;
end; {L}
'D' : begin
writeln;
write('Enter the desired number of wavelengths');
write(' divisions');
readln(x_div);
sc_clr_screen;
end; {D}
end; {case ch2}
end; {while}
end; {case y}
'V' : begin
if respon1 = 'n' then
    begin
        repeat
            write('Which plot do you want to see ? ');
            writeln(' type in ');
            write(' max T{or R} that you requested ');
            writeln(' a ');
            write(' max. possible T{or R} in thickness range ');
            writeln(' b ');
            write(' min. possible T{or R} in thickness range ');
            writeln(' c ');
            readln(append[1]);
        if (que =1) and (append[1] = 'a') then
            begin
                write('The max. T{or R} that you wanted to achieve ');
                writeln(' does not exist; ');
                write('at least to within +/-10% of this value, for ');
                writeln(' the range of thicknesses entered. ');
            writeln;
            end {if}
        else
            begin
                que := 0;
                sc_clr_screen;
                gotoxy(0,0);
                writeln('For plot requested: ');
                case append[1] of
                    'a' : begin
                        for i := 1 to tot_num_lay do
                            begin
                                write('The optimum thickness of ');
                                write(name[i]);
                                writeln(' is ', a_opt_thick[i]);
                            end; {for i}
                    'b' : begin
                        for i := 1 to tot_num_lay do
                            begin
                                write('The optimum thickness of ');
                                write(name[i]);
                                writeln(' is ', b_opt_thick[i]);
                            end; {for i}
120

end; {b}
'c': begin
for i := 1 to tot_num_lay do
begin
write('The optimum thickness of ');
write(name[i]);
writeln(' is ',
c_opt_thick[i]);
end; {for i}
end; {c}
end; {case}
end; {if else}
until (append[1] in ['a','b','c']) and
(que <> 1);
que := 1;
t_plot := concat('Data:hetroT',
append);
r_plot := concat('Data:hetroR',
append);
a_plot := concat('Data:hetroA',
append);
read(ch);
end {if response is no}
else
begin
write('This option is meant to be used
only if you ');
writeln('entered a range of ');
writeln('thicknesses for each thin
film.');
read(ch);
end; {if else response 1}
end; {case V}
end; {case ch1}
end; {while ch1}
end; {if}
{$I+} End; {plot result}

End.