

N69-37714
NASA-GR-105926
~~dry-08819~~

Technical Note BN-572

October 1968

COMPUTER PROGRAMS FOR REDUCTION OF
MICROPHOTOMETER DATA

by

R. A. Bell, Roger D. Bengtson, D. R. Branch,
D. M. Gottlieb, Randy Roig

UNIVERSITY OF MARYLAND
College Park, Maryland

CASE FILE
COPY



ASTRONOMY PROGRAM
UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND

Technical Note BN-572

October 1968

COMPUTER PROGRAMS FOR REDUCTION OF
MICROPHOTOMETER DATA *

by

R. A. Bell, Roger D. Bengtson, D. R. Branch,
D. M. Gottlieb, Randy Roig

University of Maryland
College Park, Maryland

*This work was supported in part by NASA Grant NsG-359, by NSF Grants GP5963, GP7342 and GP8698, and by NASA Grant NsG-398 to the University of Maryland Computer Science Center.

I. Introduction

During the past two or three years, the users of the Joyce-Loebl digital microphotometer at the University of Maryland have written a number of programs to reduce the data obtained from this instrument. Whilst the aims of the Astronomy group and the Fluid Dynamics group are similar in many respects the programs which each group has developed are quite different. Only the first program, which reads the tape written by the microphotometer, is common to both groups. Before discussing the programs it is desirable to discuss the details of the microphotometer and magnetic tape recorder since they have played an important role in the development of the programs.

The microphotometer is a Joyce-Loebl Mark III CS instrument which has been modified by National Instrument Laboratories so that the information on the transmission of light through a specimen may be written on magnetic tape. The magnetic tape recorder was supplied by the Digi-Data Corporation. During the time which the microphotometer has been at the University of Maryland a number of attachments have been added and the current system is a very versatile one.

The microphotometer is of the double beam type. A change in the transmission of light through the specimen is balanced by a compensating movement of an optical wedge. In the conventional instrument a pen is attached to the wedge and draws a record of the wedge position on the graph paper placed on the recording table. The motion of the wedge is accompanied by a rotation of a shaft. In the digital version an encoder is mounted on this

shaft and the wedge movement is determined from the rotation of the encoder. Full scale deflection of the wedge corresponds to a rotation of 1.74 turns of the encoder and thus, since the encoder has a resolution of a hundredth of a turn, the encoder reading, and consequently the number written on magnetic tape, will lie between 1 and 174 .

The main difference between the conventional and digital forms of the microphotometer concerns the driving of the specimen table. In the conventional form of the instrument the specimen table is linked to the recording table by means of a ratio arm and it is the recording table which is motor driven. The use of different pivot points for the ratio arm allows for different magnifications on the tracings. In some early work, for example, a ratio of $50/1$ was used whereby the movement of the recording table over its full travel of 10 inches moved the specimen a distance of 0.2 inches. After the recording table has moved through its full travel, the linkage between the two tables is unclamped and the recording table reversed to its starting point. The linkage is then clamped and the whole process repeated. Tracing a long one dimensional specimen is thus somewhat tedious since it is necessary to perform the clamping/unclamping cycle approximately every ninety seconds.

The ratio arm system is used in the digital version of the microphotometer to trace two dimensional specimens. The recording and specimen tables are clamped together permanently. After one sweep of the recording table it is reversed back to its starting point thereby bringing the specimen

table back to its starting point. Whilst the two tables are reversing, the specimen table is being stepped in the direction normal to the direction of tracing. The area of specimen which is traced in any given cycle is thus slightly offset from that traced in the previous cycle. The actual data sampling is controlled by a commutator which is linked to the back table by two timing belts, and it is possible to vary the number of data samples between 5 and 100 per inch of travel of the recording table.

One dimensional specimens, such as laboratory or astronomical spectra, are most conveniently traced with the screw drive attachment. This attachment drives the specimen table directly and no linkage is needed between the recording and specimen tables. The commutator controlling the data sampling in this mode is driven by the screw drive motor and the sampling distance between successive data samples is variable between half a micron and ten microns. A screw drive such as this is standard practice in microphotometer design. However one complication may be desirable when the instrument is used in digital form. Tracing a long specimen may result in the output record being too long for convenient input to the computer. For example, the IOCS routines for the IBM 7094 are restricted to records of 9999 words or less. The current Joyce-Loebl system allows a choice between writing records of unspecified length and writing records of 400 words. In the latter mode of operation when a counter indicates that 400 words have been written on tape the screw is declutched from its drive motor, the commutator is clamped and an inter-record gap is written on the output tape. The screw is then re-engaged with its drive motor, the commutator is unclamped and another record of 400 words written on tape.

The transmission data of a particular specimen form a number of records on tape. The data is written in BCD at a density of 200 BPI. Each transmission value is a three digit number with each digit forming one column on seven track tape. Since six columns make up one computer word, one transmission value forms only half a word. Further data, such as the number of a particular specimen, may also be written on tape. This data, known as fixed data, consists of twelve decimal digits which are manipulated by rotating wheels on the tape recorder. These data are written on tape simply by pushing a button. Further buttons allow the insertion of interrecord gaps and end of file marks. It is impossible to read the microphotometer tape by using the normal Fortran READ statement. It is however quite simple to read the tape and convert the BCD data into the normal binary computer form by writing a small program which uses assembly language as well as Fortran. This program, which has been written for an IBM 7094, also uses many IOCS routines. It would be quite possible to rewrite the program for a Univac 1108, using the ITEM HANDLER to read the tape and converting from BCD to binary by using the Fortran Field Function or using assembly language.

The following pages give details of five computer programs written to analyse digital microphotometer data. These programs are:

i) Program number 1. A general program to read the tape produced by the microphotometer, convert the data from BCD to binary format, supply a listing of the data and write a high density copy of the converted data. The listing of this program forms Appendix A.

ii) Program number 2. A program to convert the microphotometer data to an intensity scale using laboratory calibration data. The listing of this program forms Appendix B.

iii) Program number 3. A program to check the calibration used in program 2. This program is listed as Appendix C.

iv) Program number 4. A program to convert the microphotometer data to an intensity scale using calibration techniques employed in stellar spectroscopy. This program is listed as Appendix D.

v) Program number 5. A program to identify lines in stellar spectra. The input data is an estimate of the starting wavelength and dispersion and a data tape containing a list of lines which are expected to occur in the stellar spectrum. This program forms Appendix E.

Program 1

PROGRAM TO READ MICROPHOTOMETER TAPE
AND PERFORM DATA FORMAT CONVERSION.

The sole purpose of this program is to read the tape which has been written by the microphotometer, to convert the data from the BCD coding of the microphotometer to the binary coding of the IBM 7094 computer and to write out the information on tape for further processing by subsequent programs. The program is a mixture of Fortran IV and assembly language and uses some of the IBM IOCS routines. It is a pleasure to record that the original version of this program was written by Mr. H. Wendlandt.

The program begins with the request to the computer operator to mount the appropriate tapes on the tape drives. In the particular case of the listing given in appendix A he is asked to mount tape Bell 1 on A5 and a scratch tape on B6. We also ask him to put the A channel density switch on 800/200. Since each microphotometer transmission value occupies only half a word on tape, it is possible for an input record to contain a half integral number of words. This is very unlikely to happen with tapes which have been written by the 7094 and the operator is consequently warned (when such a record is read) by the TWI light flashing on the machine console. To avoid termination of the job because of this we warn the operator that the TWI light may flash. We then ATTACH drives A5 and B6, it being necessary to ATTACH A5 only because we are going to use low density. The control and ID cards then follow. The program begins with the COMMON statement defining the variables used in both the Fortran and assembly language programs. These variables are:

IN The array where the microphotometer data is stored.

- IEOF The end of file indicator, normally zero, but set equal to one after reading an end of file.
- IERR The error indicator, normally zero, but set equal to one when an error occurs on reading tape.
- IREEL The number of input files. These may be on either one input tape or several tapes. In the latter case drives A5 and A6 are used for input and the FILE card is altered to:
- 1 8 16
- A5 FILE A5A6, CK1, UT5, BCD, BLK = 9999, INPUT, REELS
- and A6 must be attached as SYSUT5. The A5 tape is processed first and when an end of file is encountered on this tape the processing of the A6 tape begins immediately.
- NER This array is used to indicate if the microphotometer data is erroneous. Normally it contains zeroes but it is set to unity if the data does not lie in the range 1 to 174.
- NF The file number on the output tape.
- NR The record number on the output tape.
- LC The number of significant data numbers in the output tape record.
- D This array contains the fixed data.
- OUT The converted microphotometer data.
- N This array is used for temporary storage in the conversion routine.
- IND This is set to unity if the first word in the input record is fixed data information.
- LI This is the length in words of the input record.

The next statement, CALL FIXIT, is inserted to stop the 7094 tape drives from making 100 retries if errors occur on reading tapes. The FIXIT routine changes the 100 tries to 3. After setting some indicators we read the number of files which are to be processed (IREEL) and OPEN the input and output files by the CALL OP statement. We then read a record by the CALL RA5 statement. We update the input record number, NR, and check to see if the record is, in fact, an end of file. We then check to see if the length of the record, LI, is greater than 9999. If it is, then we must terminate the job. Unfortunately, in view of the very large records which the microphotometer can write, it is quite possible that a huge record will be read by the 7094 and this will overwrite the program. In such a case, the ITEM HANDLER routine of the 1108 must be used to shorten the records. We then load the last word of the input record into N and call the conversion routine to see if the last half word of the input record is a genuine piece of data. If it is genuine, then the total length of the output data is $2*LI$ whilst if it is not genuine then the length is $2*LI-1$. However, an input record of 9999 words will generate 19998 words to be output, since each input word forms two output words. This is far too much data to be output as one record and so we choose to process the input data 500 words at a time. If an input record contains less than 500 words then we process the complete record. The processing is done in the DO loop beginning at card number 48 and continuing to card number 70. Any data which does not satisfy the requirement that it should be between 1 and 174 is detected and its location, in terms of file, record and word, is punched out on cards. After conversion of the input data we store information in the fixed data area D, if necessary, and we write the converted data out on tape, with the CALL WB6 statement, and also print

each record. Any data errors are also listed on the printed record. It will be noted that we again use IOCS routines to write out the data although there would be no objection, apart from the efficiency loss, to using normal Fortran WRITE statements. After having processed 500 words of the input record we check to see if we have completed processing the input data. If not, we return to card 48 and repeat the procedure whilst if processing is complete we return to card 23 and read another input record. The reading of input records is terminated by reading an end of file which causes IREEL to be diminished by 1 and then checked to see if it has become zero. If IREEL is still nonzero an end of file is written on the output tape by the CALL WIN statement after which the whole process is repeated. If IREEL is now zero we close both the input and output files and terminate the job.

A listing of this program is found in appendix A.

Program number 2 - Fluid Dynamics

Program to reduce microphotometer data to an intensity scale using a wavelength dependent calibration curve.

One of the problems of photographic recording of the light from laboratory plasmas is the reduction of this data to meaningful quantities. Most laboratory sources using photographic data recording generate much more data than can be reduced by usual hand reduction techniques. To partially alleviate this problem, a computer code has been written to reduce the microphotometer data to an intensity scale using an external wavelength dependent film calibration.

This program was originally written as an aid in the reduction of data from a gas driven shock tube in the Institute for Fluid Dynamics and Applied Mathematics although the routine is of sufficient generality that it can be used for other purposes. The majority of the research using the shock tube is concerned with the measurement of atomic transition probabilities of the light elements. Although extensive photoelectric recording is used in this experiment, a large part of the data is recorded using a time resolved drum camera spectrograph combination. The films from the high speed drum camera are developed using a standardized procedure and reduced using calibration films developed under identical conditions.

The advantages of computer reduction of the time resolved films are many. The growth of a particular spectral feature in time can be followed. An optical depth correction for bulk absorption is included in the code. One of the biggest advantages in the use of the routine is that all of the spectral features are reduced and the additional effort required to look at additional details in the same spectra is much reduced over a similar hand reduction program. As an example of the value of this program the transition probabilities of 180 lines were measured in about 1/3 the time required for a similar earlier investigation.

The purpose of this program is to reduce density data from the Joyce-Loebl microphotometer to an intensity record using a wavelength dependent film calibration curve. The input density data is the output from Program number 1. The tape read and writing procedures are similar to those in Program number 1. The following describes only the Fortran IV portion of the program as the IOCS code is similar to Program 1. The tape read and write statements are also similar to those used in Program number 1.

The calibration data for this program is generated using the carbon arc as an intensity source of unit strength. The light from the carbon arc is attenuated by a known amount using either a calibrated step filter or a neutral density wedge of known transmittance. The wavelength dependence of the attenuating material must be known so that the final characteristic curves [density vs. $\ln I$] can be normalized to the unattenuated carbon arc intensity.

Usually a characteristic curve is generated from the calibration films at wavelength interval of about 100 Å. Other wavelength intervals could be used, but 100 Å was chosen because of variations in film sensitivity with wavelength. It is not necessary to pick wavelengths in equal increments.

The film calibration data for the programs consists of values of density read at equally spaced increments of $\ln I$. Because of the iteration procedure used in the program it is necessary that all of the values of

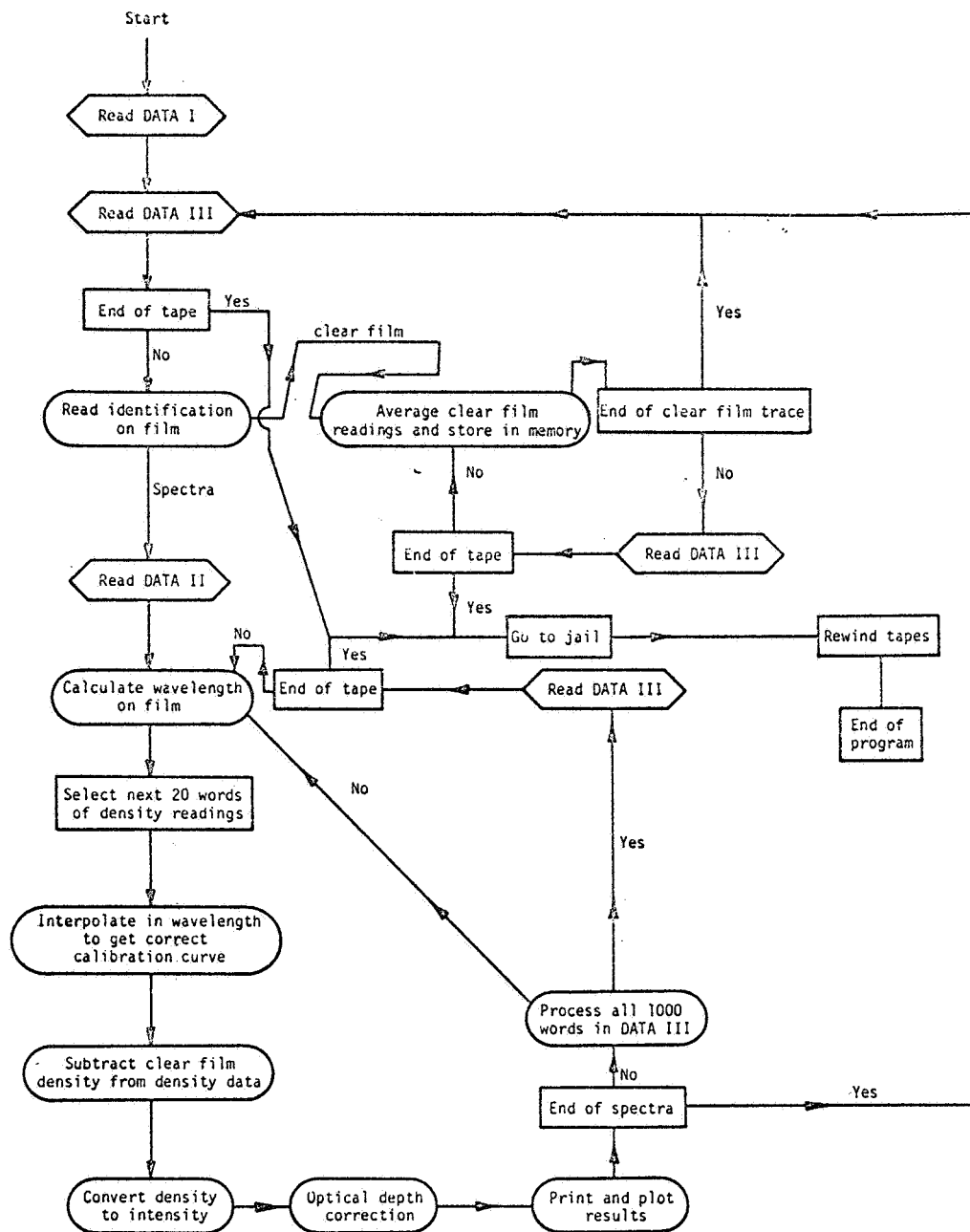
[$\ln I$] have the same increment. Each data card contains numerical data from the characteristic curve at a given wavelength. The first entry on a data card is the wavelength, then the number of points of calibration data on the card, and finally the density values on the given points of $\ln I$. The values of $\ln I$ are read in as a vector (RAD) after the density values. See the listing in appendix B for sample data and format instructions.

It should be pointed out at this point that it is necessary to have two calibration cards at lower wavelengths than the lowest wavelength from a shock tube spectra. This is necessary because of the iteration method used.

We often experienced considerable difficulty in fitting together several calibration films to cover a wide wavelength region. For this reason a computer code has been written to assist in plotting parameters associated with the calibration data and is listed as Program number 3.

Figure 1 outlines the basic procedure used in the computations. As the program is now organized, it is set up to reduce 70 mm film from a JACO f/6.3 spectrograph and drum camera combination. The reduction of films from a spectrograph with a different dispersion or different film length can be carried out by changing a few constants.

The basic method of operation of the program is (1) average clear film density data, (2) subtract averaged clear film value from reading, (3) convert to real density, (4) convert density to intensity using calibration curve, (5) optical depth correction, (6) plot and print results as a function of wavelength. These procedures are discussed in order in the following paragraphs and are shown in Figure 1.



DATA I: Calibration curves as a function of wavelength
 DATA II: Temperature, beginning wavelength, normalization factor
 DATA III: Identification plus 1000 words of density measurements

FLOW CHART FOR COMPUTERIZED FILM REDUCTION

Figure 4. Flow Chart for Computer Reduction of Data

A particular film record is marked by the twelve digit identification numbers C(x) which consist of four three digit numbers. Only the first two are important for the operation of the program. The first digit (JJ) of C(1) defines which density wedge was used to trace the film. The density wedges are numbered as follows.

| JJ | nominal density range of wedge |
|----|--------------------------------|
| 1 | .4d |
| 2 | .8d |
| 3 | 1.2d |
| 4 | 1.6d |
| 5 | 2.1d |
| 6 | 2.4d |
| 7 | 3.1d |

The second digit (JF) of C(1) is used to define the region of film which has been traced. The definitions of JF are shown below.

| JF | region scanned |
|----|--------------------------|
| 1 | clear film background |
| 2 | first trace across film |
| 3 | second trace across film |
| 4 | third trace across film |

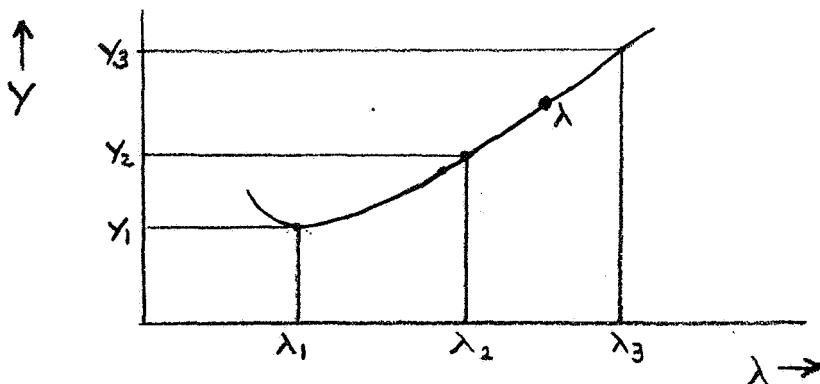
C(2) is usually used to record the position in mm of the trace from some reference point, usually taken at the point where the spectra begins. C(3) and C(4) are used to label the film. All of the numbers C(1) - C(4) are printed out on each record and on each Cal Comp plot.

The procedure for averaging the background clear film readings is to average twenty points and store as a vector in memory [VB(XXX)]. After all the clear film data has been read, and stored in memory, a "numerical filter" is used to remove any large sudden jumps in the stored background data.

The data from the scan of the spectral region is treated in groups of twenty points. All twenty points are treated as having the same wavelength (DAM). The correct calibration curve at a particular wavelength is generated with a 3-point fit using

$$y(\lambda) = y_1 - (\lambda - \lambda_1) \left[\frac{y_2 - y_1}{\lambda_2 - \lambda_1} + (\lambda - \lambda_2) \left\{ \frac{y_3 - y_1}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)} - \frac{y_2 - y_1}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_2)} \right\} \right] \quad (2-1)$$

where



Once the correct interpolated calibration curve at a given wavelength is generated, the $[\ln I]$ value (RA) corresponding to a given density is found using Eq. (2-1). The intensity is found from

$$\text{Intensity} = \text{CON} * \text{ECA} * \exp(\text{RA}) \text{ [erg cm}^{-2} \text{ sec}^{-1} \text{ \AA}^{-1}] \quad (2-2)$$

CON is a constant used to put a relative intensity scale on an absolute level and ECA is the carbon arc intensity (Blackbody at $T = 3792^\circ\text{K}$). CON is

usually found by fitting the tracing to photoelectric data. An optical depth correction is put in using

$$\text{Intensity} = \text{TAU} * \text{EBBD} \quad (2-3)$$

where

$$\text{TAU} = -\text{LOG}(1. -Y(\text{LLO})/\text{EBBD}] \quad (2-4)$$

and EBBD is the blackbody intensity at T°K.

$$\text{EBBD} = \frac{2h}{\lambda^5} \frac{c^2}{\frac{hc}{e^{-\lambda kT} - 1}} \quad (2-5)$$

The intensity values are printed out and plotted using Cal Comp plotter. Cgs units are used throughout. The output data has units of $\text{erg cm}^{-2} \text{sec}^{-1} \text{\AA}^{-1}$.

A reduced plot is made using only 1/2 of the total data points. A sample output of the reduced plot is shown in Figure 2.

INPUT DATA -- See appendix B

1. Data tape
2. Calibration curve (SE)
3. Intensity scale for calibration curve (RAD)
4. Temperature (T)
5. Beginning wavelength in \AA (Program goes from low wavelength to high wavelength. Pick beginning number where trace begins (BEGIN))
6. CON - constant necessary to put relative intensity scale on absolute basis.

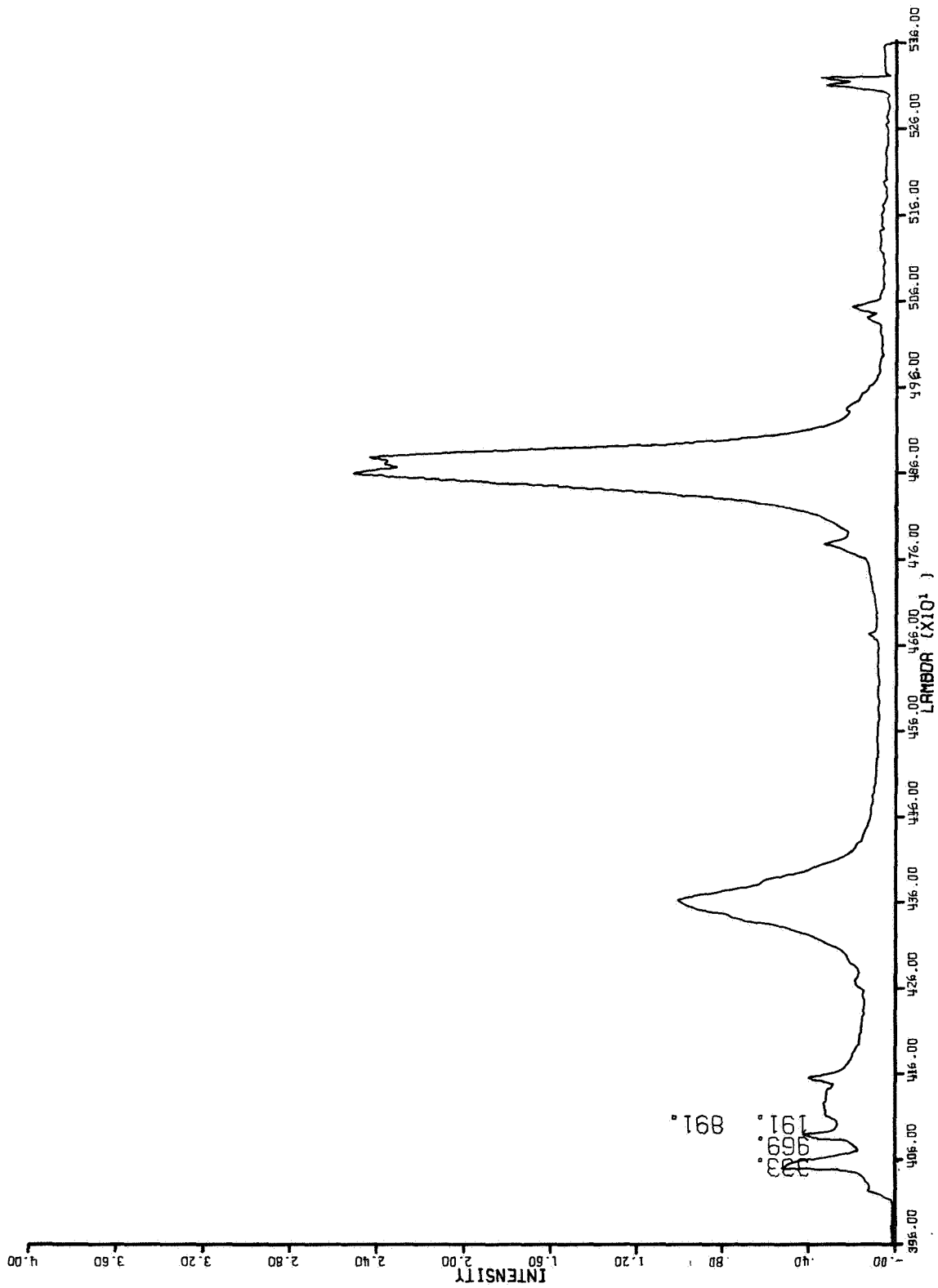


FIG. 2

DEFINITION OF TERMS

In order of appearance

| | |
|-----------------|--|
| SE (x,y) | Input film calibration matrix |
| NN (x) | Number of points of calibration data on one card |
| W (x) | Wavelength of calibration data |
| RAD (y) | Values of [ln I] needed for calibration curve |
| NF | Tape number |
| NR | Record number |
| LC | Number of words in each record |
| C (x) | Identification numbers |
| X (xxx) | Data from microdensitometer |
| FN | Same as NF but for writing on tape mounted on B6 |
| RN | " " NR " " " " " " " " |
| CL | " " LC " " " " " " " " |
| E (λ) | " " C (x) " " " " " " " " |
| S (xxxx) | " " X (xxx) " " " " " " " " |
| N | Number of film calibration data cards |
| NRD | Number of records per piece of film. For 70 mm film, NRD = 9 |
| DSW | Constant needed to convert 1 unit to a density. This constant depends upon which wedge is used |
| SCALE | Divisions/inch on plot (low scale) |
| RSCAL | Divisions/inch on plot (high scale) |
| VB (xxx) | Averaged background level in microphotometer units |
| T | Temperature °K |

| | |
|-----------|--|
| BEGIN | Wavelength where scan begins |
| CON | Constant to convert relative intensity to absolute intensity |
| NILE | Running index to find which data group is under consideration |
| DAM | Wavelength on film |
| ECA | Carbon arc intensity. (Blackbody at $T = 3792^{\circ}\text{K}$) |
| EBBD | Blackbody intensity at $T^{\circ}\text{K}$ |
| COR (x) | Interpolated calibration curve at wavelength DAM |
| XX | Density |
| RA | Interpolated $[\ln I]$ value from calibration curve |
| Y (xxx) | Intensity |
| TAU | Optical depth |
| VVV (xxx) | Intensity for reduced plot |

A listing of this program is found in appendix B.

Program number 3 - Fluid Dynamics

Program to check the calibration data used in Program number 2,

As mentioned in the previous section, there are several problems connected with generating a film calibration data over a large wavelength interval. The purpose of this program is to aid in finding flaws in the calibration data.

The three-dimensional plot is used to check the internal consistency of the film calibration. The natural logarithm of the intensity and the density are plotted against each other on the x and y axes for parametric values of wavelength on the z-axis, producing a plot as shown in figure 3. The values of $\log(I)$ along the x-axis have been made descending rather than ascending for purposes of ease in comparison. Any inconsistencies in the calibration will show up as "humps" or "valleys" in the plot.

To form the graph, the values of density are fitted to those of the logarithm of the intensity for a given wavelength, using the same method as program 2. (See equation 2-1.) The density -- $\ln(I)$ curve is plotted every 100 Angstroms until the red end of the calibration is reached.

The gamma at a given density should be a smooth function of wavelength approximately equal to a constant. For a given wavelength, it is defined as

$$\gamma = I \ln(10) \frac{dD}{dI} \quad (3-1)$$

where I = intensity and D = density . Rearranging the variables:

COMPARISON

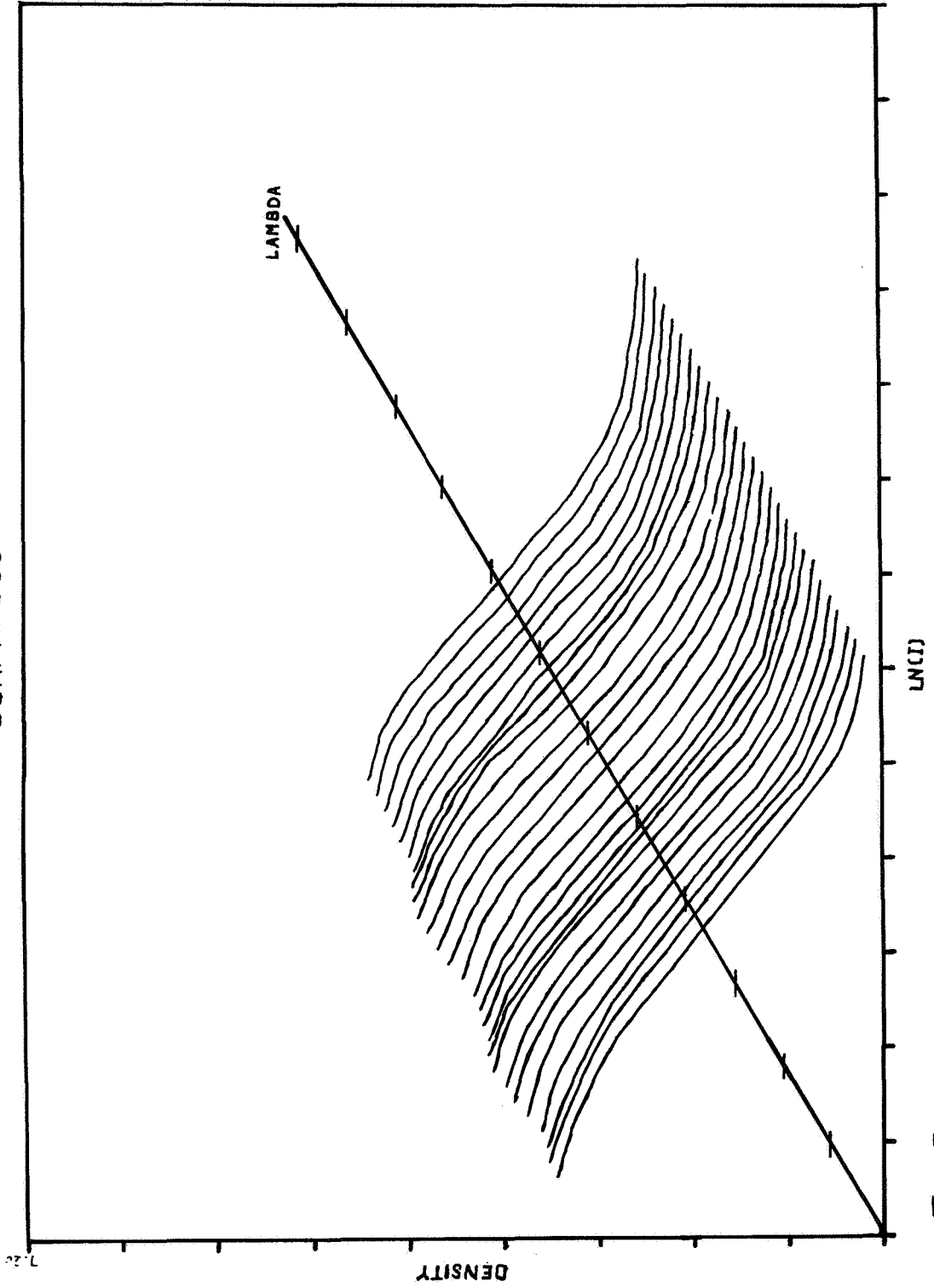


FIG. 3

$$\gamma = \ln(10) \frac{dD}{d(\ln(I))} \quad (3-2)$$

The gamma is proportional to the slope of the density -- log(I) curve just plotted. Thus, by taking the derivative of (2-1) and multiplying by the natural logarithm of 10, the gamma of the film may be obtained via a three point fit:

$$\alpha = \ln(10) \left[\frac{y_2 - y_1}{\lambda_2 - \lambda_1} + (2\lambda - \lambda_1 - \lambda_2) C \right]$$

where

(3-3)

$$C = \frac{(y_3 - y_1)}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)} - \frac{(y_2 - y_1)}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_2)}$$

A typical trace is shown in figure 4.

GAMMA COMPARISON

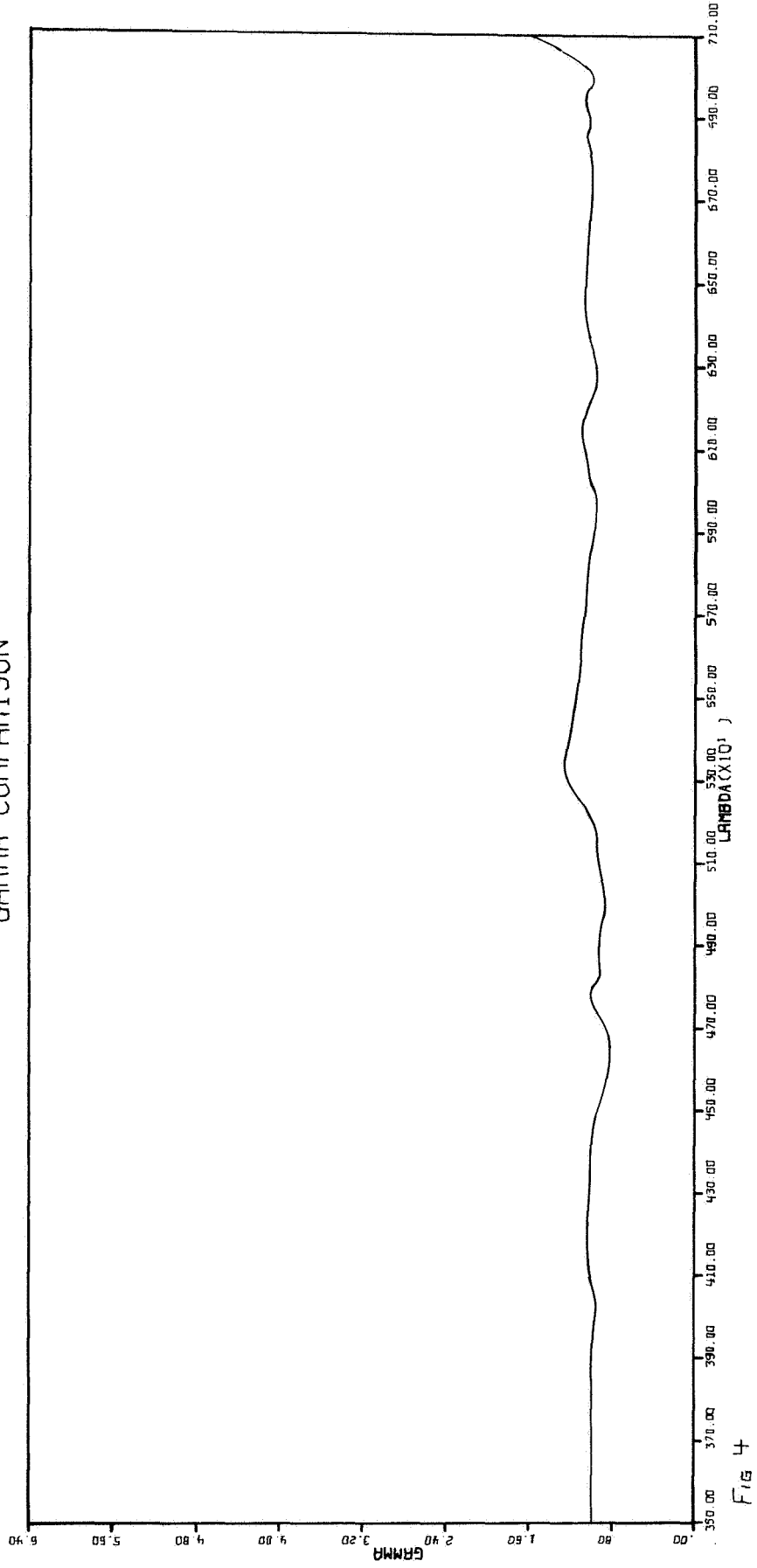


FIG 4

The final plot is used to check the film calibration's internal consistency. Any humps that appear should be investigated to check if they really belong in the calibration. The result is shown in figure 5. From plot 1, the values of density corresponding to the $\ln(I)$ for a given wavelength are taken. Then, for six values of density chosen by the user, the associated $\log(I)$ is found by use of (2-1). The isodensity points are plotted every 20 angstroms from the first calibration wavelength to the red end.

The control cards are shown in Appendix C. These cards and the tape mounting instructions apply only to the IBM 7094. For other computers, consult the manual of the computer involved.

INPUT DATA -- see appendix C for sample

1. Calibration data - Appendix B & C for samples
 2. isodensities
 3. gamma density
- } Appendix C

FILM CALIBRATION

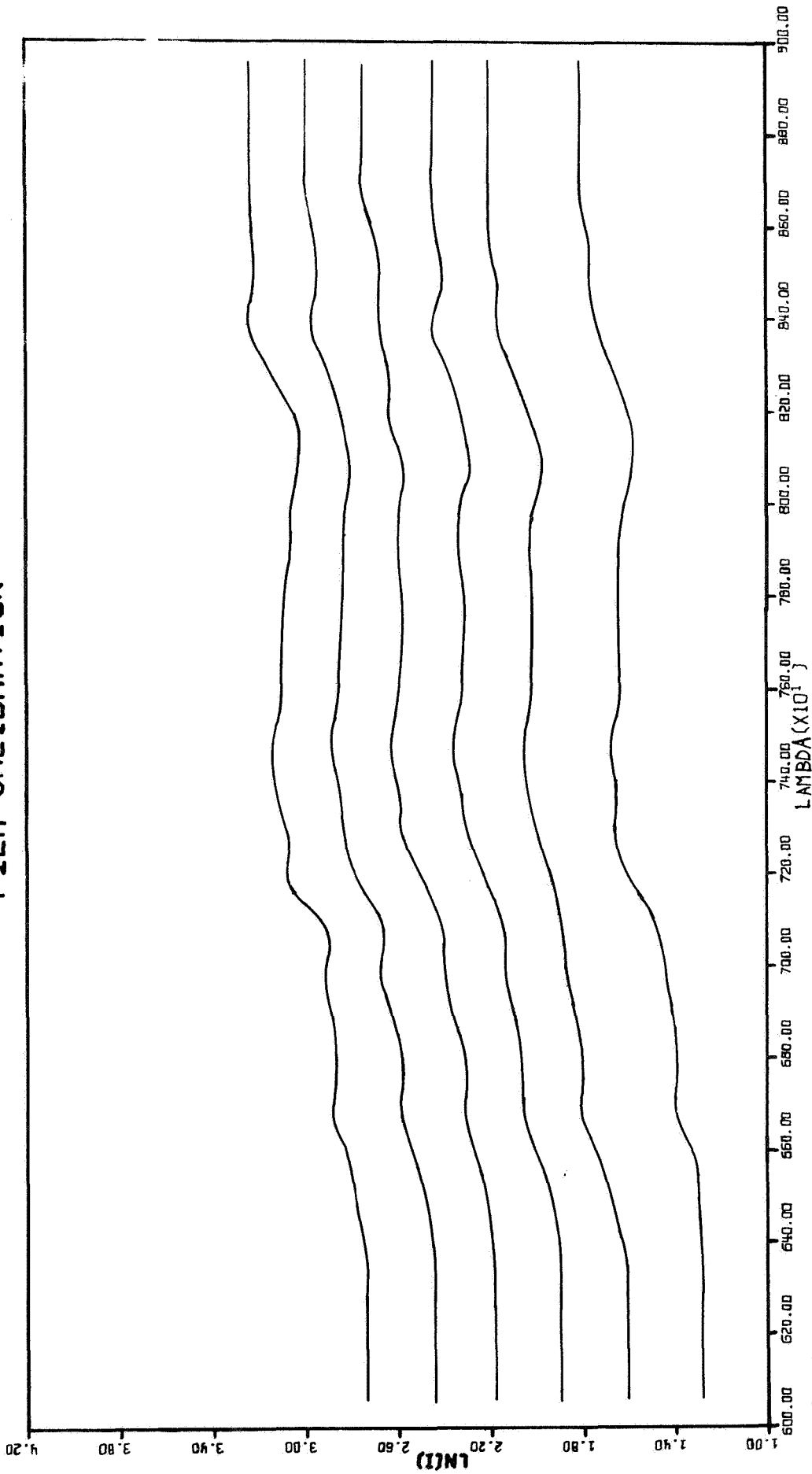


FIG. 5

Program number 4 - Astronomy

Program to convert microphotometer data to an intensity scale.

It was quite clear several years ago that digital microphotometers would be of great value in astronomy and other sciences by virtue of the great speed and convenience with which digital data can be processed in computers. However it was also equally clear that the complexity of astronomical spectra required the development of very flexible programs to take full advantage of the new instrumentation. In particular, it was clearly necessary to write a program which, given the approximate dispersion and starting wavelength of a spectral tracing, could automatically identify the spectral lines present in the spectrum.

This problem has been attacked in two different ways but has required, as a necessary condition for its solution, writing a magnetic tape containing data on the wavelength, identification, oscillator strength and excitation potential of all the lines which might be expected to be present in the spectrum of the stars being studied. It has been necessary for this tape to contain data on molecular lines, in particular the molecules CH, CN, OH, NH, C₂ and MgH, as well as atomic lines. The availability of such a tape has led to the study of such problems as the interpretation of narrow band photometry of stars. However it is now quite clear that the major gain in knowledge which the digital microphotometer will provide will come in the study of problems where a conventional microphotometer can not be used. As example of such a problem, one may mention the determination of the stellar content of a galaxy. One may trace spectra of stars of different types using a digital microphotometer and then add these spectra together in the computer. Each spectrum can be given a different weight in the addition and radial velocity dispersion can be allowed for. The resulting sum of all the spectra can then be compared with the actual spectrum of the galaxy. The contribution of the various spectral types to the final sum may be varied to optimize the

agreement between the synthesised galaxy spectrum and the actual galaxy spectrum.

It is necessary to begin the data reduction process by using a program which reduces the microphotometer data so that it is on a light intensity scale and known wavelength scale. Subsequent programs can then use this processed data to measure various quantities, such as the equivalent width of spectral lines.

The purpose of this program is the conversion of the stellar spectrum data from a photographic blackening scale to a light intensity scale. The data which are required for this are the characteristic curve of the photographic emulsion and transmission readings for the clear plate and the spectrum. These data must be written on the tape in the above order. At the present time, tracings have been made of spectra from the Mount Stromlo Observatory and the Mount Wilson Observatory. These plates are calibrated in a slightly different manner. The Mount Stromlo coude spectrograph has a wedge shaped slit on either side of the main spectrograph slit and the spectrum of a continuous light source is photographed through these slits. If this spectrum is traced normal to the dispersion, a record of the photographic blackening versus light intensity is obtained. The Mount Wilson spectrograph uses a series of slits of different widths instead of the wedge shaped slits so that in this case, a series of points define the characteristic curve. The computer reduction program differs in the sense that the Mount Stromlo program derives the curve from scratch, needing to know only which records contain the requisite data. The Mount Wilson program requires that the transmission values for each strip be entered on a data card. This latter version is the one which is listed in the Appendix, as Program number 4.

Whilst the major purpose of this program is the reduction of the microphotometer data to obtain the stellar spectrum on a light intensity scale, the program also performs a few subsidiary calculations. These are the determination of the

location of a few spectral lines and the tentative location of the continuum level. This information is used in subsequent programs. The stellar spectrum data is output on tape, using the IOCS routines, and the continuum and line location data are output on cards.

The program begins with the tape mounting instructions and the ATTACH instructions. After the control cards, which ask for the BASIC level of IOCS on the \$IBJOB card, we have the DIMENSION and COMMON cards. The variables in DIMENSION and COMMON are:

TITLE Used to store the information on the data description cards.

ZZHT The intensity data for the Mount Wilson calibration spectra, on a logarithmic scale.

XBIG The continuum level values

DEX Used as temporary working area in the determination of the average value of XBIG

I7 This array designates the input records to be ignored. NF,NR,LC,DDD and X form the input data, NF being the file number, NR the record number, LC being the number of significant data values in the array X and DDD being the fixed data. NNF,NNR,NLC,AD and Y are the corresponding output data.

IEOF is the end of file indicator

IERR is the error indicator

LOLA is a synonym for LC

CALIB is the array which stores the characteristic curve

TEMP is used as a temporary working area.

NZERO is the number of averaged zero level points.

ZERO is used to store the zero level data.

KLOLA is used to sum the LOLA values for all input records

BUN is used as a temporary working area

XXHT is used to store the calibration data.

The program begins by converting the calibration data from a logarithmic scale and opening the input and output files. We then read the first TITLE card which is included in the data deck as a means of giving running instructions.

TREEL defines the number of files to be processed, the data from one plate comprising one file. We then call for subroutine BLACK in order to determine the calibration curve. As may be seen from the listing the calibration curve, stored in the array CALIB, is derived by linear interpolation amongst the observed data points. The technique used is to compute the light intensity corresponding to transmission values ranging from 1 to 174 in steps of unity. The resulting data is printed and, after returning to the main program, we read in further TITLE cards followed by the variables I1, I2, I3 and I4. The first I1-1 records are skipped. In the Mount Stromlo program records I1 to I2-1 contain the calibration data but they are skipped in this version of the program. Records I2 to I3-1 contain the zero data and records I3 to I4-1 are the stellar data. The I7 array, denoting records which must be ignored, is read after which we set the initial values of several indices and read the first correction card. We then read a record from tape, check for an error in reading and for an end of file, check to see if any correction is needed to the numbers in the array and check to see if the record should be omitted. After this we check to see if the transmission at any point is greatly different from that at neighbouring points. Such a variation could be caused by dirt or a scratch on the plate and, provided it is restricted to a single value, it will be filtered out. Following this part we will either call the SZERO subroutine or will convert some stellar data.

The SZERO subroutine must average and store the information on the transmission of light through the clear plate. The first part of this subroutine determines the average value of the zero level for the record as a whole in order to see if any values are anomalously large. This will detect the presence of dirt or flaws in the emulsion, provided that they do not occupy a significant fraction of a record. After this we average twenty zero level readings together and store the result in the ZERO array. The length of this array is stored in NZERO. This routine is complicated by the fact

that an input record may not contain a multiple of twenty numbers and it may be necessary to store the values at the end of one record to use in conjunction with data at the start of the subsequent record.

Prior to converting the stellar data from a blackening scale to a light intensity scale, we again check the zero data for smoothness, this time over a large scale. The technique is the same as before, namely we check to see if any particular zero level is much larger than the average value of all the zero levels. We then derive the appropriate index for interpolating in the ZERO array to find the zero level for each stellar reading. We subtract the zero level from the stellar reading, convert to a fixed point number and use this to interpolate in the CALIB array to find the stellar reading in an intensity scale, storing the result in the array Y. We then search the array Y to determine which member of the array has the greatest value. This value, the tentative continuum location, is stored in XBIG and its location is stored in DEX. After printing out the record, we fill up the array Y from LOLA+1 to 1000 with zeros and call the WB6 routine to write the record out on tape. We then call the MINIM subroutine to determine the locations of minima, i.e. spectral lines, in the array Y.

In subroutine MINIM the record length is first examined since it is pointless to search for lines in a short record. The smallest number in the array $Y(J)$, where $20 \leq J \leq LOLA - 20$ is found and its value and position stored. Supposing this minimum to be $Y(J1)$, the elements of the array with index between $J1-10$ and $J1+10$ are then set equal to the huge value of 5000 after which the program finds another minimum in the Y array. A total of seven minima are found and these are then sorted into increasing order of intensity, the strongest line being the first. These data are then printed, after which we return to the main program.

The record number is checked to determine if this is the last stellar record. If not, we return to statement 10000 and process another record. When all records have been processed the raw continuum data is output after which it is smoothed and output again. Further records are read until an end of file is encountered after which we write a file mark on the output file and either terminate the job or return to statement 550 to compute a further calibration curve. The listing of this program forms Appendix D.

Program number 5. - Astronomy

Program to identify lines in stellar spectra.

The purpose of the following program is the identification of lines present in a stellar spectrum. In order to do this it is provided with a list of wavelengths of spectral lines which are expected to be present in the spectrum of the star. The program has been tested using spectra of F and G stars and appears to work satisfactorily. However, it is not known if it would work for A type stars. For B or M type stars the line list currently used should be replaced.

The program is written to process stellar spectral records of 1605 words. The first 1600 or less words are the spectrum itself and further information is contained in words 1601-1605. These records are produced by versions of programs 1 and 4 which are slightly different to those described in this paper.

Since the program is quite lengthy, a brief description is initially given, together with a list of the input data cards. This is followed by a more detailed description and list of variables. The program is listed in appendix E.

The complete program consists of the main program, POPE, 3 Fortran subroutines and a MAP routine. The MAP routine has seven entry points. OP and CLO prepare needed tapes for mounting and dismounting; WIN and WAN position one of the tapes; RA5 and RA6 read the tapes; WB6 writes an output tape.

The first section of the main routine deals with handling the tapes and reading in a list of lines which we might expect to find in the observed spectrum. This line list is generated previously using a model stellar atmosphere and is recorded on the tape which is mounted on unit A5. The tape contains the wavelength and predicted strength of the line and a code number indicating which ion is responsible for it.

The program begins by reading the quantity IFN03 and the data cards describing which files in the observed spectrum tape, on A6, are to be processed. The tape on A6 is then positioned to the beginning of the first of these files. The quantity X BEG, the estimated beginning wavelength of the first record, is read in together with a correction factor.

A title is read in from cards - this is strictly for output purposes, and is not used per se in the calculations. The complete list of predicted lines, as described above, is then read from unit A5, and stored in core.

The final preliminary step is to read a group of continuum cards, which specify the level of the continuum at a number of points in the observed spectrum.

For a complete description of the nature of this input, see the following section, entitled INPUT-OUTPUT (p.40).

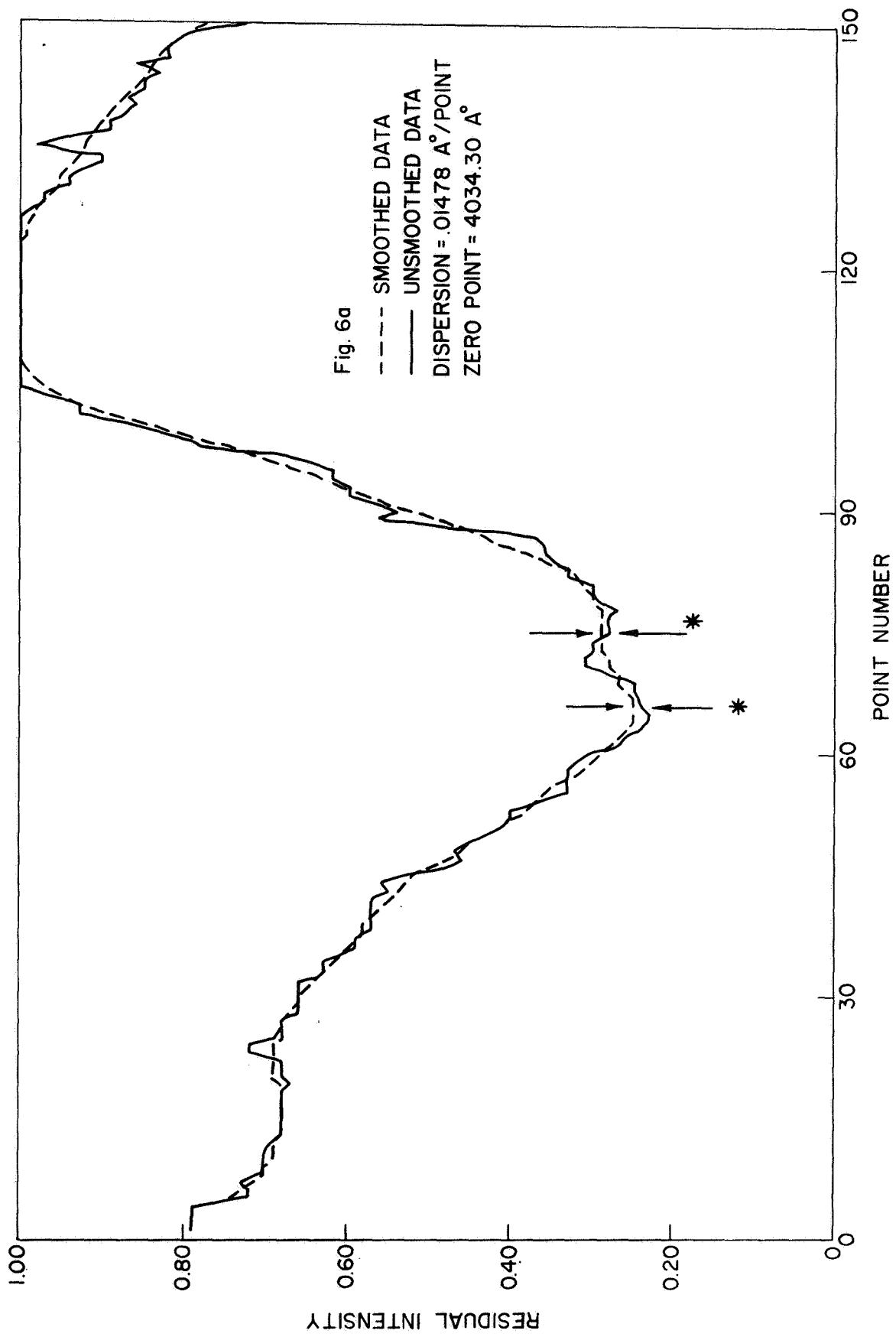
At this point we are ready to begin the real operations of POPE. We read in a single record from A6; this is the digitized output of the microphotometer, converted to an intensity scale by the use of program 4. This record contains approximately 1600 data points; the number of

data points is basically limited by two effects. Firstly, at least 8 to 10 discernible lines should be included in a block of data; secondly, no more than 1600 data points are allowable. The first restriction is caused by the method of identification of lines; the second by core storage limitations and by the assumption that the dispersion is constant within a block.

The number of significant data numbers in the record is then determined after which the continuum levels which have been previously read are used to normalize the relative depths from the tape A6. Interpolation or extrapolation is used to determine the continuum level at a given point.

The beginning wavelength of the record and the dispersion have been set up previously. For the first record the beginning wavelength, X BEG, is read from a card, and the dispersion is assigned by a single statement early in the main routine. For subsequent records the beginning wavelength is incremented as described below (using the dispersion and the number of points in the record, plus a correction factor), and the dispersion is determined from the observed and calculated wavelengths of the spectral lines in the observed spectrum.

POPE now takes the block of spectrum and determines the central wavelengths of any spectral lines contained therein. Since the data coming into the subroutine is somewhat too noisy for our purposes, and contains many spurious "lines" caused only by noise, the first step is to replace each point by the average of the nine points centered on it. This has the effect of smoothing out the noise. Figure 6 demonstrates the effect of this averaging. Here we present a single block of data for the star Delta Canis Majoris. The solid line is the unsmoothed data; the dashed line is the result of smoothing. Here the ordinate is residual intensity and the abscissa is the point number.



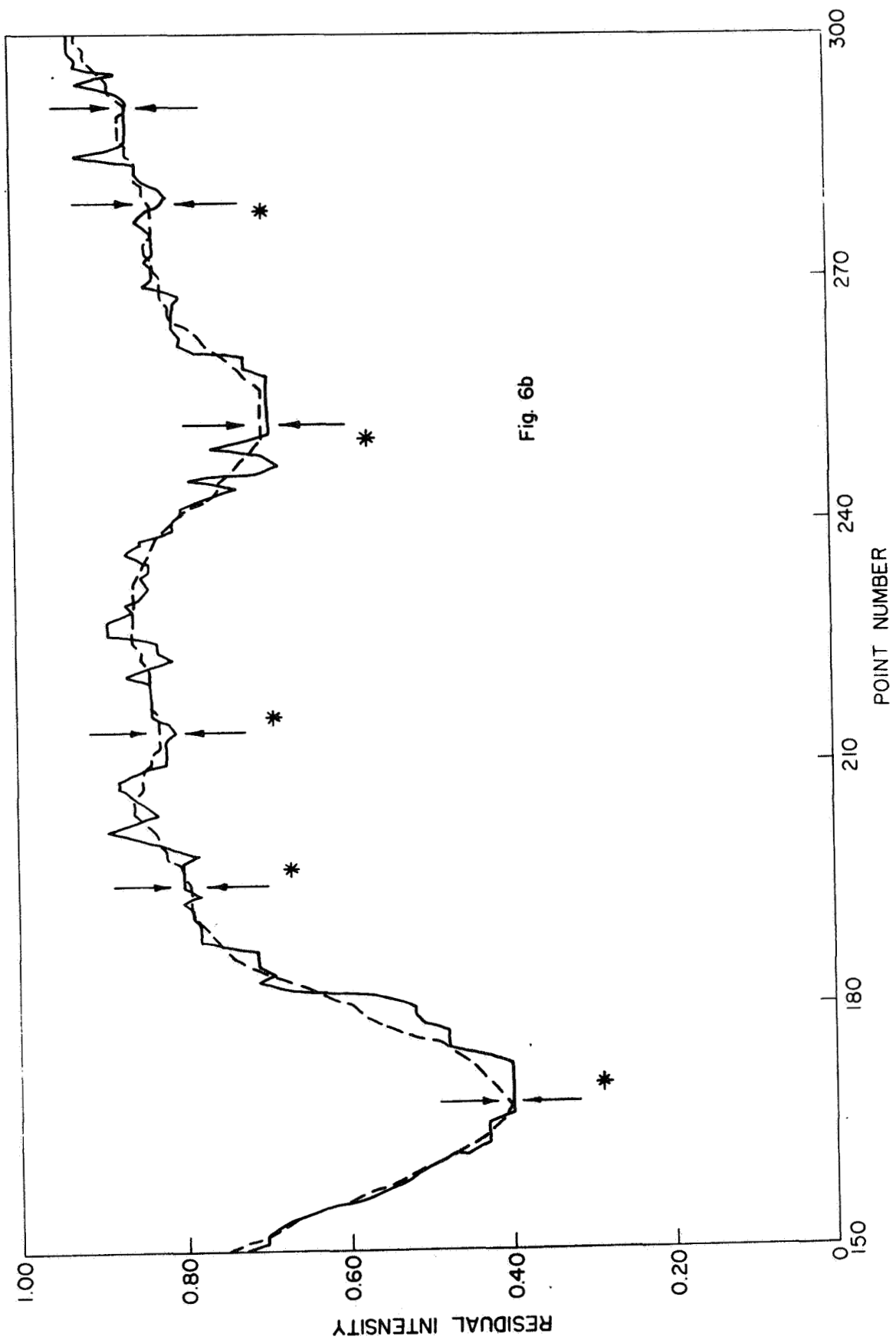
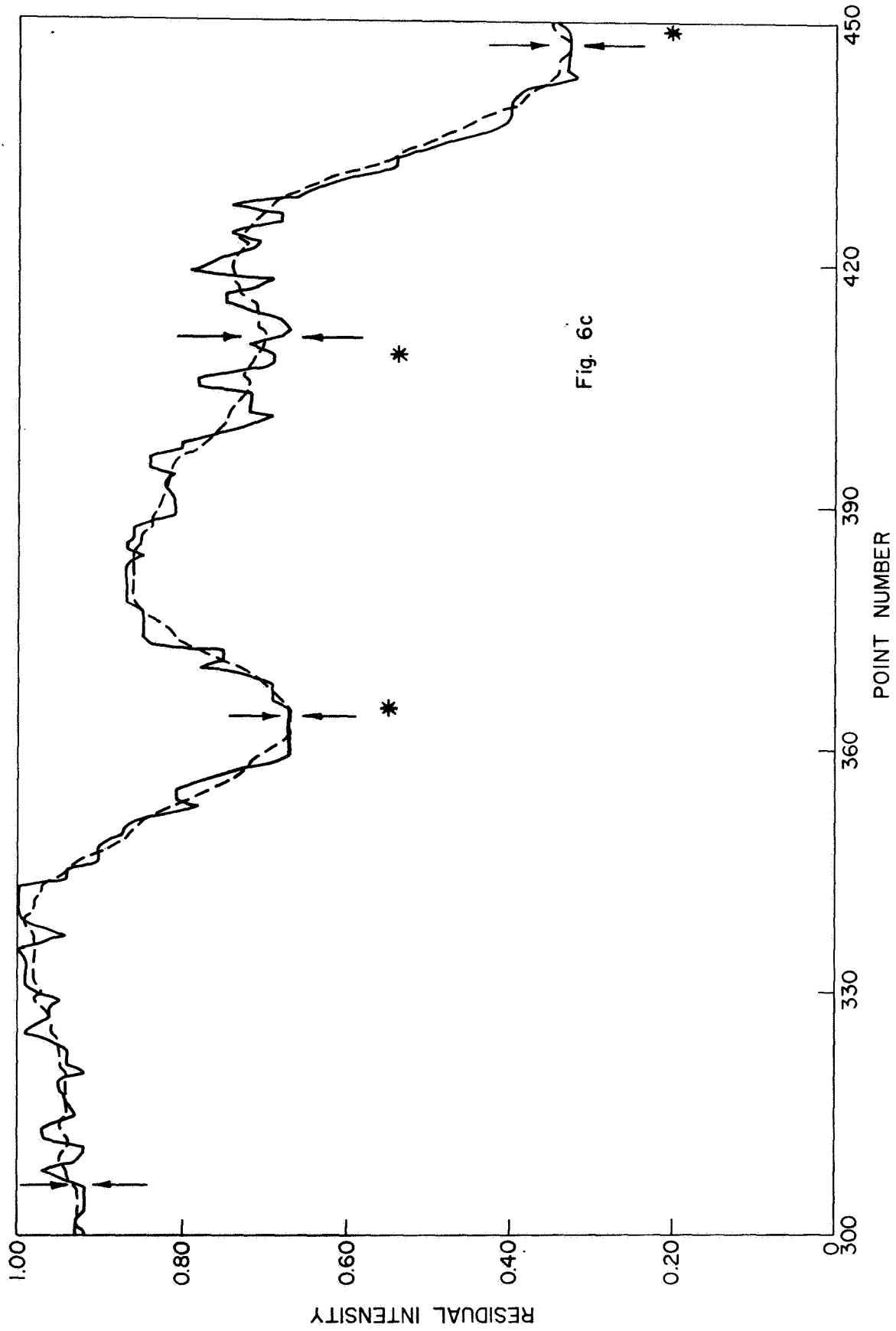
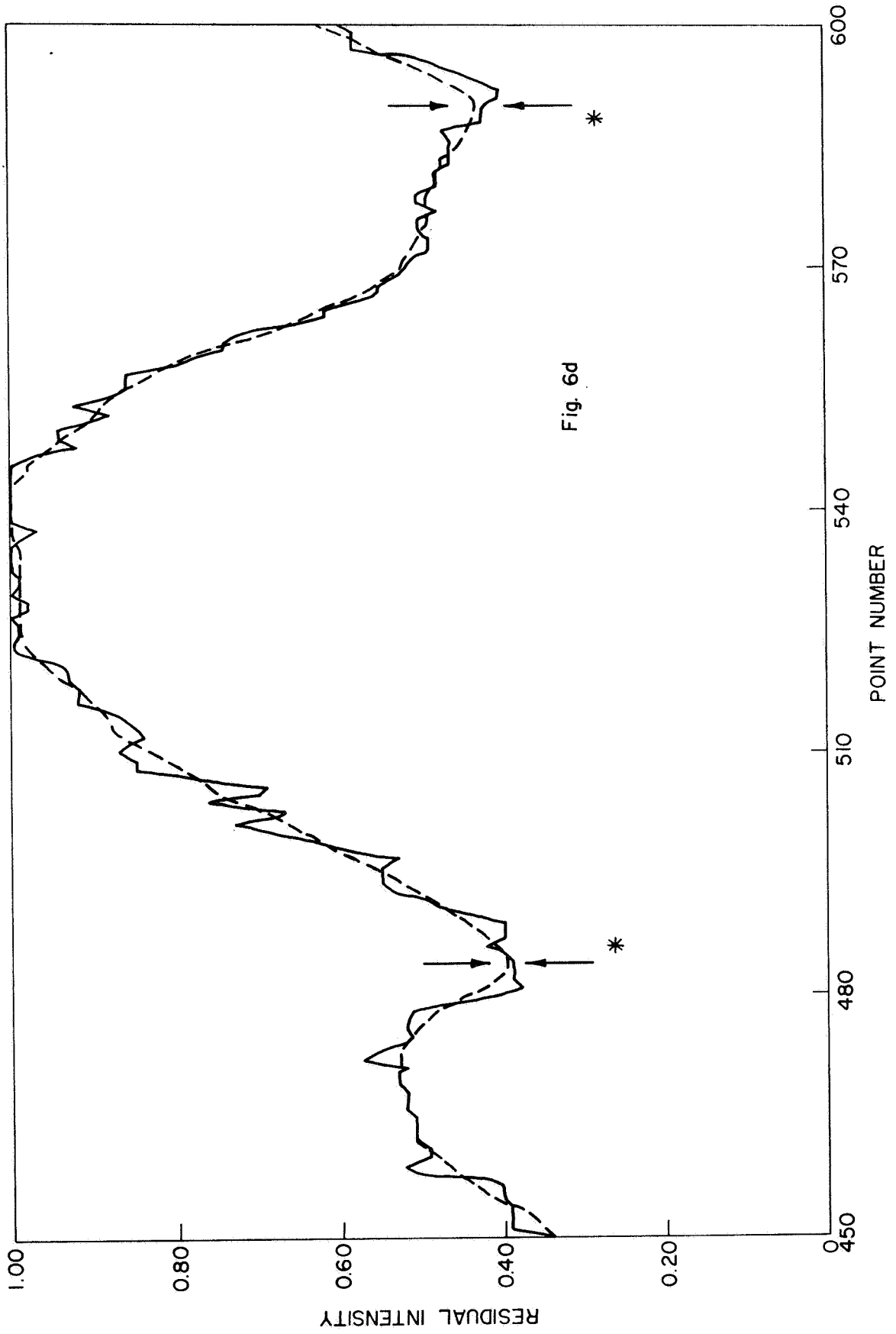


Fig. 6b





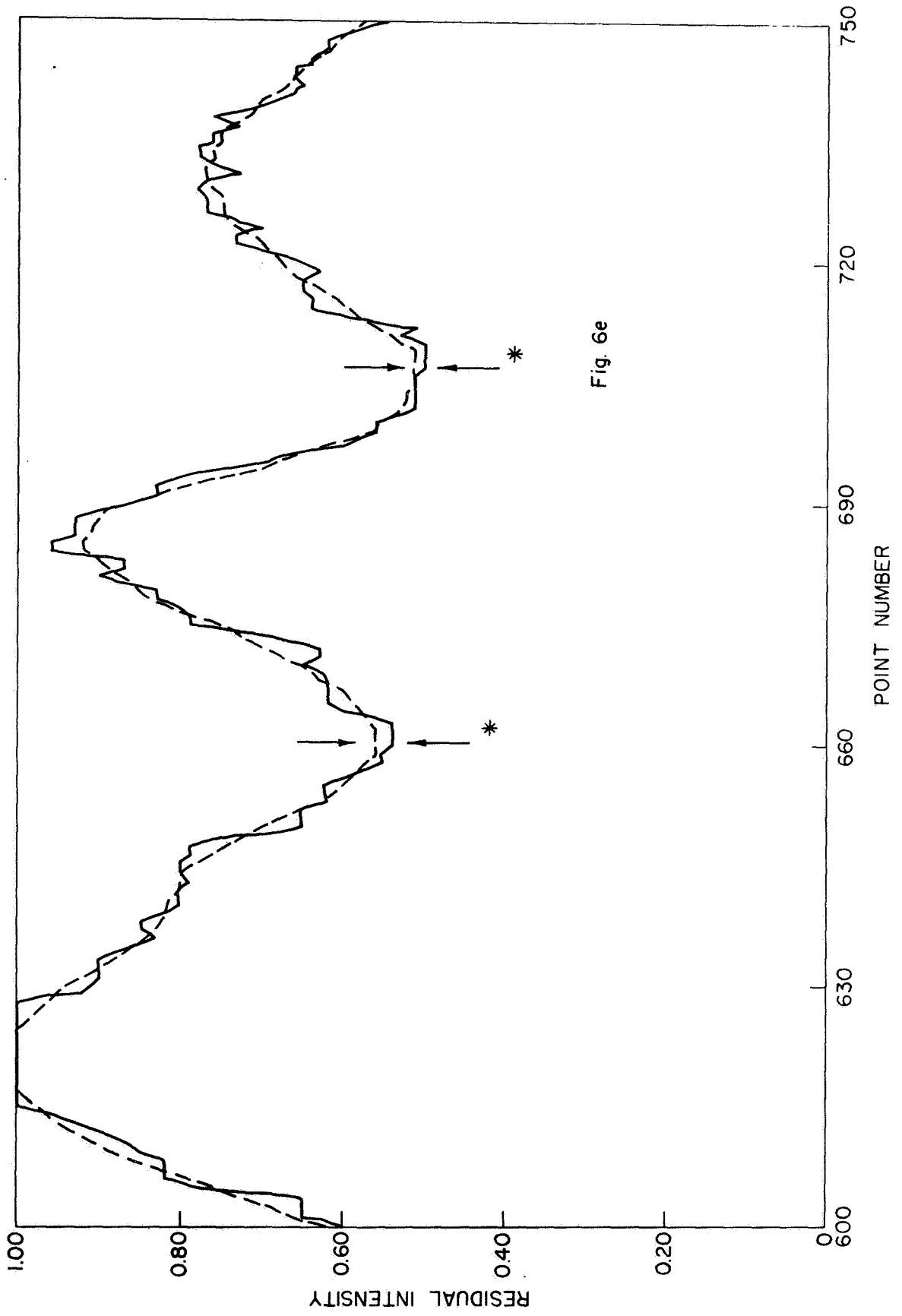
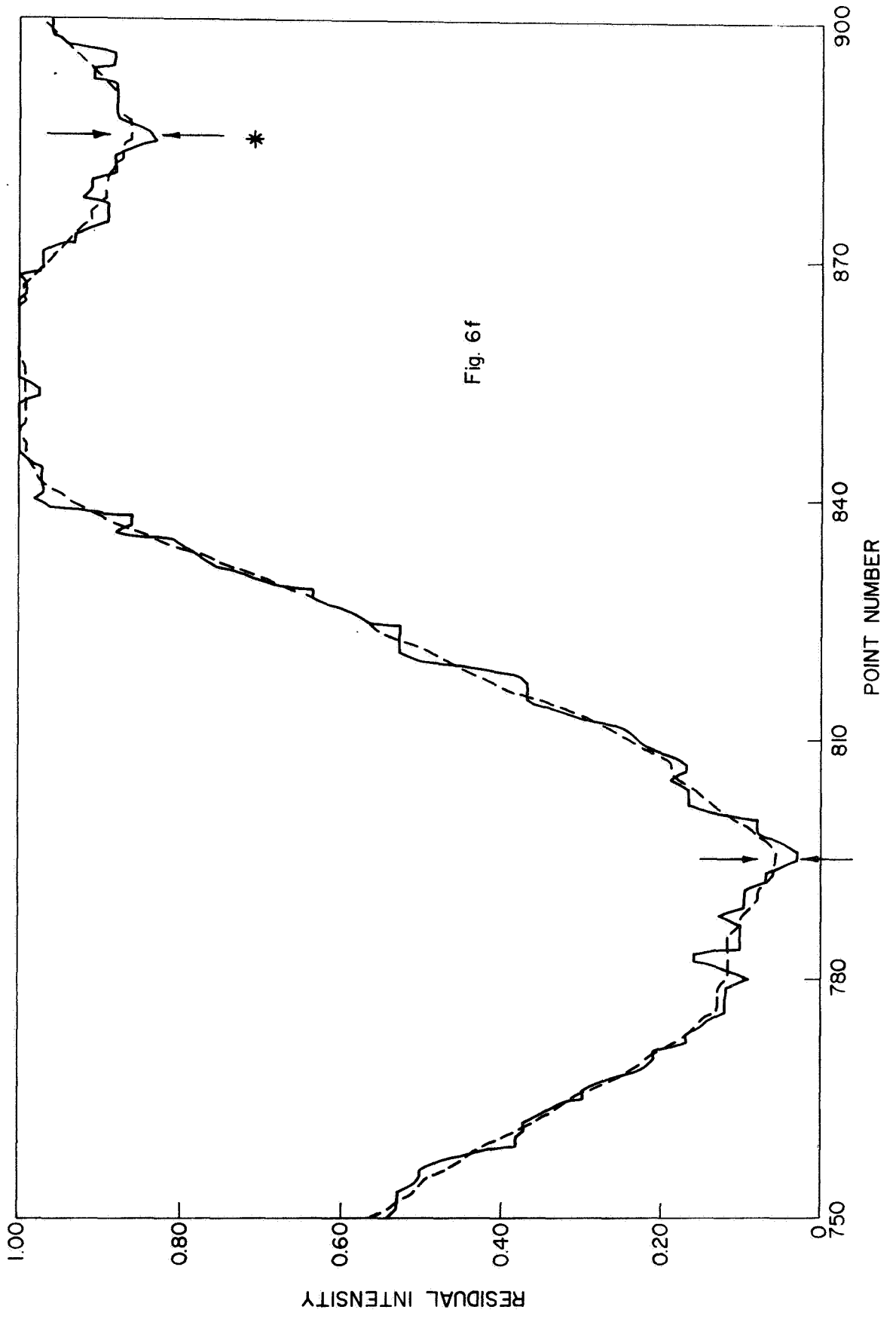


Fig. 6e



Next, the maximum and minimum values of the smoothed data are determined. A parameter, XSTOP, is defined as 15% of the difference between maximum and minimum intensity values.

We are now ready to go on to finding line centers. The criterion employed for this purpose is that a rise of at least 3% must occur in the residual intensity on each side of the proposed line center. First we pick out all the minima in residual intensity in the smoothed data. Such a procedure yields a number of possible spectral line centers. These are indicated by arrows in Figure 6. A total of 18 such were found in this particular block of data. These line centers, in wavelength units, are stored in SLINE (*,2), and their total number in SLINE (200,2). The data point numbers of the line centers are stored temporarily in SLINE (*,3).

Two criteria are used to eliminate spurious minima. First, the 3% criterion mentioned above is applied. Second, all those minima which are not at least as deep as XSTOP are removed. To suppress this last criterion, the input variable IFNO3 is set = 1; otherwise it is 0. The result, for the block of data in Figure 6, is 16 lines, indicated by stars.

Next we store those spectral lines which fall within five data points of the expected wavelength range of the record, into another array, SLINE (*,1). The total number of such lines is stored in SLINE (200,1). If there are no lines within the expected wavelength range, the record number and starting wavelength are simply incremented, and the next record read. We then return to the step where we originally read a record from A6.

We now use the unsmoothed data to determine the depth of the line, which is defined as the minimum value of the residual intensity within the range of the line. The range is defined as 1/4 of the way to adjacent lines. This intensity is stored in SLINE (*,3).

The average intensity and its mean deviation are now calculated and printed.

POPE now must correct for changes in the dispersion across the spectrum. Should this be too severe (more than about a 1% change in dispersion per 10 Å), the subroutine will malfunction. Fortunately, errors of this size are not met within the work done to date.

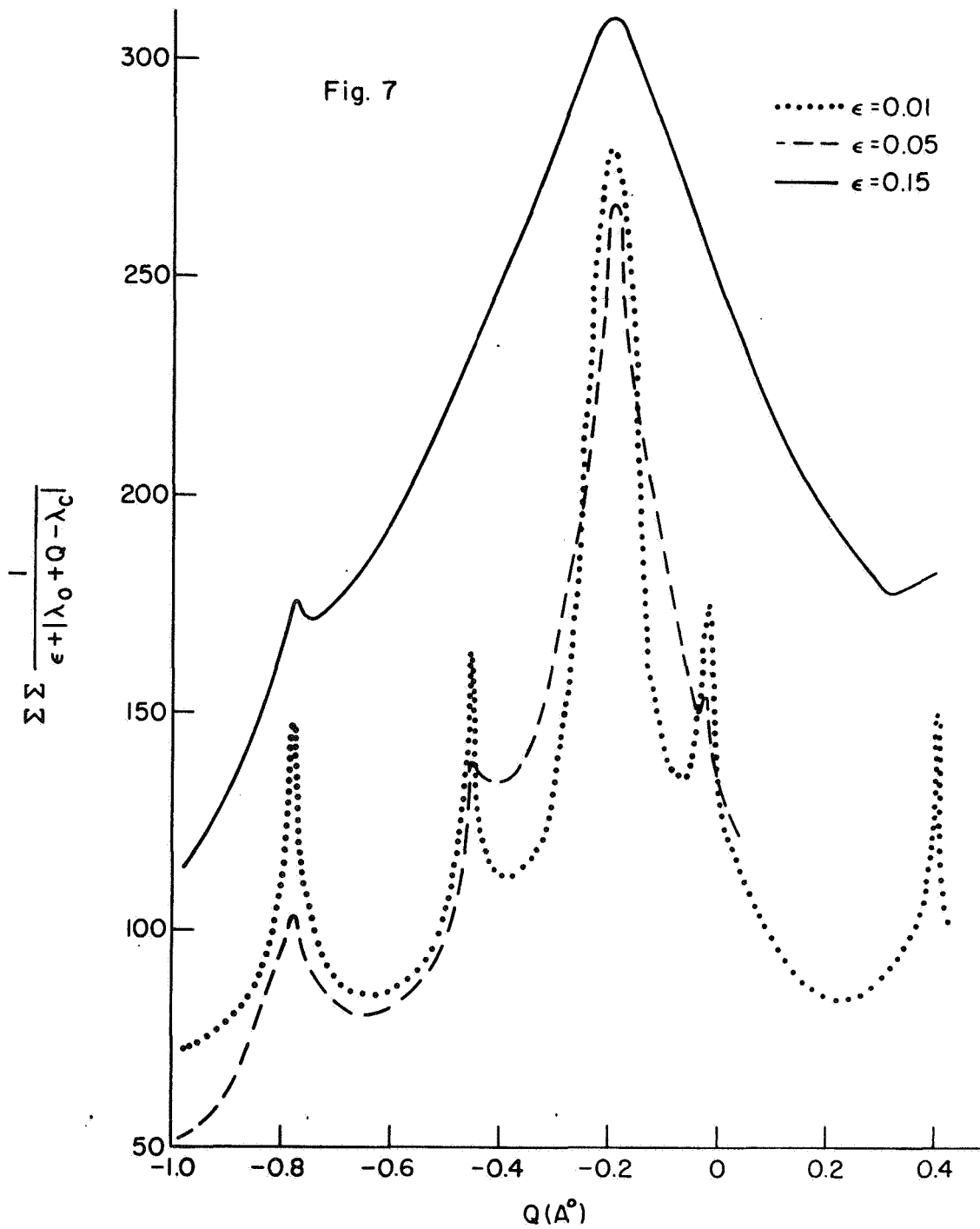
The first thing POPE does is calculate Q, the correction needed to the zero point to make the observed and calculated wavelengths agree as well as possible. To do this, a double sum is calculated:

$$\sum_{\lambda_{\text{observed}}} \sum_{\lambda_{\text{calc.}}} \frac{1}{\epsilon + |\lambda_{\text{obs}} - \lambda_{\text{calc}}| Q}$$

This quantity is a maximum when the wavelength coincidence is as good as possible.

Figure 7 presents this sum for three values of ϵ as a function of assumed Q. The secondary peaks are due to chance coincidences of a single line. On the basis of this figure ϵ equal to 0.15 was chosen for the program. There seems little chance of an error, provided that the correct value of the zero point lies within the range of wavelengths investigated (1.5 Å of the suggested wavelength for the 1st block, and 0.15 Å for subsequent blocks).

POPE now does a preliminary line identification. The residuals -- observed less calculated wavelengths for the identified lines -- are then minimized by a least squares fit, with corrections to the zero point and dispersion as the free parameters. Two-thirds of these corrections are applied. The full correction is not applied, because of errors in the correction terms, which are rather large, and which might cause untoward errors in the dispersion and zero point if they were totally credited. The criterion for identification in this section of the program is coincidence in wavelength within 0.125 Å.



We now return to the section of POPE where we store predicted lines into SLINE. This is done because a new starting wavelength is available, so that the list of lines to be searched for may be slightly revised. We select only a number of strongest lines on the list, equal to 3/4 of the number of lines present in the spectrum.

The array SLINE now contains predicted wavelengths (Col.1), observed wavelengths (Col.2), and corresponding intensities (Col.3). Another variable, IDLINE, contains the ions responsible for the predicted wavelengths. Actual line identification can now proceed.

For each predicted wavelength, we search for coincidence in wavelength among the list of observed lines. The tolerance is set at 0.05 Å; if no coincidence is found, the line is listed as NOT FOUND. In practice, coincidences are found about 75% of the time; random wavelengths give coincidences about 10% of the time.

All that remains is to print out the results of the identification search. Should less than 60% of the lines have been identified, we print out the message POSSIBLE ERROR, together with the average intensity minimum of the identified lines.

An output tape is now written on B6. This tape contains essentially the same information as the input tape A6, except that the data is now an intensity scale of 0 to 1.

Finally, the starting wavelength is incremented, and we return to read in another block of data from A6. This procedure continues until we run out of data in the file we are reading; we then go on to the next file, or `stop`, if no more files are specified.

This completes the general description of program POPE. With a few modifications, it is hoped that this procedure will supply nearly foolproof identifications. However, it is to be emphasized that the input to the program must be as specified. The following section is devoted to input-output for POPE.

INPUT - OUTPUT

All input is accomplished through POPE with its MAP subroutines.

Punched card input is as follows:

1. A single card specifying IFNO3 (Format: I1).
2. A set of cards specifying the number of files of spectrum to be read from A6, together with the numbers of these files. Format is I3/I0I3. Note that the number of files is on the first card. The file numbers are on the second and subsequent cards. In practice, less than 10 files are normally used; only two cards are needed.
3. A single card specifying the beginning wavelength (in Angstroms) of the file and a correction to the zero point of intensity, usual 0.00. Format is (F10.2 IX, F6.3).
4. A single card specifying the title, which is used for output purposes only. Format is I2A6.
5. A single card specifying the number of continuum cards to be input. Format is I6.
6. Continuum cards, equaling in number the information input in #5 above. These cards consist of a relative continuum level, and the data point in the file to which it refers. Format is 2F10.2.

Items 3 to 6 are repeated for each file to be worked.

In addition, two input tapes are needed. The first, mounted on A5, contains a list of predicted lines, the ions causing them, and their predicted strengths. The tape is written in order of increasing wavelength. Three hundred lines are blocked to form one record and each line has three parameters: wavelength, ion, strength (on a logarithmic scale). The wavelength range covered is normally 3000 Å to 6600 Å, but can easily be extended.

The second input tape, on unit A6, contains the observed spectrum.

It is written in records of 1605 words. If less than 1600 spectrum words are contained in a record the record is filled with zeros by the previous program. Both tapes must be high density (800 BPI).

The output consists of the title, block number, zero point (wavelength), and dispersion. Following this is the predicted wavelength, observed wavelength (or N/F if no line could be found at the predicted wavelength), ion code, and minimum residual intensity, for each predicted wavelength. The average intensity of the lines and their mean dispersion, the preliminary zero point correction, and the two sets of least squares transform coefficients are then printed.

The format is repeated for each block of data.

The statement POSSIBLE ERROR, together with the average intensity of the identified lines is printed out if less than 60% of the predicted lines could be identified.

Approximately 1 minute of 7094 time is needed for compilation and set-up. An additional 12 seconds is needed for a typical block of 1600 data points.

The output tape now contains spectrum, blocked as an input, but reduced to an intensity scale of 0.0 to 1.0. The tape is high density, 1605 words to a block. Words 1-1600 are the spectrum.

1601 is the number of points in the block;

1602 = block number;

1603 = beginning wavelength;

1604 = average continuum level

1605 = dispersion

DETAILED PROGRAM DESCRIPTION

The program, as listed in Appendix E, has been split into 26 sections, most of which are discussed in detail below. The comment cards with asterisks to the left are used to delineate the sections; each of these cards is numbered, with numbers running between 10 and 270. These are the "card numbers" referred to below. The flow chart (Figure 8) presents the logical flow of POPE; each section indicated by enclosure in a rectangle is discussed below.

1. Set-Up (Card #10)

- a. IFN03 is read off a data card.
- b. KKK, the number of files to be read, and IREEL (*), the file numbers, are read off a data card.
- c. The OP routine opens the files.
- d. Various quantities are initialized.
- e. The beginning wavelength of the spectrum, and FCRCT are read off a data card.
- f. Various other quantities are initialized.

NOTE: Statement 20002 is the "file transfer point".

2. Spectrum Tape Positioning (Cards #30, 50)

- a. Position spectrum tape to beginning of desired file (IREEL (JREEL)).

- b. Position search line tape.

-----Here comes the card input section-----

- c. Read STRTER, IBLSKP, DISP, LENDL from a data card.
- d. Skip IBLSKP blocks of spectrum.

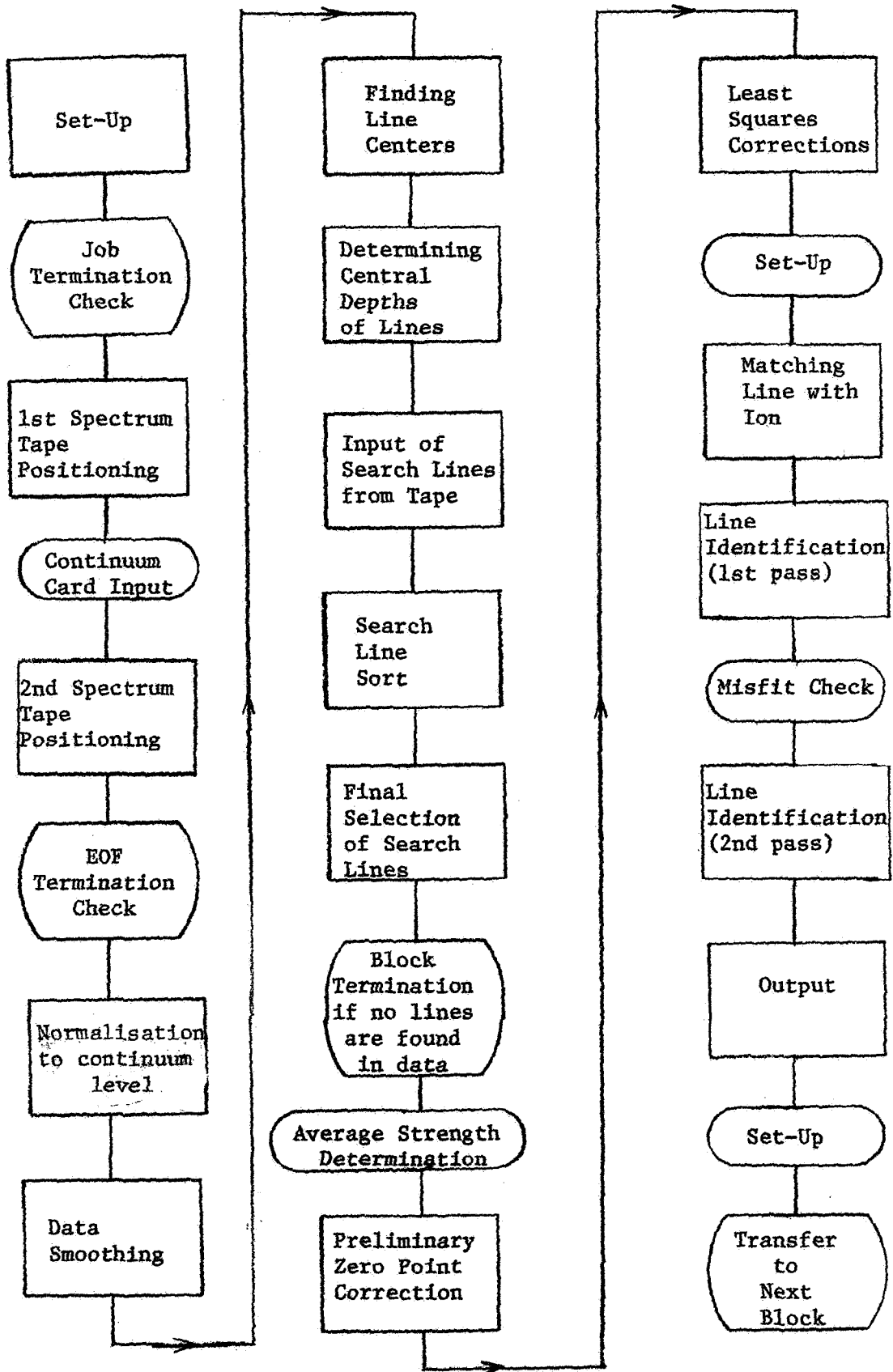


Figure 8.

3. Normalisation to continuum level

- a. Increment ILINE by the number of points in the previous block.

For each spectrum point:

- b. Calculate ZOO = the point number.
- c. If ZOO is less than the point number of the first continuum value, or greater than that of the last continuum value, then extrapolate to get ALAS, and go to statement 696.
- d. Otherwise, interpolate for ALAS, and go to 696.
- e. Determine if PT (wavelength of the data point) is within 0.06 Å of an even multiple of 20 Å. If it is, set ALASHD = ALAS.
- f. Make sure Y(J) is less than or equal to the continuum and greater than or equal to zero.
- g. Divide the spectrum value by the local continuum level (ALAS) to get the normalized intensity (X(J)).

4. Data Smoothing (Card #100)

- a. Initialize MINS(*) and NMINS.
- b. Average the nine points surrounding each data point, X(*).
- c. Assign this average to the same position in the XA(*) array.
- d. Calculate minimum and maximum values in XA(*).
- e. Calculate XSTOP for ensuing parts of program.

5. Line Center Finding (Card #110)

- a. Pick out all points, I, between the 6th and 1595th of a block such that: XA(I) less than XA(I-1) and, XA(I) less than or equal to XA(I+1).
- b. Store these points in MINS(*), and set NMINS = their total number.

- c. Between each pair of these suggested minima, find the maximum intensity. Compare this with 3% of the intensity at each of the suggested minima. If the maximum is in either case the lesser, determine which of the minima is the lesser. Remove the point number of the other minima from MINS(*), and decrease NMINS by 1.
- d. If less than 20 seconds of the time estimate is left, STOP.
- e. If IFNO3 = 1, skip (f).
- f. Compare the intensity at each suggested minima with XSTOP. If it is greater than XSTOP, remove it from MINS(*), and decrease NMINS by 1.

6. Central Depth Determination (Card #120)

- a. Store point number of minima in SLINE(*,3).
- b. Compute the wavelength of the \ast th minima and store in SLINE(*,2).
- c. Store NMINS in SLINE(200,2), and in NT.
- d. Calculate IVWMIN and IVWMAX.
- e. Find minimum value of unsmoothed normalized data between IVWMIN and IVWMAX.
- f. Store this value in SLINE(*+100,3).
- g. When this has been done for each minimum, store SLINE(*+100,3) in SLINE(*,3).

7. Tape Input of Search Lines (Card #140)

- a. Calculate the wavelength of the end of the block, and store it in XEND.
- b. If this is not the first block processed, skip (c), (d), and (e).
- c. Position line input tape to starting wavelength of block.

- d. If there are no lines on the tape before XEND, go to "block termination loop", below.
- e. Store 300 line wavelengths in WCALC(*,1), predicted strengths in WCALC(*,2), and ion identifications in ICALC(*). Then skip (f), (g), and (h).
- f. Delete lines of wavelengths shorter than XBEG from the tables. (II = the first line to be left).
- g. Advance all remaining lines II-1 spaces in the tables WCALC and ICALC.
- h. Fill in the last II - 1 spaces in these tables with lines read from the tape.
- i. Check TIME; if less than 20 seconds remain, STOP.

8. Search Line Sort (Card #150)

- a. XZ and XY are calculated, but not used until the next section.
- b. NN is set equal to 3/4 the number of lines found in the data.
- c. The WCALC array is stored in W. W(*,2) contains the number of the ICALC array pertaining to the *th number in the W array.
- d. The DO 210 loops sort the W array according to predicted strength -- strongest lines first.

9. Final Selection of Search Lines (Card #160)

- a. If less than 5 seconds remain on the internal clock, STOP.
- b. We are going to store the first NN lines of the W array in SLINE(*,1) subject to the conditions that:
 - i) Their wavelength is within the range of XBEG+1 Å to XEND - 1 Å.
 - ii) No selected line is within 0.2 Å of a stronger selected

line.

- c. The total number of search lines is stored in SLINE (200,1).

10. Preliminary Zero Point Correction (Card #190)

- a. Set $\epsilon = 0.15$.
- b. If this is the first block of data, the first try at Q is 1.5; if not, it is 0.15.
- c. Calculate the double sum, discussed above for each Q.
- d. Pick that Q which gives the maximum sum.
- e. Save this value in QNSAVE.

11. Least Squares Corrections (Card #200)

- a. Identify as many observed lines with search lines as possible, with an error tolerance in wavelength of 0.125 Å.
- b. Store the predicted wavelengths in D(1,*), and the error in wavelength in D(2,*).
- c. Correct wavelength of found lines (SLINE(*,2)) by the preliminary zero point correction.
- d. Convert D(1,*) wavelengths to units relative to XBEG.
- e. Change dispersion and zero point of wavelength by the least squares technique to minimize the errors (D(2,*)).
- f. Correct the observed wavelengths by 2/3 of the least squares corrections; correct XBEG and DISP likewise.

12. Matching of Line with Ion (Card #220)

This section goes through some "tracing back" of each search line

to find what position (ICALCX) in the ICALC array its ion is stored in. The code number of this ion is then stored in ICALCX.

13. Line Identification -- First Pass (Card #230)

- a. Set DCRIT = 0.075 \AA
- b. For each search line, see if an observed line lies within DCRIT of it in wavelength.
- c. If it does, write out the two wavelengths, the ion code, and the observed strength. Then see if the next observed line is also within DCRIT of the predicted wavelength. If it is, write out the same information for it. Add one to LCFD (ion code).
- d. If there is no identification, write out the predicted wavelength and the ion. Add 1 to XNMISS and to LCMS (ion code).

14. Line Identification -- Second Pass (Card #250)

For each found line not previously identified in the first pass:

- a. In the KSTAND array store the positions of all search lines from WCALC falling within 0.05 \AA of the found line's wavelength.
- b. If none are found, write the found wavelength and intensity; then go to card #260.
- c. Determine which of the lines represented in KSTAND is predicted to be the strongest. Write the predicted wavelength of this line, the found wavelength, ion, and observed strength.
- d. For each other line in KSTAND, write out the predicted and found wavelength, the ion, and the difference in predicted

strengths. Then go on to card #260.

15. Output Phase (Card #260)

- a. Assign values to the 1601 through 1605th points in the output array, X.
- b. Write X on tape.
- c. Write headings for the printed output.
- d. Write out the data points (normalized to intensity).

PROGRAM VARIABLES

ALAS = continuum level at a given point.

ALASHD = continuum level at the most recent multiple of 20 Å.

AONE = wavelength of the first data point in a line of output.

ATWO = wavelength of the last data point in a line of output.

AVERA = average strength of found lines in a block of data.

C(*) = the total residual in the double sum for
the *th tried value of Q.

CLO = map subroutine to close files.

CRACK(*) = amount to be added to AONE to get the wavelength of entry
column * (for print out only).

D(1,*) = wavelength of preliminary identified line, corrected by
the preliminary zero point correction.

(2,*) = predicted minus observed wavelength of this line.

DCRIT = tolerance between predicted and observed wavelength for
identification to be made.

DEX(*) = data point number for *th continuum value.

DIFF = actual difference between predicted and observed wavelengths
of a line.

DISP = dispersion of spectrum.

ϵ = A parameter used in the double sum.

ERMEAN = mean deviation of strength of found lines in a block.

FCRCT = constant to be subtracted from all intensities before they are
normalized (also subtracted from continuum value).

FMF = least squares correction to the dispersion.

FMS = least squares correction to the zero point of wavelength.

HELP(*,**,***,****) = subroutine to check that the variable * is greater
than or equal to ** and less than or equal to ***. If not, then it
prints out the number ****, and stops the program.

IBLOCK = block number in the spectrum.

IBLSKP = number of data blocks at beginning of file to be skipped.

ICALC(*) = ion causing search line.

IEOF = end of file indicator for spectrum tape ($\neq 0$ if file is at an end).

IERR = read error indicator for spectrum tape ($\neq 0$ if error occurred).

IFNO3 = input parameter. If =1, the XSTOP line discrimination criterion is deleted; if = 0, it is not.

ILINE = running total of point number at beginning of block - 1.

ILINL(*) = number in ICALC array pertaining to the *th entry in W array.

IREEL(*) = files to be worked.

IVWMIN = 1/4 way from found line to previous one (in point numbers).

IVWMAX = 1/4 way from found line to next one (in point numbers).

JBG = MINS(*-1)

JED = MINS(*)

JEOF = end of file indicator for search line tape ($\neq 0$ if EOF occurred).

JKEP = point number of the deeper minima between those at JBG and JED.

JWK = number of points in block.

KIF(*) = 1 if the found line * is identified; =0 if not.

KREEL = file number the tape is currently at.

KSTAND(*) = positions in the WCALC array of lines identified with found lines in the 2nd pass of line identification.

KTOTAL = number of significant entries in KSTAND array.

LCFD(*) = number of search lines of ion * identified.

LCMS(*) = number of search lines of ion * not identified.

LCOUT = subroutine to print what percentage of lines (of each ion) searched for were found in the spectrum.

LENDL = last block of file to be worked.

LSIN = number of continuum cards.

MINS(*) = point number of the *th minimum in the block.

NMINS = number of minima found in the block.

OP = map subroutine to open tapes.

Q = correction to wavelength zero point

QN = the Q which maximized the double sum preliminary zero point correction.

QNSAVE = QN.

RA5 = map subroutine to read one line's data from the search line tape.

RA6 = map subroutine to read one block of data from the spectrum tape.

SLINE(*,1) = wavelength of search lines
(*,2) = wavelength of lines found in spectrum
(*,3) = point number of line found in spectrum (during most of card #120 assembly), and the intensity of the found line for the rest of the program.

STRTER = pre-normalized intensity for zero inputted intensity (usually = FCRCT).

TI(*) = alphameric title used for output.

TIM(1) = time left, in seconds, on internal clock.

TIME = map subroutine which returns time left according to the internal clock.

VAL = minimum value in unsmoothed normalized data between IVWMIN and IVWMAX.

W(*,1) = predicted strength of search line.
(*,2) = *
(*,3) = wavelength of search line

WAN = map subroutine which backspaces two files on search line tape.

WB6 = map subroutine for data output on tape.

WIN = map subroutine which backspaces one file on search line tape.

WCALC(*,1) = wavelength of search line
(*,2) = predicted strength of search line

X(*) = *th normalized spectrum point. X(1601) = number of data points in block; X(1602) = block number; X(1603) = beginning wavelength; X(1604) = continuum level at center of block; X(1605) = dispersion.

XA(*) = smoothed normalized spectrum.

XBEG = beginning wavelength of block.

XBEGKP = beginning wavelength of first data block.

XBIG(*) = continuum level at input points.

XCA = An intensity value 3% above the value at JBG, a suggested minimum.

XCB = same as XCA, but for JED.

XLN = wavelength of line read from tape.

XM = maximum intensity value between two suggested minima, at JBG and JED.

XMN = minimum value of XA.

MXM = maximum value of XA.

XNMISS = number of search lines in a block not found.

XSTOP = 15% of (~~XM~~ - XMN).

Y(*) = unnormalized spectrum.

YF = maximum value in C(*).

ZOO = point number of block currently being worked on + ILINE.

PROGRAM NUMBER 1.
 READS MICROPHOTOMETER TAPE AND PERFORMS DATA FORMAT CONVERSION.

```

$IRSYS
$* MOUNT BELL 1 ON A 5 RING OUT 200 BPI REPEAT 200 BPI
$* PUT A SCRATCH ON R6 SAVE AT END
$* PUT THE A CHANNEL DENSITY SWITCH ON 800/200
$ATTACH A5
$AS SYSCK1,LOW
$ATTACH R6
$AS SYSUT6,HIGH
$IRSYS
$FXECUTE IRJOB
$ID RABELL*205/04/026*3M*200P*200C*R*TAPES
$IJOB GO,MAP,SOURCE,BASIC
$IIFTC TEST LIST,NODECK,M94,X#7
COMMON IN(9999),IEOF,IERR,IREEL,NER(2),NF,NR,LC,D(4),OUT(1000),
IN(2),IND,LI
CALL FIXIT
NF=1
NO=0
NR=0
NER(1)=0
NER(2)=0
READ(5,1)IRFFL
1 FORMAT(I2)
CALL OP
10 CALL RA5
C THIS GIVES US A RECORD
NR=NR+1
C NO IS THE INPUT RECORD NUMBER AND NR THE OUTPUT RECORD NUMBER
IF(IEOF.EQ.1)GO TO 70
IF(LI.GT.9999)GOTO60
N(1)=IN(LI)
CALL CONV
LO=2*LI
IF(N(2).EQ.0.AND.NFR(2).EQ.1)LO=LO-1
C LI IS THE INPUT RECORD LENGTH AND LO THE OUTPUT RECORD LENGTH
NER(1)=0
NFR(2)=0
LY=LO
9 WRITE(6,11)NF,NR,LI,LY,IND
11 FORMAT(10H1FILE NO.=,I5,13H RECORD NO.=,I5,15H INPUT LENGTH=,I5,
116H OUTPUT LENGTH=,I5,30X5HIND.=,I2)
IF(IERR.EQ.1)WRITE(6,12)(IN(I),I=1,LI)
12 FORMAT(17H HAD PARITY ERROR/(10(1X012)))
IERR=0
IERR=0
KK=0
LL=500
121 LEND=LL+KK
IF(LEND.GT.LI)LL=LI-KK

```

```

D020J=1,LL
I=J+KK
N(1)=IN(I)
CALL CONV
LC=2*I
L1=LC-1
OUT(L1)=N(1)
IF(NER(1).EQ.0)GO TO 15
NER(1)=0
LX=L1
14 WRITE(6,13)I,IN(I),OUT(L1),NO,LX
13 FORMAT(34H ERROR DETECTED IN INPUT WORD NO.=',I5,3X012,18H CONVERT
IED VALUE=',F6.0,22H IN OUTPUT RECORD NO.=',I5,11H WORD NO.=',I5)
IF(IND.EQ.0)GO TO 15
WRITE(7,16)NF,NO,LX
16 FORMAT(3I10)
15 IF(LC.GT.L0)GO TO 20
OUT(LC)=N(2)
IF(NER(2).EQ.0)GO TO 20
NER(2)=0
LX=LC
17 WRITE(6,13)I,IN(I),OUT(LC),NO,LX
IF(IND.EQ.0)GO TO 20
WRITE(7,16)NF,NO,LX
20 CONTINUE
IF(IND.EQ.1)GO TO 30
D025I=1,4
O(I)=OUT(I)
25 OUT(I)=0.
30 CONTINUE
IF(LL.EQ.500)GOTO35
LD=L0-2*KK+1
D031 I=LD,1000
31 OUT(I)=0.
35 CONTINUE
NO=NO+1
CALL WB6
135 WRITE(6,36)NF,NR,NO,LC,D
36 FORMAT(1X,4I6,4F6.0//)
K=1
L=20
37 IF(L.GT.(2*LL))GOTO39
WRITE(6,38)K,(OUT(LM),LM=K,L),L
38 FORMAT(1XI5,4(4X5F5.0),4XI5)
K=K+20
L=L+20
GO TO 37
39 L=2*LL
IF(K.EQ.L)GOTO101
WRITE(6,38)K,(OUT(LM),LM=K,L)
101 KK=KK+500
IF(KK.LT.L1)GOTO121
GOTO10
60 WRITE(6,61)
61 FORMAT(1X,50H GREAT LAMBORGHINI'S THE READ AREA IS TOO SMALL
65 CALL CLO
STOP
70 WRITE(6,71)NF,NR

```

```

71 FORMAT(10HIFILE NO. = ,I5,13H RECORD NO. = ,I5,8H WAS EOT)
   IREEL=IREEL-1
   IF(IREEL.EQ.0)GO TO 75
   IFOF=0
   IFRR=0
   NF=NF+1
   NR=0
   CALL WIN
   NO=0
   GO TO 10
75 WRITE(6,76)
76 FORMAT(29HOTHIS WAS THE LAST INPUT TAPE)
   GO TO 65
   END

```

```

$IBMAP MAPRO LIST,NODFCK,M94,X77
A5 FILE UNA5,CK1,RCD,BLK=9999,INPUT
B6 FILE UNB6,UT6,BIN,BLK=1007,OUTPUT
   ENTRY FIXIT
   ENTRY CONV
FIXIT SAVE 4
   CLA =00000020000000
   STD RTTCNT
   RETURN FIXIT
OP SAVE 4
   TSX .OPEN,4
   PZE A5
   TSX .OPEN,4
   PZE R6
   STZ IFOF
   STZ IFRR
   RETURN OP
CLO SAVE 4
   TSX .CLOSE,4
   PZE A5
   TSX .CLOSE,4
   PZE R6
   RETURN CLO
WB6 SAVE 4
   TSX .WRITE,4
   PZE R6,,0
   IOCT NF,,1007
   RETURN WB6
RA5 SAVE 4
AA TSX .READ,4
   PZE A5,,SYSDMP
   PZE FOF,,FRR
   IORT IN,,**
BB CAL *-1
   ANA =00777770000000
   ARS 18
   STO LI
   STZ IND
   CAL IN
   ANA =06000000000000
   TZE CC
   CLA =1
   STO IND
CC RETURN RA5

```

| | | |
|--------|--------|----------------|
| FOF | CLA | =1 |
| | STO | IEOF |
| | TRA | CC |
| FRR | CLA | =1 |
| | STO | IFRR |
| | TRA | AA |
| CONV | SAVE | 4,5,6,7 |
| | CAL | N |
| | ANA | =0171717171717 |
| | LRS | 18 |
| | STQ | N+1 |
| | LRS | 18 |
| A3A | AXT | 2,6 |
| | AXT | 2,5 |
| | CLA | IND |
| | TNZ | *+2 |
| | TIX | *+1,5,1 |
| | STZ | N+2,6 |
| | AXT | 3,7 |
| A4A | CLA | =0 |
| | LLS | 6 |
| | STQ | ZWI |
| | TZF | RL |
| | SUB | =10 |
| | TZF | CONT |
| | TMI | B |
| A5A | CLA | =1 |
| | STO | NFR+2,6 |
| | CLA | C900+3,7 |
| A6A | ADD | N+2,6 |
| | STO | N+2,6 |
| | TRA | CONT |
| R | ADD | =10 |
| | TIX | HP,5,1 |
| C | XCA | |
| | MPY | C100+3,7 |
| | XCA | |
| | TRA | A6A |
| CONT | LDQ | ZWI |
| | TIX | *+1,5,1 |
| | TIX | A4A,7,1 |
| | LDQ | N+1 |
| | TIX | A3A,6,1 |
| | RETURN | CONV |
| BL | CLA | IND |
| | TZE | CONT |
| | CLA | =1 |
| | STO | NFR+2,6 |
| | TRA | CONT |
| HP | SUR | =1 |
| | TN7 | A5A |
| | ADD | =1 |
| | TRA | C |
| WIN | SAVE | 4 |
| | TSX | •WFF,4 |
| | PZE | R6 |
| | RETURN | WIN |
| RTTCNT | BOOL | 1302 |


```
7WI   BSS      1
C900  DEC      900,90,9
C100  DEC      100,10,1
      CONTRL   //
      USE      //
IN     BSS      9999
IFOF  BSS      1
IFRR  BSS      1
IRFFL BSS      1
NFR   BSS      2
NF    BSS      7
OUT   BSS      1000
N     BSS      2
IND   BSS      1
LI    BSS      1
      USE      PREVIOUS
      END
$ENTRY      TFST
$DATA
1
!
```

PROGRAM NUMBER 2. FLUID DYNAMICS.
 REDUCES MICROPHOTOMETER DATA TO INTENSITY SCALE USING A WAVELENGTH DEPENDENT
 CALIBRATION CURVE.

```

$IRSYS
$RESTORE
$* MOUNT SCRATCH ON B6 RING IN
$* MOUNT SCRATCH ON B5 RING IN
$* MOUNT CSC 581 ON A6 RING OUT
$* RELEASE TAPE ON B6 AFTER RUN
$* PLOT TAPE = B5
$* PLOT AFTER RUN
$PAUSE
$ATTACH A6
$AS SYSUT5,HIGH
$ATTACH B6
$AS SYSUT6,HIGH
$IRSYS
$EXECUTE IRJOB
$ID BENGTSON,R.D., *303/01/011*9M*300R*TAPES $
$IBJOB GO,MAP,SOURCE,BASIC
$IBFTC THINK LIST,REF,NODECK,M94,XR7
      DIMENSION D1(20),D2(20),CO2(20)
      DIMENSION Z(1000),VB(1000),Y(1000),D(1000)
      DIMENSION VVV(1000)
      DIMENSION SE(37,20),NN(37),W(37),RAD(13)
      COMMON NF,NR,LC,C(4),X(1000),FN,RN,CL,E(4),S(1000),IEOF,IERR
C READ FILM CALIBRATION DATA
      READ(5,900) N
      READ(5,901) (W(I),NN(I),(SE(I,J),J=1,13),I=1,N)
      READ(5,920) (RAD(I),I=1,13)
      WRITE (6,912)
      WRITE(6,919) (W(I),(SE(I,J),J=1,13),I=1,N)
      WRITE(6,914) (RAD(I),I=1,13)
      CALL OP
200 IEOF=0
      IERR=0
      NRD = 32
      NILF = 0
      CALL RA6
      IF(IEOF.NE.0.OR.IERR.NE.0)GOTO 100
      LC=LC-1
C UNSCRAMBLE IDENTIFICATION ON TAPE
      CO=C(1)/100.+0.0001
      JJ=CO
      COD=JJ
      COO=C(1)-100.*COD
      CD=COO/10.
      JF=CD
      IF (LC.LT.500) GO TO 200
C WEDGE USED IS IDENTIFIED BY JJ
      IF (NR .GT.99) GO TO 200
      IF (JJ.EQ.1) GO TO 30

```

```

IF (JJ.EQ.2) GO TO 31
IF (JJ.FQ.3) GO TO 32
IF (JJ.FQ.4) GO TO 33
IF (JJ.FQ.5) GO TO 34
IF (JJ.FQ.6) GO TO 35
C   SHOT REGION IS IDENTIFIED BY JF
50  IF (JF.FQ.1) GO TO 300
    IF (JF.FQ.2) GO TO 400
    IF (JF.FQ.3) GO TO 400
    IF (JF.GF.4) GO TO 400
30  DSW = .0024
    SCALE = .008
    RSCAL = .02
    GO TO 50
31  DSW = .0052
    SCALE = .015
    RSCAL = .04
    GO TO 50
32  DSW = .0073
    RSCAL = .15
    SCALE = .05
    GO TO 50
33  DSW = .0098
    SCALE = .1
    RSCAL = .2
    GO TO 50
34  DSW = .0123
    SCALE = .15
    RSCAL = .4
    GO TO 50
35  DSW = .0151
    SCALE = .2
    RSCAL = .5
    GO TO 50
C   AVERAGE CLEAR FILM LEVEL AND STORE
300 JFA=JJ
    LLA=0
305 LL=0
    LLO=0
301 SE=0.0
    DO 302 J=1,20
    SE=SE+X(J)
    LL=LL+1
    IF (LL.EQ. LC) GO TO 390
302 CONTINUE
    LLO=LLO+1
303 QB=SE/20.0
    SE=0.0
    DO 304 J=1,20
    LLJ=LL-J+1
    IF (ABS(X(LLJ)-QB).GT.3.) X(LLJ)=QB
    SE=SE+X(LLJ)
304 CONTINUE
    LLOL=LLO+LLA
    VB(LLOL)=SE/20.0
    IF (ABS(QB-VB(LLOL)).LT.1.0) GO TO 301
    GO TO 303
390 J= LL-20*LLO

```

```

DF=J
QB=SF/DF
SE=0.0
DO 391 K=1, J
LLK=LL-K
IF (ABS(X(LLK)-QB).GT. 3.) X(LLK)=QB
SE=SE+X(LLK)
391 CONTINUE
VB(LLLOL+1)=SE/DF
LLA=LLLOL
NILE = NILE +1
IF (NILE.EQ.NRD) GO TO 350
IFRR=0
IOFF=0
CALL RA6
IF (IEOF.NE.0.OR.IERR.NE.0) GO TO 100
LC=LC-1
GO TO 305
350 LA=LLA-10
DO 355 J=11,LA
QB=0.0
DO 345 K=1,20
JAK=J+K-11
345 QB=QB+VB(JAK)
A=QB/20.
IF (ABS(VB(J)-A).GT.1.) VB(J)=A
355 CONTINUE
WRITE (6,907)
WRITE(6,910) (C(I),I=1,4)
DO 380 K=1, LLLOL,20
KO=K+19
WRITE (6,905) K,(VB(L),L=K,KO),KO
380 CONTINUE
GO TO 200
400 IF (JFA.NE. JJ) GO TO 600
C READ DATA FOR PARTICULAR SHOT
READ(5,909) T,BEGIN,CON
LLA=1
399 LLO=0
WRITE(6,911) T,BEGIN,CON
WRITE (6,903) DSW
JUMP = 0
401 P=NILE
SUMY = 0.0
PP=LLO
C CALCULATE WAVLENGTH ON FILM
DAM = BEGIN +.194*PP+155.*P
ECA=1.19E20/((DAM**5)*(EXP(3.79E4/DAM)-1.0))
EBBD=1.19E20/((DAM**5)*(EXP(1.438E7/(DAM*T))-1.0))
DO 410 K=1,N
IN=K
IF (DAM.LE.W(K)) GO TO 408
410 CONTINUE
408 NOD =NN(IN-1)
DO 413 K=1,NOD
D1(K)=SF(IN-1,K)-SF(IN-2,K)
D2(K)=SE(IN,K)-SE(IN-1,K)
DEL1=W(IN)-W(IN-1)

```

```

DEL2=W(IN-1)-W(IN-2)
DEL3=DAM-W(IN-1)
DFL4=DAM-W(IN-2)
C   FIND CORRECT CALIBRATION CURVE AT ONE WAVELENGTH
COR(K)=SE(IN-2,K)+DEL4*((D1(K)/DEL2)+DEL3*((D1(K)+D2(K))/(DEL1
1*(DEL1+DEL2)))-D1(K)/(DEL1*DEL2)))
413 CONTINUE
40  DO 420 JO=1,20
    LLO=LLO+1
C   FIND FILM DENSITY
DENS=X(LLO)-VB(LLA)
IF (DENS .LT. 0.01) DENS =0.01
X(LLO)=DENS
XX=X(LLO)*DSW
DO 414 KO=1,NOD
IF((XX-COR(KO)).GE.0.0) GO TO 415
414 CONTINUE
415 IF (KO.LE.3) KO=3
C   CONVERT DENSITY TO INTENSITY
DCO1=COR(KO)-COR(KO-1)
DCO2=COR(KO-1)-COR(KO-2)
DCO3=XX-COR(KO-2)
DCO4=XX-COR(KO-1)
RA=RAD(KO-2)+DCO3*((-.5/DCO2)+DCO4*((-1./(DCO1*(DCO1+DCO2)))+
1(.5/(DCO2*DCO1))))
Y(LLO)=CON*ECA*EXP(RA)
IF(Y(LLO).GE.EBBD) Y(LLO)=.98*EBBD
C   OPTICAL DEPTH CORRECTION
TAU = -ALOG(1.-Y(LLO)/EBBD)
Y(LLO)=EBBD*TAU
419 IF(Y(LLO).GE.(10.*SCALE)) JUMP =1
IF(LLA.EQ.LC) GO TO 450
SUMY = SUMY +Y(LLO)
420 CONTINUE
C   AVERAGE 20 INTENSITY POINTS FOR REDUCED PLOT
VVV(LLA)=SUMY/20.0
LLA= LLA +1
GO TO 401
450 IF (NILE .EQ.0) GO TO 452
459 Z(1)=1.0
DO 455 LL=2,LC
455 Z(LL)=Z(LL-1)+1.
Z(LC+1)=0.0
FLC=LC
Z(LC+2)=FLC/16.3
VVV(LLA)=Y(LLO)
CALL AXIS(0.0,0.0,6HLAMBDA,-6,16.3,0.0,Z(LC+1),Z(LC+2))
Y(LC+1)=0.0
Y(LC+2)=SCALE
IF(JUMP.EQ.1) Y(LC+2)=RSCAL
CALL AXIS(0.0,0.0,9HINTENSITY,+9,10.0,90.0,Y(LC+1),Y(LC+2))
FNILE =NILE
CALL NUMBER(0.5,0.5,.2,FNILE,0.0,0.0)
CALL LINE (Z,Y, LC, 1,0,0)
X(LC+1)=0.0
X(LC+2)=17.
CALL LINE (Z,X,LC,1,100,24)
CALL PLOT(16.3,0.0,-3)

```

```

WRITE(6,904) NR,LC,NILE
WRITE(6,913) DAM
WRITE(6,910) (C(I),I=1,4)
DO 709 K=1,LC,20
  KO=K+19
  WRITE(6,906) K,(Y(L),L=K,KO),KO
70) CONTINUE
  LLA=LLA+1
  NILE=NILE+1
  IF(NILE.EQ.NRD)GO TO 201
  IERR=0
  IOFF=0
470 CALL RA6
  IF(IEOF.NE.0.OR.IERR.NE.0) GO TO 100
  LC=LC-1
  GO TO 399
600 WRITE(6,908)
  GO TO 100
201 VVV(LLA+1)=0.0
  VVV(LLA+2)=RSCAL
  CALL PLOT(0.0,-11.0,-3)
  CALL PLOT(0.5,0.5,-3)
  CALL NUMBER(1.0,1.0,.2,C(1),90.0,0)
  CALL NUMBER(1.25,1.0,.2,C(2),90.0,0)
  CALL NUMBER(1.5,1.0,.2,C(3),90.0,0)
  CALL NUMBER(1.50,2.0,.2,C(4),90.0,0)
  Z(1)=BEGIN
  DO 203 L=2,1000
203 Z(L)=Z(L-1)+3.89
C   REDUCED PLOT (100 A/INCH)
  Z(LLA+1)=BEGIN
  Z(LLA+2)=100.
  CALL AXIS(0.0,0.0,6HLAMBDA,-6,14.0,0.0,Z(LLA+1),Z(LLA+2))
  CALL AXIS(0.0,0.0,9HINTENSITY,+9,10.0,90.0,VVV(LLA+1),VVV(LLA+2))
  CALL LINE(Z,VVV,LLA,1,0,0)
  CALL PLOT(17.5,0.0,-3)
  CALL PLOT(13.0,0.0,-3)
  GO TO 200
452 CALL PLOTS(0,1000)
  CALL PLOT(0.0,-11.0,-3)
  CALL PLOT(0.5,0.5,-3)
  CALL PLOT(2.0,0.0,-3)
  CALL NUMBER(1.0,1.0,.2,C(1),90.0,0)
  CALL NUMBER(1.25,1.0,.2,C(2),90.0,0)
  CALL NUMBER(1.5,1.0,.2,C(3),90.0,0)
  CALL NUMBER(1.50,2.0,.2,C(4),90.0,0)
  GO TO 459
900 FORMAT(12)
901 FORMAT(F5.0,I2,13F5.3)
902 FORMAT(1X,F10.0,2F10.4)
903 FORMAT(5H DSW=,F10.5)
904 FORMAT(10HIFILE NR.=,I5,8H LENGTH=,I5,6H NILE=,I5)
905 FORMAT(1XI5,4(2X5F5.1),2XI5)
906 FORMAT(1XI5,4(2X5F5.4),2XI5)
907 FORMAT(25HIBACKGROUND DATA AVERAGED)
908 FORMAT(35H TWO DIFFERENT DENSITY WEDGES USED )
909 FORMAT(F10.0,F10.0,F10.5)
910 FORMAT(4(3XF5.0))

```

```

911 FORMAT (1X,16H TEMPERATURE IS , F10.0, 7H LAMBDA,F10.0,
110H INTENSITY , F10.5 )
912 FORMAT (1H1)
913 FORMAT (8H LAMBDA=,F7.1)
914 FORMAT (/1X,13F6.3)
920 FORMAT (13F5.3)
919 FORMAT (/1X,F5.0,2X,13F6.3)
100 CALL CLO
    CALL PLOT(0,0,999)
    STOP
    FND
$IBMAP SPEC2 LIST,NODECK,M94,X#7
A6 FILE UNA6,UT5,BIN,BLK=1007,INPUT
B6 FILE UNB6,UT6,BIN,BLK=1007,OUTPUT
    ENTRY OP
    ENTRY CLO
    ENTRY RA6
    ENTRY WB6
OP SAVE 4
    TSX .OPEN,4
    PZF A6
    TSX .OPEN,4
    PZF R6
    RETURN OP
CLO SAVE 4
    TSX .CLOSE,4
    PZF A6
    TSX .CLOS,4
    PZF R6
    RETURN CLO
RA6 SAVE 4
A666 TSX .RFAD,4
    PZF A6,,SYSDMP
    PZF EOF6,,ERR6
    IOCT NF,,1007
A6A6 RETURN RA6
WB6 SAVE 4
    TSX .WRITE,4
    PZF R6
    IOCT NNF,,1007
    RETURN WB6
EOF6 CLA =2
    STO IEOF
    TRA A6A6
FRR6 CLA =2
    STO IFRR
    TRA A666
    CONTRL //
    USE //
NF BSS 1007
NNF BSS 7
Y BSS 1000
IEOF BSS 1
IFRR BSS 1
    USE PREVIOUS
    END
$DATA
37

```

PROGRAM NUMBER 3. FLUID DYNAMICS.
CHECKS CALIBRATION DATA USED IN PROGRAM NUMBER 2.

```

$IRSYS
$RESTORE
$*PLOT TAPE =P5
$* PLOT AFTER RUN
$* RELEASE AFTER PLOT
$PAUSE
$FXFCUTE      IRJOB
$ID      R.A.ROIG,*303/01/011*4M*TAPES
$IBJOB      GO,MAP,SOURCE,BASIC
$IBFTC MAIN LIST,REF, DECK,M94,XR7
      DIMENSION DENS(7),WAV(300),RA(300,7)
      DIMENSION ROK(13),SE(40,20),DIN(3),NN(40),DIM(3),W(40),RAD(13)
      DIMENSION DERIV(300),DEN(6),DFM(6)
      DIMENSION D1(200),D2(100),COR(100)
      DIMENSION DEN1(300),DEN2(300),DEN3(300),DEN4(300),DEN5(300)
      DIMENSION D(1000)
      DIMENSION DEN6(300),DFM1(30),DFM2(30)
899  FORMAT (6F12.4)
900  FORMAT (I2)
901  FORMAT (F5.0,I2,13F5.3)
902  FORMAT (I1,6F5.0)
903  FORMAT (1H0,37HTHE DENSITIES USED FOR COMPARISON ARE,6F6.2)
904  FORMAT (1H0)
905  FORMAT (1H0,6HLOG(I),IX,13F6.3)
906  FORMAT (35X,28HWAVELENGTH VS. LOG(I) MATRIX)
907  FORMAT (1H0,5HLAMBA)
908  FORMAT (1H0,44HTHE DENSITY USED FOR THE GAMMA COMPARISON IS,F6.3)
909  FORMAT (1H0,F6.0,6F12.4)
910  FORMAT(1H1,7HDENSITY,F11.3,6F12.3)
911  FORMAT (1X,8H-----)
912  FORMAT (1H1)
913  FORMAT (1X,5HLAMBA)
914  FORMAT (/1X,13F6.3)
919  FORMAT (/1X,F5.0,2X,13F6.3)
920  FORMAT (13F5.3)
C      N IS NUMBER OF WAVELENGTH CARDS
      READ(5,900) N
C      W = WAVFLNGTH
C      NN = NUMBER OF VALUES OF SE(NON-ZERO VALUES)
C      SE = CALIBRATION VALUES OF DENSTY
      READ(5,901)(W(I),NN(I),(SE(I,J),J=1,13),I=1,N)
C      RAD = LOG OF INTENSITIES CORRESPONDING TO ABOVE DENSITIFS
C      (IN DESCENDING ORDER)
      READ(5,920) (RAD(I),I=1,13)
      WRITE (6,912)
      WRITE (6,906)
      WRITE (6,905) (RAD(I),I=1,13)
      WRITE (6,911)

```

```

WRITE (6,907)
WRITE(6,919)(W(I),(SF(I,J),J=1,13),I=1,N)
C     NUM = NUMBER OF VALUES OF DENS (UP TO 6)
C     DENS = DENSITIES FOR CALIBRATION COMPARISON (IN ASCENDING ORDEP)
READ (5,902) NUM,(DENS(I),I=1,NUM)
C     DENEM = DENSITY FOR GAMMA COMPARISON
READ (5,899) DENFM
WRITE (6,904)
WRITE (6,903)(DENS(I),I=1,6)
WRITE (6,908) DENFM
C     THE NEXT FEW STEPS SET UP THE 3-DIMENSIONAL COMPARISON
CALL PLOTS(D,1000)
CALL PLOTG(0.,-11.,-3)
CALL PLOTG(10.,1.5,-3)
DO 5 I=1,13
5   ROK(I) = I
    T= RAD(1) -RAD(2)
    LO=0
    CALL AXIS(0.,0.,5HLN(I),-5,13.,0.,0.,1.)
    CALL AXIS(0.,0.,7HDENSITY,7,9., 90.,0.,.8)
    DEN(1)=0
    DEN(2)=10.30
    DEN(3)=0
    DEN(4)=0
    DEN(5)=1.
    DEN(1)=0
    DEN(2)=4.8
    DEN(3)=0
    DEN(4)=0
    DEN(5)=.8
    CALL LINE(DEN,DEN,3,1,0,0)
    CALL SYMBGL(10.25,6.25,.15,6HLAMBDA,0.,6)
    LAM=W(N)-W(1)+ 200.
    MAT=LAM/200
    FND=MAT
C     THESE STEPS SET UP THE WAVELENGTHS (EVERY 20 ANGSTROMS)
WAV(1) = W(1)
I=2
10  WAV(I) = WAV(I-1) + 20.
    IF (WAV(I)-W(N)) 20,30,30
20  I = I+1
    GO TO 10
30  CONTINUE
    WAV(I+1)= WAV(I)+20.
    KLOD = I
    WRITE (6,910) (DENS(JOB),JOB = 1,6)
    WRITE (6,911)
    WRITE (6,913)
    DO 430 I=1,KLOD
    LO=LO+1
    XJ = I-1
    DAM = WAV(I)
    DO 410 K=1,N
    IN=K
    IF(DAM.LE.W(K)) GO TO 408
410 CONTINUE
C     FOR A GIVEN WAVELENGTH, THE DENSITY VALUES ARE FOUND BY 3-POINT
C     INTERPOLATION

```

```

408   IF (IN.(T.3) IN = 3
      NOD = NN(IN-1)
      DEM1(NOD+1)=0
      DEM2(NOD+1)= 1.
      DEM2(NOD+2)= 2.
      DEM1(NOD+2)= .8
      DO 413 K=1,NOD
      D1(K)=SE(IN-1,K)-SE(IN-2,K)
      D2(K)=SE(IN,K)-SE(IN-1,K)
      DEL1=W(IN)-W(IN-1)
      DEL2=W(IN-1)-W(IN-2)
      DEL3=DAM-W(IN-1)
      DEL4=DAM-W(IN-2)
      COR(K)=SE(IN-2,K)+DEL4*((D1(K)/DEL2)+DEL3*(((D1(K)+D2(K))/(DEL1
1*(DEL1+DEL2))))-D1(K)/(DEL1*DEL2)))
      DEM1(K) =COR(K) + .005*XJ*3.
      DEM2(K) = ROK(K) + .0216*XJ*3.
413   CONTINUE
C     EVERY 100 ANGSTROMS THE 3-DIMENSIONAL COMPARISON IS PLOTTED
      IF (5*(LO/5)-LO) 40,390,40
390   CALL LINE(DEM2,DEM1,NOD,1,0,0)
40    DO 420 JO=1,20
      DEN5(7)=DENEM
      DO 420 J=1,7
      XX = DEN5(J)
      DO 414 KO=1,NOD
      IF((XX-COR(KO)).GE.0.0) GO TO 415
414   CONTINUE
C     FOR A GIVEN DENSITY AND WAVELENGTH, THE CORRESPONDING
C     LOG (I) IS FOUND BY 3-POINT INTERPOLATION (RA IS ANSWER)
41J   IF (KO.LE.3) KO=3
      DC01=COR(KO)-COR(KO-1)
      DC02=COR(KO-1)-COR(KO-2)
      DC03=XX-COR(KO-2)
      DC04=XX-COR(KO-1)
      RA(I,J)=RAD(KO-2)+DC03*((-T /DC02)+DC04*((-2.*T/(DC01*(DC01+DC02))
1)+( T/(DC02*DC01))))
      IF (J.EQ.1) DEN1(I) = RA(I,J)
      IF (J.EQ.2) DEN2(I) = RA(I,J)
      IF (J.EQ.3) DEN3(I) = RA(I,J)
      IF (J.EQ.4) DEN4(I) = RA(I,J)
      IF (J.EQ.5) DEN5(I) = RA(I,J)
      IF (J.EQ.6) DEN6(I) = RA(I,J)
      IF (J.EQ.7) X= RA(I,J)
420   CONTINUE
      DO 100 L=1,NOD
      IF (COR(L) .GT. DENEM) GO TO 100
C     DERIV = THE SLOPE OF THE ABOVE CURVE AT DENEM
      Y1=COR(L+1)
      Y2=COR(L)
      Y3=COR(L-1)
      X1=RAD(L+1)
      X2=RAD(L)
      X3=RAD(L-1)
      D=(Y2-Y1)/(X2-X1)
      C=(Y3-Y1)/((X3-X1)*(X3-X2))-(Y2-Y1)/((X2-X1)*(X3-X2))
      DERIV(I)=D+(2.*X-X1-X2)*C
      DERIV(I)=2.3026*DERIV(I)

```

```

GO TO 101
100 CONTINUE
101 WRITE (6,909) DAM,(RA(I,J),J=1,NUM)
430 CONTINUE
LAM = WAV(1) * .01
WV=LAM*100
C THE GAMMA COMPARISON IS PLOTTED HERE
CALL PLOTG(20.,0.,-3)
WAV(KL0D+1)=WV
WAV(KL0D+2)=200.
DERIV(KL0D+1)=0
DERIV(KL0D+2)=.8
CALL AXIS(0.,0.,6HLAMBDA,-6,END,0.,WV,200.)
CALL AXIS(0.,0.,5HGAMMA,5,8.,90.,0.,.8)
DEM(2)=END
DEM(3)=END
DEM(4)=0
DEM(5)=1.
DEN(1)=8.
DEN(2)=8.
DEN(3)=0
DEN(5)=1.
CALL LINE(DEM,DEN,3,1,0,0)
CALL LINE(WAV,DERIV,KL0D,1,0,0)
DEND=END/2. -1.
CALL SYMBOL(DEND,8.25,.25,16HGAMMA COMPARISON,0.,16)
DEAD =END +7.
CALL PLOTG(DEAD,0.,-3)
CALL LINE(DEM,DEN,3,1,0,0)
CALL AXIS(0.,0.,6HLAMBDA,-6,END,0.,WV,200.)
C THE DENSITY-WAVELENGTH COMPARISON IS PLOTTED HERE
RAX=RA(1,NUM)
RIN=RA(1,1)
DO 17 I=2,KL0D
RAX=AMAX1(RAX,RA(I,NUM))
RIN= AMIN1(RIN,RA(I,1))
17 CONTINUE
RIN=RIN-.5
MSC =(RAX-RIN +2.)/.8
SC=MSC
SC=SC*.1
CALL AXIS(0.,0.,6HLN (I),6,8.,90.,RIN,SC)
7 DEN6(KL0D+1)=RIN
DEN6(KL0D+2)=SC
6 DEN5(KL0D+1)=RIN
DEN5(KL0D+2)=SC
4 DEN4(KL0D+1)=RIN
DEN4(KL0D+2)=SC
3 DEN3(KL0D+1)=RIN
DEN3(KL0D+2)=SC
2 DEN2(KL0D+1)=RIN
DEN2(KL0D+2)=SC
1 DEN1(KL0D+1)=RIN
DEN1(KL0D+2)=SC
GO TO (16,15,14,13,12,11),NUM
11 CALL LINE(WAV,DEN6,KL0D,1,0,0)
12 CALL LINE(WAV,DEN5,KL0D,1,0,0)
13 CALL LINE(WAV,DEN4,KL0D,1,0,0)

```

```

14 CALL LINE(WAV,DEN3,KLOD,1,0,0)
15 CALL LINE(WAV,DEN2,KLOD,1,0,0)
16 CALL LINE(WAV,DEN1,KLOD,1,0,0)
    CALL PLOT(40,0,-3)
    CALL PLOT(0,0,999)
    STOP
    END

```

SDATA

20

```

3700.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
3800.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
3900.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
4000.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
4100.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
4100.112.2321.8001.4041.1160.9900.7560.5760.5040.4680.4680.468
4200.112.2501.8181.4401.1341.0440.7830.5940.5220.4860.4860.486
4300.112.2501.7821.3681.1161.0080.7740.5940.5220.4860.4860.486
4400.112.2411.7821.3861.1881.0710.8280.6480.5580.5220.5220.522
4500.112.2141.7641.3681.1701.0980.8460.6480.5580.5220.5220.522
4600.112.1961.7191.3141.1431.0890.7920.6120.5400.4860.4860.486
4700.112.1961.6831.3051.1341.0620.7700.6120.5040.4590.4590.459
4800.112.2141.6021.2331.0440.9540.6930.5400.4500.4140.4140.414
4900.112.052194581.1521.0080.8370.5760.4140.3420.3150.3150.315
5000.111.9941.3591.0800.9180.6930.4320.3060.2520.2520.2520.252
5100.112.0521.4761.0350.8100.6030.3690.2340.1980.1980.1980.198
5200.112.0881.4581.0800.8640.6120.3420.2160.1800.1800.1800.180
5300.112.0881.4581.0800.8640.6120.3420.2160.1800.1800.1800.180
5400.112.0881.4581.0800.8640.6120.3420.2160.1800.1800.1800.180
5500.112.0881.4581.0800.8640.6120.3420.2160.1800.1800.1800.180
7.0006.7506.5006.2506.0005.5005.0004.5004.0003.5003.000
6 .5 .9 1.1 1.5 1.9 2.3
.75

```

PROGRAM NUMBER 4. ASTRONOMY.
 CONVERTS MICROPHOTOMETER DATA TO INTENSITY SCALE.

```

$IRSYS
$*      MOUNT CSC 452 ON A6  RING OUT SAVE
$*      MOUNT BELL 17 ON B6  RING IN
$ATTACH      A6
$AS          SYSUT5,HIGH
$ATTACH      B6
$AS          SYSUT6,HIGH
$IRSYS
$EXECUTE     IRJOB
$ID  R.A.BELL*205/04/026*5M*250P*400C*TAPES
$IBJOB      GO,MAP,SOURCE,BASIC
$IBFTC SPEC1  LIST,REF,NODECK,M94,XR7
C          BRITISH GALAXY MAKERS OF NORTH AMERICA INC
C          THE FIRST 11-1 RECORDS ARE IGNORED, RECORDS 11 TO 12-1 CONTAIN
C          CALIBRATION DATA, RECORDS 12 TO 13-1 CONTAIN ZERO LEVEL DATA,
C          AND RECORDS 13 TO 14-1 ARE THE STELLAR DATA.
C          THE PROGRAMME ELIMINATES UP TO TEN RECORDS WHICH ARE SPURIOUS FOR
C          SOME REASON OR OTHER.  THESE RECORDS ARE DENOTED BY 17.
C          REMEMBER TO ATTEND TO THE FOLLOWING.....
C          (A) MOUNT APPROPRIATE IOCS TAPE ON A6.
C          (B) PREPARE DATA CARD FOR 11 TO IFILF.
C          (C) PREPARE ANOTHER FOR THE ROTTEN RECORDS.
C          (D) PREPARE THE CORRECTION CARDS.
          DIMENSION TITLE(80),ZZHT(20),XBIG(100),DEX(100),ALAS(100),I7(10)
          COMMON NF,NR,LC,DDD(4),X(1000),NNF,NNR,NLC,AD(4),Y(1000),IEOF,IERR
          1,LOLA,IDIFF,CALIB(200),TEMP(2000),NZERO,ZERO(5000),KLOLA,BUN(1000)
          2,XXHT(20)
C          SPECIAL MOUNT WILSON CALIBRATION PRE APRIL 10 1956
C          EIGHT STRIP CALIBRATION SPECTRUM
          DATA(ZZHT(1),I=1,8)/0.,.201,.395,.582,.777,.974,1.176,1.37/
          D0904 I=1,8
904 XXHT(1)=10.**ZZHT(1)
          CALL OP
          IEOF=0
          IERR=0
          READ(5,2500)(TITLE(I),I=1,80)
          READ(5,1007)IREEL
550 CALL BLACK
          READ(5,2500)(TITLE(I),I=1,80)
          WRITE(6,2501)(TITLE(I),I=1,80)
          READ(5,2500)(TITLE(I),I=1,80)
          WRITE(6,2501)(TITLE(I),I=1,80)
          READ(5,2500)(TITLE(I),I=1,80)
          WRITE(6,2501)(TITLE(I),I=1,80)
          READ(5,1029)I1,I2,I3,I4
          WRITE(6,1029)I1,I2,I3,I4
          WRITE(6,1056)
          READ(5,2500)(TITLE(I),I=1,80)
          READ(5,2500)(TITLE(I),I=1,80)
  
```

```

READ(5,1030)KKK,(I7(K),K=1,KKK)
I7(KKK+1)=I4+5
WRITE(6,2501)(TITLF(I),I=1,80)
WRITE(6,1031)(I7(K),K=1,KKK)
KA=1
IDIFF=0
LCOUNT=0
ILOLA=0
KLOLA=0
ICOUNT=0
JCOUNT=0
MINOR=1
NZFRO=0
READ(5,2500)(TITLF(I),I=1,80)
READ(5,2500)(TITLF(I),I=1,80)
READ(5,1001)NCORR,JVALUF,XCORR
DO 803 J=1,1000
803 ZFRO(J)=0.
10000 CALL RA6
IF(IERR.NE.0.OR.IFOF.NE.0)WRITE(6,1029)IEOF,IERR
IFOF=0
IERR=0
IF(NR.LT.I2)GOTO10000
1 IF(NR.LT.NCORR)GOTO20001
IF(NR.GT.NCORR)GOTO20002
X(JVALUE)=XCORR
20002 READ(5,1001)NCORR,JVALUE,XCORR
GOTO1
20001 IF(NR.NE.I7(KA))GOTO101
KA=KA+1
GO TO 10000
101 LOLA=LC
LOL=LOLA-1
DO 860 J=2,LOL
AB=X(J-1)-X(J)
AD=X(J+1)-X(J)
IF(((AB.GT.5.0).AND.(AD.GT.5.0)).OR.((AB.LT.-5.0).AND.(AD.LT.
9-5.0))) X(J)=(X(J+1)+X(J-1))/2.
860 CONTINUE
C 10001 IS THE ZERO SECTION
801 IF(NR.LT.I3)GOTO10001
C CONVERTING STELLAR DATA TO INTENSITY SCALE
IF(NR.GT.I3)GOTO551
S=0.
DO505 J=1,MANOR
505 S=S+ZERO(J)
XMANOR=MANOR
S=S/XMANOR
SS=S+3.
DO506 J=1,MANOR
IF(ZERO(J).GT.SS)ZFRO(J)=SS
506 CONTINUE
551 DO 840 J=1,LOLA
WK=KLOLA+J
IF(WK.GT.10.1)GO TO 48
HTZERO=ZERO(1)
GOTO4
48 WK=.05*WK+.5

```

```

IZFRO=WK
IF (IZERO.LT.NZFRO)GOTO3
ZERO(IZFRO)=ZERO(IZFRO-1)
ZERO(IZFRO+1)=ZERO(IZFRO)
3 GWK=IZFRO
HTZERO=ZERO(IZERO)+(WK-GWK)*(ZERO(IZERO+1)-ZERO(IZERO))
4 XHT=X(J)-HTZERO
IHT=XHT
WHT=IHT
WHT=XHT-WHT
IF (IHT.GT.0)GOTO8401
IHT=1
WHT=0.
WRITE(6,1058)NR,J,XHT
GOTO840
8401 IF (IHT.LT.174)GOTO840
IHT=NHT-1
WHT=1.
WRITE(6,1058)NR,J,XHT
40 X(J)=CALIB(IHT)+WHT*(CALIB(IHT+1)-CALIB(IHT))
BIGA=X(1)
INDX=1
DO804 J=1,LOLA
IF (X(J).LT.BIGA)GOTO804
BIGA=X(J)
INDX=J
804 CONTINUE
INDX=INDX+JCOUNT
JCOUNT=JCOUNT+LOLA
ICOUNT=ICOUNT+1
XBIG(ICOUNT)=BIGA
DEX(ICOUNT)=INDX
WRITE(6,1055)NR
WRITE(6,1098)NF
JAG=1
MBENZ=20
9 IF (MBENZ.GT.LOLA)GOTO11
WRITE(6,1054)JAG,(X(LANCIA),LANCIA=JAG,MBENZ),MBENZ
IF (MBENZ.EQ.LOLA)GOTO920
JAG=JAG+20
MBENZ=MBENZ+20
GOTO9
11 MBENZ=LOLA
WRITE(6,1054)JAG,(X(LANCIA),LANCIA=JAG,MBENZ)
920 DO921 J=1,LOLA
921 Y(J)=X(J)
IWK=LOLA+1
IF (IWK.GT.1000)GOTO901
DO902 J=IWK,1000
902 Y(J)=0.
901 CALL WR6
CALL MINIM
KLOLA=KLOLA+LOLA
IF (NR.EQ.14)GOTO20000
GOTO10000
C THIS IS THE ZERO DATA
10001 CALL SZERO
GOTO10000

```



```

200J0 WRITE(6,1050)ICOUNT
      WRITE(7,1007)ICOUNT
      WRITE(6,1006)(XBIG(J),J=1,ICOUNT)
      WRITE(6,1006)(DEX(J),J=1,ICOUNT)
      ICT=ICOUNT-1
      DO 870 J=2,ICT
870  ALAS(J)=(XBIG(J-1)+XBIG(J)+XBIG(J+1))/3.
      ALAS(1)=(2.*XBIG(1)+XBIG(2))/3.
      ALAS(ICOUNT)=(2.*XBIG(ICOUNT)+XBIG(ICT))/3.
      DO 871 J=2,ICT
871  XBIG(J)=(ALAS(J-1)+ALAS(J)+ALAS(J+1))/3.+0.5
      XBIG(1)=(2.*ALAS(1)+ALAS(2))/3.+0.5
      XBIG(ICOUNT)=(2.*ALAS(ICOUNT)+ALAS(ICT))/3.+0.5
      WRITE(7,2007)(XBIG(L),DEX(L),L=1,ICOUNT)
      WRITE(6,1006)(XBIG(L),DEX(L),L=1,ICOUNT)
20003 CALL RA6
      WRITE(6,10099)NF,NR,X(1),X(10),X(50)
      IF(IEOF.NE.0)GOTO20003
      IREEL=IREEL-1
      IEOF=0
      CALL LST
      IF(IREEL.NE.0)GOTO550
20007 CALL CLO
      STOP
1000 FORMAT((1X,10F8.2))
1001 FORMAT(2I3,F4.0)
1006 FORMAT(1X,10F8.2)
1007 FORMAT(I6)
1011 FORMAT(1X,5(F7.2,F9.2))
1029 FORMAT(1X,4I3)
1030 FORMAT(I3/(10I3))
1031 FORMAT(1X,10I4)
1050 FORMAT(1X,I6)
1054 FORMAT(1XI4,2(1X10F6.2),1XI4)
1055 FORMAT(12H1RECORD NO.=,I5)
1056 FORMAT(1H1,26H THESE RECORDS ARE REJECTS)
1058 FORMAT(1X,2I5,E16.8)
1098 FORMAT(1X,I6)
2007 FORMAT(5(F6.2,F7.0))
10099 FORMAT(1X,2I6,3F10.3)
2500 FORMAT(80A1)
2501 FORMAT(1X,80A1)
      END
$IBFTC LEVEL LIST,REF,NODECK,M94,XR7
      SUBROUTINE SZERO
      COMMON NF,NR,LC,DDD(4),X(1000),NNF,NNR,NLC,AD(4),Y(1000),IEOF,IERR
      1,LOLA,IDIFF,CALIB(200),TEMP(2000),NZERO,ZERO(5000),KLOLA,BUN(1000)
      2,XXHT(20)
      S=0.
      DO500 J=1,LOLA
      BUN(J)=X(J)
500  S=S+X(J)
      XLOLA=LOLA
      S=S/XLOLA+3.
503  SS=0.
      SSN=0.
      INDY=0
      DO501 J=1,LOLA

```

```

      IF(BUN(J).GT.S)GOTO516
      IF(BUN(J).GT..05)SSN=SSN+1.
      SS=SS+BUN(J)
      GOTO501
516  INDY=INDY+1
      BUN(J)=0.
501  CONTINUE
      IF(INDY.EQ.0)GOTO502
      S=SS/SSN+3.
      GOTO503
502  SS=S-3.
      DO 504 J=1,LOLA
      IF(X(J).GT.S)X(J)=SS
504  CONTINUE
10001 IF(NZERO.EQ.0)GOTO799
      DO 798 J=1,LOLA
      JQ=LOLA+IDIFF+1-J
      JX=LOLA+1-J
798  X(JQ)=X(JX)
      DO 797 J=1,IDIFF
797  X(J)=TEMP(J)
      LOLA=LOLA+IDIFF
      GOTO10
799  IWK=0
      10  MINOR=MINOR+IWK
      IWK=LOLA/20
      MANOR=MINOR+IWK-1
      NCOUNT=0
      DO 806 J=MINOR,MANOR
      NCOUNT=NCOUNT+1
      MO=20*(NCOUNT-1)+1
      MU=MO+19
      ZERT=0.
      DO 802 JX=MO,MU
802  ZERT=ZERT+X(JX)
806  ZERO(J)=ZERT/20.
      IDIFF=LOLA-20*IWK
      DO 805 J=1,IDIFF
      JQ=20*IWK+J
805  TEMP(J)=X(JQ)
      MCOUNT=MCOUNT+1
      JZERO=JZERO+IWK
      NZERO=MANOR
      RETURN
      END
$IBFTC CURVE  LIST,REF,NODECK,M94,XR7
      SUBROUTINE BLACK
      COMMON NF,NR,LC,DDD(4),X(1000),NNF,NNR,NLC,AD(4),Y(1000),IEOF,IERR
1  ,LOLA,IDIFF,CALIB(200),TEMP(2000),NZERO,ZERO(5000),KLOLA,BUN(1000)
2  ,XXHT(20)
      DIMENSION YYHT(20)
      READ(5,2500)(TITLE(I),I=1,80)
2500  FORMAT(80A1)
      READ(5,1005)(YYHT(I),I=1,8)
1005  FORMAT(8F4.0)
      M=2
      DO2 I=1,174
      X=I

```

```

4 IF(X.GT.YYHT(M))GOTO3
  CALIB(I)=XXHT(M-1)+(XXHT(M)-XXHT(M-1))*(X-YYHT(M-1))/(YYHT(M)-YYHT
  1(M-1))
  GOTO2
3 M=M+1
  IF(M.EQ.9)GOTO10
  GOTO4
2 CONTINUE
10 WRITE(6,1000)I
1000 FORMAT(1X,43H CHARACTERISTIC CURVE EXTRAPOLATED FROM PT ,I5)
  WK=CALIB(I-1)-CALIB(I-2)
  DO11 J=1,174
  11 CALIB(J)=CALIB(J-1)+WK
  WRITE(6,1002)
1002 FORMAT(1X,52H FOLLOWING IS CALIBRATION CURVE FOR MT WILSON PLATES)
  WRITE(6,1003)
1003 FORMAT(1X,38H COMPUTED AT UNIT DEFLECTION INTERVALS)
  WRITE(6,1006)(CALIB(I),I=1,174)
1006 FORMAT(1X,10F8.3)
10000 RETURN
      FND
$IBFTC FIT      LIST,REF,NODECK,M94,XR7
      SUBROUTINE MINIM
      DIMENSION XMINM(10),XIDA(10)
      COMMON NF,NR,LC,DDD(4),X(1000),NNF,NNR,NLC,AD(4),Y(1000),IEOF,IFERR
      1,LOLA,IDIFF,CALIB(200),TEMP(2000),NZERO,ZERO(5000),KLOLA,BUN(1000)
      2,XXHT(20)
      IF(LOLA.GT.100)GOTO50
      WRITE(6,1004)
1004 FORMAT(1X,24H THE RECORD IS TOO SHORT )
      GOTO51
      50 XONE=.03*X(1)
      DO10 I=2,LOLA
      WK=ABS(X(I)-X(1))
      IF(WK.GT.XONE)GOTO11
      10 CONTINUE
      11 ICONST=I-2
      WRITE(6,1002)ICONST,KLOLA
      WRITE(7,1003)ICONST
1002 FORMAT(1X,2I6)
      WLOLA=KLOLA
1003 FORMAT(I6)
      LLOLA=LOLA-20
      L=1
      6 XMIN=X(20)
      INDX=20
      DO1 I=20,LLOLA
      IF(XMIN.LE.X(I))GOTO1
      XMIN=X(I)
      INDX=I
      1 CONTINUE
      MINDX=INDX-10
      NINDX=INDX+10
      IF(MINDX.LT.20)MINDX=20
      IF(NINDX.GT.LLOLA)NINDX=LLOLA
      WY=.05*XMIN
      DO16 I=MINDX,NINDX
      IF(X(I).GT.4900.)GOTO17

```

```

16 CONTINUE
   GOTO18
17 DO20 I=MINDX,NINDX
20 X(I)=5000.
   GOTO6
18 DO2 I=MINDX,NINDX
   WK=ABS(X(I)-XMIN)
   IF(WK.LT.WY)GOTO3
2 CONTINUE
3 I STOR=I
   DO4 I=I STOR,NINDX
   WK=ABS(X(I)-XMIN)
   IF(WK.GT.WY)GOTO15
4 CONTINUE
15 JSTOR=I-1
   IWK=I STOR+JSTOR
   XWK=IWK
   XWK=XWK/2.
   IWK=IWK/2
   XMINM(L)=X(IWK)
   XIDA(L)=XWK+WLOLA
   DO5 I=MINDX,NINDX
5 X(I)=5000.
   L=L+1
   IF(L.LE.7)GOTO6
C ARRANGING IN SORTED ORDER,XIDA=DISTANCE,XMINM=RESID INTENSITY
   NN=1
23 SORTXX=XIDA(NN)
   INDX=NN
   DO22 I=NN,7
   IF(XIDA(I).GT.SORTXX)GOTO22
   INDX=I
   SORTXX=XIDA(I)
22 CONTINUE
   TEMP1=XIDA(NN)
   TEMP2=XMINM(NN)
   XIDA(NN)=XIDA(INDX)
   XMINM(NN)=XMINM(INDX)
   XIDA(INDX)=TEMP1
   XMINM(INDX)=TEMP2
   NN=NN+1
   IF(NN.LT.7)GOTO23
   WRITE(6,1000)(XMINM(L),L=1,7)
   WRITE(6,1000)(XIDA(L),L=1,7)
   WRITE(7,1001)(XMINM(L),L=1,7)
   WRITE(7,1001)(XIDA(L),L=1,7)
1001 FORMAT(7F9.2)
1000 FORMAT(1X,7F11.3)
   DO8 L=1,7
   XIDA(L)=XIDA(L)+WLOLA
8 XIDA(L)=4000.+XIDA(L)*.01478
   WRITE(6,1000)(XIDA(L),L=1,7)
51 RETURN
   FND
$IRMAP SPEC2 LIST,NODECK,M94,X#7
A6 FILE UNA6,UT5,BIN,BLK=1007,INPUT
B6 FILE UNB6,UT6,BIN,BLK=1000,OUTPUT
OP SAVE 4

```

```

      TSX      .OPEN,4
      PZF      A6
      TSX      .OPEN,4
      PZF      R6
      RETURN   OP
LST   SAVE    4
      TSX      .WEF,4
      PZE      R6
      RETURN   LST
CLO   SAVE    4
      TSX      .CLOSE,4
      PZF      A6
      TSX      .CLOSE,4
      PZF      R6
      RETURN   CLO
RA6   SAVE    4
A666  TSX      .READ,4
      PZF      A6,,SYSDMP
      PZF      EOF6,,ERR6
      IOCT     YF,,1007
A6A6  RETURN   RA6
WB6   SAVE    4
      TSX      .WRITE,4
      PZF      R6
      IOCT     Y,,1000
      RETURN   WB6
EOF6  CLA     =2
      STO     IFOF
      TRA     A6A6
FRR6  CLA     =2
      STO     IERR
      TRA     A666
      CONTRL  //
      USE     //
YF    RSS     1007
NMF   RSS     7
Y     RSS     1000
IFOF  RSS     1
IERR  RSS     1
      USE     PREVIOUS
      END

```

\$DATA

THE NEXT CARD IS THE NUMBER OF FILES TO BE PROCESSED

000002

TRANSMISSION VALUES OF THE CALIBRATION STRIPS

002.008.014.026.048.080.127.147.

THE NEXT CARD IS I1,I2,I3,I4, WHERE THE FIRST I1-1 RECORDS ARE IGNORED,
RECORDS I1 TO I2-1 ARE CALIBRATION DATA ,RECORDS I2 TO 93 1 ARE ZERO DATA
AND RECORDS I3 TO I4-1 ARE STELLAR DATA . FORMAT(1X,4I3)

002004016026

002004016026200200200

THE NEXT CARD IS N,(I7(K),K=1,N) WHERE THE N RECORDS FORMING THE ARRAY I7
OMITTED FORMAT(I3/(10I3))

001

200

THE NEXT CARDS CORRECT DIGITISER ERRORS DO NOT CORRECT RECORDS WITH RECORD
NUMBER .LT.I2 RECORD NUMBER,WORD NO, CORRECT VALUE FORMAT(2I3,F4.0

017676117.

200200001.

TRANSMISSION VALUES OF THE CALIBRATION STRIPS

002.004.005.016.030.071.103.142.

THE NEXT CARD IS I1,I2,I3,I4 WHERE THE FIRST I1-1 RECORDS ARE IGNORED,
RECORDS I1 TO I2-1 ARE CALIBRATION DATA ,RECORDS I2 TO I3-1 ARE ZERO DATA
AND RECORDS I3 TO I4-1 ARE STELLAR DATA . FORMAT(1X,4I3)

002003055102

THE NEXT CARD IS N,(I7(K),K=1,N) WHERE THE N RECORDS FORMING THE ARRAY I7 ARE
OMITTED FORMAT(I3/(10I3))

004

003027054093

THE NEXT CARDS CORRECT DIGITISER ERRORS DO NOT CORRECT RECORDS WITH RECORD
NUMBER .LT.I2 RECORD NUMBER,WORD NO, CORRECT VALUE FORMAT(2I3,F4.0

200200001.

PROGRAM NUMBER 5. ASTRONOMY.
 IDENTIFIES LINES IN STELLAR SPECTRA.

```

$IRSYS
$* MOUNT CSC 2687 ON A6 RING OUT SAVE
$* MOUNT CSC 1769 ON B6 RING IN HIGH SAVE
$* MOUNT CSC 307 ON A5 RING OUT SAVE
$*
$PAUSE
$ATTACH A6
$AS SYSUT5,HIGH
$ATTACH B6
$AS SYSUT6,HIGH
$IRSYS
$EXECUTE IRJOB
$ID DMGOTT*205/02/202*7M*200P*200C*B*TS
$IRJOB GO,MAP
$IRETC POPE LIST,M94,XR7
      DIMENSION SLINE(200,3),CRACK(20),WCALC(300,2),LCFD(500),
      *LCMS(500),W(300,3),ILINL(200),C(300),D(2,25),XA(1610),MINS(200),
      *KIF(200),KSTAND(200),ICALC(300),XMISSV(40,2),WAVE1(150),X1(150),
      *WAVE2(150),X2(150),TI(12),XBIG(100),DEX(100),IREFL(10)
      EQUIVALENCE(SLINE(1,1),Y(1))
      EQUIVALENCE(Y(601),W(1,1))
      COMMON Y(1605),IEOF,IERR,XLN,IDLN,SLN,IDLNX,JEOF,ZLINE,IDLONE,FRAO
      *,FG,FH,TIM(2),X(1610)
      COMMON/LC/LCFD,LCMS
  
```

C BEGIN SET-UP

***** 010 *****

```

      READ(5,6679) IFN03
      READ(5,1011)KKK,(IREEL(K),K=1,KKK)
      CALL HELP(KKK,1,10,100)
      CALL OP
      IEOF=0
      IERR=0
      JEOF=0
      KRFEI=1
      JREEL=1
20002 CONTINUE
      READ(5,303) XBEG,FCRCT
      NNNNNN=1
      DO 502 I=1,200
502 KIF(I)=0
      XBEGKP=XBEG
      JSTST=0
      DO 610 I=1,500
610 LCMS(I)=0
      QQ=0.0
      ILINE=0
      SUM=0.
  
```

IBLOCK=1

```

C          JOB TERMINATORS                                ***** 020 *****
          IF(XBEG.LT.0.0) GO TO 100
20012 IF(JREEL.GT.KKK)GOTO100

C          BEGIN FIRST SPECTRUM TAPE POSITIONING          ***** 030 *****
          CALL HELP(JREEL,1,10,200)
          IF(IRFFL(JRFFL).NE.KRFFL) GO TO 1101
          GO TO 1112
1101 CALL RA6
          IF(IEOF.EQ.0) GO TO 1101
          IEOF=0
          KRFFL=KREEL+1
          GO TO 20012
1112 IF(JEOF.NE.0) GO TO 850
          CALL WIN
          GO TO 851
850 CALL WAM
851 JEOF=0
          IEOF=0
          IERR=0
          JRFFL=JRFFL+1

C          BEGIN CONTINUUM CARD INPUT                    ***** 040 *****
102 READ(5,1009)(TI(I),I=1,12)
105 READ(5,2006)LSIN
          CALL HELP(LSIN,2,100,300)
          DO 691 J=1,LSIN
691 READ(5,2008)XBIG(J),DEX(J)

C          BEGIN SECOND SPECTRUM TAPE POSITIONING        ***** 050 *****
          READ(5,2100) STRTER,IBLSKP,DISP,LENDL
          DISPHD=DISP
          IF(IBLSKP.EQ.0) GO TO 2102
          DO 2101 III=1,IBLSKP
          CALL RA6
2101 CONTINUE
2102 CONTINUE
10000 CONTINUE

C          STATEMENT NUMBER 104 IS THE 'BLOCK TRANSFER POINT' ***** 060 *****
104 CALL RA6

          END OF FILE TERMINATION CHECK                ***** 070 *****
          IF(IBLOCK.GT.LENDL) IEOF=3
          IF(IEOF.NE.0) GO TO 103

          BEGIN NORMALIZATION TO CONTINUUM LEVEL      ***** 080 *****
          IF(IBLOCK.NE.1) ILINE=ILINE+JWK
          JWK=Y(1601)

```



```

JWKHD=JWK
CALL HFLP(JWK,1,1600,400)
DO 4 J=1,JWK
ZOO=ILINE+J
IF(ZOO.LT.DEX(1)) GO TO 699
IF(ZOO.GT.DEX(LSIN)) GO TO 698
IPS=0
697 IPS=IPS+1
CALL HELP(IPS,1,100,500)
IF(DEX(IPS).LT.ZOO)GO TO 697
ALAS=XBIG(IPS-1)+(XBIG(IPS)-XBIG(IPS-1))*(ZOO-DEX(IPS-1))/
7(DEX(IPS)-DEX(IPS-1))
GO TO 696
699 ALAS=XBIG(1)-(XBIG(2)-XBIG(1))*(DEX(1)-ZOO)/(DEX(2)-DEX(1))
GO TO 696
698 ALAS=XBIG(LSIN)+(XBIG(LSIN)-XBIG(LSIN-1))*(ZOO-DEX(LSIN))/
6(DEX(LSIN)-DEX(LSIN-1))
696 PT=XBEG+FLOAT(J)*DISP
IPT=PT*100.0+0.5
IPTA=IPT-(IPT/1000)*1000
IF(IPTA.GT.6) GO TO 860
IPTB=(IPT-(IPT/10000)*10000-IPTA)/1000
IF(IPTB.EQ.(IPTB/2*2))GO TO 860
ALASHD=ALAS-FCRCT
860 CONTINUE
IF(Y(J).GT.ALAS)Y(J)=ALAS
IF(Y(J).LT.0.001) Y(J)=STRTER
4 X(J)=(Y(J)-FCRCT)/(ALAS-FCRCT)
GO TO 101

```

C EOF BOOKKEEPING -- NOT PART OF MAIN PROGRAM FLOW

***** 090 *****

```

103 CONTINUE
IEOF=0
KREEL=KREEL+1
CALL LCOUT
GO TO 20012
100 CALL CLO
STOP
101 CONTINUE

```

C DATA SMOOTHING

***** 100 *****

```

DO 135 IU=1,200
135 MINS(IU)=0
NMIN=0
IM=-4
IP=+4
IMP=-IM+1
IPM=JWK-IP
IRUN=IP-IM+1
CALL HELP(IMP,1,1600,1)
CALL HELP(IPM,1,1600,101)
DO 110 I=IMP,IPM
SSUM=0.
IG=I+IM-1
DO 111 K=1,IRUN
KA=IG+K

```

```

CALL HELP(KA,1,1600,201)
111 SSUM=SSUM+ X(KA)
112 XA(I)=SSUM/FLOAT(IRUN)
110 CONTINUE
  XMN=1
  XMX=0
  JWKM=JWK-50
  CALL HELP(JWK,1,1600,2)
  DO 270 I=50,JWK
  IF(XA (I).GT.XMX) XMX=XA (I)
270 IF(XA (I).LT.XMN) XMN=XA (I)
  XSTOP=XMX-0.15*(XMX-XMN)

```

C BEGIN LINE CENTER FINDING

***** 110 *****

```

  J=1
  DO 114 I=6,1595
  IM=I-1
  IP=I+1
  IF(XA (I).LT.XA (IM).AND.XA (I).LF.XA (IP)) GO TO 115
  GO TO 116
115 MINS(J)=I
  J=J+1
116 CONTINUE
114 CONTINUE
  NMINS=J-1
  CALL HELP(NMINS,1,200,3)
  DO 117 J=2,NMINS
121 CONTINUE
  JRG=MINS(J-1)
  JED=MINS(J)
  XM=-1.1
  CALL HELP(JBG,1,1600,4)
  CALL HELP(JED,1,1600,4)
  DO 118 JA=JRG,JED
  IF(XA (JA).GE.XM) XM=XA (JA)
118 CONTINUE
  XCA=XA (JRG)+.03
  XCB=XA (JED)+.03
  IF(XM.GT.XCA.AND.XM.GT.XCB) GO TO 117
  JKEP=JBG
  IF(XCA.GE.XCB) JKFP=JED
  MINS(J-1)=JKEP
  IF(J.GE.NMINS) GO TO 119
  JP=J+1
  CALL HELP(JP,1,200,5)
  CALL HELP(NMINS,1,200,5)
  DO 120 JR=JP,NMINS
120 MINS(JR-1)=MINS(JR)
  CALL TIME
  IF(TIM(1).LT.20.0) STOP
  NMINS=NMINS-1
  GO TO 121
117 CONTINUE
119 CONTINUE
124 CONTINUE
  IF(IFNO3.EQ.1) GO TO 6678
  CALL HELP(NMINS,1,200,6)

```

```

DO 122 I=1,NMINS
MINSM=MINS(I)
CALL TIME
IF(TIM(1).LT.20.0) STOP
CALL HELP(MINSM,1,1600,7)
IF(XA (MINSM).LT.XSTOP) GO TO 125
IP=I+1
IF(I.NE.NMINS) GO TO 140
MINS(I)=0
NMINS=NMINS-1
GO TO 141
140 CONTINUE
CALL HFLP(IP,1,200,8)
CALL HFLP(NMINS,1,200,8)
DO 133 IA=IP,NMINS
133 MINS(IA-1)=MINS(IA)
NMINS=NMINS-1
GO TO 124
125 CONTINUE
122 CONTINUE
6678 CONTINUE
141 CONTINUE

```

C BEGIN CENTRAL DEPTH DETERMINATION

***** 120 **

```

CALL HELP(NMINS,1,200,9)
DO 123 I=1,NMINS
SLINE(I,3)=MINS(I)
123 SLINE(I,2)=XBEG +FLOAT(MINS(I))*DISP
SLINE(200,2)=NMINS
NT=SLINE(200,2)
CALL HFLP(NT,1,100,10)
DO 173 I=1,NT
IVWMIN=(SLINE(I,3)*3.0+SLINE(I-1,3))/4.0
IF(I.EQ.1) IVWMIN=5
IVWMAX=(SLINE(I,3)*3.0+SLINE(I+1,3))/4.0
IF(I.EQ.NT) IVWMAX=JWK-10
VAL=2.00
CALL HELP(IVWMIN,1,1600,11)
CALL HELP(IVWMAX,1,1600,11)
DO 174 J=IVWMIN,IVWMAX
IF(VAL.LT.X (J)) GO TO 174
VAL=X(J)
174 CONTINUE
173 SLINE(I+100,3)=VAL
DO 176 I=1,99
176 SLINE(I,3)=SLINE(I+100,3)

```

C BEGIN SEARCH LINE SUB-ASSEMBLY

***** 130 **

IF(IBLOCK.NE.1) GO TO 200

C BEGIN TAPE INPUT OF SEARCH LINES

***** 140 **

```

NPTS=JWK
XEND=XBEG+FLOAT(NPTS)*DISP
CALL RA5
ILN=XLN
202 IF(XLN.GE.XBEG) GO TO 201
CALL RA5
GO TO 202

```

```

201 CONTINUE
  IF(XEND.GE.XLN) GO TO 203
  NLINES=0
  WRITE(6,500)
  GO TO 559
203 CONTINUE
  DO 204 I=1,300
  CALL RA5
  WCALC(I,1)=XLN
  WCALC(I,2)=SLN
204 ICALC(I)=IDLN
  GO TO 300
200 CONTINUE
  DO 205 I=1,300
  IF(WCALC(I,1).GE.XBEG) GO TO 206
205 CONTINUE
206 II=I
  II=II-20
  IF(II.LE.0) II=2
  J=1
  CALL HELP(II,1,300,12)
  DO 207 I=II,300
  CALL HELP(J,1,300,13)
  WCALC(J,1)=WCALC(I,1)
  WCALC(J,2)=WCALC(I,2)
  ICALC(J)=ICALC(I)
  J=J+1
207 CONTINUE
  CALL HELP(J,1,300,14)
  WCALC(J,1)=XLN
  WCALC(J,2)=SLN
  ICALC(J)=IDLN
  J=J+1
  CALL HELP(J,1,300,15)
  DO 208 I=J,300
  CALL RA5
  WCALC(I,1)=XLN
  WCALC(I,2)=SLN
  ICALC(I)=IDLN
208 CONTINUE
300 CONTINUE
  CALL TIME
  IF(TIM(1).LT.20.) STOP
  NLINES=300

```

C BEGIN SEARCH LINE SORT (BY STRENGTH)

***** 150 ***

```

XZ=XEND-1.0
XY=XBEG+1.0
NN=FLOAT(NMINS)*0.75+0.5
DO 209 I=1,300
  W(I,1)=WCALC(I,2)
  W(I,3)=WCALC(I,1)
209 W(I,2)=FLOAT(I)+0.1
  DO 210 I=1,299
  IP=I+1
  DO 210 J=IP,300
  IF(W(I,1).GT.W(J,1)) GO TO 211

```

```

DO 216 M=1,3
WT=W(I,M)
W(I,M)=W(J,M)
W(J,M)=WT
216 CONTINUE
211 CONTINUE
210 CONTINUE

```

C BEGIN FINAL SELECTION OF SEARCH LINES

***** 160 **

```

J=1
L=0
CALL TIME
IF(TIM(1).LT.5.0) STOP
212 CONTINUE
CALL HFLP(J,1,300,16)
IF(W(J,3).LT.XY.OR.W(J,3).GT.XZ) GO TO 213
IF(L.NE.0) GO TO 214
SLINE(1,1)=W(J,3)
ILINL(1)=W(J,2)
L=1
GO TO 215
214 CONTINUE
CALL HELP(L,1,200,18)
DO 217 M=1,L
CALL HELP(J,1,300,17)
IF(ABS(W(J,3)-SLINE(M,1)).LT.0.2) GO TO 218
217 CONTINUE
GO TO 219
213 CONTINUE
218 J=J+1
IF(J.GT.300) GO TO 220
GO TO 212
219 CONTINUE
CALL HELP(J,1,300,19)
CALL HELP(L,1,200,20)
SLINE(L+1,1)=W(J,3)
ILINL(L+1)=W(J,2)
L=L+1
J=J+1
IF(J.GT.300) GO TO 220
215 CONTINUE
IF(L.GE.NN) GO TO 220
GO TO 212
220 CONTINUE
SLINE(200,1)=NN
559 CONTINUE

```

C BLOCK TERMINATION IF NO LINES IN DATA

***** 170 **

```

IF(NLINES.NE.0) GO TO 704
IBLOCK=IBLOCK+1
XBEG=XBEG+DISP*FLOAT(JWK)
ILINE=ILINE+JWK
GO TO 104
704 CONTINUE

```

C DETERMINATION OF AVERAGE STRENGTH OF FOUND LINES

***** 180 **

```

SUM=0.0
NA=SLINE(200,2)
CALL HELP(NA,1,200,21)
DO 312 IA=1,NA
312 SUM=SUM+SLINE(IA,3)
AVERA=SUM/FLOAT(NA)
SUM=0
DO 313 IA=1,NA
313 SUM=SUM+ABS(AVERA-SLINE(IA,3))
ERMEAN=SUM/FLOAT(NA)
WRITE(6,314) AVFRA,FRMEAN

```

C BEGIN PRELIMINARY ZERO POINT CORRECTION

***** 190 ***

```

F=0.15
NON=SLINE(200,1)
IF(NON.LT.5) GO TO 187
NTW=SLINE(200,2)
GF=0.
Q=0.15
L=136
IF(IRBLOCK.EQ.1) Q=1.5
IF(IRBLOCK.EQ.1) L=1
169 SUM=0
CALL HELP(NTW,1,200,22)
CALL HELP(NON,1,200,23)
DO 170 I=1,NTW
DO 171 J=1,NON
171 SUM=SUM+1.0/(E+ABS(SLINE(J,1)-SLINE(I,2)+Q))
170 CONTINUE
CALL HELP(L,1,300,24)
C(L)=SUM
L=L+1
Q=Q-0.01
IF(L.EQ.301) GO TO 1200
IF(L.EQ.166.AND.IRBLOCK.NE.1) GO TO 1200
GO TO 169
1200 YF=-9999.
DO 1201 I=1,300
IF((I.LT.136.OR.I.GT.165).AND.IRBLOCK.NE.1) GO TO 1201
IF(C(I).LT.YF) GO TO 1201
YF=C(I)
QN=1.51-0.01*FLOAT(I)
1201 CONTINUE
WRITE(6,1202) QN
QNSAVE=QN

```

C BEGIN LEAST SQUARES CORRECTIONS

***** 200 **

```

1212 NT=0
DO 1203 I=1,NON
DO 1204 J=1,NTW
HS=SLINE(I,1)-SLINE(J,2)+QN
HSV=ABS(HS)
IF(HSV.GT.0.125) GO TO 1205
NT=NT+1
CALL HELP(NT,1,25,25)

```

```

D(1,NT)=SLINE(I,1)
D(2,NT)=HS
GO TO1203
1205 CONTINUE
1204 CONTINUE
1203 CONTINUE
14752 CONTINUE
DO1206 I=1,NTW
SLINE(I,2)=SLINE(I,2)-QN
1206 CONTINUE
CALL HELP(NT,1,25,26)
DO1207 I=1,NT
D(1,I)=D(1,I)-XBEG
1207 CONTINUE
SA=0.
SB=0.
SC=0.
SD=0.
DO1208 I=1,NT
SA=SA+D(1,I)
SB=SB+D(2,I)
SC=SC+D(1,I)**2
1208 SD=SD+D(1,I)*D(2,I)
FMF=SD-SA*SB/FLOAT(NT)
FMF=FMF/(SC-SA*SA/FLOAT(NT))
FMFSV=FMF
FMS=(SB-FMF*SA)/FLOAT(NT)
WRITE(6,1209) FMF,FMS
DO1211 I=1,NTW
1211 SLINE(I,2)=SLINE(I,2)+FMF/1.5*(SLINE(I,2)-XBEG)+FMS/1.5
QN=0.
DISP=DISP*(1.0+FMF/1.5)
XBEG=XBEG+FMS/1.5
XBEG=XBEG-QNSAVE
187 CONTINUE

```

C ADDITIONAL SET-UP

***** 210 *****

```

IF(IBLOCK.EQ.1) XBEGKP=XBEG
WRITE(6,5) (TI(II),II=1,12),IBLOCK,XBEG,DISP
NRR=SLINE(200,2)
NR=SLINE(200,1)
321 SUML=0.
XNMISS=0.

```

C BEGIN MATCHING OF LINE WITH ION

***** 220 *****

```

CALL HELP(NR,1,200,27)
DO 177 K=1,NR
IF(K.NE.1) GO TO 182
DO 183 L=1,300
183 IF(ABS(SLINE(K,1)-WCALC(L,1)).LT.0.0005) GO TO 184
184 LHOLD=K
186 ICALCX=ILINL(LHOLD)
CALL HELP(LHOLD,1,200,28)
ICALCX=ICALC(ICALCX)
GO TO 185
132 LHOLD=LHOLD+1

```

GO TO 186
185 CONTINUE

C BEGIN LINE IDENTIFICATION (FIRST PASS)

***** 230 ****

DCRIT=0.075
CALL HELP(NRR,1,199,29)
385 DO 178 L=1,NRR
CALL HELP(ICALCX,1,500,129)
DIFF=ABS(SLINE(K,1)-SLINE(L,2))
IF(DIFF.LE.DCRIT) GO TO 179
178 CONTINUE
311 GO TO 180
179 CONTINUE
WRITE(6,6) SLINE(K,1),SLINE(L,2),ICALCX,SLINE(L,3)
LCFD(ICALCX)=LCFD(ICALCX)+1
KIF(L)=1
IF(DIFF.GT.0.031) GO TO 1222
1222 CONTINUE
DIFFA=ABS(SLINE(K,1)-SLINE(L+1,2))
IF(DIFFA.GT.DCRIT) GO TO 322
WRITE(6,318) SLINE(K,1),SLINE(L+1,2),ICALCX,SLINE(L+1,3)
322 CONTINUE
SUML=SUML+SLINE(L,3)
GO TO 181
180 CONTINUE
WRITE(6,7) SLINE(K,1),ICALCX
LCMS(ICALCX)=LCMS(ICALCX)+1
323 XNMISS=XNMISS+1.0
181 CONTINUE
177 CONTINUE

C BEGIN MISFIT CHECK

***** 240 ****

XNRR=NR
XNRR=XNRR*0.4
IF(XNMISS.LT.XNRR) GO TO 315
WRITE(6,316)
IF(FLOAT(NR)-XNMISS) 340,315,340
340 ERCX=SUML/(FLOAT(NR)-XNMISS)
WRITE(6,411) ERCX
315 CONTINUE
WRITE(6,20)

CALL HELP(NRR,1,200,30)
DO1500 I=1,NRR
IF(KIF(I).EQ.1) GO TO 501
K=1
J=1
505 WMAY=WCALC(J,1)
CALL HELP(J,1,300,31)
IF(ABS(WMAY-SLINE(I,2)).LT.0.05) GO TO 503
IF(WMAY.GT.SLINE(I,2)) GO TO 504
J=J+1
IF(J.EQ.300) GO TO 504
GO TO 505
503 KSTAND(K)=J


```

CALL HELP(K,1,200,32)
K=K+1
J=J+1
GO TO 505
504 KTOTAL=K-1
IF(KTOTAL.EQ.0) GO TO 506
KBIGG=1
IF(KTOTAL.EQ.1) GO TO 507
CALL HELP(KBIGG,1,200,33)
DO 508 KIDN=2,KTOTAL
CALL HELP(KIDN,1,200,34)
KSTONE=KSTAND(KBIGG)
KSTTWO=KSTAND(KIDN)
CALL HELP(KSTONE,1,300,35)
CALL HELP(KSTTWO,1,300,36)
IF(WCALC(KSTONE,2).GT.WCALC(KSTTWO,2)) GO TO 508
KIDN=KIDN
508 CONTINUE
507 CONTINUE
KTHREE=KSTAND(KBIGG)
WRITE(6,6) WCALC(KTHREE,1),SLINE(I,2),ICALC(KTHREE),SLINE(I,3)
CALL HELP(KTOTAL,1,200,37)
DO 509 IA=1,KTOTAL
IF(IA.EQ.KBIGG) GO TO 510
KFOUR=KSTAND(IA)
CALL HELP(KFOUR,1,300,38)
FACTDW=WCALC(KTHREE,2)-WCALC(KFOUR,2)
WRITE(6,18) WCALC(KFOUR,1),SLINE(I,2),ICALC(KFOUR),FACTDW
510 CONTINUE
509 CONTINUE
GO TO 501
506 WRITE(6,21) SLINE(I,2),SLINE(I,3)
501 CONTINUE
1500 CONTINUE

```

C BEGIN OUTPUT PHASE

***** 260 *****

```

CALL HELP(NRR,1,200,39)
X(1601)=FLOAT(JWK)+0.1
X(1602)=FLOAT(IBLOCK)+0.1
X(1603)=XBEG
X(1604)=ALASHD
X(1605)=DISP
CALL WR6
WRITE(6,81) IBLOCK,JWK,XBEG,ALASHD
DO 670 I=1,20
670 CRACK(I)=(FLOAT(I)-1.0)*DISP
WRITE(6,82) (CRACK(I), I=2,20)
DO 671 I=1,80
IF(MOD(I,5).NE.1) GO TO 672
WRITE(6,83)
672 CONTINUE
IONE=(I-1)*20+1
ITWO=IONE+19
IF(IONE.GT.JWK) GO TO 671
AONE=XBEG+FLOAT(IONE)*DISP-DISP
ATWO=XBEG+FLOAT(ITWO)*DISP-DISP
WRITE(6,84) AONE,IONE,(X(J),J=IONE,ITWO),ITWO,ATWO

```

671 CONTINUE

C SET-UP FOR NEXT BLOCK AND TRANSFER

***** 270 ***

```

DO 511 I=1,200
511 KIF(I)=C
WRITE(7,80) IBLOCK,XBEG
WRITE(6,22)
IBLOCK=IBLOCK+1
XBEG=XBEG+DISP*FLOAT(JWK)
GO TO 104
5 FORMAT(1H1,12A6/14H0BLOCK NUMBER ,I2/14H ZERO POINT = ,F7.2/14H DI
*SPERSION = ,F8.6///35H SEARCH FOUND ION INTENSITY/)
6 FORMAT(1H ,F7.2,2X,1X,F7.2,3X,I3,2X,F5.3)
7 FORMAT(1H ,F7.2,3X,3HN/F,7X,I3,2X)
20 FORMAT(1H0///12H SECOND PASS//)
21 FORMAT(1X,3HN/T,7X,F7.2,3X,3HN/A,2X,F5.3)
22 FORMAT(1H0///)
80 FORMAT(I2,1X,F7.2,60X,1H4)
81 FORMAT(7H1BLOCK ,I3/18H NUMBER OF POINTS ,I4/22H BEGINNING WAVELEN
*GTH ,F7.2/11H CONTINUUM ,F9.3///)
82 FORMAT(1H0,15X,5H+,.00 ,9(1H+,F3.2,1X),1X,10(1H+,F3.2,1X)/)
83 FORMAT(1H )
84 FORMAT(1X,F7.2,2X,I4,2X,10(F4.2,1X),1X,10(F4.2,1X),1X,I4,2X,F7.2)
303 FORMAT(F10.2,1X,F6.3)
314 FORMAT(30H0AVERAGE INTENSITY OF LINES = ,F4.2,23H AND MEAN DISPERS
*ION = ,F5.3)
316 FORMAT(15H0POSSIBLE ERROR)
318 FORMAT(4H OR ,F7.2,2X,F8.2,3X,I3,2X,F5.3)
402 FORMAT(1H ,F7.2,2X,1X,F7.2,3X,I3,2X,F5.3,46X,1H1)
403 FORMAT(4H OR ,F7.2,2X,F8.2,3X,I3,2X,F5.3,45X,1H2)
404 FORMAT(1H ,F7.2,3X,3HN/F,7X,I3,2X,53X,1H3)
411 FORMAT(41H0AVERAGE INTENSITY OF IDENTIFIED LINES = ,F5.4)
500 FORMAT(10H RETURN A )
1001 FORMAT(1X,16H MEAN DISPERSION)
1002 FORMAT(1X,F20.8)
1009 FORMAT(12A6)
1011 FORMAT(I3/(10I3))
1090 FORMAT(1H ,2F15.5)
1202 FORMAT(36H0PRELIMINARY ZERO POINT CORRECTION =,F7.2)
1209 FORMAT(20H0TRANSFORM COFFFS. ,F7.4,2X,F7.4)
2006 FORMAT(I6)
2008 FORMAT(2F10.2)
2100 FORMAT(F7.4,1X,I3,1X,F10.6,1X,I3)
2110 FORMAT(1X,I4,2X,10(F4.2,1X),1X,10(F4.2,1X),2X,I4)
6679 FORMAT(I1)
END
$IBFTC CRUTCH LIST,M94,XR7
SUBROUTINE HELP(INDEX,IA,IB,N)
IF(INDEX.GE. IA.AND. INDEX.LE. IB) RETURN
WRITE(6,1) N
1 FORMAT(1H1,I3)
CALL LCOU
STOP
END
$IBFTC FINIS LIST,M94,XR7
SUBROUTINE LCOU
COMMON/LC/LCFD(500),LCMS(500)

```

```

WRITE(6,611)
611 FORMAT(12H ION SUMMARY///44H ION FOUND MISSED TOTAL PER CENT
* FOUND//)
DO 612 I=1,500
LTOT=LCFD(I)+LCMS(I)
IF(LTOT.EQ.0) GO TO 612
PER=FLOAT(LCFD(I))/FLOAT(LTOT)*100.0
WRITE(6,613) I,LCFD(I),LCMS(I),LTOT,PER
613 FORMAT(1X,I3,I7,2I8,7X,F5.1)
612 CONTINUE
RETURN
END

$IBMAP GRAF1 LIST,REF,DFCK,M94,XR7
A5 FILE UNA5,CK1,BIN,BLK=400,INPUT
A6 FILE UNA6,UT5,BIN,BLK=1605,INPUT
B6 FILE UNB6,UT6,BIN,BLK=1605,OUTPUT
ENTRY OP
ENTRY CLO
ENTRY RA6
OP SAVE 4
TSX .OPFN,4
PZF A6
TSX .OPFN,4
PZE A5
TSX .OPFN,4
PZF R6
RETURN OP
CLO SAVE 4
TSX .CLOSE,4
PZF A5
TSX .CLOSE,4
PZE A6
TSX .CLOSE,4
PZF R6
RETURN CLO
RA5 SAVE 4
A55 TSX .RFAD,4
PZF A5,,SYSDMP
PZE EOF5,,ERR5
IOCT WLINE,,4
A5A5 RETURN RA5
EOF5 CLA =1
STO JEOF
TRA A5A5
FRR5 CLA =1
STO IERR
TRA A55
WAN SAVE 4
TSX .RSF,4
PZE A5,,2
RETURN WAN
WANX SAVE 4
TSX .RSF,4
PZE A6,,2
RETURN WANX
WIN SAVE 4
TSX .RSF,4
PZF A5,,1

```

| | RETURN | WIN |
|-------|--------|------------|
| TIME | SAVE | 4 |
| | CLA | 5 |
| | COM | |
| | LRS | 8 |
| | ORA | CC.2 |
| | STO | SEC+1 |
| | CLA | CC.3 |
| | LLS | 8 |
| | FAD | SFC+1 |
| | FDP | =60.0 |
| | CLA | SFC |
| | STO | SFC |
| | FSR | SFC |
| | STO | SEC+1 |
| | RETURN | TIME |
| RA6 | SAVE | 4 |
| A666 | TSX | .READ,4 |
| | PZF | A6,,SYSDMP |
| | PZE | EOF6,,ERR6 |
| | IOCT | Y,,1605 |
| A6A6 | RETURN | RA6 |
| FOF6 | CLA | =2 |
| | STO | ↑FOF |
| | TRA | A6A6 |
| ERR6 | CLA | =2 |
| | STO | ↑FRR |
| | TRA | A666 |
| WB6 | SAVE | 4 |
| | TSX | .WRITE,4 |
| | PZF | B6 |
| | IOCT | X,,1605 |
| | RETURN | WB6 |
| | CONTRL | // |
| | USE | // |
| Y | BSS | 1605 |
| ↑FOF | BSS | 1 |
| ↑FRR | BSS | 1 |
| WLINE | BSS | 4 |
| JFOF | BSS | 1 |
| ZLINE | BSS | 5 |
| SEC | BSS | 2 |
| X | BSS | 1605 |
| | USE | PREVIOUS |
| | END | |

\$DATA

0

001

004

0003750.50 00.706

ALPHA BOOTES-1 AUGUST 7, 1968 (A.M.) FILE 1 CONDENSED FILE 4

000003

0000003.22 00800.00

0000005.00 10800.00

0000009.83 22400.00

00.7060 000 000.014280 099

-999999.99

0