93467

GENERATION AND REDUCTION OF THE DATA FOR THE ULYSSES GRAVITATIONAL WAVE EXPERIMENT

R. AGRESTI, P. BONIFAZI, L. IESS, G.B. TRAGER

IFSI-87-20

December 1987

(NASA-CR-184578) GENERATION AND REDUCTION OF THE DATA FOR THE ULYSSES GRAVITATIONAL WAVE EXPERIMENT (Jet Propulsion Lab.) 77 p

N89-12273

Unclas
G6/61 0175097

GENERATION AND REDUCTION OF THE DATA FOR THE ULYSSES GRAVITATIONAL WAVE EXPERIMENT

R. AGRESTI, P. BONIFAZI, L. IESS

-Istituto di Fisica dello Spazio Interplanetario, Frascati

G.B. TRAGER

-Jet Propulsion Laboratory, Pasadena, U.S.A.

IFSI-87-20

December 1987

INDEX

SUMMARY

We describe a procedure for the generation and reduction of the radiometric data known as 'REGRES' files. These data are a current output of NASA's Orbit Determination Program. The software package was developped in view of the data analysis of the gravitational wave experiment, (GWE) planned for the European spacecraft 'Ulysses'.

INTRODUCTION

Doppler tracking of interplanetary spacecraft is routinely performed by the stations of NASA's Deep Space Network (DSN). An electromagnetic carrier of highly stable frequency is sent from a ground station to a spacecraft, which retransmits it back coherently to the Earth by means of a trasponder.

The presently used radio link configurations for precision Doppler tracking are of two types: in the first one, called 'two-way' operation, the same station both transmits and receives the radio signal, in the second one, called 'three-way', the transmitting and receiving station are different. The signal is usually transmitted in S band (2.1 GHz) and received both in S and X band (8.4 GHz).

We call Doppler residual the quantity obtained by subtracting the measured frequency shift of the carrier to the predicted one. In the REGRES data, the predictions are obtained a posteriori by means of a regression method: the received radiometric data are used to improve the spacecraft ephemeris in order to minimise the least square deviation between the observables and the predictions.

In this paper we report a description of the REGRES data generation and of the software we have developped to reduce and display the data. This software has been applied to the analysis of four days (DOY 308 - DOY 313, 1980) of REGRES data relative to the Voyager I spacecraft.

The software package was implemented on an HP-1000F computer, with an RTE-6-VM operative system. The graphical unit was an HP-9872T plotter. A general purpose graphical routine, developed at IFSI-CNR, was used to plot output data.

REGRES FILE GENERATION

REGRES files are a current output of the Orbit Determination Program (ODP). The volume and the structure of REGRES files are not fixed, but depend on the way the ODP is run. Standard references on the subject are (Moyer, 1971) and (Khatib et al., 1972). ODP is actually a set of many links, which, to some extent, may be run independently. Its functions are integrated by a number of allied programs, the most important of which is DPTRAJ. The process which leads to the computation of the trajectory of the spacecraft is described in the following.

The fundamental quantities handled by the ODP are:

$\underline{X}=(\underline{r},\dot{\underline{r}})$: spacecraft state vector, i.e. position and velocity of the spacecraft with respect to a center of integration (usually the baricenter of a celestial body).

$\underline{q}=(\underline{X}_0,\underline{a})$: parameter vector, containing the initial state of the spacecraft ($\underline{X}_0$) and a set ($\underline{a}$) of physical parameters (see below) needed for the integration of the equations of motion.

$\underline{z}$: vector of observable quantities (Doppler, range, angle, ΔDOR).

The main physical parameters $\underline{a}$ are:

- the gravitational parameters $\mu_i=GM_i$ of Sun, planets and satellites of the solar system;
- the coefficients of the harmonic expansion of the

planets' gravitational potential;

- the quantities used in modelling the probe acceleration due to small forces and manoeuvres;
- atomic station time -> UTC conversion parameter;
- coordinates of the tracking stations;
- the relativistic parameter γ.

Many other quantities may be used, depending on the particular spacecraft: there are parameters relating to atmospheric models, mass anomalies, gas leaks, solar radiation pressure, etc.

The vector $\underline{a}$ is usually stored as a part of a file called Generalised Input File (GIN file), characterising the physical world in which the probe moves.

The ODP not only performs the integration of the equations of motion, but also allows the redeterminations of the parameters $\underline{q}$ and, therefore, the determination of a better state vector $\underline{X}(t)$. This goal is obtained by comparing predictions and observations in a regressive analysis made up of several steps.

STEP 1 PATH-VARY (or PVDRIVE)

This link has actually a twofold purpose and may be separated for clarity into two sub-units (PATH and VARY). PATH integrates the equations of motion of the probe, starting from given injection parameters $\underline{X}o$ and physical parameters $\underline{a}$. The probe acceleration results from the following contributions:

- Newtonian gravitational forces (leading term) from the relevant celestial bodies.
- Oblatenesses of planets.
- Mascons.
- Relativistic effects.
- Solar radiation pressure.
- Attitude control.

- Orbital manoeuvers.
- Atmospheric drag (in case of planetary encounter).

The center of integration may be any planet, or center of mass of planetary system (i.e. planet with satellites). The output of PATH is a propagation of the initial state vector X along the trajectory as a function of time and of the parameters $\underline{q}$:

$$\underline{X} = \underline{X}\ (t,\ \underline{q})$$

Its output $\underline{X}(t)=(\underline{r}(t),\dot{\underline{r}}(t))$ may be stored in the Probe Ephemeris Tape (PET). It is also used as input to the program PREDIX which provides observable predictions to the tracking stations. These predictions are used to generate the pseudoresiduals contained in the Archival Tracking Data Files (ATDF).

The other subunit (VARY) integrates the variational equations

$$\frac{\partial \ddot{\underline{r}}}{\partial \underline{q}} = \frac{\partial \ddot{\underline{r}}}{\partial \underline{r}}\frac{\partial \underline{r}}{\partial \underline{q}} + \frac{\partial \ddot{\underline{r}}}{\partial \dot{\underline{r}}}\frac{\partial \dot{\underline{r}}}{\partial \underline{q}} + \left[\frac{\partial \ddot{\underline{r}}}{\partial \underline{q}}\right]_{\underline{r},\dot{\underline{r}}=\text{ const}}$$

$$\ddot{\underline{Z}} = A\underline{Z} + B\dot{\underline{Z}} + C \qquad\qquad \underline{Z} = \frac{\partial \underline{r}}{\partial \underline{q}}$$

whose solution gives $\underline{Z}$ and $\underline{Z}$ as functions of time. These quantities are needed in the subsequent step (REGRES) to get the partial derivatives of the observable quantities with respect to the parameters. The matrices A, B and C are obtained from the equations of celestial mechanics.

STEP 2 (REGRES)

In this step, the regression partial derivatives of the observables (Doppler, range, angle) with respect to the parameters $\underline{q}$ are formed. The regression partial derivatives are needed to adjust the quantities $\underline{X}o$ and $\underline{a}$ in order to obtain a better estimate of the spacecraft trajectory (i.e. better residuals).

To this end, for each acquired value of the observable $\underline{r}$ at the reception time t_3 , the corresponding ground transmission time t_1 and spacecraft reception time t_3 are computed. The times t_1 and t_2 are obtained by solving the light time problem in the metric of the solar system baricenter. For each leg (uplink and downlink) the transmission and reception times are given by (neglecting terms of order $1/c^5$):

$$t_{rec} - t_{trans} = \frac{r_{ij}}{c} + \frac{1+\gamma}{c^3} \mu s \ln \frac{(r_i+r_j+r_{ij})}{(r_i+r_j-r_{ij})}$$

where

$$r_i = || \underline{r}_i (t_{trans}) ||$$

$$r_j = || \underline{r}_j (t_{trec}) ||$$

$$r_{ij} = || \underline{r}_j (t_{rec}) - \underline{r}_i (t_{trans}) ||$$

$\mu_S = GM_S$ = gravitational parameter of the Sun

γ = post newtonian parameter (in general relativity $\gamma = 1$)

The vectors $\underline{r}$ have origins in the solar system baricenter. In the last version of REGRES bending terms as well as Jupiter and Saturn terms are included.

The solution of the light time problem must be performed for each acquired point, and requires therefore considerable computer time.

The regression derivatives $\partial \underline{r}/\partial \underline{q}$ are then formed by considering that the observable $\underline{r}$ is a function of the state vectors of the Earth stations and the spacecraft;

$$Z = Z[\underline{X}_{st}(t_3,\underline{q}),\underline{X}_{sc}(t_2,\underline{q}),\underline{X}_{st}(t_1,\underline{q});\underline{q}]$$

It may be seen that the partial derivatives $\partial \underline{r}/\partial \underline{q}$ are actually the sum of many terms, originating from explicit and implicit dependence of $\underline{Z}$ on the parameter vector. We refer to (Moyer, 1982) for a detailed discussion of the subject. Here we just point out that the solution of the variational equations, which provides the quantity $\partial \underline{X}_{sc}/\partial \underline{q}$, is required for the computation of $\partial \underline{Z}/\partial \underline{q}$.

A typical REGRES output file contains:

a) values of the parameters $\underline{q}$ for which partials have been computed

b) the selected observables and their residuals

c) the selected partial derivatives $\partial \underline{r}/\partial \underline{q}$.

It must be noted that at this step the parameters $\underline{q}$ are divided into two groups: "solve-for" parameters, for which new values are computed; "consider" parameters, which are not corrected but whose errors are taken into account in the computation of the errors for the "solve-for" parameters.

STEP 3 (ACCUME-SOLVE)

These two links use the partials obtained from the previous step to compute the parameter vector $\underline{q}$ and its covariance matrix. With the new q's, a new GIN file may be created, and a new iteration, PV, REGRES, ACCUME, SOLVE (PVRAS), may be started.

The REGRES file which we analysed has not been generated by the complete sequence PVRAS, which is complicated and unnecessary for our purposes. The link REGRES was run in the so called "simulation mode", i.e. using already existing PV and GIN files and bypassing the computation of the partial derivatives $\partial \underline{z}/\partial \underline{q}$ The following steps have been performed to get the final REGRES file:

a) select the GIN file to be used, which contains an initial state vector $(\underline{r}_o,\dot{\underline{r}}_o)$ of the spacecraft and a set of physical parameters $\underline{a}$ that already gave satisfactory residuals.

b) select the corresponding PV file (containing probe ephemeris)

c) select the file (ATDF or IDR) containing the observable quantities for which the residuals have to be computed (we actually used the same ATDF which was previously analysed at Frascati).

d) run the program STRIPPER which, starting from an ATDF or IDR, generates an OD file (a file which can be used as an input to REGRES)

e) run REGRES in the "simulation mode", using PV, GIN and OD files as input (together with the ephemeris of the bodies of the solar system).

The output of the step (e) is the required REGRES file.

SOFTWARE

In this section we report on the software package implemented on the HP1000F System at CNR-Frascati to handle the REGRES files.

These data, generated by an Univac 1108 System at JPL and stored on magnetic tape, have been converted using the Univac computer of the University of Rome in ASCII characters files stored on magnetic tape. The list of this Fortran program (named UNIHP) is shown in appendix B.

The main program implemented on the HP1000F (named DECODEREGRESS) decodes the REGRES data (ASCII characters) and selects the relevant items for the GWE data. The format of the output data (called STANDARD format throughout the paper) is described in (Iess and Armstrong 1985).

The software package is completed by a set of programs for the display and validation of the data and for the storage of Doppler residuals (our observable) in the HP disk memory. These programs are named: STATREGRESS, PLOTREGRESS and FFREGRESS.

STATREGRESS reads the STANDARD tapes and produces an output list for each day of data. This list contains hourly and total number of samples for each active station, band and mode.

PLOTREGRESS provides plots of the STANDARD tapes.

In order to simplify the access to the data, the STANDARD files can be copied on a disk using the FFREGRES program. Each disk file contains the data relative to a single pass.

The program listings are reported in Appendix B. In the following we report a description of the software.

DECODEREGRES

DECODEREGRESS reads the REGRES tapes in ASCII format, selects the items relevant to the GWE and generates the STANDARD tapes.

DATA STRUCTURE

INPUT

BUFFER DIMENSION: ISK(40,100)

RECORD LENGTH: 8000 characters

RECORD FORMAT: Each data record is composed of 100 cards of 80 characters each. The 80 characters of each card have the following meaning:

char. 1-5 : measurement group identifier N1, ranging from to 99999

char. 6-8 : card identifier N2, ranging from 1 to 100

char. 9-32 : data value, written in the format F24.18

char. 33-56: as above

char. 57-80: as above

DATA FORMAT: In the original tapes generated at JPL, twenty quantities were associated to each Doppler data acquisition (see tab. 1). We will refer to this set of 20 quantities as to a "measurement". Therefore each record contains 15 measurements (i. e.: 3*100/20), (see tab. 2). A data block is defined as a set of 40 measurements. The block identifier number N1, which appear in tab. 2, refers to the record generated at JPL with the Univac routine NTRAN.

See tab. 3 for an example of ISK (40,100) record.

OUTPUT

The output data, stored on magnetic tapes, are structured as a sequence of fixed length records with eof.

BUFFER DIMENSION: LMAT(160,15)

RECORD LENGTH: 2400 double integers number, i.e. 4800 16-bit words.
RECORD FORMAT: Each of the 160 rows contains 15 double integer items.

DATA FORMAT: Each row contains information about one measurement in both bands (S and X). Tab. 4 shows the data structure of the STANDARD records.

In tab. 5 we report, the items of the REGRES tapes generated at JPL and their corresponding items in the STANDARD records. Tab. 6 shows an example of STANDARD record (they e.c. refers to the same REGRES data in tab. 3).

ALGORITHM

The program reads in sequence the ISK(40,100) records. The 20 quantities corresponding to a single measurement are transferred in the INTEGER*4 buffer ISK1(240). This buffer is decoded and the relevant items of each measurement are transferred in a second INTEGER*4 buffer LMAT(160,15). It is important to note that two subsequent measurements in input usually refers to the S and X band respectively. In order to properly fill the eighth column of the matrix LMAT (S-3/11X), the program requires that S-band datum must precede the X-band datum in the input data. In fig. 2a and 2b is reported the flowchart of the program.

The program is structured as a main routine, managing the input data buffer, and a subroutine, managing the output buffer. Those two program segments are now described in more detail, referring to fig. 2a and 2b.

MAIN PROGRAM

1) INFORMATION FOR INPUT/OUTPUT

1a) options and parameters for the job:
sequential number of the required initial REGRES

record, sequential number of the required final REGRES record;
tape logical unit number;
etc.

1b) search for initial record to analyse

2) READ A ISK (40.100) RECORD
this step is made using the system routine XTAPE (EXEC)
check on the record length
eof causes jump to step #8
read error causes jump to next record

3) TEST FOR COLUMN IDENTIFIER
if the block identifier is changed, the program resets the decodifying buffer pointer IP. This step is needed to eliminate the free item present at every changing group

4) DECODIFICATION AND STORAGE OF THE DATA
this step is performed by the subroutine CONTROLLO which transfers and decodes the data

5,6,7) REPEAT STEPS 1 TO 4 FOR ALL REQUIRED RECORDS AND TAPES

8) FINAL OPERATIONS
8a) stores the last data in memory
8b) writes eof on STANDARD tape and rewinds the tapes
8c) prints information on the work just carried out
9) STOP

STATREGRESS

This program reads the STANDARD tapes and produces an output list for each day. The list contains hourly and total number of samples for each active station, band and mode. An example of an output list is reported in tab. 7.

ALGORITHM

The input data are contained in the STANDARD files generated by DECODEREGRESS. The algorithm uses a buffer of sample counters structured as a matrix 48*24. Each row represents one of the combinations of 6 DSN, stations 4 tracking modes and 2 bands, while the 24 columns represent the hours of the day. The identification of the sample counter for a given DSN station, band, mode and hour of day combination is given by:

ROW = (S - 1) * 8 + (M - 1) * 2 + B
COL = HOUR OF DAY

where B is 1 or 2 for S or X band respectively M represent the 4 different radio link modes, while S=1,2,3,4,5,6 represents the six DSS: 12, 14, 42, 43, 61 and 63 respectively. The decodification of the row number can be made as follows:

STATION = INT ((ROW - 1) / 8) + 1
BAND = ROW's PARITY
MODE = INT ((ROW - BAND - (STATION - 1) * 8) / 2) + 1

In fig. 3 is reported the flow chart of the program, whose main steps are described below.

1) ASK INFORMATION FOR INPUT/OUTPUT
   tape logical unit number

number of records to analyse
initial and final day and year
etc.

2) READ A RECORD I4 (160, 15)

3, 4) SEARCH FOR FIRST RELEVANT DATA

5, 6, 7) ANALYSE DATA AND FILL THE LIST

8) OUTPUT OPERATIONS

if the list (i. e. the output buffer) of the required day is empty, (if there are no data, for example) the program prints a warning, otherwise it prints the required list.

10) START ANALYSIS OF A NEW REQUIRED DAY (if any)

11) STOP

PLOTREGRESS

This program provides plots of STANDARD data for each day and mode.

DATA STRUCTURE

The input data are the STANDARD file generated by the program DECODEREGRESS. The user selects the passes to be plotted for each desired day. DOY, mode and the DSS are required as input parameters. The data appear into three curves (S-band, X-band and S-3/11*X) as shown in fig. 4.

ALGORITHM

The program flow chart is shown in fig. 5.

1) ASK INFORMATION FOR INPUT/OUTPUT

1a) options and parameters for the job as:
    initial and final REGRES records
    tape logical unit number
    required days and modes
    etc.

1b) search for the initial record to analyse

2) REWIND TAPE

3) READ A RECORD

4) CHECK FOR DAY AND MODE
   if data refers to the required day and mode, they are stored in a buffer

5) DO DATA REFER TO A LATER DAY?
   in this case the selected data are plotted

6) LAST RECORD?
   the search for the relevant data is repeated for

all records;
the search ends if the data of a record refer to a later day;
if there are no data to plot, the program prints a warning and goes to step #8.

7) PLOTTING
this step is carried out by a software package developped at IFSI

8) OTHER PLOTS?
in this case, steps (1-7) are repeated.

9) STOP

FFREGRES

This program stores S and X band Doppler residuals and oscillator frequency from STANDARD tapes into disk files. Each file contains data referring to a given pass. Besides Doppler residuals, each file contains a header which reports spacecraft angular data, RTLT and other information. Some of these quantities are retrieved from the file DATA/FILE#1 (see Iess and Armstrong, 1985)

DATA STRUCTURE

INPUT : The input data are the STANDARD data and angular data FILE#1 (see Tab. 8)

OUTPUT: In output one obtains ASCII disk files relative to a single pass

BUFFER DIMENSION: ISTORE(3,730)

RECORD LENGTH: 2190 double integers, i.e. 4380 16-bit word

RECORD FORMAT: each of the 3 rows contains 730 double integers. The file name has the following mnemonic code:

DAT/RODTTRRDDD

where:

DAT is the data subdirectory;

ROD stands for REGRES Original Doppler resid.;

TT is the code for transmitting DSS;

RR is the code for receiving DSS;

DDD are three characters indicating the DOY.

DATA FORMAT: each file (see tab. 9) is composed by a header, with general information about the pass, followed by S and X band residuals and oscillator frequency in the first, second and third row respectively. Residuals are given in mmHz while the oscillator frequency is given in Hz. The header contains initial and final time, number of data, receiving and transmitting DSS, tracking mode, RTLT, spacecraft's right ascension and declination at the middle of the pass. An example of residual file is reported in tab. 10.

ALGORITHM

The algorithm looks for data referring to the required receiving DSS, mode and begin of tracking day. Then it selects data from FILE#1 to build the header (see fig. 6).

For the Voyager I data the sampling rate was of 1 pt/min: therefore each Doppler residual refers to a given minute of the day (MOD). The time T (in MOD) of every datum can be obtained from the following relation:

$$T = \text{module} (I - 11 - T_{in} , 1440)$$

where I is the column number of the datum in the array ISTORE, and T_{in} the MOD of the first datum of the file. Since the header must be skipped, 11 must be subtracted from I to get the right position. Finally, the remainder of the division of I - 11 -Tin by 1440 provides the MOD of the datum (if I-11-Tin > 1440, the MOD refers to the next day).

ACKNOWLEDGEMENTS

We are grateful to B. Bertotti (P.I. of the Ulysses GWE) for his help in the preparation of this report. Special thanks are also due to J. Brenkle, and A. Khatib for their cooperation during the stay one of us (L. Iess) at JPL, in the summer 1985. We thank also M. Stazi for the preparation of the figures.

REFERENCES

Iess, L. and Armstrong, J.W.: internal memo IFSI-GWE, 1985.

Khatib, A.R, Null, G.W. and Zielenbach, J.W.: "Engineering Planning Document - The alphabet System", JPL 900-578, Oct. 1972

Moyer, T.D.: "Mathematical Formulation of the Double Precision Orbit Determination Program", JPL-TR 32-1527, May 1971.

APPENDIX A

LIST OF ACRONYMS AND NAMES

ATDF	Archival Tracking Data File
DOY	Day Of the Year
DSN	Deep Space Network
DSS	Deep Space Station
EOF	End Of File
GIN	Generalised INput file
GWE	Gravitational Wave Experiment
IDR	Intermediate Data Records
MOD	Minute Of the Day
ODP	Orbit Determination Program
PET	Probe Ephemeris Tape
PV	Path-Vary
REGRES	Regression files
RTLT	Round Trip Light Time

APPENDIX B

PROGRAM LISTINGS

We report in this appendix the fortran program listings of the computer programs used in the data analysis.

```
C        PROGRAM UNIHP
C
C ******************************************************************
C *  TRASFORMAZIONE DEI NASTRI REGRESS IN FORMATO UNIVAC           *
C ******************************************************************
C *                                                                *
C *  QUESTO PROGRAMMA OPERA SUI DATI REGRESS                       *
C *                                                                *
C ******************************************************************
C
C DICHIARAZIONI
      COMMON /DATI/A(801),AOUT(100),NROUT
      DOUBLE PRECISION ARRAY(20,40),A
      CHARACTER*80 AOUT
C
      IOUT=10
      IOUT2=11
      PRINT 1
 1    FORMAT(2X,'N. RECORD INITIAL AND FINAL?')
      READ(5,*) NIN,NFI
      NRIN=NFI-NIN+1
      PRINT 2
 2    FORMAT(2X,' DOYOU WANT PRINTOUT OF A(20,40)? (1=SI)')
      READ(5,*) ISTAM
      NR=0
      NROUT=0
      LL=0
      N1=NIN-1
      IF(N1.EQ.0) GO TO 100
 200  CONTINUE
      CALL NTRAN$(IOUT,2,1600,ARRAY,L,22)
      NR=NR+1
      IF(L.LT.0) GO TO 500
      IF(NR.LT.N1) GO TO 200
 100  CONTINUE
      CALL NTRAN$(IOUT,2,1600,ARRAY,L,22)
      NR=NR+1
      IF(L.LT.0) GO TO 500
      IF(L.EQ.-2) GO TO 610
      IF(ISTAM.NE.1) GO TO 150
      WRITE(8,300) NR
      WRITE(8,400)
      WRITE(8,700) (ARRAY(L,1),L=1,20)
      WRITE(8,400)
      WRITE(8,700) (ARRAY(L,40),L=1,20)
      WRITE(8,400)
 150  CONTINUE
      A(801)=0.D0
      II=0
      DO 10 I=1,40
      DO 10 J=1,20
      II=II+1
      A(II)=ARRAY(J,I)
```

```
 10    CONTINUE
       DO 20 L=1,801,3
       L1=L+2
       CALL FILBUF(LL,IOUT2)
       ENCODE(80,800,AOUT(LL),JJ,ERR=900) NR,LL,(A(K),K=L,L1)
       IF(ISTAM.NE.1) GO TO 20
       IF(L1.GT.9.AND.L1.LT.793) GO TO 20
       WRITE(8,850) LL,AOUT(LL)
 850   FORMAT(2X,'LL=',I3,2X,A80)
 20    CONTINUE
 800   FORMAT(I5,I3,3D24.18)
 950   FORMAT(2X,'ERROR WRITE K,LL,NR,NROUT',4I9)
C
       IF(NR.GT.NFI) GO TO 600
       GO TO 100
C
 300   FORMAT(2X,'RECORD NUMBER =',I10)
 400   FORMAT(//)
 700   FORMAT(2X,3D24.18)
C
 500   CONTINUE
       PRINT 3,NR,NROUT
 3     FORMAT(2X,'HARDWARE ERROR, NR,L NROUT',2X,3I9)
       STOP
 610   CONTINUE
       PRINT 6,NR,NROUT
 6     FORMAT(2X,'EOF AFTER N. RECORDS =',I10,2X,' NROUT = ',I10)
 600   CONTINUE
C
       CALL NTRAN$(IOUT2,1,2000,AOUT,L,22)
       NROUT=NROUT+1
       PRINT 5,NROUT,NRIN
       WRITE(8,400)
       WRITE(8,5) NROUT,NRIN
 5     FORMAT(2X,'NUMBER OF RECORDS WRITTEN AND READ',2X,2I10)
       CALL NTRAN$(IOUT2,9,2000,AOUT,L,22)
       GO TO 1000
 900   PRINT 950,K,LL,NR,NROUT
 1000 CONTINUE
C
C FINE DEL MAIN
       STOP
       END
C
C ********************************************************************
C QUESTA SUBROUTINE RIEMPIE IL BUFFER
       SUBROUTINE FILBUF(LL,IOUT2)
C
       COMMON /DATI/A(801),AOUT(100),NROUT
       CHARACTER*80 AOUT
       DOUBLE PRECISION A
C
       LL=LL+1
```

```
      IF(LL.LT.100) RETURN
      CALL NTRAN$(IOUT2,1,2000,AOUT,L,22)
      IF(L.LT.0) PRINT 10,NROUT
      NROUT=NROUT+1
      NO=(NROUT/50)*50-NROUT
      IF(NO.EQ.0) PRINT 20,NROUT
 10   FORMAT(2X,'HARDWARE ERROR AT RECORD =',I10)
 20   FORMAT(2X,'NUMBER OF RECORDS WRITTEN =',I10)
      LL=1
C
C RITORNO AL MAIN
      RETURN
C
C FINE DI FILBUF
      END
```

```
FTN7X,S
      PROGRAM DECODEREGRESS(),                 <880329.1415>
C
C ********************************************************************
C *  DECODIFICA SEQUENZIALE DEI DATI REGRESS DA NASTRO                *
C ********************************************************************
C *                                                                  *
C *  QUESTO PROGRAMMA OPERA SUI DATI REGRESS.                        *
C *                                                                  *
C *  ESSO LEGGE DA NASTRO RECORDS DI 4000 PAROLE, LI DECODIFICA      *
C *  ED IMMAGAZZINA SU NASTRO D'USCITA COME INTEGER*4.               *
C *                                                                  *
C ********************************************************************
C
C DICHIARAZIONI
      DIMENSION ISK(40,100)
      COMMON ISK1(240),IP,IRM,IRC,IC3,IPRINT1,IPRINT2,ITAPE1
      COMMON IWR1,NIN,ILM1,ILM2
      INTEGER*4 LMAT(160,15)
      DIMENSION IA(4)
C
C ESEGUIBILE
C INFORMAZIONI DI INGRESSO/USCITA
  111 WRITE(1,11)
   11 FORMAT(1X,21/)
      WRITE(1,*) ''
      WRITE(1,*) '    DECODIFICA SEQUENZIALE DEI DATI REGRESS'
      WRITE(1,*) '    **************************************'
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*) ' QUESTO PROGRAMMA LEGGE DA NASTRO RECORDS DI'
      WRITE(1,*) ' 4000 PAROLE FORNENDO IN USCITA, A SCELTA:'
      WRITE(1,*)
      WRITE(1,*) '  1) SEMPLICE LISTATO DEI DATI'
      WRITE(1,*) '  2) LISTATO DEI DATI DECODIFICATI'
      WRITE(1,*) '  3) IMMAGAZZINAMENTO DEI DATI DECODIFICATI'
      WRITE(1,12)
   12 FORMAT(1X,/,80'*',/)
      WRITE(1,'("RECORD REGRESS INIZIALE: ")')
      READ(1,*) NIN
      WRITE(1,'("RECORD REGRESS FINALE: ")')
      READ(1,*) NFI
      WRITE(1,'("UNITA'' NASTRO IN LETTURA: ")')
      READ(1,*) ITAPE
      REWIND ITAPE
      WRITE(1,'("VUOI IL SEMPLICE LISTATO? <1=SI>: ")')
      READ(1,*) IPRINT
      IF(IPRINT.EQ.1) THEN
       WRITE(1,'("UNITA'' STAMPANTE?: ")')
       READ(1,*) IWR
       WRITE(IWR,*)
       WRITE(1,'("DESIDERI SOLO QUESTA STAMPA? <1=SI>: ")')
       READ(1,*) IONLY
```

```
      ENDIF
      IF(IONLY.NE.1) THEN
       WRITE(1,'("VUOI IMMAGAZZINARE I DATI STANDARD? (1=SI): ")')
       READ(1,*) IPRINT2
       IF(IPRINT2.EQ.1) THEN
        WRITE(1,'("UNITA'' NASTRO IN SCRITTURA?: ")')
        READ(1,*) ITAPE1
        IF(ITAPE.EQ.ITAPE1) THEN
         WRITE(1,*) 'ATTENTO! HAI FORNITO LA STESSA UNITA'' NASTRO'
         WRITE(1,*) 'SIA IN SCRITTURA CHE IN LETTURA!'
         GO TO 111
        ENDIF
       ENDIF
       WRITE(1,'("VUOI UNA STAMPA DEI DATI STANDARD? (1=SI): ")')
       READ(1,*) IPRINT1
       IF(IPRINT1.EQ.1) THEN
        WRITE(1,'("UNITA'' STAMPANTE?: ")')
        READ(1,*) IWR1
       WRITE(IWR1,*)
       ENDIF
      ENDIF
C
C RICERCA DEL RECORD INIZIALE RICHIESTO
      NRC=NIN-1
      IF(NRC.NE.0) WRITE(1,*) 'RICERCA DEL RECORD INIZIALE'
      DO 1 I=1,NRC
       CALL XTAPE(1,ITAPE,ISK,4000,LEN)
       IF(LEN.NE.4000) THEN
        WRITE(1,100)I,LEN
100     FORMAT(2X,'ERRORE IN LETTURA REC# ',I5,' LUN.# ',I5)
       ENDIF
    1 CONTINUE
      WRITE(1,200) NRC
  200 FORMAT('NUMERO DI RECORDS REGRESS SALTATI:',I5)
      WRITE(1,300)
  300 FORMAT(1X,/,/,80'*')
C
C PER CIASCUN RECORD RICHIESTO
      DO 2 II=NIN,NFI
      IRC=IRC+1
C
C LETTURA DI UN RECORD
       CALL XTAPE(1,ITAPE,ISK,4000,LEN)
       IF (LEN.NE.4000) THEN
        IF (LEN.EQ.-1) THEN
         WRITE(1,*) 'EOF IN LETTURA'
         GO TO 10
        ENDIF
        WRITE(1,100) 'ERRORE IN LETTURA RECORD #',II,' LUN.=',LEN
        GO TO 2
       ENDIF
       WRITE(1,*) 'LETTO DALLA UNITA''',ITAPE,' IL RECORD #',II
C
```

```
C STAMPA IL RECORD REGRESS SE RICHIESTO
        IF(IPRINT.EQ.1) THEN
         IF(IWR.NE.1) WRITE(1,*) 'SCRITTURA SULLA UNITA''',IWR,
     +'  REC #',II
         WRITE(IWR,400) II
  400    FORMAT(1X,'      RECORD # ',I5)
         WRITE(IWR,500) ((ISK(I,J),I=1,40),J=1,100)
  500    FORMAT(40A2)
         WRITE(IWR,600)
  600    FORMAT(/)
         IF(IONLY.EQ.1) GO TO 2
        ENDIF
C
C PER CIASCUNA COLONNA DEL RECORD
        DO 3 ICL=1,100
C
C CONTROLLO SUGLI IDENTIFICATORI
         CALL SMOVE(ISK(1,ICL),1,8,IA,1)
         DECODE(8,700,IA,ERR=20) IC1,IC3
  700    FORMAT(I5,I3)
         IF(IC1.NE.IC2) THEN
          IC2=IC1
          IP=1
         ENDIF
C
C TRANSFERIMENTO DI UNA COLONNA DI 'ISK' IN 'ISK1' CONTROLLANDO,
C DATO PER DATO, SE IL BUFFER E' PIENO (TEST SU IP)
         CALL SMOVE(ISK(1,ICL),9,32,ISK1(1),IP)
         CALL CONTROLLO(LMAT)
         CALL SMOVE(ISK(1,ICL),33,56,ISK1(1),IP)
         CALL CONTROLLO(LMAT)
         CALL SMOVE(ISK(1,ICL),57,80,ISK1(1),IP)
         CALL CONTROLLO(LMAT)
C
C SI RIPETE PER UN' ALTRA COLONNA
    3  CONTINUE
C
C SI RIPETE PER UN ALTRO RECORD
    2 CONTINUE
C
C OPERAZIONI FINALI
   10 REWIND ITAPE
      IF(IONLY.NE.1) THEN
C
C SI IMMAGAZZINANO I DATI RIMASTI IN MEMORIA
        IF(IPRINT2.EQ.1) THEN
         ILM2=ILM2+1
         CALL XTAPE(2,ITAPE1,LMAT,4800,LEN)
         IF(LEN.NE.4800) THEN
          WRITE(1,*) 'ERRORE IN SCRITTURA SU ',ITAPE1
          WRITE(1,*) 'LMAT #',ILM2,' LEN =',LEN
         ENDIF
         WRITE(1,*) '**SCRITTO SU',ITAPE1,'  #',ILM2,'RECORDS STANDARD'
```

```
      ENDIF
      IF(IPRINT1.EQ.1) THEN
       ILM1=ILM1+1
       WRITE(1,*) 'SCRITTURA SU',IWR1,' DEL RECORD #',ILM1
       DO 4 MM=1,160
    4  WRITE(IWR1,800) MM,(LMAT(MM,NN),NN=1,15)
  800  FORMAT(I3,I6,I7,I9,I7,I9,I8,I9,I6,I7,I8,5I7)
      ENDIF
C
C SCRITTURA DELL' EOF SUL NASTRO IN USCITA
      IF(IPRINT2.EQ.1) THEN
       WRITE(1,'("VUOI UN ALTRO NASTRO REGRESS? (1=SI): ")')
       READ(1,*) NY
       IF(NY.EQ.1) GO TO 111
       ENDFILE(UNIT=ITAPE1,IOSTAT=KKK,ERR=30)
       WRITE(1,*) '**SCRITTO EOF SU',ITAPE1
       REWIND ITAPE1
      ENDIF
     ENDIF
C
C FINE REGOLARE DELL' ANALISI
     WRITE(1,300)
     WRITE(1,*) '*****'
     WRITE(1,*) '***LETTI DA',ITAPE,' ',NFI-NIN+1,' R. RECORDS'
     IF(IPRINT.EQ.1) THEN
      WRITE(1,*) '*****'
      WRITE(1,*) 'SCRITTI SU',IWR,' ',IRC,' R. RECORDS'
     ENDIF
     IF(IPRINT2.EQ.1) THEN
      WRITE(1,*) '*****'
      WRITE(1,*) 'IMMAGAZZINATI SU',ITAPE1,' ',ILM2,' S. RECORDS'
     ENDIF
     IF(IPRINT1.EQ.1) THEN
      WRITE(1,*) '*****'
      WRITE(1,*) 'SCRITTI SU',IWR1,' ',ILM1,' S. RECORDS'
     ENDIF
     WRITE(1,*) '*****'
     WRITE(6,300)
     WRITE(6,*) '*****'
     WRITE(6,*) '***LETTI DA',ITAPE,' ',NFI-NIN+1,' R. RECORDS'
     IF(IPRINT.EQ.1) THEN
      WRITE(6,*) '*****'
      WRITE(6,*) 'SCRITTI SU',IWR,' ',IRC,' R. RECORDS'
     ENDIF
     IF(IPRINT2.EQ.1) THEN
      WRITE(6,*) '*****'
      WRITE(6,*) 'IMMAGAZZINATI SU',ITAPE1,' ',ILM2,' S. RECORDS'
     ENDIF
     IF(IPRINT1.EQ.1) THEN
      WRITE(6,*) '*****'
      WRITE(6,*) 'SCRITTI SU',IWR1,' ',ILM1,' S. RECORDS'
     ENDIF
     WRITE(6,*) '*****'
```

```
      WRITE(1,900)
  900 FORMAT(1X,/,50'*','FINE REGOLARE DELL''ANALISI')
      REWIND ITAPE
      REWIND ITAPE1
      STOP
C
C ERRORE DI DECODIFICA
   20 WRITE(1,*) '**ERRORRE DI DECODIFICA**'
      STOP
C
C ERRORE IN EOF
   30 WRITE(1,*) 'ERRORE SU',ITAPE1,' IN EOF #',KKK
      STOP
C
C FINE DEL MAIN
      END
C
C *************************************************************************
C QUESTA SUBROUTINE SLITTA L'INDICE DI POSIZIONE NEL BUFFER
C E LO DECODIFICA SE QUESTO E' PIENO
      SUBROUTINE CONTROLLO(LMAT)
C
C DICHIARAZIONI
      COMMON ISK1(240),IP,IRM,IRC,IC3,IPRINT1,IPRINT2,ITAPE1
      COMMON IWR1,NIN,ILM1,ILM2
      INTEGER*4 LMAT(160,15)
      INTEGER*4 LYD,LHMS,LTRM,LE,LB,LFRQ
      DOUBLE PRECISION DTIM,DOBS,DFRQ,DPAS,DCOM,DRTL,DHRG,DDEC,DAZM
      DOUBLE PRECISION DELV,DG1,DGM3,DGM1,DRES,DRJC,DWGH,DCRS,DMOD
      CHARACTER A1,A2,A3,A4,A(10)
C
C SLITTAMENTO DELL' INDICE DI POSIZIONE NEL BUFFER
      IP=IP+24
C
C CONTROLLO SUL RIEMPIMENTO DEL BUFFER
      IF(IP.EQ.481) THEN
C
C IL BUFFER E' PIENO: CONTIENE TUTTI I DATI RELATIVI AD UNA MISURA
C SI EFFETTUA LA DECODIFICA
       IP=1
       DECODE(480,100,ISK1,ERR=900) DTIM,A1,LYD,LHMS,(A(K),K=1,10),
     +IB,IT,IR,IM,A2,A3,A4,DOBS,DFRQ,DPAS,DCOM,DRTL,DHRG,DDEC,DAZM,
     +DELV,DG1,DGM3,DGM1,DRES,DRJC,DWGH,DCRS,DMOD
  100  FORMAT(D24.18,A2,I5,I7,10A2,4I2,3A2,17D24.18)
C
C CONTROLLO SULLA BONTA' DEI DATI
       IF(DRJC.NE.0.D0.OR.DWGH.NE.0.D0) THEN
        ICO=IRC+NIN-1
        WRITE(1,*) ' RECORD #',ICO,IC3
        WRITE(1,*) ' ERRORE NEI DATI: REJCOD =',DRJC,' WEIGHT =',DWGH
        WRITE(1,*) ' --------------- '
       ENDIF
C
```

```
C ELABORAZIONE DEI DATI
       LHMS=LHMS/10.
       LTRM=IT*10000.+IR*100.+IM
       CALL ELEV(IT,DHRG,DDEC,EL)
       IELUP=DG1*100.
       IELDW=EL*100.
       LE=IELUP*10000.+IELDW
       LB=DRES*1000000.
       LFRQ=DFRQ
C
C TRASFERIMENTO DEI DATI NEL BUFFER DI USCITA
       IF(LMAT(IRM,1).EQ.LYD.AND.LMAT(IRM,2).EQ.LHMS.AND.LMAT(IRM,3)
     + .EQ.LFRQ.AND.LMAT(IRM,4).EQ.LTRM.AND.LMAT(IRM,5).EQ.LE) THEN
C
C MISURA CONTEMPORANEA ALLA PRECEDENTE
        IF(IB.EQ.11) THEN
C S-BAND
         LMAT(IRM,6)=LB
         LMAT(IRM,9)=DOBS
                    ELSE
C X-BAND
         LMAT(IRM,7)=LB
         LMAT(IRM,10)=DOBS
         IF(IB.NE.21) THEN
          LMAT(IRM,12)=999999999
                     ELSE
          TBD=LMAT(IRM,6)-3./11.*LMAT(IRM,7)
          LMAT(IRM,8)=TBD
         ENDIF
        ENDIF
                                                                 ELSE
C TRASFERIMENTO IN UNA NUOVA RIGA DEL BUFFER LMAT
        IRM=IRM+1
C
C CONTROLLO SUL RIEMPIMENTO DEL BUFFER DI USCITA
        IF(IRM.EQ.161) THEN
         IF(IPRINT2.EQ.1) THEN
          ILM2=ILM2+1
          CALL XTAPE(2,ITAPE1,LMAT,4800,LEN)
          IF(LEN.NE.4800) THEN
           WRITE(1,*) 'ERRORE IN SCRITTURA SU',ITAPE1
           WRITE(1,*) 'LMAT #',ILM2,' LUN =',LEN
          ENDIF
          WRITE(1,*) '**SCRITTO SU',ITAPE1,' S. RECORD #',ILM2
         ENDIF
         IF(IPRINT1.EQ.1) THEN
          ILM1=ILM1+1
          WRITE(1,*) 'SCRITTURA SU',IWR1,' S. RECORD #',ILM1
          DO 1 MM=1,160
    1     WRITE(IWR1,200) MM,(LMAT(MM,NN),NN=1,15)
  200    FORMAT(I3,I6,I7,I9,I7,I9,I8,I9,I6,I7,I8,5I7)
         ENDIF
         DO 2 MM=1,160
```

```
            DO 2 NN=1,15
     2     LMAT(MM,NN)=0
            IRM=1
          ENDIF
C
C MISURA NON CONTEMPORANEA ALLA PRECEDENTE
          LMAT(IRM,1)=LYD
          LMAT(IRM,2)=LHMS
          LMAT(IRM,3)=DFRQ
          LMAT(IRM,4)=LTRM
          LMAT(IRM,5)=LE
          LMAT(IRM,11)=DRJC*10.**6
          LMAT(IRM,12)=DWGH*10.**6
          IF(IB.EQ.11) THEN
C S-BAND
           LMAT(IRM,6)=LB
           LMAT(IRM,9)=DOBS
                        ELSE
C X-BAND
           LMAT(IRM,7)=LB
           LMAT(IRM,10)=DOBS
          ENDIF
         ENDIF
C
C RITORNO AL MAIN
       ENDIF
       RETURN
C
C ERRORE DI DECODIFICA DEL BUFFER
  900 WRITE(1,*) '**ERRORE DI DECODIFICA DEL RECORD**'
       STOP
C
C FINE DI CONTROLLO
       END
C
C ***********************************************************************
C QUESTA ROUTINE CALCOLA L' ELEVAZIONE
       SUBROUTINE ELEV(IT,DHRG,DDEC,EL)
C
C DICHIARAZIONI
       DOUBLE PRECISION DHRG,DDEC
C
C RICERCA DELLA LATITUDINE DELLA STAZIONE IN QUESTIONE
       IF(IT.EQ.12) THEN
        LATD=35
        LATP=17
                        ELSE
        IF(IT.EQ.14) THEN
         LATD=35
         LATP=25
                         ELSE
         IF(IT.EQ.42) THEN
          LATD=-35
```

```
          LATP=24
                             ELSE
          IF(IT.EQ.43) THEN
           LATD=-35
           LATP=24
                              ELSE
           IF(IT.EQ.61) THEN
            LATD=40
            LATP=25
                               ELSE
            IF(IT.EQ.63) THEN
             LATD=40
             LATP=25
            ENDIF
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
C
C CALCOLA L' ELEVAZIONE
       PIG=3.141592654
       CON=PIG/180.
       IF(LATD.GT.0) THEN
        ALAT=CON*(LATD+LATP/60.)
                         ELSE
        ALAT=CON*(LATD-LATP/60.)
       ENDIF
       RHRG=DHRG*CON
       RDEC=DDEC*CON
       SINEL=COS(RDEC)*COS(ALAT)*COS(RHRG)+SIN(RDEC)*SIN(ALAT)
       EL=ASIN(SINEL)
       EL=EL/CON
C
C RITORNO AL MAIN
       RETURN
C
C FINE DI ELEV
       END
```

```
FTN7X,S
C
      PROGRAM  STATREGRESS(),                              (880329.1415)
C
C *******************************************************************
C *  LISTA RIASSUNTIVA DEI DATI STANDARD                              *
C *******************************************************************
C *                                                                   *
C *  QUESTO PROGRAMMA OPERA SUI DATI STANDARD                         *
C *                                                                   *
C *  ESSO PRODUCE IN USCITA UNA LISTA RIASSUNTIVA DEL NUMERO          *
C *  DI DATI STANDARD SEPARANDOLI, PER OGNI GIORNO DI DATI            *
C *  RICHIESTO, PER ORA, STAZIONE, BANDA E MODO DI COLLEGAMENTO       *
C *                                                                   *
C *******************************************************************
C
C
C DICHIARAZIONI
      INTEGER*4 LMAT(160,15),IYD,ZERO
      DIMENSION NATR(48,24)
      CHARACTER B
C
C INFORMAZIONI DI INGRESSO/USCITA
      WRITE(1,100)
  100 FORMAT(1X,20/,80'*')
      WRITE(1,*) ' LISTA RIASSUNTIVA DEI DATI STANDARD'
      WRITE(1,*) ' ***********************************'
      WRITE(1,*)
      WRITE(1,*) ' QUESTO PROGRAMMA PRODUCE ITERATIVAMENTE,'
      WRITE(1,*) ' PER OGNI GIORNO DI DATI STANDARD RICHIESTO,'
      WRITE(1,*) ' UNA LISTA RIASSUNTIVA DEL NUMERO DI DATI'
      WRITE(1,*) ' SEPARANDOLI PER ORA, STAZIONE, BANDA E MODO.'
      WRITE(1,101)
  101 FORMAT(1X,/,80'*',/)
   10 WRITE(1,'("NASTRO DI LETTURA (ENTER 8 OR 9) #:  ")')
      READ(1,*) ITAPE
      IF(ITAPE.NE.8.AND.ITAPE.NE.9) GO TO 10
      REWIND ITAPE
   30 WRITE(1,'("GIORNO INIZIALE (DA 1 TO 365):  ")')
      READ(1,*) IDAYI
      IF(IDAYI.LE.0.OR.IDAYI.GE.366) GO TO 30
   40 WRITE(1,'("GIORNO FINALE (DA 1 TO 365) #:  ")')
      READ(1,*) IDAYF
      IF(IDAYF.LE.0.OR.IDAYF.GE.366) GO TO 40
   50 WRITE(1,'("ANNO:19 ")')
      READ(1,*) IYEAR
      IF(IYEAR.GE.100) GO TO 50
      WRITE(1,'("STAMPANTE #:  ")')
      READ(1,*) IWR
      ZERO=0
C
C PER CIASCUN GIORNO RICHIESTO
      DO 8 ID=IDAYI,IDAYF
```

```
      IYD=IYEAR*1000.+ID
C
C INIZIALIZZAZIONE
      NRC=0
      DO 1 II=1,24
       DO 1 JJ=1,48
    1 NATR(JJ,II)=0
      CALL EXEC(3,ITAPE+0200B)
      IPAGE=12
      WRITE(IWR,200) IPAGE
  200 FORMAT(A2)
C
C LETTURA DI UN RECORD
    2 CALL XTAPE(1,ITAPE,LMAT,4800,LEN)
      NRC=NRC+1
      IF(LEN.EQ.-1) THEN
       WRITE(1,*) 'EOF IN LETTURA'
       GO TO 60
      ENDIF
      IF(LEN.NE.4800) WRITE(1,*) '**ERRORE IN LETTURA: LUN.=',LEN
C
C PER CIASCUNA RIGA DEL RECORD
      DO 3 I=1,160
       IRG=I
C
C RICERCA DEL GIORNO RICHIESTO
       IGO=LMAT(I,1)-IYD
       IF(IGO.LT.0) GO TO 3
       IF(IGO.GT.0) GO TO 60
C
C ANALISI DEI DATI PER LA MTRICE
       IRM=LMAT(I,4)-LMAT(I,4)/10000*10000
       IR=IRM/100
       IM=IRM-IR*100-10
       IF(IR.EQ.12) THEN
        IRR=1
                    ELSE
        IF(IR.EQ.14) THEN
         IRR=2
                     ELSE
         IF(IR.EQ.42) THEN
          IRR=3
                      ELSE
          IF(IR.EQ.43) THEN
           IRR=4
                       ELSE
           IF(IR.EQ.61) THEN
            IRR=5
                        ELSE
            IF(IR.EQ.63) IRR=6
           ENDIF
          ENDIF
         ENDIF
```

```
          ENDIF
         ENDIF
         IH=LMAT(I,2)/10000
         IMS=LMAT(I,2)-IH*10000
         IBT=IH*60+IMS/100+1
C
C RIEMPIMENTO DELLA LISTA
         I2=(IBT-1)/60.+1
         IB=1
         I1=(IRR-1)*8+(IM-1)*2+IB
         IF(LMAT(I,6).NE.ZERO) NATR(I1,I2)=NATR(I1,I2)+1
         IB=2
         I1=(IRR-1)*8+(IM-1)*2+IB
         IF(LMAT(I,7).NE.ZERO) NATR(I1,I2)=NATR(I1,I2)+1
C
C SI RIPETE PER UN'ALTRA RIGA
    3   CONTINUE
C
C SI RIPETE PER UN ALTRO RECORD
        GO TO 2
C
C FINE DELLA RICERCA DEI DATI
   60   IYEAR1=LMAT(IRG,1)/1000.
        IDAY1=LMAT(IRG,1)-IYEAR1*1000.
        WRITE(1,*) 'FINE RICERCA IL',IDAY1,' DELL'' ANNO ',IYEAR1
C
C SCRITTURA DELLA MATRICE DI USCITA
        DO 4 JJ=1,48
         DO 4 II=1,24
          IF(NATR(JJ,II).NE.0) IPRINT1=1
    4   CONTINUE
        IF(IPRINT1.EQ.1) THEN
         IPRINT1=0
         WRITE(IWR,300)
  300    FORMAT(2X,132'.',/,/)
         WRITE(IWR,400) IYEAR,ID
  400 FORMAT(2X,'YEAR = ',I3,5X,'DAY = ',I4,9X,'HOUR =   0  1  2  3  4'
     *,'  5 :   6  7  8  9 10 11  :   12 13 14 15 16 17 :  18'
     *,' 19 20 21 22 23  NS DAY',/,/)
         DO 7 JJ=1,48
          DO 5 II=1,24
    5     IF(NATR(JJ,II).NE.0) IPRINT=1
          IF(IPRINT.EQ.1) THEN
           IPRINT=0
           IS=(JJ-1)/8.+1
           IF(IS.EQ.1) THEN
            ISS=12
                      ELSE
            IF(IS.EQ.2) THEN
             ISS=14
                       ELSE
             IF(IS.EQ.3) THEN
              ISS=42
```

```
                       ELSE
             IF(IS.EQ.4) THEN
              ISS=43
                         ELSE
              IF(IS.EQ.5) THEN
               ISS=61
                           ELSE
               IF(IS.EQ.6) ISS=63
              ENDIF
             ENDIF
            ENDIF
           ENDIF
          ENDIF
          IP=JJ/2.
          IF(IP*2.EQ.JJ) THEN
           B='X'
           IBB=2
                         ELSE
           B='S'
           IBB=1
          ENDIF
          IMM=(JJ-IBB-(IS-1)*8)/2+1
          ITOT=0
          DO 6 LL=1,24
    6      ITOT=ITOT+NATR(JJ,LL)
          IF(IS1.NE.IS) WRITE(IWR,*)
          IS1=IS
          WRITE(IWR,500) ISS,B,IMM,(NATR(JJ,LL),LL=1,24),ITOT
  500 FORMAT(2X,'DOP   :     DSS =',I3,'       Band-Mode   ',A1,I2,4X,6I3,
     *,' : ',6I3,'  :  ',6I3,' : ',6I3,I8)
         ENDIF
    7   CONTINUE
        WRITE(IWR,*)
        WRITE(IWR,300)
       ENDIF
C
C SI RIPETE PER UN ALTRO GIORNO
       IF(LEN.EQ.-1) GO TO 70
    8 CONTINUE
C
C FINE REGOLARE DELL' ANALISI
   70 WRITE(IWR,200) IPAGE
      WRITE(IWR,200) IPAGE
      WRITE(1,*)
      WRITE(1,*) '**FINE REGOLARE DELL' ANALISI**'
      STOP
C
C FINE DEL MAIN
      END
```

```
FTN7X,S
C
      PROGRAM PLOTREGRESS(),                                (880329.1415)
C
C ******************************************************************
C *  TRACCIAMENTO SEQUENZIALE DEI DATI STANDARD                      *
C ******************************************************************
C *                                                                *
C *  QUESTO PROGRAMMA OPERA SUI DATI STANDARD                        *
C *                                                                *
C *  ESSO TRACCIA TUTTI I DATI RELATIVI AL GIORNO, ALLA STAZIONE ED   *
C *  AL MODO RICHIESTO.                                              *
C *  QUESTI VENGONO TRACCIATI SEPARATI PER BANDA IN TRE GRAFICI   .  *
C *  NEL MASSIMO FORMATO.                                            *
C *                                                                *
C ******************************************************************
C
C DICHIARAZIONI
      EMA BS(720),BX(720),BD(720),BT(720)
      INTEGER*4 LMAT(160,15),IYD
      DIMENSION V(29),IST(6),IMODE(4)
      CHARACTER*50 N1,N2
      INTEGER YN6
      CHARACTER*2 ANNO
C
C ESEGUIBILE
C INFORMAZIONI PER INGRESSO/USCITA
      WRITE(1,100)
  100 FORMAT(1X,20/,80'*',/)
      WRITE(1,*) '  TRACCIAMENTO SEQUENZIALE DEI DATI STANDARD'
      WRITE(1,*) '  ******************************************'
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*) '  QUESTO PROGRAMMA TRACCIA TUTTI I DATI'
      WRITE(1,*) '  RELATIVI A CIASCUN GIORNO RICHIESTO'
      WRITE(1,*) '  IN SUCCESSIONE PER STAZIONI E MODI'
      WRITE(1,*)
      WRITE(1,*) '  I GRAFICI SONO 3 PER OGNI FOGLIO'
      WRITE(1,*) '  IL FORMATO E'' IL MASSIMO'
      WRITE(1,*)
      WRITE(1,*) ' '
      WRITE(1,*) '  PENNINI DEL PLOTTER:'
      WRITE(1,*) '   1: PER LE INTESTAZIONI'
      WRITE(1,*) '   2: COMPLESSO 10'
      WRITE(1,*) '   3: COMPLESSO 40'
      WRITE(1,*) '   4: COMPLESSO 60'
      WRITE(1,*)
      WRITE(1,101)
  101 FORMAT(1X,/,80'*',/)
   10 WRITE(1,'("NASTRO DI LETTURA (8 O 9) #: ")')
      READ(1,*) ITAPE
      IF(ITAPE.NE.8.AND.ITAPE.NE.9) GO TO 10
      REWIND ITAPE
```

```
 30 WRITE(1,'("GIORNO INIZIALE (DA 1 A 365) #: ")')
    READ(1,*) IDAYI
    IF(IDAYI.LE.0.OR.IDAYI.GE.366) GO TO 30
 40 WRITE(1,'("GIORNO FINALE (DA 1 A 365) #: ")')
    READ(1,*) IDAYF
    IF(IDAYF.LE.0.OR.IDAYF.GE.366) GO TO 40
 50 WRITE(1,'("ANNO:19 ")')
    READ(1,*) IYEAR
    IF(IYEAR.GE.100) GO TO 50
    WRITE(ANNO,102) IYEAR
102 FORMAT(I2)
    WRITE(1,'("VUOI L'' ECO SULLA 6? (Y=1): ")')
    READ(1,*) YN6
    IF(YN6.EQ.1) THEN
     IPAGE=12
     WRITE(6,200) IPAGE
200  FORMAT(A2)
    ENDIF
    NR=0
    NM=0
    DO I=1,6
     NR=NR+1
     WRITE(1,*) '***** 0 PER TERMINARE'
     WRITE(1,'("STAZIONE RICEVENTE? (12,14,42,43,61,63): ")')
     READ(1,*) IST(NR)
     IF(IST(NR).EQ.0) THEN
      NR=NR-1
      GO TO 60
     ENDIF
     IF(IST(NR).NE.12.AND.IST(NR).NE.14.AND.IST(NR).NE.42.AND.
   +IST(NR).NE.43.AND.IST(NR).NE.61.AND.IST(NR).NE.63) THEN
      NR=NR-1
      WRITE(1,*)
      WRITE(1,*) '*****STAZIONE NON RICONOSCIUTA!*****'
     ENDIF
    END DO
 60 DO I=1,4
     NM=NM+1
     WRITE(1,*) '***** 0 PER TERMINARE'
     WRITE(1,'("MODO RICHIESTO? (DA 1 A 4): ")')
     READ(1,*) IMODE(NM)
     IF(IMODE(NM).EQ.0) THEN
      NM=NM-1
      GO TO 70
     ENDIF
     IF(IMODE(NM).LT.1.OR.IMODE(NM).GT.4) THEN
      NM=NM-1
      WRITE(1,*)
      WRITE(1,*) '*****MODO NON RICONOSCIUTO!*****'
     ENDIF
    END DO
 70 CONTINUE
C
```

```
C PER CIASCUN GIORNO RICHIESTO
      DO 7 ID=IDAYI,IDAYF
      IYD=IYEAR*1000.+ID
C
C PER CIASCUN MODO RICHIESTO
      DO 5 IMM=1,NM
       IM=IMODF(IMM)
C
C PER CIASCUNA STAZIONE RICHIESTA
      DO 6 K=1,NR
      IS=IST(K)
C
C INIZIALIZZAZIONE
      IPO=0
      NRC=0
      DO 1 I=1,720
      BS(I)=0.
      BX(I)=0.
      BD(I)=0.
      BT(I)=0.
    1 CONTINUE
      IF(YN6.EQ.1) THEN
       WRITE(6,*)
       WRITE(6,*)
       WRITE(6,*) 'DOY =',ID,'  ANNO =',IYEAR,'  STAZIONE RIC.=',IS,
     +'  MODO =',IM
       WRITE(6,*)
       WRITE(6,*) 'MINUTO   S-BAND     X-BAND   NON RIPORTATE'
      ENDIF
      WRITE(1,*) 'DOY =',ID,'  ANNO =',IYEAR,'  STAZIONE RIC.=',IS,
     +'  MODO =',IM
      REWIND ITAPE
C
C LEGGE UN RECORD
    2 CALL XTAPE(1,ITAPE,LMAT,4800,LEN)
      NRC=NRC+1
      IF(LEN.NE.4800) THEN
       IF(LEN.EQ.-1) THEN
        WRITE(1,*) '**EOF SU NASTRO',ITAPE
        GO TO 4
                    ELSE
        WRITE(1,*) '**ERRORE IN LETTURA: LUN=',LEN,' SU ',ITAPE
       END IF
      END IF
C
C PER TUTTE LE RIGHE DEL BUFFER
      DO 3 I=1,160
       IRG=I
C
C RICERCA DEL GIORNO RICHIESTO
       IGO=LMAT(I,1)-IYD
       IF(IGO.LT.0) GO TO 3
       IF(IGO.GT.0) GO TO 4
```

```
C
C CONTROLLO DEL MODO E DELLA STAZIONE
        IRM=LMAT(I,4)-LMAT(I,4)/10000*10000
        IR=IRM/100
        IM1=IRM-IR*100-10
C
C SE IL DATO E' QUELLO CERCATO
        IF(IM.EQ.IM1.AND.IS.EQ.IR) THEN
C
C IMMAGAZZINAMENTO DEL DATO PER LA GRAFICA
         IPO=IPO+1
         BS(IPO)=LMAT(I,6)/1000000.
         IF(BS(IPO).GT..5) BS(IPO)=.5
         IF(BS(IPO).LT.-.5) BS(IPO)=-.5
         BX(IPO)=LMAT(I,7)/1000000.
         IF(BX(IPO).LT.-2.5) BX(IPO)=-2.5
         IF(BX(IPO).GT.2.5) BX(IPO)=2.5
         BD(IPO)=LMAT(I,8)/1000.
         IF(BD(IPO).GT.20) BD(IPO)=20
         IF(BD(IPO).LT.-20) BD(IPO)=-20
         IH=LMAT(I,2)/10000
         IMS=LMAT(I,2)-IH*10000
         BT(IPO)=IH*60+IMS/100+1
         IF((BS(IPO).EQ.0.OR.BX(IPO).EQ.0).AND.YN6.EQ.1) THEN
          WRITE(6,300) BT(IPO),LMAT(I,6),LMAT(I,7)
  300     FORMAT(2X,F5.0,I11,I11)
          IPO=IPO-1
         ENDIF
        ENDIF
C
C SI RIPETE PER UN' ALTERA RIGA
    3 CONTINUE
C
C SI  RIPETE PER UN ALTRO RECORD
      GO TO 2
C
C FINE RICERCA DEI DATI
    4 IYEAR1=LMAT(IRG,1)/1000.
      IDAY1=LMAT(IRG,1)-IYEAR1*1000
      WRITE(1,*) 'FINE RICERCA IL:',IDAY1,' DELL'' ANNO',IYEAR1
C
C IN ASSENZA DI DATI NON SI TRACCIA IL GRAFICO
      IF(BT(1).EQ.0) THEN
       WRITE(1,*) '**NON CI SONO DATI PER QUESTO INGRESSO**'
       IF(YN6.NE.1) GO TO 6
       WRITE(6,*) '**NON CI SONO DATI DA GRAFICARE PER QUESTO INPUT**'
       GO TO 6
                    ELSE
       WRITE(1,*) '**GRAFICI IN USCITA:',IPO,' PUNTI**'
       IF(YN6.EQ.1) THEN
        WRITE(6,*) '**GRAFICI IN USCITA:',IPO,' PUNTI'
       ENDIF
      ENDIF
```

```
C
C TRACCIAMENTO GRAFICI
      V(1)=0
      V(2)=-.5
      V(3)=.5
      V(4)=0
      V(5)=0
      V(6)=1440
      V(7)=50
      V(8)=30
      V(9)=350
      V(10)=90
      V(11)=2
      V(12)=3
      V(13)=60.
      V(14)=.02
      V(15)=4
      V(16)=5
      V(17)=1
      V(18)=0
      V(19)=0
      V(20)=0
      V(21)=0
      V(22)=1
      V(23)=3
      V(24)=4
      IF(IS.EQ.12.OR.IS.EQ.14) V(25)=2
      IF(IS.EQ.42.OR.IS.EQ.43) V(25)=3
      IF(IS.EQ.61.OR.IS.EQ.63) V(25)=4
      V(26)=1
      V(27)=1
      V(28)=0
      V(29)=1
      WRITE(1,*) 'TRACCIAMENTO PRIMO GRAFICO'
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BS,IPO,1.,0,1)
      WRITE(1,*) 'TRACCIATO PRIMO GRAFICO'
      N2=' '
      N1='  X-BAND [Hz]'
      V(2)=-2.5
      V(3)=2.5
      V(8)=100
      V(10)=160
      V(14)=.1
      WRITE(1,*) 'TRACCIAMENTO SECONDO GRAFICO'
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BX,IPO,1.,0,1)
      WRITE(1,*) 'TRACCIATO SECONDO GRAFICO'
      N2=' '
      N1='S-3/11*X [mHz]'
      V(2)=-20.
      V(3)=20.
      V(8)=170
```

```
      V(10)=230
      V(14)=1
      WRITE(1,*) 'TRACCIAMENTO TERZO GRAFICO'
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BD,IPO,1.,0,1)
      WRITE(1,*) 'TRACCIATO TERZO GRAFICO'
C
C SI RIPETE PER UN' ALTRA STAZIONE
      IF(YN6.EQ.1) WRITE(6,200) IPAGE
    6 CONTINUE
C
C COMPLETAMENTO DEL GRAFICO
      N2='   VOYAGER     19'//ANNO//'                REGRESS FILES'
      N1='  S-BAND [Hz]'
      V(1)=0
      V(2)=-.5
      V(3)=.5
      V(4)=0
      V(5)=0
      V(6)=1440
      V(7)=50
      V(8)=30
      V(9)=350
      V(10)=90
      V(11)=2
      V(12)=3
      V(13)=60.
      V(14)=.02
      V(15)=4
      V(16)=5
      V(17)=1
      V(18)=0
      V(19)=1
      V(20)=0
      V(21)=1
      V(22)=1
      V(23)=3
      V(24)=4
      V(25)=0
      V(26)=1
      V(27)=1
      V(28)=0
      V(29)=1
      WRITE(1,*) 'COMPLETAMENTO DEL GRAFICO'
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BS,IPO,1.,0,1)
      N2=' '
      N1='  X-BAND [Hz]'
      V(2)=-2.5
      V(3)=2.5
      V(8)=100
      V(10)=160
      V(14)=.1
```

```
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BX,IPO,1.,0,1)
      N2=' '
      N1='S-3/11*X [mHz]'
      V(2)=-20.
      V(3)=20.
      V(8)=170
      V(10)=230
      V(14)=1
      CALL SETPAR(V,N1,N2)
      CALL EGRAPHY(BT,BD,IPO,1.,0,1)
      CALL ZDLIM(0.,370.,0.,280.,IERR)
      IF(IERR.NE.0) WRITE(1,*) 'ZDLIM ERROR #',IERR
      CALL ZASPK(370.,280.)
      CALL ZVIEW(0.,370.,0.,280.)
      CALL ZWIND(0.,370.,0.,280.)
      CALL ZCOLR(1)
      CALL SYMBOL(43.,25.,-2.,'00.00',0.,5)
      CALL SYMBOL(93.,25.,-2.,'04.00',0.,5)
      CALL SYMBOL(143.,25.,-2.,'08.00',0.,5)
      CALL SYMBOL(193.,25.,-2.,'12.00',0.,5)
      CALL SYMBOL(243.,25.,-2.,'16.00',0.,5)
      CALL SYMBOL(293.,25.,-2.,'20.00',0.,5)
      CALL SYMBOL(343.,25.,-2.,'24.00',0.,5)
      CALL SYMBOL(38.,30.,-2.,'-.5',0.,3)
      CALL SYMBOL(35.,100.,-2.,'-2.5',0.,4)
      CALL SYMBOL(38.,170.,-2.,'-20.',0.,4)
      FDAY=FLOAT(ID)
      FDSS=FLOAT(IS)
      FMOD=FLOAT(IM)
      CALL SYMBOL(70.,260.,-3.,'DAY =',0.,5)
      CALL NUMBER(95.,260.,-4.,FDAY,0.,3,'(I3)')
      IF(NR.EQ.1) THEN
       CALL NUMBER(155.,260.,-4.,FDSS,0.,2,'(I2)')
       CALL SYMBOL(130.,260.,-3.,'DSS =',0.,5)
      ENDIF
      CALL SYMBOL(190.,260.,-3.,'MODE =',0.,6)
      CALL NUMBER(220.,260.,-4.,FMOD,0.,1,'(I1)')
      CALL ZNEWF
C
C SI RIPETE PER UN ALTRO MODO
    5 CONTINUE
C
C SI RIPETE PER UN ALTRO GIORNO
    7 CONTINUE
C
C DISATTIVAZIONE DEL PLOTTER
      CALL ZDEND
      CALL ZEND
C
C FINE REGOLARE DELL' ANALISI
      WRITE(1,*)
      WRITE(1,*) '**FINE REGULARE DELL''ANALISI**'
```

```
      REWIND ITAPE
      STOP
C
C FINE DEL MAIN
      END
```

```
FTN7X,S
$FILES(0,1)
C
      PROGRAM FFREGRESS(),                         (880329.1415)
C
C ********************************************************************
C * CREAZIONE DI DISC FILES DEI DATI STANDARD                         *
C ********************************************************************
C *                                                                  *
C * QUESTO PROGRAMMA OPERA SUI DATI STANDARD                         *
C *                                                                  *
C *                                                                  *
C * ESSO CREA DISC FILES A PARTIRE DAI DATI IMMAGAZZINATI SU NASTRO  *
C * CIASCUN FILE CONTIENE DATI RELATIVI AD UN SINGOLO PASSAGGIO      *
C * AVENDO CIOE' FISSATO LE STAZIONI ATTIVE, IL MODO ED IL GIORNO .  *
C * DELL'INIZIO DEL PASSAGGIO                                        *
C * IL FILE CONTIENE TUTTI I DATI NELLE BANDE S E X E LA FREQUENZA   *
C * DELL' OSCILLATORE.                                               *
C *                                                                  *
C * N.B. SI E' ASSUNTO CHE FRA UN PASSAGGIO E L'ALTRO CI SIANO 8 ORE *
C *                                                                  *
C ********************************************************************
C
C DICHIARAZIONI
      INTEGER*4 LMAT(160,15),ISTORE(3,730),ANGDAT(120,6)
      INTEGER*4 IYEAR,IYD,ZERO
      DIMENSION IDSR(6),IDST(6),IMODE(4)
      CHARACTER*3 NY
      COMMON IPO,IPOS,IPOX,IAUTO,ISTORE,ANGDAT,LU
C
C ESEGUIBILE
C INFORMAZIONI DI INGRESSO/USCITA
      WRITE(1,100)
  100 FORMAT(1X,20/,80'*',/)
      WRITE(1,*) '       CREAZIONE DI DISC FILES DEI DATI STANDARD'
      WRITE(1,*) '       *****************************************'
      WRITE(1,*)
      WRITE(1,*)
      WRITE(1,*) '  QUESTO PROGRAMMA CREA DISC FILES A PARTIRE DAI'
      WRITE(1,*) '  NASTRI STANDARD.'
      WRITE(1,*) '  CIASCUN FILE CONTIENE DATI RELATIVI AD UN SINGOLO'
      WRITE(1,*) '  PASSAGGIO NELLE DUE BANDE S E X E LA FREQUENZA'
      WRITE(1,*) '  DELL''OSCILLATORE.'
      WRITE(1,200)
  200 FORMAT(1X,/,80'*',/)
   10 WRITE(1,*) '***PARAMETRI PER L'' ANALISI:'
      WRITE(1,'("NASTRO IN LETTURA (8 O 9) #: ")')
      READ(1,*) ITAPE
      IF(ITAPE.NE.8.AND.ITAPE.NE.9) GO TO 10
      REWIND ITAPE
      WRITE(1,'("QUANTI RECORDS? : ")')
      READ(1,*) NFI
      WRITE(1,'("VUOI L''ANALISI AUTOMATICA? (1=SI): ")')
```

```
      READ(1,*) IAUTO
      WRITE(1,'("STAMPANTE #: ")')
      READ(1,*) LU
      IF(LU.NE.1) THEN
       WRITE(LU,300)
300    FORMAT(5X,'FILE NAMES',6X,'IN DAY    FIN DAY',6X,
     +'YEAR      IN MIN    FIN MIN',6X,'MODE    TRS DSS    REC DSS',
     +5X,'R ASC',7X,'DEC',6X,'RTLT')
       WRITE(LU,*)
      ENDIF
   20 WRITE(1,'("GIORNO INIZIALE? (DA 1 A 365): ")')
      READ(1,*) IDAYI
      IF(IDAYI.LT.1.OR.IDAYI.GT.365) GO TO 20
   30 WRITE(1,'("DELL''ANNO: 19 ")')
      READ(1,*) IYEAR
      IF(IYEAR.GT.99.OR.IYEAR.LT.1) GO TO 30
      ZERO=0
      NR=0
      NT=0
      NM=0
      DO 1 I=1,6
       NR=NR+1
       WRITE(1,*) '**** 0 PER TERMINARE'
       WRITE(1,'("STAZIONE RICEVENTE? (12,14,42,43,61,63): ")')
       READ(1,*) IDSR(NR)
       IF(IDSR(NR).EQ.0) THEN
        NR=NR-1
        GO TO 40
       ENDIF
       IF(IDSR(NR).NE.12.AND.IDSR(NR).NE.14.AND.IDSR(NR).NE.42.AND.
     +IDSR(NR).NE.43.AND.IDSR(NR).NE.61.AND.IDSR(NR).NE.63) THEN
        NR=NR-1
        WRITE(1,*)
        WRITE(1,*) '****STAZIONE NON RICONOSCIUTA!*****'
       ENDIF
    1 CONTINUE
   40 DO 2 I=1,4
       NM=NM+1
       WRITE(1,*) '**** 0 PER TERMINARE'
       WRITE(1,'("MODO? (DA 1 A 4): ")')
       READ(1,*) IMODE(NM)
       IF(IMODE(NM).EQ.0) THEN
        NM=NM-1
        GO TO 50
       ENDIF
       IF(IMODE(NM).LT.1.OR.IMODE(NM).GT.4) THEN
        NM=NM-1
        WRITE(1,*)
        WRITE(1,*) '****MODO NON RICONOSCIUTO!*****'
       ENDIF
    2 CONTINUE
   50 DO I=1,6
       NT=NT+1
```

```
         WRITE(1,*) '**** 0 PER TERMINARE'
         WRITE(1,'("STAZIONE TRASMITTENTE? (12,14,42,43,61,63): ")')
         READ(1,*) IDST(NT)
         IF(IDST(NT).EQ.0) THEN
          NT=NT-1
          GO TO 51
         ENDIF
         IF(IDST(NT).NE.12.AND.IDST(NT).NE.14.AND.IDST(NT).NE.42.AND.
     +IDST(NT).NE.43.AND.IDST(NT).NE.61.AND.IDST(NT).NE.63) THEN
          NT=NT-1
          WRITE(1,*)
          WRITE(1,*) '****STAZIONE NON RICONOSCIUTA'
         ENDIF
        END DO
C
C LEGGE I DATI ANGOLARI
   51 WRITE(1,*)
      WRITE(1,*) 'LETTURA FILE DAT/FILE#1'
      OPEN(70,FILE='DAT/FILE#1',IOSTAT=IOSO,STATUS='OLD',ERR=60)
      READ(70,*) ANGDAT
      CLOSE(70)
      GO TO 70
   60 WRITE(1,*) 'ERRORE #',IOSO,' IN OPEN IL FILE DAT/FILE#1'
      WRITE(1,'(" VUOI ABORTIRE IL PROGRAMMA? (Y/N): ")')
      READ(1,400) NY
  400 FORMAT(A3)
      IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1')
     +STOP '***FINE IRREGOLARE DELL'' ANALISI***'
      GO TO 10
C
C SI FISSANO I PARAMETRI
   70 DO 5 I1=1,NR
        IDSSR=IDSR(I1)
        IDSST=IDSSR
        I3=0
        DO 5 I2=1,NM
   71    IMODES=IMODE(I2)
         IF(IMODES.GT.2) THEN
          I3=I3+1
          IF(I3.GT.NT) GO TO 5
          IDSST=IDST(I3)
          IF(IDSST.EQ.IDSSR) GO TO 71
         ENDIF
         WRITE(1,*)
         WRITE(1,*) 'RICERCA: MODO =',IMODES,' DSS RICEVENTE=',IDSSR,
     +' DSS TRASMITTENTE=',IDSST
C
C INIZIALIZZAZIONE
         REWIND ITAPE
         IPO=0
         IPOS=0
         IPOX=0
         IBTF=0
```

```
      IBTA=0
      INIT=0
      NEWDAY=0
      IDAY=IDAYI
      DO 3 II=1,3
       DO 3 JJ=1,730
       ISTORE(II,JJ)=0
    3 CONTINUE
      IYD=IYEAR*1000.+IDAY
      ITC=480
C
C LEGGE UN RECORD
      WRITE(1,*)
      WRITE(1,*) 'LEGGENDO DAL NASTRO',ITAPE
      NRC=0
 1000 CALL XTAPE(1,ITAPE,LMAT,4800,LEN)
      NRC=NRC+1
      IF(LEN.EQ.-1) GO TO 2000
      IF(LEN.NE.4800) WRITE(1,*) '**ERRORE IN LETTURA: LUN=',LEN,'**'
C
C PER CIASCUNA RIGA DEL BUFFER
   90 CONTINUE
      DO 4 I=1,160
C
C RICERCE DEL GIORNO RICHIESTO
          IGO=LMAT(I,1)-IYD
          IF(IGO.LT.0) GO TO 4
          IF(IGO.GT.0) THEN
           IDAY=IDAY+1
           IYD=IYD+1
           IF(INIT.EQ.0) GO TO 90
           NEWDAY=1
           IBTA=1440-IBTF
           ISTORE(2,9)=IBTF
           IBTF=0
          ENDIF
C
C ANALISI DEI DATI
          IST=LMAT(I,4)/10000
          IRM=LMAT(I,4)-IST*10000
          ISR=IRM/100
          IM=IRM-ISR*100-10
          IH=LMAT(I,2)/10000
          IMS=LMAT(I,2)-IH*10000
          IBT=IH*60+IMS/100
C
C FINE DEL PASSAGGIO
          IF((IBT-IBTF+IBTA).GT.ITC) THEN
           CALL CREA
           NEWDAY=0
           IBTF=IBT
           IBTA=0
          ENDIF
```

```
C
C DATO DA TRASFERIRE NEL BUFFER DI USCITA
         IF(ISR.EQ.IDSSR.AND.IM.EQ.IMODES.AND.IST.EQ.IDSST) THEN
          IPO=IPO+1
C
C PRIMO DATO PER IL BUFFER DI USCITA (RIEMPIMENTO DELLA TESTATA)
          IF(IPO.EQ.1) THEN
           IF(IAUTO.NE.1) THEN
            WRITE(1,*)
            WRITE(1,*) 'DATI PER QUESTO INPUT'
           ENDIF
           ISTORE(1,8)=IDAY
           ISTORE(1,9)=IDAY
           ISTORE(2,8)=IBT
           ISTORE(3,1)=2
           ISTORE(3,2)=IYEAR
           ISTORE(3,3)=IDSST
           ISTORE(3,4)=IDSSR
           ISTORE(3,5)=IMODES
           INIT=1
           NEWDAY=0
          ENDIF
C
C IMMAGAZZINAMENTO DEI DATI
          IBTF=IBT
          ISTORE(1,9)=IDAY
          ISTORE(2,9)=IBT
          IT=IBT-ISTORE(2,8)+11
          IF(NEWDAY.EQ.1) IT=IT+1440
          IF(IT.GT.721) THEN
           WRITE(1,'("VUOI ABORTIRE IL PROGRAMMA? (Y/N): ")')
           READ(1,400) NY
           IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1')
     +STOP '** FINE IRREGOLARE DELL''ANALISI***'
          ENDIF
          ISTORE(1,IT)=LMAT(I,6)
          ISTORE(2,IT)=LMAT(I,7)
          ISTORE(3,IT)=LMAT(I,3)
C
C AGGIORNAMENTO DEI DATI NELLA TESTATA
          IF(LMAT(I,6).NE.ZERO) THEN
           IPOS=IPOS+1
           ISTORE(1,2)=IDAY
           ISTORE(1,4)=IBT
           ISTORE(1,7)=IT
           IF(IPOS.EQ.1) THEN
            ISTORE(1,1)=IDAY
            ISTORE(1,3)=IBT
            ISTORE(1,6)=IT
           ENDIF
          ENDIF
          IF(LMAT(I,7).NE.ZERO) THEN
           IPOX=IPOX+1
```

```
           ISTORE(2,2)=IDAY
           ISTORE(2,4)=IBT
           ISTORE(2,7)=IT
           IF(IPOX.EQ.1) THEN
            ISTORE(2,1)=IDAY
            ISTORE(2,3)=IBT
            ISTORE(2,6)=IT
           ENDIF
          ENDIF
         ENDIF
C
C SI RIPETE PER UN'ALTRA RIGA
    4   CONTINUE
C
C SI RIPETE PER UN ALTRO RECORD
        IF(NRC.NE.NFI) GO TO 1000
C
C FINE RICERCA DEI DATI
 2000   CALL CREA
        IYEAR1=LMAT(160,1)/1000.
        IDAY1=LMAT(160,1)-IYEAR1*1000.
        WRITE(1,*)
        WRITE(1,*) 'FINE AL GIORNO:',IDAY1,' DELL''ANNO',IYEAR1
        WRITE(1,*)
C
C SI PROCEDE CON ALTRI PARAMETRI
      IF(I3.GT.0) GO TO 71
    5 CONTINUE
C
C OPZIONE PER RIPETERE L' ANALISI CON ALTRI DATI
      IF(IAUTO.EQ.1) WRITE(1,*) ''
      WRITE(LU,*)
      REWIND ITAPE
      WRITE(1,'("VUOI IMMAGAZZINARE ALTRI DATI? (Y/N): ")')
      READ(1,400) NY
      IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1') GO TO 10
C
C FINE REGOLARE DELL'ANALISI
      REWIND ITAPE
      WRITE(1,*)
      WRITE(1,*) '**FINE REGOLARE DELL''ANALISI**'
      STOP
C
C FINE DEL MAIN
      END
C
C ***********************************************************************
C QUESTA SUBROUTINE CREA I DISK FILES
      SUBROUTINE CREA
C
      INTEGER*4 ISTORE(3,730),ANGDAT(120,6)
      CHARACTER*14 NFILE
      CHARACTER*3 NY
```

```
      COMMON IPO,IPOS,IPOX,IAUTO,ISTORE,ANGDAT,LU
C
C SE CI SONO DATI
      IF(IPO.NE.0) THEN
       IF(IAUTO.NE.1) WRITE(1,*) 'CI SONO',IPO,' DATI'
C
C RICERCA DEI DATI ANGOLARI
       CALL SEARCH
C
C CREAZIONE DEL FILENAME
       WRITE(NFILE,100)ISTORE(3,3),ISTORE(3,4),ISTORE(1,8)
  100  FORMAT('DAT/ROD',I2,I2,I3)
C
C SALTO DELLE RICHIESTE DA INPUT SE IL LAVORO E' AUTOMATICO
       IF(IAUTO.EQ.1) THEN
        NY='YES'
        GO TO 10
       ENDIF
C
C SCRIVE IL DISC FILE RICHIESTO
       WRITE(1,*)
       WRITE(1,*) 'IL FILE ',NFILE,' CONTIENE I SEGUENTI DATI:'
       IF(ISTORE(3,1).EQ.2) WRITE(1,*) '***VALORI REGRESS'
       WRITE(1,*) '***BANDE: S, X E FREQUENZA'
       WRITE(1,*) '***GIORNO INIZIALE:',ISTORE(1,8)
       WRITE(1,*) '***GIORNO FINALE:',ISTORE(1,9)
       WRITE(1,*) '***ANNO:',ISTORE(3,2)
       WRITE(1,*) '***MINUTO INIZIALE:',ISTORE(2,8)
       WRITE(1,*) '***MINUTO FINALE:',ISTORE(2,9)
       WRITE(1,*) '***MODO:',ISTORE(3,5)
       WRITE(1,*) '***DSS TRASMITTENTE:',ISTORE(3,3)
       WRITE(1,*) '***DSS RICEVENTE:',ISTORE(3,4)
       WRITE(1,*) '***ASCENSIONE RETTA:',ISTORE(3,7)
       WRITE(1,*) '***DECLINAZIONE:',ISTORE(3,8)
       WRITE(1,*) '***RTLT:',ISTORE(3,6)
       WRITE(1,*)
       WRITE(1,'(" OK AD IMMAGAZZINARE? <Y/N>: ")')
       READ(1,200) NY
  200  FORMAT(A3)
       IF(NY.EQ.'A') STOP '***FINE IRREGOLARE DELL''ANALISI***'
C
C SCRITTURA DEL FILE SE RICHIESTO
   10  IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1') THEN
       IF(LU.NE.1) WRITE(LU,300) NFILE,ISTORE(1,8),ISTORE(1,9),
     +ISTORE(3,2),ISTORE(2,8),ISTORE(2,9),ISTORE(3,5),ISTORE(3,3),
     +ISTORE(3,4),ISTORE(3,7),ISTORE(3,8),ISTORE(3,6)
  300  FORMAT(1X,A16,11(I10))
        WRITE(1,*)
        WRITE(1,*) 'SCRITTURA DEL FILE ',NFILE
        ISTORE(1,5)=IPOS
        ISTORE(2,5)=IPOX
        ISTORE(3,9)=IPO
        OPEN(70,FILE=NFILE,IOSTAT=IOSO,ERR=20)
```

```
        WRITE(70,*) ISTORE
        CLOSE(70,IOSTAT=IOSC,ERR=30)
                                        ELSE
        WRITE(1,*) 'FILE ',NFILE,' NON CREATO'
       ENDIF
C
C REINIZIALIZZAZIONE DEI BUFFERS
       IPO=0
       IPOS=0
       IPOX=0
       DO 1 II=1,3
        DO 1 JJ=1,730
         ISTORE(II,JJ)=0
    1  CONTINUE
      ENDIF
C
C RITORNO AL MAIN
      RETURN
C
C ERRORR IN OPEN FILE 70
   20 WRITE(1,*) '**ERRORE IN OPEN FILE ',NFILE,' #',IOSO,'**'
      STOP
C
C ERRORE IN CLOSE FILE 70
   30 WRITE(1,*) '**ERRORE IN CLOSE FILE ',NFILE,' #',IOSC,'**'
      STOP
C
C FINE DI CREA
      END
C
C ******************************************************************
C QUESTA SUBROUTINE CERCA I DATI ANGOLARI
      SUBROUTINE SEARCH
C
      INTEGER*4 ISTORE(3,730),ANGDAT(120,6)
      INTEGER*4 SUMMIN,MINMED,IDHM,FDHM
      CHARACTER*3 NY
      COMMON IPO,IPOS,IPOX,IAUTO,ISTORE,ANGDAT,LU
C
C RICERCA DEI DATI ANGOLARI
      SUMMIN=ISTORE(2,8)+ISTORE(2,9)+1.
      IF(ISTORE(1,8).NE.ISTORE(1,9)) SUMMIN=SUMMIN+1440.
      MINMED=SUMMIN/2
      IF(MINMED.GT.1440) MINMED=MINMED-1440
      D4=10000
      I60=60
      IDHM=ISTORE(1,8)*D4+MINMED/I60*100+MOD(MINMED,I60)
      KK=0
      DO 1 K=1,120
       IF(ISTORE(3,2).LT.ANGDAT(K,1)/1000) GO TO 10
       FDHM=(ANGDAT(K,1)-ISTORE(3,2)*1000)*D4+ANGDAT(K,2)/100
       IF(FDHM.LT.IDHM) GO TO 1
       KK=K
```

```
        GO TO 10
      1 CONTINUE
C
C DATI NON TROVATI
   10  IF(KK.EQ.0) THEN
        WRITE(1,*) '**DATI ANGOLARI NON TROVATI!**'
        IF(LU.NE.1) WRITE(LU,*) 'DATI ANGOLARI NON TROVATI'
        IF(IAUTO.NE.1) THEN
         WRITE(1,'("VUOI ABORTIRE IL PROGRAMMA? <Y/N>: ")')
         READ(1,200) NY
  200    FORMAT(A3)
         IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1')
     +STOP '***IRREGOLARE FINE DEL PROGRAMMA***'
        ENDIF
        ISTORE(3,7)=ANGDAT(120,5)
        ISTORE(3,8)=ANGDAT(120,4)
        ISTORE(3,9)=ANGDAT(120,3)
        RETURN
       ENDIF
       KKK=KK-1
       IF(KKK.EQ.0) THEN
        ISTORE(3,7)=ANGDAT(1,5)
        ISTORE(3,8)=ANGDAT(1,4)
        ISTORE(3,9)=ANGDAT(1,3)
        RETURN
       ENDIF
C
C SI RIEMPIE IL BUFFER
       FS1=ANGDAT(KKK,2)-(ANGDAT(KKK,2)/100)*100
       FH1=ANGDAT(KKK,2)/D4
       FM1=(ANGDAT(KKK,2)-FH1*D4)/100
       FS2=ANGDAT(KK,2)-(ANGDAT(KK,2)/100)*100
       FH2=ANGDAT(KK,2)/D4
       FM2=(ANGDAT(KK,2)-FH2*D4)/100
       ITS=30
       ITD=IDHM/D4
       ITH=(IDHM-ITD*D4)/100
       ITM=(IDHM-ITD*D4-ITH*100)
       IF(FH2-FH1.LT.0) THEN
        FH2=FH2+24
        IF(IDHM-FDHM.LT.0) ITH=ITH+24
       ENDIF
       T2=FS2/60.+FM2+FH2*60.
       T1=FS1/60.+FM1+FH1*60.
       T=ITS/60.+ITM+ITH*60.
       DT=T2-T1
       IF(DT.EQ.0) THEN
        WRITE(1,*) '**RIPETIZIONE NEI DATI ANGOLARI**'
        IF(LU.NE.1) WRITE(LU,*) 'RIPETIZIONE NEI DATI ANGOLARI'
        IF(IAUTO.NE.1) THEN
         WRITE(1,'("VUOI ABORTIRE IL PROGRAMMA? <Y/N>: ")')
         READ(1,200) NY
         IF(NY.EQ.'Y'.OR.NY.EQ.'YES'.OR.NY.EQ.'1')
```

```
     +STOP '**FINE IRREGOLARE DELL'' ANALISI***'
      ENDIF
      ISTORE(3,7)=ANGDAT(KK,5)
      ISTORE(3,8)=ANGDAT(KK,4)
      ISTORE(3,9)=ANGDAT(KK,3)
     ENDIF
     ST=ANGDAT(KK,5)-ANGDAT(KKK,5)
     ISTORE(3,7)=ST/DT*(T-T1)+ANGDAT(KKK,5)
     ST=ANGDAT(KK,4)-ANGDAT(KKK,4)
     ISTORE(3,8)=ST/DT*(T-T1)+ANGDAT(KKK,4)
     ST=ANGDAT(KK,3)-ANGDAT(KKK,3)
     ISTORE(3,6)=(ST/DT*(T-T1)+ANGDAT(KKK,3))/10
C
C RITORNO ALLA CHIAMATA
      RETURN
C
C FINE DI SEARCH
      END
```

TAB. 1

REGRESS MEASUREMENT STRUCTURE

1 TIME 1: UTC (seconds 1950)

2 TIME 2: composite word containing UTC coded as year, day of year, hour, minute, second and fractions of second; these information are contained in the decimal part of time 2 as:

YYDDDHHMMSSfffff

3 IDENTIFICATION WORD: composite word containing sampling rate in hundredths of seconds in the first seven decimal figures; the 8th figure represents the downlink band (1=S, 2=X), the 9th one the tracking network (DSN=1), the 10th and 11th the transmitting station, the 12th and 13th the receiving station, the 14th the data type (1=Doppler), the 15th the tracking link mode (2=two way, 3=three way), the last figures are not used; the scheme is:

1.SSSSSSSBNTTRRKW

4 OBSERVABLE: observed frequency shift in Hertz, (same quantity as in the ATDF tapes).

5 FREQUENCY: oscillator frequency in Hz. The transmitted frequency is obtained multiplying FREQUENCY by 96.

6 PASS IDENTIFIER: composite word containing spacecraft identification code.

7 COMPUTED: frequency shift predicted by ODP

8 RTLT: round trip light time in seconds

9 HOUR ANGLE: composite word containing the spacecraft hour angle in degrees, minutes, seconds and fractions of seconds in the first figures; the code is:

1.DDMMSSffff

10 DECLINATION: composite word containing spacecraft declination in degrees, minutes, seconds and fractions of seconds; the code is the same as in item 9.

11 AZIMUTH: composite word containing spacecraft azimuth in in degrees, minutes, seconds and fractions of seconds; code as in item 9.

12 ELEVATION AT RECEIVING TIME: composite word containing spacecraft elevation at the receiving time in degrees, minutes, seconds and fractions of seconds; code as in item 9;

13 ELEVATION AT TRANSMITTING TIME: composite word containing spacecraft elevation at the transmitting time; code as in item 9.

14 DERIVATIVE OF THE ELEVATION AT RECEIVING TIME: in deg./s

15 DERIVATIVE OF THE ELEVATION AT TRANSMITTING TIME: in deg./s

16 RESIDUAL: Doppler residual in Hz, computed from OBSERVABLE minus COMPUTED (item 4 minus item 7).

17 REJECTION CODE: if this code is 0 the data are ok.

18 WEIGHT: data quatity indicator (used by ODP).

19 CALIBRATION: media calibration to Doppler residuals (in Hz).

20 CALIBRATED RESIDUAL: calibrated residual in Hz. It is computed by subtracting: RESIDUAL minus CALIBRATION (items 16 and 19).

TAB. 2

STRUCTURE OF REGRESS RECORDS

rows --->	1	6	9	33	57
columns 1	N1	N2	TIME TAG 1	TIME TAG 2	ID. WORD
2	N1	N2	OBSERVABLE	FREQUENCY	PASS ID.
3	N1	N2	COMPUTED	RTLT	HOUR ANGLE
4	N1	N2	DECLINATION	AZIMUTH	ELEV. T3
5	N1	N2	ELEV. T1	ELEV. DER. T1	ELEV. DER.T3
6	N1	N2	RESIDUAL	REJECTION	WEIGHT
7	N1	N2	CALIB. COMP.	MODELS	TIME TAG 1
8	.	.	.	.	.
.	.	.	omissis		
100	.	.	.	.	.

TAB. 3

EXAMPLE OF REGRESS RECORD

The data start from time 20h 00' 29" of the DOY 308 year 1980

```
1  1 .973281629999999999+009 .803082000299999998+016 .100060001114121300+017
1  2-.269094662833319999+005 .220277400000000000+008 .103080310204800000+017
1  3-.269090225907072255+005 .101894300547065213+005 .410241037489606019+002
1  4-.610047258413681025-001 .236355807440611173+003 .000000000000000000
1  5 .544776425522538426+002-.283824054044811239-002 .164946371770699485-003
1  6-.443692624774428168+000 .000000000000000000     .000000000000000000
1  7 .000000000000000000     .000000000000000000     .973281629999999999+009
1  8 .803082000299999998+016 .100060002114121300+017-.986680276999990000+005
1  9 .220277400000000000+008 .103080310204800000+017-.986664161639321474+005
1 10 .101894300547065213+005 .410241037489606019+002-.610047258413681014-001
1 11 .236355807440611173+003 .000000000000000000     .544776425522538426+002
1 12-.283824054044811239-002 .164946371770699485-003-.161153606685252271+001
1 13 .000000000000000000     .000000000000000000     .000000000000000000
1 14 .000000000000000000     .973281629999999999+009 .803082000299999998+016
1 15 .100060001114141200+017-.269156408999980000+005 .220277400000000000+008
```

. . . omissis

```
1 94 .000000000000000000     .973281749999999999+009 .803082002299999998+016
1 95 .100060001114121300+017-.268720277666650000+005 .220277400000000000+008
1 96 .103080310204800000+017-.268715821903272889+005 .101894286495735869+005
1 97 .415250900514604227+002-.609566732206611566-001 .236819925387191006+003
1 98 .000000000000000000     .544943592287294470+002-.285344413867476923-002
1 99 .113775830962588772-003-.445576337711003134+000 .000000000000000000
1100 .000000000000000000     .000000000000000000     .000000000000000000
```

TAB. 4

STRUCTURE OF STANDARD RECORDS

1	YYDDD	year and day
2	HHMMSS	hour, minute and second
3	FREQCY	oscillator frequency in Hz
4	TTRRNM	trans. and rec. stations, network and mode
5	ELupELdw	elevation in up and down link in degrees
6	S-RES	S-band residual in mmHz
7	X-RES	X-band residual in mmHz
8	S-3/11*X	quantity used in plasma calibration
9	S-OBS	S-band observable (in Hz)
10	X-OBS	X-band observable (in Hz)
11	REJCOD	rejection code
12	WEIGHT	datum weight
13		not used
14		not used
15		not used

TAB. 5

DATA TRANSFER FROM REGRESS FILE TO STANDARD FILE

INPUT DATA	OUTPUT DATA
1 TIME 1	not transferred
2 TIME 2	1) YYDDD 2) HHMMSS
3 ID. WORD	band indicator 4) TTRRMM
4 OBSERVABLE	9) S-OBS or 10) X-OBS
5 FREQUENCY	3) FREQCY
6 PASS ID.	not transferred
7 COMPUTED	not transferred
8 RTLT	not transferred
9 HOUR ANGLE	not transferred
10 DECLINATION	not transferred
11 AZIMUTH	not transferred
12, 13 ELEVATIONS	5) ELupELdw
14, 15 EL. DERIVATIVES	not transferred
16 RESIDUAL	6) S-RES or 7) X-RES
17 REJECTION CODE	11) REJCOD
18 WEIGHT	12) WEIGHT
19 CALIBRATION	not transferred,
20 CAL. RESIDUAL	not transferred.

TAB. 6

EXAMPLE OF STANDARD RECORD

Data start from time 20h 00' 29" of the DOY 308 year 1980

```
1 80308 200029 22027740 141213 54473792  -443692  -1611536  -4182
-26909  -98668       0       0       0       0       0

2  80308  200029  22027740 141412 54473792 -447566 -1624393 -4549
-26915  -98690       0       0       0       0       0

3 80308 200029 22027740 144213 54472920  -450149  -1638823  -3197
-31147 -114208       0       0       0       0       0

4  80308  200029  22027740 144313 54472920 -446804 -1639150   236
-31147 -114208       0       0       0       0       0

5 80308 200129 22027740 141213 54483776  -442122  -1610828  -2805
-26890  -98599       0       0       0       0       0

6  80308  200129  22027740 141412 54483776 -446126 -1623059 -3473
-26896  -98621       0       0       0       0       0

7 80308 200129 22027740 144213 54482936 -447568  -1638656    -661
-31130 -114146       0       0       0       0       0

8  80308  200229  22027740 141213 54493752 -445576 -1602485 -8534
-26872  -98530       0       0       0       0       0

9 80308 200229 22027740 141412 54493760  -447826  -1620098  -5981
-26878  -98553       0       0       0       0       0

10  80308  200229 22027740 144213 54492960 -446743 -1636628  -389
-31114 -114085       0       0       0       0       0

. . . omissis

160 80308 204029 22027740 141412 53653088 -445258 -1612198  -5567
-26197  -96058       0       0       0       0       0
```

EXAMPLE OF REGRESS TABLE

Doppler residuals available for the DOY 309 of the year 1980

YEAR = 80	DAY = 309		HOUR =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NS DAY
DOP :	DSS = 12	Band-Mode	S 3	0	0	0	0	0	0	0	0	0	0	0	0	55	55	60	56	60	59	60	60	18	40	27	0	550
DOP :	DSS = 12	Band-Mode	X 3	0	0	0	0	0	0	0	0	0	0	0	0	55	55	60	56	60	59	60	60	18	42	28	0	553
DOP :	DSS = 14	Band-Mode	S 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	60	60	54	46	60	19	28	0	384
DOP :	DSS = 14	Band-Mode	X 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	60	60	54	46	60	19	28	0	384
DOP :	DSS = 14	Band-Mode	S 3	0	0	0	0	0	0	0	0	0	0	0	10	60	57	59	0	0	0	0	0	0	23	1	0	210
DOP :	DSS = 14	Band-Mode	X 3	0	0	0	0	0	0	0	0	0	0	0	10	60	57	59	0	0	0	0	0	0	23	1	0	210
DOP :	DSS = 42	Band-Mode	S 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	60	60	21	29	0	209
DOP :	DSS = 42	Band-Mode	X 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	60	60	21	29	0	208
DOP :	DSS = 43	Band-Mode	S 2	60	44	60	60	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	21	60	377
DOP :	DSS = 43	Band-Mode	X 2	60	44	60	60	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	25	60	381
DOP :	DSS = 43	Band-Mode	S 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42	42	58	19	28	0	189
DOP :	DSS = 43	Band-Mode	X 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	42	58	19	28	0	190
DOP :	DSS = 61	Band-Mode	S 3	0	0	0	0	21	16	60	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	128
DOP :	DSS = 61	Band-Mode	X 3	0	0	0	0	21	16	60	31	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	129
DOP :	DSS = 63	Band-Mode	S 2	0	0	0	0	0	0	0	19	58	58	57	51	51	49	58	0	0	0	0	0	0	0	0	0	401
DOP :	DSS = 63	Band-Mode	X 2	0	0	0	0	0	0	0	19	58	58	57	51	51	49	58	0	0	0	0	0	0	0	0	0	401
DOP :	DSS = 63	Band-Mode	S 3	0	0	0	0	29	18	60	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137
DOP :	DSS = 63	Band-Mode	X 3	0	0	0	0	29	18	60	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137

TAB. 7

TAB. 8

ANGULAR DATA STRUCTURE

1 - TIME TAG (high part), coded as YYDDD

2 - TIME TAG (low part), coded as HHMMSS

3 - RTLTx10 it is given in seconds with an approximation of .1 sec; the stored value is IFIX(RTLT*10.)

4 - RAx100 the right ascension is given in degrees with an approximation of .01 degrees; the stored value is IFIX(RA*100.)

5 - DECx100 the declination is given in degrees with an approximation of .01 degrees; the stored value is IFIX(DEC*100.)

6 - ELOx100 the elongation is given in degrees with an approximation of .01 degrees; the stored value is IFIX(ELO*100.)

TAB. 9

LOGICAL FORMAT OF DOPPLER DATA FILES (ROD)

columns ↓ / rows ---->	1	2	3
1	S INITIAL DAY	X INITIAL DAY	DATA ID. (1=A 2=R)
2	S FINAL DAY	X FINAL DAY	YEAR
3	S INITIAL MINUTE	X INITIAL MINUTE	TRANSMITTING DSS
4	S FINAL MINUTE	X FINAL MINUTE	RECEIVING DSS
5	# of S-band DATA	# of X-band DATA	MODE
6	LOCATION OF THE FIRST S-BAND ITEM	LOCATION OF THE FIRST X-BAND ITEM	RTLT
7	LOCATION OF THE LAST S-BAND ITEM	LOCATION OF THE LAST X-BAND ITEM	RIGHT ASCENSION
8	INITIAL DAY	INITIAL MINUTE	DECLINATION
9	FINAL DAY	FINAL MINUTE	# of S or X DATA
10	0	0	0
11	:	:	:
.	S-BAND DATA :	X-BAND DATA :	OSCILLATOR FREQUENCY :
.	mmHz :	mmHz :	Hz :
730	V	V	V

N.B.:

1) S and X stand for S-band and X-band respectively
2) The value of the item DATA IDENTIFIER is 1 for ATDF data and 2 for REGRESS data

TAB. 10

FILE DAT/ROD1414309

1	309	309	2
2	309	309	80
3	903	903	14
4	1347	1347	14
5	384	384	2
6	11	11	10187
7	455	455	18571
8	309	903	-22
9	309	1347	384
10	0	0	0
11	-1716015139	2147483647	22027744
12	2147483647	2147483647	22027744
13	2147483647	2147483647	22027744
14	-458095	-1670888	22027720
15	-457467	-1671856	22027720
16	-457156	-1669117	22027720
17	-453513	-1655412	22027720
.	.	.	.
730	0	0	0

FIGURE CAPTIONS

Fig. 1 The Deep Space Network stations.

Fig. 2a Flow chart of the program DECODEREGRES.
2b Zoom of block A of the previous flow chart.

Fig. 3 Flow chart of the program STATREGRES.

Fig. 4 Output of the program PLOTREGRES.

Fig. 5 Flow chart of the program PLOTREGRES.

Fig. 6 Flow chart of the program FFREGRES.

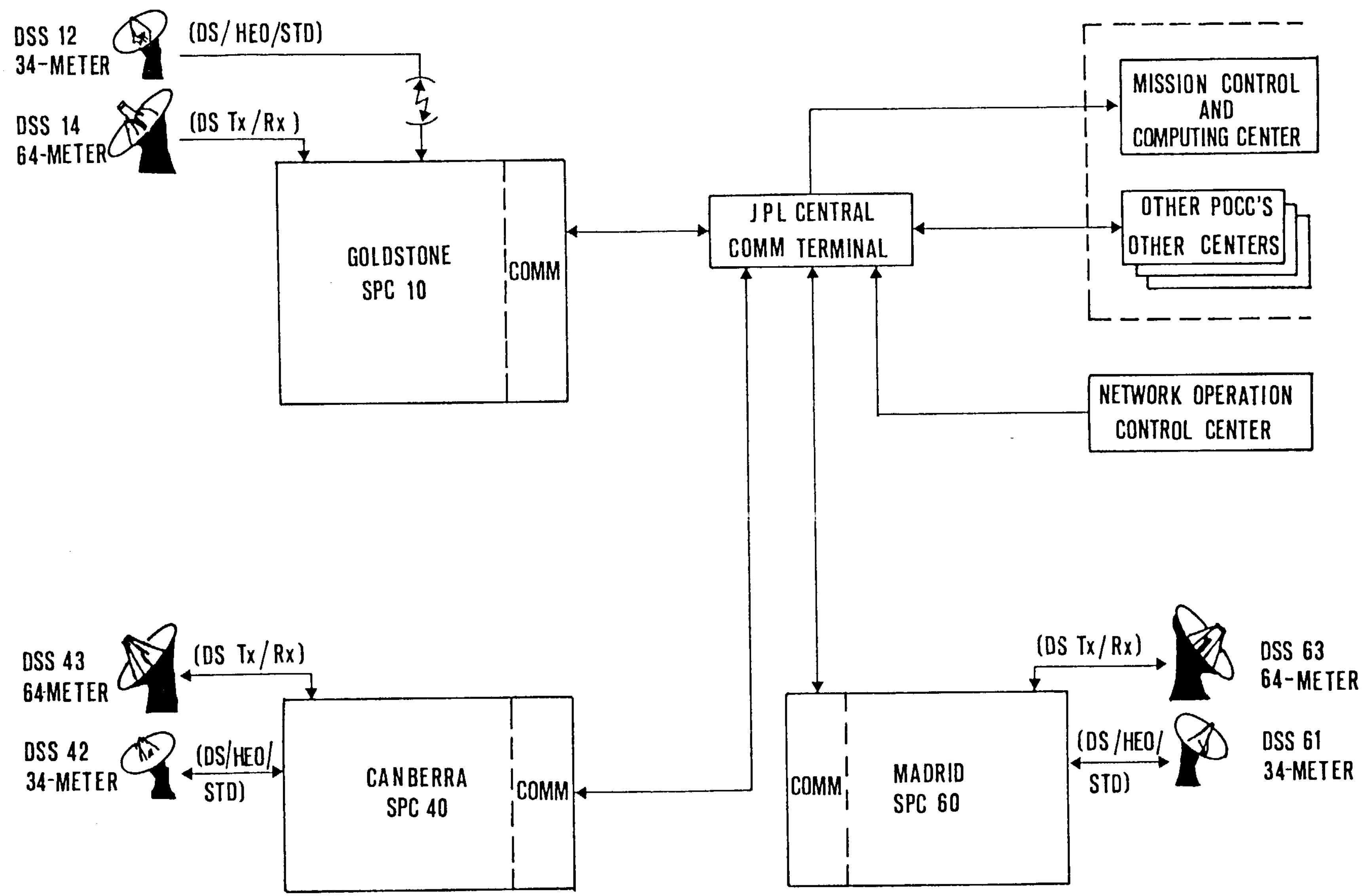
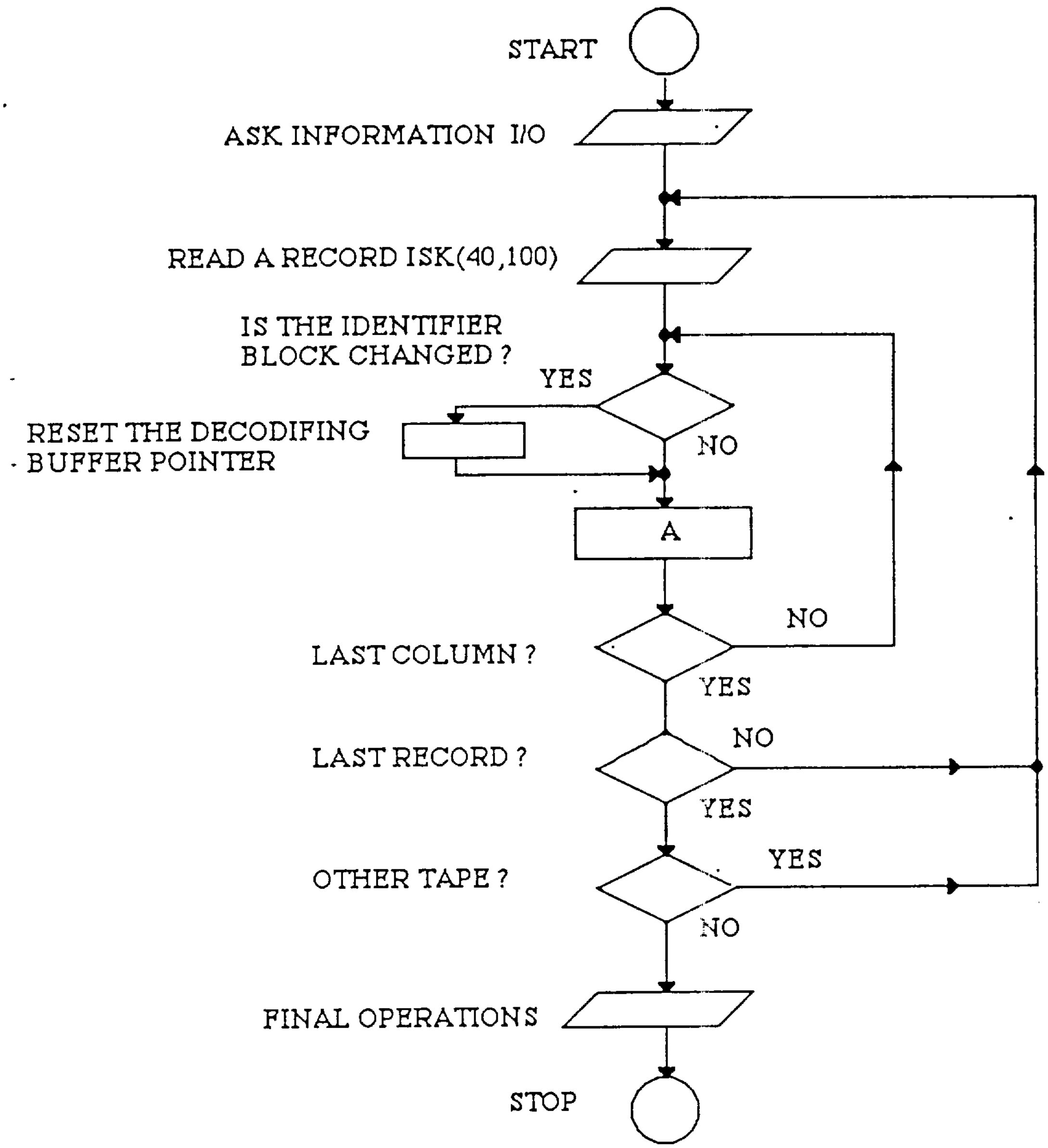

DSS 12
34-METER
(DS/HEO/STD)
DSS 14
64-METER
(DS Tx/Rx)
GOLDSTONE
SPC 10
COMM
JPL CENTRAL
COMM TERMINAL
MISSION CONTROL
AND
COMPUTING CENTER
OTHER POCC'S
OTHER CENTERS
NETWORK OPERATION
CONTROL CENTER
DSS 43
64METER
(DS Tx/Rx)
DSS 42
34-METER
(DS/HEO/
STD)
CANBERRA
SPC 40
COMM
COMM
MADRID
SPC 60
(DS Tx/Rx)
DSS 63
64-METER
(DS/HEO/
STD)
DSS 61
34-METER

FIG. 1



START
ASK INFORMATION I/O
READ A RECORD ISK(40,100)
IS THE IDENTIFIER BLOCK CHANGED ?
YES
RESET THE DECODIFING BUFFER POINTER
NO
A
NO
LAST COLUMN ?
YES
NO
LAST RECORD ?
YES
YES
OTHER TAPE ?
NO
FINAL OPERATIONS
STOP

FIG. 2 a

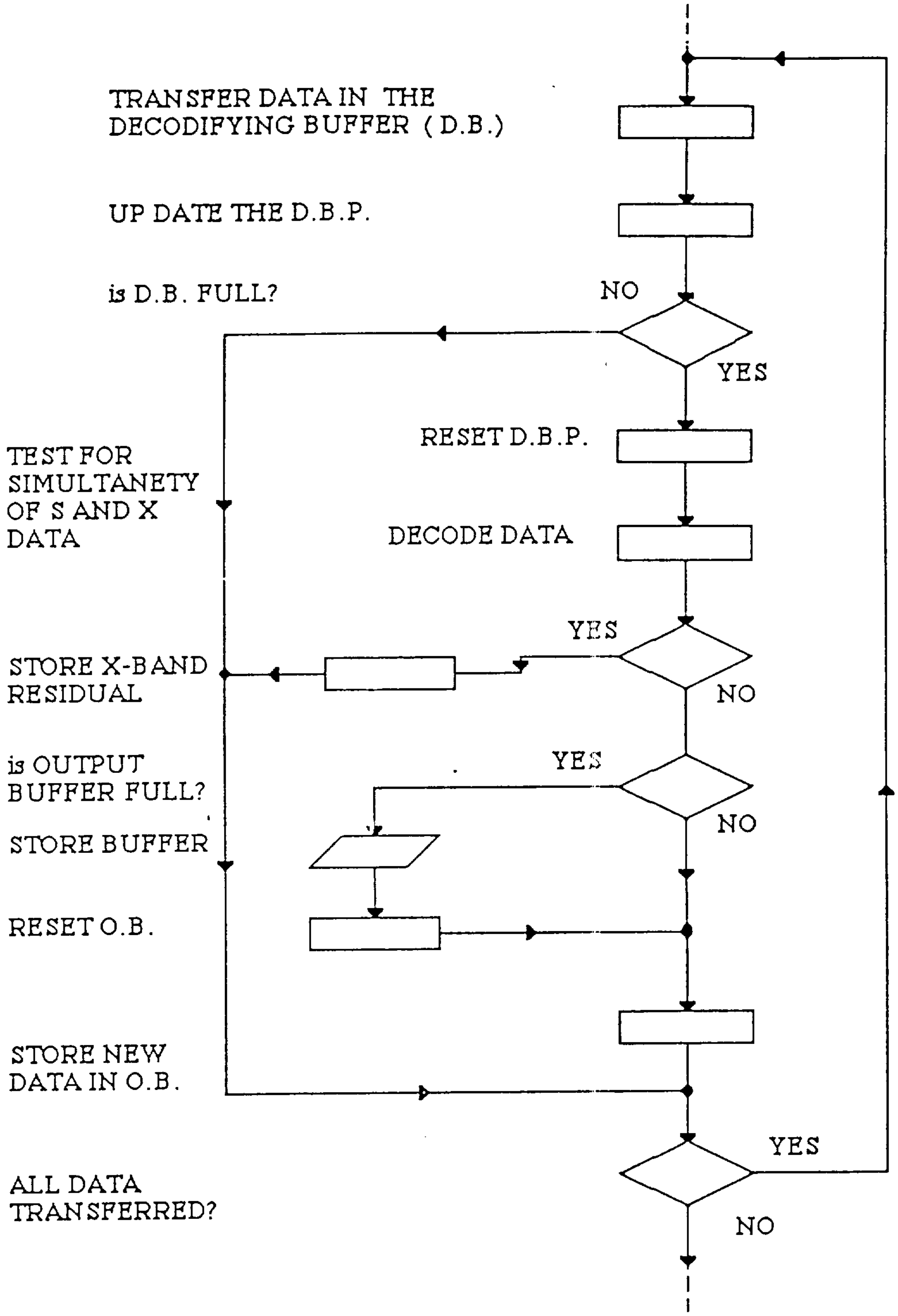

TRANSFER DATA IN THE DECODIFYING BUFFER (D.B.)
UP DATE THE D.B.P.
is D.B. FULL?
NO
YES
RESET D.B.P.
TEST FOR SIMULTANETY OF S AND X DATA
DECODE DATA
YES
STORE X-BAND RESIDUAL
NO
is OUTPUT BUFFER FULL?
YES
NO
STORE BUFFER
RESET O.B.
STORE NEW DATA IN O.B.
YES
ALL DATA TRANSFERRED?
NO

FIG. 2 b

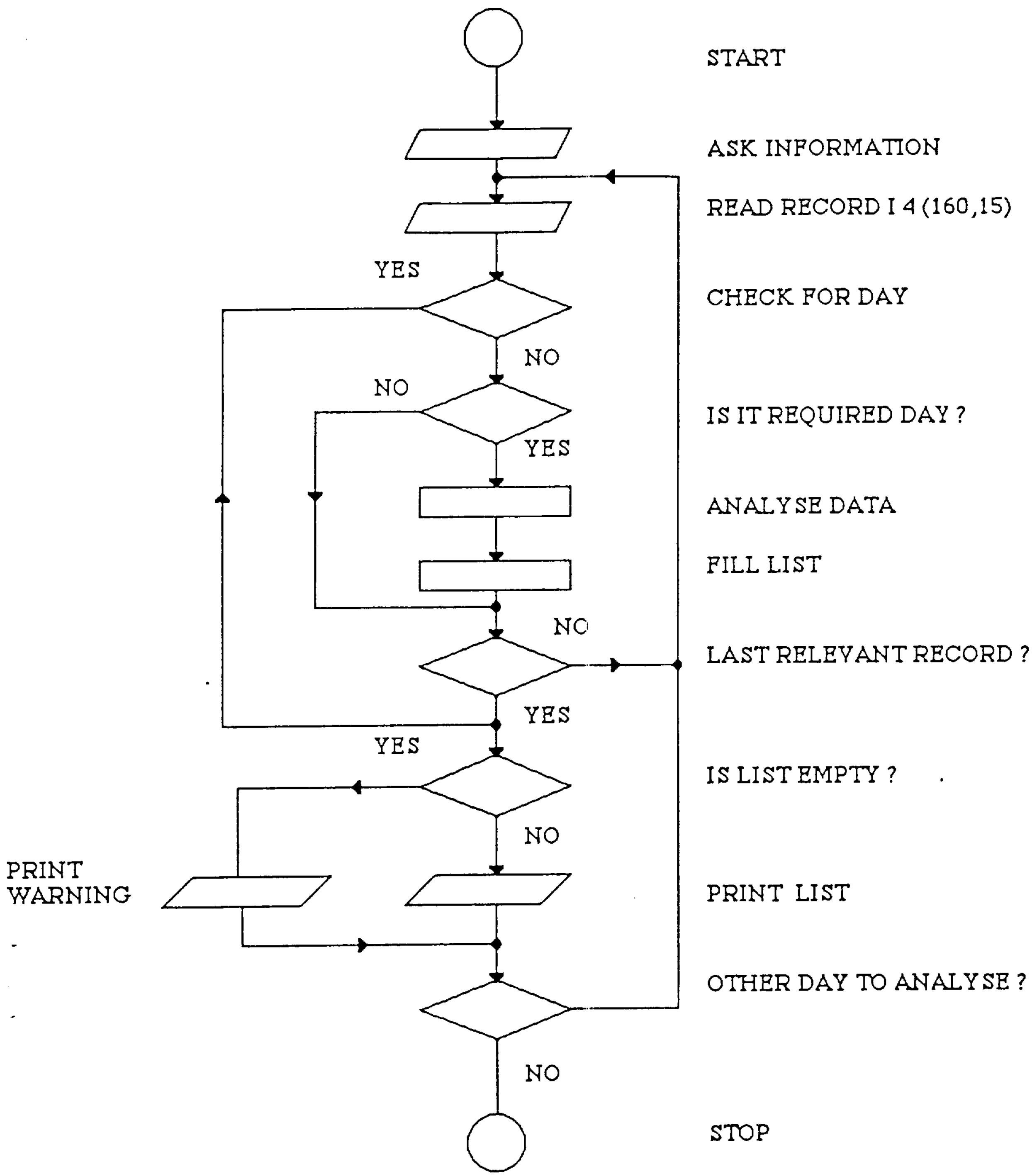
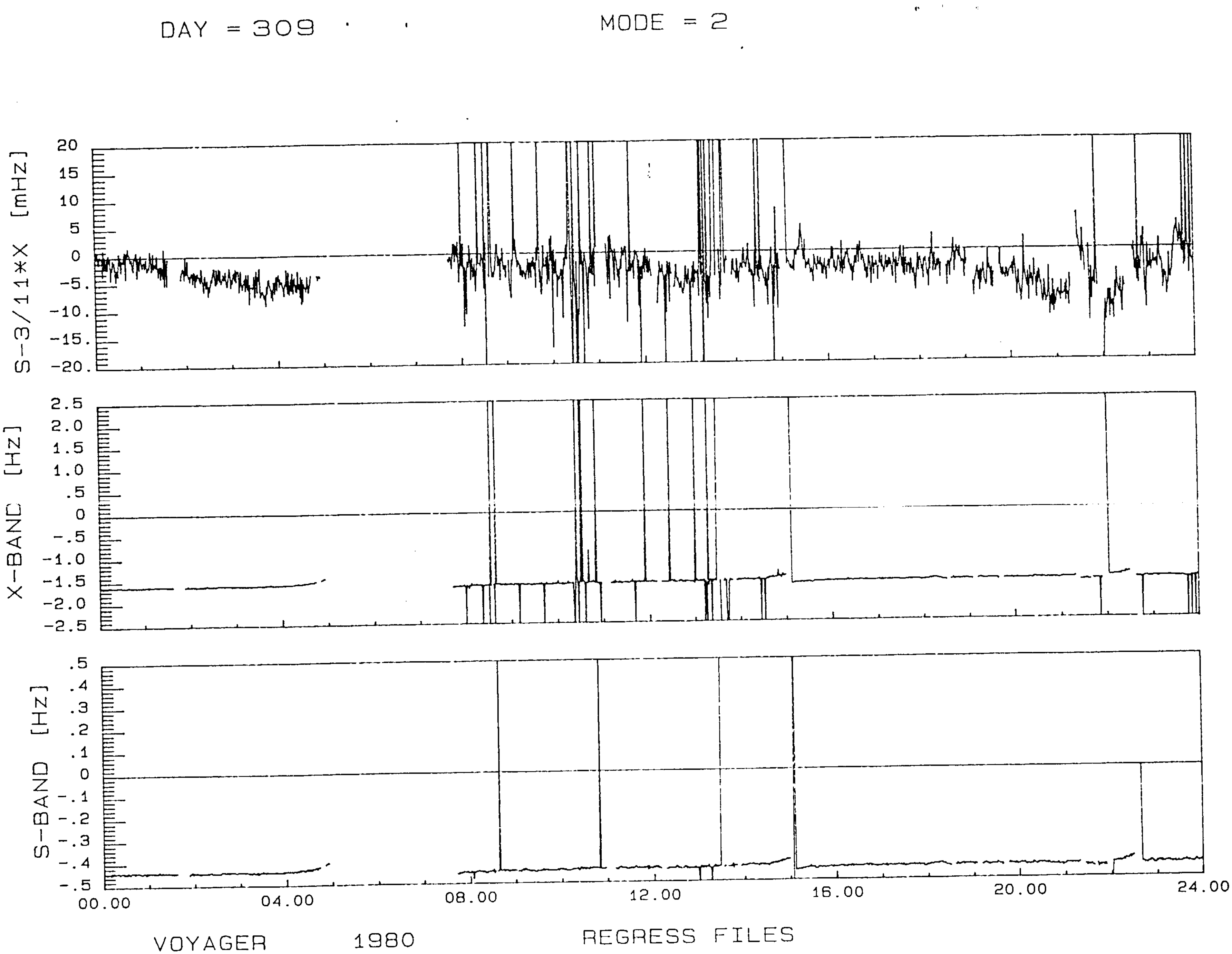

START
ASK INFORMATION
READ RECORD I 4 (160,15)
YES
CHECK FOR DAY
NO
NO
IS IT REQUIRED DAY ?
YES
ANALYSE DATA
FILL LIST
NO
LAST RELEVANT RECORD ?
YES
YES
IS LIST EMPTY ?
NO
PRINT WARNING
PRINT LIST
OTHER DAY TO ANALYSE ?
NO
STOP

FIG. 3



DAY = 309
MODE = 2
S-3/11*X [mHz]
20
15
10
5
0
-5.
-10.
-15.
-20.
X-BAND [Hz]
2.5
2.0
1.5
1.0
.5
0
-.5
-1.0
-1.5
-2.0
-2.5
S-BAND [Hz]
.5
.4
.3
.2
.1
0
-.1
-.2
-.3
-.4
-.5
00.00
04.00
08.00
12.00
16.00
20.00
24.00
VOYAGER 1980
REGRESS FILES

FIG. 4

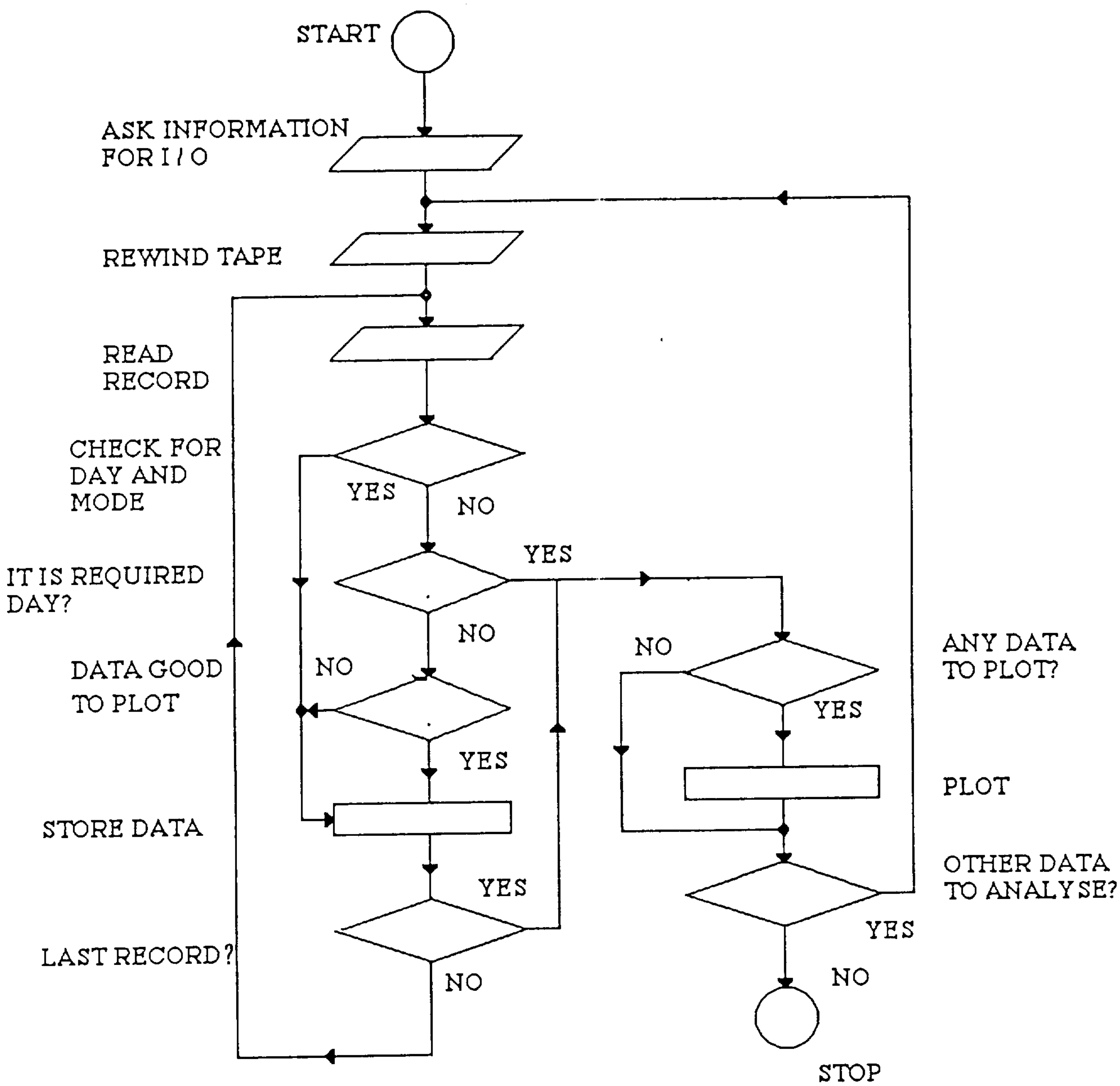

START
ASK INFORMATION
FOR I / O
REWIND TAPE
READ
RECORD
CHECK FOR
DAY AND
MODE
YES
NO
IT IS REQUIRED
DAY?
YES
NO
DATA GOOD
TO PLOT
NO
YES
STORE DATA
YES
LAST RECORD?
NO
NO
ANY DATA
TO PLOT?
YES
PLOT
OTHER DATA
TO ANALYSE?
YES
NO
STOP

FIG. 5

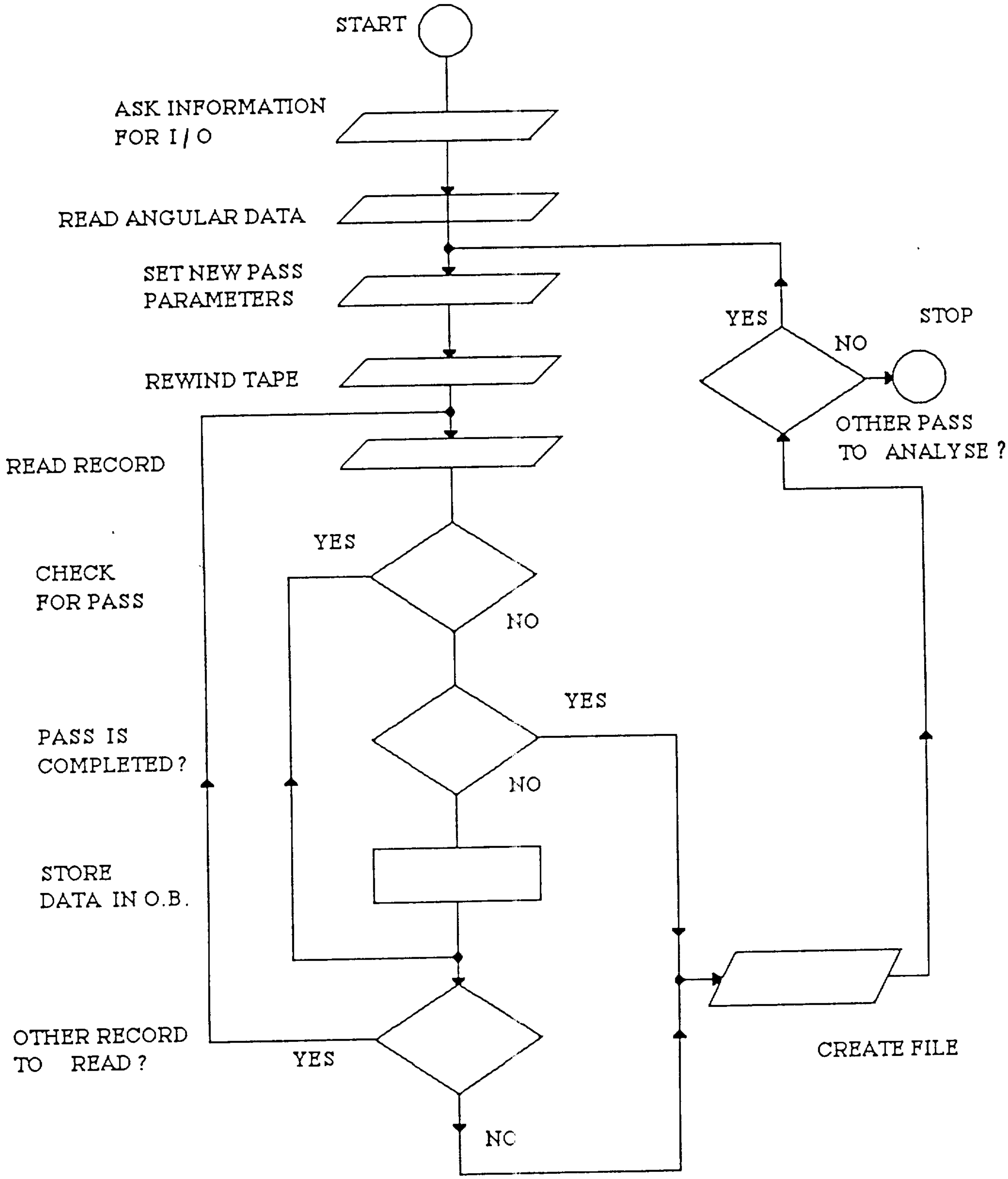

START
ASK INFORMATION FOR I/O
READ ANGULAR DATA
SET NEW PASS PARAMETERS
REWIND TAPE
READ RECORD
CHECK FOR PASS
YES
NO
PASS IS COMPLETED ?
YES
NO
STORE DATA IN O.B.
OTHER RECORD TO READ ?
YES
NO
CREATE FILE
OTHER PASS TO ANALYSE ?
YES
NO
STOP

FIG. 6