PART II

Ephemeris Reconstruction Software

by

Brian D. Cuthbertson
Departments of Astronomy & Aerospace Engineering
University of Texas
Austin, Texas 78712

1 INTRODUCTION

Like many software development efforts, the Transportable Laser Ranging System (TLRS) satellite ephemeris reconstruction software discussed in this paper is a creature of evolution. Its earliest ancestor was installed in the newly-completed TLRS Nova minicomputer in late 1979, and performed serviceably, if slowly, during the initial testing of the TLRS system. Since that time there have been several descendents, each of which has evolved to enhance the speed and/or accuracy of its predecessor. The present software satisfactorily fulfills present TLRS requirements for tracking the LAGEOS satellite, and in fact is adequate for use with other lower satellites as well. This paper summarizes the operational features of the present software, and the environment in which the software operates as part of the overall TLRS tracking system.

2 T.L.R.S. OPERATING ENVIRONMENT

The TLRS ephemeris reconstruction software can perhaps be best introduced through an explanation of its roll in the larger TLRS software tracking system. Naturally, the overall goal of the tracking system is to provide accurate real-time coordinates of a satellite (LAGEOS) during a given pass. The process may be outlined as follows: First, a LAGEOS ephemeris must be supplied to the TLRS crew. This ephemeris is presently generated at the University of Texas Department of Aerospace Engineering through the UTOPIA orbital analysis system, using full geopotential fields and detailed perturbation forces. The ephemeris is accurate for up to several months, and provides state vectors in a metric INTER RANGE VECTOR format at periodic intervals. Originally this interval was 3 hours, but integrator package accuracy improvements have allowed for an increase to 1 day for LAGEOS. Preceding a given LAGEOS pass, the TLRS crew will use the ephemeris and the ephemeris reconstruction software to generate a computer file of predicted altitudes, elevations, and ranges at 1-minute intervals. Often this will be done at the beginning of a day for all passes to be observed. During an actual pass, the TLRS tracking software will use the computer file and an interpolation algorithm to provide continuous real-time altitude, elevation, and range estimates to the tracking hardware. Hence the ephemeris reconstruction software itself is not required to provide real-time performance during actual tracking operations.

3 ACCURACY REQUIREMENTS

Prior to development of the initial 1979 software package, accuracy requirement discussions led to adoption of a roughly 100-meter total maximum pointing error. This pointing accuracy must naturally be coupled with a reasonable integration period. (Any ephemeris reconstruction package can meet any pointing accuracy requirement for a short enough integration time.) The minimum tolerable integration time period was thus set at 3 hours, since 3 hours appeared to be the shortest period desirable for successive state vectors on a LAGEOS IRV ephemeris. As mentioned previously, this 3 hour minimum period proved to be too conservative. The present reconstruction software maintains accuracy well within the 100-meter limit for periods on the order of 24 hours. Hence, the 24-hour period has been adopted as the standard interval for ephemeris state vectors.

4 FORCE MODELING

Any satellite ephemeris reconstruction software package is essentially completely defined by the force models and the integrator it uses. For the TLRS software, both of these components have evolved since the development of the initial version in 1979. The force modeling in the initial model consisted strictly of a geopotential field model using conventional terms including up to 4th degree and order. This field was evaluated using a Pines (rectangular) geopotential formulation [Pines,1973]. Though adequate for 3-hour integrations, that first force model has been considerably expanded in the present "24-hour" version. In particular, the geopotential model now uses normalized terms including up to 7th degree and order. This allows inclusion of several higher order terms of particular significance to LAGEOS. The field is evaluated using a new Pines normalized geopotential formulation developed especially for this application by Dr. Bob Schutz of the University of Texas Department of Aerospace Engineering. In addition to geopotential perturbations, the present package also models lunar and solar perturbations using an analytical model that does not require lunar and solar ephemerides. The combination of a 7 by 7 geopotential field with lunar and solar ephemerides produces the accuracies desired in the present "24-hour" ephemeris reconstruction software.

5 INTEGRATOR

The integrator used in the TLRS software was originally a Runge Kutta second order integrator. Although adequate in terms of accuracy, it was relatively slow; a 24-hour integration required on the order of 20 minutes in the first software package run on the TLRS Nova minicomputer. In the first revision of the software package, the Runge Kutta integrator was replaced with the much more sophisticated Krogh-Shampine-Gordon integrator, a 14th order multi-step integrator very similar to the one used in the U.T. Aerospace Department's UTOPIA orbital analysis system [Lundberg, 1981]. Although the new integrator is much larger in terms of source code, it is also much faster since it employs a table interpolation scheme. A 24-hour integration, for example, requires on the order of only 5 minutes of TLRS Nova minicomputer time.

6 OTHER FEATURES

In addition to force modeling and the integrator itself, there are several other features worth noting. First, the mean equatorial system has been chosen as the inertial system of integration. In the initial reconstruction software an arbitrary inertial system coincident with the initial state vector was used, but this later became awkward when evaluations of lunar and solar perturbations were added. Second, pseudo-evaluations have been incorporated into the routine calculating perturbing accelerations.

In ended as a time saver, this feature saves each time and total perturbing acceleration for possible re-use during the next integrator call, thus preventing unnecessary and time-consuming re-evaluations of the geopotential and lunar/solar perturbations at duplicate times. Third, the software takes advantage of predicted X and Y polar motion data, and earth rotation rate change data, that are provided in the U.T. Aerospace ephemeris in addition to the IRV state vector for each ephemeris time point. The X and Y polar motion data are used in the conversion of the satellite pseudo body-fixed state to true body-fixed state for the calculation of satellite azimuths, altitudes, and ranges. It might also be noted that the predicted range is corrected for refraction effects. The earth rotation rate change is applied to a constant base, which is then used in various coordinate system transformations. The fact that both polar motion and earth rotation predictions appear on U.T. Aerospace ephemerides reflects on the significant work put forth at U.T. during the past few years on extrapolating these values, work done in conjunction with the TLRS software development effort. Fourth and finally, it might be mentioned that the geopotential field used in the system at present is the GEM-10 normalized field. One flexible feature of the software is that the geopotential field coefficients, as well as other basic geophysical constants, are read from a data file, thus allowing an update of geopotential field or constants without re-compilation of code.

7 NOTES ON SOURCE CODE

The development of the TLRS ephemeris reconstruction software package was not done directly on the TLRS Nova minicomputer on which it was installed, but rather on a similar sized PDP 11/60 minicomputer in the U.T. Dept. of Aerospace Engineering. The Aerospace system offered the benefits of ease-of-access plus a direct link to the powerful U.T. Cyber computers, on which the UTOPIA orbital analysis system was used in generating satellite ephemerides needed for the development work. The FORTRAN source code available with this paper is taken from the PDP 11/60 development system, and reflects the "test bed" orientation of that system. Options allow either interactive or file input of initial, intermediate, and final comparison state vectors, and allow output of radial, transverse, and normal residuals, or of altitude, elevation, and ranges. Program size permits operation without overlays with 32000 16-bit words of memory.

8 CONCLUDING REMARKS

The TLRS ephemeris reconstruction software has proved itself an effective tool during TLRS field operations over the past two years, particularly when used for LAGEOS satellite operations, given LAGEOS ephemeris information generated by the University of Texas Department of Aerospace Engineering. Its performance is satisfactory enough that no modifications to the existing software are anticipated in the near future. The package is also being installed in the University of Texas Astronomy Department's Mobile Laser Ranging System (MLRS), now in place at McDonald Observatory, for satellite and lunar laser ranging work.

9 REFERENCES

Lundberg, J.B., "Multistep Integration Formulas for the Numerical Integration of the Satellite Problem," Inst. for Advanced Study in Orbital Mechanics, The University of Texas at Austin, TR81-1, 1981

Pines, S., "Uniform Representation of the Gravitational Potential and its Derivatives," AIAA J. 11 (11), 1508, 1511, 1973

Spencer, J.L., "Pines Nonsingular Gravitational Potential: Derivation, Description, and Implementation," McDonnell Douglas Technical Services Company, Inc., Report MDC W0013, NASA contract NAS 9-13970, 1976.

10 APPENDIX A

```
********************************************************************************
TAPE FORMAT:-
9-TRACK, 1600 BPI
LOGICAL RECORD SIZE = 90 ASCII BYTES/RECORD
BLOCK SIZE = 100 RECORDS/BLOCK
 (THE LAST BLOCK OF EACH FILE MAY BE SHORT)
********************************************************************************
```

TAPE CONTENTS:

TAPE CONTAINS FILES OF FORTRAN SOURCE CODE WHICH CAN GENERATE TWO PROGRAMS, IRVINT AND RESGEN. THE FORTRAN SOURCE CODE WAS DESIGNED TO RUN ON A DIGITAL EQUIPMENT CORPORATION PDP 11/60 MINICOMPUTER UNDER THE RSX11-M OPERATING SYSTEM. FOR ADDITIONAL INFORMATION ON EITHER PROGRAM OR ITS OPERATION, CONTACT DR. BOB SCHUTZ, RICHARD EANES, OR BRIAN CUTHBERTSON, AT THE UNIVERSITY OF TEXAS DEPARTMENT OF AEROSPACE ENGINEERING, UNIVERSITY OF TEXAS AT AUSTIN, AUSTIN, TEXAS 78712, U.S.A.

PROGRAM IRVINT:
(IRVINT IS AN ACRONYM FOR INTER-RANGE VECTOR INTEGRATION PROGRAM)
THIS PROGRAM CONTAINS AN INTERACTIVE VERSION OF THE SATELLITE INTEGRATION PACKAGE DEVELOPED FOR FIELD MINICOMPUTER USE IN LASER-RANGING OPERATIONS USING THE TLRS (TRANSPORTABLE LASER RANGING SYSTEM) AND MLRS (MOBILE LASER RANGING SYSTEM) OPERATED BY THE UNIVERSITY OF TEXAS MCDONALD OBSERVATORY. ONCE INTEGRATION ALGORITHMS ARE VERIFIED IN THIS PACKAGE, THE RELEVANT ROUTINES ARE TRANSFERRED TO THE TLRS AND MLRS NOVA MINICOMPUTERS FOR INSTALLATION IN THE FIELD SOFTWARE PACKAGES.

PROGRAM RESGEN:
(RESGEN IS AN ACRONYM FOR RESIDUAL-GENERATION PROGRAM)
THIS PROGRAM IS A MODIFIED VERSION OF IRVINT, USED AT THE U.T. DEPT. OF AEROSPACE ENGINEERING TO DO PRELIMINARY COMPARISONS OF OBSERVED SATELLITE RANGE VALUES (FROM INCOMING QUICK-LOOK RANGING DATA) TO CALCULATED (INTEGRATED) RANGE VALUES. RESIDUAL DIFFERENCES BETWEEN OBSERVED AND CALCULATED RANGES ARE PRODUCED.

THESE DRIVER PROGRAMS ARE INCLUDED TO DEMONSTRATE THE USE OF THE VARIOUS SUBROUTINES SO THAT MORE VERSATILE PACKAGES TAILORED TO EACH USERS SPECIFIC NEEDS CAN BE CONSTRUCTED.

AS THIS IS OUR FIRST ATTEMPT TO EXPORT THE TLRS/MLRS PREDICTION SOFTWARE, ANY FEEDBACK CONCERNING PROBLEMS YOU ENCOUNTER IN ADAPTING THE CODE TO USE ON OTHER SYSTEMS WILL BE APPRECIATED AND CAN BE TRANSMITTED TO THE ABOVE MENTIONED ADDRESS.

```
********************************************************************************
```

SOURCE CODE REQUIRED FOR PROGRAM COMPILATIONS:

IRVINT REQUIRES THE FOLLOWING TAPE FILES OF SOURCE CODE FOR COMPILATION:
IRVINT, KSG, GEOPINN, ROTLIB, AZELVR, DERIV, ASKIRV, SUNMON, HOLIB

RESGEN REQUIRES THE FOLLOWING TAPE FILES OF SOURCE CODE FOR COMPILATION:
RESGEN, KSG, GEOPINN, ROTLIB, AZELVR, DERIV, ASKIRV, SUNMON, HOLIB

```
******************************************************************************

COMPLETE LIST OF FILES ON THIS TAPE:

FILE NO.:  FILE NAME:     DESCRIPTION:

    1      INTRO          TAPE CONTENT INFORMATION (THIS FILE).

    2      IRVINT         PROGRAM IRVINT; ROUTINES SITES,ASKAZ.

    3      RESGEN         PROGRAM RESGEN; ROUTINES SITES, READOB, FNDIRV.

    4      AZELVR         ROUTINE AZELVR.

    5      ASKIRV         ROUTINES ASKIRV, GETIRV, RAOGU.

    6      GEOPINN        ROUTINES GEOPINN, GEOSET.

    7      DERIV          ROUTINE DERIV.

    8      KSG            ROUTINES INTEGR, KSG, KSGDR, SGSTRT, SGSTEP, SGNTRP,
                          KSGCO, BCKDIF.

    9      SUNMON         ROUTINES SUNMON, ECLEQ.

   10      HOLIB          ROUTINES EQN, KEP, PMPTRB.

   11      ROTLIB         ROUTINES ROTATE, RTNROT, MA3331.

   12      GEM10N         GEOPHYSICAL DATA FILE; INPUT FOR IRVINT AND RESGEN.

   13      SITE           LASER-RANGING SITE DATA FILE; INPUT FOR
                          IRVINT AND RESGEN.

   14      IRV            SAMPLE METRIC INTER-RANGE VECTOR DATA FILE;
                          INPUT FOR IRVINT AND RESGEN.

   15      OBS            SAMPLE QUICK-LOOK RANGE DATA FILE (SEASAT
                          DECIMAL FORMAT); ONE PASS FROM HALEAKALA (7210 )
                          AND ONE PASS FROM WETTZELL (7834) ON 08 JULY 82;
                          INPUT OBSERVATION FILE FOR RESGEN.

   16      RES            SAMPLE RESGEN OUTPUT RESIDUAL FILE.
                          RESIDUAL FILES FROM 3 EXECUTIONS OF RESGEN ARE
                          APPENDED TOGETHER WITH COMMENTS.
******************************************************************************

END OF FILE INTRO.

******************************************************************************
```

Real-Time Data and Quick-Look for the CERGA LLR System

by

J. Kovalevsky and S. Lengelle
CERGA
Grasse, France

1 Object of the Programs

Two programs are used in CERGA in order to control, in real-time, the data acquisition of the lunar laser and to perform an almost real-time quick look treatment of the data obtained, so that the operator may know its approximate quality.

1.1 Program "VISTIR"

The program "VISTIR" (visulization des tirs = firing visualisation) performs the following tasks:

- Acquisition of the time of laser firings as registered by the event-timer.
- Control of the electronic gate after each firing so that it is opened around the return time as forecast by the ephemerides.
- Acquisition of the events registered by the event-timer and computation of the (o - c) by comparison of the ephemerides.
- Display of the (o - c) on the screen on a histogram divided in 50 channels corresponding each to 1/50 of the gate width.

At the end of a series of laser shots, the observer may:

- Request a print-out of the number of non-isolated (o - c) in 5 nanoseconds channels.
- Reduce at will the channel width of the histogram and recenter around any indicated old channel, producing a new histogram.
- Print the time, the measured delay and the corresponding (o - c) for all events in the channel of the maximum of the last histogram and in the four surrounding channels.

If this first quick-look shows for for one or several series of shots on the same retroreflector that there are probable returns above the noise, it is possible to analyze more accurately together these series of events using the next program.

1.2 Program "TIRESU"

The program "TIRESU" (tirs-resultats = firing results) performs the following task:

- Computation of the (o - c) of all events by comparison with ephemerides,

- Display of rejected events (¦(o - c)¦ > 10 microseconds),
- Print-out of the number of all (o - c) in every 5 nanosecond channel,
- Display on the screen of the histogram of (o - c) (see VISTIR),
- Reduction of the channel width at request of the operator and display of a new histogram centered at any old channel indicated by the operator.

When the operator judges that no new histogram is needed, and on request of the operator, the program:

- Prints the time, the measured delay and the corresponding (o - c) for all events in the channel of the maximum of the last histogram and in the four surrounding channels.
- Computes by least squares method a regression straight line representing (o - c) as function of time. This is a good representation of the normal trend of (o - c) in function of time due to (UT1-UTC) difference if the duration of the observations does not exceed 20 minutes. The coefficients of the straight line are given and the residuals of the selected events with respect to this line are printed. The standard deviation of these residuals is also printed.
- The operator can examine these residuals and may order the program to ignore some of the events, for instance if their residuals are too large. Then, the whole procedure is started again.
- When the operator does not request more rejection, he enters in the computer the parameters of the observation (temperature, pressure,...) that are printed.
- Finally, on request of the operator, all events (time and delay) are printed.

2 Specifications of the Programs

Both programs are written in FORTRAN IV for the Data General Eclipse 5200 system composed of:

- CPU with 32 K byte
- Disk unit
- Alphanumeric Tektronix display
- 16 bits interface (digital input/output)

- Teletype

- RS232C interface (ALM 8).

The software support is the real-time disk operator system (RDOS), with the multitask package for FORTRAN.

However, one subroutine, (WPORTE) controlling the data flow to the electronic gate is written in assembly language.

Figure 1 shows the connections used in real-time data acquisition "VISTIR".

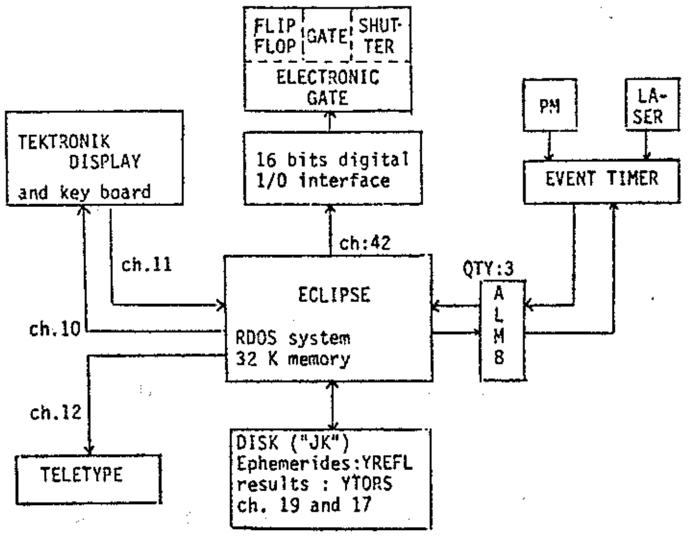


Figure 1: Connections with the "Eclipse" for VISTIR program.

They have the following characteristics:

- Connection with the event-timer through ALM 8: channel QTY: 3.

- Connection with the electronic gate through the 16 bit input/output interface: channel 42.

Other connections are common for both programs "VISTIR" and "TIRESU".

- The Tektronix display: channel WRITE = 10 and channel READ = 11.

- The teletype: channel WRITE = 12.

The program and files are in the directory "JK". It is also assumed that the ephemerides are prepared and recorded in an ephemeris file: "JK : YREFL", channel 19.

For the program "VISTIR" the observations are recorded in an observation file: "JK : YTORS", channel 17. All the observation files that are to be used in "TIRESU" will have to be previously appended in a file "JK : YTIRS".

Let us now study successively each of the two programs.

3 Real Time Data Acquisition and Visualization Program "VISTIR"

3.1 Structure of the program

The program has a multitask organization that permits to activate various parts of the computation in a given order defined by their priorities and the occurrence of exterior events.

The program starts with the initialization task that is "VISTIR" proper.

This task is the initialization of the overall program. It includes:

- Reading of the ephemerides into commons,
- Preparation of the gate treatment (request of its half-with),
- Drawing the histogram axes. This is done using the subroutine "WHIST" initialized by "HIS" reset to zero (subroutine WHIST is used to draw a histogram of 50 numbers written in the common "HIS").

At the end of the task, the three other task are activated. These are:

- ANALY : 1st priority
- SOLO 2 : 2nd priority
- SOLO 1 : 3rd priority

Before the initialization is killed, ANALY is held, and control given to SOLO 2.

The aim of SOLO 2 is to stop the treatment when the operator informs the computer that the firings are over. SOLO 2 is held until a "1" is typed on the Tektronix keyboard. Until this happens, the control is shared by the other two tasks, as shown on figure 2.

3.1.1 SOLO 1

The first task, SOLO 1, reads one by one the data present in the event-timer and recognizes whether it is a firing or a return event. The structure of the event-timer record is the following:

- 1st digit: tens of hours if it is an event; tens of hours plus 8 if it is a firing,

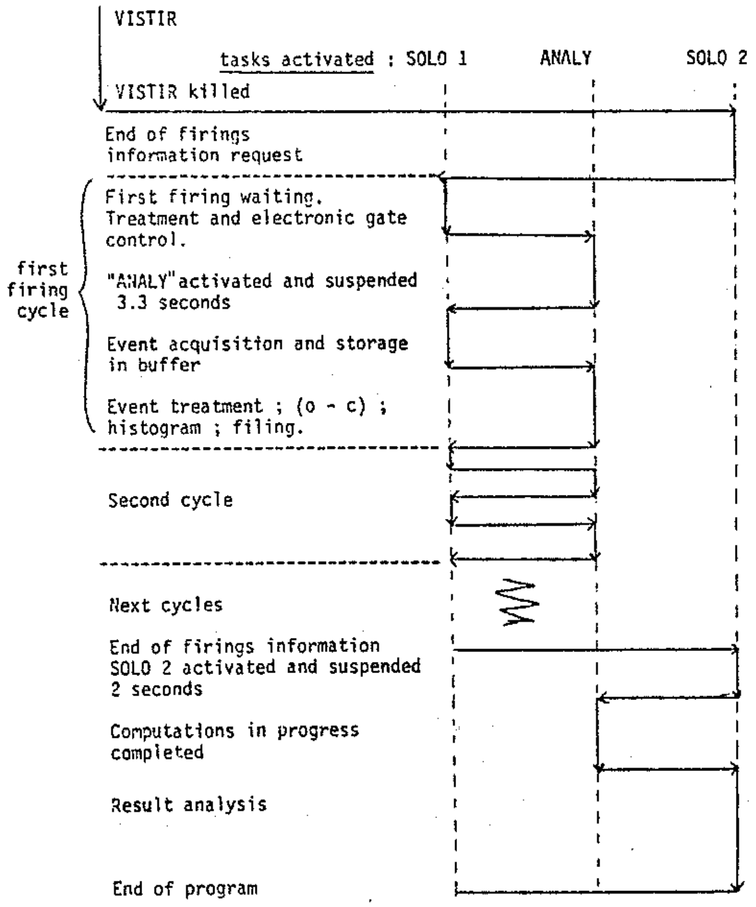

VISTIR
tasks activated : SOLO 1
ANALY
SOLO 2
VISTIR killed
End of firings information request
first firing cycle
First firing waiting. Treatment and electronic gate control.
"ANALY"activated and suspended 3.3 seconds
Event acquisition and storage in buffer
Event treatment ; (o - c) ; histogram ; filing.
Second cycle
Next cycles
End of firings information SOLO 2 activated and suspended 2 seconds
Computations in progress completed
Result analysis
End of program

Figure 2: "VISTIR" task sequence.

- next digits: hours (1), minutes (2), seconds (2), hundreds of picoseconds (10).

If the first digit is "8", "9", or "A", the event is recognized as a firing. The expected delay is computed by interpolation of the ephemerides. The time is stored in the buffer. The delay "D" is also sent to the electronic gate using the subroutine "WPORTE". This subroutine is written in assembly language. It sends to the gate the eight BCD digits of (D - 2) seconds expressed in 10^{-8}s.

Finally, the control is given to ANALY that is, then suspended during 3.3 seconds, so as to leave time to SOLO 1 to treat the return events. If the first digit is "0", "1" or "2", the event is recognized as a return. It is stored in the buffer.

3.1.2 ANALY

The second task, ANALY, computes the (o - c) for each event present in the buffer and increments by 1 the corresponding channel of the histogram. The observations are filed in "JK : YTORS". If the (o - c) differs by less than 10 nanoseconds of a previous one or if it is at least the fourth in a channel, the bell rings.

When all the events filed in the buffer are treated, the control is returned to SOLO 1. Warning is given to the operator when the observation files are 75% complete (150 events). If they are full, the computer turns SOLO 1 off. No more events can be treated and control is given to SOLO 2.

3.1.3 SOLO 2

The third task, SOLO 2, is called when the operator punches a "1" on the keyboard to indicate the end of the series of observations or when the observation files are full. Another two seconds are given to ANALY to finish the computation in progress before these tasks are killed. If there are less than four events, the program is not processed. If there are at least four events, SOLO 2 analyzes the (o - c) and displays the results as follows:

- A print-out on the teletype of the distribution of possible returns. The (o - c) range between $\pm$ the half width of the electronic gate. This interval is divided in 100 ns sections. Each section is divided in 20 channels of five nanosecond width. The numbers of (o - c) in all channels of a section are printed whenever:

    1. there is one (o - c) in the first or last two channels of the section.

    2. there are two (o - c) or more in at least one channel of the section.

    3. there are (o - c) in two consecutive channels or separated by one channel.

- The operator may request a change in the range of the histogram. The minimum and the maximum (o - c) of this range, the number of rejected (but not suppressed) events falling outside this new range, the new channel width of the histogram are displayed and, then, the new histogram itself (using subroutine "WHIST".) This request can be iterated as many times as desired. The channel width is usually diminished to observe more details in the distribution of the residuals, but it may also be widened without inconvenience.

- If the operator judges that at a certain stage there is a significant maximum in the histogram, the teletype prints out the events corresponding to the channel of the maximum in the last histogram displayed as well as to the four surrounding channels.

3.2 Operating Procedure

The assembly of the program is done with the following statement: "RLDR JK : <VISTIR WHIST SOLO1 WPORTE ANALY SOLO2> FMT.LB 10/C 5/K FORT.LB".

The program must be called when all the peripherals and hardware connections are ready, and the ephemeris file placed in "JK : YREFL". The procedure is then straight forward, since the computer displays in full words (in French) the actions that the operator should take. These are, in sequence, the following:

1. NUMERO DE SERIE ? FORMAT.., PUIS RETURN: request of serial number of the observation series, so as to recognize it in the appended file of observations.

2. QUELLE DEMI-LARGEUR DE PORTE ? FORMAT .... DIZAINES DE NANOSECONDE: request of the half width of the electronic gate in 10^{-8} second.

3. APPUYER SUR "RESET" PUIS SUR O ET "RETURN": operator is requested to clear the display unit. Minimum and maximum (o - c) are displayed as well as the axes of the histogram.

4. A LA FIN DES TIRS, TAPER 1 ET "RETURN": the operator is now authorized to start the laser operation. He has nothing to do with the keyboard until he judges that the series is over. He then should punch "1-RETURN" to end the data acquisition. During the firings, the histogram of (o - c) is built and the bell will ring to warn that there are at least four events in a channel or that there are two events within 10 nanoseconds.

When the analysis starts, a new conversation between the operator and the console takes place:

1. RESULTATS SAUVEGARDES DANS JK : YTORS: the computer informs the operator that the observations are saved and that the analysis will take place (if not, the program stops). While the teletype prints the distribution of possible returns, the console displays rejected events outside the histogram limits and the characteristics of the next histogram:

   - (o - c) minimum and maximum in the histogram,

   - width of the histogram,

   - number of suppressed events,

   - width of a channel of the histogram.

2. APPUYER SUR "RESET" PUIS SUR O ET "RETURN": the operator is requested to clear the display unit for the next histogram. Minimum and maximum (o - c) and the channel width are displayed, and then, the histogram itself.

3. FAUT-IL UN HISTOGRAMME PLUS FIN?/OUI: FAIRE 1; NON: FAIRE O: the operator is requested to state whether he wants a more detailed histogram. If the answer is "YES":

4. NUMERO DU PAS DU MAXIMUM?: the operator must give the two digits of the channel number around which the new histogram should be displayed.

5. VALEUR DU PAS ? FORMAT ..., L'ANCIEN ETAIT DE: XXX NANOSECONDS: the operator is requested to give the channel width (in nanoseconds) he wishes for the new histogram. Then the new minimum and maximum values for (o - c) as well as the number of rejected events are displayed. The operation is resumed at phase 2).

6. If the answer to the question 3) is "NO", Y-A-T-IL UN RESULTAT PROBABLE ? OUI: FAIRE 1; NON: FAIRE 0: the operator should answer that there is a probable result if there is a well defined maximum of the histogram and if all probable results are within two channels of this maximum (if not, one should have chosen another channel width). If the answer is "YES", these observations are printed and the program stops. If it is "NO", the program stops.

3.3 Test Data

It is not possible to provide test data for "VISTIR", since it is a real-time program and requires the presence of the hardware (laser, event-timer, electronic gate). However, most of the functions of SOLO 2 (the only task that does not depend on the hardware) are present in TIRESU, for which we give test data.

4 Quick-look Treatment Program "TIRESU"

4.1 Structure of the Program

The program has a simple almost linearly constructed flow-chart. Only one subroutine is called: "WHIST" for drawing the histograms (as in "VISTIR").

Figure 3 that gives the flow-chart of the program, the main successive functions of the program being the following:

- Reading the ephemerides and the observation files and computing the (o - c) for each observation.

- Suppression and display of all observations that are outside the range $\pm$ 10 microseconds.

- Print out on the teletype of the distribution of all (o - c). All sections of 100 ns in the interval of (o - c) are divided in 20 channels of five nanosecond width. The numbers of (o - c) in all channels of a section are printed if there is at least one (o - c) in the corresponding interval.

- Display of the histogram of all the nonsuppressed (o - c).

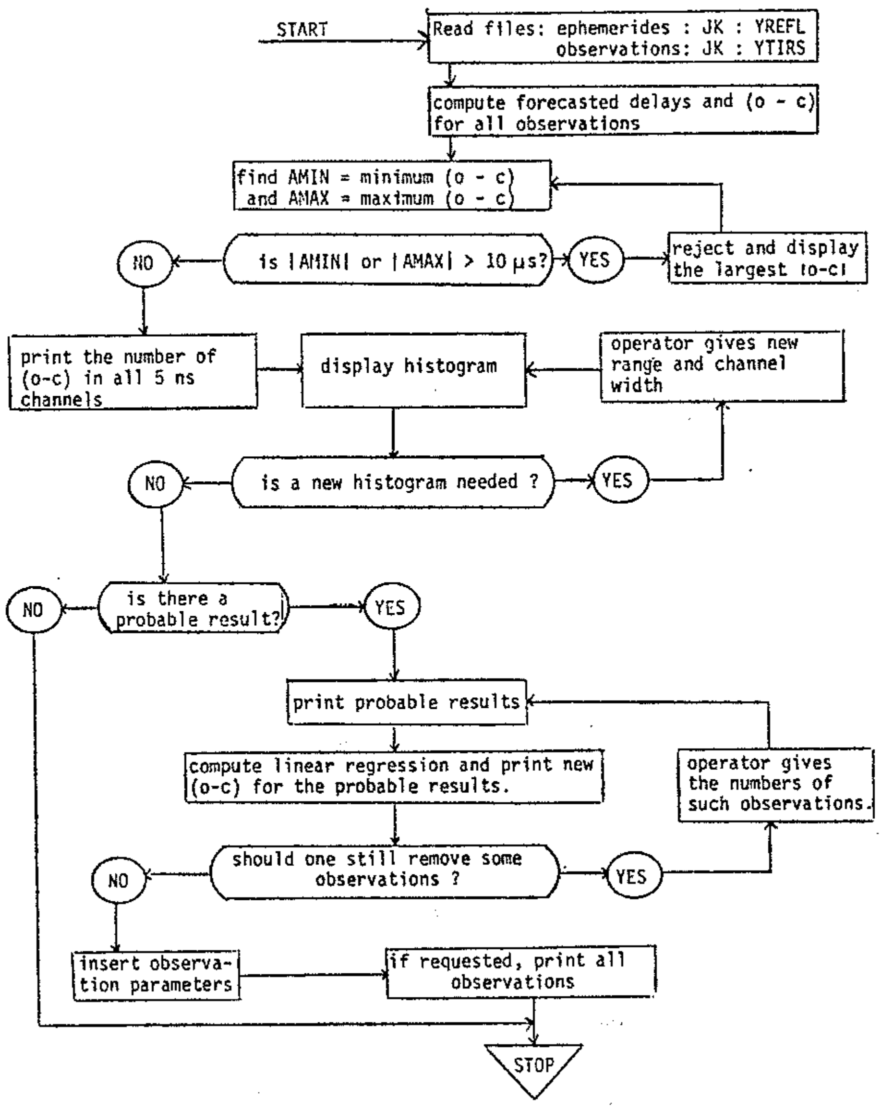

START
Read files: ephemerides : JK : YREFL
observations: JK : YTIRS
compute forecasted delays and (o - c) for all observations
find AMIN = minimum (o - c) and AMAX = maximum (o - c)
is |AMIN| or |AMAX| > 10 μs?
YES
reject and display the largest |o-c|
NO
print the number of (o-c) in all 5 ns channels
display histogram
operator gives new range and channel width
is a new histogram needed ?
YES
NO
is there a probable result?
YES
NO
print probable results
compute linear regression and print new (o-c) for the probable results.
operator gives the numbers of such observations.
should one still remove some observations ?
YES
NO
insert observation parameters
if requested, print all observations
STOP

Figure 3: "TIRESU" flow-chart.

- The operator may request a change in the range of the histogram. The minimum and maximum (o - c) of the range and the number of rejected (but not suppressed) events falling outside the new range, the new channel width and finally, the new histogram itself are displayed.

This procedure can be iterated as many times as desired. The channel width is usually diminished to observe more details of the distribution of the residuals. It may also be widened, the "rejected" observations are then recovered.

- It the operator judges that, at a certain stage, there is a significant maximum in the histogram, the teletype prints out the events corresponding to the channel of the maximum in the last histogram displayed as well as to the four surrounding channels.

- A linear least squares fit of the (o - c) as function of the observation times is computed in the form {(o - c) = Y + X (T - TMM)} where T is the observation time and TMM is the mean observation time. X, Y, and TMM are printed, then the (o - c), the new residuals with respect to this regression straight line and their standard deviation.

- The operator may reject some of these observations if he judges that their residuals are not consistent with the others. Then the procedure starts again with the print out of the remaining events.

- When this iterative process is stopped by the operator, he must give the computer the observation parameters (crater used for tracking, pressure, temperature, calibration and mean noise). These are printed.

- The operator may also request a print out of all the observations.

4.2 Operating Procedures

The assembly of the program is done with the following statement: RLDR JK : <TIRESU WHIST> FORT.LB .

The ephemeris file "JK : YREFL" and the observation file "JK : YTIRS" must be prepared beforehand. Then, the procedure is dictated by the conversation with the display console. The actions to be taken by the operator start when the characteristics of the first histogram (minimum and maximum non-suppressed (o - c), extension and channel width of the histogram, number of suppressed events) are displayed and the teletype has started to print the number of (o - c) channel by channel.

1. APPUYER SUR "RESET" PUIS SUR O ET "RETURN": the operator is requested to clear the display unit for the histogram. Minimum and maximum (o - c) and the channel width are displayed and, then, the histogram itself.

2. FAUT-IL UN HISTOGRAMME PLUS FIN? OUI: FAIRE 1; NON: FAIRE 0: the operator is requested to state whether he wants a more detailed histogram.

3. If the answer is "YES"---> NUMERO DU PAS DU MAXIMUM?: the operator must give the two digits of the channel number around which the new histogram should be displayed.

4. VALEUR DU PAS ? FORMAT: ..., L'ANCIEN ETAIT DE: XXX NANOSECONDES: the operator is requested to give the channel width (in nanoseconds) he wishes for the new histogram. Then, the new minimum and maximum values for (o - c) as well as the number of rejected events are displayed. The operation is resumed at phase 1).

5. If the answer to the question 2) is "NO"---> Y-A-T-IL UN RESULTAT PROBABLE ? OUI: FAIRE 1; NON: FAIRE 0: If the answer is "NO", the program comes to its end. If it is "YES", all the observations corresponding to the maximum of the histogram and within two channels of this maximum are considered as probable. It is therefore necessary to optimize the last histogram range. After the list of events, the residuals with respect to the linear regression and the standard deviation are printed by the teletype, the operator must examine them and decide whether some of them appear to be abnormally large. These may be rejected in a new round.

6. FAUT-IL SUPPRIMER UN POINT? OUI: FAIRE 1; NON: FAIRE 0:

7. If the answer is "YES" ---> QUEL NO. ? FORMAT XXX: the operator gives the number of the observation to be rejected. The computer comes back to the question 6) as many times as it is necessary until the answer is "NO". Then it starts to print the remaining observations and computes the new linear regression. After this, a new iteration with question 6) is started.

8. If the answer to the question 6) is "NO" so that there is no rejection, the operator is asked a number of questions, the answers of which are necessary for the scientific use of the data.

9. NO ET NOM DU CRATERE SUIVI: XX AAAAAAAA. Number and name of the crater used for the tracking.

10. PRESSION : XXXX. Atmospheric pressure in millimeters of mercury.

11. TEMPERATURE : XXX. Temperature in degrees Celsius

12. CALIBRATION : XXX. Calibration in nanoseconds.

13. COMPTAGE DE BRUIT/SECONDE EN MILLIER D'EVT/SEC : XXX. Noise in kilohertz.

14. FAUT-IL IMPRIMER LES EVENEMENTS? OUI: FAIRE 1; NON :FAIRE 0. If the answer is "YES", all the observations are printed. If "NO" the program ends.

4.3 Test data

The following test data correspond to an actual observation that was made on July 7, 1981 by the CERGA lunar laser.

1. Ephemerides: The ephemeris data to be filed in "JK : YREFL" are given in table 1.

2. Observations: The observation data to be filed in "JK : YTIRS" are given in table 2.

```
 3
 7  7 31
13 30  0
 0.2444792500000D  7
13 30  0.25993231092759D  1  0.24447932708330D  7  0.13191560D  3  0.34799000D  1
19  0  0.26019693960665D  1  0.24447932915700D  7  0.13208330D  3  0.33349000D  1
19 30  0.26051117913340D  1  0.24447933125000D  7  0.13225960D  3  0.32396000D  1
20  0  0.26085191595641D  1  0.24447933333300D  7  0.13244070D  3  0.31940000D  1
20 30  0.26124442959939D  1  0.24447933541700D  7  0.13253120D  3  0.30933000D  1
21  0  0.26165353655435D  1  0.24447933750000D  7  0.13283090D  3  0.30024000D  1
21 30  0.26203353631440D  1  0.24447933958300D  7  0.13304030D  3  0.29065000D  1
22  0  0.26252379912710D  1  0.24447934166700D  7  0.13325010D  3  0.28103000D  1
22 30  0.26293287543330D  1  0.24447934375000D  7  0.13349030D  3  0.27152000D  1
99  0  0.00000000000000D  0  0.24447935000000D  7  0.39999900D  3  0.30395012D-20
```

Table 1: Apollo 15 - 7 July 1981; File: "JK:YREFL"

```
.710418027361513D  5 .710444094443219D  5
.710418027361513D  5 .710444094447877D  5
.710663427100116D  5 .710689494675104D  5
.710724696681012D  5 .710750764381634D  5
.710786273755526D  5 .710812341576017D  5
.710970431653864D  5 .710996429803854D  5
.711215570180306D  5 .711241638793969D  5
.711215570180306D  5 .711241638850397D  5
.711276942359855D  5 .711303011134857D  5
.711276942359855D  5 .711303011137195D  5
.711338323174038D  5 .711364392071134D  5
.711461072770986D  5 .711487141961091D  5
.711522259919125D  5 .711548329175773D  5
.711645021103463D  5 .711671090596896D  5
.711767788546231D  5 .711793858283247D  5
.711767788546231D  5 .711793858285609D  5
.711828981145443D  5 .711855050950139D  5
.711890566598214D  5 .711916636575676D  5
.712013149731651D  5 .712039219910543D  5
.712136324498718D  5 .712162394974926D  5
.712136324498718D  5 .712162394957956D  5
.712197711393090D  5 .712223781917431D  5
.712259096258886D  5 .712285166889164D  5
.712320488971583D  5 .712346559792107D  5
.712320488971583D  5 .712346559736236D  5
.712504806496095D  5 .712530877662363D  5
.712504806496095D  5 .712530877708274D  5
.712565999587552D  5 .712592070890611D  5
.712565999587552D  5 .712592070842694D  5
.712627378376316D  5 .712653449747822D  5
.712627378376316D  5 .712653449772636D  5
.712688556869579D  5 .712714628413667D  5
.712688556869579D  5 .712714628405390D  5
.712872674636206D  5 .712898746542712D  5
.712872674636206D  5 .712898746547207D  5
.712872674636206D  5 .712898746572922D  5
.713056410283163D  5 .713082482551665D  5
.713178981103605D  5 .713205053613771D  5
.713362933074042D  5 .713389005947275D  5
.713485639028205D  5 .713511762121433D  5
.714037541951761D  5 .714063616108843D  5
.714344105483402D  5 .714370130298840D  5
.714650404302753D  5 .714676479726539D  5
.715385906153006D  5 .715411983041407D  5
.715385906153006D  5 .715411982293862D  5
.715447106269697D  5 .715473183286716D  5
.715630895439871D  5 .715656972791056D  5
.715692271832714D  5 .715718349348653D  5
.715692271832714D  5 .715718349332945D  5
.715876018480821D  5 .715902096325473D  5
.999900000000000D  6 .999900000000000D  6
```

Table 2: Observation data - 7 July 1981 (File: "JK:YTIRS"

3. Print-out of (o - c) channels: The matrix giving the number of events per 5 nanosecond channel as printed by the teletype is given in table 3. Sections with no event are not printed.

4. Histogram: We give, figures 4 to 7, the four histograms that are obtained if the following answers are successively given in the

```
CANAL, 0-C NBRE D' EVENEMENTS PAR 5  NANO

  1  -5991 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 14  -4691 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 15  -4591 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 18  -4291 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0
 20  -4091 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 22  -3891 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 23  -3791 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
 26  -3491 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 32  -2891 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 41  -1991 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 43  -1791 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 44  -1691 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 48  -1291 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
 53   -791 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 54   -691 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 60    -91 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 62    109 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 64    309 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 67    609 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 68    709 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 70    909 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 4 6 2 2
 71   1009 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 73   1209 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 75   1409 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 77   1609 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 83   2209 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 86   2509 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
 87   2609 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
100   3909 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
101   4009 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Table 3: Number of events in each 5 nanosecond channel.

histogram conversion.

a. Fig. 4: histogram appearing in the first place.

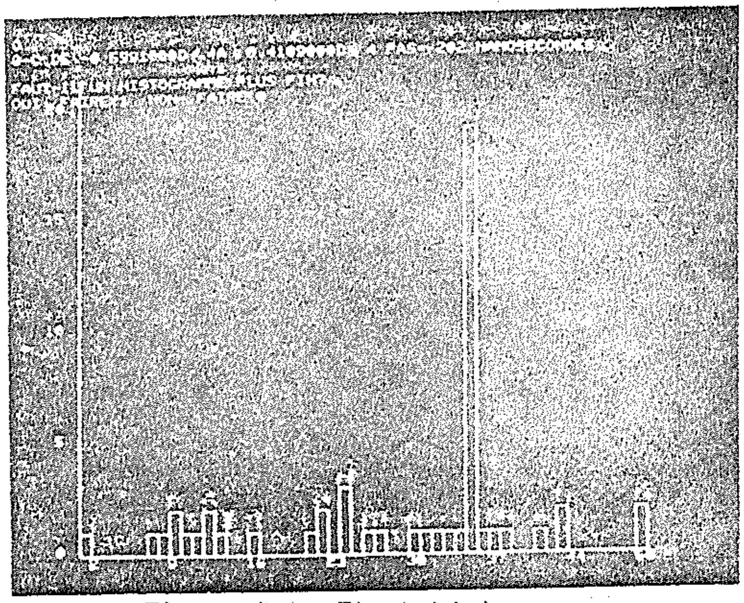
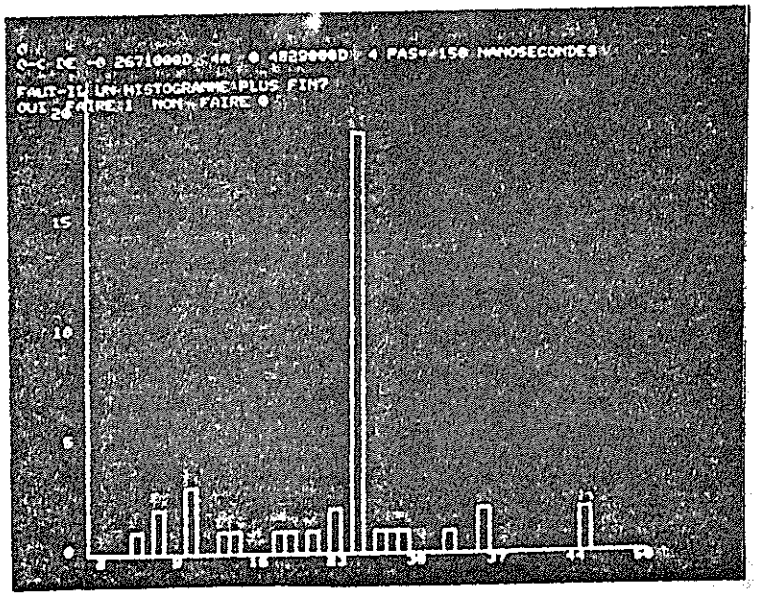


Figure 4: First histogram.

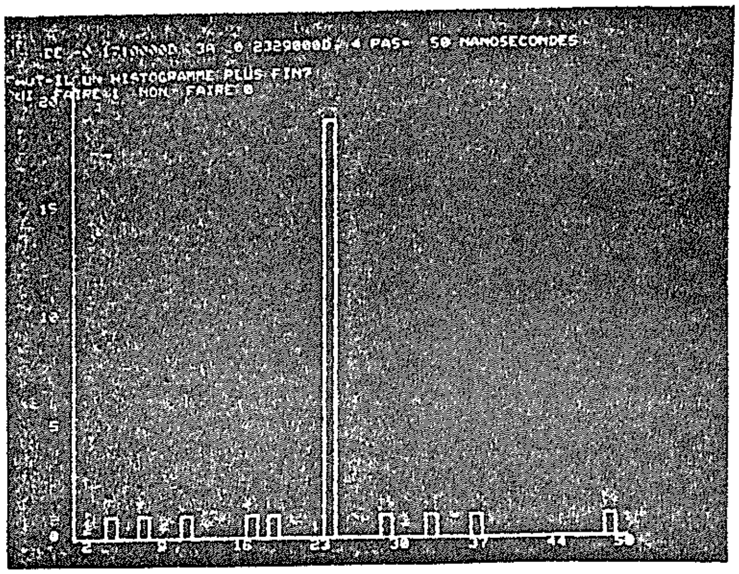
b. Request a new histogram. Central channel: 35; channel width: 150 ns. Figure 5 gives this second histogram.

c. Request a new histogram. Central channel: 24; channel width: 50 ns. Figure 6 gives this last histogram.



```
C-C DE -0 2671000D 4A 0 4829000D 4 PAS=150 NANOSECONDES
FAUT-IL UN HISTOGRAMME PLUS FIN?
OUI FAIRE=1 NON FAIRE 0
```

Figure 5: Second histogram.



```
DE -0 1210000D 3A 0 2329000D 4 PAS= 50 NANOSECONDES
FAUT-IL UN HISTOGRAMME PLUS FIN?
OUI FAIRE=1 NON FAIRE 0
```

Figure 6: Third histogram

d. Request a new histogram. Central channel: 24; channel width: 15 ns. Figure 7 gives this last histogram.

After this, do not request a new histogram and indicate that there is a probable result. At this stage, the teletype prints 19 measurements, the coefficients of the linear regression, the residual and the mean quadratic error (table 4).

5. Answer "OUI" three times when rejection are requested. Reject the observations numbered 11, 24, and 44. Then answer that there are no more rejection. The teletype prints the 16 remaining measurements and the other data as given in 5). They are

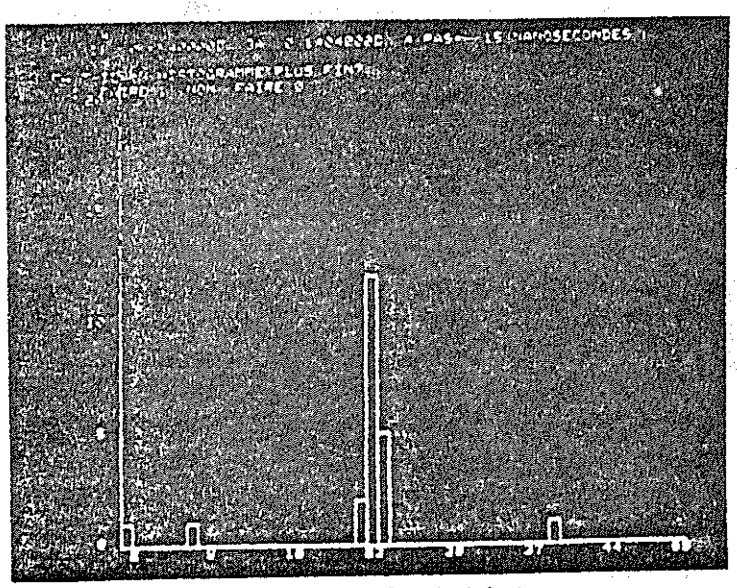


Figure 7: Last histogram.

```
RESULTATS DU  7   7  81 JJ A Ø H =    0.24447925D  7

NØ      J.JUL.                     MESURE                          J-C
 4 0.2444793322598O3D  7     .26067700622D  1              0.98788D -6
 5 0.244479332266930D  7     .26067820491D  1              0.98550D -6
10 0.244479332323720D  7     .26068777340D  1              0.99229D -6
11 0.244479332330824D  7     .26068897096D  1              0.98105D -6
13 0.244479332352113D  7     .26069256648D  1              0.99208D -6
15 0.244479332380531D  7     .26069737016D  1              0.99323D -6
18 0.244479332394741D  7     .26069977462D  1              0.99357D -6
21 0.244479332423186D  7     .26070459238D  1              0.99563D -6
24 0.244479332444501D  7     .26070820524D  1              0.98130D -6
28 0.244479332472917D  7     .26071303059D  1              0.99730D -6
32 0.244479332487101D  7     .26071544088D  1              0.99681D -6
34 0.244479332508411D  7     .26071906506D  1              0.99867D -6
37 0.244479332529677D  7     .26072268502D  1              0.99836D -6
38 0.244479332543864D  7     .26072510166D  1              0.99612D -6
39 0.244479332565154D  7     .26072873233D  1              0.10017D -5
42 0.244479332678716D  7     .26074815438D  1              0.10058D -5
43 0.244479332714167D  7     .26075423786D  1              0.10076D -5
44 0.244479332799295D  7     .26076888401D  1              0.10004D -5
49 0.244479332834754D  7     .26077500231D  1              0.10103D -5

X=  0.4570D-10 Y=  0.9932D -6 TM=0.712200D  5 OU=19 H47

NØ    Ø-C SEC     RESIDUS DTE
 4    0.98788D -6  -0.14109D -8
 5    0.98550D -6   0.12513D -8
10    0.99229D -6  -0.32995D -8
11    0.98105D -6   0.82193D -8
13    0.99208D -6  -0.19640D -8
15    0.99323D -6  -0.19983D -8
18    0.99357D -6  -0.17692D -8
21    0.99563D -6  -0.27111D -8
24    0.98130D -6   0.12464D -7
28    0.99730D -6  -0.24199D -8
32    0.99681D -6  -0.13646D -8
34    0.99867D -6  -0.23879D -8
37    0.99836D -6  -0.12377D -8
38    0.99612D -6   0.15652D -8
39    0.10017D -5  -0.31880D -8
42    0.10058D -5  -0.28207D -8
43    0.10076D -5  -0.31499D -8
44    0.10004D -5   0.73779D -8
49    0.10103D -5  -0.11561D -8

   ERR. QUADR. MØYENNE = 0.4582D -8
```

Table 4: Print-out after 4th histogram; 19 events retained.

reproduced in table 5.

```
RESULTATS DU  7   7  81 JJ A 3 H =   0.244479250  7

NO       J.JUL.                      MESURE                         O-C
 4 0.244479332259803D  7     .260677006220  1               0.98788D -6
 5 0.244479332266930D  7     .260678204910  1               0.98550D -6
10 0.244479332323720D  7     .260687773400  1               0.99229D -6
13 0.244479332352113D  7     .260692566480  1               0.99208D -6
15 0.244479332380531D  7     .260697370160  1               0.99323D -6
18 0.244479332394741D  7     .260699774620  1               0.99357D -6
21 0.244479332423186D  7     .260704592380  1               0.99563D -6
28 0.244479332472917D  7     .260713030590  1               0.99730D -6
32 0.244479332497101D  7     .260715440880  1               0.99681D -6
34 0.244479332508411D  7     .260719065060  1               0.99867D -6
37 0.244479332529677D  7     .260722685020  1               0.99836D -6
38 0.244479332543864D  7     .260725101660  1               0.99612D -6
39 0.244479332565154D  7     .260728732330  1               0.10017D -5
42 0.244479332678716D  7     .260748154380  1               0.10058D -5
43 0.244479332714167D  7     .260754237860  1               0.10076D -5
49 0.244479332834754D  7     .260775002310  1               0.10103D -5

X=  0.4745D-10 Y=  0.9949D -6 TM=0.712200D  5 OU=19 H47

NO   O-C SEC    RESIDUS DTE
 4   0.98788D -6   0.47492D-11
 5   0.98550D -6   0.26777D -8
10   0.99229D -6  -0.17872D -8
13   0.99208D -6  -0.40883D -9
15   0.99323D -6  -0.40017D -9
18   0.99357D -6  -0.14957D -9
21   0.99563D -6  -0.10485D -8
28   0.99730D -6  -0.68208D -9
32   0.99681D -6   0.39471D -9
34   0.99867D -6  -0.59633D -9
37   0.99836D -6   0.58598D -9
38   0.99612D -6   0.34103D -8
39   0.10017D -5  -0.13106D -8
42   0.10058D -5  -0.77171D -9
43   0.10076D -5  -0.10472D -8
49   0.10103D -5   0.11288D -8

 ERR. QUADR. MOYENNE = 0.1450D -8
```

Table 5: Print-out after the rejection of obs. 11, 24, and 44.

6. The rest of the conversation is straightforward. Answer:

   - Reflector : 03
   - Crater : 09 POSIDO-A
   - Pressure : 0660
   - Temperature : 012
   - Calibration : 0198
   - Noise : 0050

   These numbers are printed as indicated in table 6.

7. If the print-out of events is requested, this is done as shown in table 7.

```
NØ REFLECTEU  :    3

CRATERE SUIVI NØ :  9     PØSIDØ-A

PRESSIØN :      660

TEMPERATURE :       12

CALIBRATIØN :       198

BRUIT :     50    KILOHERTZ
```

Table 6:   Observation parameter print-out.

```
JJ  INSTANT DE TIR EN SEC.    A.R. EVENEMENT
 1      .710418027361513D  5   0.26067081706D  1
 2      .710418027361513D  5   0.26067086364D  1
 3      .710663427100116D  5   0.26067574988D  1
 4      .710724696681012D  5   0.26067700622D  1
 5      .710786273755526D  5   0.26067820491D  1
 6      .710970431653864D  5   0.26068150020D  1
 7      .711215570180306D  5   0.26068613663D  1
 8      .711215570180306D  5   0.26068670091D  1
 9      .711276942359855D  5   0.26068775002D  1
10      .711276942359855D  5   0.26068777340D  1
11      .711338323174038D  5   0.26068897096D  1
12      .711461072770936D  5   0.26069090105D  1
13      .711522259919125D  5   0.26069256648D  1
14      .711645021103463D  5   0.26069493433D  1
15      .711767788546231D  5   0.26069737016D  1
16      .711767788546231D  5   0.26069739378D  1
17      .711823981145443D  5   0.26069804696D  1
18      .711890566593214D  5   0.26069977462D  1
19      .712013149731651D  5   0.26070178892D  1
20      .712136324498718D  5   0.26070476208D  1
21      .712136324498718D  5   0.26070459238D  1
22      .712197711393090D  5   0.26070524341D  1
23      .712259096253886D  5   0.26070630278D  1
24      .712320488971583D  5   0.26070820524D  1
25      .712320488971583D  5   0.26070764653D  1
26      .712504806496095D  5   0.26071166268D  1
27      .712504806496095D  5   0.26071212179D  1
28      .712565999587552D  5   0.26071303059D  1
29      .712565999587552D  5   0.26071255142D  1
30      .712627378376316D  5   0.26071371506D  1
31      .712627378376316D  5   0.26071396320D  1
32      .712688556869579D  5   0.26071544088D  1
33      .712688556869579D  5   0.26071535811D  1
34      .712872674636206D  5   0.26071906506D  1
35      .712872674636206D  5   0.26071911001D  1
36      .712872674636206D  5   0.26071936716D  1
37      .713056410283163D  5   0.26072268502D  1
38      .713178981103605D  5   0.26072510166D  1
39      .713362933074042D  5   0.26072873233D  1
40      .713485689028205D  5   0.26073093225D  1
41      .714037541951761D  5   0.26074157082D  1
42      .714344105483402D  5   0.26074815433D  1
43      .714650404302753D  5   0.26075423796D  1
44      .715385906153006D  5   0.26076888401D  1
45      .715385906153006D  5   0.26076840856D  1
46      .715447106269697D  5   0.26077017019D  1
47      .715630895439871D  5   0.26077351185D  1
48      .715692271832714D  5   0.26077515939D  1
49      .715692271832714D  5   0.26077500231D  1
50      .715876018480821D  5   0.26077844672D  1
```

Table 7:   Print-out of all events.

SAO Prediction and Data Review Algorithms

by

James H. Latimer
Smithsonian Astrophysical Observatory
Cambridge, MA 02138

```
C       CODE FOR GENERATION OF THE SAO 333 QUICK LOOK OBSERVATION MESSAGE
C       HAS THE FOLLOWING FORM:  FOR EACH PASS,
      CALL GENHED(ISTORE,PRECAL,ISTAT,DATE,SAT,ISKY,
     + IHUM,ITEMP,IPRESS,ICHECK,LEX,IENC)
C       FOR EACH OBSERVATION TO BE TRANSMITTED:
      CALL GENRAN(ISTORE,NTOTGOOD,NPUNCH,ICHECK,IENC,LEX,
     + RANGE,POSEC,JHR,JMIN,ISEC,ICONF)
C       AND AT THE END OF A PASS
        CALL NEWLINE(ISTORE,"<5><7>END<15><15><12><0>",1,1)
C       NOTE THAT SUBROUTINE NEWLINE, WHOSE DETAILS ARE NOT IMPORTANT,
C       IS THE DATA SINK ROUTINE
        COMPILER DOUBLE PRECISION
        SUBROUTINE GENHED(ISTORE,PRECAL,ISTAT,DATE,SAT,ISKY,
     + IHUM,ITEMP,IPRESS,ICHECK,LEX,IENC)
        DIMENSION LINE(30),ICHECK(2)
C
C GENERATE A HEADER TO BE PUT INTO THE STORAGE AREA
C FOR LATER PUNCHING
C
        ICHECK(1)=0
        ICHECK(2)=0
        CALL NEWLINE(ISTORE,"<1><15><15><15><12>..LASER<15><15><12>",1,1)
        CALL HEDENC(LEX,LINE,IENC,ISTAT,DATE,ICHECK)
        CALL NEWLINE(ISTORE,LINE,1,1)
        CALL SATENC(LEX,LINE,IENC,SAT,ISKY,IHUM,ITEMP,IPRESS,
     + PRECAL,ICHECK)
        CALL NEWLINE(ISTORE,LINE,1,1)
        END

        COMPILER DOUBLE PRECISION
        SUBROUTINE HEDENC(LEX,IAR,IENC,ISTAT,DATE,ICHECK)
        DIMENSION IAR(2)
C
C ENCODES THE FIRST HEADING LINE OF THE QUICK-LOOK MESSAGE
C
        CALL ENCODE(IAR,22)0029 WRITE(IENC,1) ISTAT,DATE
1       FORMAT("<2><24>33333 ",I4,F7.0"<15><15><12>")
C MOVE LOW 5 DIGITS OF DATE DOWN ONE SPACE, REMOVING "."
        I=19
10      IF(I.LE.14) GO TO 20
        CALL PUTC(IAR,I,IGETC(IAR,I-1))
        I=I-1
             1 OT OG
             20      CALL PUTC(IAR,14,40K)
             C ADD IN THE CHARACTERS TO THE CHECKSUM
                     CALL BCHECK(IAR(2),3,ICHECK)
        END

        COMPILER DOUBLE PRECISION
        SUBROUTINE SATENC(LEX,IAR,IENC,SAT,ISKY,IHUM,ITEMP,
     + IPRESS,CAL,ICHECK)
        DIMENSION IAR(2)
C
C ENCODES THE SATELLITE LINE DATA IN THE QUICK-LOOK FORMAT
```

```
C
        XCAL=CAL*10.
        CALL ENCODE(IAR,40)
        WRITE(IENC,1) SAT,ISKY,IHUM,ITEMP,IPRESS,XCAL
1       FORMAT(" <3><46>"F8.0,I1,I2,I5,I6,F7.0" 00000<15><15><12>")
C MOVE PIECES OF FLOATING PT#'S, TO REMOVE DECIMAL PTS.
        DO 20 J=10,31,21
        I=J
        K=8
        IF(I.EQ.31) K=26
10      IF(I.LE.K) GO TO 20
        CALL PUTC(IAR,I,IGETC(IAR,I-1))
        I=I-1
        GO TO 10
20      CONTINUE
C MAKE SURE LEADING ZEROES WRITTEN
        DO 110 I=3,37
110     IF(IGETC(IAR,I).EQ.40K) CALL PUTC(IAR,I,60K)
C PUT SPACING IN
        DO 120 I=8,32,6
120     CALL PUTC(IAR,I,40K)
        CALL BCHECK(IAR(2),6,ICHECK)
        END

        COMPILER DOUBLE PRECISION
        SUBROUTINE GENRAN(ISTORE,NTOTGOOD,NPUNCH,ICHECK,IENC,
     + LEX,RANGE,POSEC,JHR,JMIN,ISEC,ICONF)
C
C GENERATE THE RANGE LINE TO POSSIBLY BE INSERTED INTO THE
C STORAGE AREA FOR LATER PUNCHING
C
        DIMENSION LINE(30)
C
        CALL RANENC(LEX,LINE,IENC,JHR,JMIN,ISEC,POSEC,ICONF,RANGE,ICHECK)
        CALL NEWLINE(ISTORE,LINE,NTOTGOOD,NPUNCH)
        END

        COMPILER DOUBLE PRECISION
        SUBROUTINE RANENC(LEX,IAR,IENC,IHR,IMIN,ISEC,POSEC,
     + ICONF,RANGE,ICHECK)
        DIMENSION ICHECK(2),IAR(2)
C
C ENCODES AND PUNCHES THANGE LINE FOR THE QUICK-LOOK MESSAGE
C
        CALL ENCODE(IAR,34)
        JSEC=ISEC/10
        KSEC=MOD(ISEC,10)
        SEC=POSEC*1.E6
        RAN=RANGE*10.
        WRITE(IENC,1) IHR,IMIN,JSEC,KSEC,SEC,ICONF,RAN
1       FORMAT(" <4><40>",2I2,I1,I2,F7.0,I3,F12.0"<15><15><12>")
C MOVE PIECES OF THE FLOATING PT NUMBERS, REMOVING DECIMAL PTS.
        DO 20 J=16,31,15
        I=J
        K=14
        IF(I.EQ.31) K=26
10      IF(I.LE.K) GO TO 20
        CALL PUTC(IAR,I,IGETC(IAR,I-1))
        I=I-1
        GO TO 10
20      CONTINUE
C MAKE SURE LEADING ZEROES WRITTEN
        DO 110 I=3,32
110     IF(IGETC(IAR,I).EQ.40K) CALL PUTC(IAR,I,60K)
C PUT IN SPACING
        DO 120 I=8,31,6
120     CALL PUTC(IAR,I,40K)
C ADD IN FINAL CHECKSUM, INSERT INTO LINE TO PUNCH
        CALL BCHECK(IAR(2),5,ICHECK)
        CALL PUTC(IAR,17,ICHECK(1)+60K)
        CALL PUTC(IAR,18,ICHECK(2)+60K)
C RESET CHECKSUM TO START OVER
        ICHECK(1)=0
        ICHECK(2)=0
        END

        COMPILER DOUBLE PRECISION
        SUBROUTINE BCHECK(IAR,N,ICHECK)
        DIMENSION ICHECK(2)
C
C SUBROUTINE TO TAKE N GROUPS OF 5 CHARACTERS, SEPARATED WIT
HC A BLANK, AND BUILD A 2 DIGIT 10'S MODULUS CHECKSUM WITH EACH
C LINE. ANY CHARACTERS LESS THAN 60 OCTAL (CHARACTER ZERO)
C ARE CHECKSUMMED AS ZERO.
C
        NCHAR=N*6
        J=1
        DO 200 I=1,NCHAR
        IF(MOD(I,6).EQ.0) GO TO 200
        J=J+1
        L=MOD(J,2)+1
        ICHECK(L)=MOD(ICHECK(L)+MAX0(0,IGETC(IAR,I)-60K),10)
200     CONTINUE
        END

C       CODE TO READ THE SAO333 QL FORMAT AND THE NASA QL FORMAT
      SUBROUTINE OBSCARD
C
c*****************************************************************
c
c       Obscard initiates passes. It searches the DATA RECORDS
c       until a pass identifier record is found. Then it
c       calls the appropriate subroutine to process that TYPE OF
c       observation. PASSES MAY ALSO BE INITIATED WITHIN EACH DATA
C       TYPE (IN LASERA, NASA, OR BAKER NUNN). COMMENTS IN RERUN
C       FILES (ALWAYS PREFACED BY ....) ARE DELETED. LINE_FEEDS
C       IN FIRST CHARACTER POSITION ARE DELETED. SPECIAL LOGIC
C       ENSURES THAT "UNDETECTED" WETTZELL PASSES ARE REJECTED IN FULL
```

```
C       TO AVOID THE OBSERVATION TIME BIAS PROBLEM.
c
c****************************************************************
c
      COMMON /PARAM/ FORMIN,NOBJS,OBJ1(100),OBJ2(100),LOBS(100),
     1NOBSV(100)
      COMMON /LASER/ SAT(25),NUMSAT,PASS(25),SUM(25)
      COMMON/LIMIT/ISTA(25),PRLIM(25,2),TMPLIM(25,2),CALLIM(25,2,2),
     1            RANLIM(25,2),EXPDATE(25)
        common/seq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
     1          TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
        common/nasac/nassat(25),saosat(25),nassta(25),saosta(25)
      common/lascoc/RCRD,pcar
        COMMON/WETZCOM/WCAL,ICORR,WETZFLG,WFLAG,RAN2WT
        DATA BLANK/'     '/
      character*80 RCRD,pcar*30,BLANK*5
        CHARACTER*4 nassat,saosat*7,nassta,saosta
      logical*1 WETZFLG(3),WFLAG,WNDFLG,PASSFLG
        double precision WCAL,RAN2WT
      INTEGER FORMIN,OBJ1,OBJ2,BUFF(8)
      INTEGER PASS,SUM,sat,SAONUM
        integer*2 passeq,OBSONE,BADHFLG
C
C-----INITIALIZE FOR PROCESSING OBSERVATIONS
C
      DO 6 J=1,NOBJS
      NOBSV(J)=0
 6    CONTINUE
      NGARB=0
        LPASS=0
        WFLAG=.FALSE.
c
c------parse observation RECORDS for an identifier and transfer to
c------the appropriate pass processing subroutine.
c
 100  READ(2,101,err=200)RCRD
 101  FORMAT(A)
      NGARB=NGARB+1
        J=INDEX(RCRD,'....')   ! DELETE COMMENT RECORDS (RERUNS)
        IF (J.NE.0) GOTO 100
121     continue
        NDETFLG=0   ! FIND NASA OBS WITHOUT PROPER HEADER
        IF (RCRD(1:1).NE.CHAR(10)) GOTO 123   ! DELETE A LINE_FEED
        RCRD(1:80)=RCRD(2:80)//BLANK(1:1)
123     J=INDEX(RCRD,'33333')
        IF (J-1) 153,152,151
151     RCRD(1:80)=RCRD(J:80)//BLANK(1:J-1)   ! LEFT SHIFT RECORD
152     IF (LASTC(RCRD,80).LE.8) GOTO 103     ! WETTZELL PASS DETECTED
        IF (RCRD(J+6:J+9).NE.'7834') GOTO 106    ! SAO PASS DETECTED
        WRITE(11,902) NGARB-1   ! UNCORRECTABLE WETTZELL PASS
        NGARB=0
        WRITE(11,162) RCRD(1:20)
162     FORMAT(A20/'....WETTZELL PASS UNDETECTED OR NO CORRECTIONS')
```

```
      CALL WNCOREJ(RCRD,IFLG)
      NGARB=0
      IF (IFLG.EQ.1) GOTO 121
      GOTO 100
153   IF (INDEX(RCRD,'LASERQL')) 154,154,109    ! NASA PASS DETECTED
154   IF (INDEX(RCRD,'..LASERQL')) 155,155,109 ! NASA PASS DETECTED
155   IF (INDEX(RCRD,'LASER QL')) 156,156,109  ! NASA PASS DETECTED
156   if(lastc(RCRD,80).eq. 0) go to 100
      J=INDEX(RCRD,'END')
      IF (J.EQ.0.OR.J.GT.2) GOTO 157
      WRITE(11,101) RCRD   ! END OF AN UNLABELED PASS (REJECTED)
      WRITE(11,161) NGARB
161   FORMAT('....NUMBER OF GARBAGE RECORDS =',I3)
      NGARB=0
      GOTO 100
157   IF (RCRD(1:3).NE.'1BB') GOTO 111
      NDETFLG=1
      GOTO 109
111   WRITE(11,101) RCRD     ! GARBAGE RCRDS DUMPED
      IF (LPASS.EQ.PASSEQ) GOTO 100
      LPASS=PASSEQ   ! SUPPRESS SEQUENCES OF REJECT MESSAGES
      WRITE(11,102)
102   FORMAT('....REJECT/ GARBAGE RECORD')
      GO TO 100
C
C     CALL WETTZELL STATION
C
103   IF (NGARB.GT.5) WRITE(11,902) NGARB-1
902   FORMAT('....NUMBER OF GARBAGE RECORDS = ',I4)
      CALL WETZEL(RCRD,*100)   ! PROCESS CORRECTIONS FOR WETTZELL OBSERVATIONS
      GOTO 107
120   ngarb=0
      LPASS=0
      goto 121
C
c     call lasera
C
106   IF (NGARB.GT.5) WRITE(11,902) NGARB-1
107   CALL LASERA(*120)   ! PROCESS SAO LASER OBSERVATIONS
      NGARB=0
      LPASS=0
      WFLAG=.FALSE.
      GO TO 100
C
C    CALL NASA
C
109   IF (NGARB.GT.5) WRITE(11,902) NGARB-1
      CALL NASA   ! PROCESS NASA LASER OBSERVATIONS
      NGARB=0
      LPASS=0
      GO TO 100
C
C...PROCESSING TERMINATES NORMALLY ONLY WHEN FOR002.DAT IS EMPTY
C
```

```
200     CONTINUE
        RETURN
        END

        SUBROUTINE LASERA(*)
c
c******************************************************************
c
c       lasera reads the observation records for a pass. each
c       record is parsed for type, i.e. date(header), satellite
c       (header),or range and related observations. the
c       appropriate processing subroutine is called for each
c       record. EACH DATE RECORD INITIATES A NEW PASS.
C       COMMENT RECORDS (....) ARE DELETED FROM RERUN FILES.
C       PART OF THE FIRST OBSERVATION AND THE NUMBER OF OBSERVATIONS
C       IN THE ORIGINAL DATA SET ARE CARRIED THROUGH MULTIPLE RERUNS
C       BY "____" CHARACTERS. RECORDS ARE SHIFTED LEFT IF NECESSARY.
C       A PASS IS REJECTED IF THERE IS ANY ERROR IN A HEADER RECORD.
C       AN OBSERVATION IS REJECTED IF THERE IS ANY ERROR IN THE
C       GIVEN OBSERVATION RECORD. WETTZELL OBSERVATIONS ARE TIME-
C       SHIFTED AND TIME UNBIASED.
c
c******************************************************************
c
        common/seq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
     1          TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
      COMMON/LASER/SAT(25),NUMSAT,PASS(25),SUM(25)
      COMMON/LASCOM/CODE(4),DECADE,DT,C,Y,D,M,HR,MIN
     1  ,SEC,IDENT,ILLCODE,IFIRST,JSTA,NOGOOD,OBS,PR,RANGE,RELHUM,
     2     STAT,TEMP,PRESS,HUM,TCENT,JSAT,TM,CAL,OBSFLAG
     3  ,ACOEFF,BCOEFF,PRECAL,POSTCAL,ALOGSS,DIFF,NDIFF,PPCAL
     4    ,NCHKSM1,NCHKSM2
      common/lascoc/card,pcar
      COMMON/SOLVE/DATER(14),STORE(43),WORK(115),ARRAY(6),NRANGE
      COMMON/BLOCK/ IBLOCK(800)
      COMMON/SWITCH/GO,SW1,SW2,KILL1,KILL2
      COMMON /PNTS/ IPNTS,NUMPTS,NUMGPTS
        COMMON/WETZCOM/WCAL,ICORR,WETZFLG,WFLAG,RAN2WT
        DATA BLANK/'     '/
      INTEGER DELAY,SAONUM,FLAG,GO,HR,OBS
      INTEGER PASS,SAT,SEC,SUM,SW1,SW2,TEMP
      logical*1 WETZFLG(3),WFLAG,OBSFLAG(17),WNDFLG,PASSFLG
        integer*2 passeq,OBSONE,BADHFLG
      LOGICAL CHKCHR
      LOGICAL TMPCHK,PRCHK,CALCHK,RANCHK
      character*30 pcar
      character*80 card
        character*10 itest,BLANK*5
        double precision r,range,WCAL,RAN2WT
      INTEGER CODE
C
C-----INITIALIZE
C
      C=2.997925E8
```

```
        IRRFLG=0   ! RERUN FLAG (0,1,2)
      ITEST='           '
      IPNTS=0
      NUMPTS=0
      NUMGPTS=0
      KILL1=0
      KILL2=0
      SW1=0
      SW2=0
      GO=0
      SW2=-1
      do 500 j=1,17
500   obsflag(j)=.false.
      IFIRST=0
      NCHKSM1=0
      NCHKSM2=0
      PASSFLG=.FALSE.
      go to 110
C
C-----READ A RECORD AND DETERMINE ITS DISPOSITION
C
  100 CONTINUE
      obsflag(17)=.false.
      READ(2,3,err=250,END=800)card
    3 FORMAT(a)
      J=INDEX(CARD,'....')
      IF (J.NE.0) GOTO 100   ! DELETE COMMENT RECORDS
106   J=INDEX(CARD,'____')
      IF (J.EQ.0) GOTO 107   ! SAVE SPECIAL COMMENTS OVER MULTIPLE RERUNS
      CALL REJREC(CARD)
      IRRFLG=IRRFLG+1
      OBSONE=1
      GO TO 100
107   IF (CARD(1:1).NE.CHAR(10)) GOTO 108   ! DELETE LINE_FEED IN 1ST LOCATION
      J=LASTC(CARD,80)
      CARD(1:J-1)=CARD(2:J)
      CARD(J:J)=' '
      GOTO 106
108   J=0
      DO 102 I=1,5
        IF (CARD(I:I).NE.' ') GOTO 103
102     J=J+1
103     IF (J.GT.0) THEN   ! SHIFT LEFT IF NECESSARY
        CARD(1:80-J)=CARD(J+1:80)
        CARD(81-J:80)=BLANK(1:J)
        ENDIF
      IF(card(1:5).EQ.'     ') GOTO 100      ! DELETE BLANK RECORDS
      IF(card(1:3).EQ.'END') GOTO 250        ! OBS. PASS COMPLETED
        IF (CARD(1:5).NE.'33333') GOTO 110
        IW=0
        IF (LASTC(CARD,45).NE.5) GOTO 405
        IW=1
        GOTO 410
405     IF (LASTC(CARD,45).NE.17) GOTO 110
```

```
        IF (WFLAG) GOTO 110
410     IF (IRRFLG.GT.1) GOTO 422
        WRITE(11,411) NSEQ
411     FORMAT('____NUMBER OF OBSERVATIONS IN PASS =',I3/'END')
422     CALL INCPASS
        IF (IW.EQ.1) GOTO 300
c       squeeze out blanks, go from 35 to 30 characters in pcar
110     pcar=card(1:5)//card(7:11)//card(13:17)//card(19:23)//
        1card(25:29)//card(31:35)
C
C-----SETTER SPECIFIES RECORD TYPE (BASED ON THE NUMBER OF DIGITS)
C-----VIA THE PARAMETER JUMP
C------JUMP   RECORD TYPE
C       1       DATE (AND STATION)
C       2       SATELLITE (AND ITS ENVIRONMENTAL PROPERTIES)
C       3       RANGE (AND OBSERVATION TIME AND RELATED VALUES)
C       4,5     GARBAGE RECORD
C
      CALL SETTER(JUMP,OBSFLAG,wflag)
C
      GO TO (130,130,130,300,200),JUMP
300     WRITE(11,253)
253     FORMAT('END')
        return 10

C-----SETSW SETS THE SWITCHES, AND GIVES THE PLACE TO JUMP AT THE END
C
130 CALL SETSW(JUMP)
      IF( JUMP.LT.3 .AND. IPNTS.NE.0 ) GOTO 260
      GO TO (160,170,180,200,200),JUMP
C
C-----PROCESS THE DATE RECORD
C
  160 CALL DATECRD
      GO TO 100
C
C-----PROCESS THE SATELLITE RECORD
C
  170 CALL SATCRD
      GO TO 100
C
C-----PROCESS THE RANGE RECORD
C
  180 CALL RANCRD
      GO TO 100
C
C-----REJECT GARBAGE RECORDS TO RERUN FILE
C
200     WRITE(11,3) card
        CALL REJERS(OBSFLAG,17,17)
      GO TO 100
C
C-----SAVE THE TOTAL NUMBER OF OBSERVATIONS IN ORIGINAL DATA SET
C
```

```
250     IF (WNDFLG) WNDFLG=.FALSE.
        IF (IRRFLG.GT.1) GOTO 252
        WRITE(11,259) NSEQ
259     FORMAT('____NUMBER OF OBSERVATIONS IN PASS =',I3)
252     WRITE(11,3) CARD
         CALL INCPASS
C       IF( IPNTS.EQ.0 ) RETURN
    JUMP=3
260     CONTINUE
      IPNTS=0
        NUMPTS=0
        NUMGPTS=0
      GOTO (160,170,999), JUMP
800     WRITE(11,259) NSEQ
999     RETURN
      END
       SUBROUTINE DATECRD
c
c************************************************************
C
c       process date records. after checking for illegal characters
c       convert station, year, month, and day to integer
c       type, and CHECK that each is a legitimate value.
c       APPROPRIATE obsflag element is set .TRUE. if illegal station
C       or date COMPONENT is found.
c
c************************************************************
c
      COMMON/SOLVE/DATER(14),STORE(43),WORK(115),ARRAY(6),NRANGE
        common/seq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
        1           TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
      COMMON/LASER/SAT(25),NUMSAT,PASS(25),SUM(25)
      COMMON/LASCOM/CODE(4),DECADE,DT,C,Y,D,M,HR,MIN
     1  ,SEC,IDENT,ILLCODE,IFIRST,JSTA,NOGOOD,OBS,PR,RANGE,RELHUM,
     2     STAT,TEMP,PRESS,HUM,TCENT,JSAT,TM,CAL,OBSFLAG
     3  ,ACOEFF,BCOEFF,PRECAL,POSTCAL,ALOGSS,DIFF,NDIFF,PPCAL
     4,NCHKSM1,NCHKSM2
        common/lascoc/card,pcar
      COMMON/BLOCK/IBLOCK(600)


      COMMON/SWITCH/GO,SW1,SW2,KILL1,KILL2


      COMMON/YEAR/IMO,IDY,IYEAR,IFYEAR


        COMMON/WETZCOM/WCAL,ICORR,WETZFLG,WFLAG,RAN2WT
        DATA DPMO/31,28,31,30,31,30,31,31,30,31,30,31/
      logical*1 WETZFLG(3),WFLAG,OBSFLAG(17),WNDFLG,PASSFLG
        character*80 card
        character*30 pcar
        double precision r,range,WCAL,RAN2WT
        INTEGER*2 PASSEQ,OBSONE,DPMO(12),BADHFLG
      INTEGER CODE,Y,D
      INTEGER SNUM,DECade,stat
      INTEGER PASS,SAT,SEC,SUM,SW1,SW2,TEMP
      INTEGER DELAY,SAONUM,FLAG,GO,HR,OBS
      LOGICAL CHKCHR
C
        DO 5 I=1,17
        OBSFLAG(I)=.FALSE.
5       CONTINUE
        BADHFLG=0
        OBSONE=0
        J=LASTC(CARD,40)
      WRITE(11,10) card(1:J)
   10 FORMAT(a)
C
C-----COMPUTE CHECKSUMS
C
30    DO 32 I=1,15,2
32    NCHKSM1=NCHKSM1 + ichar(pcar(i:i))-48
      DO 34 I=2,14,2
34    NCHKSM2=NCHKSM2 + ichar(pcar(i:i))-48
  100 CONTINUE
C
C-----CHECK STATION NUMBER
C
        IF(PCAR(6:6).EQ.' ') PCAR(6:6)='0'
      IF(CHKCHR(pcar,6,9)) GO TO 210
      JSTA=ICONV(pcar,6,9)
C      IF WETTZELL STATION, VERIFY IT WAS DETECTED IN OBSCARD, ELSE
C      THE CALIBRATION AND TIME SHIFT (AND CORRECTION) WOULD NOT BE MADE.
C     IF NOT DETECTED, ALL RAW WETTZELL DATA WILL BE AVAILABLE ON FOR011.DAT
C      FILE.
        IF (JSTA.NE.7834) GOTO 45
        IF (JSTA.EQ.7834.AND.WFLAG.EQ..TRUE.) GOTO 45
        WNDFLG=.TRUE.
        GOTO 1000
45      Y=0
        M=0
        D=0
      CALL TARGDIS(JSTA,DIS,Y,M,D)
C
C-----IF DIS=-99., THEN THE STATION WAS NOT FOUND
C
      IF(DIS.EQ.-99.) go to 210
        stat=jsta
  115 CONTINUE
C
C-----CHECK FOR ILLEGAL MONTH.
C
      IF(CHKCHR(pcar,12,13)) GO TO 118
      M=iconv(pcar,12,13)
      IF(M.le.12.and.M.ge.1) go to 120
  118 OBSFLAG(14)=.true.
      GO TO 120
```

```
C
C------CHECK ILLEGAL YEAR DATE
C
 120 CONTINUE
      IF(CHKCHR(pcar,10,11)) GO TO 125
      DECADE=ichar(pcar(10:10))-48
      IF(decade.EQ.0.and.pcar(11:11).gt.'6' ) DECADE=7
      IF(decade.EQ.0 .AND.pcar(11:11).LE.'6') DECADE=8
        pcar(10:10)=char(decade+48)
      Y=iconv(pcar,10,11)
      CALL YRCHK(Y,M,OBSFLAG)
      IF(.not.OBSFLAG(15)) go to 130
C     year change problem causes no calc of sec, obsin rancrd
125    OBSFLAG(15)=.true.
  130 CONTINUE
C
C------CHECK FOR ILLEGAL DAY.
C
      IF(CHKCHR(pcar,14,15)) GO TO 135
      D=iconv(pcar,14,15)
        K=0
        ITM=Y-76
        IF ((ITM/4)*4.EQ.ITM) K=1
        IF (OBSFLAG(14)) GOTO 999
        IF(M .EQ. IMO .AND. D .GT. IDY)GO TO 135
      IF(D.le.DPMO(M)+K.and.D.ge.1) go to 999
  135 OBSFLAG(13)=.true.
200   GO TO 999
210   OBSFLAG(16)=.true.
      GO TO 115
C
  999 CONTINUE
        do 500 i=13,17
        if(obsflag(i)) BADHFLG=1
500     CONTINUE
        IF (BADHFLG.EQ.1) GOTO 1000
        IF (WFLAG) CALL OBSTMSH
1000    CALL REJERS(OBSFLAG,13,17)
        RETURN
        END

      SUBROUTINE SATCRD
C
c*****************************************************************
c
C------process a SATELLITE record (HEADER). convert ASCII DATA FIELDS TO
C------VARIABLES (satellite identification,skycode, humidity,
c------temperature, pressure and calibration).  IF ANY ERROR,
C------ENTIRE PASS IS REJECTED.
C                OBSFLAG(I)      MEANING IF TRUE
C                       7        BAD CALIBRATION
C                       8        BAD PRESSURE
C                       9        BAD TEMPERATURE
C10     BAD HUMIDITY
```

```
C                         11        BAD SKYCODE
C                         12        BAD SATELLITE IDENTIFICATION

C
c*************************************************************
c
        common/seq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
     1          TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
      COMMON/LASER/SAT(25),NUMSAT,PASS(25),SUM(25)
      COMMON/LASCOM/CODE(4),DECADE,DT,C,Y,D,M,HR,MIN
     1  ,SEC,IDENT,ILLCODE,IFIRST,JSTA,NOGOOD,OBS,PR,RANGE,RELHUM,
     2     STAT,TEMP,PRESS,HUM,TCENT,JSAT,TM,CAL,OBSFLAG
     3  ,ACOEFF,BCOEFF,PRECAL,POSTCAL,ALOGSS,DIFF,NDIFF,PPCAL
     4    ,NCHKSM1,NCHKSM2
        common/lascoc/card,pcar
      COMMON/SOLVE/DATER(14),STORE(43),WORK(115),ARRAY(6),NRANGE
      COMMON/BLOCK/IBLOCK(600)
      COMMON/SWITCH/GO,SW1,SW2,KILL1,KILL2
      COMMON/LIMIT/ISTA(25),PRLIM(25,2),TMPLIM(25,2),CALLIM(25,2,2),
     1           RANLIM(25,2),EXPDATE(25)
        COMMON/CONN/CON(25)
        CHARACTER*1 CON
        character*80 card
        character*30 pcar
      double precision r,range
        integer*2 passeq,OBSONE,BADHFLG
        INTEGER TMUNPTS,TMUNGPTS
      INTEGER ALPHA,DELAY,SAONUM,FLAG,GO,HR,OBS
      INTEGER PASS,SAT,SEC,SUM,SW1,SW2,TEMP,stat
      INTEGER CODE,Y,D
      logical*1 OBSFLAG(17),WNDFLG,PASSFLG
      LOGICAL TMPCHK,PRCHK,CALCHK,RANCHK,chkchr
      INTEGER PSURE
C
        J=LASTC(CARD,40)
      WRITE(11,20) card(:J)
   20 FORMAT(a)
      PRECAL=0.
        POSTCAL=0.
      DIFF=0.0
        do 500 i=1,17
500     obsflag(i)=.false.
        IF (WNDFLG) GOTO 400
C
C-----COMPUTE CHECKSUMS
C
   30 DO 32 I=1,29,2
   32 NCHKSM1=NCHKSM1 +ichar(pcar(i:i))-48
      DO 34 I=2,30,2
   34 NCHKSM2=NCHKSM2+ichar(pcar(i:i))-48
   36 CONTINUE
C
C-----CHECK FOR BAD SATELLITE IDENTIFICATION
C
```

```
      IF(.not.CHKCHR(pcar,1,7)) go to 100
      GO TO 118
  100 IDENT=ICONV(pcar,1,7)
      DO 115 i=1,NUMSAT
      jsat=i
      IF(SAT(i).EQ.IDENT) GO TO 120
      jsat=0
  115 CONTINUE
  118 OBSFLAG(12)=.true.
  120 CONTINUE
C
C------GET TARGET DISTANCE IF POSSIBLE
C
      IF(OBSFLAG(16)) GO TO 300
      CALL TARGDIS(JSTA,decade,Y,M,D)
      IF(decade.EQ.-99.) GO TO 300
  125 CONTINUE
C
C------CHECK FOR BAD SKYCODE
C
      ILLCODE=ichar(pcar(8:8))-48
      IF(ILLCODE.LE.2.and.ILLCODE.GE.0) GO TO 130
      OBSFLAG(11)=.true.
  130 CONTINUE
C
C----CHECK FOR BAD HUMIDITY CHARACTERS
C
      RELHUM=.50
        iHUM=50
      IF(CHKCHR(pcar,9,10)) GO TO 135
      iHUM=iconv(pcar,9,10)
        hum=floatj(ihum)
    IF(HUM.LT.0.0.OR.HUM.GT.100.0) GO TO 135
        relhum=hum/100.
      GO TO 140
  135 OBSFLAG(10)=.true.
  140 CONTINUE
C
C------CHECK FOR BAD TEMPERATURE
C
      IF(CHKCHR(pcar,11,14)) GO TO 145
        IF (PCAR(11:11).GE. '2') GOTO 145
      TEMP=ICONV(pcar,11,14)
C
C------pcar(11:11) IS 0 OR 1 DEPENDING ON + OR - TEMPERATURE, SO THESE STEPS
C------CONVERT THIS TO THE CORRECT TEMPERATURE.
C
      K=1000-TEMP
      TCENT=FLOAT(TEMP)/10.
      IF(K.LE.0) TCENT=FLOAT(K)/10.
C
C------PUT SIGN ON THE TEMPERATURE
C
      pcar(11:11)=' '
      IF(K.LT.0) pcar(11:11)='-'
      IF(OBSFLAG(16)) GO TO 150
        IF(.NOT. TMPCHK(TCENT,JSTA))GO TO 145
142     temp=tcent
      GO TO 150
  145 TCENT=25.
      OBSFLAG(9)=.true.
C
C------CHECK FOR BAD PRESSURE CHARACTERS
C
150     CONTINUE
      IF(CHKCHR(pcar,16,19)) GO TO 155
        pr=floatj(iconv(pcar,16,19))
        IF(.NOT. PRCHK(PR,JSTA))GO TO 155
152     press=pr
        go to 160
  155 PR=1000.
      OBSFLAG(8)=.true.
        GOTO 152
C
C------CHECK FOR BAD CALIBRATION CHARACTERS
C
160     CONTINUE
        IF(CHKCHR(pcar,20,25)) GO TO 164
        IF(CHKCHR(pcar,26,30)) GO TO 164
        DO 162 I=1,25
        IF (JSTA.EQ.ISTA(I)) GOTO 163
162     CONTINUE
        GOTO 164
163     K=I
        IF (CON(K).EQ.'X') K=K+1
        tm1=float(iconv(pcar,20,25))/10.
        tm2=float(iconv(pcar,26,30))/10.
        IF(JSTA .EQ. 7835) THEN
                        TM1 = TM1*2.0
                        TM2 = TM2*2.0
                ENDIF


        IF(OBSFLAG(16)) GOTO 164
C
        IF (TM1.EQ.0.) THEN
                IF (TM2.EQ.0.) GOTO 167 ! TM1=0.  TM2=0.
                GOTO 166                  ! TM1=0.  TM2>0.
        ELSE
    IF (TM2.EQ.0.) GOTO 165   ! TM1>0.  TM2=0.
                GOTO 167                  ! TM1>0.  TM2>0.
        ENDIF
C
  164   TM=1000.
        OBSFLAG(7)=.true.
        GOTO 170
165     IF (TM1.LT.CALLIM(K,2,1).OR.TM1.GT.CALLIM(K,1,1)) GOTO 164
        TM=TM1
        GOTO 170
166     IF (TM2.LT.CALLIM(K,2,1).OR.TM2.GT.CALLIM(K,1,1)) GOTO 164
        TM=TM2
        GOTO 170
167     TM2=float(ICONV(pcar,20,20))*10000.+TM2
        IF (TM1.LT.CALLIM(K,2,1).OR.TM1.GT.CALLIM(K,1,1)) GOTO 164
        IF (TM2.LT.CALLIM(K,2,1).OR.TM2.GT.CALLIM(K,1,1)) GOTO 164
        TM=(TM1+TM2)/2.
      PRECAL=TM1-TM
      POSTCAL=TM2-TM
      DIFF=PRECAL-POSTCAL
      DIFF=AMOD(DIFF,100.0)
      NDIFF=0
C
C------COMPUTE CALIBRATION
C
170     CONTINUE
      CAL=FUNCAL(TM,TCENT,RELHUM,PR,decade)
C
C------ADVANCE THE PASS NUMBER
C
175     CONTINUE
        do 508 i=7,12
508     if(obsflag(i))go to 180
      PASS(JSAT)=PASS(JSAT)+1
      IFIRST=1
C
C------FOR THIS OBSERVATION PUT BAD DATA COMMENTS TO REJECT FILE
C
180     CONTINUE
        if(obsflag(16)) CALL REJERS(OBSFLAG,16,16)
        do 501 i=7,12
501     if(obsflag(i)) BADHFLG=1
400     CONTINUE
        CALL REJERS(OBSFLAG,7,12)
      RETURN
C
300     CONTINUE
      decade=100.
      OBSFLAG(16)=.true.
      GO TO 125
      end


        SUBROUTINE RANCRD
C
c*****************************************************************
c
C------process A RANGE DATA record. COMPUTE THE ONE-WAY RANGE IN
c------centimers. also process hours, minutes, seconds, confidence code,
c------and calibration.
C------THEN, via stdfrm, write THE REDUCED OBSERVATION DATA from  LASCOM
c------to THE DATA STREAM file for001.dat.
C
```

```
C                OBSFLAG(I)      MEANING
C                       1        SKY CODE
C                       2        RANGE (DATA OR VALUE)
C                       3        SECONDS
C                       4        ILLEGAL MINUTES
C                       5        ILLEGAL HOURS
C                       6        ILLEGAL CHECKSUM
C
C*****************************************************************
c
        COMMON/SEq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
     1          TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
        COMMON/LASER/SAT(25),NUMSAT,PASS(25),SUM(25)
      COMMON/LASCOM/CODE(4),DECADE,DT,C,Y,D,M,HR,MIN
     1  ,SEC,IDENT,ILLCODE,IFIRST,JSTA,NOGOOD,OBS,PR,RANGE,RELHUM,
     1     STAT,TEMP,PRESS,HUM,TCENT,JSAT,TM,CAL,OBSFLAG
     2  ,ACOEFF,BCOEFF,PRECAL,POSTCAL,ALOGSS,DIFF,NDIFF,PPCAL
     3    ,NCHKSM1,NCHKSM2
        common/lascoc/card,pcar
      COMMON/SOLVE/DATER(7,2),STORE(43),WORK(115),ARRAY(6),NRANGE
      COMMON/BLOCK/IBLOCK(600)
      COMMON /PNTS/ IPNTS,NUMPTS,NUMGPTS
        integer*2 passeq,OBSONE,BADHFLG
      INTEGER CHKSUM
        logical*1 obsflag(17),WNDFLG
      INTEGER DELAY,SAONUM,FLAG,GO,HR,OBS
      INTEGER PASS,SAT,SEC,SUM,SW1,SW2,TEMP,stat
      INTEGER CODE,Y,D
      double precision r,range
        LOGICAL*1 PASSFLG
      LOGICAL TMPCHK,PRCHK,CALCHK,RANCHK
      LOGICAL CHKCHR
        character*80 card
        character*30 pcar
      DATA KPASS/0/
C
C       SAVE FIRST OBSERVATION OF A PASS ONLY OVER MULTIPLE RERUNS
C
        IF (OBSONE.EQ.1) GOTO 630
        WRITE(11,629) CARD(1:11)
629     FORMAT('_____',A11)
        OBSONE=1
630 IF (WNDFLG) GOTO 500
        do 641 k=1,17
641     obsflag(k)=.false.
   50   CONTINUE
C
      IF( KPASS.NE.PASSEQ ) IUPDATE=0
      KPASS=PASSEQ
C
C-----CHECK IF ILLEGAL HOURS CHARACTERS
C
      IF(CHKCHR(pcar,1,2)) GO TO 110
      hr=iconv(pcar,1,2)
```

```
C
C------CHECK IF HOURS LEGAL
C
      IF(HR.LE.23.AND.HR.GE.0) GO TO 116
110     CONTINUE
        obsflag(5)=.true.
      GO TO 122
  116 CONTINUE
C
C     ADVANCE DATE BY ONE DAY IF MIDNIGHT IS CROSSED
C
      IF( HR.EQ.23 ) IUPDATE=1
      IF( IUPDATE.EQ.0 .OR. HR.EQ.23 ) GO TO 122
      IF( HR.NE.0 ) GOTO 122
      CALL MIDNIT(Y,M,D)
      IUPDATE=0
C
C
C------CHECK IF MIN,SEC LEGAL
C
122     CONTINUE
      IF(CHKCHR(pcar,3,4)) GO TO 118
        min=iconv(pcar,3,4)
      IF(MIN.LE.59.AND.MIN.GE.0)  GO TO 120
118     CONTINUE
        obsflag(4)=.true.
120     IF(CHKCHR(pcar,5,12)) GO TO 125
      IF(pcar(5:5).LE.'5') go to 130
125     CONTINUE
        obsflag(3)=.true.
130     CONTINUE
    sec=iconv(pcar,5,12)
C
C------GET UNCORRECTED 2-WAY RANGE
C
     IF(.NOT.CHKCHR(pcar,16,25)) GO TO 140
     obsflag(2)=.true.
140     CONTINUE
      CODE(1)=ICONV(pcar,15,15)
      IF( pcar(15:15).LE.'2' .AND. pcar(15:15).GE.'0' ) GOTO 150
      obsflag(1)=.true.
150     CONTINUE
      NCHKSM1=0
        NCHKSM2=0
      CHKSUM=0
      IF( pcar(13:14).EQ.'00') GOTO 160
c  delete next line when stations give correct checksum for first
c  OBSERVATION record IN A PASS.
      if (nseq.eq.3) goto 160
      if (pcar(13:13).eq.'0'.or.pcar(14:14).eq.'0') goto 160
      DO 152 I=1,25,2
  152 nchksm1=nchksm1 + ichar(pcar(i:i))-ichar('0')
      nchksm1=nchksm1 - ichar(pcar(13:13))+ichar('0')
      CHKSUM = MOD(NCHKSM1,10)*10
      DO 153 I=2,24,2
  153 nchksm2=nchksm2 + ichar(pcar(I:i))-ichar('0')
      nchksm2=nchksm2 - ichar(pcar(14:14))+ichar('0')
      CHKSUM = MOD(NCHKSM2,10) + chksum
  158 NCHKSM1=0
        NCHKSM2=0
        if(chksum.EQ.iconv(pcar,13,14))go to160
        OBSFLAG(6)=.TRUE.
160     CONTINUE
      r=dflotj(ICONV(pcar,16,20))*1.d05 + dflotj(iconv(pcar,21,25))
      r=(r/1.d01)+dble(CAL)
      r=r*1.d-09
      RANGE =2.997925d08*r
C
C------CONVERT TO 1-WAY RANGE IN CM.
C
      RANGE=RANGE*5.d01
      IF(OBSFLAG(12).OR.OBSFLAG(2)) GO TO 165
      IF(RANGE.GE.9900000000.) GO TO 300
      IF(RANCHK(RANGE,JSAT,OBSFLAG))go to 165
300       CONTINUE
      obsflag(2)=.true.
  165 CONTINUE
        if(ifirst.eq.0)go to 170
        do 888 j=4,17
888     if(obsflag(j))go to 170
      SYSCAL= TM - FUNCAL(0.0,TCENT,RELHUM,PR,RS)
C
C_____PREPARE A TRANSIT DATA SORT MAP RECORD
C
        CALL TRANSMAP(lndx)
        TRNFLG=1   !IF 0, WILL BYPASS TRANSORT IN BOLO1
C
      IFIRST=0
170     CONTINUE
        IPNTS=1
        do 891 j=1,6
891     if(obsflag(j))go to 600
        goto 180
600     CALL REJREC(CARD)
        CALL REJERS(OBSFLAG,1,6)
        GOTO 999
180     CONTINUE
        IF (BADHFLG.EQ.1) GOTO 600
C
C------KEEP COUNT OF GOOD AND BAD DATA POINTS
C
        OBS=SAONUM(JSAT)
        NUMPTS=NUMPTS+1
        IF (CODE(1).NE.3) NUMGPTS=NUMGPTS+1
        CALL TRNPTS(LNDX)
C
c------build the standard binary output form.
C
```

```
      CALL stdfrm
C
C------ADVANCE OBSERVATION NUMBER.
C
      K=SAONUM(JSAT)+1
      IF(K.GE.30000)K=20001
      SAONUM(JSAT)=K
C
  999 CONTINUE
      RETURN
C
500   CONTINUE
        CALL REJREC(CARD)
        WRITE(11,502)
502     FORMAT(6X,'REJECT; WETTZELL OBS, NO CORRECTIONS')
        GOTO 999
      END

        SUBROUTINE NASA
C
C...REDUCE NASA QUICK LOOK LASER DATA TO  SAO FORMAT
C
C               FOR002.DAT      INPUT OBSERVATIONS DATA
C               FOR001.DAT      OUTPUT REDUCED OBSERVATIONS DATA STREAM
C               FOR011.DAT      OUTPUT REJECTED OBSERVATIONS, HEADER
C                               RECORDS AND COMMENTS. IF CORRECTED, CAN
C                               RERUN FOR011.DAT AS FOR002.DAT
C...THE OBSFLGN ARRAY FLAGS OBSERVATION DATA ERRORS. AN OBSERVATION
C...IS REJECTED IF ANY DATA ERROR OCCURS.
C               OBSFLGN(I) = .FALSE. IF THERE ARE NO DATA ERRORS
C                            .TRUE.  IF THERE ARE ANY DATA ERRORS
C               OBSFLGN(I)      MEANING
C       1       STATUS---PRIME LASER
C                       2       STATION
C                       3       SATELLITE
C                       4       YEAR
C                       5       DAYS PER YEAR
C                       6       DAY OF THE MONTH
C                       7       SECONDS PER DAY
C                       8       FRACTION OF A SECOND
C                       9       TWO WAY TIME RANGE
C                      10       RANGE
C                   11          BAD RECORD
C____________________________________________________________
C
        EXTERNAL ICONV,CHKCHR,RANCHK
        DOUBLE PRECISION FMJD,SECOND,FSEC,RANGE,RNGTIME,PRANGE
        CHARACTER*80 RCRD,PCRD*30
        CHARACTER*4 CSTA,NASSTA(25),SAOSTA(25),CND1*1,CYR*2
        CHARACTER*4 CSAT,NASSAT(25),SAOSAT(25)*7,CDAPYR*3,CSECPDA*5
        CHARACTER*6 CFSEC
        LOGICAL*1 LEAPYR,CHKCHR,FLAG(17),OBSFLGN(11),WNDFLG,PASSFLG
        LOGICAL *1 RANCHK
        REAL*4 NOISE
      INTEGER*2 JNASTA,STA,TDPY,YR,MONTH,DAY,HOUR,MIN,IDPMT(12,2)
      INTEGER*2 PASSEQ,TYPE,COL57,COL58,COL62,CNDX,DUM(6),OBSONE
      INTEGER*2 BADHFLG
      INTEGER*4 SAONUM,OBSNO,PRES
      COMMON/SEQ/PASSEQ,SAONUM(25),NASNUM(25),NNASSAT,NNASSTA,
     1        TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
      COMMON/NASAC/NASSAT,SAOSAT,NASSTA,SAOSTA
      COMMON/LIMIT/ISTA(25),PRLIM(25,2),TMPLIM(25,2),CALLIM(25,2,2),
     1          RANLIM(25,2),EXPDATE(25)
      COMMON/LASCOC/RCRD,PCRD
      DATA IDPMT/31,59,90,120,151,181,212,243,273,304,334,365,
     1        31,60,91,121,152,182,213,244,274,305,335,366/
      DATA REFCOR,TIMPRE,CENMAS,HUMID,ATEMP,ELEVANG,
      1    AZIMUTH,NOISE,BIAS,PRES/9*0.0,0/
      DATA PRANGE,CNDX/0.D00,0/
C_______________________________________________________
C...BEGIN A NASA PASS
      KOUNT=1
      OBSONE=0
      IRRFLG=0
C__________________________________________________________
C...OUTPUT HEADER RECORD
80    IF (NDETFLG.EQ.0) GOTO 83
      WRITE(11,81)
81    FORMAT('LASERQL')
      NDETFLG=0
      GOTO 85
83    CALL REJREC(RCRD)
C____________________________________________________
C...GET THE NEXT OBSERVATION RECORD
100   READ(2,101,END=38) RCRD
101   FORMAT(A)
85    J=INDEX(RCRD,'....')   ! DELETE A COMMENT
      IF (J.NE.0) GOTO 100
      J=INDEX(RCRD,'____')   ! SAVE SPECIAL COMMENT OVER MULTIPLE RERUNS
      IF (J.EQ.0) GOTO 103
      IRRFLG=IRRFLG+1
      WRITE(11,101) RCRD
      OBSONE=1
      GOTO 100
103   DO 102 J=1,11              ! ERROR FLAGS
102   OBSFLGN(J)=.FALSE.
C...IS THIS RECORD AN OBSERVATION?
      J=LASTC(RCRD,80)
      IF (J.EQ.0) GOTO 100       ! DELETE EMPTY RECORDS
      J=INDEX(RCRD,'END') ! END OF PASS?
      IF (J) 104,104,38
104   J=INDEX(RCRD,'1BB') ! FIND BEGINNING OF DATA ON OBS RECORD
      IF (J-1) 107,106,105
105   RCRD(1:81-J)=RCRD(J:80)
106     IF(RCRD(1:3).EQ.'1BB'.AND.RCRD(70:72).EQ.'4FF') THEN
      IF (OBSONE.EQ.1) GOTO 122
      IF (IRRFLG.GT.0) GOTO 122
      WRITE(11,121) RCRD(1:14)
```

```
121      FORMAT('____',A14)
         OBSONE=1
         GOTO 122
         ELSE
107      IF (OBSONE.EQ.1) GOTO 120
         WRITE(11,121) RCRD(1:14)
         OBSONE=1
120      OBSFLGN(11)=.TRUE.
         GOTO 2000       ! REJECT RECORD AND CONTINUE
         ENDIF
C_____________________________________________________________
C...PROCESS THE RECORD; STATUS FIELD
122      CND1=RCRD(15:15)
         IF(CND1.EQ.'0'.OR.CND1.EQ.'4') GOTO 200
         OBSFLGN(1)=.TRUE.
         GOTO 2000
C_____________________________________________________________
C...PROCESS THE STATION
200      CONTINUE
         if (RCRD(10:10).eq.' ') RCRD(10:10)='0'
         IF (CHKCHR(RCRD,10,13))  GOTO 210
         CSTA=RCRD(10:13)
         DO 220 J=1,NNASSTA
         IF (CSTA.EQ.NASSTA(J)) GOTO 230
220      CONTINUE
210      CONTINUE
         OBSFLGN(2)=.TRUE.
         GOTO 2000
230      CSTA=SAOSTA(J)
         STA=ICONV(CSTA,1,4)
C_____________________________________________________________
C...PROCESS THE SATELLITE
300      CONTINUE
         IF (RCRD(4:4).EQ.' ') RCRD(4:4)='0'
         IF (CHKCHR(RCRD,4,7)) GOTO 310
         CSAT=RCRD(4:7)
         DO 320 L=1,NNASSAT
         IF (CSAT.EQ.NASSAT(L)) GOTO 330
320      CONTINUE
310      CONTINUE
         OBSFLGN(3)=.TRUE.
         GOTO 2000
330      CONTINUE
         LNASAT=L
         IDENT=ICONV(SAOSAT(L),1,7)
C_____________________________________________________________
C...PROCESS THE YEAR
400      CONTINUE
         IF (CHKCHR(RCRD,24,25)) GOTO 410
         CYR=RCRD(24:25)
         YR=ICONV(RCRD,24,25)
         IF (YR.GT.76) GOTO 430
410      CONTINUE
         OBSFLGN(4)=.TRUE.
```

```
      GOTO 2000
430   CONTINUE
      TDPY=365
      LEAPYR=.FALSE.
      ITM=YR-76
      IF ((ITM/4)*4.EQ.ITM) TDPY=366
      IF (TDPY.EQ.366) LEAPYR=.TRUE.
C_____________________________________________________________
C...PROCESS DAYS PER YEAR INTO MONTH AND DAY OF MONTH
500   CONTINUE
      IF (CHKCHR(RCRD,26,28)) GOTO 510
      CDAPYR=RCRD(26:28)
      IDAPYR=ICONV(RCRD,26,28)
      IF (IDAPYR.GE.1.AND.IDAPYR.LE.TDPY)GOTO 530
510 CONTINUE
      OBSFLGN(5)=.TRUE.
      GOTO 2000
C...IS CURRENT YEAR A LEAP YEAR ?
530   CONTINUE
      J=1
      IF (LEAPYR) J=2
      DO 535 I=1,12
      IF (IDAPYR.LE.IDPMT(I,J)) GOTO 540
535   CONTINUE
      OBSFLGN(6)=.TRUE.
      GOTO 2000
540   DAY=IDAPYR
      MONTH=I
      IF (I.EQ.1) GOTO 600
      DAY=IDAPYR-IDPMT(I-1,J)
C_____________________________________________________________
C...PROCESS SECONDS PER DAY INTO HOURS, MINUTES, AND SECONDS
600   CONTINUE
      IF (CHKCHR(RCRD,29,33)) GOTO 610
      CSECPDA=RCRD(29:33)
      ISECPDA=ICONV(RCRD,29,33)
      IF(ISECPDA.GE.0.AND.ISECPDA.LE.86400) GOTO 630
610   CONTINUE
      OBSFLGN(7)=.TRUE.
      GOTO 2000
630   INTM=ISECPDA
      HOUR=INTM/3600
      IHR=HOUR
      INTM=INTM-3600*IHR
      MIN=INTM/60
      MINT=MIN
      ISEC=INTM-60*MINT
C_____________________________________________________________
700   CONTINUE
      CFSEC=RCRD(34:39)
      IF (.NOT.CHKCHR(RCRD,34,39)) GOTO 710
      OBSFLGN(8)=.TRUE.
      GOTO 2000
710   FSEC=DFLOTJ(ICONV(RCRD,34,39))/1.D06
```

```
        SECOND=DFLOTJ(ISEC)+FSEC
C_____________________________________________________________
C...COMPUTE MEAN JULIAN DAY, AND FRACTION THEREOF
800     CONTINUE
        INTM=(YR-73)/4
        MJD=41682+INTM*366+(YR-73-INTM)*365+IDAPYR
        FMJD=(DFLOTJ(ISECPDA)+FSEC)/86400.D00
C_____________________________________________________________
C...PROCESS RANGE (2-WAY TIME (NANOSEC) INTO 1-WAY DISTANCE (METERS)
900     CONTINUE
        IF (.NOT.CHKCHR(RCRD,55,66)) GOTO 910
        OBSFLGN(9)=.TRUE.
        GOTO 2000
910     RNGTIME=DFLOTJ(ICONV(RCRD,55,60))
        RNGTIME=RNGTIME*1.D06+DFLOTJ(ICONV(RCRD,61,66))
        RANGE=.299792458D00*RNGTIME/2.D00    ! METERS
        IF (RANCHK(RANGE,LNASAT,FLAG)) GOTO 950
        OBSFLGN(10)=.TRUE.
        GOTO 2000
950     RANGE=RANGE/1.D02    ! METERS
C_____________________________________________________________
C...PREPARE INTERNAL DATA STREAM
1000    CONTINUE
        OBSNO=NASNUM(LNASAT)
        COL57=4
        COL58=8
        CALSTAB=0
        COL62=0
        TYPE=8
        RANPRE=2
        TIMPRE=0.0002
        CNDX=1
C_____________________________________________________________
C...OUTPUT THE INTERNAL DATA STREAM, UNFORMATTED
        WRITE(1,ERR=2000) IDENT,OBSNO,STA,MJD,FMJD,TYPE,RANGE,
     1    REFCOR,TIMPRE,RANPRE,COL57,COL58,CALSTAB,COL62,CENMAS,
     2    PRES,HUMID,ATEMP,ELEVANG,PRANGE,AZIMUTH,CNDX,NOISE,
     3    BIAS,YR,MONTH,DAY,HOUR,MIN,SECOND,PASSEQ,DUM
C_____________________________________________________________
C...INITIALIZE FOR NEXT OBSERVATION
        KOUNT=KOUNT+1
        K=NASNUM(LNASAT)+1
        IF (K.GE.40000) K=30001
        IF (K.LE.30000) K=30001
        NASNUM(LNASAT)=K
       GOTO 100
C_____________________________________________________________
C...REJECT RECORD AND OUTPUT DATA ERROR COMMENT
2000    CONTINUE
        CALL REJREC(RCRD)
        CALL REJERN(OBSFLGN)
        KOUNT=KOUNT+1
        GOTO 100
C_____________________________________________________________
```

```
C...PASS COMPLETED, RETURN TO OBSCARD
38      IF (IRRFLG.GT.0) GOTO 40
        WRITE(11,37) KOUNT-1
37      FORMAT('____NUMBER OF OBSERVATIONS IN PASS',I3)
40      CALL REJREC(RCRD)
        K=PASSEQ+1
        IF (K.GE.30000) K=1
        PASSEQ=K
        RETURN
        END

        subroutine stdfrm
c
c*****************************************************************
c
c       collect the necessary values, change a few DATA types,
c       and write the standard SAO OBSERVATION RECORD, unformated,
C       TO THE data stream IN for001.dat file.
c
c*****************************************************************
c
        external mjdf
        LOGICAL*1 PASSFLG
        integer*4 ident,obsno,mjd,stat,obs,code,y,d,sec,pass,
     1  sat,temp,chksum,SAONUM,hr,
     2  tenmm,ihum,ipres,col53,col5455,
     3  ccal
        integer*2 type,col57,col58,col62,cndx,year,month,day,sta,
     1 hour,minute,passeq,paseqn,OBSONE,BADHFLG
        integer*2 dy(52)
        integer*2 i2zero
        integer*4 i4zero
        real*4 r4zero
        double precision r8zero
        real refcor,timpre,ranpre,calstab,cenmas,humid,tcent,
     2  elevang,azimuth,noise,bias
        double precision range,second,mjdfr,prange,orange
        logical*1 obsflag(17),WNDFLG
        common/seq/passeq,saonum(25),nasnum(25),nnassat,nnassta,
     1          TRNFLG,WNDFLG,OBSONE,BADHFLG,PASSFLG,NSEQ
        COMMON /LASER/ SAT(25),NUMSAT,PASS(25),SUM(25)
        COMMON/LASCOM/CODE(4),DECADE,DT,C,Y,D,M,HR,MIN
     1  ,SEC,IDENT,ILLCODE,IFIRST,JSTA,NOGOOD,OBS,PR,RANGE,RELHUM,
     1     STAT,TEMP,PRESS,HUM,TCENT,JSAT,TM,CAL,OBSFLAG
     2  ,ACOEFF,BCOEFF,PRECAL,POSTCAL,ALOGSS,DIFF,NDIFF,PPCAL
     3    ,NCHKSM1,NCHKSM2
        equivalence (orange,dy(1))
        equivalence (refcor,dy(5))
        equivalence (timpre,dy(7))
        equivalence (ranpre,dy(9))
        equivalence (col57,dy(11))
        equivalence (col58,dy(12))
        equivalence (calstab,dy(13))
        equivalence (col62,dy(15))
```

```
      equivalence (cenmas,dy(16))
      equivalence (pres,dy(18))
      equivalence (humid,dy(20))
      equivalence (atemp,dy(22))
      equivalence (elevang,dy(24))
      equivalence (prange,dy(26))
      equivalence(azimuth,dy(30))
      equivalence (cndx,dy(32))
      equivalence (noise,dy(33))
      equivalence (bias,dy(35))
      equivalence (year,dy(37))
      equivalence (month,dy(38))
      equivalence (day,dy(39))
      equivalence (hour,dy(40))
      equivalence (minute,dy(41))
      equivalence (second,dy(42))
      equivalence (paseqn,dy(46))
      data i2zero,i4zero,r4zero,r8zero/4*0/
      data refcor,timpre,ranpre,ctrmas/4*0/
      data elevang,azimuth,prange/3*0/
      data noise,bias/2*0.0/
C
      PASSFLG=.TRUE.
c     if (code(1).ne.3) obsno=-obsno
      sta=stat
      col58=8
      col57=0
      calstab=0
      col62=1
      ndx=mjdf(y,m,d,ismd)
      mjd=ndx*ismd
      atemp=tcent
      pres=pr
      humid=hum
      orange=range*1.d-2
      year=y
      month=m
      day=d
      hour=hr
      minute=min
      second=dflotj(sec)*1.d-6
      mjdfr=((second/60.+min)/60.+hr)/24.
      type=8
      cndx=0
      timpre=.0002
      ranpre=2.0
      paseqn=PASSEQ
      write(1,err=100)ident,obs,sta,mjd,mjdfr,type,dy
      goto 999
C
C     I/O ERROR IN WRITE TO DATA STREAM RECORD. ANALYSIS PROGRAM WILL
C     DETECT SUCH ERRORS AND DELETE THE OBSERVATION RECORD.
C
100   CALL ERRSNS(IFNUM,IA,IB,IC,ID)
```

```
        WRITE(11,101) IFNUM,IA,IB
101     FORMAT('....WRITE ERROR IN DATA STREAM OBSERVATION  RECORD',
        1       ' TO FOR001.DAT'/'.......FORTRAN I/O ERROR # ',I2/
        2       '.......RMS I/O ERROR COMPLETION STATUS, (HEX) ',Z8/
        3       '.......RMS I/O ERROR STATUS VALUE, (HEX) ',Z8)
999     return
        end

              SUBROUTINE TCODE(NUMPASS)
C
C     *****************************************************************
C
C     PREPARES THE LASERA RECORD FROM THE PASS HEADER DATA STORED IN
C     COMMON LASHED AND FROM THE POINT OBS DATA STORED IN COMMON LASOBS
C
C     OUTPUT_
C          BUFF = ARRAY CONTAINING THE LASERA VARIABLES AS EQUIVALENCED
C
C
C     *****************************************************************
C
C
      INTEGER*2  BUFF(64)
        INTEGER*2 STATION,STATIONX, SKYC, YEAR, DAY, MONTH,HOUR,MINUTE,STATUS
     1  ,TYPE,EPOCH_SYS,OBS_UNIT,CALIB_INDEX,SPEED_LIGHT
     1  ,YR,MO,DY,HR,MIN,L
        INTEGER*4 NSAT,NEXOBS,MJD,FILL(3),NSATX,NEXOBSX
        REAL*4 RFRCORR,TIME_PREC,RANGE_PREC,CALIB_STABIL,CEN_MASS
     1  ,PRESSURE,HUMIDITY,TEMP,ELEV_ANGLE,AZIMUTH,RANGE_NOISE,RANGE_BIAS
     2  ,HUM,PRESS,TMP,RFRCORRX,WT,COMCORR,ELEV,AZIM
        REAL*8 FRAC_DAY,RANGE,PREDIC_RANGE,SECOND,RANGEX,PRERANG,SEC
        LOGICAL RNGCM
        EQUIVALENCE (NSAT,BUFF(1))
        EQUIVALENCE (NEXOBS,BUFF(3))
        EQUIVALENCE (STATION,BUFF(5))
        EQUIVALENCE (MJD,BUFF(6))
        EQUIVALENCE (FRAC_DAY,BUFF(8))
        EQUIVALENCE (TYPE,BUFF(12))
        EQUIVALENCE (RANGE,BUFF(13))
        EQUIVALENCE (RFRCORR,BUFF(17))
        EQUIVALENCE (TIME_PREC,BUFF(19))
        EQUIVALENCE (RANGE_PREC,BUFF(21))
        EQUIVALENCE (EPOCH_SYS,BUFF(23))
        EQUIVALENCE (OBS_UNIT,BUFF(24))
        EQUIVALENCE (CALIB_STABIL,BUFF(25))
        EQUIVALENCE (CALIB_INDEX,BUFF(27))
        EQUIVALENCE (CEN_MASS,BUFF(28))
        EQUIVALENCE (PRESSURE,BUFF(30))
        EQUIVALENCE (HUMIDITY,BUFF(32))
        EQUIVALENCE (TEMP,BUFF(34))
        EQUIVALENCE (ELEV_ANGLE,BUFF(36))
        EQUIVALENCE (PREDIC_RANGE,BUFF(38))
        EQUIVALENCE (AZIMUTH,BUFF(42))
        EQUIVALENCE (SPEED_LIGHT,BUFF(44))
        EQUIVALENCE (RANGE_NOISE,BUFF(45))
        EQUIVALENCE (RANGE_BIAS,BUFF(47))
        EQUIVALENCE (YEAR,BUFF(49))
        EQUIVALENCE (MONTH,BUFF(50))
        EQUIVALENCE (DAY,BUFF(51))
        EQUIVALENCE (HOUR,BUFF(52))
        EQUIVALENCE (MINUTE,BUFF(53))
        EQUIVALENCE (SECOND,BUFF(54))
        EQUIVALENCE (STATUS,BUFF(58))
        EQUIVALENCE (FILL(1),BUFF(59))
      COMMON /LASHED/ STATIONX,DIST,YR,MO,DY,NSATX,SKYC,HUM,TMP,
     $                PRESS,PRECAL,POSTCAL,PPCAL,DIFF,NDIFF,ACOEFF,
     $ BCOEFF,REFAREA
      COMMON /LASOBS/ NEXOBSX,HR,MIN,SEC,AZIM,ELEV,PRERANG,RANGEX,WT,L,
     $                RFRCORRX,COMCORR
       COMMON /UNITS/ C,RNGCM
        FILL(1)=0
        FILL(2)=0
        FILL(3)=0
C
C
C
C
C  SET UP EQUIVALENCED VARIABLES--LASHED
        STATION=STATIONX
        YEAR=YR
        MONTH=MO
        DAY=DY
        HUMIDITY=HUM
        TEMP=TMP
        PRESSURE=PRESS
        NSAT=NSATX
C
C
C
C
C  SET UP EQUIVALENCED VARIABLES--LASOBS
        NEXOBS=NEXOBSX
        HOUR=HR
        MINUTE=MIN
        SECOND=SEC
        AZIMUTH=AZIM
        ELEV_ANGLE=ELEV
        PREDIC_RANGE=PRERANG
        RANGE=RANGEX/10.00
        RANGE_PREC=WT
        CALIB_INDEX=L
        RFRCORR=RFRCORRX
        CEN_MASS=COMCORR
        RANGE_NOISE=0.0
        RANGE_BIAS=0.0
        STATUS=NUMPASS
        FRAC_DAY=(DFLOAT(HOUR)+(DFLOAT(MINUTE)+SECOND/60.D00)/60.D00)/24.D00
        II=(MJDF(YEAR,MONTH,DAY,MJD))
```

```
C
C
C
      TIME_PREC=.0001
C
        TYPE=8
        EPOCH_SYS=4
      OBS_UNIT = 4
      IF( RNGCM ) OBS_UNIT = 7
      CALIB_STABIL=ABS(DIFF)
C
C
C
C
C
C
C
        WRITE(7) BUFF
C
C
      RETURN
      END

        PROGRAM NASABIN
C
C       THIS PROGRAM INVOKES SUBROUTINE GETREC WHICH READS SUCCESSIVE
C       RANGE RECORDS FROM TAPE IN NASA BINARY FORMAT; 9-TO-9 HAS BEEN
C       USED TO PRODUCE A VAX READABLE TAPE FROM THE ORIGINAL
C       IBM TAPE.  THE VARIOUS
C       QUANTITIES ARE UNSCRAMBLED AND CONVERTED FROM IBM TO DEC VAX
C       FORMAT, THEN WRITTEN TO TAPE IN SAO FORMAT WITH SUBROUTINE PUTREC.
C
C       INPUT: UNIT FOR001   OUTPUT: FOR002
C
        BYTE RECORD(68),B4(4),B5(4),B6(4)
        INTEGER*2 MDRTK, MDRAP, MDRRH, TIMESYS
        INTEGER*2 IVSHORT
        REAL*8 GMT, OBSVAL, AIBM, C_OLD
        EQUIVALENCE (IB4,B4(4)), (IB5,B5(4)),(IB6,B6(4))
        DATA C_OLD/2.997925D8/  !METERS/SEC. NASA USES OLD S.O.L.
C
        NREC = 0
        NBAD = 0
        DO 11 I=1,4
        B4(I)='00'X
        B5(I)='00'X
11      B6(I)='00'X
1       GO TO (100, 200, 300, 400) IGETREC(RECORD)
C
C       SUCCESSFUL TAPE READ:
100     NREC = NREC + 1
        IDSAT =  IVINT(RECORD(1))
        TIMESYS = IVSHORT(RECORD(7))
        NUMSTAT = IVINT(RECORD(9))
      MJD = IVINT(RECORD(17))
      GMT = AIBM(RECORD(21))
      OBSVAL = AIBM(RECORD(29))
C
C     NASA DATA GIVES EPOCH AS TIME AT SATELLITE, SAO AS
C     TRANSMISSION TIME.  CONVERT:
      GMT = GMT - (OBSVAL/C_OLD )/86400.D0
      IF (GMT.LT.0.D0) THEN
              GMT = GMT + 1.D0
              MJD = MJD - 1
      ENDIF
      TROPREF = SIBM(RECORD(49))
C
C     NASA SIGN CONVENTION ON CENTER OF MASS CORRECTION APPEARS
C     TO BE OPPOSITE SAO, SO WE CHANGE THE SIGN.
      CMASS = -SIBM(RECORD(65))
C
C     NOW WE UNSCRAMBLE THE METEOROLOGICAL DATA REPORT WHICH IS PACKED
C     IN BYTES 53-56. NOTE THAT DIVIDING (MULTIPLYING) BY 2**N SHIFTS
C     A BYTE RIGHT (LEFT) BY N BITS.
C     BITS 1-7 OF BYTE53 GIVE THE RELATIVE HUMIDITY; WE ARE TOLD
C     BIT 0 IS ALWAYS 0, AND WE SO ASSUME.
C
      MDRRH = RECORD(53)
C
C     MDRTK CONSISTS OF BYTE 54 PLUS FIRST HALF OF BYTE 55;
C     MDRAP IS SECOND HALF OF BYTE 55 PLUS BYTE 56.
C
C     PUT BYTES 54,55,56 OF RECORD IN BYTE ARRAYS EQUIVALENCED
C     TO IB4,IB5,IB6:
C
      B4(4)=RECORD(54)
      B5(4)=RECORD(55)
      B6(4)=RECORD(56)
C
C     IMAGINE THE 4 HALF-BYTES IN BYTES 54 AND 55 LABELLED A,B,
C     C,D IN ASCENDING ORDER. WE NOW ISOLATE THESE:
C
      IC = IB5/16
      IA = IB4/16
      IB = IB4 - 16*IA
      ID = IB5 - 16*IC
C
C     NOW CONSTRUCT THE PRESSURE AND TEMPERATURE:
C
      MDRTK = IC + 16*(IB+16*IA)
      MDRAP = IB6 + 256*ID
      CALL PUTREC(IDSAT, TIMESYS, NUMSTAT, MJD, GMT, OBSVAL,
     1  TROPREF, MDRRH, MDRTK, MDRAP, CMASS)
C
C     RECYCLE TO READ A NEW LOGICAL RECORD:
      GO TO 1
C
C     END OF TAPE SENSED; NORMAL TERMINATION:
```

```
C
 200   WRITE(6,201) NREC
 201   FORMAT('0   END OF TAPE SENSED. NORMAL TERMINATION AFTER '
     1 , 'READING', I6, ' LOGICAL RECORDS.')
        STOP
C
C       UNRECOGNIZABLE RECORD READ:
C
300     WRITE(6,301) NREC, (RECORD(I), I = 1, 68)
301     FORMAT('0  RECORD READ IS NEITHER LASER NOR ANGLE DATA:'/
     1  10X, 'NREC=',I7/ 10X, 9(1X,4Z2)/ 10X, 8(1X,4Z2))
        NBAD = NBAD+1
        IF (NBAD.LT.20) GOTO 1
        STOP
C
C       ABNORMAL STATUS RETURNED FROM TAPE READ. MESSAGE WRITTEN BY GETREC.
400     STOP
        END !NASABIN

        INTEGER FUNCTION IGETREC(RECORD)
C
C       IGETREC READS A TAPE CONTAINING DATA IN IBM/NASA BINARY FORMAT
C       (TRANSLATED TO VAX READABLE FORM BY 9-TO-9).  SUCCESSIVE CALLS
C       RETURN SUCCESSIVE 68-BYTE LOGICAL RECORDS OF LASER RANGING
C       DATA IN BYTE ARRAY "RECORD"; LOGICAL RECORDS OF ANGLE (POINTING)
C       DATA ARE IGNORED.  THE VALUE OF IGETREC ON RETURN INDICATES:
C         1     SUCCESSFUL RETURN
C         2     END OF TAPE
C         3     UNRECOGNIZABLE RECORD (NEITHER RANGING NOR ANGLE).
C               THE RECORD IS RETURNED IN "RECORD."
C         4     ABNORMAL STATUS ON RETURN FROM READ. MESSAGE WRITTEN.
C
        BYTE RECORD(68), BBUF(8196)
        LOGICAL EOF, FIRST
        INTEGER BUFCOUNT
        DATA FIRST/.TRUE./, LUN/1/
C
C       ON FIRST SUBROUTINE CALL WE MUST READ A RECORD
C
        IF (FIRST) THEN
                FIRST= .FALSE.
                NEOF = 0
100             CALL BREAD(LUN, ISTAT, NBYTES, BBUF, EOF) !READ A RECORD
                IF (EOF) THEN  !WE HAVE REACHED END OF TAPE
                        IGETREC = 2
                        RETURN
                ENDIF
                IF (ISTAT.NE.'1'X .AND. ISTAT.NE.'870'X .AND.
        1       ISTAT.NE.'878'X) THEN
                        WRITE(6,150) ISTAT     ! ABNORMAL ISTAT FROAD
150                     FORMAT('0*** ABNORMAL STATUS RETURNED FROM',
        1       ' TAPE READ ATTEMPT:', Z10, ' (HEX)'/)
                        IGETREC = 4
                        RETURN
                ENDIF
                IF (NBYTES.GE.76) THEN   !IBM FILE MARK LT 76 BYTES
                        BUFCOUNT = 5 !COUNTER TO FST BYTE OF LR1
                ELSE
                        NEOF = NEOF + 1  !WE HAVE READ FILE MARK
                        IF (NEOF.LT.2) THEN
                                GO TO 100 !ONE FM SO TRY AGAIN
                        ELSE
                                IGETREC = 2  !2 FILE MARKS = EOT
                                RETURN
                        ENDIF
        ENDIF
        ENDIF
C
C       TEST TO SEE IF WE HAVE USED UP LAST RECORD READ; IF SO, GET ANOTHER
C
200     IF (BUFCOUNT.GT.NBYTES) THEN
                NEOF = 0
300             CALL BREAD(LUN, ISTAT, NBYTES, BBUF, EOF) !READ A RECORD
                IF (EOF) THEN  !WE HAVE REACHED END OF TAPE
                        IGETREC = 2
                        RETURN
                ENDIF
                IF (ISTAT.NE.'1'X .AND. ISTAT.NE.'870'X .AND.
        1       ISTAT.NE.'878'X) THEN
                        WRITE(6,150) ISTAT     ! ABNORMAL ISTAT FROAD
                        IGETREC = 4
                        RETURN
                ENDIF
                IF (NBYTES.GE.76) THEN   !IBM FILE MARK LT 76 BYTES
                        BUFCOUNT = 5 !COUNTER TO FST BYTE OF LR1
                ELSE
                        NEOF = NEOF + 1  !WE HAVE READ FILE MARK
                        IF (NEOF.LT.2) THEN
                                GO TO 300 !ONE FM SO TRY AGAIN
                        ELSE
                       IGETREC = 2  !2 FILE MARKS = EOT
                           RETURN
                        ENDIF
                ENDIF
        ENDIF
C
C       WE NOW TEST TO SEE IF THE LOGICAL RECORD POINTED TO BY BUFCOUNT
C       IS A LASER RECORD, ANGLE RECORD OR UNRECOGNIZABLE
C
        IF (BBUF(BUFCOUNT+9).EQ.'46'X) THEN  !TEXT FOR ANGLE
                BUFCOUNT = BUFCOUNT + 72        !IF ANGLE, LOOK AT NEXT REC.
                GO TO 200
        ENDIF
C
C       IF WE HAVE A LASER RECORD,SET IGETREC = 1, OTHERWISE = 3;
C       IN EITHER CASE, WE WILL FILL THE OUTPUT ARRAY 'RECORD'
C
        IF (BBUF(BUFCOUNT+9).EQ.'14'X) THEN
```

```
                IGETREC = 1
        ELSE
                IGETREC = 3
        ENDIF
        DO 400 I = 1, 68     !FILL OUTPUT ARRAY; IGNORE FIRST 4 BYTES
400     RECORD(I) = BBUF(BUFCOUNT+3+I)
        BUFCOUNT = BUFCOUNT +72   !INSURE WE ARE AT RIGHT PLACE ON REENTRY
        RETURN
        END

        SUBROUTINE PUTREC(IDSAT, TIMESYS, NUMSTAT, MJD, GMT, OBSVAL,
     1 TROPREF, MDRRH, MDRTK, MDRAP, CMASS)
C
C       PUTREC ACCEPTS THE VARIOUS QUANTITIES ASSOCIATED WITH A
C       NASA LASER RANGING RECORD, CONVERTS THEM TO FORMATS
C       CONSISTENT WITH SAO BINARY FORMAT WHERE NECESSARY,
C       COMPUTES ASSOCIATED QUANTITIES AND PACKS THE RESULTS INTO
C       BYTE ARRAY "BY," WHICH IS THEN WRITTEN (UNFORMATTEDLY)
C       TO UNIT FOR002.
C
        INTEGER*2 MDRTK, MDRAP, MDRRH, TIMESYS
        REAL*8 GMT, OBSVAL
        BYTE BY(128)
        INTEGER*2 STATNO,YY, MM, DD, HH, MIN, CALIND, SOL
        INTEGER*2 I8, I5, I2
        REAL*8 GMTP, RANGE, PRERNG, SEC
        LOGICAL FIRST
        DATA FIRST/.TRUE./, LUOUT/2/
        EQUIVALENCE (IDSATP,BY(1)), (NSEQ, BY(5)), (STATNO,BY(9)),
      1 (MJDP,BY(11)), (GMTP,BY(15)), (I8,BY(23)), (RANGE,BY(25)),
      2 (TROPREFP,BY(33)), (TIMEPREC,BY(37)), (RANGPREC,BY(41)),
      3 (I5,BY(45)), (I2,BY(47)), (CALSTAB,BY(49)),
      4 (CALIND,BY(53)), (CMASSP,BY(55)), (ATPRES,BY(59)),
      5 (RELHUM,BY(63)), (ATTEMP,BY(67)), (PREEL,BY(71)),
      6 (PRERNG,BY(75)), (PREAZ,BY(83)), (SOL,BY(87)),
      7 (RNGNOIS,BY(89)), (RNGBIAS,BY(93)), (YY,BY(97)),
      8 (MM,BY(99)), (DD,BY(101)), (HH,BY(103)),
      9 (MIN,BY(105)), (SEC,BY(107))
C
C       ON FIRST ENTRY WE SET THE BYTES WHICH WILL NOT CHANGE.
C       ALSO, SET INITIAL SEQUENCE NUMBER.
C
        IF(FIRST) THEN
        FIRST = .FALSE.
        NSEQ = 0
        I8 = 8           !LASER OBSERVATION
        TIMEPREC = 1.E-6         !NASA FIGURE
        RANGPREC = 0.1   !NASA
        I5 = 5           !INDEX
        2 = 2            !INDEX
        CALSTAB = 0.0    !NO CALIBRATION INFORMATION
        CALIND = 0       !"
        PREEL = 0.0      !NO PREDICTIONS
        PRERNG = 0.D0    !"
      PREAZ = 0.0     !"
      SOL =00 !OLD SPEED OF LIGHT WAS USED
      RNGNOIS = .070710677    !SET TO GIVE NASA PRECISION OF
      RNGBIAS = .070710677    !0.1/SQRT(2)
      DO 20 I = 115,128       !PASS SEQUENCE NUMBER AND
20    BY(I) = '00'X           !SPARE BYTES SET TO 0
      ENDIF
C
C     NOW WE SET THE VARIABLE QUANTITIES
C
      IDSATP = IDSAT
      NSEQ = NSEQ + 1 !ARBITRARY OBSERVATION NUMBER
      IF (NSEQ.GT.69999) NSEQ = 1     !NSEQ TOO HIGH
      STATNO = NUMSTAT
      MJDP = MJD
      GMTP = GMT
      RANGE = OBSVAL
      TROPREFP= TROPREF       !TROPOSPHER REFRACTION CORR
      CMASSP = CMASS          !CENTER OF MASS CORR
      ATPRES = FLOAT(MDRAP)
      RELHUM = FLOAT(MDRRH)/100.   !PERCENT TO FRACTION
      ATTEMP = FLOAT(MDRTK)-273.2  !KELVIN TO CELSIUS
      CALL INVSAO(MJD, IY,IM,ID)   !FULL IN MONTH, DAY, YEAR
      YY = IY
      MM = IM
      DD = ID
      SGMT = 24*GMT           !HOURS, MINUTES, SECONDS.  CODE LOOKS
      HH = IIFIX(SGMT)        !ODD BECAUSE FIX ONLY TAKES REAL*4 ARGS.
      SGMT = 60*(24*GMT-HH)
      MIN = IIFIX(SGMT)0034   SEC = 60*(60*(24*GMT-HH)-MIN)
C
C     FINISHED!  WRITE TO TAPE AND RETURN
C
      WRITE(LUOUT) BY
      RETURN
      END

      REAL*8 FUNCTION AIBM(BY)

      IMPLICIT REAL*8 (A-H,O-Z)
      BYTE BY(8)
      INTEGER*4 IT(8)



      AIBM=0.D0
      DO 101 I=1,8
      IT(I)=BY(I)
      IF(IT(I).LT.0)IT(I)=IT(I)+256
  101 CONTINUE
      IF(IT(1).GT.'7F'X)THEN
              SIGN=-1.D0
              IT(1)=IT(1)-'80'X
      ELSE
```

```
                 SIGN=+1.D0
         ENDIF
         DO 102 I=2,8
  102    AIBM=256.D0*AIBM+DFLOAT(IT(I))
C        FIXUP FOR CASE OF INPUT IBM F.P. ZERO'
         IF (AIBM.EQ.0.D0) RETURN
         AIBM=SIGN*AIBM*16.D0**(IT(1)-'4E'X)
         RETURN
         END     ! AIBM

         REAL*4 FUNCTION SIBM(BY)

         BYTE BY(4), BYT(8)
         REAL*8 AIBM
         DO 100 I =1, 4
         BYT(I) = BY(I)
100      BYT(I+4) = '00'X
         SIBM = AIBM(BYT)
         RETURN
         END  !SIBM

         subroutine invsao(isao,iy,mo,iday)
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         dimension month(12)
         logical leap
         data month/0,31,59,90,120,151,181,212,243,273,304,334/
         idy(iy)=iy*365+iy/4-iy/100+iy/400
         is=isao+678576
         if(is .le. 0) return
         iy=ifix(float(is)/365.2424)+1            !in the form 19xx
         leap=mod(iy,4) .eq. 0 .and. (mod(iy,100).ne. 0 .or.
     1   mod(iy,400) .eq. 0)                !is this a leap yr?
         id=is-idy(iy-1)
         if(id .gt . 0)go to 4                    !valid day number.
         iy=iy-1                                  !bad day; fix it
         id=is-idy(iy-1)
         go to 6
4        if(id .le. 365 .or. (leap .and. (id .le. 366)))go to 6
         iy=iy+1
         id=is-idy(iy-1)
6        do 7 i=1,12
7        if(month(i) .LT. id)mo=i
         iday=id-month(mo)
         if(.not. leap .or. (leap .and. mo .le . 2))go to 50
         iday=iday-1
         if(iday .gt. 0) go to 50
         mo=mo-1                                  !back up one month.
         iday=id-1-month(mo)
         if(id .eq. 60)iday=29
50       iy=iy-1900
100      return
         end

C  LASERG PROGRAM FOR FINAL DATA.
```

```
C
C**** NOTE:     THIS IS NOT AN OPERATIONAL VERSION.  CODE NOT RELEVANT
C****           TO IN PUT/OUTPUT FORMATS AND DATA TRANSFORMATIONS HAS
C****           BEEN LARGELY OMITTED (INDICATED BY  . . . ).
C                        G. GULLAHORN   1 OCT 81
C
C  THIS PROGRAM TAKES THE INTERNAL FORMAT FILE CREATED BY Q,FLAC,LASERT,
C  etc. AND PRODUCES TWO OUTPUT FILES.  THE FIRST,CALLED SAODATA.DAT
C  IS IN THE 'STANDARD' SAO FORMAT; THE SECOND, CALLED GODDARD.DAT,
C  IS THE STANDARD NASA FORMAT.
C
C  G. GULLAHORN,  MAR 1980:
C  SAODATA.DAT IS NOW CALLED SO<sat mnem><mo><y>.DAT
C  GODDARD.DAT IS NOW CALLED GO<sat mnem><mo><yr>.DAT, E.G.
C  SOG31280.DAT FOR GEOS3 DATA DECEMBER 1980
C
 . . .
C
        DATA TIMING/0.,.003,.002,.005,.02,.05,.2,.5,2.0/
        DATA TCODE/1,2,3,4,5,6,7,8,9/
        INTEGER*4 SATID,OBSNO,NANORNG
        INTEGER*2 STAT,OBSTYPE,EPCHSYST,OBSUNITS,CALINDX,CINDX
        INTEGER*2 YR,MO,DAY,HR,MIN,TCODE(9),TIMCODE,TMICRO,SIGMA
        INTEGER*2 TSYSIND
        CHARACTER*17 INPUT
        REAL*4 REFCORR,TIMEPRE,RNGPRE,CALSTAB,CMCORR,PRESS,HUM
        REAL*4 TEMP,PELEV,PAZ,RNGNOISE,RNGBIAS,TIMING(9)
        REAL*8 OBSRNG,PRANGE,SEC,C,CONVRNG
        BYTE BARRAY(128)
C
        EQUIVALENCE(SATID,BARRAY(1)),(OBSNO,BARRAY(5)),(STAT,BARRAY(9))
     $  ,(OBSTYPE,BARRAY(23)),(OBSRNG,BARRAY(25)),(REFCORR,BARRAY(33)),
     $  (TIMEPRE,BARRAY(37)),(RNGPRE,BARRAY(41)),(EPCHSYST,BARRAY(45)),
     $  (OBSUNITS,BARRAY(47)),(CALSTAB,BARRAY(49)),(CALINDX,BARRAY(53)),
     $  (CMCORR,BARRAY(55)),(PRESS,BARRAY(59)),(HUM,BARRAY(63)),
     $  (TEMP,BARRAY(67)),(PELEV,BARRAY(71)),(PRANGE,BARRAY(75)),
     $  (PAZ,BARRAY(83)),(CINDX,BARRAY(87)),(RNGNOISE,BARRAY(89)),
     $  (RNGBIAS,BARRAY(93)),(YR,BARRAY(97)),(MO,BARRAY(99)),
     $  (DAY,BARRAY(101)),(HR,BARRAY(103)),(MIN,BARRAY(105)),
     $  (SEC,BARRAY(107))
C
 . . .
C
        TYPE *,' ENTER NAME OF INPUT FILE(USE QUOTES)'
        ACCEPT *,INPUT
        OPEN(UNIT=20,NAME=INPUT,TYPE='OLD',FORM='UNFORMATTED')
C
 . . .
C**** (FILE NAMES "SFILE" "GFILE" ARE GENERATED IN OMITTED CODE.)
 . . .
C  OPEN OUTPUT FILES,FORMERLY SAODATA.DAT, GODDARD.DAT
        OPEN(UNIT=21,NAME=SFILE,TYPE='NEW',RECORDSIZE=108)
        OPEN(UNIT=22,NAME=GFILE,TYPE='NEW',RECORDSIZE=90)
C
```

```
 . . .
C
C  SET UP CONSTANTS USED FOR GODDARD FORMAT.
      IZERO=0
      IONE=1
      C=299792.458D+00
      MEASTYPE=20
      TSYSIND=23
      ICOL=151
      CONVRNG=(C/(2.0D+08))*(1.0D+04)
C
C  MAJOR LOOP ::  READ AN OBSERVATION FROM INPUT FILE:::
C
10    READ(20,END=200)BARRAY
C
 . . .
C
      IREFCORR=NINT(REFCORR*100.)
      DO 50 I=1,8
      IF(TIMEPRE.GE.TIMING(I).AND.TIMEPRE.LE.TIMING(I+1))THEN
         TIMCODE=TCODE(I)
         GO TO 60
      END IF
50    CONTINUE
      TIMCODE=9
60    IRNGNOISE=NINT(25*(LOG10(RNGNOISE)+3))
      IRNGBIAS=NINT(25*(LOG10(RNGBIAS)+3))
      SIGMA=NINT(25*(LOG10(RNGPRE)+3))
      ISEC=NINT(SEC*10000000.)
      NANORNG=NINT(OBSRNG*10.)
      ICALSTAB=NINT(ABS(CALSTAB*10.))
      ICMCORR=NINT(CMCORR*100.)
      IPRESS=NINT(PRESS)
C     ONE-SHOT PROCESSING OF GENE & ALAN'S JAN/FEB 79 DATA REQUIRES
C     FOLLOWING PATCH TEMPORARILY:
      IF(HUM .LT. 1.0)HUM=HUM*100.
      IHUM=NINT(HUM)
      ITEMP=NINT(TEMP*10.)
      IPELEV=NINT(PELEV*1000.)
      IPRANGE=NINT(PRANGE*10.)
      IPAZ=NINT(PAZ*1000.)
C
C  WRITE A RECORD IN SAO FORMAT.
      WRITE(21,1000)SATID,OBSNO,STAT,YR,MO,DAY,HR,MIN,ISEC,
     $ NANORNG,IREFCORR,TIMCODE,SIGMA,OBSTYPE,EPCHSYST,OBSUNITS,
     $ ICALSTAB,CALINDX,ICMCORR,IPRESS,IHUM,ITEMP,IPELEV,IPRANGE,
     $ IPAZ,CINDX,IRNGNOISE,IRNGBIAS
1000  FORMAT(I7,2(I5),5(I2),I9.9,I10,2X,I4,I1,I2,3(I1),I3.3,I1,2(I4),
     $ I2,I4,I5,I11,1X,I7,1X,3(I2))
C
C  NOW SET UP FOR GODDARD OUTPUT FORMAT.
C  REFER TO THE GODDARD NOTES FOR A DESCRIPTION OF THE INFORMATION
C  IN EACH COLUMN.
      CALL FINDDAY(IYD,YR,MO,DAY)
```

```
        IGSEC=HR*3600.+MIN*60.+SEC
        J=SEC
        IFRSEC=NINT((SEC-J)*1000000.)
        IRNG=NINT(OBSRNG*CONVRNG)
        KTEMP=NINT(TEMP+273.15)
        MRNGBIAS=NINT(RNGBIAS*1000.)
        IGREFCORR=NINT(REFCORR*1000.)
        IGCMCORR=NINT(CMCORR*1000.)
        IGTIMPRE=LOG10((TIMEPRE+.0000005)*10**6)
C
C  WRITE A RECORD IN GODDARD FORMAT.
        WRITE(22,1001)SATID,MEASTYPE,TSYSIND,STAT,YR,IYD,IGSEC,
     $ IFRSEC,ICOL,IRNG,IZERO,IPRESS,KTEMP,IHUM,MRNGBIAS,
     $  IGREFCORR,IONE,IONE,IGCMCORR,IGTIMPRE
1001    FORMAT(I7,2(I2),I5,I2,I3,I5,I6.6,I3,5X,I10,4X,I1,1X,I4,2(I3)
     $  ,2X,I5,2X,I5,2(I1),I6,I2)
        GO TO 10
C
C       TERMINATE
C
 200    CLOSE(UNIT=20,DISP='SAVE')
        CLOSE(UNIT=21,DISP='SAVE')
        CLOSE(UNIT=22,DISP='SAVE')
  . . .
        END

        SUBROUTINE FINDDAY(GDDAY,YEAR,OMNTH,DAY)
C
C  THIS WAS ADAPTED FROM GENE CAMPBELL ROUTINE CALLED SMTHDAY
C  WHICH WAS USED BY HIS 6400 LASERG PROGRAM.
C  ALL I WANT TO DO WITH THIS ROUTINE IS RETURN THE DAY OF THE
C  YEAR(e.g. 93,281,365,etc.) FOR THE GODDARD FORMAT.
C
        INTEGER*2 YEAR,OMNTH,DAY,MO,J
        INTEGER GDDAY
        DIMENSION DZ(10)
        DATA DZ/334.,304.,273.,243.,212.,181.,151.,120.,90.,59./
C
        Y =YEAR
        M=OMNTH+.8
        D=DAY
        X=365.25*Y+.9
        J=X
        A=J
        MO=M-2
C
        IF(MO.GT.0)THEN
                IF((X-A-.8).LE.0.0)THEN
                    C=0.
0               ELSE
                C=1.
                ENDIF
                J=13-M
                E=DZ(J)+C
```

```
      ELSE IF(MO.EQ.0)THEN
      E=31.
      ELSE IF(MO.LT.0)THEN
      E=0.
      END IF
      GDDAY1=15018.+A+D+E
      GDDAY2=15018.+A
      GDDAY=GDDAY1-GDDAY2
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
C
C LUNAR PERTURBATION OVERLAY
C INCORPORATED GRIPE LUNAR PERTURBATION INTO MINI  ON 23 MARCH 1981
C COMPLETED WORKING VERSION ON APRIL 1, 1981.  THIS VERSION
C USES THE LUNI-SOLAR PERTURBATION THEORY DEVELOPED BY
C Y. KOZAI AS REPORTED IN SMITHSONIAN SPECIAL REPORT 349.  ALL SOURCE
C CODES ARE ON FLPPS DISK ONE.  THE SUBROUTINES USED IN THIS OVERLAY
C ARE:
C
C RDZON---LOADS REGISTERS WITH ZONAL HARMONICS
C INST---CALCULATES INSTANTANEOUS ELEMENTS AT EPOCH OF OBSERVATION
C GETSMA---CALCULATES SEMI-MAJOR AXIS
C NFINC---CALCULATES INCLINATION FUNCTION
C FACCAL---COMPUTES FACTORIALS
C HANSEN---COMPUTES ECCENTRICITY FUNCTION
C LUNAR---CALCULATES LONG PERIOD AND SHORT PERIOD LUNI-SLAR PRTS
C SETUP---ASSIGN VARIABLES FOR INTEGRATION
C KIND---ROUTINE TO INTEGRATE TERMS FOR LUNAR PERTS
C SUNVECT---CALCULATES VECTOR TO SUN
C LUNVECT---CALCULATES VECTOR TO MOON
C PRECESS---CALCULATES TERMS DUE TO PRECESSION
C EVA---CO MPUTES SIN AND COS TERMS FOR ECC AND MA
C
C THE FUNCTIONS USED IN THIS OVERLAY ARE
C
C CONSOC---STORES THE CONSTANTS USED FOR OVERLAY
C ERIQA---SOLVES KEPLER'S EQUATION
C LOAD---TAKES LOWER ORDER 4 BYTES FROM A REAL*8 AND PUTS
C        THEM INTO AN INTEGER*4
C PUT---PUTS INTEGER*4 INTO BITS 0-3 OF REAL*8 WORD.
C ASIN---FINDS ARCSIN
C ATANG---FINDS ARCTAN
C
C THE LIBRARIES USED IN THIS OVERLAY ARE
C
C STDLUNLIB---CONTAINS BINARY FOR ALL OF ABOVE
C WFWLTUTIL---SYSTEM UTILITIES
C WFWRUN---FORTRAN UTILITIES
C WFWSOS---SYSTEM UTILITIES
C
      DIMENSION PM(8,12),TEMP(22),T(2),IP(6),TLUN(2)
      DIMENSION NAMST(10),ISTP(3),IANOM(23),ERLAT(5)
      COMMON ORB(300)
      COMMON/IOBLK/NXT,IN,IOUT
C
A    .EXTN;.DSI,.STTY,LTODC,LT1DC,.LEXD

      CALL READY;;LOAD POWER FAIL CODE
      CALL FOPEN(6,"$LEX")
      CALL FOPEN(10,"$TTO")
      WRITE(6,1)
    1 FORMAT(" COMPUTING COEFFICIENTS FOR LUNAR PERTURBATIONS")
```

```
      WRITE(6,11)
  11  FORMAT("S.D.V. VERSION 6.3B 9/9/81")
C
C INITIALIZE THE I/O FILES
C
      CALL RUBOUT("SCRAT0")
      CALL FOPEN(1,"SCRAT1") ;INPUT FILE
      CALL FOPEN(0,"SCRAT0,0") ;OUTPUT FILE
      IN=1
      IOUT=0
      JJ=1
      J=0
      ISTP(1)=2H99
      ISTP(2)=2H90
      CALL PUTC(ISTP(3),1,0)
      CALL PUTC(ISTP(3),2,0)
C
C  READ IN ID#'S OF SATELLITES AND # OF SECONDS FOR EARLY/LATE
C  CORRECTION
C
   23 IGO=IRDWX(35,IANOM(JJ))
      IF(ISTEQ(IANOM(JJ),ISTP)) GO TO 25
      JJ=JJ+4
      J=J+1
      IGO=IRDWX(36,ERLAT(J))
      GO TO 23
C
C READ IN PASS START TIME, # DAYS
C
   25 IGO=IRDWX(25,T)
   10 CONTINUE
C
C PREDICTION TIME ADJUSTED TO ACCOUNT FOR
C PASS START BEFORE TIME OF PREDICTION
C
      TLUN(1)=T(1)-.125
      TLUN(2)=T(2)
      IGO=IRDXL(51,NAMST,LEN,ITYPE)
      IF(IGO.EQ.0) GO TO 90
      CALL IWRX(51,NAMST,LEN,ITYPE)
      WRITE(6,2) (NAMST(I),I=1,LEN)
    2 FORMAT(1X,10A2)
      IGO=IRDWX(52,TEMP)
      IGO=IRDWX(53,IP)
      IGO=IRDX(54,TEMP(3))
      ISTEP=IP(2)+1
C
C  TEST EPOCH TO MAKE SURE THE CORRECT YEAR WAS TYPED IN.
C
       AHILM=TEMP(1)+TEMP(2)+30.
       ALOLM=TEMP(1)+TEMP(2)-30.
       IF(T(1).LE.AHILM.AND.T(1).GE.ALOLM) GO TO 28
       CALL PUTC(IRING,1,7)
       CALL PUTC(IRING,2,7)
      DO 27 I=1,10
   27 WRITE(6,1000) IRING
 1000 FORMAT(1X,A2)
      I=IOK(" YOUR INPUT TIME IS MORE THAN 30 DAYS DIFFERENT FROM
     1  THE EPOCH OF THE ELEMENTS. DO YOU WANT TO CONTINUE?(Y OR N)")
      IF(I.EQ.0) CALL CHAIN("GO")
C
C  IF MAKING A CORRECTION TO THE MEAN ANOMALY TERM, ADD CORRECTION
C  TO TEMP(IP(4)+3).
C
   28 IF(J.EQ.0) GO TO 35
      M=0
      DO 30 K=1,JJ,4
      M=M+1
      IF(ISTEQ(NAMST,IANOM(K))) GO TO 32
   30 CONTINUE
      GO TO 35
   32 CORR=-ERLAT(M)
      ISCND=IP(4)+4
      REV=TEMP(ISCND)
      SECRV=86400./REV
      AMNAN=CORR/SECRV
      IFRST=ISCND-10034
          TEMP(IFRST)=TEMP(IFRST)+AMNAN
C
   35 CONTINUE
      CALL IWRX(57,TEMP,76,4)
C  CHANGE NODE TO NODE OF 1950
C   TEMP(ISTEP)=TEMP(ISTEP)-(TEMP(1)-33281.)*3.508E -5
      IGO=IRDWX(16,MOON)
      IF(MOON.EQ.0) GO TO 10
C
C  STARTING GRIPE LUNAR CODE, 23 MARCH 1981
C
      DO 14 I=1,300
   14 ORB(I)=0.0
      NOD=12
      NUN=0
      IND=1
      ORB(265)=TEMP(1)+TEMP(2)
      ORB(266)=0.30
      ORB(267)=0.30
C
C INTERVAL OVER WHICH TABLE OF PERTURBATION VALUES ARE PREPARED
C FOR EPHEMIS T(2)+. 25.  T(2) IS AMOUTNT OF TIME OVER WHICH
C PREDICTIONS ARE WANTED, .25 IS 1/8TH DAY ALLOWANCES TO COMPLETE
C PASS IN EITHER DIRECTION
C
      DENT=(T(2)+.25)/11.
      ORB(1)=PUT(10)
      ORB(2)=PUT(21)
      ORB(3)=PUT(111)
      ORB(4)=PUT(121)
      ORB(5)=PUT(201)
      ORB(6)=PUT(250)
      ORB(7)=PUT(222)
      ORB(9)=PUT(260)
      III=LOAD(ORB(3))
      IAA=LOAD(ORB(4))
      IXX=LOAD(ORB(2))
      ORB(287)=PUT(0)
C
C  THE FOLLOWING ARE INTEGRAL AND FRACTIONAL EPOCH AT WHICH
C  ELEMENTS ARE DEFINED.  THEY WILL NOT CHANGE IN RUN
C
      ORB(10=TEMP(1)
      ORB(11)=TEMP(2)
      ORB(130)=TEMP(1)
      ORB(131)=TEMP(2)
C
C  DIFFERENCE IN TIME FROM EPOCH OF ELEMENTS AND EPOCH OF PREDICTION
C
      ORB(121)=T(1-(TEMP(1)+TEMP(2))
      DELT=ORB(121)
      DO 240 I=143,158
 240  ORB(I)=TEMP(I-140)
      CALL RDZON
      ORB(138)=IP(1)
      ORB(139)=(IP(2)-IP(1))
      ORB(40)=(IP(3)-IP(2))
      ORB(141)=IP(4)-IP(3)
      ORB(142)=IP(5)-IP(4)
      ORB(III+2)=0
      ORB(III+3)=PUT(1)
      TLUN(2)=0.0
      DO 250 J=1,NOD
      CALL INST
      DO 249 I=1,5
  249 ORB(IXX+I-1)=ORB(IAA+I-1)
      ORB(IXX+5)=ORB(IAA+6)
      ORB(IXX+10)=ORB(IAA+7)
      ORB(IXX+14)=ORB(IAA+5)
      ORB(IXX+11)=ORB(IAA+8)
      CALL LUNAR(TLUN)
C
C PLACE EPOCH AND PERTURBATIONS IN PM ARRAY FOR USE IN EPHEM
C
      DO 444 JW=2,6
      PM(1,IND)=TLUN(1)+TLUN(2)
      PM(JW,IND)=ORB(69+JW)
  444 CONTINUE
C CONVERT MA PERTS TO RADIANS
      PM(6,IND)=PM(6,IND)*CONSOC(2)
      PM(7,IND)=ORB(79)
      PM(8,IND)=ORB(80)
      IND=IND+1
      DELT=DELT+DENT
      TLUN(1)=TLUN(1)+DENT
      ORB(IAA)=DINT(DELT)
      ORB(IAA+1)=DELT-ORB(IAA)

  250 CONTINUE
C
C  END OF GRIPE LUNAR MODIFICATIONS
C
      CALL IWRX(17,PM,384,4)
      GO TO 10
   90 CALL IWRX(0,0,0,0)
      CALL FCLOS(0)
      CALL FCLOS(1)
      CALL FCLOS(10)
      CALL CHAIN("TESSER")
      END



      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE  RDZON
C
C EMGA GAPOSCHKIN
C
C      LOAD REGISTERS WITH ZONAL HARMONICS
       COMMON ORB(300)
       COMMON/LABE2/ZON(21)
       DATA  ZON/17.043570D0,  1.08262679D-03, -2.5356D-06,-1.6234D-06,
     1 -2.2759D-07,  5.4337D-07, -3.6066D-07, -2.0702D-07, -1.2002D-07,
     1-2.4111D-07,  2.3295D-07, -1.9312D-07, -2.2861D-07,  1.2378D-07, -
     17.9890D-09,  4.1823D-08, -9.9068D-08, -6.0868D-08, -1.2610D-09, -1
     1.5175D-07, -6.7624D-10  /
       I=LOAD(ORB(5))
       DO  1  J=1,21
       K=I+J-1
     1 ORB(K)=ZON(J)
C
       RETURN
       END


       COMPILER DOUBLE PRECISION
       COMPILER NOSTACK
       SUBROUTINE INST
C
       REAL II
       REAL DX(7)
       COMMON DUMMY(110),II(10),AA(80),ZON(21),PER(8,3),DUCO(55)
       COMMON/INSTL/TFP,TIP,TZIP,TZFP,IFF,PIP,PFP,A,B,L,XL,
      1QQQQ,SUM,ISAVE,ICF,XG
C * * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * * * * *
C           THESE MODIFICATIONS BY E.M.G.
       II(5)=PUT(9)
       ISTAR=9
       RADIAN=CONSOC(7)
C
C     ICF POINTS TO COEFFICIENTS
      ICF=ISTAR+13
C  IPT POINTS TO POLYNOMIAL DEGREE
      IPT=ISTAR+8
C     STAR+8
C     SAVE T
      T1=AA(1)
      T2=AA(2)
      T=T1+T2
      I=ISTAR
C     SET UP TIME
      TIME=(AA(I+1)+T)+AA(I+2)
C     LOOP OVER 5 ELEMENTS (ICF IS CONTINUALLY UPDATED AS SUCCESSIVE
C     COEFFICIENS ARE REFERENCED.)
      DO 139 J=1,5
      IF(J-1)400,400,401
  400 AA(8)=AA(ICF+2)/RADIAN
C  STORE RATE OF PERIGEE (AT EPOCH)
      WDOT=AA(8)
  401 IF(J-2)403,402,403
  402 AA(9)=AA(ICF+2)/RADIAN
  403 CONTINUE
      SU= 6
      SUM=0.
      ISAVE=ICF
      IPT=IPT+10014
          K=AA(IPT)
      DO 160 M=1,K
      I=ICF+K+1-M
  160 SUM=SUM*T+AA(I)
      ICF=ICF+K
C     IF AP, AN, OR I, TAKE MOD 360 DEGREES, AND CONVERT TO RADIANS
      IF(J-3)166,166,167
  166 L=SUM/360.
       XL=L*360
      SUM=(SUM-XL)/RADIAN
      GO TO 205
C     IF MA, TAKE MOD. 1
  167 IF(J-5)205,204,205
  204 L=SUM
      XL=L
      SUM=SUM-XL
      IF(SUM)206,205,205
  206 SUM=1.+SUM
  205 I=J
  139 AA(I)=SUM
C     COMPUTE MEAN MOTION (DERIVATIVE OF POLYNOMIAL PART
C     OF MEAN ANOMALY).
      AA(6)=0.
      K=AA(IPT)-1
      DO 168 L=1,K
      IC=K+2-L
      I=ISAVE+IC
      XL=IC-1
  168 AA(6)=AA(6)*T+AA(I)*XL
C     COMPUTE SEMI-MAJOR AXIS
CMEAN MOTION IN REV/CTU
       XN=AA(6)*CONSOC(6)
C  RATE OF PERIGEE IN DEGREES PER CU
      XG=AA(8)*RADIAN*CONSOC(6)
      LEND=20
C
      CALL GETSMA(ZON,XN,XG,AA(4),AA(3),AA(7),LEND,II(2))
C
C  CONVERT TO MEGAMETERS
       AA(7)=AA(7)*CONSOC(5)
       RETURN
       END

      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE GETSMA(ZON,RMM,RMG,ECC,AI,SMA,LEND,IFLAG)
C
      REAL IFLAG
      REAL N,N1
      DIMENSION ZON(1)
      COMMON/SECT/SEC(2,20)
      COMMON/DIF/IFIRST
C
      DATA IFIRST/0/
C
      N=RMM*CONSOC(2)+RMG*CONSOC(1)/180.
      C=COS(AI)
      S=SIN(AI)
      C2=C*C
      C4=C2*C2
      E2=ECC*ECC
      ETA2=1.-E2
      ETA=SQRT(ETA2)
      A=(1./N**2)**(1./3.)
      IEND=LEND/2
C
      IF(IFLAG.EQ.1.AND.IFIRST.EQ.1) GO TO 11
      IFLAG=1
      IFIRST=1
C  COMPUTE SEC ONLY ON 1ST OBS/ITERATION OR FIRST TIME GETSMA CALLED
      DO 10 I=1,IEND
      L=I*2
      CALL NFINC(L,0,I,S,C,FXI,DFXI,DD)
      CALL HANSEN(-L-1,L-2*I,L-2*I,ECC,GE,DGE)
      SEC(1,I)=(-GE*DFLOAT(L+1)*2.+ETA2/ECC*DGE)*FXI*ZON(L)
      SEC(2,I)=(C/S/ETA*DFXI*GE-ETA/ECC*FXI*DGE)*ZON(L)
C
   10 CONTINUE
   11 CONTINUE
      T=10.*(1.-6.*C2+13.*C4)-5.*(5.-18.*C2+5.*C4)*E2+16.*ETA*(1.-3.
     1*C2)**2
      T=3./128.*T*ZON(2)**2/ETA**7
      W=35.-90.*C2-385.*C4+4.*ETA*(-6.+48.*C2-90.*C4)+ETA2*(-25.+12
     16.*C2-45.*C4)
      W=-W*3./128./ETA2**4*ZON(2)**2^^^^^^^^^^^
      DO 20 J=1,10
      X=1.
      DO 21 I=1,IEND
      L=I*2
      AL=A**L
   21 X=X+(SEC(1,I)+SEC(2,I))/AL
      X=X+(T+W)/A**4
      N1=N/X
      A1=(1./N1**2)**(1./3.)
      IF(ABS(A-A1).LT.1.D-9) GO TO 30
      A=A1
   20 CONTINUE
   30 SMA=A10011
          RETURN
      END


     COMPILER DOUBLE PRECISION
     COMPILER NOSTACK
     SUBROUTINE HANSEN(N,M,I,E,HA,DHA)
C-----------HA IS HANSEN COEFFICIENT.
C-----------DHA IS DERIVATIVE OF HA WITH REPECT TO ECCENTRICITY.
C----------- REFERENCE  ,TISSERAND,VOL.1,P.260.
C----------H.KINOSHITA      10 APRIL 1975
C-----------REVISED IN MAY 1976
      ETA=DSQRT(1.-E**2)
      BE=E/(1.+ETA)
      DBE=1./(1.+ETA)+(E/(1.+ETA))**2/ETA
      AN=ETA*DFLOAT(I)
      DX=E/ETA*DFLOAT(I)
      IM=IABS(I-M)
      ND=N-I
      NDD=N+I
      XD=-AN
      XDD=AN
      IF(I-M)5,20,20
    5 ND=N+I
      NDD=N-I
      XD=AN
      XDD=-AN
      DX=-DX
   20 K=0
      CALL PQ(ND,IM,XD,DX,P,DP)
      E0=P
      HA=P
      DHA=DP
      IF(E0.EQ.0.)GO TO 900
   10 K=K+1
      CALL PQ(ND,IM+K,XD,DX,P,DP)
      CALL PQ(NDD,K,XDD,-DX,Q,DQ)
      E1=P*Q*BE**(K*2)
      EPS=DABS(E1/E0)

      HA=HA+E1
      DHA=DHA+(DP*Q+P*DQ)*BE**(K*2)+P*Q*BE**(K*2-1)*DFLOAT(K*2)*DBE
      IF(EPS.GT.1.D-6)GO TO 10
      FAC=(1.-BE**2)**(N*2+3)/(1.+BE**2)**(N+1)*(-BE)**IM
      DFAC=BE**(IM-1)*(1.-BE**2)**(N*2+2)*(1.+BE**2)**(-N-2)*(DFLOAT
     1(IM)-DFLOAT(2*(3*N+4))*BE**2-DFLOAT(IM+2*(N+2))*BE**4)
      DFAC=DFAC*DBE*(-1.)**IM
      DHA=DFAC*HA+FAC*DHA
      HA=FAC*HA
  900 RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE NFINC(K,M,L,S,C,FXI,DFXI,DDFXI)
C----------H.KINOSHITA JAN 1973
C---------REVISED APRIL 1976
C---------WHEN I=0,THIS SUBROUTINE DOES NOTWORK.
      DIMENSION FAC(0:25),U(25)
      COMMON/IW1/IFACT
      COMMON/B/FAC,U
      IF(IFACT.EQ.11) GO TO 100
      CALL FACCAL(25,25)
      IFACT=11
  100 CONTINUE
      FAC(0)=1
      I=(K-M+1)/2+1
      I1=(K-L)*2+1
      I2=K-L+1
      N=K
      MD=K-L*2
      L1=L+1
      XNKD1=DFLOAT(K)
      C1=U(I)*FAC(I1)/FAC(L1)/FAC(I2)/(2.**XNKD1)
      SS=Q(K,M,MD,C)
      FXI=SS*C1
      W1=Q(K,M-1,MD,C)
      W2=Q(K,M-2,MD,C)
      NC1=(N+M)*(N-M+1)
      C2=DFLOAT(NC1)
      DFXI=(C2*W1+SS*(DFLOAT(MD)-DFLOAT(M)*C)/S)*C1
      C3=DFLOAT((N+M-1)*(N-+2))*C2
      C4=C2*(DFLOAT(MD*2)-DFLOAT(M*2-1)*C)/S
      C5=(DFLOAT(M)-DFLOAT(MD)*C+(DFLOAT(MD)-DFLOAT(M)*C)**2)/S**2
      DDFXI=(W2*C3+W1*C4+SS*C5)*C1
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE FACCAL(N,M)
      DIMENSION FAC(0:25),UP(25)
      COMMON/B/FAC,UP
C  THIS THE MAXIMUM NUMBER THAT CAN BE FACTORIALIZED CURRENTLY
C  WITHOUT OVERFLOW
      COMMON/MAXFA/MAXFAC
      DATA MAXFAC/25/
      NMAX=N
      MMAX=M
      IF(NMAX.LE.MAXFAC) GO TO 6
C  FACTORIAL OUT OF BOUNDS STOP
      STOP
    6 CONTINUE
C
      FAC(1)=1.
      DO 10 I=2,NMAX
   10 FAC(I)=FAC(I-1)*DFLOAT(I-1)
      IF(M.LE.MAXFAC) GO TO 160
C   FACTORIAL OUT OF BOUNDS STOP
      STOP
   16 CONTINUE
      UP(1)=1.
      DO 20 I=2,MMAX
   20 UP(I)=UP(I-1)*(-1.)
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE PQ(N,K,X,DX,P,DP)
      DIMENSION F(0:25),UP(25)
      COMMON/IW1/IFACT
      COMMON/B/F,UP
      IF(IFACT.EQ.11) GO TO 20
      CALL FACCAL(25,25)
      IFACT=11
   20 CONTINUE
      IF(K)40,30,40
   30 P=1.
      DP=0.
      RETURN
   40 CONTINUE
      IF(X.EQ.0.)GO TO 60
      P=0.
      P=0.
      K1=K+1
      DO 50 J1=1,K1
      J=J1-1
      KJ=K-J+1
      HP= FN(N+1+K,N+1+J)/F(J1)/F(KJ)*X**J
      P=P+HP
      DP=DP+HP*DFLOAT(J)/X
   50 CONTINUE
      DP=DP*DX
      RETURN
   60 CONTINUE
      P=FN(N+1+K,N+1)/F(K+1)
      DP=0.
      RETURN
      END
      COMPILER DOUBLE PRECISIION
      COMPILER NOSTACK
      FUNCTION Q(N,M,MD,C)
      DIMENSION FAC(0:25),U(25)
      COMMON/B/FAC,U
      IF(M)  10,11,11
   10 IF(MD)  20,21,21
   20 M1=-M
      MD1=-MD
      I=M1-MD1
      IF(I)  22,23,23
   22 I=-I+1
      GO TO 24
   23 I=I+1
   24 I1=N+MD1+1
      I2=N-MD1+ 1
      I3=N-M1+1
      I4=N+M1+1
      Q  =QQQP(N,M1,MD1,C)*U(I)*FAC(I1)/FAC(I2)*FAC(I3)/FAC(I4)
      RETURN
   21 M1=-M
      I=N-MD+1
      I1=N-M1+1
      I2=N+M1+1
      DC=-C
      Q=  QQQP(N,M1,MD,DC)*U(I)*FAC(I1)/FAC(I2)
      RETURN
   11 IF(MD)  30,31,31
   30 MD1=-MD
      DC=-C
      I=N-M+1
      I1=N+MD1+1
      I2=N-MD1+1
      Q=  QQQP(N,M,MD1,DC)*U(I)*FAC(I1)/FAC(I2)
      RETURN
   31 Q=  QQQP(N,M,MD,C)
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION QQQP(N,M,MD,C)
      DIMENSION FAC(0:25),U(25)
      COMMON/B/FA,U
      IF(M-MD)  10,11,11
   10 M1=MD
      MD1=M
      I=MD-M+1
      I1=N+M+1
      I2=N-M+1
      I3=N-MD+1
      I4=N+MD+1
```

```
     .QQP=QP(N,M1,MD1,C)*U(I)*FAC(I1)/FAC(I2)*FAC(I3)/FAC(I4)
      RETURN
   11 QQQP=QP(N,M,MD,C)
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION QP(N,M,MD,C)
C    H.  KINOSHITA  JAN  U1973
      DIMENSION FAC(0:25),UP(25)
      COMMON/B/FAC,UP
      S=0.
      C1=1.D30
      A=1.-C
      B=1.+C
      IRE=N-M+1
      DO  10  I=1,IRE
      IR=I-1
      I1=M-MD+IR+1
      I2=N+MD-IR+1
      I3=N-M-IR+1
      IF(I3.EQ.0) I3=1
      S=S+C1*UP(I)/FAC(I1)/FAC(I2)/FAC(I)/FAC(I3)*FQP(A,IR)*FQP(B,I3-
     11)
  10  CONTINUE
      I1=N+M+1
      I2=N-MD+1
      C2=1./C1*FAC(I1)*FAC(I2)/2.**N
      C3=FQP(A,M-MD)*FQP(B,M+MD)
      C3=DSQRT(C3)
      QP=S*C2*C3
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION FQP(A,M)
      IF(M.EQ.0)GO TO 10
      FQP=A**M
      RETURN
   10 FQP=1.
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION FN(N,M)
      FN=1.
      ID=IABS(N-M)
      IF(ID)20,10,20
   10 RETURN
   20 DO 30 I=1,ID
      FN=FN*DFLOAT(N-I+1)
   30 CONTINUE
      RETURN
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION CONSOC(I)
      REAL ITEM(10)
      COMMON/ITE/ITEM,GM,ROTE,SECDAY
      DATA ITEM/
     1  3.141592653589793D0,
     2  6.283185307179586D0,
     3  0.D0,0018      4  1.D20,
     5  6.378136D0,
     6  0. D0,
     7  57.29577951308232D0,
     8  0.D0,
     9  1.D20,
     A  1.D20/
C  GM  (MM**3/SEC**2)
      DATA GM/3.986005D-4/
C  ROTATION RATE OF EARTH (RAD/SEC)
      DATA ROTE/.7292115085D-4/
C  SEC/DAY
      DATA SECDAY/86400.D0/
C
C
  100 CONTINUE
      CONSOC=ITEM(I)
      IF(ITEM(I).NE.0) RETURN
C  CALCULATE CONSOC(3), CONSOC(6), CONSOC(8)
C    GE DAYS/CTU
      ITEM(6)=DSQRT(ITEM(5)**3/GM)/SECDAY
C  ROT EARTH (RAD/CTU)
      ITEM(8)=ROTE*ITEM(6)*SECDAY
C  GM (MM**3 (REV/DAY)**2 )
      ITEM(3)=GM*(SECDAY/ITEM(2))**2
      GO TO 100
      END


      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION ERIQA (AMA,ECC,SINE)
C     SOLVING KEPLER"S EQUATION
C     USING THEDOI SCHEME
C     RUDOLF LOESER, APRIL 1965
      KOUNT=0
      CONV=1.D-12
      MA=AMA
      YMA=MA
      XMA=(AMA-YMA)*CONSOC(2)
      EP=XMA
      DEP=0.
  100 SINE=DSIN(EP)
      KOUNT=KOUNT+1
      ERIQA=XMA+ECC*SINE
      DE=ERIQA-EP
      IF(ABS(DE)-CONV  )104,101,101
  101 IF(DEP)103,102,103
  102 DEP=DE
      EP=ERIQA
      GO TO 100
  103 EP=EP+DE*(DEP/(DEP-DE))
      DEP=0.
      GO TO 100
  104 CONTINUE
      RETURN
      END

      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK

      SUBROUTINE EVA (AMA,ECC,SINE,COSE,SINV,COSV,E,V)
C     RUDOLF LOESER, 1 MAR 66
C     USES THE VALUES OF AMA AND ECC TO COMPUTE THE VALUES OF
C     SIN(E), COS(E), SIN(V) AND COS(V).
      PI=CONSOC(1)
      TOPI=CONSOC(2)
      E=ERIQA(AMA,ECC,SINE)
      COSE=COS(E)
      C=1.-ECC*COSE
      SINV=SQRT(1.-ECC*ECC)*SINE/C
      COSV=(COSE-ECC)/C
      V=ASIN(SINV)
      IF(SINV)102,100,100
  100 IF(COSV)101,104,104
  101 V=PI-V
      GO TO 104
  102 IF(COSV)103,104,104
  103 V=-PI-V
  104 IF(V)105,106,106
  105 V=TOPI+V
  106 CONTINUE
      RETURN
      END

      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE KIND(TIME,ZZ,ROA)
      COMMON ORB(300)
      COMMON/VAR1/W,OM,INC,E,MA,N,A,SW,CW,S2W,C2W,S3W,C3W,
     X  E2,EFAC,SINI,COSI,TANI
      COMMON/VAR2/NMOON,AMOON,NSUN,ASUN,FACMOON,FACSUN,AE,
     X  J2,TWOPI,RACORR,DWE,DWI,DME,DMI,DOE,DOI
      COMMON/VAR3/B,,CJ,CT,DELT,POINT,DTMAX,K1,K2,K3,GMM
      COMMON/VAR5/ALF,AP,AOR,AOR2,AOR3,ARAT,ARAT2,ARAT3,AA,A2,
     X  AB2,APB,AMP,PB2,ACPBS,ACMBC
      COMMON/VAR6/DEC,SD,CD,BB,DAO,DBI,DBO,B2,SPEC1,SPEC2,SPEC3,
     X  X,Y,Z,V,TEMP
```

```
C
      DIMENSION B(5)C(5,3),XX(10,9),ZZ(3)
      DIMENSION BT(4,2,2),CJ(3,2),CT(16,3)
      EQUIVALENCE (BT,CT(1,3))
      EQUIVALENCE(ORB(21),XX)
      REAL INC,J2,MA,N,NMOON,NSUN
C
      INTEGER POINT
C
C
C         EVALUATE THE DERIVATIVES
C
      DO 90 KINDI=1,2
      IF(KINDI.EQ.2) GO TO 27
      CALL LUNVECT(TIME,ZZ,ROA)
      ALF=ATANG(ZZ(2),ZZ(1))
      DEC=ASIN(ZZ(3))
      AP=AMOON
      FAC=FACMOON
      GO TO 30
   27 CALL SUNVECT(TIME,ZZ,ROA)
      CALL PRECESS(ZZ,ZZ,TIME,1)
      ALF=ATANG(ZZ(2),ZZ(1))
      DEC =ASIN(ZZ(3)/ROA)
      AP=ASUN
      FAC=FACSUN
   30 AOR=1./ROA
      AOR2=AOR**2
      AOR3=AOR*AOR2
      FAC=FAC*AOR3*TWOPI
      ARAT=A/AP
      ARAT2=ARAT**2
      ARAT3=ARAT*ARAT2
      SD=SIN(DEC)
      CD=COS(DEC)
      IF(K1.EQ.1) ALF=ALF-RACORR
      X=SIN(OM-ALF)
      BB=-CD*COSI*X+SD*SINI
      DAO=-CD*X
      DBI=CD*SINI*X+SD*COSI
      X=COS(OM-ALF)
      AA=CD*X
      DBO=-CD*COSI*X
      A2=AA*AA
      B2=BB*BB
      AB2=2.*AA*BB
      APB=A2+B2
      AMB=A2-B2
      APB2=APB*APB
      ACPBS=AA*CW+BB*SW
      ASMBC=AA*SW-BB*CW
C
      X=AMB*S2W-AB2*C2W
      TEMP=15./4.*E*X
      X=(4.+3.*E2)*(5.*APB-4.)
      Y=(A2-3.*B2)*A
      Z=(3.*A2-B2)*BB
      V=3.*X*ASMBC+105.*E2*(Y*S3W-Z*C3W)
      TEMP=TEMP-5./64.*AOR*ARAT*V
      X=7.*APB-6.
      V=X*(AMB*S2W-AB2*C2W)
      TEMP=TEMP+105./32.*AOR2*ARAT2*E*V
      X=21.*APB2-28.*APB+8.
      V=X*ASMBC
      TEMP=TEMP-105./128.*AOR3*ARAT3*V
      B(4)=FAC*EFAC*TEMP
C
      SPEC1=AA*DAO+BB*DBO
      SPEC2=AA*DAO-BB*DBO
      SPEC3=AA*DBO+BB*DAO
      X=2.+3.*E2
      TEMP=-3./4.*SPEC1*X-15./4. *E2*(SPEC2*C2W+SPEC3*S2W)
      X=3.*(4.+3.*E2)
C
      Y=15.*A2+5.*B2-4.
      Y=Y*DAO+5.*AB2*DBO
      Z=5.*A2+15.*B2-4.
      Z=Z*DBO+5.*AB2*DAO
      V=X*(Y*CW+Z*SW)
      Y=AMB*DAO-AB2*DBO
      Z=AMB*DBO+AB2*DAO
      V=V+105.*E2*(Y*C3W+Z*S3W)
      TEMP=TEMP+5./64.*E*AOR*ARAT*V
      X=7.*APB-4.
      Y=1.+5.*E2
      V=2.*X*SPEC1*Y
      X=AA**3*DAO- BB**3*DBO
      X=7.*X-3.*SPEC2
      Y=3.*AA**2*BB*DAO+AA**3*DBO
      Y=Y+3.*AA*BB**2*DBO+BB**3*DAO
      Y=7.*Y-6.*SPEC3
      Z=2.*X*C2W+Y*S2W
      V=V+7.*E2*Z
      TEMP=TEMP-15./32.*AOR2*ARAT2*V
      X=28.*(3.*APB-2.)
      V=X*SPEC1*ACPBS
      Y=21.*APB2-28.*APB+8.
      Z=DAO*CW+DBO*SW
      V=V+Y*Z
      TEMP=TEMP+105./128.*E*AOR3*ARAT3*V
      B(3)=FAC/EFAC/SINI*TEMP
      X=E/TANI/(1.-E2)
      B(3)=B(3)-X*B(4)
C
      X=BB*(2.+3.*E2)
      Y=E2*(BB*C2W-AA*S2W)
      TEMP=3./4.*X-15./4.*Y
      X=4.+3.*E2
      Y=5.*A2+15.*B2-4.
      V=X*(5.*AB2*CW+Y*SW)
      X=AB2*C3W-AB*S3W
      V=V-35.*E2*X
      TEMP=TEMP-15./64.*E*AOR*ARAT*V
      X=7.*APB-4.
      Y=1.+5.*E2
      V=2.*BB*X*Y
      X=2.*BB*(7.*B2-3.)
      Y=7.*A2+21.*B2-6.
      V=V-7.*E2*(X*C2W-Y*AA*S2W)
      TEMP=TEMP+15./32.*AOR2*ARAT2*V
      X=14.*AB2*(3.*APB-2.)
      Y=21.*A2*A2+126.*A2*B2
      Y=Y+105.*B2*B2-28.*A2
      Y=Y-84.*B2+8.
      V=X*CW+Y*SW
      TEMP=TEMP-105./128.*E*AOR3*ARAT3*V
      B(2)=FAC/EFAC/SINI*DBI*TEMP
C
      X=3.*APB-2.
      TEMP=3./4.*X+15./4.*(AMB*C2W+AB2*S2W)
      X=4.+9.*E2
      Y=5.*APB-4.
      V=X*Y*ACPBS
      X=A2-3.*B2
      Y=3.*A2-B2
      Z=X*AA*C3W+Y*BB*S3W
      V=V+35.*E2*Z
      TEMP=TEMP-15./64.*AOR*ARAT/E*V
      V=35.*APB2-40.*APB+8.
      X=7.*APB-6.
      V=V+7.*X*(AMB*C2W+AB2*S2W)
      TEMP=TEMP+15./32.*AOR2*ARAT2*V
      X=21.*APB2-28.*APB+8.
      V=X*ACPBS
      TEMP=TEMP-105./128./E*AOR3*ARAT3*V
      B(1)=FAC*EFAC*TEMP
      B(1)=B(1)-COSI*B(2)
C
      X=7.+3.*E2
      V=-X/4.*(3.*APB-2.)
      X=1.+E2
      TEMP=V-15./4.*X*(AMB*C2W+AB2*S2W)
      X=4.+29.*E2
      Y=5.*APB-4.
      V=X*Y*ACPBS
      X=A2-3.*B2
      X=3.*A2-B2
      Z=X*AA*C3W+Y*BB*S3W
C  CHANGE CODE ACCORDING TO KOZAI 8/19/80
      X=1.+E2
      V=V+35.*E2*Z*X
      TEMP=TEMP+15./64.*AOR*ARAT/E*V
```

```
C  CHANGE CODE ACCORDING TO KOZAI 8/19/80
      X=9.+15.*E2
      Y=35.*APB2-40.*APB+8.
      V=X*Y
C  CHANGE CODE ACCORDING TO KOZAI 8/19/80
      X=1.+3.*E2
      Y=7.*APB-6.
      Z=AMB*C2W+AB2*S2W
      V=V+35.*X*Y*Z
      TEMP=TEMP-3[illegible]32.*AOR2*ARAT2*V
      X=21.*APB2[illegible]*APB+8.
      V=X*ACPB3
      TEMP=TEMP+105./128.*AOR3*ARAT3/E*V
      B(5)=FAC*TEMP/TWOPI
C
C         SAVE DERIVATIVES
C
      IF(KINDI.EQ.2) GO TO 42
      DO 40 J=1,5
      C(J,POINT)=B(J)
   40 CONTINUE
      GO TO 50
   42 DO 45 J=1,5
      C(J,POINT)=C(J,POINT)+B(J)
   45 CONTINUE
C
C         GET THE SHORT-PERIOD PERTURBATIONS
C
   50 IF(DELT.NE.0.)GO TO 60
      IF(POINT.NE.3) GO TO 60
      AM=MA*TWOPI
      X=AM+2.*W
      SINM=SIN(X)
      COSM=COS(X)
      X=X+AM
      SIN2M=SIN(X)
      COS2M=COS(X)
      X=X+AM
      SIN3M=SIN(X)
      COS3M=COS(X)
      X=X+AM
      COS4M=COS(X)
      SIN4M=SIN(X)
      XTEMP1=9.*COSM-COS3M
      XTEMP2=9.*SINM-SIN3M
      Y=1.-1.5D0*APB
C
C  GET IN IN RAD/AY FOR THIS SECTION (S.P. PERTS)
      N=N*TWOPI
C
C  INDENTED LINES ARE ADDITIONAL TERMS INVOLVING FIRST ORDER
C  ECCENTRICITY DEPENDENCE DEVELOPED BY KOZAI 8/13/79
C
      TEMP=AMB*COS2M+AB2*SIN2M
```

```
C
      DA=A*1.5D0*FAC*TEMP/N
C
      TEMP=Y*COS(AM)
      X=9.*COSM+COS3M
      TEMP=TEMP+.25D0*AMB*X
      X=9.*SINM+SIN3M
      TEMP=TEMP+AB2/4.*X
      DE=FAC*TEMP/N
C
      TEMP=-Y*SIN(AM)
      X=9.*SINM-SIN3M
      TEMP=TEMP+.25D0*AMB*X
      X=9.*COSM-COS3M
      TEMP=TEMP-AB2/4. *X
      DM=FAC*TEMP/N/E
C
      X=AMB*SIN2M-AB2*COS2M
      TEMP=-21./8.*X
      X=BB*SIN2M+AA*COS2M
      TEMP=TEMP+.75D0/TANI*DBI*X
      DW=FAC*TEMP/N
C
      TEMP=DBI*X/SINI
      DOMEGA=-.75D0*FAC*TEMP/N
C
      X=AMB*COS2M+AB2*SIN2M
      TEMP=COSI*X
      TEMP=TEMP-SPEC2*SIN2M
      TEMPTEMP+SPEC3*COS2M
      DI=.75D0*FAC/SINI*TEMP/N
C
C
C  CALCULATE S.P. EFFECTS DUE TO TIDES  (KOZAI 8/13/79)
C
      AAERAT=A/AE
      FACT=(FAC/N)/(AAERAT**5)*ORB(265+KINDI)
      Y=1.-1.5D0*APBO

      X=-3.D0*Y*E*COS(AM)
      X=X-.75D0*AMB*(E*COSM-2.*COS2M-7.*E*COS3M)
      X=X-.75D0*AB2*(E*SINM-2.*SIN2M-7.*E*SIN3M)
      DAT=A*FACT*X
C
      X=-.75*Y*(2.*COS(AM)+3.*E*COS(2.*AM))
      X=X+1./16.D0*AMB*(6.*COSM-6.*E*COS2M+14.*COS3M+51.*E*COS4M)
      X=X+1./16.D0*AB2*(6.*SINM-6. *E*SIN2M+14.*SIN3M+51.*E*SIN4M)
      DET=FACT*X
C
      X=.75*Y*(2.*SIN(AM)+3.*E*SIN(2*AM))
      X=X+1./16.D0*AMB*(6.*SINM+48.*E*SIN2M-14.*SIN3M-51.*E*SIN4M)
      X=X-1./16.D0*AB2*(6.*COSM+48.*E*COS2M-14.*COS3M-51.*E*COS4M)
      DMT=(FACT/E)*X
C
```

```
      XTEMP1=3.*E*COSM-3.*COS2M-7.*E*COS3M
      XTEMP2=3.*E*SIN M-3.*SIN2M-7.*E*SIN3M
      X=-21./4.D0*Y*E*SIN(AM)
      X=X-3./8.D0*AMB*XTEMP2
      X=X+3./8.D0*AB2*XTEMP1
      X=X-1./16.D0*E*AMB*(3.*SINM-7.*SIN3M)
      X=X+1./16.D0*E*AB2*(3.*COSM-7.*COS3M)
      TEMP=-9./2.D0*BB*E*SIN(AM)
      TEMP=TEMP.25D0*BB*XTEMP2
      TEMP=TEMP-.25D0*AA*XTEMP1
      X=X+TEMP*DBI/TANI
      DMWT=FACT*X
C
      X=9./2.D0*(AA*DAO+BB*DBO)*E*SIN(AM)
      X=X-.25*AMB*COSI*XTEMP1
      X=X-.25*AB2*COSI*XTEMP2
      X=X+.25*(AA*DAO-BB*DBO)*XTEMP2
      X=X-.25*(AA*DBO+BB*DAO)*XTEMP1
      DIT=FACT*X/SINI
C
      X=9./2.D0*BB*E*SIN(AM)
      X=X+.25*BB*XTEMP2
      X=X+.25*AA*XTEMP1
      DOMEGAT=FACT*X*DBI/SINI
C
C  ADD S.P. TIDAL PERTS TO EXISTING S.P. PERTS
C
      DA=DA+DAT
      DE=DE+DET
      DM=DM+DMT
      DMW=DMW+DMWT
      DI=DI+DIT
      DOMEGA=DOMEGA+DOMEGAT
C
C
C  ADD S.P. PERTS TO EXISTING PERTS FOR NODE,INC, U, AND R
C
      XX(2,6)=XX(2,6)+DOMEGA
      XX(3,6)=XX(3,6)+DI
C
      IF(KINDI.EQ.2) GO TO 55
      CALL EVA(MA,E,SE,CE,SV,CV,EA,VEE)
   55 ROA=1.-E*CE
      DW=DMW-DM
      X=1./ROA+1./EFAC/EFAC
      X=DW+X*SV*DE
      XX(9,6)=XX(9,6)+X+EFAC/ROA/ROA*DM
      X=ROA*DA-A*CV*DE
      XX(10,6)=XX(10,6)+X+A*E*SV/EFAC*DM
C
C
C
      N=N/TWOPI
C
```

```
C
C        ADD L.P. TERMS DUE TO THE BODY TIDES.a
C
   60 ALF=ALF+ORB(267+KINDI)*TWOPI/360.D0
      X=SIN(OM-ALF)
      BB=-CD*COSI*X+SD*SINI
      DAO=-CD*X
      DBI=CD*SINI*X+SD*COSI
      X=COS(OM-ALF)
      AA=CD*X
      DBO=-CD*COSI*X
      X=AA*AA+BB*BB
      SPEC=3.*(.75D0*X-.5D0)
      X=AE/A
      X=X**5
      Y=1.-E2
      Y=Y*Y
      FF=FAC*X*ORB(265+KINDI)/Y
C
      B(4)=0.
      X=AA*DAO+BB*DBO
      B(3)=-1.5D0*FF/SINI*X
      B(2)=1.5D0*FF/SINI*BB*DBI
      B(1)=-COSI*B(2)+F*SPEC
      B(5)=FF*SPEC*EFAC/TWOPI
      Y=ORB(265+KINDI)
      IF(Y.EQ.0) Y=1
      DO 63 J=1,3
      BT(J,1,KINDI)=B(J)/Y
   63 CONTINUE
      BT(4,1,KINDI)=B(5)/Y
C
      Y=AA**2-DAO**2+BB**2
      Y=Y-BB*SINI*SD-DBO**2
      BT(3,2,KINDI)=-1.5D0*FF/SINI*Y
      Y=AA*BB*SINI+DBO*DBI
      Y=-1.5D0*FF/SINI*Y
      B T(2,2,KINDI)=Y
      BT(1,2,KINDI)=-COSI*Y-4.5D0*FF*X
      BT(4,2,KINDI)=-4.5D0*FF*X*EFAC/TWOPI
C
C  ADD BODY TIDES
      DO 65 J=1,5
      C(J,POINT)=C(J,POINT)+B(J)
  65  CONTINUE
C
   90 CONTINUE
      RETURN
      END


      COMPILER NOSTACK
      FUNCTION LOAD(I)
C  FUNCTION LOAD TAKES THE LOWER ORDER 4 BYTES FROM A REAL*8 AND PUTS
```

```
C  THEM INTO AN INTEGER*4
      INTEGER I(2)
      LOAD=I(1)
      RETURN
      END

      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE LUNARK(TT)
C
C       SPECIAL VERSION INC. S.P. INT. TERMS FOR SUN AND MOON
C
C COMPUTE LUNI-SOLAR PERTURBATIONS BY THE NEW METHOD OF
C       KOZAI (SEE SPECIAL REPORT 349).
C
      COMMON ORB(300)
      COMMON/VAR1/W,OM,INC,E,MA,N,A,SW,CW,S2W,C2W,S3W,C3W,
     X  E2,EFAC,SINI,COSI,TANI
      COMMON/VAR2/NMOON,AMOON,NSUN,ASUN,FACMOON,FACSUN,AE,
     X  J2,TWOPI,RACORR,DWE,DWI,DME,DMI,DOE,DOI
      COMMON/VAR3/B,C,CJ,CT,DELT,POINT,DTMAX,K1,K2,K3,GMM
      COMMON/VAR4/INIT
C
      DIMENSION B(5),C(5,3),TT(2),XX(10,9),ZZ(3)
      DIMENSION BT(4,2,2),CJ(3,2),CT(16,3)
      EQUIVALENCE (BT,CT(1,3))
      EQUIVALENCE(ORB(21),XX)
      REAL INC,J2,MA,N,NMOON,NSUN
      INTEGER POINT
C
      DATA INIT/0/
      DATA GMM/76298.43D0
      DATA DTMAX,K1,K2,K3/.25D0,0,0,1/
C
      NO=10
      TWOPI=CONSOC(2)
C  SET INIT TO 1 FOR ALL SUBSEQUENT OBS
C
C  EPSILON FOR NUTATION ININC TERMS (SEC OF ARC)
      EPSILON=ORB(286)
      IF(EPSILON.LE.0.OR.EPSILON.GE.1.) EPSILON=.001D0
C
C         SET CONSTANTS
C
C  USE SIDEREAL PERIOD FOR MOON
      NMOON=1./27.321661D0
      AMOON=384.4D0
      NSUN=1./365.25964D0
      ASUN=1.4959884D5
      AE=CONSOC(5)
C                   SET ARRAY POINTERS
      ORB(112)=PUT(1)
      J2=ORB(202)
C                   INITIALIZE
```

```
      EP=ORB(265)
      X=ORB(10)+ORB(11)-33282.
      RACORR=(3.506D-5)*X/57.2957795D0
      TL=EP
      TIN=TT(1)+TT(2)
      DELT=TIN-TL
      SIGN1=1.
      IF(DELT.LT.0.) SIGN1=-SIGN1
      N=XX(5,2)
      FACMOON=.0121835D0*NMOON**2/N
      FACSUN=NSUN**2/N
      TIME=TL
      DO 14 J=1,6
      XX(J,6)=0.
   14 CONTINUE
      XX( 9,6)=0.
      XX(10,6)=0.
C
C         PREPARE DT FOR THE INTEGRATION STEP
C
C   INIT=0 ON FIRST PASS THROUGH LOOP
      POINT=2
      IF(INIT.EQ.0) GO TO 25
   20 X=ABS(DELT)
      DT=X
      IF(X.GT.DTMAX) DT=DTMAX
      DT=DT*SIGN1
      DT2=DT/2.
C   RETURN FOR 2ND HALF OF INT STEP
   22 IF(POINT.EQ.3) DELT=DELT-DT
      TIME=TIME+DT2
   25 CONTINUE
      ORB(121)=(TIME-ORB(10))-ORB(11)
      ORB(122)=0.
      ORB(111)=PUT(4)
      ISAVE=LOAD(ORB(113))
      ORB(113)=PUT(1)
      CALL INST
      ORB(113)=PUT(ISAVE)
      CALL SETUP
C
      CALL KIND(TIME,ZZ,ROA)
C
C         INTEGRATE
C
      IF(INIT.EQ.1) GO TO 100
C   INTERACTION WITH J2
      X=ORB(263)
      Y=ORB(262)
      CJ(1,2)=DWE*X+DWI*Y
      CJ(2,2)=DOE*X+DOI*Y
      CJ(3,2)=DME*X+DMI*Y
      INIT=1
      GO TO 110
```

```
  100 IF(POINT.EQ.3) GOTO 104
C                                RECALL THE EQUIVALENCE OFBT AND CT(1,3)
      DO 102 J=1,16
      CT(J,2)=CT(J,3)
  102 CONTINUE
      POINT=3
      GO TO 22
C  INT DIR EFF USING SIMPSONS RULE
  104 DO 106 J=1,5
      X=(C(J,1)+4.*C(J,2)+C(J,3))
      ORB(260+J-1)=ORB(260+J-1)+X*DT2/3.
  106 CONTINUE
C  INT INTERACTION WITH J2 BY TRAPEZ
      X=ORB(263)
      Y=ORB(262)
      CJ(1,2)=DWE*X+DWI*Y
      CJ(2,2)=DOE*X+DOI*Y
      CJ(3,2)=DME*X+DMI*Y
      DO 107 J=1,3
      K=J
      IF(J.EQ.3) K=5
      X=(CJ(J,1)+CJ(J,2))/2.
      ORB(260+K-1)=ORB(260+K-1)+X*DT
  107 CONTINUE
C  INT TIDE BY SIMPSONS RULE
      DO 108 J=1,16
      X=(CT(J,1)+4.*CT(J,2)+CT(J,3))
      K=J+10
      ORB(260+K-1)=ORB(260+K-1)+X*DT2/3.
  108 CONTINUE
C
C  SHIFT DERIVS FOR DIRECT AND PERTS
  110 DO 113 J=1,5
      C(J,1)=C(J,POINT)
      IF(J.GT.3) GO TO 113
C  DERIVS FOR J2 INTERACTION
      CJ(J,1)=CJ(J,2)
  113 CONTINUE
C  DERIVS FOR TIDE
      DO 116 J=1,16
      CT(J,1)=CT(J,3)
  116 CONTINUE
      IF(DELT.EQ.0.) GO TO 120
      POINT=2
C  RETURN FOR NEXT TIME STEP
      GO TO 20
C
  120 TL=TIN
      DO 125 J=1,5
      XX(J,6)=XX(J,6)+ORB(260+J-1)
 125 CONTINUE
      ORB(265)=TL
C
C
```

```
C
      RETURN
C
      END



      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE PRECESS (XZ,X,T,JAY)
C
C              XZ IS VECTOR REFERRED TO EQUINOX OF 1950.
C              X  IS VECTOR REFERRED TO EQUINOX OF SMITHSONIAN DAY = T
C
C              JAY=1 MEANS XZ IS INPUT VECTOR AND X IS OUTPUT
C              JAY=2 MEANS X IS INPUT VECTOR AND XZ IS OUTPUT
C   JAY=3 MEANS REPEAT THE LAST CALL TO PRECESS USING NEW
C VALUES OF COMPONENTS IN THE INPUT VECTOR.
C
      DIMENSION V(3),W(3),X(3,XZ(3)
      IF(JAY-2)1,1,2
    1 KAY=JAY
      D=(T-33281.923D0)/36524.22D0
      D2=D*D
      D3=D2*D
      XX=1.-.00029696D0*D2-.00000013D0*D3
      YX=-.02234941D0*D-.00000676D0*D2+.00000221D0*D3
      ZX=-.00971690D0*D+.00000207D0*D2+.00000096D0*D3
      YY=1.-.00024975D0*D2-.00000015D0*D3
      ZY=-.00010858D0*D2
      ZZ=1.-.00004721D0*D2
    2 IF(KAY-1)3,3,6
C                             XZ IS INPUT
    3 DO 4 J=1,3
    4 W(J)=XZ(J)
      V(1)=ZX*W(3)+YX*W(2)+XX*W(1)
      V(2)=ZY*W(3)+YY*W(2)-YX*W(1)
      V(3)=ZZ*W(3)+ZY*W(2)-ZX*W(1)
      DO 5 J=1,3
    5 X(J)=V(J)
      RETURN
C X IS INPUT
    6 DO 7 J=1,3
    7 V(J)=X(J)
      W(1)=XX*V(1)-YX*V(2)-ZX*V(3)
      W(2)=YX*V(1)+YY*V(2)+ZY*V(3)
      W(3)=ZX*V(1)+ZY*V(2)+ZZ*V(3)
      DO 8 J=1,3
    8 XZ(J)=W(J)
      RETURN
C
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION PUT(I)
      COMMON/EQUI/IY
      EQUIVALENCE (IY,Y)
C  C  FUNCTION THAT ON THE VAX PUTS INTEGER*4 INTO BITS 0-31 OF
C  REAL*8 WORD. (SEE FORTRAN USER'S GUIDE PGS A1-A3)


      Y=0
      IY=I
      PUT=Y
      RETURN
      END
```

```
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION PUT(I)
      COMMON/EQUI/IY
      EQUIVALENCE (IY,Y)
C C  FUNCTION THAT ON THE VAX PUTS INTEGER*4 INTO BITS 0-31 OF
C  REAL*8 WORD. (SEE FORTRAN USER'S GUIDE PGS A1-A3)


      Y=0
      IY=I
      PUT=Y
      RETURN
      END
```

```
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE SETUP
C
      COMMON ORB(300)
      COMMON/VAR1/W,OM,INC,E,MA,N,A,SW,CW,S2W,C2W,S3W,C3W,
     X  E2,EFAC,SINI,COSI,TANI
       COMMON/VAR2/NMOON,AMOON,NSUN,ASUN,FACMOON,FACSUN,AE,
     X  J2,TWOPI,RACORR,DWE,DWI,DME,DMI,DOE,DOI
C
      REAL INC,J2,MA,N,NMOON,NSUN
      W=ORB(121)
      OM =ORB(122)
      INC=ORB(123)
      E  =ORB(124)
      MA =ORB(125)
      N  =ORB(126)
      A  =ORB(127)
      SW =SIN(W)
      CW =COS(W)
      S2W=2.*SW*CW
      C2W=CW*CW-SW*SW
      S3W=SW*C2W+CW*S2W
      C3W=CW*C2W-SW*S2W
      E2=E*E
      EFAC=DSQRT(1.-E2)
      SINI=SIN(INC)
      COSI=COS(INC)
      TANI=SINI/COSI
C
C         GET THE CORRECTION TERMS INVOLVING J2
C
      X=AE/A
      Y=1.-E2
      X=J2*N*TWOPI*X*X
      X=X/Y/Y
      DOE=-6.*X*E*COSI/Y
      DOI=1.5D0*X*SINI
      X=1.-5.*COSI**2
      DWE=X*DOE/(2.*COSI)
      DWI=-5.*COSI*DOI
      X=1.-3.*COSI**2
      DME=3./8.*EFAC*X*DOE/COSI
      DMI=-3.*COSI*EFAC*DOI
      DME=DME/TWOPI
      DMI=DMI/TWOPI
      RETURN
      END



      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      FUNCTION ATANG (OVER,UNDER)
C
      DIMENSION C(3)
      C(1)=CONSOC(1)/2.
      C(2)=CONSOC(1)
      C(3)=CONSOC(2)
C
      A=OVER
      B=UNDER
      D=DSQRT(A*A+B*B)
      A=A/D
      B=B/D
      IF(ABS(B)-.00000001)1,4,4
    1 D=C(1)
      IF(A)2,3,3
    2 D=D+C(2)
    3 ATANG=D
      RETURN
    4 D=DATAN(A/B)
      IF(B)2,5,5
    5 IF(A)6,3,3
    6 D=D+C(3)
      GO TO 3
C
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE LUNVECT (TIME,VECTOR,ROVERA)
C
C           GETS DIRECTION COSINES OF THE MOON WITH RESPECT TO THE
C EQUINOX AND EQUATOR OF THE EPOCH TIME (EXPRESSED IN
C           SMITHSONIAN DAYS).
C                                             R.E.BRIGGS -- MAY 1973
C
C
C  REVISED MAR 77 TO COMPUTE POSITION FROM BROWNS THEORY
C
      DIMENSION VECTOR(3),XX(3)
      REAL LONGS,MS
      REAL LONG,M
      SPR=206264.806
C
      RADIAN = CONSOC(7)
      REVOL=360.
      T=(TIME-15019.5)/36525.
      T2=T*T
C
      X=1336.85523095*T
      J=X
      Y=J
      X=(X-Y)*REVOL
      Y=270.434164+X-.001133*T2
      LONG=Y/RADIAN
C
      X=11.30287231*T
      J=X
      Y=J
      X=(X-Y)*REVOL
      Y=334.329556+X-.010325*T2
      GAMMA=Y/RADIAN
C
       X=5.37261669*T
      J=X
      Y=J
      X=(X-Y)*REVOL
      Y=259.183275-X+.02078*T2
      OMEGA=Y/RADIAN
C
C                    COMPUTE POSITION FROM BROWNS THEORY.
C
      M=LONG-GAMMA
      X=129602768.13*T+1.089*T2
      X=279.69668+X/3600.
      LONGS=X/RADIAN
      X=129596579.10*T-0.54*T2
      X=358.47583+X/3600.
      MS=X/RADIAN
      FF=LONG-OMEGA
      DD=LONG-LONGS
C LONGITUDE FIRST
      DLONG=2369.912*SIN(2.*DD)+191.953*SIN(M+2.*DD)
      DLONG=DLONG+22639.500*SIN(M)-4586.465*SIN(M-2.*DD)
      DLONG=DLONG-38.428*SIN(M-4.*DD)-668.146*SIN(MS)
      DLONG=DLONG-165.145*SIN(MS-2.*DD)-125.154*SIN(DD)
      DLONG=DLONG+769.016*SIN(2.*M)-211.656*SIN(2.*M-2.*DD)
      DLONG=DLONG-30.773*SIN(2.*M-4.*DD)-109.673*SIN(M+MS)
      DLONG=DLONG-205.962*SIN(M+MS-2.*DD)+147.687*SIN(M-MS)
      DLONG=DLONG-411.608*SIN(2.*FF)-55.173*SIN(2.*FF-2.*DD)
      DLONG=DLONG+36.124*SIN(3.*M)-45.099*SIN(M+2.*FF)
      DLONG=DLONG+39.528*SIN(M-2.*FF)
      DLONG=DLONG-24.420*SIN(MS+2.*DD)+14.387*SIN(2.*M+2.*DD)
      DLONG=DLONG+14.577*SIN(M-MS+2.*DD)+28.475*SIN(M-MS-2.*DD)
      DLONG=DLONG+18.609*SIN(M-DD)+18.023*SIN(MS+DD)
      DLONG=DLONG-13.193*SIN(3.*M-2.*DD)
      DLONG=DLONG/SPR
      LONG=LONG+DLONG
C                               NOW THE LATITUDE
      XLAT=117.2608*SIN(FF+2.*DD)+18461.3493*SIN(FF)
      XLAT=XLAT-623.6553*SIN(F F-2.*DD)+1010.1724*SIN(M+FF)
      XLAT=XLAT-166.5729*SIN(M+FF-2.*DD)+199.4806*SIN(-M+FF+2.*DD)
      XLAT=XLAT-999.6848*SIN(-M+FF)-33.3628*SIN(-M+FF-2.*DD)
      XLAT=XLAT-29.6546*SIN(MS+FF-2.*DD)+61.9131*SIN(2.*M+FF)
      XLAT=XLAT-31.7627*SIN(-2.*M+FF)
      XLAT=XLAT+15.1194*SIN(M+FF+2.*DD)+12.1245*SIN(-MS+FF-2.*DD)
      XLAT=XLAT-15.5659*SIN(2.*M+FF-2.*DD)
      XLAT=XLAT/SPR
C                               THE PARALLAX
      DP=28.2333*COS(2.*DD)+3.0861*COS(M+2.*DD)
      DP=DP+186.5398*COS(M)+34.3117*COS(M-2.*DD)
      DP=DP+1.9178*COS(MS-2.*DD)+10.1657*COS(2.*M)
      DP=DP+1.4437*COS(M+MS-2.*DD)+1.1528*COS(M-MS)
      DP=DP-0.9781*COS(DD)-0.9490*COS(M+MS)
      DP=DP+0.6215*COS(3.*M)-0.7136*COS(M-2.*FF)
C
      X=COS(XLAT)
      XX(1)=X*COS(LONG)
      XX(2)=X*SIN(LONG)
      XX(3)=SIN(XLAT)
      X=1.+DP/3422.452
      ROVERA=1./X
C^^^^^^^^^
C                 DIR. COSINES N THE EQUATORIAL SYSTEM
C
      EPS=(23.452294-.0130125*T)/RADIAN
      SE=SIN(EPS)
      CE=COS(EPS)
      VECTOR(1)=XX(1)
      VECTOR(2)=XX(2)*CE-XX(3)*SE
      VECTOR(3)=XX(3)*CE+XX(2)*SE
      RETURN
C
      END
      COMPILER DOUBLE PRECISION
      COMPILER NOSTACK
      SUBROUTINE SUNVECT (T,X,R)
C
C     T= TIME IN SMITHSONIAN DAYS
C              X=VECTOR TO SUN REFERRED TO EQUINOX AND EQUATOR OF 1950.0
C              R=MAGNITUDE OF X VECTOR    (LENGTHS ARE IN A.U.)
C
      DIMENSION X(3)
C
      A=6.2291402+.0172021242*(T-39125.)
      E=A
      DO 1 J=1,6
    1 E=A+.016728643*SIN(E)
      SE=SIN(E)
      CE=COS(E)
      R=1.0000002-.016728643*CE
      X(1)=.20947919*CE+.97767652*SE-.003504301
      X(2)=-.89708519*CE+.19215772*SE+.015007012
      X(3)=.43367832*X(2)
      RETURN
      END

      COMPILER DOUBLE PRECISION
      SUBROUTINE EPHLUN(T,ELE,PERT,RNU)
C
      DIMENSION T(2),ELE(6),PERT(6),RNU(2)
C
C     VERSION TO INTERPOLATE GRIPE LUNAR PERTURBATION TABLE
C     INCORPORATED IN MARCH 26, 1981 BY SAEQA DIL
C
      COMMON/MOON/PM(8,12)
C
      J1=1
C
      TEMP=T(1)+T(2)
C
      IF(TEMP.LT.PM(1,1)) STOP TIME LESS THAN TABLE
      IF(TEMP.GT.PM(1,12)) STOP TIME GREATER THAN TABLE
C
  100 CONTINUE
C
      IF(TEMP.GE.PM(1,J1).AND.TEMP.LE.PM(1,J1+1)) GO TO 200
      IF(TEMP.GT.PM(1,J1+1) J1=J1+1
C
      GO TO 100
C
  200 CONTINUE
C
      DO 300 I=1,5
      TEMP1=(PM(I+1,J1+1)-PM(I+1,J1))/(PM(1,J1+1)-PM(1,J1))
      TEMP2=TEMP1*(TEMP-PM(1,J1)
      PERT(I)=TEMP2+PM(I+1,J1)
  300 CONTINUE
      DO 400 I=1,2
      TEMP1=(PM(I+6,J1+1)-PM(I+6,J1))/(PM(1,J1+1)-PM(1,J1))
      TEMP2=TEMP1*(TEMP-PM(1,J1))
      RNU(I)=TEMP2+PM(I+6,J1)
  400 CONTINUE
      RETURN
      END
```