

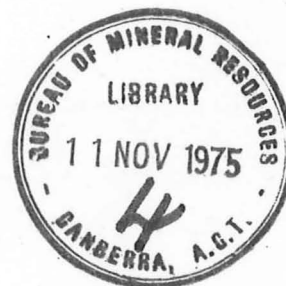
1975/60
COPY 4

DEPARTMENT OF
MINERALS AND ENERGY

056223



BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS



RECORD 1975/60

GEOPHYSICAL SURVEYS OF THE CONTINENTAL MARGINS OF
AUSTRALIA, GULF OF PAPUA, AND BISMARCK SEA -
SATELLITE FIX PROCESSING METHODS

by

R.A.P. GARNETT

The information contained in this report has been obtained by the Department of Minerals and Energy as part of the policy of the Australian Government to assist in the exploration and development of mineral resources. It may not be published in any form or used in a company prospectus or statement without the permission in writing of the Director, Bureau of Mineral Resources, Geology and Geophysics.

RECORD 1975/60

GEOPHYSICAL SURVEYS OF THE CONTINENTAL MARGINS OF
AUSTRALIA, GULF OF PAPUA, AND BISMARK SEA -
SATELLITE FIX PROCESSING METHODS

by

R.A.P. GARNETT

CONTENTS

	<u>Page</u>
SUMMARY	
1. INTRODUCTION	1
2. SATELLITE FIX DATA FILES	3
3. SATELLITE FIX COMPUTATION METHODS	5
4. TROPOSPHERIC REFRACTION CORRECTION	10
5. HEIGHT OF THE RECEIVER ABOVE THE REFERENCE SPHEROID	13
6. OFFSET FREQUENCY OF THE REFERENCE OSCILLATOR	15
7. DETERMINATION OF SHIP VELOCITY AS AN UNKNOWN VARIABLE	17
8. USE OF 2 MINUTE SHIP POSITIONS IN FIX COMPUTATION	20
9. CONCLUSION	24
10. REFERENCES	26

FIGURES

1. Listing of a satellite fix data file
2. Fix error due to error in ship velocity
3. Position plot of fixes in port at Newcastle
4. Position plot of fixes in port at Port Adelaide
5. Velocity comparisons for 33 fixes recorded over the shelf

APPENDICES

1. DRIVING INSTRUCTIONS FOR PROGRAM BMRFIX
2. SAMPLE LISTINGS OF PRINTOUTS FROM BMRFIX FOR EACH COMPUTATION ROUTINE OPTION
3. FORTRAN LISTING OF PROGRAM BMRFIX
4. SATELLITE LONG-TERM OFFSET FREQUENCY PLOTS

SUMMARY

The accuracy with which a ship's position may be calculated from satellite Doppler measurements depends primarily on the accuracy with which certain parameters are known or may be computed. These parameters are the height of the receiver above the reference spheroid, the offset frequency of the satellite transmitter, and the ship's surface velocity. They may be inserted as fixed values or computed as variables in the calculation of the fix position.

The height of the receiver, which varies only with geoid height, is known to sufficient accuracy. The offset frequency must be computed but the derived value contains errors due to the errors in the other fix parameters. Since these are random, however, and the transmitter frequency drift is approximately linear, an improved value is obtained by long-term smoothing of the offset frequency plot.

The greatest problem is to define the ship's velocity with sufficient accuracy, to the order of 0.1 knot if a positional accuracy of 0.1 nautical mile is to be achieved. Over the shallow waters of the continental shelf the absolute short-term accuracy of the sonar Doppler should be sufficient. In deep water, however, the effect of ocean currents is uncertain and may introduce a large error, in some cases exceeding 1 nautical mile.

Two approaches to the problem were considered. The first was to treat both ship's velocity components as unknown variables. This produced erratic results of poor accuracy, but a considerable improvement was made by keeping fixed the least sensitive variable, easterly velocity. Results were still far below the accuracy of a stationary fix, however, and it is probable that the use of shorter and more numerous counts will be necessary to effect an improvement.

The second approach was initially to improve the accuracy of the fix by defining the motion of the ship from the navigation data as a series of relative positions every 2 minutes of time rather than as an average velocity, thus accounting for changes in course or speed during the fix. The remaining error due to ocean currents may then be reduced through an iterative process. The navigation data are tied to the fix positions and, thus improved, are used to recalculate the fixes to which they are again fitted. One iteration should be sufficient to achieve an accuracy of 0.1 nautical mile.

The latter method increases the time and cost of reducing the navigation data, but is the only way in which the desired accuracy may be achieved with the present data.

1. INTRODUCTION

The Fortran program presently used by BMR to compute satellite fixes is called BMRFIX and derives originally from a program acquired from Ray Geophysics in 1968. This program initially performed only the basic fix computation, but it was extensively modified by R. Whitworth to include a variety of other facilities, and restructured into a simpler and more comprehensible form. The major modifications include a routine for computing the effects on fix positions of changes in ship's velocity and geoid height and another routine for eliminating Doppler counts systematically until the root mean square (r.m.s.) of the ship-satellite range residuals reduces to an acceptable value.

Between August 1973 and August 1974, investigations were made into the various techniques available for computing satellite fixes with a view to choosing the technique that would maximize the accuracy of the navigation data acquired during the Continental Margin Survey around Australia. About 9000 satellite passes were recorded at sea during the survey, and those which are reliable will be used to tie-down the dead-reckoning navigation systems. The time interval between good fixes may range from half an hour to over five hours but is usually about two hours.

The position from a single good satellite fix on land may now be determined to an accuracy of 20-40 metres r.m.s. (Black, 1973) and considerably better when multiple fixes are taken at the same position. On a moving vessel, however, the accuracy of a fix is reduced by any uncertainty in the motion of the receiver during the time period through which it may record signals from the satellite (up to 16 minutes) and by the fact that the fix cannot be repeated. As a result of these and other factors the position obtained from a fix of poor quality at sea may be in error by half a nautical mile or more.

Sluiter (1969) has outlined the effects of a variety of factors on the accuracy of a satellite fix, and some of the routines in BMRFIX

derive from his results. In this Record the developments and tests which have led to the establishment of the program in its present form are described and discussed. The way in which this program will be used in attaining the final positional data is outlined, and user driving instructions together with a listing of the program itself are included in Appendices 1 and 3 respectively.

It is not within the purpose of this Record to describe the Transit satellite navigation system or the basic theory of fix computation; that is described by Stansell (1968) and Regnaudin (1969). However, certain parts of the computation theory are explained where they are directly relevant to the topics discussed. Chapter 3 includes a detailed explanation of the method by which a solution to a satellite fix is obtained by an iterative process, but a knowledge of the basic principles of satellite Doppler measurements is assumed.

SEGMENT *17.131350

1	*	8243928.0					
2	*	3724457.0					
3	*	1645167.0					
4	*	19847.0					
5	*	5125.0					
6	*	745416.0					
7	*	605541.0					
8	*	-553.0					
9	*	13398.0					
10	*	1396055.0					
11	*	939910.0					
12	*	3025540.0	2000.0				
13	*	3161416.0	1996.0				
14	*	3413126.0	1992.0				
15	*	3795844.0	2008.0				
16	*	4197033.0	2010.0				
17	*	4479171.0	2005.0				
18	*	4634322.0	2008.0				
19	*	4711336.0	2000.0				
20	*	29.0	304.0	-15.0			
21	*	40.0	304.0	4.0			
22	*	52.0	295.0	-10.0			
23	*	61.0	278.0	-0.0			
24	*	69.0	253.0	-4.0			
25	*	74.0	223.0	-15.0			
26	*	76.0	190.0	3.0			
27	*	75.0	154.0	-25.0			
28	*	72.0	118.0	9.0			
29	*	64.0	84.0	46.0			
30	*	54.0	54.0	14.0			
31	*	1350	28.0S	114E	328.0	9.0	-36.6

LISTING OF A SATELLITE FIX DATA FILE

2. SATELLITE FIX DATA FILES

The basic data necessary for the computation of each satellite fix are stored in card image format on magnetic tape. Each file, which consists of 32 card images, contains the data for one satellite fix, and all the fixes for a cruise are stored on a single magnetic tape. There are 36 tapes in all.

The data are formatted in the following way.

File header	:	The lock-on time in survey time format SS.DDHHMM (showing survey number, survey day, GMT hour and minutes)
Cards 1 - 11	:	Each contains a fixed parameter of the satellite orbit.
Cards 12 - 19	:	The Doppler and refraction counts for each 2-minute period.
Cards 20 - 30	:	These contain the variable correction data for the satellite orbit.
Card 31	:	This card contains manually inserted values for lock-on time, approximate latitude and longitude, ship's speed and heading, and the geoid height.

Figure 1 shows a listing of the data for a typical fix.

The data tapes were produced in this format on a CDC 3600 computer from the original field tapes and later converted to a structure compatible with the CDC 6600. Fairly extensive editing was required on these 6600 tapes before computation of the fixes could continue. The most commonly occurring errors may be summarized as follows.

- (1) Frequently the same fix might appear on the tape two or more times sequentially. This stemmed from the original on-board computation when an attempt was made to improve the fix position by deleting suspect counts or manually changing input data before recalculating the fix. In all cases the fix with the most complete data was retained and the others deleted.
- (2) Owing to an error in re-formatting the data during the making of the 3600 tapes, one of the orbital parameters occasionally overflowed its

format and filled the word with stars. These were replaced after referring to original listings.

(3) Most of the other errors originated from the field tapes and usually occurred in the last card. Corrections were made by referring to the on-board teletype listings, which by necessity were assumed to be correct. It was also found quite often that data from one fix would be mixed up with those from the following fix. This usually applied to the last few cards of the fix and frequently resulted in data being out of format in the last card.

All these errors were corrected in the data tapes using edit cards and program NFILE. This program produces a new data tape with the corrections incorporated in it. It is thought that all the worst errors have been corrected by this process; that is, all those which either make the fix impossible to compute or produce a fix position which is obviously in error. However, there may still be minor errors in the data which have a less obvious effect on the fix position, and these could be found only by detailed comparison of the data tape listings with the original teletype listings, a process which would be prohibitively lengthy. It is expected that examination of the navigation during the later phases of the processing will reveal which fixes still contain data errors.

If it is necessary to make alterations or corrections to the fix data for the computation only, without altering the data files, it may be achieved by using edit cards during runs of BMRFIX. Subroutine EDITFIX has been written into the program for this purpose; it enables any card in a data file to be changed discretely.

3. SATELLITE FIX COMPUTATION METHODS

The basic method by which fixes are computed in EMRFIX is one of iteration. Each Doppler count, after correction for the offset frequency (the frequency difference between the satellite transmitter and the reference oscillator on the ship), gives an actual measurement of the change in ship-satellite range during the period of that count. If approximate values are then assumed for the latitude and longitude of the ship's position (say to the nearest degree) it is possible, knowing the path of the satellite, to compute the theoretical range change during each count. The differences between the theoretical and actual range changes are called residuals, and from these residuals adjustments to the estimated position may be calculated. The adjustments are applied and the process repeated in an iterative loop which stops only when the adjustments fall below a specified value or twenty iterations have been completed without reaching this value.

In this way the approximate values which are initially given to the unknown variable parameters are systematically improved to the required precision. These variables will always include latitude and longitude, but there are other parameters which are not always known accurately and which it may be useful to treat as unknown variables in order to derive values. Such parameters include the offset frequency, the geoid height, and the north and east components of ship velocity during the fix. It will be shown later how the solution to a fix is derived by first producing a number of independent linear equations that equal the number of 2-minute Doppler counts recorded, up to a maximum of eight. The unknowns in these equations are the adjustments to the unknown variable parameters. In order to solve these equations, therefore, the number of variable parameters used in a computation must be less than or equal to the number of equations and thus to the number of counts recorded.

The various stages of the fix computation will now be explained in detail.

(1) The positions of the satellite at the start and finish of each 2-minute count are calculated from the orbital parameters and the variable correction terms. This is done in subroutine SETXYZ while the standard corrections to the Doppler counts for the ionospheric refraction effect (Halamandaris & Gilbert, 1971) are made in subroutine SETDATA.

(2) Subroutines POS1, POS2, POS3, and POS4 perform the actual fix computations, each for a different set of variable parameters. Any one of these routines may be selected at the option of the user.

POS1 - latitude, longitude.

POS2 - latitude, longitude, offset frequency.

POS3 - latitude, longitude, geoid height.

POS4 - latitude, longitude, ship's northerly velocity.

Typical printouts from each of these routines are shown in Appendix 2. It is quite possible to derive a routine which includes the ship's easterly as well as its northerly velocity as a variable, or for that matter any other combination which may be desired. In order to follow through the computation POS2 will be considered.

(3) Using the approximate values of latitude and longitude the theoretical position of the ship at the beginning and end of each count is derived. At these times the theoretical ship-satellite range can then be found. The range is corrected for the tropospheric refraction effect (see Chapter 4) and the change in range during each count obtained. These values are then subtracted from the actual change in range found from the Doppler count, giving the residual at each count, $DL_1, DL_2 \dots DL_n$, where n is the number of counts taken during the fix.

(4) Finding a solution to the fix means finding the adjustments to the approximate values of the variable parameters which will reduce the residuals to a minimum. Therefore equations must be set up relating the residuals to the errors in approximate latitude, longitude, and offset frequency - $\Delta\phi, \Delta\lambda$

and Δf . Let the ship-satellite range at the start of count k be related to the three variable parameters by the function

$$R_k = F_k(\phi, \lambda, f)$$

If each variable is written as an approximate value plus an error term we have,

$$R_k = F_k(\bar{\phi} + \Delta\phi, \bar{\lambda} + \Delta\lambda, \bar{f} + \Delta f)$$

Assuming the errors to be small this may be expressed to first-order accuracy (Whittaker & Robinson, 1924) as

$$R_k = F_k(\bar{\phi}, \bar{\lambda}, \bar{f}) + \frac{\partial F_k}{\partial \phi} \Delta\phi + \frac{\partial F_k}{\partial \lambda} \Delta\lambda + \frac{\partial F_k}{\partial f} \Delta f$$

Now the difference between the measured change in range during the k th count and that computed from the approximate values may be expressed from the above equation and is equivalent to the residual DL_k .

$$\begin{aligned} DL_k &= [R_k - R_{k+1}] - [F_k(\bar{\phi}, \bar{\lambda}, \bar{f}) - F_{k+1}(\bar{\phi}, \bar{\lambda}, \bar{f})] \\ &= \left(\frac{\partial F_k}{\partial \phi} - \frac{\partial F_{k+1}}{\partial \phi} \right) \Delta\phi + \left(\frac{\partial F_k}{\partial \lambda} - \frac{\partial F_{k+1}}{\partial \lambda} \right) \Delta\lambda + \left(\frac{\partial F_k}{\partial f} - \frac{\partial F_{k+1}}{\partial f} \right) \Delta f \end{aligned}$$

We may simplify this equation by substituting for the change in each partial differential during a count as follows:

$$C_{1k} = \frac{\partial F_k}{\partial \phi} - \frac{\partial F_{k+1}}{\partial \phi} \quad C_{2k} = \frac{\partial F_k}{\partial \lambda} - \frac{\partial F_{k+1}}{\partial \lambda} \quad C_{3k} = \frac{\partial F_k}{\partial f} - \frac{\partial F_{k+1}}{\partial f}$$

Values for these three terms can be calculated for each count using the approximate values of the variable parameters. (They are of course also approximate but become increasingly accurate as the iteration converges).

(5) We can now establish a linear equation for each count.

$$DL_k = C_{1k} \Delta\phi + C_{2k} \Delta\lambda + C_{3k} \Delta f \quad \text{for } k = 1 \text{ to } n$$

If there are more equations than there are unknowns there will not usually

be a unique solution since there are inherent errors in the equations. These errors derive not only from the approximations made in the development of the equations but also from errors in the definition of the satellite path and in the measurement of the Doppler counts. In fact errors are likely to derive from any parameter which is not treated as a variable. An optimum solution must therefore be obtained such that the sum of the squares of the errors in all the equations is a minimum. Substituting x , y , and z for $\Delta \phi$, $\Delta \lambda$, and Δf the error in the above equation may be expressed as

$$E_k = C_{1k} \cdot x + C_{2k} \cdot y + C_{3k} \cdot z - DL_k$$

The sum of the squares of the errors is

$$S = \sum_{k=1}^n E_k^2 = \sum_{k=1}^n (C_{1k} \cdot x + C_{2k} \cdot y + C_{3k} \cdot z - DL_k)^2$$

For S to be a minimum the following condition must hold:

$$\frac{\partial S}{\partial x} = 0, \quad \frac{\partial S}{\partial y} = 0, \quad \frac{\partial S}{\partial z} = 0$$

Partially differentiating with respect to each variable we obtain three equations:

$$\frac{\partial S}{\partial x} = \sum C_{1k}^2 \cdot x + \sum C_{1k} \cdot C_{2k} \cdot y + \sum C_{1k} \cdot C_{3k} \cdot z - \sum C_{1k} \cdot DL_k = 0$$

$$\frac{\partial S}{\partial y} = \sum C_{2k} \cdot C_{1k} \cdot x + \sum C_{2k}^2 \cdot y + \sum C_{2k} \cdot C_{3k} \cdot z - \sum C_{2k} \cdot DL_k = 0$$

$$\frac{\partial S}{\partial z} = \sum C_{3k} \cdot C_{1k} \cdot x + \sum C_{3k} \cdot C_{2k} \cdot y + \sum C_{3k}^2 \cdot z - \sum C_{3k} \cdot DL_k = 0$$

Since the number of equations now equals the number of unknowns, a unique solution may be found which will fit the least-squares condition. The solution is most easily found by matrix inversion. Substituting back for the variables $\Delta \phi$, $\Delta \lambda$, and Δf we may express the equations in the following matrix form:

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \times \begin{bmatrix} \Delta \phi \\ \Delta \lambda \\ \Delta f \end{bmatrix}$$

where $B_i = \sum_{k=1}^n D L_k \cdot C_{ik}$

and $A_{ij} = \sum_{k=1}^n C_{ik} \cdot C_{jk}$

The solution to the unknowns $\Delta \phi$, $\Delta \lambda$, and Δf is found by inverting matrix A and multiplying it with matrix B. Subroutine MATINV performs the inversion and multiplication, and returns the solutions so long as the matrix A is not singular.

The errors $\Delta \phi$, $\Delta \lambda$, and Δf are added to the original values of latitude, longitude, and offset frequency respectively, and processes (3) to (5) are repeated. This iterative process is necessary because of the approximations assumed in deriving the original linear equations which are accurate only for infinitely small differences in ϕ , λ , and f . Initially these differences are too large to be of sufficient accuracy, but as the computation is repeated they progressively reduce until they become smaller than the precision desired of the derived variables. The values of the variables at this point will be of the required precision. If this does not occur after twenty iterations the process is terminated with an informative message, but convergence is normally obtained in no more than four iterations.

4. TROPOSPHERIC REFRACTION CORRECTION

The Doppler shift observed by a receiver on the Earth's surface in a signal transmitted from a satellite is not as it would be in a vacuum. The path and the speed of transmission of the signal are affected by variation in the refractivity of the atmosphere, and the error may be considered as comprising two discrete components. One is the component due to refraction in the ionosphere, which depends on the frequency of the transmitted signal, and the other is that due to tropospheric refraction, which is not frequency dependent. The first may be computed from Doppler counts taken at two different frequencies, but the second must be computed independently.

Tropospheric refraction depends on the density and humidity of the air through which it passes. The Earth's atmosphere is assumed to vary only with height, but even so the full expression for range correction is still complex and heavily dependent on high computer precision and memory capacity for its computation. Perhaps the simplest approximation to this expression has been put forward by Hopfield (1969) and it is this approximation which is currently used in function TROPO to compute the tropospheric range correction in BMRFIX.

The total range correction $\Delta \rho_{tro}$ is considered to be the sum of the so-called 'dry' and 'wet' components, here subscripted d and w:

$$\Delta \rho_{tro} = (\Delta \rho_{tro})_d + (\Delta \rho_{tro})_w$$

The approximations for these terms are:

$$(\Delta \rho_{tro})_d = K_d P \operatorname{cosec} \sqrt{E^2 + \theta_d^2} \text{ km}$$

$$(\Delta \rho_{tro})_w = K_w \operatorname{cosec} \sqrt{E^2 + \theta_w^2} \text{ km}$$

E is the elevation of the satellite from the receiver and θ_d and θ_w are

empirical parameters (in the same units as E). The above expressions give very good approximations of the components for elevation above 5° , and quite good ones as low as 2° , with the following parameter values.

$$\theta_d = 2.5^\circ$$

$$\theta_w = 1.5^\circ$$

P is the observed local pressure at antenna height, and K_d is a constant whose value is known to quite high precision. Its best current value is

$$K_d = 2.278 \times 10^{-8} \text{ km/millibars}$$

K_w varies with latitude, season, and weather and a value may be chosen according to the following list of observed averages.

K_w	Place, time
$0.28 \times 10^{-3} \text{ km}$	Tropics or mid-latitude summer
$0.20 \times 10^{-3} \text{ km}$	Mid-latitude spring or fall
$0.12 \times 10^{-3} \text{ km}$	Mid-latitude winter
$0.05 \times 10^{-3} \text{ km}$	Polar regions

On EMR surveys up till now the recording of satellite fixes has not been accompanied by the measurement of atmospheric pressure so that the following approximations are made on the basis of average sea-level pressure

$$K_d P = 2.31 \times 10^{-3} \text{ km}$$

$$K_w = 0.20 \times 10^{-3} \text{ km}$$

The range correction for a single ray path is about 2.5 metres at 90° elevations and 90 metres at the horizon. Over a complete fix this has the effect of shifting the position of the tracking station about 20 m towards the orbit along the slant-range vector for a high pass, and about 80 m for a 15° pass. If the height of the station above the geoid is

fixed, the resultant horizontal shift in position becomes about 80 m for a 75° pass and the same for a 15° pass. An error of 1 percent in the pressure P affects the total range correction by about 1 percent, and an error of 0.1×10^{-3} in the value of K_w affects it by about 4 percent.

Hopfield estimates that if the pressure P is known the residual error in the total correction will be less than 10 percent and this almost entirely due to the wet component. If the pressure is not measured and may thus be in error by up to 3000 millibars the induced error may be increased by about 3 percent. Except for extremely high or low passes the position error attributable to tropospheric correction in this case will still not exceed about 12 metres.

This error is negligible compared with other errors inherent in taking fixes from a moving vessel. The approximate equations used in BMRFIX would therefore seem to be quite adequate. It would be advisable in future, however, to measure atmospheric pressure at the time of a fix in order to eliminate any error due to the dry component. It may also be possible to choose a more accurate value for K_w according to the table shown above instead of simply using a fixed average value.

If it is considered necessary to reduce the error in the tropospheric correction even more an algorithm derived by Yionoulis (1970) may be used. In order to give high accuracy over all elevation angles there are in fact two expressions, one for high angles and one for low with a considerable area of overlap. The algorithms are in the form of numerical summations which may be taken to the number of terms sufficient to give the desired accuracy. This method eliminates the necessity for double-precision computation and enables the calculation to be carried out on a computer without introducing rounding errors.

5. HEIGHT OF THE RECEIVER ABOVE THE REFERENCE SPHEROID

Any position on the Earth's surface may be defined by reference to a spheroid of best fit. Since the geoid is an irregular figure such a position cannot be defined by latitude and longitude alone, but must also be specified by its radial distance from the spheroid surface. The radial distance of the geoid from the reference spheroid is called the geoid height, which must be known in order to solve a satellite fix.

Several spheroids have been derived from attempts to fit either the geoid as a whole or only a particular part of its surface. The APL (Applied Physics Laboratory) spheroid with an equatorial radius of 6378144.0 metres and a flattening of $1/298.23$ gives an optimum worldwide fit to the geoid but the difference over the Australian region shows a significant north-south gradient. An Australian National spheroid has been developed which is biased to give best fit to the Australian region. It has an equatorial radius of 6378160.0 metres, a flattening of $1/298.25$ and an origin shift from the APL spheroid of -115, -42, and 147 metres in the X, Y, and Z directions respectively. No significant rotational difference between these two spheroids has yet been detected.

If the same position on the geoid is referred to different spheroids the latitude and longitude of that position in each case will be different. In the case of the APL and the Australian National spheroids the differences approximate to 5 seconds of latitude and 4 seconds of longitude over the Australian region but may be computed precisely for any position. The Transit satellite navigation system is based on the APL spheroid and therefore it is convenient to calculate positions which are referred to that system.

Geoid heights for the APL spheroid have been obtained at one-degree intervals over the whole Australian region. These heights are now used in the computation in preference to the geoid heights originally written onto the satellite data files by the Compagnie Generale de Geophysique (CGG) and whose source is somewhat obscure. A comparison of the two

revealed differences as much as 50 metres in places. The APL geoid heights are stored on a file which may be accessed by BMRFIX to give the geoid height at any position by interpolation.

Subroutine POS3 derives the geoid height by treating it as an unknown parameter together with latitude and longitude, but initial tests indicate that as a variable it is too sensitive to changes in range, and the derived value too inaccurate for it to be of any use. It appears that with a good fix of relatively high altitude the computed geoid height may be several hundred metres in error, while with poorer, low-altitude fixes the error may rise to a few thousand metres. Comprehensive tests have yet to be carried out, but it would seem that as a method of determining the height of the geoid, even by filtering over large areas, this technique holds little promise for fixes taken at sea.

6. OFFSET FREQUENCY OF THE REFERENCE OSCILLATOR

The Doppler shift of the satellite transmitter is measured by reference to a stable oscillator at the receiving station. It is thus vitally important to know with considerable accuracy the difference in frequency between the two oscillators. This is called the offset frequency and is approximately 32000 Hertz. It will, however, vary from satellite to satellite and change with time as the oscillator frequencies drift.

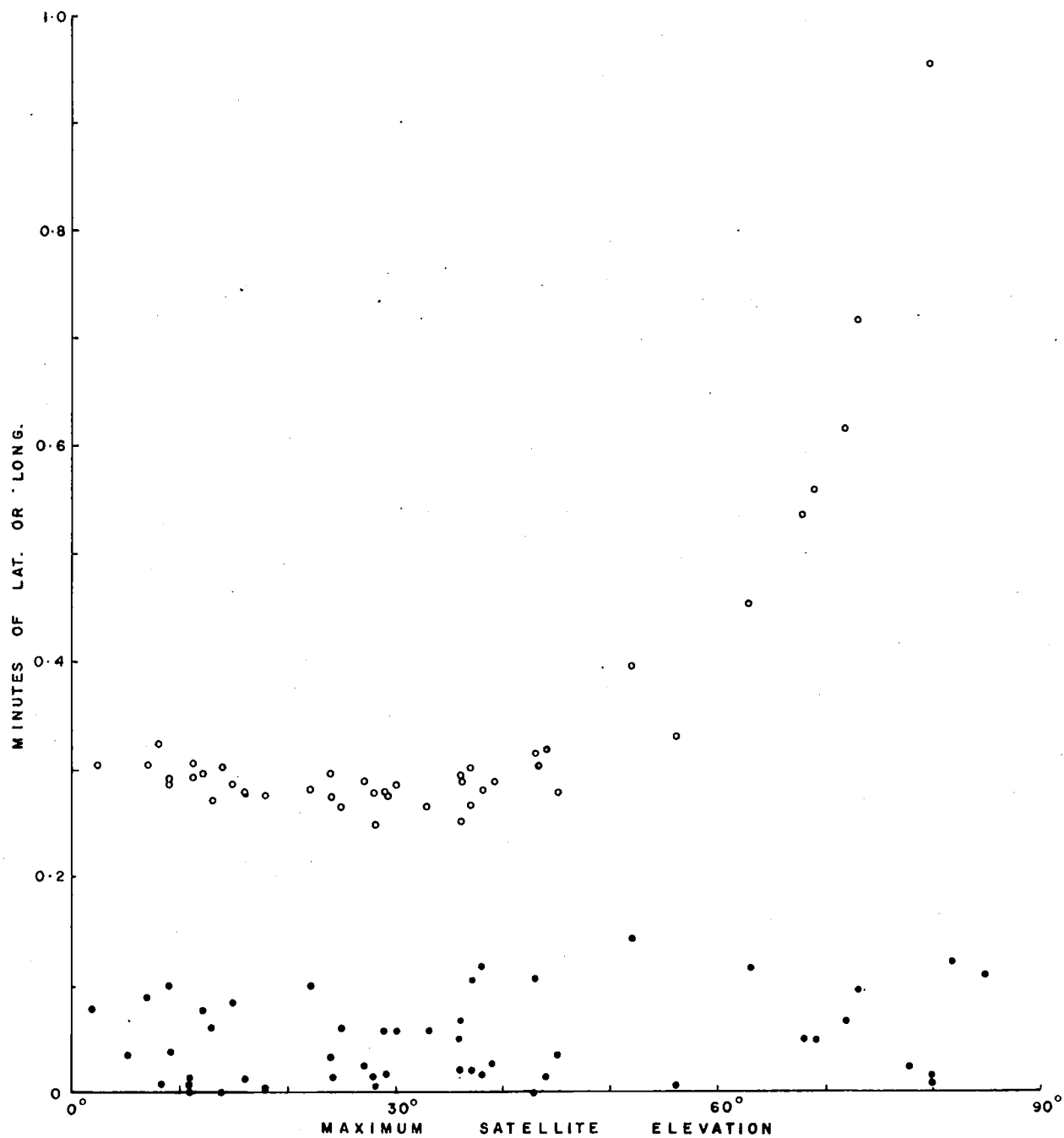
One way of obtaining the offset frequency at a particular fix is to let it be an unknown variable with latitude and longitude in the computation. Since it is a very sensitive parameter the resulting value will be of considerable accuracy if all the parameters are known to a similar accuracy. But if there are errors in the ship's velocity or the geoid height or in the positional data of the satellite itself, these errors will be reflected in the derived value of the offset frequency.

Where the computed offset frequencies from a series of fixes taken from a particular satellite are plotted against time it becomes apparent that the long-term drift is virtually linear (Appendix 4). The scatter of individual fixes about this line may be attributed to the errors in other parameters as mentioned above or to the poor quality of a fix due to an unfavourable pass altitude or an insufficient number of Doppler counts.

It is generally considered that the oscillators themselves have an extremely long drift period, and although occasional jumps may occur there will in general be no short-period variation of the offset frequency. On this basis it has been suggested by Sluiter (1969) that a more accurate value of offset frequency would be that taken from the long-term drift curve. Statistical analysis of repeated fixes at a known position shows that there is indeed a significant reduction of scatter when such a procedure is followed.

Accordingly the offset frequencies at all fixes were computed and plotted against time for each satellite separately. Long-term drift

curves were then hand-drawn through each plot over the whole length of the survey and values taken off at intervals of five days. In one or two instances the drift curve of a particular satellite appeared to change unusually rapidly for a few days and it was decided not to use the satellite over the length of time that such erratic behaviour remained apparent. All other values were compiled into a file which may be accessed from BMRFIX and the offset frequency for a satellite at a required time determined by linear interpolation between values.



FIX SHIFT PER KNOT ERROR IN NORTHERLY SHIP VELOCITY

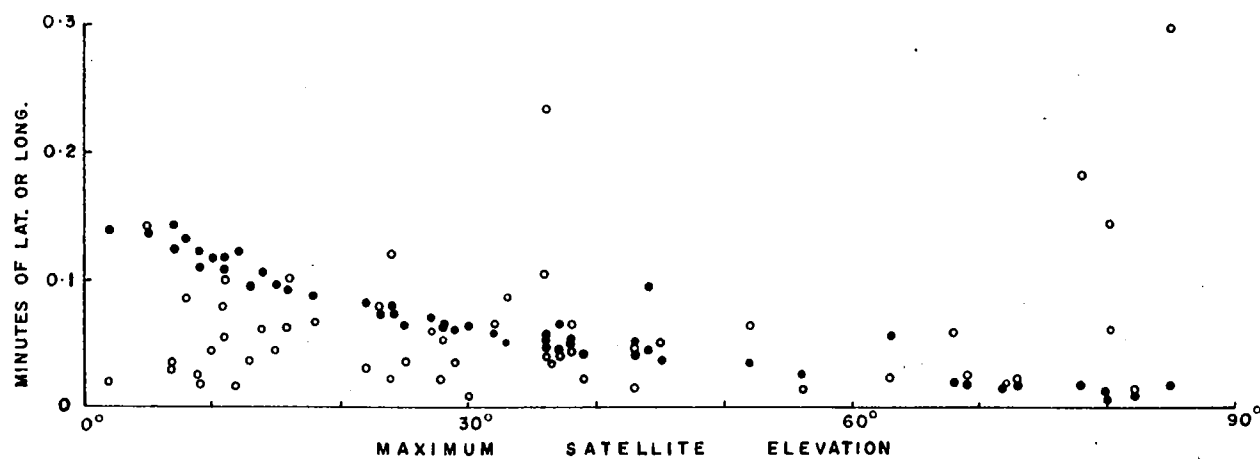


Fig.2 FIX SHIFT PER KNOT ERROR IN EASTERLY SHIP VELOCITY

- Modulus of latitude change
- " " longitude "

7. DETERMINATION OF SHIP VELOCITY AS AN UNKNOWN VARIABLE

Possibly the most important factor in the determination of satellite fixes from a moving vessel is the accuracy with which the velocity of that vessel over the Earth's surface is known. Sluiter (1969) has outlined the theoretical effect of velocity errors on a fix position for various headings and various satellite elevations. In general the response is such that errors in northerly velocity have the greatest effect and predominantly on the fix longitude, whereas easterly velocity errors have a smaller effect mainly on the latitude of the fix. The error due to northerly velocity reaches a minimum at 45° satellite elevation while that due to easterly velocity reduces continuously with satellite elevation. The magnitude of the effect is such that a 1-knot error in northerly velocity will alter the position derived from an average five-count fix by about 0.25 nautical miles for a 15° elevation, 0.2 n.m. for a 45° elevation and 0.5 n.m. for a 75° elevation.

These theoretical velocity error ellipses appear to be confirmed in practice. An error ellipse for each fix is computed in BMRFIX, and results for about 50 fixes have been compiled by Hudspeth (pers. comm.) and are displayed in Figure 2. They demonstrate the close relation of northerly velocity with longitude and easterly velocity with latitude, as maximum satellite elevation changes. Cross relations, however, appear largely erratic.

Determination of the ship's velocity is therefore a vital factor in the computation of a fix, and an accuracy of 0.1 knot should be aimed for if navigation is to be good to within 0.1 n.m. Ship's speed may in fact be measured in continental shelf areas to such a precision using sonar Doppler and velocity logs. In deep-water areas where sonar Doppler cannot read off the water bottom the velocity component due to ocean currents introduces an unknown quantity. As these currents may reach 2 knots or more the effect on a satellite fix position can be considerable.

One way of obtaining an absolute ship velocity is to make it an unknown variable in the fix computation, or rather two variables, northerly and easterly velocity. As long as at least four counts are used it is possible to solve for latitude, longitude, and both velocity vectors.

In order to assess the value of such a method tests were done on a series of fixes recorded in port, where the correct velocity is of course zero. Ten fixes at Newcastle and eleven fixes at Port Adelaide were computed in this way and the positions were plotted (Figs. 3, 4). It should be noted that in most fixes more counts were taken during the decline of the satellite than during its ascent due to late lock-on but otherwise they represent an average selection as regards satellite elevation and number of counts recorded.

All the Adelaide fixes and six out of ten of the Newcastle fixes lay within 0.2 n.m. of the correct position but the scatter otherwise was wide. It was also apparent that easterly velocity was the most insensitive of the variables, with occasional errors in the calculated value of 10 knots or more. Since the errors in the easterly vector seemed in general to be larger than most errors expected from ocean currents it was decided to try northerly velocity only as a variable with easterly velocity fixed as the dead reckoned value.

The positions derived by this method are also shown in Figures 3 and 4. As expected the predominant position error is in longitude but all the Newcastle fixes are within 0.2 n.m. of the correct position while the Adelaide fixes are all within 0.1 n.m.. These results are encouraging but are of course improved by the fact that the easterly velocity was correct in this case since it was known exactly. Greater divergency would be expected with errors in the easterly velocity.

Computation with both velocity components as variables therefore seems impracticable since reasonable results are obtained only from fixes of very good quality with at least six counts. The application of

FIX	SAT No.	SAT ELEV.	No. OF COUNTS
1	14	41°	7
2	13	51°	6
3	14	25°	6
4	15	42°	6
5	14	47°	8
6	13	47°	8
7	15	25°	6
8	13	51°	7
9	12	50°	7
10	14	35°	8
11	18	35°	5

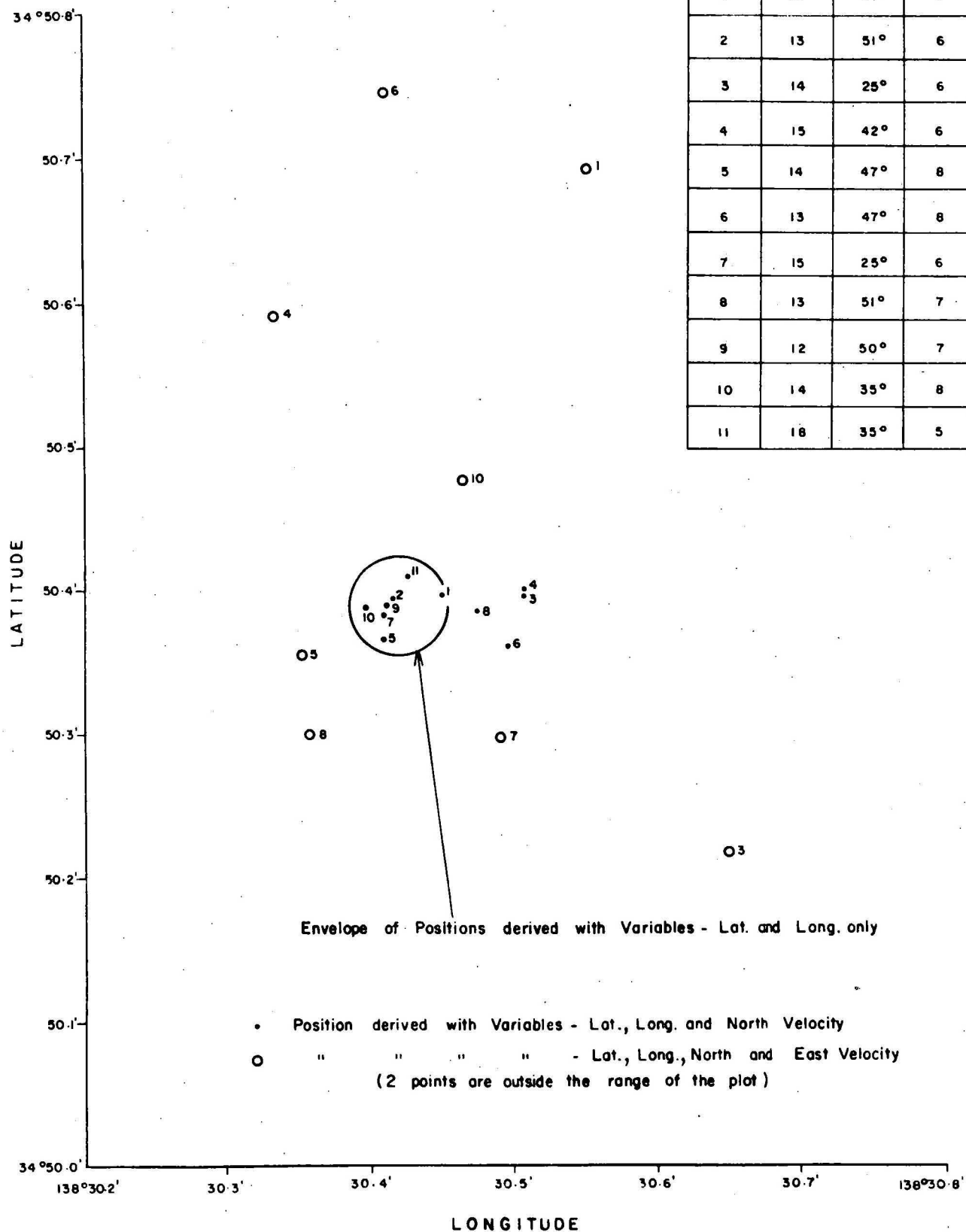


Fig. 4 FIXES IN PORT ADELAIDE (16.280534 - 16.300746)

northerly velocity alone as a variable yields considerably improved results but even good fixes may give a velocity up to 0.5 knot in error. In deep ocean areas of strong currents this latter technique might be usefully applied for on-board computation, but for post-processing of the navigation there are better methods of obtaining accurate velocities. These are described in the next chapter.

Serious consideration should be given to the length of time over which a Doppler count is recorded. At present counts are of two-minute duration, which permits no more than eight counts during a single pass. By reducing the length of each count and thereby increasing their number it may well be possible to increase the statistical reliability of a computed position since the ratio of equations to unknowns will be improved. Some reduction in the accuracy of each individual count must be expected, but it is quite likely that a more favourable compromise can be found. This in turn may cause some improvement in the determination of unknown variables, especially ship's velocity, the accuracy of which appears to depend heavily on the number of counts used in its computation. A count duration of 24 seconds has been found to give good results but it is thought that optimum precision may yet be achieved using counts of 20 seconds or less (Dennis, 1974).

8. USE OF 2-MINUTE SHIP POSITIONS IN FIX COMPUTATION

If the components of ship velocity are not to be treated as variable it is necessary to decide how the motion of the ship may best be defined. The values for the course and speed of the ship which are stored on the satellite data files are estimates taken on board from the dead-reckoning navigation systems and put in manually at the time of each fix. There was naturally no allowance for the effect of ocean currents in deep water, but even on the continental shelf there is considerable doubt as to the accuracy of these values.

A sequence of 33 fixes taken on the shelf in the Great Australian Bight was examined to establish an order of magnitude for the velocity errors. Since the sonar Doppler signals reflect from the bottom in these waters, an absolute velocity, acquired by averaging over the period of the fix (16 minutes), should be accurate to 0.1 knot. The difference between sonar Doppler velocity recorded in the general survey data files and that manually inserted on satellite data files was plotted for each fix (Fig. 5). The differences almost all lie within a 1-knot radius and are fairly evenly scattered. The inaccuracy of the values manually inserted in the satellite files is clearly evident and hard to explain since the same sonar Doppler information was available at the time of each fix.

The opportunity was also taken to make some assessment of the effectiveness of variable velocity computation of fixes recorded at sea. Since correct positions for each fix were not known a comparison of the derived velocities was made with the relatively accurate sonar Doppler velocities.

When both northerly and easterly velocity were treated as variable

Difference between Ships Velocity from the Sonar Doppler
and the Velocity on the Satellite Data Files

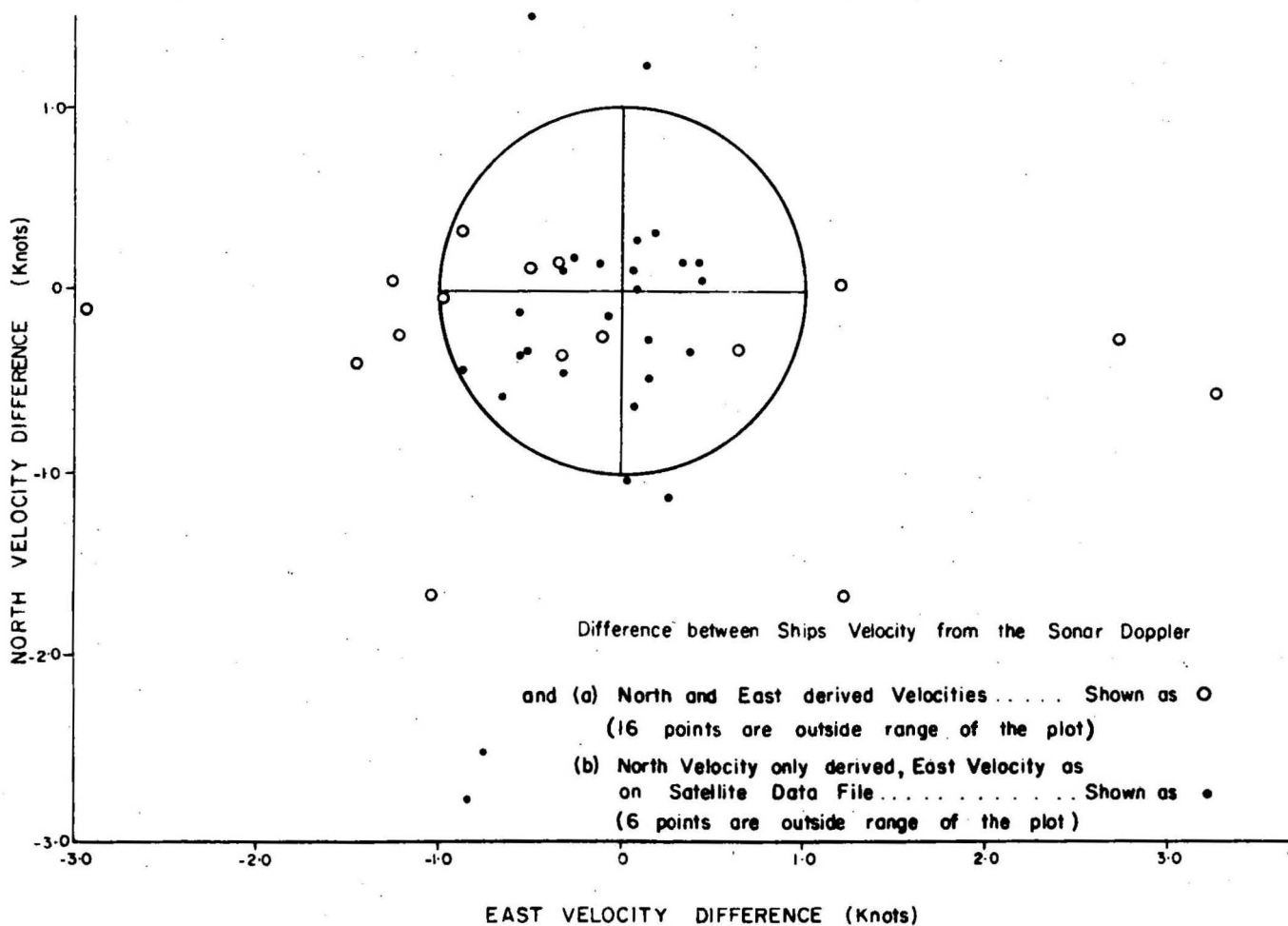
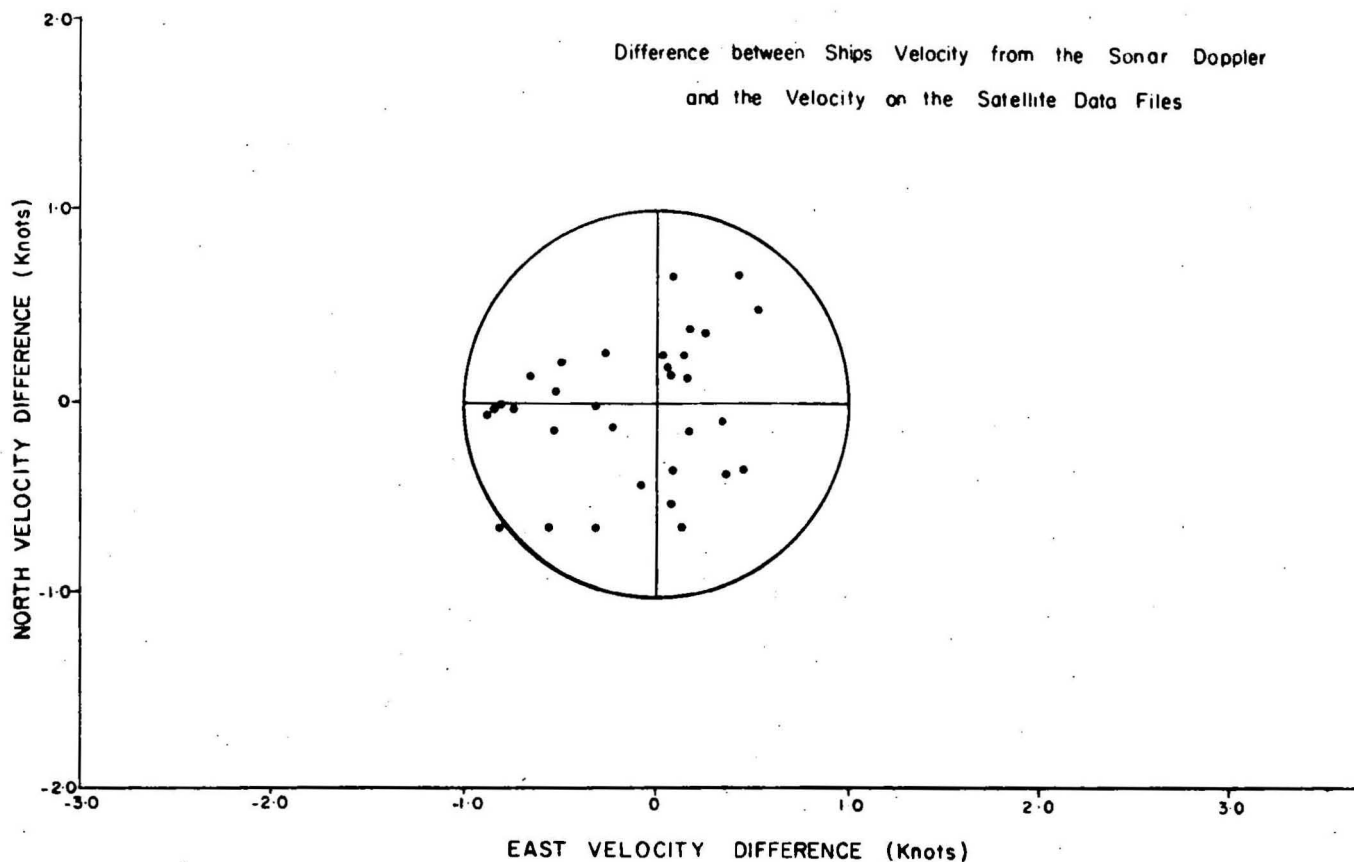


Fig. 5

VELOCITY COMPARISONS FOR 33 FIXES
RECORDED OVER THE SHELF
(16.750238 - 16.761656)

the differences with sonar Doppler velocity were extremely large as might be expected, especially the easterly component. With northerly velocity variable and the easterly component fixed as that on the satellite data file there is a considerable improvement in the scatter. Apart from the obviously bad values due to very poor fixes, where counts are few or elevations unfavourable, the errors in northerly velocity compare favourably with the errors in the corresponding velocities on the satellite data file.

In areas of deep ocean with strong current, such a method of acquiring velocity might prove as good as or better than simply using the value from the sonar Doppler, but overall the latter would seem preferable. Velocity derived from the sonar Doppler or one of the other dead-reckoning systems is most conveniently obtained from the one-minute sample data tapes. They can be set up as explained below to provide dead-reckoned positions whose absolute accuracy may be very low but whose relative accuracy over the period of a fix will be high.

Given such data it is then possible to improve the fix computation technique. The essential requirement for the computation is that the relative position of the ship be known at the beginning and end of each count. It is normally valid to assume that the velocity remains constant throughout the fix, in which case the ship's position varies linearly with time, but if significant changes in speed or course occur during the 16-minute fix period a non-linear error will be introduced into the residuals which may considerably reduce the accuracy of the fix. But if we obtain the relative position of the ship at the start and finish of each count directly from the dead-reckoning navigation, any variation of ship velocity will be automatically accounted for in the calculation.

The FIXNAV tapes which contain survey data at 1-minute intervals

also contain dead-reckoned latitudes and longitudes at these times. Subroutine VELDATA in EMRFIX reads off these positions over the period of a fix and derives eastings and northings of the ship at 2-minute intervals. These are expressed in metres and are relative to the position of the ship at the time for which the fix is to be calculated. The uncertainty in the absolute values of latitude and longitude does not affect the values for eastings and northings.

In essence, the method of computing a fix is not changed. Only the calculation of the ship co-ordinates and the partial derivatives of the variables for each count is somewhat different. All computation routines have been altered to use this method including POS4 which in addition creates the change in northerly velocity as a variable.

This method of accounting for the motion of the ship should reduce the final residuals below those derived using a uniform velocity. Where both methods have been tried on a fix it has generally been found that this is so.

Although relative motions of the ship have been accounted for, the velocity component due to ocean currents in deep water is still unknown and this will introduce an error into the fix position. In the worst case of a north-south current for a fix of moderately poor elevation the error is about 0.25 n.m. per knot of current. Currents around Australia generally don't exceed about 2 knots although there are examples of 4-knot currents off the east coast. The fix error from this source should thus be less than 0.5 n.m. normally but on occasion may be as much as 1 n.m.

The dead-reckoning navigation is adjusted to the satellite fix positions by using an Akima interpolation (Akima, 1974). If the fix positions are correct the effect of long-period current and systematic instrument error on the dead-reckoned positions would be almost entirely removed and thus the ship velocities derived from these corrected positions would be of good accuracy. If there is an error of 0.5 n.m.

in the fix position, however, it will generate an average velocity error over a 2-hour fix interval of 0.25 knots. Were these new dead-reckoned positions to be used in a recomputation of satellite fixes as described above, the errors in the new fix positions would not normally exceed 0.05 n.m., and even in extreme cases few fixes should be in error by more than 0.1 n.m. except in areas of rapidly varying current. Such an accuracy is within the requirements of the survey.

This is the method proposed in the reduction of the navigation data. It is the only method that will satisfactorily reduce the satellite fixes to the order of accuracy deemed necessary, although it will mean a significant increase in the computing cost and time spent producing the final results.

9. CONCLUSION

The satellite fix computation program BMRFIX is now set up in such a way that the user has considerable choice in how a fix may be computed. He also has several facilities available to aid him in assessing the quality and accuracy of a fix. The original quality of the satellite pass geometry and the number and accuracy of the Doppler counts may not be improved, but estimates of other parameters used in the fix computation may be.

The height of the antenna above the reference spheroid is not a very critical factor but should be known to about 10 metres for the position error to remain below 0.025 n.m. for most passes. The geoid heights supplied by the APL are considered to be good to this accuracy. The offset frequency is a much more sensitive parameter. However, plots of the computed values for each satellite over the whole period of the survey show clearly that the drift is mostly linear. The offset frequency may therefore be established most accurately by taking values from the long-term trend line.

The correction for tropospheric refraction is adequately computed by the simplified equations used in the program. Considering the accuracy expected and desired for a fix at sea it seems unnecessary for more complicated algorithms to be used, but small improvement may easily be made in future work by measuring and incorporating in the computation the atmospheric pressure at the time of the fix.

The most critical parameter in the computation of a fix recorded at sea is the velocity of the ship. In deep water where the sonar Doppler does not reflect off the sea bottom but is back-scattered from within the sea water the component of ship velocity due to ocean current is not known. Experiments using both northerly and easterly velocity components as variables met with mixed success. The derived velocities

especially the easterly component, showed error often in excess of any likely current. Using the dead-reckoned value for easterly velocity and varying only the northerly component, the error in the derived value of the latter for most fixes was within 0.5 knots and for a good fix may be much smaller. Such a method of computing fixes would seem to offer most advantage for quick on-line computation in deep-water areas. It should be possible to derive more accurate values of both velocity components in the future by recording a greater number of Doppler counts of shorter length, thereby improving the statistical reliability of the computation.

From post-processing of the present navigation data the most accurate results would seem to be obtained by using an iterative process. The original dead-reckoning is used in computing the satellite fixes initially. The dead-reckoning is then improved by tying it to the fix positions, and the fixes are recomputed using the new dead-reckoning. Such a process will of course add considerably to the cost and the time spent in processing the navigation but it is the only way of optimizing its accuracy. It is estimated that the second computation of the fixes should reduce most position errors to within 0.1 n.m.

10. REFERENCES

- AKIMA, H., 1970 - A new method of interpolation and smooth curve fitting based on local procedures. J. Assoc. Comput. Mach. 17(4), 589-602.
- BLACK, H.D., et al, 1973 - Recent (1973) improvements in the navy navigation satellite system.
Proceedings of the 1973 ION National Aerospace Meeting, March 1973.
- DENNIS, A.R., 1974 - A second generation navy satellite marine navigation system.
Navigation, 21 (1), pp. 37 - 43.
- HALAMANDARIS, H., & GILBERT, R.K., 1971 - The effect of ionospheric corrections to positioning accuracy using satellites.
Paper presented at the Offshore Technology Conference, Houston, Texas, April 1971.
- HOPFIELD, H.S., 1969 - Approximation to the tropospheric range correction.
Internal Memorandum (12.11.69), John Hopkins University, APL.
- HUDSPETH, J., in prep. - Procedures for assessing the accuracies of satellite fixes recorded at sea.
Bur. Miner. Resour. Aust. Rec.
- REGNAUDIN, J., 1969 - The Transit System.
Publication by Compagnie Generale de Geophysique, Paris, July 1969.
- SLUITER, P.G., 1969 - Relative weighting of satellite fixes. Paper presented at Marine Geodasy Symposium, New Orleans, Nov. 1969.
- STANSELL, T.A., Jr , 1968 - The navy navigation satellite system: Description and status. Navigation, 15 (3) pp. 229 - 243.
- YIONOULIS, S.M., 1970 - Algorithm to compute tropospheric refraction effects on range measurements. J. geophys. Res., 75(36), pp. 7636 - 7637.

PROGRAM BMRFIX

DRIVING INSTRUCTIONS
WRITTEN BY R.GARNETT ON 1.2.75

THE PROGRAM COMPUTES AND ASSESSES SATELLITE FIXES.
DATA IS INPUT ON LUN 40 FROM SATELLITE FIX DATA FILES.
THIS DATA MAY BE ALTERED FOR THE COMPUTATION USING EDIT CARDS.
A FILE CONTAINING ONE MINUTE D-R POSITIONS MAY BE INPUT ON LUN 1.
A FILE CONTAINING GEOID HEIGHTS MUST BE INPUT ON LUN 41.
A FILE CONTAINING OFFSET FREQUENCIES MUST BE INPUT ON LUN 42.
CARD IMAGES CONTAINING FIX TIME, COMPUTED POSITION, AND VALUES OF
ASSOCIATED PARAMETERS ARE OUTPUT ON LUN 50.

***** CONTROL CARDS FOR RUNNING ON THE CYBER 7600 *****

FIRST CARD SENDS JOB TO THE CYBER.....*CY,CHARGE CODE FOR TERMINAL,JOBNAME
BMRFIX,T50,NL2.
ACCOUNT(CHARGE CODE)

***** REQUEST TAPE CONTAINING LATEST ONE MINUTE D-R POSITIONS.
***** THIS IS OPTIONAL.
LABEL,TAPE1,R,L=STAPE LABELS.
STAGE,TAPE1,NT,HD,PRE,ST=OLT,VSN=TAPE NO.

***** REQUEST TAPE CONTAINING SATELLITE FIX DATA FILES.
LABEL,TAPE40,R,L=STAPE LABELS.
STAGE,TAPE40,NT,HD,PRE,ST=OLT,VSN=TAPE NO.

***** ATTACH FILE CONTAINING APL GEOID HEIGHTS.
ATTACH,TAPE41,GEOID,ID=CCDMRXMD.

***** ATTACH FILE CONTAINING LONG-TERM OFFSET FREQUENCIES.
ATTACH,TAPE42,LTOF,ID=CCDMRXMD.

***** REQUEST PERMANENT FILE FOR LUN 50 IF OUTPUT FILE TO BE SAVED
REQUEST,TAPE50,*PF.

***** ATTACH AND LOAD BINARY LIBRARIES CONTAINING BMRFIX ROUTINES.
ATTACH,LIB2,PHASE2,ID=CCDMRXMD.
ATTACH,LIB5,SERVICE,ID=CCDMRXMD.
LIBRARY(LIB2,LIB5)

***** EXECUTE BMRFIX.
BMRFIX.

***** CATALOG LUN 50 FILE IF IT IS TO BE SAVED.
CATALOG,TAPE50,FILE NAME,ID=CCDMRXMD,CN=LEAVE,MD=STRICTLY,EX=ALONE,RP=200.

- - - - - EOS - 7/8/9 PUNCHED IN COLUMN-1 - - - - -

***** DATA CARDS *****

***** CARD 1 - INDICATES WHETHER OR NOT A D-R NAVIGATION FILE
***** HAS BEEN ATTACHED.
FILE HEADER OF D-R NAVIGATION FILE IN 8A10 FORMAT
***** OR
*NO NAVIGATION FILE

***** CARD 2 - SPECIFIES THE OUTPUT FILE HEADER ON LUN 50
***** EVEN IF THE FILE IS NOT TO BE SAVED.
OUTPUT FILE HEADER IN 8A10 FORMAT

DRIVING INSTRUCTIONS FOR PROGRAM BMRFIX

SS.DDHHMM SS.DDHHMM

*EDITFIX SS.DDHHMM

999911.0							11
4779574.0	1989.0						19
1728	35.05	138E	150.0	8.0	0.0		31

***** IF A *EDITFIX CARD IS INCLUDED WITH NO EDITING CARDS THE
***** EFFECT WILL BE TO PREVENT ANY ASSESSMENT COUNT DELETIONS
***** IN FINAL COMPUTATION.
***** ANY NUMBER OF FIXES IN THE SEQUENCE MAY BE EDITED IN THIS
***** FASHION BUT EDIT CARDS MUST BE ARRANGED IN THE SAME ORDER
***** THAT THE FIXES ARE COMPUTED.
***** A SINGLE FIX MAY BE RECOMPUTED ANY NUMBER OF TIMES WITH
***** DIFFERENT EDITS BY REPEATING THE *EDITFIX CARD FOLLOWED
***** BY THE APPROPRIATE EDITING CARDS.

..... EOS - 7/8/9 PUNCHED IN COLUMN 1 - - - - -

***** SPECIFY ANY FURTHER SEQUENCES OF FIXES WHICH ARE TO BE
***** PROCESSED BY REPEATING CARD 3 PLUS ANY EDITING INSTRUCTIONS
***** FOR FIXES IN THAT SEQUENCE.
***** TERMINATE EACH SEQUENCE WITH AN END OF SECTION.
***** EACH SEQUENCE MUST CHRONOLOGICALLY FOLLOW THE PRECEDING ONE.

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ EOI - 6/7/8/9 PUNCHED IN COLUMN 1 ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

B.M.R. SATELLITE NAVIGATION PROGRAM - FIX ON 1972757715 AT 1350 USING SATELLITE 14 - SURVEY TIME 17,131350

FIXED DATA

8243928.0
3724457.0
1645067.0
19847.0
5125.0
745416.0
605541.0
-553.0
13396.0
1326055.0
999910.0

DOPPLER AND REFRACTION

3025540.0 2000.0
3151410.0 1995.0
3413126.0 1992.0
3795844.0 2003.0
4197033.0 2010.0
4479171.0 2005.0
4634322.0 2003.0
4711336.0 2000.0

VARIABLE DATA

25.0 304.0 -15.0
40.0 304.0 4.0
52.0 295.0 -10.0
61.0 278.0 -0.0
69.0 253.0 -4.0
74.0 223.0 -15.0
76.0 190.0 3.0
75.0 154.0 -25.0
72.0 118.0 9.0
64.0 84.0 46.0
54.0 54.0 14.0

ASSESSMENT OF R.M.S. RESIDUAL (IN METRES) USING LONG-TERM OFFSET FREQUENCY AND REMOVING ONE COUNT

ALTITUDE	ALL COUNTS	COUNT 1	COUNT 2	COUNT 3	COUNT 4	COUNT 5	COUNT 6	COUNT 7	COUNT 8
3. TO 10.	13.72	*****	11.86	17.38	13.62	14.44	14.95	16.04	6.65
10. TO 17.	-7.16	-4.14	*****	-1.91	-7.33	-7.05	-6.00	-4.67	-15.13
17. TO 24.	12.17	13.30	7.40	*****	11.90	10.55	12.47	13.67	6.25
24. TO 25.	-1.46	1.10	-2.22	4.92	*****	-4.60	-2.08	-1.83	1.74
25. TO 19.	-9.98	-10.79	-2.91	-3.07	-10.21	*****	-13.04	-13.93	-1.65
19. TO 12.	-7.70	-7.59	-5.63	-9.30	-7.82	-12.07	*****	-11.92	3.13
12. TO 5.	-16.42	-13.32	-15.20	-17.88	-16.47	-19.45	-18.68	*****	-7.39
5. TO -1.	62.52	63.89	53.62	60.94	62.50	60.46	60.86	59.95	*****
R.M.S. RESIDUALS	24.30	25.37	25.79	25.35	25.98	25.55	25.75	25.07	7.35

REJECTION OF COUNT 8 REDUCES R.M.S. RESIDUAL TO 7.4 METRES - BELOW ACCEPTABLE LIMIT OF 15.0 METRES

LOCK-ON TIME = 1350 GMT
EST. LAT = -27.8 DEGREES
EST. LONG = 113.8 DEGREES

COURSE = 328.1 DEGREES
SPEED = 9.11 KNOTS
GEOID HT INC. ANTENNA = 13.3 METRES

LONG-TERM OFFSET FREQUENCY = 32069.88 HERTZ
SATELLITE MOVING - SOUTH
MAXIMUM ALTITUDE - 25, DEGREES TO WEST

FIX TIME = 1400 GMT
LATITUDE = 28 12.817' SOUTH
LONGITUDE = 113 54.658' EAST

FIX SHIFT PER KNOT ERROR - NORTH	DLAT	DLONG
FIX SHIFT PER KNOT ERROR - NORTHEAST	.063' N	.268' E
FIX SHIFT PER KNOT ERROR - EAST	.004' S	.215' E
FIX SHIFT PER KNOT ERROR - SOUTHEAST	.066' S	.036' E
SHIFT CAUSED BY 100M ERROR IN GEOID	.089' S	.164' W
SHIFT CAUSED BY DERIVING DVN	.074' N	.050' E
SHIFT CAUSED BY USING AVERAGE VELOCITY	.013' S	.058' W
	.003' N	.003' E

DVN .22 KNOTS TO THE SOUTH

RESIDUALS IN METRES

ALTITUDE	INPUT DATA	NORTH	NORTH-EAST	EAST	SOUTH-EAST	GEOID HT	DERIVED VN	AVG VEL
3. TO 10.	6.65	20.30	18.51	9.78	-7.78	7.93	3.69	7.17
10. TO 17.	-15.13	-15.32	-13.83	-13.10	-13.56	-15.12	-15.10	-13.05
17. TO 24.	6.25	-3.70	-5.55	6.45	13.43	5.40	8.41	4.17
24. TO 25.	1.24	-1.17	-1.95	-1.86	1.46	.99	1.76	1.04
25. TO 19.	-1.65	8.50	4.43	-2.61	-8.50	.10	-2.63	.46
19. TO 12.	3.13	6.78	6.64	4.44	1.47	3.49	2.34	2.55
12. TO 5.	-7.38	-16.77	-10.31	-2.14	2.96	-8.08	-5.35	-6.90
*** TO ***	*****	*****	*****	*****	*****	*****	*****	*****
R.M.S. RESIDUALS	7.35	12.31	10.06	6.98	8.02	7.55	7.13	6.49

17.131400 28 12.82 113 54.66 1350 328. 9.1 13.3 14 S 25. W 32069.88

SAMPLE LISTING OF COMPUTATION WITH VARIABLES — LATITUDE AND LONGITUDE

FIXED DATA

423928.0
3724457.0
1645067.0
19347.0
5125.0
745416.0
605541.0
-553.0
13398.0
1396055.0
999910.0

DOPPLER AND REFRACTION

3025540.0 2000.0
3151416.0 1995.0
3413125.0 1992.0
3755844.0 2003.0
4137033.0 2010.0
4479171.0 2005.0
4634322.0 2003.0
4711336.0 2000.0

VARIABLE DATA

29.0 304.0 -15.0
40.0 304.0 4.0
32.0 295.0 -10.0
51.0 273.0 -0.0
69.0 253.0 -4.0
74.0 223.0 -15.0
76.0 190.0 3.0
75.0 154.0 -25.0
72.0 118.0 9.0
64.0 84.0 46.0
54.0 54.0 14.0

ASSESSMENT OF R.M.S. RESIDUAL (IN METRES) USING LONG-TERM OFFSET FREQUENCY AND REMOVING ONE COUNT

ALTITUDE	ALL COUNTS	COUNT 1	COUNT 2	COUNT 3	COUNT 4	COUNT 5	COUNT 6	COUNT 7	COUNT 8
3. TO 10.	13.72	*****	11.86	17.38	13.62	14.44	14.95	16.04	6.65
10. TO 17.	-7.16	-4.14	*****	-1.91	-7.33	-7.05	-6.30	-4.67	-15.13
17. TO 24.	-12.17	15.30	9.40	*****	11.90	10.55	12.47	13.67	6.25
24. TO 25.	-1.46	1.10	-2.22	4.92	*****	-4.69	-2.08	-1.83	1.24
25. TO 19.	-9.98	-10.79	-9.91	-9.07	-10.21	*****	-13.04	-13.93	-1.65
19. TO 12.	-7.70	-9.59	-6.63	-9.30	-7.82	-12.07	*****	-11.92	3.13
12. TO 5.	-16.42	-18.32	-15.20	-17.88	-16.47	-19.45	-18.68	*****	-7.39
5. TO -1.	62.52	60.99	53.62	60.94	62.50	60.46	60.86	59.95	*****
R.M.S. RESIDUALS	24.30	25.37	25.79	25.35	25.98	25.55	25.75	25.07	7.35

REJECTION OF COUNT 8 REDUCES R.M.S. RESIDUAL TO 7.4 METRES - BELOW ACCEPTABLE LIMIT OF 15.0 METRES

LOCK-ON TIME = 1350 GMT COURSE = 328.1 DEGREES LONG-TERM OFFSET FREQUENCY = 32069.88 HERTZ
EST. LAT = -27.8 DEGREES SPEED = 9.11 KNOTS SATELLITE MOVING - SOUTH
EST. LONG = 113.8 DEGREES GEOD HT INC. ANTENNA = 13.3 METRES MAXIMUM ALTITUDE = 25. DEGREES TO WEST

FIX TIME = 1400 GMT LATITUDE = 28 12.809' SOUTH COMPUTED OFFSET FREQUENCY = 32069.83 HERTZ
LONGITUDE = 113 54.658' EAST

FIX SHIFT PER KNOT ERROR - NORTH DLAT .060' N DLONG .268' E
FIX SHIFT PER KNOT ERROR - NORTHEAST DLAT .004' S DLONG .215' E
FIX SHIFT PER KNOT ERROR - EAST DLAT .066' S DLONG .036' E
FIX SHIFT PER KNOT ERROR - SOUTHEAST DLAT .089' S DLONG .164' W
SHIFT CAUSED BY 100M ERROR IN GEOD DLAT .004' N DLONG .050' E
SHIFT CAUSED BY DERIVING DVN DLAT .013' S DLONG .058' W DVN .22 KNOTS TO THE SOUTH
SHIFT USING LATITUDE-LONGITUDE ONLY DLAT .009' S DLONG .000' E

RESIDUALS IN METRES

ALTITUDE	NEW OFFSET	NORTH	NORTH-EAST	EAST	SOUTH-EAST	GEOD HT	DERIVED VN	INPUT DATA
3. TO 10.	9.65	20.33	18.51	9.78	-7.78	7.93	3.69	6.65
10. TO 17.	-13.51	-13.32	-13.83	-13.10	-13.56	-15.12	-15.10	-15.13
17. TO 24.	5.80	-3.73	-6.65	6.45	13.43	5.40	8.41	6.25
24. TO 25.	-6.62	-1.17	-1.95	-8.86	1.46	.99	1.76	1.24
25. TO 19.	-1.41	8.53	4.43	-2.61	-8.50	.10	-2.63	-1.65
19. TO 12.	4.54	6.73	6.64	4.44	1.47	3.49	2.34	3.13
12. TO 5.	-4.45	-16.77	-10.31	-2.14	2.96	-8.08	-5.35	-7.38
*** TO ***	*****	*****	*****	*****	*****	*****	*****	*****
R.M.S. RESIDUALS	7.09	12.31	10.06	6.98	8.02	7.55	7.13	7.35

17,131400 28 12.81 113 54.66 1350 328. 9.1 13.3 14 S 25. W 32069.83

SAMPLE LISTING OF COMPUTATION WITH VARIABLES - LATITUDE, LONGITUDE AND OFFSET FREQUENCY

B.M.R. SATELLITE NAVIGATION PROGRAM - FIX ON 1972/07/15 AT 1350 USING SATELLITE 14 - SURVEY TIME 17,131350

FIXED DATA

8243928.0
3724457.0
1645067.0
19847.0
5125.0
745416.0
605541.0
-553.0
13398.0
1396055.0
999910.0

DOPPLER AND REFRACTION

3025540.0 2003.0
3161416.0 1995.0
3413126.0 1992.0
3795844.0 2003.0
4197033.0 2013.0
4479171.0 2003.0
4634322.0 2003.0
4711336.0 2003.0

VARIABLE DATA

29.0 304.0 -15.0
40.0 304.0 4.0
52.0 295.0 -10.0
61.0 278.0 -0.0
69.0 253.0 -4.0
74.0 223.0 -15.0
76.0 190.0 3.0
75.0 154.0 -25.0
72.0 118.0 9.0
64.0 84.0 46.0
54.0 54.0 14.0

ASSESSMENT OF R.M.S. RESIDUAL (IN METRES) USING LONG-TERM OFFSET FREQUENCY AND REMOVING ONE COUNT

ALTITUDE	ALL COUNTS	COUNT 1	COUNT 2	COUNT 3	COUNT 4	COUNT 5	COUNT 6	COUNT 7	COUNT 8
3. TO 10.	13.72	*****	11.86	17.38	13.62	14.44	14.95	16.04	6.65
10. TO 17.	-7.16	-4.14	*****	-1.91	-7.33	-7.05	-6.00	-4.67	-15.13
17. TO 24.	12.17	15.30	9.40	*****	11.90	10.55	12.47	13.67	6.25
24. TO 25.	-1.46	1.10	-2.22	4.92	*****	-4.69	-2.08	-1.83	1.24
25. TO 19.	-9.98	-10.79	-9.91	-8.07	-10.21	*****	-13.04	-13.93	-1.65
19. TO 12.	-7.70	-9.59	-6.63	-5.30	-7.82	-12.07	*****	-11.92	3.13
12. TO 5.	-16.42	-18.32	-15.20	-17.88	-16.47	-19.45	-18.68	*****	-7.39
5. TO -1.	62.52	60.89	53.62	60.94	62.50	60.46	60.86	59.95	*****
R.M.S. RESIDUALS	24.30	25.37	25.79	25.35	25.98	25.55	25.75	25.07	7.35

REJECTION OF COUNT 8 REDUCES R.M.S. RESIDUAL TO 7.4 METRES - BELOW ACCEPTABLE LIMIT OF 15.0 METRES

LOCK-ON TIME = 1350 GMT
EST. LAT = -27.8 DEGREES
EST. LONG = 113.8 DEGREES

COURSE = 328.1 DEGREES
SPEED = 9.11 KNOTS
GEOID HT INC. ANTENNA = 13.3 METRES

LONG-TERM OFFSET FREQUENCY = 32069.88 HERTZ
SATELLITE MOVING - SOUTH
MAXIMUM ALTITUDE = 25. DEGREES TO WEST

FIX TIME = 1400 GMT

LATITUDE = 28 12.826' SOUTH
LONGITUDE = 113 54.543' EAST

COMPUTED GEOID HEIGHT INC. ANTENNA = -217.8 METRES

FIX SHIFT PER KNOT ERROR - NORTH
FIX SHIFT PER KNOT ERROR - NORTHEAST
FIX SHIFT PER KNOT ERROR - EAST
FIX SHIFT PER KNOT ERROR - SOUTHEAST
SHIFT CAUSED BY 100M ERROR IN GEOID
SHIFT CAUSED BY DERIVING DVN
SHIFT USING LATITUDE-LONGITUDE ONLY

DLAT .060' N DLONG .268' E
DLAT .004' S DLONG .215' E
DLAT .065' S DLONG .036' E
DLAT .089' S DLONG .164' W
DLAT .004' N DLONG .050' E
DLAT .013' S DLONG .058' W
DLAT .009' N DLONG .115' E

DVN .22 KNOTS TO THE SOUTH

RESIDUALS IN METRES

ALTITUDE	NEW GEOID	NORTH	NORTH-EAST	EAST	SOUTH-EAST	GEOID HT	DERIVED VN	INPUT DATA
3. TO 10.	3.70	20.33	18.51	9.78	-7.78	7.93	3.69	6.65
10. TO 17.	-15.17	-15.32	-13.83	-13.10	-13.56	-15.12	-15.10	-15.13
17. TO 24.	8.23	-3.73	-1.65	6.45	13.43	5.40	8.41	6.25
24. TO 25.	1.84	-1.17	-1.95	-1.86	1.46	.99	1.76	1.24
25. TO 19.	-2.36	8.53	4.43	-2.61	-8.50	.10	-2.63	-1.65
19. TO 12.	2.30	6.79	6.64	4.44	1.47	3.49	2.34	3.13
12. TO 5.	-5.78	-16.77	-10.31	-2.14	2.96	-8.08	-5.35	-7.39
*** TO ***	*****	*****	*****	*****	*****	*****	*****	*****
R.M.S. RESIDUALS	7.16	12.31	10.06	6.98	8.02	7.55	7.13	7.35

*17,131400 28 12.83 113 54.54 1350 328. 9.1***** 14 S 25. W 32069.88*

SAMPLE LISTING OF COMPUTATION WITH VARIABLES — LATITUDE, LONGITUDE AND GEOID HEIGHT

FIXED DATA

8243928.0
3724457.0
1645067.0
19847.0
5125.0
745416.0
615541.0
-553.0
13398.0
1396055.0
909910.0

DOPPLER AND REFRACTION

3025540.0 2000.0
3161416.0 1996.0
3413126.0 1992.0
3795844.0 2008.0
4197033.0 2010.0
4479171.0 2005.0
4634322.0 2008.0
4711336.0 2000.0

VARIABLE DATA

29.0 304.0 -15.0
40.0 304.0 4.0
52.0 295.0 -10.0
61.0 278.0 -0.0
69.0 253.0 -4.0
74.0 223.0 -15.0
76.0 190.0 3.0
75.0 154.0 -25.0
72.0 118.0 9.0
64.0 84.0 46.0
54.0 54.0 14.0

ASSESSMENT OF R.M.S. RESIDUAL (IN METRES) USING LONG-TERM OFFSET FREQUENCY AND REMOVING ONE COUNT

ALTITUDE	ALL COUNTS	COUNT 1	COUNT 2	COUNT 3	COUNT 4	COUNT 5	COUNT 6	COUNT 7	COUNT 8
3. TO 10.	13.72	*****	11.86	17.38	13.62	14.44	14.95	16.04	6.65
10. TO 17.	-7.16	-4.14	*****	-1.91	-7.33	-7.05	-6.00	-4.67	-15.13
17. TO 24.	12.17	15.30	9.40	*****	11.90	10.55	12.47	13.67	6.25
24. TO 25.	-1.46	1.10	-2.72	4.92	*****	-4.69	-2.08	-1.83	1.24
25. TO 19.	-9.98	-10.79	-9.01	-8.07	-10.21	*****	-13.04	-13.93	-6.65
19. TO 12.	-7.70	-9.59	-6.63	-8.30	-7.82	-12.07	*****	-11.92	3.13
12. TO 5.	-16.42	-18.32	-15.20	-17.88	-16.47	-19.45	-18.68	*****	-7.39
5. TO -1.	62.52	60.89	63.62	60.94	62.50	60.46	60.86	59.95	*****
R.M.S. RESIDUALS	24.30	25.37	25.79	25.35	25.98	25.55	25.75	25.07	7.35

REJECTION OF COUNT 8 REDUCES R.M.S. RESIDUAL TO 7.4 METRES - BELOW ACCEPTABLE LIMIT OF 15.0 METRES

LOCK-ON TIME = 1350 GMT
EST. LAT = -27.8 DEGREES
EST. LONG = 113.8 DEGREES

COURSE = 328.1 DEGREES
SPEED = 9.11 KNOTS
GEOID HT INC. ANTENNA = 13.3 METRES

LONG-TERM OFFSET FREQUENCY = 32069.88 HERTZ
SATELLITE MOVING - SOUTH
MAXIMUM ALTITUDE - 25. DEGREES TO WEST

FIX TIME = 1400 GMT

LATITUDE = 28 12.830' SOUTH
LONGITUDE = 113 54.600' EAST

COMPUTED NORTH VELOCITY = 0.00 KNOTS

FIX SHIFT PER KNOT ERROR - NORTH
FIX SHIFT PER KNOT ERROR - NORTHEAST
FIX SHIFT PER KNOT ERROR - EAST
FIX SHIFT PER KNOT ERROR - SOUTHEAST
SHIFT CAUSED BY 100M ERROR IN GEOID
SHIFT CAUSED BY DERIVING DVN
SHIFT USING LATITUDE-LONGITUDE ONLY

DIAT .060' N DLONG .268' E
DIAT .004' S DLONG .215' E
DIAT .066' S DLONG .036' E
DIAT .089' S DLONG .164' W
DIAT .004' N DLONG .050' E
DIAT .013' S DLONG .058' W
DIAT .013' N DLONG .058' E

DVN .22 KNOTS TO THE SOUTH

RESIDUALS IN METRES

ALTITUDE	NEW DVN	NORTH	NORTH-EAST	EAST	SOUTH-EAST	GEOID HT	DERIVED VN	INPUT DATA
3. TO 10.	3.69	20.30	18.51	9.78	-7.78	7.93	3.69	6.65
10. TO 17.	-15.09	-15.32	-13.83	-13.10	-13.56	-19.12	-15.10	-15.13
17. TO 24.	8.41	-3.70	-6.65	6.45	13.43	5.40	8.41	6.25
24. TO 25.	1.76	-1.17	-1.95	-1.86	1.46	.99	1.76	1.24
25. TO 19.	-2.63	8.50	4.43	-2.61	-8.50	.10	-2.63	-6.65
19. TO 12.	2.34	6.78	6.64	4.44	1.47	3.49	2.34	3.13
12. TO 5.	-5.35	-16.77	-10.31	-2.14	2.96	-8.08	-5.35	-7.39
*** TO ***	*****	*****	*****	*****	*****	*****	*****	*****
R.M.S. RESIDUALS	7.13	12.31	10.06	6.98	8.02	7.55	7.13	7.35

17.131400 28 12.83 113 54.60 1350 328. 9.1 13.3 14 S 25. W 32069.88

SAMPLE LISTING OF COMPUTATION WITH VARIABLES - LATITUDE, LONGITUDE AND SHIP'S NORTHERLY VELOCITY


```

PROGRAM BMRFIX(INPUT,TAPE43=INPUT,OUTPUT,TAPE43,TAPE41,TAPE42,
1 TAPE50,TAPE1)
2
3 DIMENSION LABEL(8),NAME(8),LIST1(4,3),LIST2(4,4)
4 COMMON/ORB17/ TP,XNDOT,FP,DFP,EP51,AD,WM,DWM,COS5,NG,SINS
5 COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
6 COMMON/FIX/TLOC,QN,XLAT,XLAT,XLON,XLON,SH,SV,QM,N,NDOP(8),TO
7 COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELOM,GMSAVE,OFFSAVE
8 COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTNAX,ISIGN
9 COMMON/LUN/ MTIN,LUNDOC,LUNGM,LUNLTOF,LUNSAVE
10 COMMON/CONST1/ B4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
11 COMMON/CONST2/ INORTH,ISOUTH,IEAST,INEST,DUBIOUS,UNKNOWN
12 EQUIVALENCE (ARRAY,TP)
13 DATA (MTIN=1),(LUNDOC=40),(LUNSAVE=50),(LUNGM=41),(LUNLTOF=42)
14 DATA (E4=1,E=4),(E6=1,E=6),(E7=1,E=7),(XMM=30.86667)
15 DATA (DEGRA=0.01745329252),(RAMIN=3437.7468),(FIVESEC=0.00000005)
16 DATA (INORTH=7M,NORTH),(ISOUTH=7M,SOUTH),(IEAST=6M,EAST),
17 1 (INEST=6M,WEST),(DUBIOUS=1,OE9),(UNKNOWN=1,OE10),(ANTENNA=18,3)
18 DATA (LIST1(1,1)=40HALL FIXES ASSESSED, PROCESSED AND LISTED)
19 DATA (LIST1(1,2)=40HALL FIXES ONLY PROCESSED AND LISTED )
20 DATA (LIST1(1,3)=40HALL FIXES PROCESSED, EDITED FIXES LISTED)
21 DATA (LIST2(1,1)=40HALLITUDE AND LONGITUDE ONLY )
22 DATA (LIST2(1,2)=40HALLITUDE, LONGITUDE AND OFFSET FREQUENCY)
23 DATA (LIST2(1,3)=40HALLITUDE, LONGITUDE AND GEOID HEIGHT )
24 DATA (LIST2(1,4)=40HALLITUDE, LONGITUDE AND NORTH VELOCITY )
25
26 CALL ERRSEY (KK,1000)
27 REWIND MTIN $ REWIND IUNDOC $ REWIND LUNSAVE
28 PRINT 90
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

```

FORTRAN LISTING OF PROGRAM BMRFIX

```

SUBROUTINE LOOKUP(ICARD,LUN,NEOF,NPARITY,NFLAG)
C      VERSION CREATED 01/06/77
C      THIS SUBROUTINE HAS TWO ENTRY POINTS
C      1. LOOKUP - SEARCHES A MAGNETIC TAPE SEQUENTIALLY FOR A
C      SUPPLIED FILE NAME.
C      ICARD = FILE LABEL TO BE FOUND
C      LUN = LOGICAL UNIT FOR SEARCH
C      NEOF = NUMBER OF CONSECUTIVE EOFs AT END OF PROCESSING FILE
C      NPARITY = PARITY OF SEARCH DONE FOR LUN CONCERNED
C      1 = ODD PARITY (BUFFER)
C      0 = EVEN PARITY (BUFFER)
C      NFLAG = INDICATOR, 0=FILE FOUND, 1= FILE NOT FOUND
C
C      2.SETCOPY = ALLOWS TAPE CONTENTS UP TO A REQUIRED FILE TO BE
C      COPIED ONTO A NEW FILE, CALLING PARAMETRS AHEAD
C      CALL SETCOPY(ICARD,LUNSAVE,NEOF,NPARITY,NWORDS)
C      ICARD = ARRAY IN WHICH BUFFERED DATA IS TEMPORARILY STORED
C      LUNSAVE = LOGICAL UNIT FOR OUTPUT COPY
C      NWORDS = NUMBER OF WORDS IN BUFFER ON READ
C
C      DIMENSION LABEL(8),ICARD(8)
C      DATA (ICOPY = 0)
C      DATA (ISKIP=1)
C
C      5 IF (ICARD(1),EQ,1RM+REWIND) 1 12,20
C
C      10 REWIND LUN 5 NFLAG = 1 5 RETURN
C
C      20 IF (NPARITY,GE,0) 30,40
C      30 BUFFER IN(LUN,NPARITY) (LABEL(1),LABEL(8))
C      IF (UNIT(LUN)) 70,200,70
C      35 40 READ(LUN,50) LABEL
C      50 FORMAT(8A10)
C      IF (EOF(LUN)) 200,60
C      40 IF (IOCHECK(LUN)) 70,70
C
C      70 DO 40 K=1,8
C      IF (ICARD(K),EQ,LABEL(K)) 80,100
C      40 CONTINUE
C
C      PRINT 00, ICARD,LUN
C      90 FORMAT(///,10X,=FILE=,5X,1M=.8A10,1M=.5X,=FOUND ON LUN=,10,/)
C      NFLAG = 0 5 RETURN
C
C      100 IF (ICOPY,EQ,1) 190,110
C      110 IF (ISKIP,EQ,0) 115,130
C      115 PRINT 120,LABEL,LUN
C
C      120 FORMAT(///,10X,=FILE=,5X,1M=.8A10,1M=.5X,=SKIPPED ON LUN=,10,/)
C
C      130 CALL SKIP66(LUN,NPARITY)
C      IF (NEOF,EQ,1) 20,100
C
C      140 DO 140 K=2,NEOF
C      IF (NPARITY,GE,0) 150,160
C      150 BUFFER IN(LUN,NPARITY) (LABEL(1),LABEL(8))
C      IF (UNIT(LUN)) 170,140,130
C      160 READ(LUN,50) LABEL
C      IF (EOF(LUN)) 140,170
C      170 IF (IOCHECK(LUN)) 130,130
C      180 CONTINUE
C      GO TO 20
C
C      190 PRINT 200, LABEL,LUN,LUNSAVE
C      200 FORMAT(///,10X,=FILE=,5X,1M=.8A10,1M=.5X,=COPIED FROM=,10,= TO=,10,=
C      1/)
C
C      210 BACKSPACE LUN
C      220 IF (NPARITY,GE,0) 220,240
C      220 BUFFER IN (LUN,NPARITY) (LABEL(1),LABEL(NWORDS))
C      IF (UNIT(LUN)) 230,270,230
C      230 NM = LENGTH(LUN) 8 NEOF = 0
C      BUFFER OUT (LUNSAVE,NPARITY) (LABEL(1),LABEL(NM))
C      IF (UNIT(LUNSAVE)) 220,220
C
C      240 READ (LUN,50) LABEL
C      IF (EOF(LUN)) 270,250
C      250 IF (IOCHECK(LUN)) 260,260
C      260 NEOF = 0
C      WRITE (LUNSAVE,50) LABEL
C      GO TO 240
C
C      270 ENDFILE LUNSAVE
C      NEOF = NEOF + 1
C      IF (NEOF,EQ,NEOF) 20,210
C
C      280 PRINT 290, ICARD,LUN
C      290 FORMAT(///,30X,75(1M=1,///,10X,=FILE=,5X,1M=.8A10,1M=.5X,
C      1=NOT FOUND=,///,10X,=IN SEQUENTIAL SEARCH OF MULTIPLE FILE ON LUN=
C      2,10,= IF MAY EXIST IN FILE BEFORE STARTING POINT OF SEARCH=,
C      3///,30X,75(1M=1,///)
C      REWIND LUN 5 NFLAG = 1 5 RETURN
C
C      ENTRY SETCOPY
C
C      LUNSAVE = LUN 8 ICOPY = 1
C      NWORDS = NFLAG
C
C      IF (NPARITY,EQ,-1,AND,NWORDS,GT,8) PRINT 300, NWORDS
C      300 FORMAT(///,10X,=NUMBER OF WORDS FOR BCD OUTPUT IS =,10,= ONLY A MA
C      1XIMUM OF 8 IS ALLOWED=,/)
C
C      RETURN
C      ENTRY SETSKIP
C      ISKIP=0 5 RETURN
C
C      END

```

LOOKUP 2
 LOOKUP 3
 LOOKUP 4
 LOOKUP 5
 LOOKUP 6
 LOOKUP 7
 LOOKUP 8
 LOOKUP 9
 LOOKUP 10
 LOOKUP 11
 LOOKUP 12
 LOOKUP 13
 LOOKUP 14
 LOOKUP 15
 LOOKUP 16
 LOOKUP 17
 LOOKUP 18
 LOOKUP 19
 LOOKUP 20
 LOOKUP 21
 LOOKUP 22
 LOOKUP 23
 LOOKUP 24
 LOOKUP 25
 LOOKUP 26
 LOOKUP 27
 LOOKUP 28
 LOOKUP 29
 LOOKUP 30
 LOOKUP 31
 LOOKUP 32
 LOOKUP 33
 LOOKUP 34
 LOOKUP 35
 LOOKUP 36
 LOOKUP 37
 LOOKUP 38
 LOOKUP 39
 LOOKUP 40
 LOOKUP 41
 LOOKUP 42
 LOOKUP 43
 LOOKUP 44
 LOOKUP 45
 LOOKUP 46
 LOOKUP 47
 LOOKUP 48
 LOOKUP 49
 LOOKUP 50
 LOOKUP 51
 LOOKUP 52
 LOOKUP 53
 LOOKUP 54
 LOOKUP 55
 LOOKUP 56
 LOOKUP 57
 LOOKUP 58
 LOOKUP 59
 LOOKUP 60
 LOOKUP 61
 LOOKUP 62
 LOOKUP 63
 LOOKUP 64
 LOOKUP 65
 LOOKUP 66
 LOOKUP 67
 LOOKUP 68
 LOOKUP 69
 LOOKUP 70
 LOOKUP 71
 LOOKUP 72
 LOOKUP 73
 LOOKUP 74
 LOOKUP 75
 LOOKUP 76
 LOOKUP 77
 LOOKUP 78
 LOOKUP 79
 LOOKUP 80
 LOOKUP 81
 LOOKUP 82
 LOOKUP 83
 LOOKUP 84
 LOOKUP 85
 LOOKUP 86
 LOOKUP 87
 LOOKUP 88
 LOOKUP 89
 LOOKUP 90
 LOOKUP 91
 LOOKUP 92
 LOOKUP 93
 LOOKUP 94
 LOOKUP 95
 LOOKUP 96
 LOOKUP 97
 LOOKUP 98
 LOOKUP 99
 LOOKUP 100
 LOOKUP 101
 LOOKUP 102
 LOOKUP 103
 LOOKUP 104
 LOOKUP 105
 LOOKUP 106
 LOOKUP 107
 LOOKUP 108
 LOOKUP 109
 LOOKUP 110
 LOOKUP 111
 LOOKUP 112

	SUBROUTINE TODAY(I DATE)	TODAY	2
		TODAY	3
C	THIS SUBROUTINE REFORMATS THE DATE FROM DD/MM/YY TO THE	TODAY	4
C	INTERNATIONAL STANDARD YYYY/MM/DD.	TODAY	5
		TODAY	6
	CALL DATE(I VALUE)	TODAY	7
	DECODE(10,10,I VALUE) DD,MM,YY	TODAY	8
10	FORMAT(1X,3(A2,1X))	TODAY	9
		TODAY	10
	E+CODE(10,20,I DATE) YY,MM,DD	TODAY	11
20	FORMAT(=10=,A2,=0=,A2,=0=,A2)	TODAY	12
		TODAY	13
	RETURN	TODAY	14
		TODAY	15
	END	TODAY	16

G 449-120A

```

SUBROUTINE EDITFIX(FIXEDIT)
  DIMENSION DEK(11),DAK(11),ZSK(11)
  DIMENSION ICARD(7),ARRAY(11),STORE(60),LIST(4)
  COMMON/ORB1/ TP,XMDOT,FP,DFP,EPSI,A0,MN,DWN,COS5,W0,SINS
  COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
  COMMON/XYZ/ LOCOSN,XLAT,M(1),XT,XLON,XLON,MH,SV,QH,N,NDDP(8),TO
  COMMON/SAVE/FIXTIME,IDENT(11),OFFSET,SAVELAT,SAVELOH,QHSAVE,OFFSAVE
  EQUIVALENCE (DEK,XSAT),(DAK,YSAT),(ZSK,ZSAT)
  EQUIVALENCE (ARRAY(1),TP)
  DATA (LIST*10MIXED DATA,7MDOPPLER,8MVARIABLE,10MSHIPS DATA)
  DATA (FIXLAST=0.0)

  IF (ABS(FIXEDIT-FIXLAST).LT.1.0E-8) 10,15
10 CALL MOVLEV(STORE(1),TP,60) $ GO TO 1140
15 CALL MOVLEV(TP,STORE(1),60)
1140 PRINT 510, IDENT,TLOCKON,(ARRAY(1),DOP(1),REF(1),DEK(1),DAK(1),
1 ZSK(1),1=1,8),(ARRAY(1),DEK(1),DAK(1),ZSK(1),1=9,11)
20 510 FORMAT(1H1,9X,'R.M.R. SATELLITE NAVIGATION PROGRAM - FIX ON ',
1 A10.0,AT,'R4.' USING SATELLITE=14,' - SURVEY TIME=F12.6,
2 //,10X,'FIXED DATA=9X,'DOPPLER AND REFRACTION=20X,
3 'VARIABLE DATA=//,8//,10X,F10.1,10X,2F10.1,10X,3F10.1),
4 3//,10X,F10.1,40X,3F10.1)

25 ENTRY EDITSET
20 READ 30, ICARD,NRECORD
30 FORMAT(7A10,110)
IF (BOF(60)) 40,90
30 40 FIXEDIT = 1.0E10
RETURN

90 IF (ICARD(1),EQ,10M=EDITFIX) 100,120
100 FIXLAST = FIXEDIT
DECODE(10,110,ICARD(2)) FIXEDIT
110 FORMAT (F10.6)
RETURN

120 IF (NRECORD,LT,1.0R,NRECORD,GT,31) 250,130
250 PRINT 260, NRECORD
260 FORMAT (//,10X,10(1H=),* RECORD NO.=,13,*ON EDITING CARD IS ILLEGAL
1 - NO EDIT DONE=//)
GO TO 20

45 130 IF (NRECORD,LT,12) 140,160
140 PRINT 135, LIST(1),NRECORD,ICARD
135 FORMAT(5X,10(1H=),5X,EDIT OF *,A10,* CARD=,14,4H *,7A10,1H=)
DECODE(10,190,ICARD(1)) ARRAY(NRECORD)
150 FORMAT (F10.0)
GO TO 20

160 IF (NRECORD,LT,20) 170,190
170 I = NRECORD-11 $ PRINT 135, LIST(2),1,ICARD
DECODE(20,180,ICARD(1)) DAP(1),REF(1)
180 FORMAT (2F10.1)
GO TO 20

190 IF (NRECORD,LT,31) 200,220
200 I = NRECORD-19 $ PRINT 135, LIST(3),1,ICARD
DECODE(30,210,ICARD(1)) DEK(1),DAK(1),ZSAT(1)
210 FORMAT (3F10.1)
GO TO 20

220 I = 1 $ PRINT 135, LIST(4),1,ICARD
230 DECODE(50,240,ICARD(1)) HOURS,XMIN,XLAT,XLON,XLON,SNLAT,SNLON,SN,SV
240 FORMAT (F8.0,F2.0,2(F9.0,A1),2F10.1)
GO TO 20

END

```

```

801TFIX 2
801TFIX 3
801TFIX 4
801TFIX 5
801TFIX 6
801TFIX 7
801TFIX 8
801TFIX 9
801TFIX 10
801TFIX 11
801TFIX 12
801TFIX 13
801TFIX 14
801TFIX 15
801TFIX 16
801TFIX 17
801TFIX 18
801TFIX 19
801TFIX 20
801TFIX 21
801TFIX 22
801TFIX 23
801TFIX 24
801TFIX 25
801TFIX 26
801TFIX 27
801TFIX 28
801TFIX 29
801TFIX 30
801TFIX 31
801TFIX 32
801TFIX 33
801TFIX 34
801TFIX 35
801TFIX 36
801TFIX 37
801TFIX 38
801TFIX 39
801TFIX 40
801TFIX 41
801TFIX 42
801TFIX 43
801TFIX 44
801TFIX 45
801TFIX 46
801TFIX 47
801TFIX 48
801TFIX 49
801TFIX 50
801TFIX 51
801TFIX 52
801TFIX 53
801TFIX 54
801TFIX 55
801TFIX 56
801TFIX 57
801TFIX 58
801TFIX 59
801TFIX 60
801TFIX 61
801TFIX 62
801TFIX 63
801TFIX 64
801TFIX 65
801TFIX 66
801TFIX 67
801TFIX 68
801TFIX 69
801TFIX 70

```

SUBROUTINE NEXTDATE(TIME,DATE,TYPE)

```

  DIMENSION NDAYS(13,2),NYEAR(20),NSTART(20)
  DATA (NDAYS=0,31,59,90,120,151,181,212,243,273,304,334,365,
1 0,31,60,91,121,152,182,213,244,274,305,335,366)
  DATA (NYEAR=1969,2(1967),1968,6(1970),5(1971),4(1972))
  DATA (NSTART=207,2(265),267,246,5(346),46,96,162,281,361,
1 88,164,269,347)

10 GO TO (10,100) (TYPE)

10 ISURVEY = TIME $ IDAY = (TIME-ISURVEY)*100.0+0.5
IF (ISURVEY,GT,0. AND,ISURVEY,LT,21) 20,80
20 IDAY = NSTART((ISURVEY)-IDAY) $ IYEAR = NYEAR((ISURVEY))
15 30 ICOL = 1 $ IF ((IYEAR/4)*4,EQ,IYEAR) ICOL = 2

DO 40 K=1,12
IF ((IDAY,GT,NDAYS(K,1,ICOL)) 40,90
40 CONTINUE
IDAY = IDAY-NDAYS(K,1,ICOL) $ IYEAR = IYEAR-1
GO TO 30

50 IDAY = IDAY-NDAYS(K,1,ICOL)+1000 $ IMONTH = K+1000
ENCODE(10,60,IDAY) IDAY
25 ENCODE(10,60,IMONTH) IMONTH
ENCODE(10,60,IYEAR) IYEAR
60 FORMAT(110)
ENCODE(10,70,DATE) IYEAR,IMONTH,IDAY
30 70 FORMAT(R4.0//,R2.0//,R2.0//,R2.0)

RETURN

80 IDATE = 10M=UNKNOWN $ RETURN

35 100 DECODE(10,110,DATE) IYEAR,IMONTH,IDAY
110 FORMAT(14,2(1X,12))
ICOL = 1 $ IF ((IYEAR/4)*4,EQ,IYEAR) ICOL = 2
IDAY = IDAY-NDAYS(IMONTH,ICOL)

40 DO 130 K=1,20
IF (NYEAR(K)-IYEAR) 130,120,140
120 IF (IDAY,GT,NSTART(K)) 130,140
130 CONTINUE
K = 21

45 140 IF (NYEAR(K-1),LT,IYEAR) 150,160
150 IYEAR = IYEAR-1
ICOL = 1 $ IF ((IYEAR/4)*4,EQ,IYEAR) ICOL = 2
IDAY = IDAY-NDAYS(13,ICOL)
50 160 ISURVEY = K-1 $ IDAY = IDAY-NSTART(K-1)
TIME = ISURVEY+0.01*IDAY
IF (IDAY,GT,99) TIME = 1.0E10

55 RETURN
END

```

```

NEWDATE 2
NEWDATE 3
NEWDATE 4
NEWDATE 5
NEWDATE 6
NEWDATE 7
NEWDATE 8
NEWDATE 9
NEWDATE 10
NEWDATE 11
NEWDATE 12
NEWDATE 13
NEWDATE 14
NEWDATE 15
NEWDATE 16
NEWDATE 17
NEWDATE 18
NEWDATE 19
NEWDATE 20
NEWDATE 21
NEWDATE 22
NEWDATE 23
NEWDATE 24
NEWDATE 25
NEWDATE 26
NEWDATE 27
NEWDATE 28
NEWDATE 29
NEWDATE 30
NEWDATE 31
NEWDATE 32
NEWDATE 33
NEWDATE 34
NEWDATE 35
NEWDATE 36
NEWDATE 37
NEWDATE 38
NEWDATE 39
NEWDATE 40
NEWDATE 41
NEWDATE 42
NEWDATE 43
NEWDATE 44
NEWDATE 45
NEWDATE 46
NEWDATE 47
NEWDATE 48
NEWDATE 49
NEWDATE 50
NEWDATE 51
NEWDATE 52
NEWDATE 53
NEWDATE 54
NEWDATE 55
NEWDATE 56
NEWDATE 57

```

```

SUBROUTINE SETDATA
DIMENSION DEK(11),DAK(11),ZSK(11)
COMMON/ORBIT/ TP,XMDOT,FP,DFP,EPSI,AO,WN,DWN,COSS,WG,SINS
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,NLAT,XLON,NLON,SM,SV,GH,N,NDOP(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELON,GH,SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMAX,ISIGN
COMMON/CONST1/ F4,E6,E7,XMMH,DEGRA,RAHIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IWEST,DUBIOUS,UNKNOWN
EQUIVALENCE (DEK,XSAT),(DAK,YSAT),(ZSK,ZSAT)
EQUIVALENCE (ARRAY,TP)

TP=TP+E4
XMDOT=XMDOT+E7+3.0)*DEGRA
FP=FP+E4*DEGRA
DFP=DFP+E7*DEGRA
EPSI=EPSI+E6
AO=AO+10.0
WN=WN+E4*DEGRA
DWN=DWN+E7*DEGRA
COSS=COSS+E6
WG=WG+E4*DEGRA
SINS=SINS+E6
DNG=.0043752695
WN=WN-WG
DWN=DWN-DWG

ITIME = AMOD(TLOCKON+1.0E6,100.0)+0.5
JJ = 4 $ IF ((ITIME/4)=4,Eq,ITIME) JJ = 5
ZSK(JJ-2) = 0.375*ZSK(JJ-3)+0.750*ZSK(JJ-1)-0.125*ZSK(JJ+1)
DO 440 J=JJ,19,2
ZSK(J) = 0.375*ZSK(J-1)+0.750*ZSK(J-1)-0.125*ZSK(J-3)
440 CONTINUE
DO 450 J=1,9
DEK(J) = DEK(J+1) $ DAK(J) = DAK(J+1) $ ZSK(J) = ZSK(J+1)
450 CONTINUE

N = 0
DO 560 J=1,8
IF (DOP(J).LT.1.0) 530,550
530 ALT(1,J) = ALT(2,J) = UNKNOWN
DO 540 K=1,8
RESID(K,J) = UNKNOWN
540 CONTINUE
GO TO 560
550 DOP(J)=DOP(J)+(REF(J)-2000.0)*(-1.0)+24.0/55.0
DOP(J) = DOP(J)+0.74948234
N = N+1 $ NDOP(N) = J
560 CONTINUE

RETURN
END

```

```

SFYDATA 2
SETDATA 3
SETDATA 4
SETDATA 5
SETDATA 6
SETDATA 7
SETDATA 8
SETDATA 9
SETDATA 10
SETDATA 11
SETDATA 12
SETDATA 13
SETDATA 14
SETDATA 15
SETDATA 16
SETDATA 17
SETDATA 18
SETDATA 19
SETDATA 20
SETDATA 21
SETDATA 22
SETDATA 23
SETDATA 24
SETDATA 25
SETDATA 26
SETDATA 27
SETDATA 28
SETDATA 29
SETDATA 30
SETDATA 31
SETDATA 32
SETDATA 33
SETDATA 34
SETDATA 35
SETDATA 36
SETDATA 37
SETDATA 38
SETDATA 39
SETDATA 40
SETDATA 41
SETDATA 42
SETDATA 43
SETDATA 44
SETDATA 45
SETDATA 46
SETDATA 47
SETDATA 48
SETDATA 49
SETDATA 50
SETDATA 51
SETDATA 52
SETDATA 53
SETDATA 54
SETDATA 55

```

```

SUBROUTINE SETXYZ
DIMENSION DEK(11),DAK(11),ZSK(11)
COMMON/ORBIT/ TP,XMDOT,FP,DFP,EPSI,AO,WN,DWN,COSS,WG,SINS
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,NLAT,XLON,NLON,SM,SV,GH,N,NDOP(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELON,GH,SAVE,OFFSAVE
COMMON/CONST1/ F4,E6,E7,XMMH,DEGRA,RAHIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IWEST,DUBIOUS,UNKNOWN
EQUIVALENCE (DEK,XSAT),(DAK,YSAT),(ZSK,ZSAT)

C COMPUTE TIME SINCE PERIGEE

TLOCK = TMINS(TLOCKON)
TFIX = INT((TLOCK-NDOP(N))*0.1+0.5)+10
FIXTIME = TSURVEY(TFIX)
TO = TFIX-TLOCK $ T1 = AMOD(TLOCK,1440.0)
TK = T1-TP
IF (TK.LT.-480.0) 590,600
590 TK = TK+1440.0 $ GO TO 620
600 IF (TK.GT.1440.0-6.2831857/XMDOT) TK = TK-1440.0

620 DO 650 J=1,9
DEK(J)=DEK(J)+E4*DEGRA
DAK(J)=DAK(J)+10.0
ZSK(J) = ZSK(J)+10.0
QMK=XMDOT*TK
EK=QMK*EPSI+SIN(QMK)-DEK(J)
AK=AO-DAK(J)
UK=AK-COS(EK)-EPSI
VK=AK-SIN(EK)
FK=FP-DFP*TK
XKP=UK-COS(FK)-VK-SIN(FK)
YKP=UK-SIN(FK)+VK-COS(FK)
BK=WN-DWN*TK
XSAT(J) = XKP-COS(BK)-YKP-COSS-SIN(BK)+ZSK(J)+SINS-SIN(BK)
YSAT(J) = XKP-SIN(BK)+YKP-COSS-COS(BK)-ZSK(J)+SINS-COS(BK)
ZSAT(J) = YKP-SINS-ZSK(J)+COSS
TK=TK+2.0
650 CONTINUE

RETURN
END

```

```

SFTXYZ 2
SETXYZ 3
SETXYZ 4
SETXYZ 5
SETXYZ 6
SETXYZ 7
SETXYZ 8
SETXYZ 9
SETXYZ 10
SETXYZ 11
SETXYZ 12
SETXYZ 13
SETXYZ 14
SETXYZ 15
SETXYZ 16
SETXYZ 17
SETXYZ 18
SETXYZ 19
SETXYZ 20
SETXYZ 21
SETXYZ 22
SETXYZ 23
SETXYZ 24
SETXYZ 25
SETXYZ 26
SETXYZ 27
SETXYZ 28
SETXYZ 29
SETXYZ 30
SETXYZ 31
SETXYZ 32
SETXYZ 33
SETXYZ 34
SETXYZ 35
SETXYZ 36
SETXYZ 37
SETXYZ 38
SETXYZ 39
SETXYZ 40
SETXYZ 41
SETXYZ 42
SETXYZ 43
SETXYZ 44
SETXYZ 45

```

```

SURROUTINE VELDATA(IFVEL)
DIMENSION STORE(32,120)
COMMON/ORBIT/ TP,XMDOT,FP,DFP,EP,SI,AD,MN,DWN,COSS,NG,SINS
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),CN(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,M,AT,XLON,NLON,BH,SV,GH,N,NDOP(8),TG
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELOM,GH,SAVE,OFFSAVE
COMMON/LUN/ MTIN,LUNDOC,LUNGH,LUNLTOF,LUNSAVE
COMMON/CONST1/ E4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IMEST,DURIOUS,UNKNOWN
DATA (ISTART=0),(CONST=111120.0),(CVEL=0.00202484)

IF (IFVEL.LT.3) 5,160

5 TFIRST = AMIN1(TLOCKON, FIXTIME)
IF (ISTART.EQ.0) 10,90

10 BUFFER IN (MTIN,1) (STORE(1),STORE(1920))
IF (UNIT(MTIN)) 40,220,20
20 PRINT 30, STORE(1)
30 FORMAT(/,10X,10(1M),10X,=PARITY ERROR IN RECORD STARTING=.F15.6)
40 IF (TFIRST.GT.STORE(1.60)*FIVESEC) 10,60

25 60 BUFFER IN (MTIN,1) (STORE(1921),STORE(3840))
IF (UNIT(MTIN)) 80,220,70
70 PRINT 30, STORE(1921)
80 ISTART = 1

30 90 IF (TFIRST.GT.STORE(1.120)*FIVESEC) 10,100
100 IF (TFIRST.LT.STORE(1.60)*FIVESEC) 110,105

105 DO 106 I=1,1920
STORE(I) = STORE(I-1920)
35 106 CONTINUE
GO TO 60

110 TSTART = THINS(STORE(1,1))
JFIX = THINS(FIXTIME)-TSTART+1.5
JSTART = THINS(TLOCKON)-TSTART+1.5 $ JSTOP = JSTART+16
IF (JFIX.GT.0.AND.JSTART.GT.0) 180,160

150 SLAT = -STORE(19,JFIX) $ SLOW = STORE(20,JFIX)
COSL = COS(SLAT*DEGRA)

45 KOUNT = 0 $ IFVEL = 2
DO 200 J=JSTART,JSTOP,2
KOUNT = KOUNT+1
IF (STORE(19,J).LT.DUBIOUS) 190,160
190 DN(KOUNT) = (-STORE(19,J)-SLAT)*CONST
DE(KOUNT) = (STORE(20,J)-SLOW)*COSL*CONST
200 CONTINUE

XLAT = SLAT $ XLON = SLOW
SUMN = DN(9)-DN(1) $ SUME = DE(9)-DE(1)
SV = SORT(SUMN**2+SUME**2)*CVEL
SH = (ATAN2(SUMN,SUMN))/DEGRA
IF (SH.LT.0.0) SH = SH+360.
IFVEL = 1 $ RETURN

60 220 IFVEL = 3
150 PRINT 170, FIXTIME
170 FORMAT(/,10X,10(1M),10X,=NO NAVIGATION DATA AT TIME=.F16.6)
IF (NLAT.EQ.1M) XLAT = -XLAT
IF (NLON.EQ.1M) XLON = -XLON

65 ENTRY VELSET

RSH = SH*DEGRA
RSV = SV*XMM
SVN = RSV*COS(RSH)
SVE = RSV*SIN(RSH)
DO 300 J=1,9
DT = (J-1)*2-TD
DN(J) = DT*SVN
DE(J) = DT*SVE
75 300 CONTINUE
RETURN

80 END

```

VELDATA 2
VELDATA 3
VELDATA 4
VELDATA 5
VELDATA 6
VELDATA 7
VELDATA 8
VELDATA 9
VELDATA 10
VELDATA 11
VELDATA 12
VELDATA 13
VELDATA 14
VELDATA 15
VELDATA 16
VELDATA 17
VELDATA 18
VELDATA 19
VELDATA 20
VELDATA 21
VELDATA 22
VELDATA 23
VELDATA 24
VELDATA 25
VELDATA 26
VELDATA 27
VELDATA 28
VELDATA 29
VELDATA 30
VELDATA 31
VELDATA 32
VELDATA 33
VELDATA 34
VELDATA 35
VELDATA 36
VELDATA 37
VELDATA 38
VELDATA 39
VELDATA 40
VELDATA 41
VELDATA 42
VELDATA 43
VELDATA 44
VELDATA 45
VELDATA 46
VELDATA 47
VELDATA 48
VELDATA 49
VELDATA 50
VELDATA 51
VELDATA 52
VELDATA 53
VELDATA 54
VELDATA 55
VELDATA 56
VELDATA 57
VELDATA 58
VELDATA 59
VELDATA 60
VELDATA 61
VELDATA 62
VELDATA 63
VELDATA 64
VELDATA 65
VELDATA 66
VELDATA 67
VELDATA 68
VELDATA 69
VELDATA 70
VELDATA 71
VELDATA 72
VELDATA 73
VELDATA 74
VELDATA 75
VELDATA 76
VELDATA 77
VELDATA 78
VELDATA 79
VELDATA 80
VELDATA 81

```

SUBROUTINE GEOIDHT(FIXTIME,XLAT,XLONG,HEIGHT,NOVAL)
DIMENSION GH(51,71)
COMMON/LUN/ MTIN,LUNDOC,LUNGH,LUNLTOF,LUNSAVE
DATA (ISTART=0)

IF (ISTART.EQ.0) 10.60

10 REWIND LUNGH
DO 20 I = 1,7
  K = I+10 $ J = K-9
  BUFFER IN(LUNGH,1) (GH(I,1),GH(I,K))
  IF (UNIT(LUNGH)) 20,30,20
20 CONTINUE
  BUFFER IN(LUNGH,1) (GH(I,71),GH(51,71))
  IF (UNIT(LUNGH)) 50,30,50
30 PRINT 40, LUNGH
40 FORMAT(//,10(1H*),* EOF FOUND ON GEOID HEIGHT FILE ON LUN*,13)
20 CALL ABORT
  ISTART = 1

60 RLAT = XLAT*1.0
  RLONG = XLONG*99.0
25 LAT1 = RLAT*0.0001 $ LONG1 = RLONG*0.0001
  LAT2 = LAT1*1 $ LONG2 = LONG1*1
  IF (LAT1.LT.1.OR.LAT2.GT.51) 90,75
  IF (LONG1.LT.1.OR.LONG2.GT.71) 90,80
30 GH1 = GH(LAT1,LONG1)-(GH(LAT2,LONG1)-GH(LAT1,LONG1))*(RLAT-LAT1)
  GH2 = GH(LAT1,LONG2)-(GH(LAT2,LONG2)-GH(LAT1,LONG2))*(RLAT-LAT1)
  HEIGHT = (GH1+(GH2-GH1)*(RLONG-LONG1))*0.3048
  NOVAL = 1 $ RETURN

35 90 PRINT 420, FIXTIME,XLAT,XLONG
  420 FORMAT(//,10X,10(1H*),10X,*FIX AT*,F15.6,* LAT*,F7.2,* LONG*,
  1 F8.2,* NOT WITHIN RANGE OF AVAILABLE GEOID HEIGHTS)

  NOVAL = 2 $ RETURN
40 END
GEOIDHT 2
GEOIDHT 3
GEOIDHT 4
GEOIDHT 5
GEOIDHT 6
GEOIDHT 7
GEOIDHT 8
GEOIDHT 9
GEOIDHT 10
GEOIDHT 11
GEOIDHT 12
GEOIDHT 13
GEOIDHT 14
GEOIDHT 15
GEOIDHT 16
GEOIDHT 17
GEOIDHT 18
GEOIDHT 19
GEOIDHT 20
GEOIDHT 21
GEOIDHT 22
GEOIDHT 23
GEOIDHT 24
GEOIDHT 25
GEOIDHT 26
GEOIDHT 27
GEOIDHT 28
GEOIDHT 29
GEOIDHT 30
GEOIDHT 31
GEOIDHT 32
GEOIDHT 33
GEOIDHT 34
GEOIDHT 35
GEOIDHT 36
GEOIDHT 37
GEOIDHT 38
GEOIDHT 39
GEOIDHT 40
GEOIDHT 41
GEOIDHT 42

SUBROUTINE OFFREQ(FIXTIME,IDENT,OFFSET,IFLTOF)
COMMON/OFFREQ/ ISURV,NSAT,NVAL,ISATNUM(5),LTOF(21,5)
COMMON/LUN/ MTIN,LUNDOC,LUNGH,LUNLTOF,LUNSAVE
REAL LTOF
DATA (ISURVEY=10000)

NSURVEY = FIXTIME*0.001
IF (NSURVEY.EQ.0) 100.10

10 ISURVEY = NSURVEY $ REWIND LUNLTOF
20 BUFFER IN(LUNLTOF,1) (ISURV,NVAL)
  IF (UNIT(LUNLTOF)) 50,30,50
30 PRINT 40, ISURVEY
40 FORMAT(//,10X,10(1H*),* EOF ON OFFSET READ - SURVEY*,13,* NOT ON LT
  1 OF FILE*)
  STOP 50
30 IF (ISURV.EQ.0) 50 IN 60
  CALL SKIP66(LUNLTOF,1)
  GO TO 20

60 BUFFER IN(LUNLTOF,1) (ISATNUM(1),ISATNUM(NSAT))
  IF (UNIT(LUNLTOF)) 80,70,80
70 STOP 60
25 50 BUFFER IN(LUNLTOF,1) (LTOF(1,1),LTOF(NVAL,NSAT))
  IF (UNIT(LUNLTOF)) 100,90,110
  90 STOP 70

100 DO 110 I = 1,NSAT
  IF (IDENT.EQ.ISATNUM(I)) 130,110
110 CONTINUE
  PRINT 120, IDENT
120 FORMAT(//,10X,10(1H*),* SATELLITE NO.*,13,* NOT FOUND IN LTOF FILE*)
  1 //)
35 IFLTOF = 2 $ RETURN

130 ISAT = I
  RDAY = (FIXTIME-ISURVEY)*170.0
  J = (RDAY-1.0)/5.0*1.0001
  IF (J.LT.1.OR.J*1.GT.NVAL) 140,160
40 NDAY = (NVAL-1)*5+1
  PRINT 150, FIXTIME,NDAY
150 FORMAT(//,10X,10(1H*),* FIXTIME*,F10.6,* LTOF FILE ONLY EXTENDS T
  10 DAY*,13,/)
  IFLTOF = 2 $ RETURN
45

160 IDAY = RDAY*0.1
  RMOUR = (RDAY-IDAY)*100.0
  IMOUR = RMOUR*0.1
  IMIN = (RMOUR-IMOUR)*100.0*0.1

  LASTDAY = (J-1)*5+1
  MIN = (IDAY-LASTDAY)*1440+IMOUR*60+IMIN

55 IF (LTOF(J,ISAT).EQ.0.0.OR.LTOF(J*1,ISAT).EQ.0.0) 170,190
  170 NEXTDAY = LASTDAY+5
  PRINT 180, LASTDAY,NEXTDAY,ISATNUM(ISAT)
180 FORMAT(//,10X,10(1H*),* NO LTOF AVAILABLE BETWEEN DAYS*,13,* AND*,
  1 13,* FOR SATELLITE*,13,/)
  IFLTOF = 2 $ RETURN
60

190 OFFSET = LTOF(J,ISAT)-(LTOF(J*1,ISAT)-LTOF(L,ISAT))*MIN/7200.0

  IFLTOF = 1 $ RETURN
65 END
OFFREQ 2
OFFREQ 3
OFFREQ 4
OFFREQ 5
OFFREQ 6
OFFREQ 7
OFFREQ 8
OFFREQ 9
OFFREQ 10
OFFREQ 11
OFFREQ 12
OFFREQ 13
OFFREQ 14
OFFREQ 15
OFFREQ 16
OFFREQ 17
OFFREQ 18
OFFREQ 19
OFFREQ 20
OFFREQ 21
OFFREQ 22
OFFREQ 23
OFFREQ 24
OFFREQ 25
OFFREQ 26
OFFREQ 27
OFFREQ 28
OFFREQ 29
OFFREQ 30
OFFREQ 31
OFFREQ 32
OFFREQ 33
OFFREQ 34
OFFREQ 35
OFFREQ 36
OFFREQ 37
OFFREQ 38
OFFREQ 39
OFFREQ 40
OFFREQ 41
OFFREQ 42
OFFREQ 43
OFFREQ 44
OFFREQ 45
OFFREQ 46
OFFREQ 47
OFFREQ 48
OFFREQ 49
OFFREQ 50
OFFREQ 51
OFFREQ 52
OFFREQ 53
OFFREQ 54
OFFREQ 55
OFFREQ 56
OFFREQ 57
OFFREQ 58
OFFREQ 59
OFFREQ 60
OFFREQ 61
OFFREQ 62
OFFREQ 63
OFFREQ 64
OFFREQ 65
OFFREQ 66
OFFREQ 67
OFFREQ 68

```

```

SUBROUTINE ASSESS1(IFEDIT,ILIST)
DIMENSION NSET(8)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DM(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,M/LAT,XLON,NLON,SM,SV,GM,N,NDOP(8),TO
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMX,ISIGN
COMMON/CONST1/ #4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IWEST,DUBIOUS,UNKNOWN
DATA (KOUNT+9*COUNT),(DMAX+15.0)
DMIN=1.0E10
CALL POS1(XLAT,XLON,SVN,SV,GM,TO,1,NDOP,N)
IF (RMS(1),LT,DMAX,AND,ILIST,EQ,3,AND,IFEDIT,EQ,1) RETURN
IF (ILIST,LT,3,OR,IFEDIT,EQ,2) PRINT 10
10 FORMAT(/,3X,=ASSESSMENT OF R.M.S. RESIDUAL (IN METRES) USING LONG-
TERM OFFSET FREQUENCY AND REMOVING ONE COUNT=)
DO 20 K=2,N
NSET(K-1) = NDOP(K)
20 CONTINUE
DO 100 K=1,N
RESID(K=1,NDOP(K)) = UNKNOWN
CALL POS1(XLAT,XLON,SVN,SV,GM,TO,K-1,NSET,N-1)
NSET(K) = NDOP(K)
IF (RMS(K=1),LT,DMIN) 90,100
90 DMIN = RMS(K=1) $ KMIN = K
100 CONTINUE
IF (ILIST,LT,3,OR,IFEDIT,EQ,2) 120,165
120 PRINT 130, (KOUNT,NDOP(K),K=1,N)
130 FORMAT(/,3X,=ALTITUDE=,7X,=ALL COUNTS=,3X,8(A5,13,4X) )
N1 = N-1
DO 150 J=1,8
PRINT 140,(ALT(I,J),I=1,2),(RESID(I,J),I=1,N1)
140 FORMAT(5X,F3.0,= TO =,F3.0,5X,(F9.2,3X))
150 CONTINUE
PRINT 160,(RMS(I),I=1,N1)
160 FORMAT(/,3X,=R.M.S. RESIDUALS =,9(F9.2,3X))
165 IF (RMS(1),LT,DMAX,OR,IFEDIT,EQ,2) RETURN
KDELETE = NDOP(KMIN)
ALT(1,KDELETE) = ALT(2,KDELETE) = UNKNOWN
DO 190 J=1,9
RESID(J,KDELETE) = UNKNOWN
190 CONTINUE
N=N-1
DO 200 K=KMIN,N
NDOP(K)=NDOP(K+1)
200 CONTINUE
IF (DMIN,LT,DMAX) 210,240
210 IF (ILIST,LT,3) PRINT 220,KDELETE,DMIN,DMAX
220 FORMAT(/,10X,10(1H=),10X,=REJECTION OF COUNT=,13,
1 = REDUCES R.M.S. RESIDUAL TO,F6.1,
1 = METRES = BELOW ACCEPTABLE LIMIT OF,F5.1,= METRES=)
RETURN
240 RMS(1) = DMIN
DO 250 K=1,N
RESID(1,K) = RESID(KMIN+1,K)
250 CONTINUE
IF (ILIST,LT,3) PRINT 170,KDELETE,DMIN
170 FORMAT(/,10X,10(1H=),10X,=REJECTION OF COUNT=,14,
1 = REDUCING R.M.S. RESIDUAL TO,F6.1,= METRES=)
IF(N,GT,3) 60,260
260 PRINT 270, DMIN,TLOCKON
270 FORMAT(/,10X,10(1H=),10X,=UNABLE TO REDUCE R.M.S. RESIDUAL BELOW,
1F6.1,= METRES FOR FIX AT,F15.6)
RETURN
END
```

```

SUBROUTINE ASSESS2(IPOS)
DIMENSION LIST(4),IT(8)
COMMON/FIX/TLOCKON,XLAT,M/LAT,XLON,NLON,SM,SV,GM,N,NDOP(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELONG,GHSAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMX,ISIGN
COMMON/CONST1/ #4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
DATA (IT(1),I=2,7)/10H NORTH,10HNORTH-EAST,10H EAST,
1 10HSOUTH-EAST,10H GEOID HT,10HDERIVED VN/
DATA (LIST=10HINPUT DATA,10HNEW OFFSET,9HNEW GEOID,9H NEW DVN)
IT(1) = LIST(IPOS)
GHKEEP = GM $ OFFKEEP = OFFSET
GM = GHSAVE $ OFFSET = OFFSAVE
IF (IPOS,GT,1) 10,20
10 CALL POS1(XLAT,XLON,0.0,0.0,GM,TO,8,NDOP,N)
SAVELAT = SLAT $ SLAT = XLAT+RAMIN
SAVELONG = SLONG $ SLONG = XLON+RAMIN
20 CALL ERROR(=30.87, 0.00, 0.0,1,2)
CALL ERROR(=21.83,=21.83, 0.0,2,3)
CALL ERROR( 0.00,=30.87, 0.0,3,4)
CALL ERROR(=21.83,=21.83, 0.0,4,5)
CALL ERROR( 0.00, 0.00,100.0,5,6)
CALL ERROR( 0.00, 0.00, 0.0,6,7)
IF (IPOS,EQ,1) 100,110
100 CALL ERROR( 0.00, 0.00, 0.0,7,8)
IT(8) = 9H AVGE VEL $ GO TO 120
110 CALL ERROR( 0.00, 0.00, 0.0,8,8)
IT(8) = 10HINPUT DATA
120 PRINT 230, (IT(I),I=1,8)
230 FORMAT(/,19X,=RESIDUALS IN METRES=,/,6X,=ALTITUDE=,3X,8(A10))
DO 250 J=1,8
PRINT 240, (ALT(I,J),I=1,2),(RESID(I,J),I=1,8)
240 FORMAT(5X,F3.0,= TO =,F3.0,8(5X,F9.2))
250 CONTINUE
PRINT 260, (RMS(I),I=1,8)
260 FORMAT(/,3X,=R.M.S. RESIDUALS =,8(F9.2,3X))
GM = GHKEEP $ OFFSET = OFFKEEP
RETURN
END
```

```

SUBROUTINE LISTFIX(IPOS)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),FN(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,NLAT,XLON,NLON,SH,SV,GM,N,NDOPI(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELONG,GM,SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMX,ISIGN
COMMON/CONST1/ E4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IWEST,DUBIOUS,UNKNOWN
LISTFIX 2
LISTFIX 3
LISTFIX 4
LISTFIX 5
LISTFIX 6
LISTFIX 7
LISTFIX 8
LISTFIX 9
LISTFIX 10
LISTFIX 11
LISTFIX 12
LISTFIX 13
LISTFIX 14
LISTFIX 15
LISTFIX 16
LISTFIX 17
LISTFIX 18
LISTFIX 19
LISTFIX 20
LISTFIX 21
LISTFIX 22
LISTFIX 23
LISTFIX 24
LISTFIX 25
LISTFIX 26
LISTFIX 27
LISTFIX 28
LISTFIX 29
LISTFIX 30
LISTFIX 31
LISTFIX 32
LISTFIX 33
LISTFIX 34
LISTFIX 35
LISTFIX 36
LISTFIX 37
LISTFIX 38
LISTFIX 39
LISTFIX 40
LISTFIX 41
LISTFIX 42
LISTFIX 43
LISTFIX 44
LISTFIX 45
LISTFIX 46
LISTFIX 47
LISTFIX 48
LISTFIX 49
LISTFIX 50
LISTFIX 51
LISTFIX 52

```

```

SUBROUTINE ERROR(DVN,DVE,DGM,ITYPE,KOUNT)
DIMENSION LABEL(4,8)
COMMON/FIX/TLOCKON,XLAT,NLAT,XLON,NLON,SH,SV,GM,N,NDOPI(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELONG,GM,SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMX,ISIGN
COMMON/CONST1/ E4,E6,E7,XMM,DEGRA,RAMIN,FIVESEC
COMMON/CONST2/ INORTH,ISOUTH,IEAST,IWEST,DUBIOUS,UNKNOWN
DATA (LABEL(1,1)+40HFIX SWIFT PER KNOT ERROR - NORTH )
DATA (LABEL(1,2)+40HFIX SWIFT PER KNOT ERROR - NORTHEAST )
DATA (LABEL(1,3)+40HFIX SWIFT PER KNOT ERROR - EAST )
DATA (LABEL(1,4)+40HFIX SWIFT PER KNOT ERROR - SOUTHEAST )
DATA (LABEL(1,5)+40HSWIFT CAUSED BY 100M ERROR IN GEOID )
DATA (LABEL(1,6)+40HSWIFT CAUSED BY DERIVING DVN )
DATA (LABEL(1,7)+40HSWIFT CAUSED BY USING AVERAGE VELOCITY )
DATA (LABEL(1,8)+40HSWIFT USING LATITUDE-LONGITUDE ONLY )
GO TO (10,10,10,10,10,20,30,40) ITYPE
10 CALL POS1(XLAT,XLON,DVN,DVE,GM,DGM,TO,KOUNT,NDOPI,N) $ GO TO 70
20 CALL POS4(XLAT,XLON,EVN,D,GM,TO,KOUNT,NDOPI,N)
LABEL3 = INORTH $ IF (EVN,LT,0.0) LABEL3 = ISOUTH
DIFFVN = ABS(EVN/XMM) $ GO TO 70
30 CALL VELSET(UU)
CALL POS1(XLAT,XLON,0.0,0.0,GM,TO,KOUNT,NDOPI,N) $ GO TO 70
40 XLAT = SLAT/RAMIN $ SLAT = SAVELAT
XLON = SLONG/RAMIN $ SLONG = SAVELONG
70 DLAT = XLAT*RAMIN $ DLONG = XLON*RAMIN
DLAT = DLAT-SLAT $ DLONG = DLONG-SLONG
LABEL1 = INORTH $ IF (DLAT,LT,0.0) LABEL1 = ISOUTH
LABEL2 = IEAST $ IF (DLONG,LT,0.0) LABEL2 = IWEST
DLAT = ABS(DLAT) $ DLONG = ABS(DLONG)
PRINT 80, (LABEL(1,ITYPE),I=1,4),DLAT,LABEL1,DLONG,LABEL2
80 FORMAT(10X,4A10.0,DLAT,F8.3,A3,5X,DLONG,F8.3,A3)
IF (ITYPE,EO,6) PRINT 110, DIFFVN,LABEL3
110 FORMAT(10X,90X,0DVN,F7.2,0 KNOTS TO THE,R9)
RETURN
END
ERROR 2
ERROR 3
ERROR 4
ERROR 5
ERROR 6
ERROR 7
ERROR 8
ERROR 9
ERROR 10
ERROR 11
ERROR 12
ERROR 13
ERROR 14
ERROR 15
ERROR 16
ERROR 17
ERROR 18
ERROR 19
ERROR 20
ERROR 21
ERROR 22
ERROR 23
ERROR 24
ERROR 25
ERROR 26
ERROR 27
ERROR 28
ERROR 29
ERROR 30
ERROR 31
ERROR 32
ERROR 33
ERROR 34
ERROR 35
ERROR 36
ERROR 37
ERROR 38
ERROR 39
ERROR 40
ERROR 41
ERROR 42
ERROR 43
ERROR 44
ERROR 45
ERROR 46
ERROR 47

```

```

SUBROUTINE POS1(RLAT,RLOX,SVN,SVE,GH,TC,KOUNT,NDOP,N)
DIMENSION NDOP(8)
DIMENSION A(2,2),B(2)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),PN(9),DE(9)
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELOX,G+SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMAX,ISIGN
NO SOL = 0
EQUATOR = 6378144.0 $ POLE = 6356757.0
ESQ = 1.0-(POLE/EQUATOR)**2 $ ALPHA=1.0-ESQ
ROFF = OFFSET*120.0*0.74948234
DO 70 ITER=1,20
B(1) = B(2) = 0.0
A(1,1) = A(1,2) = A(2,2) = 0.0
SINL = SIN(RLAT) $ COSL = COS(RLAT)
SINW = SIN(RLOX) $ COSW = COS(RLOX)
BETA = 1.0-ESQ*SINL**2
RPV = EQUATOR/SQRT(BETA)
RMP = RPV*ALPHA/BETA
XSTART = (RPV*GW)*COSL*COBW $ DXDL = -(RMP*GH)*SINL*COBW
YSTART = (RPV*GW)*COSL*SINW $ DYDL = -(RMP*GH)*SINL*SINW
ZSTART = (ALPHA*RPV*GW)*SINL $ DZDL = (RMP*GH)*COCL
DXDN = -SINL*COBW $ DXDE = -SINW
DYDN = -SINL*SINW $ DYDE = COSW
DZDN = COSL
D2XDLDN = -COSL*COBW
D2YDLDN = -COSL*SINW
D2ZDLDN = -SINL
ALTMAX = SUMSQ = 0.0
DO 20 I=1,N
DR = C1 = C2 = 0.0
K = 0 $ J1 = NDOP(1) $ J2 = J1+1
DO 10 J=J1,J2
K = K+1 $ DT = (J-1)*2-T0
DNJ = DN(J)*DT*SVN $ NEJ = DE(J)*DT*SVE
XSWIP = XSTART + DXDN*DNJ + DXDE*DEJ
YSWIP = YSTART + DYDN*DNJ + DYDE*DEJ
ZSWIP = ZSTART + DZDN*DNJ
XGRAD = DXDL + D2XDLDN*DNJ
YGRAD = DYDL + D2YDLDN*DNJ
ZGRAD = DZDL + D2ZDLDN*DNJ
C COMPUTE SLANT RANGE DIFFERENCES
DX = XSAT(J)-XSWIP
DY = YSAT(J)-YSWIP
DZ = ZSAT(J)-ZSWIP
DS = SQRT(DX**2+DY**2+DZ**2)
COSX = DX/DS $ COSY = DY/DS $ COSZ = DZ/DS
C COMPUTE TROPOSPHERIC CORRECTION
THETA = ASIN ((COSW*COCL*DX+SINW*COCL*DY+SINL*DZ)/DS)
DR = DS*TROPO(THETA,RLAT) -DR
C1 = -(COSX*XGRAD+COSY*YGRAD+COSZ*ZGRAD)-C1
C2 = COSX*YSWIP-COSY*XSWIP-C2
ALT(K,J1) = THETA * 57.3 $ ALTMAX = AMAX1(ALTMAX,ALT(K,J1))
ISIGN = 1 $ IF (1-SINW*DX+COSW*DY,LT.0.0) ISIGN = -1
70 CONTINUE
DL = DOP(J1)-ROFF-DR
A(1,1) = A(1,1)+C1*C1
A(1,2) = A(1,2)+C1*C2
A(2,2) = A(2,2)+C2*C2
B(1) = B(1)+DL*C1
B(2) = B(2)+DL*C2
RESID(KOUNT,J1) = DL
SUMSQ = SUMSQ+DL**2
20 CONTINUE
A(2,1) = A(1,2)
CALL MATINV (A(1,1),2,B(1),1,DET,IRR)
IF (IRR.NE.0) 30,50
30 PRINT 40, FIXTIME
40 FORMAT(5X,5(1H+),5X,*NO SOLUTION TO SATFIX MATRIX FOR FIX AT TIME*
1,F10.6)
NO SOL = 1
RETURN
90 DLAT = B(1) $ DLON = B(2)
RLAT = RLAT+DLAT
RLOX = RLOX+DLON
RMS(KOUNT) = SQRT(SUMSQ/N)
C SOLUTION WAS CONVERGED
IF (ABS(DLAT).LT.0.0000001) 60,70
60 IF (ABS(DLON).LT.0.0000001) RETURN
70 CONTINUE
C ITERATIVE SOLUTION DOES NOT CLOSE FAST ENOUGH
80 PRINT 100, FIXTIME
100 FORMAT(5X,5(1H+),5X,*NO SOLUTION AFTER 20 ITERATIONS FOR FIX*,
1,AT TIME*,F10.6)
NO SOL = 1
RETURN
END

```



```
SUBROUTINE POS2(RLAT,RLON,SVN,SVE,GH,TC,KOUNT,NDOP,N)
DIMENSION NDDP(8)
DIMENSION A(3,3),B(3)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),TW(9),DE(9)
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELONG,G+SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMAX,ISIGN
NO SOL = 0
EQUATOR = 6378144.0 $ POLE = 6356757.0
ESQ = 1.0-(POLE/EQUATOR)**2 $ ALPHA=1.0-ESQ
ROFF = OFFSET*120.0+0.74948234 $ C3 = 1.0
DO 80 ITER=1,20
  B(1) = B(2) = B(3) = 0.0
  A(1,1) = A(1,2) = A(1,3) = A(2,2) = A(2,3) = A(3,3) = 0.0
  SINL = SIN(RLAT) $ COSL = COS(RLAT)
  SINW = SIN(RLON) $ COSW = COS(RLON)
  BETA = 1.0-ESQ*SINL**2
  RPV = EQUATOR/SQRT(BETA)
  RMP = RPV*ALPHA/BETA
  XSTART = (RPV*GH)*COSL*COSW $ DXDL = -(RMP*GH)*SINL*COSW
  YSTART = (RPV*GH)*COSL*SINW $ DYDL = -(RMP*GH)*SINL*SINW
  ZSTART = (ALPHA*RPV*GH)*SINL $ DZDL = (RMP*GH)*COSL
  DXDN = -SINL*COSW $ DXDE = -SINW
  DYDN = -SINL*SINW $ DYDE = COSW
  DZDN = COSL
  D2XDLDN = -COSL*COSW
  D2YDLDN = -COSL*SINW
  D2ZDLDN = -SINL
  ALTMAX = SUMSQ = 0.0
  DO 20 I=1,N
    DR = C1 + C2 = 0.0
  K = 0 $ J1 = NDOP(1) $ J2 = J1+1
  DO 10 J=J1,J2
    K = K+1 $ DT = (J-1)*2-T0
    DNJ = DN(J)+DT*SVN $ DEJ = DE(J)+DT*SVE
    XSWIP = XSTART + DXDN*DNJ + DXDE*DEJ
    YSWIP = YSTART + DYDN*DNJ + DYDE*DEJ
    ZSWIP = ZSTART + DZDN*DNJ
    XGRAD = DXDL + D2XDLDN*DNJ
    YGRAD = DYDL + D2YDLDN*DNJ
    ZGRAD = DZDL + D2ZDLDN*DNJ
  C COMPUTE SLANT RANGE DIFFERENCES
  DX = XSAT(J)-XSWIP
  DY = YSAT(J)-YSWIP
  DZ = ZSAT(J)-ZSWIP
  DS = SQRT(DX**2+DY**2+DZ**2)
  COSX = DX/DS $ COSY = DY/DS $ COSZ = DZ/DS
  C COMPUTE TROPOSPHERIC CORRECTION
  THETA = ASIN ((COSW*COSL*DX+SINW*COSL*DY+SINL*DZ)/DS)
  DR = DS*TROPD(THETA,RLAT) -DR
  C1 = -(COSX*XGRAD+COSY*YGRAD+COSZ*ZGRAD)-C1
  C2 = COSX*YSWIP-COSY*XSWIP-C2
  ALTK(J,1) = THETA + 57.3 $ ALTMAX = AMAX1(ALTMAX,ALTK(J,1))
  ISIGN = 1 $ IF (-SINW*DX+COSW*DY,LT,0.0) ISIGN = -1
10 CONTINUE
  DL = DOP(J1)-ROFF-DR
  A(1,1) = A(1,1)+C1*C1
  A(1,2) = A(1,2)+C1*C2
  A(1,3) = A(1,3)+C1*C3
  A(2,2) = A(2,2)+C2*C2
  A(2,3) = A(2,3)+C2*C3
  A(3,3) = A(3,3)+C3*C3
  B(1) = B(1)+DL*C1
  B(2) = B(2)+DL*C2
  B(3) = B(3)+DL*C3
  RESID(KOUNT,J1) = DL
  SUMSQ = SUMSQ+DL**2
20 CONTINUE
  A(2,1) = A(1,2) $ A(3,1) = A(1,3) $ A(3,2) = A(2,3)
  CALL MATINV (A(1,1),3,B(1),1,DET,IRR)
  IF (IRR.NE.0) 30,50
30 PRINT 40, FIXTIME
40 FORMAT(5X,51H=),5X,=NO SOLUTION TO SATFIX MATRIX FOR FIX AT TIME=
  1,F10.6)
  NO SOL = 1
  RETURN
50 DLAT = B(1) $ DLON = B(2)
  DOFF = B(3)
  RLAT = RLAT+DLAT
  RLON = RLON+DLON
  ROFF = ROFF+DOFF
  RMS(KOUNT) = SQRT(SUMSQ/N)
  OFFSET = ROFF/(120.0+0.74948234)
  C SOLUTION WAS CONVERGED
  IF (ABS(DLAT).LT,0.00000001) 60,80
  IF (ABS(DLON).LT,0.00000001) 70,80
  IF (ABS(DOFF).LT,0.001) RETURN
80 CONTINUE
  C ITERATIVE SOLUTION DOES NOT CLOSE FAST ENOUGH
  90 PRINT 100, FIXTIME
  100 FORMAT(1X,51H=),5X,=NO SOLUTION AFTER 20 ITERATIONS FOR FIX=,
  1,AT TIME=F10.6)
  NO SOL = 1
  RETURN
END
```

```

SUBROUTINE POS3(RLAT,RLOM,SVN,SVE,GH,TD,KOUNT,NDOP,N)
DIMENSION NDOP(8)
DIMENSION A(3,3),B(3)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELOM,G-SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMAX,ISIGN
NO SOL = 0
EQUATOR = 6378144.0 $ POLE = 6356757.0
ESQ = 1.0-(POLE/EQUATOR)**2 $ ALPHA=1.0-ESQ
ROFF = OFFSET*170.0+0.74948234
DO 80 ITER=1,20
B(1) = B(2) = B(3) = 0.0
A(1,1) = A(1,2) = A(1,3) = A(2,2) = A(2,3) = A(3,3) = 0.0
SINL = SIN(RLAT) $ COSL = COS(RLAT)
SINW = SIN(RLOM) $ COSW = COS(RLOM)
BETA = 1.0-ESQ*SINL**2
RPV = EQUATOR/SQRT(BETA)
RMP = RPV*ALPHA/BETA
XSTART = (RPV*GH)*COSL*COSW $ DXDL = -(RMP*GH)*SINL*COSW
YSTART = (RPV*GH)*COSL*SINW $ DYDL = -(RMP*GH)*SINL*SINW
ZSTART = (ALPHA*RPV*GH)*SINL $ DZDL = (RMP*GH)*COSL
DXDN = -SINL*COSW $ DXDE = -SINW
DYDN = -SINL*SINW $ DYDE = COSW
DZDN = COSL
DZXDLN = -COSL*COSW
DZYDLN = -COSL*SINW
DZZDLN = -SINL
DXGW = COSL*COSW
DYGW = COSL*SINW
DZGW = SINL
ALTMAX = SUMSQ = 0.0
DO 20 J=1,N
DR = C1 = C2 = C3 = 0.0
K = 0 $ J1 = NDOP(1) $ J2 = J1-1
DO 10 J=J1,J2
K = K+1 $ DT = (J-1)*2-10
DNJ = DN(J)*DT*SVN $ NEJ = DE(J)*DT*SVE
XSHIP = XSTART + DXDN*DNJ + DXDE*DEJ
YSHIP = YSTART + DYDN*DNJ + DYDE*DEJ
ZSHIP = ZSTART + DZDN*DNJ
XGRAD = DXDL + DZXDLN*DNJ
YGRAD = DYDL + DZYDLN*DNJ
ZGRAD = DZDL + DZZDLN*DNJ
C
COMPUTE SLANT RANGE DIFFERENCES
DX = XSAT(J)-XSHIP
DY = YSAT(J)-YSHIP
DZ = ZSAT(J)-ZSHIP
DS = SQRT(DX**2+DY**2+DZ**2)
COSX = DX/DS $ COSY = DY/DS $ COSZ = DZ/DS
C
COMPUTE TROPOSPHERIC CORRECTION
THETA = ASIN ((COSX*COSL+DX*SINW+COSL*DY*SINL+DZ)/DS)
DR = DS*TROPO(THETA,RLAT) -DR
C1 = -(COSX*XGRAD+COSY*YGRAD+COSZ*ZGRAD)-C1
C2 = COSX*YSHIP-COSY*XSHIP-C2
C3 = -(DXGW+COSX*DYGW+COSY*DZGW+COSZ)-C3
ALT(K,J1) = THETA * 57.3 $ ALTMAX = AMAX1(ALTMAX,ALT(K,J1))
ISIGN = 1 $ IF (-SINW*DX+COSW*DY.LT.0.0) (SIGN = -1
10 CONTINUE
DL = DOP(J1)-ROFF-DR
A(1,1) = A(1,1)+C1*C1
A(1,2) = A(1,2)+C1*C2
A(1,3) = A(1,3)+C1*C3
A(2,2) = A(2,2)+C2*C2
A(2,3) = A(2,3)+C2*C3
A(3,3) = A(3,3)+C3*C3
B(1) = B(1)+DL*C1
B(2) = B(2)+DL*C2
B(3) = B(3)+DL*C3
RESID(KOUNT,J1) = DL
SUMSQ = SUMSQ+DL**2
20 CONTINUE
A(2,1) = A(1,2) $ A(3,1) = A(1,3) $ A(3,2) = A(2,3)
CALL MATINV (A(1,1),3,B(1),1,DET,IRR)
IF (IRR.NE.0) 30,50
30 PRINT 40, FIXTIME
40 FORMAT(5X,51H=),5X,=NO SOLUTION TO SATFIX MATRIX FOR FIX AT TIME=
1,F10.6)
NO SOL = 1
RETURN
90 DLAT = B(1) $ DLON = B(2)
DOM = B(3)
RLAT = RLAT+DLAT
RLOM = RLOM+DLON
GH = GH+DOM
RMS(KOUNT) = SQRT(SUMSQ/N)
C
SOLUTION HAS CONVERGED
IF (ABS(DLAT).LT.0.0000001) 60,80
80 IF (ABS(DLON).LT.0.0000001) 70,80
70 IF (ABS(DOM).LT.0.1) RETURN
80 CONTINUE
C
ITERATIVE SOLUTION DOES NOT CLOSE FAST ENOUGH
90 PRINT 100, FIXTIME
100 FORMAT(7X,51H=),5X,=NO SOLUTION AFTER 20 ITERATIONS FOR FIX=,
1 = AT TIME=F10.6)
NO SOL = 1
RETURN
120 END

```

```

SUBROUTINE POS4(RLAT,RLON,SVN,SVE,GH,TO,KOUNT,NDOF,N)
DIMENSION NDOF(8)
DIMENSION A(3,3),B(3)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVELOM,GHSAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9),RMS(9),ALT(2,8),ALTMAX,ISIGN
NO SOL = 0
EQUATOR = 6378144.0 $ POLE = 6356757.0
ESO = 1.0-(POLE/EQUATOR)**2 $ ALPHA=1.0-ESO
ROFF = OFFSET*220.0+0.74948234
SVN = 0.0
DO 80 ITER=1,20
  B(1) = B(2) = B(3) = 0.0
  A(1,1) = A(1,2) = A(1,3) = A(2,2) = A(2,3) = A(3,3) = 0.0
  SINL = SIN(RLAT) $ COSL = COS(RLAT)
  SINW = SIN(RLON) $ COSW = COS(RLON)
  BETA = 1.0-ESO*SINL**2
  RPV = EQUATOR/SQRT(BETA)
  RMP = RPV*ALPHA/BETA
  XSTART = (RPV*GH)+COSL*CORNW $ DXDL = -(RMP*GH)*SINL*COSW
  YSTART = (RPV*GH)+COSL*SINW $ DYDL = -(RMP*GH)*SINL*SINW
  ZSTART = (ALPHA*RPV*GH)*SINL $ DZDL = (RMP*GH)*COSL
  DXDN = -SINL*COSW $ DXDE = -SINW
  DYDN = -SINL*SINW $ DYDE = COSW
  DZDN = COSL
  D2XDLOM = -COSL*COSW
  D2YDLOM = -COSL*SINW
  D2ZDLOM = -SINL
  ALTMAX = SUMSQ = 0.0
  DO 20 I=1,N
    DR = C1 = C2 = C3 = 0.0
    K = 0 $ J1 = NDOF(I) $ J2 = J1+1
    DO 10 J=J1,J2
      K = K+1 $ DT = (J-1)*2-T0
      DNJ = DN(J)*DT*SVN
      XSHIP = XSTART + DXDN*DNJ + DXDE*DE(J)
      YSHIP = YSTART + DYDN*DNJ + DYDE*DE(J)
      ZSHIP = ZSTART + DZDN*DNJ
      XGRAD = DXDL + D2XDLOM*DNJ
      YGRAD = DYDL + D2YDLOM*DNJ
      ZGRAD = DZDL + D2ZDLOM*DNJ
      XGRADVN = DT*DXDN
      YGRADVN = DT*DYDN
      ZGRADVN = DT*DZDN
      C COMPUTE SLANT RANGE DIFFERENCES
      DX = XSAT(J)-XSHIP
      DY = YSAT(J)-YSHIP
      DZ = ZSAT(J)-ZSHIP
      DS = SQRT(DX**2+DY**2+DZ**2)
      COSX = DX/DS $ COSY = DY/DS $ COSZ = DZ/DS
      C COMPUTE TROPOSPHERIC CORRECTION
      THETA = ASIN ((COSW*COSL*DX+SINW*COSL*DY+SINL*DZ)/DS)
      DR = DS-TROPO(THETA,RLAT) -DR
      C1 = -(COSX*XGRAD+COSY*YGRAD+COSZ*ZGRAD)-C1
      C2 = COSX*YSHIP-COSY*XSHIP-C2
      C3 = -(COSX*XGRADVN+COSY*YGRADVN+COSZ*ZGRADVN)-C3
      ALT(K,J1) = THETA * 57.3 $ ALTMAX = AMAX1(ALTMAX,ALT(K,J1))
      ISIGN = 1 $ IF (-SINW*DX+COSW*DY.LT.0.0) ISIGN = -1
10 CONTINUE
      DL = DOP(J1)-ROFF-DR
      A(1,1) = A(1,1)+C1*C1
      A(1,2) = A(1,2)+C1*C2
      A(1,3) = A(1,3)+C1*C3
      A(2,2) = A(2,2)+C2*C2
      A(2,3) = A(2,3)+C2*C3
      A(3,3) = A(3,3)+C3*C3
      B(1) = B(1)+DL*C1
      B(2) = B(2)+DL*C2
      B(3) = B(3)+DL*C3
      RESID(KOUNT,J1) = DL
      SUMSQ = SUMSQ+DL**2
20 CONTINUE
      A(2,1) = A(1,2) $ A(3,1) = A(1,3) $ A(3,2) = A(2,3)
      CALL MATINV (A(1,1),3,B(1),1,DET,IRR)
      IF (IRR.NE.0) 30,50
30 PRINT 40, FIXTIME
40 FORMAT(5X,5(IH+),5X,NO SOLUTION TO SATFIX MATRIX FOR FIX AT TIME=
1, F10.6)
NO SOL = 1
RETURN
50 DLAT = B(1) $ DLOM = B(2)
DVN = B(3)
RLAT = RLAT+DLAT
RLON = RLON+DLOM
SVN = SVN+DVN
RMS(KOUNT) = SQRT(SUMSQ/4)
C SOLUTION HAS CONVERGED
IF (ABS(DLAT).LT.0.0000001) 60,80
80 IF (ABS(DLOM).LT.0.0000001) 70,80
70 IF (ABS(DVN).LT.0.01) RETURN
80 CONTINUE
115 C ITERATIVE SOLUTION DOES NOT CLOSE FAST ENOUGH
90 PRINT 100, FIXTIME
100 FORMAT(/,5X,5(IH+),5X,NO SOLUTION AFTER 20 ITERATIONS FOR FIX=,
1, F10.6)
NO SOL = 1
RETURN
END

```

	SUBROUTINE MATINV(A,N,B,I,D,ERROR)	MATINV	2
	DIMENSION A(N,N),B(N,1),IPIV(24),IND(24,2)	MATINV	3
	A IS AN N*N MATRIX TO BE INVERTED,OR CONTAINING EQUATION COEFFS	MATINV	4
	B IS AN N*N RHS MATRIX FOR EQUATIONS	MATINV	5
5	C IF L=0,INVERSE ONLY GIVEN,L POSITIVE,SOLUTIONS ONLY,L NEGATIVE	MATINV	6
	C BOTH, M=ABS(L)	MATINV	7
	C D CONTAINS THE DETERMINANT OF THE A MATRIX ON EXIT	MATINV	8
	C A IS REPLACED BY THE INVERSE ,B BY THE SOLUTIONS,	MATINV	9
10	C METHOD OF GAUSS-JORDON PIVOTAL ELIMINATION	MATINV	10
	M=ABS(L)	MATINV	11
	D=1.0	MATINV	12
	DO 10 I=1,N	MATINV	13
	10 IPIV(I)=0	MATINV	14
	DO 220 I=1,N	MATINV	15
15	AMAX=0.0	MATINV	16
	C SEARCH SUB-MATRIX FOR LARGEST ELEMENT AS PIVOT	MATINV	17
	DO 60 J=1,N	MATINV	18
	IF (IPIV(J)) 70,20.60	MATINV	19
20	DO 50 K=1,N	MATINV	20
	IF (IPIV(K))=1 30,50.70	MATINV	21
	C THIS ROW COLUMN HAS NOT BEEN A PIVOT	MATINV	22
	30 IF (ABS (A(J,K))-AMAX) 50,50.40	MATINV	23
	40 IROW=J	MATINV	24
	ICOL=K	MATINV	25
25	AMAX=ABS (A(J,K))	MATINV	26
	50 CONTINUE	MATINV	27
	60 CONTINUE	MATINV	28
	C PIVOT FOUND	MATINV	29
	IPIV(ICOL)=IPIV(ICOL)+1	MATINV	30
30	IF (AMAX-1.0E-90) 70,70.00	MATINV	31
	C MATRIX SINGULAR,ERROR RETURN	MATINV	32
	70 ERROR=1	MATINV	33
	RETURN	MATINV	34
	80 IF (IROW=ICOL) 90,130.90	MATINV	35
35	C MAKE PIVOT A DIAGONAL ELEMENT BY ROW INTERCHANGE.	MATINV	36
	90 D=-D	MATINV	37
	DO 100 K=1,N	MATINV	38
	AMAX=A(IROW,K)	MATINV	39
	A(IROW,K)=A(ICOL,K)	MATINV	40
40	100 A(ICOL,K)=AMAX	MATINV	41
	IF (M) 130,130.110	MATINV	42
	110 DO 120 K=1,N	MATINV	43
	AMAX=B(IROW,K)	MATINV	44
	B(IROW,K)=B(ICOL,K)	MATINV	45
45	120 B(ICOL,K)=AMAX	MATINV	46
	130 IND(1,1)=IROW	MATINV	47
	IND(1,2)=ICOL	MATINV	48
	AMAX=A(ICOL,ICOL)	MATINV	49
	D=D*AMAX	MATINV	50
50	A(ICOL,ICOL)=1.0	MATINV	51
	C DIVIDE PIVOT ROW BY PIVOT	MATINV	52
	DO 140 K=1,N	MATINV	53
	140 A(ICOL,K)=A(ICOL,K)/AMAX	MATINV	54
	IF (M) 170,170.150	MATINV	55
55	DO 160 K=1,N	MATINV	56
	160 B(ICOL,K)=B(ICOL,K)/AMAX	MATINV	57
	C REDUCE NON-PIVOT ROWS	MATINV	58
	DO 220 J=1,N	MATINV	59
	IF (J=ICOL) 180,220.180	MATINV	60
60	180 AMAX=A(J,ICOL)	MATINV	61
	A(J,ICOL)=0.0	MATINV	62
	DO 190 K=1,N	MATINV	63
	190 A(J,K)=A(J,K)-A(ICOL,K)*AMAX	MATINV	64
	IF (M) 220,220.200	MATINV	65
65	DO 210 K=1,N	MATINV	66
	210 B(J,K)=B(J,K)-B(ICOL,K)*AMAX	MATINV	67
	220 CONTINUE	MATINV	68
	C AFTER N PIVOTAL CONDENSATIONS,SOLUTIONS LIE IN B MATRIX	MATINV	69
	IF (L) 230,230.270	MATINV	70
70	C FOR INVERSE OF A, INTERCHANGE COLUMNS	MATINV	71
	DO 260 I=1,N	MATINV	72
	J=N+1-I	MATINV	73
	IF (IND(J,1)=IND(J,2)) 240,260.240	MATINV	74
75	240 IROW=IND(J,1)	MATINV	75
	ICOL=IND(J,2)	MATINV	76
	DO 250 K=1,N	MATINV	77
	AMAX=A(K,IROW)	MATINV	78
	A(K,IROW)=A(K,ICOL)	MATINV	79
	250 A(K,ICOL)=AMAX	MATINV	80
80	260 CONTINUE	MATINV	81
	270 ERROR=0	MATINV	82
	RETURN	MATINV	83
	END	MATINV	84
		MATINV	85

```

SUBROUTINE SATSAVE(IPRINT)
COMMON/XYZ/ DOP(8),REF(8),XSAT(11),YSAT(11),ZSAT(11),DN(9),DE(9)
COMMON/FIX/TLOCKON,XLAT,M1AT,XLON,M1ON,SM,SV,GM,N,NDDP(8),TO
COMMON/SAVE/FIXTIME,IDENT(3),OFFSET,SAVELAT,SAVEON,GS,SAVE,OFFSAVE
COMMON/ASSESS/SLAT,SLONG,RESID(9,8),RMS(9),ALT(2,8),ALTMAX,ISIGN
COMMON/LUN/ MTIN,LUNDOC,LINGM,LUNLTOF,LUNSAVE
COMMON/CONST/ F4,E6,E7,XMM,DEORA,RANIN,FIVESEC

10      DLAT = ABS(SLAT)      $ DIONG = ABS(SLONG)
      LAT = DLAT/60.0      $ DIAT = DLAT-60.0*LAT
      LONG = DLONG/60.0      $ DIONG = DLONG-60.0*LONG

15      LABEL1 = 1MW      $ IF (ZSAT(9).LT.ZSAT(1)) LABEL1 = 1WS
      LABEL2 = 1ME      $ IF (ISIGN.LT.0.0) LABEL2 = 1MW
      IF (XLAT.GT.0.0) LAT = -LAT
      IF (XLON.LT.0.0) LONG = -LONG

20      IF (IPRINT.EQ.1) PRINT 80,FIXTIME,LAT,DLAT,SLONG,DLONG,IDENT(2),
      1 SM,SV,GM,IDENT(3),LABEL1,ALTMAX,LABEL2,OFFSET
80      FORMAT(/,10X,1M,9.6,1X,2(14,F6.2),1X,R4,F5.0,2F5.1,15.4X,A1,
      1 F5.0,4X,A1,F10.2,1M)

25      WRITE(LUNSAVE,90) FIXTIME,LAT,DLAT,SLONG,DLONG,IDENT(2),SM,SV,GM,
      1 IDENT(3),LABEL1,ALTMAX,LABEL2,OFFSET
90      FORMAT(F9.6,1X,2(14,F6.2),1X,R4,F5.0,2F5.1,15.4X,A1,F5.0,4X,A1,
      1 F10.2)

30      RETURN
      END

```

SATSAVE 2
SATSAVE 3
SATSAVE 4
SATSAVE 5
SATSAVE 6
SATSAVE 7
SATSAVE 8
SATSAVE 9
SATSAVE 10
SATSAVE 11
SATSAVE 12
SATSAVE 13
SATSAVE 14
SATSAVE 15
SATSAVE 16
SATSAVE 17
SATSAVE 18
SATSAVE 19
SATSAVE 20
SATSAVE 21
SATSAVE 22
SATSAVE 23
SATSAVE 24
SATSAVE 25
SATSAVE 26
SATSAVE 27
SATSAVE 28
SATSAVE 29
SATSAVE 30
SATSAVE 31
SATSAVE 32

```

SUBROUTINE SATLIST
DIMENSION LABEL(8),ICARD(8)
COMMON/LUN/ MTIN,LUNDOC,LINGM,LUNLTOF,LUNSAVE

5      REWIND LUNSAVE      $ KOUNT = 0

      READ(LUNSAVE,10) LABEL
10      FORMAT(8A10)
      IF (.EOF.(LUNSAVE)) 999,20
20      IF (.DCEC(LUNSAVE)) 30,50
30      PRINT 40, LABEL
40      FORMAT(/,10X,10(1M),10X,4PARITY ERROR IN *.1M,8A10,1M)

15      50 READ(LUNSAVE,10) ICARD
      IF (.EOF.(LUNSAVE)) 999,60
60      IF (.DCEC(LUNSAVE)) 70,80
70      PRINT 40, ICARD

20      90 IF ((KOUNT/50)*50.EQ.KOUNT) PRINT 90, LABEL,LUNSAVE
90      FORMAT(1M,/,10X,4FILE *.1M,8A10,1M,*, SAVED ON LUN*,14,/,
      1 19X,4FIXTIME LATITUDE LONGITUDE LOCK SHIP SHIP GEOID SAT*,
      2 4 4 SAT E OFFSET,/,10X,4SS,DDMMHH DEG MIN DEG MIN*,
      3 4 TIME WDD SPD HT NO $ ELEV M FREQ*)
25      IF ((KOUNT/10)*10.EQ.KOUNT) PRINT 100
100      FORMAT(10X)

      KOUNT = KOUNT+1
      PRINT 110, KOUNT,ICARD
30      110 FORMAT(10X,15.2X,1M,8A10,1M)
      GO TO 90

999      RETURN

35      END

```

SATLIST 2
SATLIST 3
SATLIST 4
SATLIST 5
SATLIST 6
SATLIST 7
SATLIST 8
SATLIST 9
SATLIST 10
SATLIST 11
SATLIST 12
SATLIST 13
SATLIST 14
SATLIST 15
SATLIST 16
SATLIST 17
SATLIST 18
SATLIST 19
SATLIST 20
SATLIST 21
SATLIST 22
SATLIST 23
SATLIST 24
SATLIST 25
SATLIST 26
SATLIST 27
SATLIST 28
SATLIST 29
SATLIST 30
SATLIST 31
SATLIST 32
SATLIST 33
SATLIST 34
SATLIST 35
SATLIST 36

```

SUBROUTINE SKIP66(LUN,NPARITY)

C      THIS SUBROUTINE HAS TWO ENTRY POINTS
C      1. SKIP66 = SKIP FORWARD ALONG LOGICAL UNIT UNTIL AN
C      END-OF-FILE MARK DETECTED
C      2. BACK66 = MOVE BACK ALONG TAPE UNTIL EOF DETECTED

C      LUN = LOGICAL UNIT ON WHICH SEARCH IS TO BE MADE
C      NPARITY = PARITY OF SEARCH MODE FOR LUN CONCERNED
C      1 = ODD PARITY (AS PER BUFFER)
C      2 = EVEN PARITY (AS PER BUFFER)
C      -1 = IN READ MODE FOR FORMATTED DATA

15      DIMENSION A(2)

      IF (NPARITY.GE.0) 10,30

10      BUFFER IN(LUN,NPARITY) A(1),A(2)
      IF (UNIT(LUN))10,20,10
20      RETURN

30      READ(LUN,40) A(1)
40      FORMAT(A10)
      IF (.EOF.(LUN)) 60,50
25      50 IF (.DCEC(LUN)) 30,30
60      RETURN

      ENTRY BACK66

30      BACKSPACE LUN
      IF (NPARITY.GE.0) 70,100

70      BUFFER IN(LUN,NPARITY) A(1),A(2)
      IF (UNIT(LUN)) 80,90,80
35      80 BACKSPACE LUN
      BACKSPACE LUN
      GO TO 70
90      BACKSPACE LUN
      RETURN

40      100 READ(LUN,40) A(1)
      IF (.EOF.(LUN)) 130,110
110      IF (.DCEC(LUN)) 120,120
120      BACKSPACE LUN
      BACKSPACE LUN
      GO TO 100
130      BACKSPACE LUN
      RETURN

45      END

```

SKIP66 2
SKIP66 3
SKIP66 4
SKIP66 5
SKIP66 6
SKIP66 7
SKIP66 8
SKIP66 9
SKIP66 10
SKIP66 11
SKIP66 12
SKIP66 13
SKIP66 14
SKIP66 15
SKIP66 16
SKIP66 17
SKIP66 18
SKIP66 19
SKIP66 20
SKIP66 21
SKIP66 22
SKIP66 23
SKIP66 24
SKIP66 25
SKIP66 26
SKIP66 27
SKIP66 28
SKIP66 29
SKIP66 30
SKIP66 31
SKIP66 32
SKIP66 33
SKIP66 34
SKIP66 35
SKIP66 36
SKIP66 37
SKIP66 38
SKIP66 39
SKIP66 40
SKIP66 41
SKIP66 42
SKIP66 43
SKIP66 44
SKIP66 45
SKIP66 46
SKIP66 47
SKIP66 48
SKIP66 49
SKIP66 50
SKIP66 51

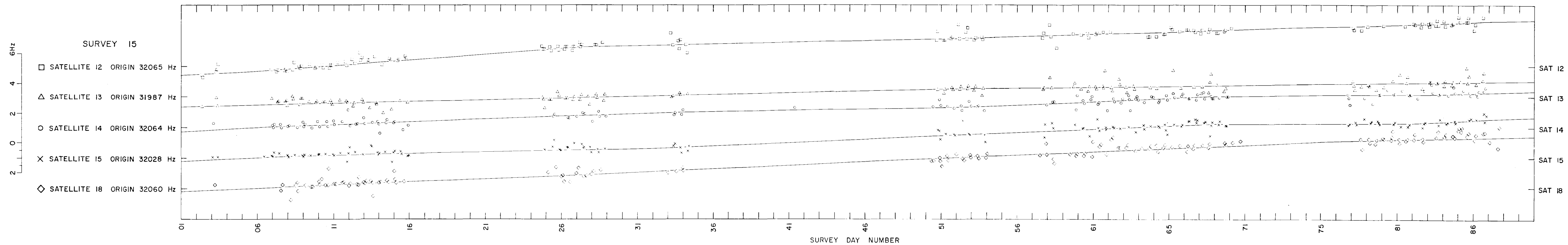
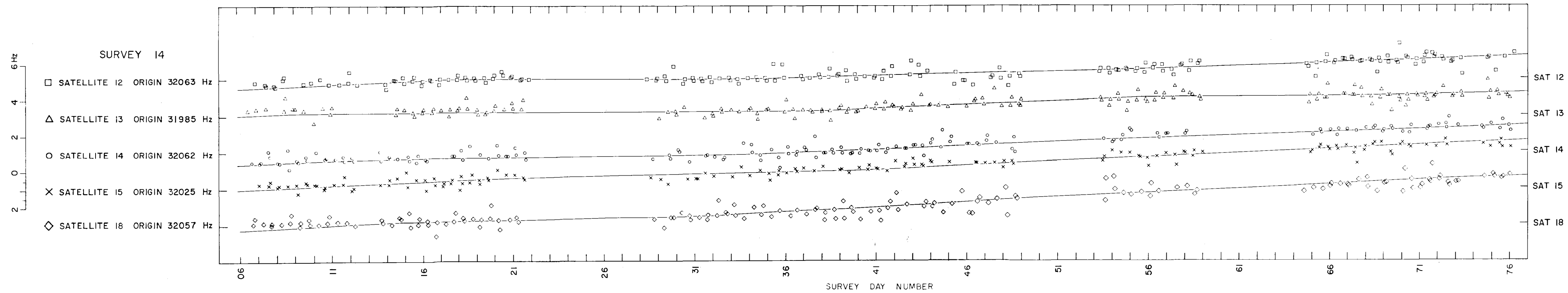
	SUBROUTINE HIGHSAT(ALTMAX)	HIGHSAT	2
	COMMON/FIX/TLOCKON,XLAT,XLAT,XLON,XLON,BH,SV,GH,N,NDOP(8),T0	HIGHSAT	3
5	TSTART = TMIN(TLOCKON)	HIGHSAT	4
	ALTMAX = 0.0	HIGHSAT	5
	DO 20 K=1,9	HIGHSAT	6
	CALL ALTSAT(TSTART,XLAT,XLON,GH,ALT)	HIGHSAT	7
10	TIME = TSURVEY(TSTART)	HIGHSAT	8
	IF (ALT,GT,ALTMAX) 10.48	HIGHSAT	9
	10 ALTMAX = ALT	HIGHSAT	10
	TSTART = TSTART+2.0	HIGHSAT	11
15	20 CONTINUE	HIGHSAT	12
	GO TO 80	HIGHSAT	13
	40 TSTART = TSTART-4.0	HIGHSAT	14
	ALTMAX = 0.0	HIGHSAT	15
20	DO 60 K=1,21	HIGHSAT	16
	CALL ALTSAT(TSTART,XLAT,XLON,GH,ALT)	HIGHSAT	17
	IF (ALT,GT,ALTMAX) 50.70	HIGHSAT	18
	50 ALTMAX = ALT	HIGHSAT	19
	TSTART = TSTART+0.2	HIGHSAT	20
25	60 CONTINUE	HIGHSAT	21
	GO TO 80	HIGHSAT	22
	70 RETURN	HIGHSAT	23
30	80 PRINT 130	HIGHSAT	24
	130 FORMAT(/,10X,10(1H),10X,=MAXIMUM ALTITUDE NOT FOUND=)	HIGHSAT	25
	ALTMAX = 1.0E10 \$ RETURN	HIGHSAT	26
	END	HIGHSAT	27
		HIGHSAT	28
		HIGHSAT	29
		HIGHSAT	30
		HIGHSAT	31
		HIGHSAT	32
		HIGHSAT	33
		HIGHSAT	34
		HIGHSAT	35

	SUBROUTINE ALTSAT(TIME,RLAT,RLONG,GH,ALT)	ALTSAT	2
	COMMON/ORB17/ TP,XMDOT,FP,DFP,EP,SI,AD,WN,DWN,CSS,WG,SINS	ALTSAT	3
	DATA (TSTART=0)	ALTSAT	4
5	IF (TSTART,EO,0) 10.20	ALTSAT	5
	10 EQUATOR = 6378144.0 \$ POLE = 6356757.0	ALTSAT	6
	ESQ = 1.0-(POLE/EQUATOR)**2	ALTSAT	7
10	ALPHA = 1.0-ESQ \$ TSTART = 1	ALTSAT	8
	20 SINL = SIN(RLAT) \$ COSL = COS(RLAT)	ALTSAT	9
	SINW = SIN(RLONG) \$ COSW = COS(RLONG)	ALTSAT	10
	BETA = 1.0-ESQ*SINL**2	ALTSAT	11
15	RPV = EQUATOR/SQRT(BETA)	ALTSAT	12
	RWP = RPV*ALPHA/BETA	ALTSAT	13
	XSHIP = (RPV*GH)*COSL*COGW	ALTSAT	14
	YSHIP = (RPV*GH)*COSL*SINW	ALTSAT	15
20	ZSHIP = (ALPHA*RPV*GH)*SINL	ALTSAT	16
	TK = AMOD(TIME,1440.0)-TP	ALTSAT	17
	IF (TK,LT,-480.0) 590,800	ALTSAT	18
	590 TK = TK+1440.0 \$ GO TO 620	ALTSAT	19
25	800 IF (TK,GT,1440.0-6.2831852/XMDOT) TK = TK-1440.0	ALTSAT	20
	620 QMK = XMDOT*TK	ALTSAT	21
	EK = QMK-EP*SI*SIN(QMK)	ALTSAT	22
	UK = AD*(COS(EK)-EP*SI)	ALTSAT	23
30	VK = AD*SIN(EK)	ALTSAT	24
	FK = FP-DFP*TK	ALTSAT	25
	XNP = UK*(COS(FK)-VK*SIN(FK))	ALTSAT	26
	YKP = UK*SIN(FK)+VK*COS(FK)	ALTSAT	27
	BK = WN*DWN*TK	ALTSAT	28
35	XSAT = XNP*(COS(BK)-YKP*COS(SIN(BK))	ALTSAT	29
	YSAT = XNP*(SIN(BK)+YKP*COS(COS(BK))	ALTSAT	30
	ZSAT = YKP*SINS	ALTSAT	31
	DX = XSAT-XSHIP \$ DY = YSAT-YSHIP \$ DZ = ZSAT-ZSHIP	ALTSAT	32
40	DS = SQRT(DX**2+DY**2+DZ**2)	ALTSAT	33
	SINALT = (COSL*COGW*DX+COSL*SINW*DY+SINL*DZ)/DS	ALTSAT	34
	IF (SINALT,GT,1.0) SINALT = 1.0	ALTSAT	35
45	ALT = ASIN(SINALT)*57.3	ALTSAT	36
	RETURN	ALTSAT	37
	END	ALTSAT	38
		ALTSAT	39
		ALTSAT	40
		ALTSAT	41
		ALTSAT	42
		ALTSAT	43
		ALTSAT	44
		ALTSAT	45
		ALTSAT	46
		ALTSAT	47
		ALTSAT	48
		ALTSAT	49

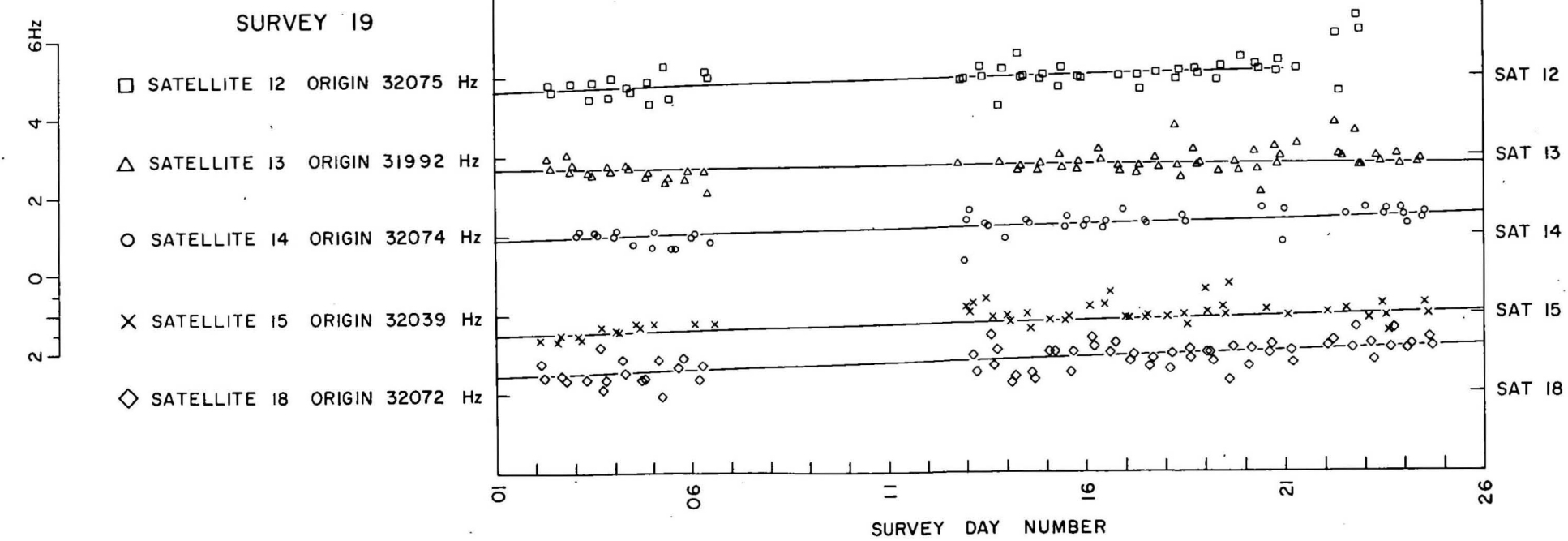
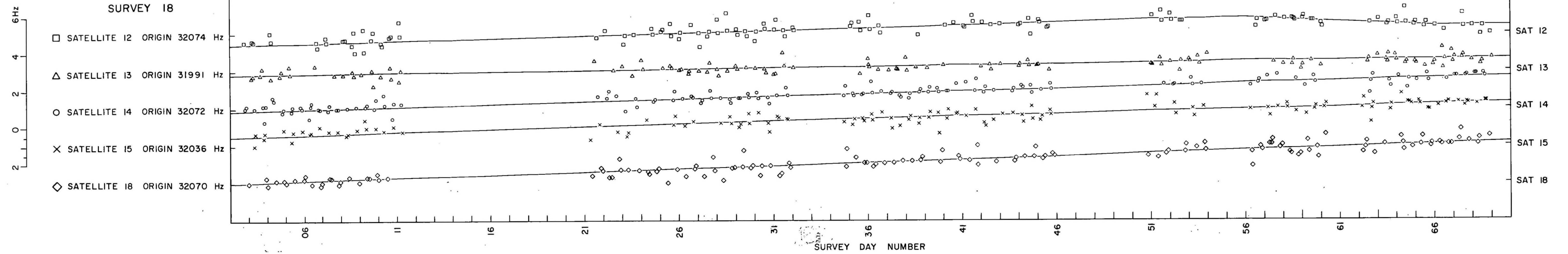
	FUNCTION TROPO(THETA,RLAT)	TROPO	2
	DATA (1START=0)	TROPO	3
5	IF (1START.EQ.0) 10,20	TROPO	4
	10 C1 = 2.5/57.2957795	TROPO	5
	C2 = 1.5/57.2957795	TROPO	6
	1START = 1	TROPO	7
10	20 THETA0 = SORT(THETA**2+C1**2)	TROPO	8
	THETA0 = SORT(THETA**2+C2**2)	TROPO	9
		TROPO	10
		TROPO	11
		TROPO	12
		TROPO	13
		TROPO	14
15	RD = 2.31/SIN(THETA0)	TROPO	15
	RH = 0.20/SIN(THETA0)	TROPO	16
		TROPO	17
	TROPO = RD+RH	TROPO	18
		TROPO	19
20	RETURN	TROPO	20
	END	TROPO	21
		TROPO	22

	FUNCTION TSURVEY(RHINS)	TSURVEY	2
	C** THIS FUNCTION CONVERTS CONTINUOUS RUNNING TIME IN MINUTES	TSURVEY	3
	C** INTO SURVEY TIME IN FORMAT "SS.DDMMHCC" WHERE	TSURVEY	4
5	C** SS = SURVEY NUMBER	TSURVEY	5
	C** DD = SURVEY DAY	TSURVEY	6
	C** HH = HOURS G.M.T.	TSURVEY	7
	C** MM = MINUTES G.M.T.	TSURVEY	8
	C** CC = SECONDS G.M.T.	TSURVEY	9
10		TSURVEY	10
	ISEC = RHINS*60.8 + 0.5	TSURVEY	11
		TSURVEY	12
	IMIN = ISEC/60	TSURVEY	13
	IMOU = IMIN/60	TSURVEY	14
15	IDAY = IMOU/24	TSURVEY	15
	ISURVEY = IDAY/100	TSURVEY	16
		TSURVEY	17
	ISEC = ISEC - IMIN*60	TSURVEY	18
	IMIN = IMIN - IMOU*60	TSURVEY	19
20	IMOU = IMOU - IDAY*24	TSURVEY	20
	IDAY = IDAY - ISURVEY*100	TSURVEY	21
		TSURVEY	22
	ITIME = ISURVEY*100000000+IDAY*1000000+IMOU*10000+IMIN*1000+ISEC	TSURVEY	23
25	TSURVEY = ITIME*1.0E-8	TSURVEY	24
		TSURVEY	25
	RETURN	TSURVEY	26
	END	TSURVEY	27
		TSURVEY	28
		TSURVEY	29
		TSURVEY	30

	FUNCTION THINS(TIME)	THINS	2
	C** THIS FUNCTION CONVERTS SURVEY TIME IN FORMAT "SS.DDMMHCC"	THINS	3
	C** INTO CONTINUOUS RUNNING TIME IN MINUTES	THINS	4
5	C** SS = SURVEY NUMBER	THINS	5
	C** DD = SURVEY DAY	THINS	6
	C** HH = HOURS G.M.T.	THINS	7
	C** MM = MINUTES G.M.T.	THINS	8
	C** CC = SECONDS G.M.T.	THINS	9
10		THINS	10
	ITIME = TIME*1.0E8 + 0.5	THINS	11
		THINS	12
	ISURVEY = ITIME/100000000	THINS	13
	ITEMP1 = ITIME/1000000	THINS	14
15	ITEMP2 = ITIME/10000	THINS	15
	ITEMP3 = ITIME/100	THINS	16
		THINS	17
	IDAY = ITEMPL - ISURVEY*100	THINS	18
	IMOU = ITEMPL - ITEMPL*100	THINS	19
20	IMIN = ITEMPL - ITEMPL*100	THINS	20
	ISEC = ITIME - ITEMPL*100	THINS	21
		THINS	22
	ISEC = ISURVEY*60*60*60+IDAY*60*60+IMOU*3600+IMIN*60+ISEC	THINS	23
25	THINS = ISEC/60.8	THINS	24
		THINS	25
	RETURN	THINS	26
	END	THINS	27
		THINS	28
		THINS	29
		THINS	30



SATELLITE OFFSET FREQUENCY PLOT



SATELLITE OFFSET FREQUENCY PLOTS