SQT

NASA CONTRACTOR REPORT

NASA CR-141432

SEAHT

A COMPUTER PROGRAM FOR THE USE OF INTERSECTING ARCS OF ALTIMETER DATA FOR SEA SURFACE HEIGHT REFINEMENT

C P Allen
C F Martin

Prepared Under Contract No NAS6-2639 by

EG&G/Washington Analytical Services Center, Inc.
Wolf Research & Development Group
Clarke and Vine Streets
Pocomoke City, Maryland 21851

NASA
National Aeronautics and Space Administration

Wallops Flight Center
Wallops Island, Virginia 23337
AC 804 824-3411

November 1977

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA CR-141432		

4 Title and Subtitle

SEAHT - A COMPUTER PROGRAM FOR THE USE OF INTERSECTING ARCS OF ALTIMETER DATA FOR SEA SURFACE HEIGHT REFINEMENT

5 Report Date
November 1977

6 Performing Organization Code

7 Author(s)

C. P. Allen and C. F. Martin

8 Performing Organization Report No

9 Performing Organization Name and Address

EG&G/Washington Analytical Services Center, Inc.
Wolf Research & Development Group
Clarke and Vine Streets
Pocomoke City, Maryland 21851

10 Work Unit No

11 Contract or Grant No
NAS6-2639

12 Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Wallops Flight Center
Wallops Island, Virginia 23337

13 Type of Report and Period Covered
Contractor Report

14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

The SEAHT program is designed to process multiple passes of altimeter data with intersecting ground tracks, with the estimation of corrections for orbital errors to each pass such that the data has the best overall rss agreement at the crossover points. Orbit error for each pass is modeled as a polynomial in time, with optional orders of 0, 1, or 2. One or more passes may be constrained in the adjustment process, thus allowing passes with the best orbits to provide the overall level and orientation of the estimated sea surface heights. Intersections which disagree by more than an input edit level are not used in the error parameter estimation.

In the program implementation, passes are grouped into South-North passes and North-South passes, with the North-South passes partitioned out for the estimation of orbit error parameters. Computer core utilization is thus dependent on the number of parameters estimated for the set of South-North arcs, but is independent on the number of North-South passes.

Estimated corrections for each pass are applied to the data at its input data rate (normally 1/sec.) and an output tape is written which contains the corrected data.

17 Key Words (Suggested by Author(s))

GEOS-3
Altimeter
Geoid

18 Distribution Statement

Unclassified - unlimited

STAR Category - 61

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price*
Unclassified	Unclassified	116	

*For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

SECTION 1.0
INTRODUCTION

The primary objective in the development of the SEAHT program is the improvement of sea surface height values obtained from altimeter data along intersecting arcs (i.e., arcs whose groundtracks cross). The basic concept in the improvement scheme is that, at the crossovers, the altimeter measurements are to the same point on the surface and crossover differences are due to differences in:

- orbit error
- tides
- currents
- propagation effects
- altimeter noise and bias
- eddies and other temporal sea surface height effects

Of these effects, all except orbit error can be corrected or have amplitudes below the approximately 1 m level, and the effects of measurement noise and propagation effects are expected to be reduced to near the 10 cm level. The satellite orbital height error is expected to be the dominant error source in most cases. Over short time periods (less than a quarter of a revolution), the orbit error can be modeled as a polynomial in time, and a set of orbit error parameters can be estimated such that the set of intersections are as consistant as allowed by unmodeled (or improperly modeled) temporal sea surface height variations. In general, this orbit adjustment would be expected to be made relative to one or more passes whose orbits were known to be very good. This allows the estimated sea surface heights to be given an absolute interpretation rather than being only a set of relative heights.

Given a set of arcs, algorithms have been developed to categorize and find the intersections of altimeter passes. Algorithms are also developed for the estimation of the orbit error parameters using a Bayesian least squares technique.

SECTION 2.0
CROSSING ARC ADJUSTMENT ANALYSIS

2.1 FORMULATION OF CROSSOVER DIFFERENCES

In the study of sets of intersecting arcs, it is necessary to construct matrices relating to a pass at the time that it crosses the groundtrack for another pass. This will be accomplished through double subscripts, with the first denoting the pass of interest, and the second the pass which is being intersected. E.g., g_{ij} then refers to the i^{th} pass at the time of intersection of the groundtrack from the j^{th} pass. In all cases that will be considered, the arc lengths will be chosen sufficiently short so that the intersections are unique, and one arc does not cross another more than once.

Assuming that tide corrections have been made to altimeter sea surface height measurements, the result is the altimeter geoid measurement contaminated by, in general, orbit error. This geoid height measurement can be expressed as

$$g_{ij} = g^*_{ij} + \Delta g_{ij} + n_{ij} \tag{1}$$

where

g_{ij} is the geoid height measurement

g^*_{ij} is the actual geoid height

Δg_{ij} is the error in the geoid height due to satellite height error plus any biases in the altimeter measurement

n_{ij} is the error in the measurement due to noise in the altimeter measurement and to temporal sea surface height effects.

Although no analytical model is available for g_{ij}, it may be represented empirically by a polynomial in time,

$$\Delta g_{ij} = a_{i0} + a_{i1} t_{ij} + a_{i2} t_{ij}^{2} + \ldots + a_{iK_i} t_{ij}^{K_i} \qquad (2)$$

where

t_{ij} is the time along the i^{th} arc at point (i,j)

Considering now the geoid height measurement at the point (i,j) along the j^{th} arc, the model equation is

$$g_{ji} = g_{ji}^{*} + \Delta g_{ji} + n_{ji} \qquad (3)$$

But, by definition $g_{ij}^{*} = g_{ji}^{*}$. Therefore, using (1) and (3), we have

$$d_{ij} = g_{ij} - g_{ji} = \Delta g_{ji} - \Delta g_{ji} + n_{ij} - n_{ji}$$

where d_{ij} is the difference between the geoid height measurements at the (i,j) point. Substituting from Eqn. (2),

$$d_{ij} = \sum_{e=0}^{K_i} a_{ie}\, t_{ij}^{e} - \sum_{e=0}^{K_j} a_{je}\, t_{ji}^{e} + n_{ij} - n_{ji} \qquad (4)$$

Since Equation (4) is expressed in terms of geoid height differences, a constraint must be used to anchor the solution. For a one parameter solution, one constrained revolution is sufficient. All other revolutions are adjusted such that the differences in heights between each of these and the constrained revolution will be minimized. Since orbital errors have predominantly a once per revolution type of variation, a low order polynomial should serve as a satisfactory model for short arcs. In particular, a quadratic would be expected to be the maximum required order for arc lengths not exceeding a quarter of a revolution (~25 minutes for GEOS-3) and shorter arcs should require only a first (or zero) order polynomial.

The objective is then to solve for the set of coefficients which will minimize the d_{ij}'s in some sense, subject to the constraint that certain arcs are not to be adjusted. The constraints can be applied through several different techniques. We choose the Bayesian approach since it is consistent with available software and amounts to nothing more than supplying additional measurements to the estimation scheme. The least squares technique will be chosen for the coefficient estimation. This procedure will also be minimum variance if the noise difference, $n_{ij} - n_{ji}$, is uncorrelated from one crossing to another and its variance is used to weight the intersection data. The degree to which the errors are uncorrelated depends primarily upon the correlation between temporal sea surface height variations but, in any case, correlation information is not expected to be available. The least squares technique then requires the minimization of

$$S = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \delta_{ij} \left\{ \left(d_{ij} - \sum_{e=0}^{K-1} a_{ie} \, t_{ij}^{e} + \sum_{e=0}^{K-1} a_{je} \, t_{ji}^{e} \right)^2 w_{ij} \right\} \quad (5)$$

where

N is the total number of arcs

δ_{ij} = 1 if arc j intersects i
= 0 if not

w_{ij} = weighting factor based on $n_{ij} - n_{ji}$
$= 1/E\ [n_{ij} - n_{ji}]^2$

with E the expected value operator. Additionally, we have assumed that

$$K_i = K_j = K-1$$

so that the number of orbit error parameters is taken to be the same for all arcs. This is not an essential assumption, but substantially simplifies the software implementation. Constraints can be used to effectively vary the number of adjusted parameters from one arc to the next.

2.2 IMPLEMENTATION CONSIDERATIONS

Before discussing the solution of (5) it is pertinent to consider the number of intersections required to generate a solution of a given order. First, note that the total number of intersections is

$$N_I = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \delta_{ij}, \qquad (6)$$

whereas the number of coefficients to be determined is

$$N_c = \sum_{i=1}^{N} K = NK \tag{7}$$

Obviously, for this method to be practical we must require that

$$N_I \geq N_c$$

or

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} \delta_{ij} \geq NK \tag{8}$$

In addition, we cannot solve for K coefficients along the i^{th} arc if there are fewer than K arcs intersecting arc i. Therefore, we also require

$$\sum_{j=1}^{N} \delta_{ij} \geq K \text{ for all i.} \tag{9}$$

In order to assess the number of arcs which it may be feasible to process in a computer, consider the case of two sets of arcs with each arc in the first set intersecting each arc in the second set. We will denote the number of arcs in the first set by N_a and the number of arcs in the second set by N_b. For a solution, we require that the number of intersections be greater than or equal to the number of coefficients, or

$$N_a N_b \geq K(N_a + N_b),$$

from which

$$N_a \geq \frac{K\ N_b}{N_b - K} \qquad (10)$$

Minimum values of N_a for various values of K and N_b are listed in Table 1.

K	N_b	Minimum N_a
1	2	2
1	3	2
1	5	2
2	3	6
2	4	4
2	5	4
2	10	3
2	20	3
3	4	12
3	5	8
3	10	5
3	20	4
3	50	4

Table 1. Minimum Numbers of Crossing Arcs for Different Numbers of Adjusted Parameters

Note that the limiting value of N_a for large values of N_b is K+1. This can be deduced from Equation (10), since

$$\lim_{N_b \to \infty} \frac{K\ N_b}{N_b - K} = K \qquad (11)$$

For all finite values of N_b, the ratio is greater than K so N_a must be at least K+1.

Looking now at the approximate core requirements, we consider the normal matrix to be the array which primarily limits the number of arcs which can be simultaneously processed. Allowing for this array to be double precision, but with only the diagonal and above elements stored, the number of words required for the M dimensional normal matrix is

$$N_T = \frac{M(M+1)}{2} (2) = M(M+1)$$

Substituting $M = K(N_a + N_b)$ for the number of parameters, the core required is

$$N_T = K^2(N_a + N_b)^2 + K(N_a + N_b) \tag{12}$$

Table 2 shows the core required for the normal matrix for two different relations of N_a to N_b. In one case, we set $N_a = N_b$, and in the other case we set $N_b = 1$ as is appropriate for the program implementation in which the arcs in one direction are partitioned, as will be discussed in Section 2.3.

Considering the maximum core available on the Wallops computers to be on the order of 70K words, it is evident from Table 2 that the limitation on N_b for the unpartitioned solution is approximately 100 passes for a one parameter solution and approximately 50 passes for a two parameter solution. For a partitioned solution, these limits are almost doubled, allowing approximately 200 one parameter arcs or 100 two parameter arcs.

K	N_a	N_T	
		$N_b \doteq N_a$	$N_b = 1$
1	2	20	12
	5	110	42
	10	420	132
	20	1640	462
	50	10100	2652
	100	40200	10302
	150	90300	22952
	200	160400	40602
2	4	272	110
	10	1640	506
	20	6480	1806
	50	40200	10506
	100	160400	41006
3	6	1332	462
	10	3660	1122
	20	14520	4032
	50	90300	23562
	100	360600	92112

Table 2. Normal Matrix Storage Requirements
for Different Numbers of Arcs and Parameters Per Arc

2.3 LEAST SQUARES ESTIMATION OF ORBIT IMPROVEMENT PARAMETERS

The least squares estimate of the parameters which minimize the altimeter crossovers is obtained by finding the set of a's which minimize the sum of squares of crossover differences given by Equation (5). Equivalently, we can consider the least squares solution of the set of linear equations given by Equation (4), which may be written symbolically in matrix form as

$$\bar{Y} = B\bar{Z} + \bar{\varepsilon} \tag{13}$$

where

$\bar{Y}$ is the vector of geoid height differences.

B is the matrix of partial derivatives of the geoid height differences with respect to the orbit error parameters. This matrix will consist of powers of time from some reference time for each pass.

$\bar{Z}$ is the vector of orbit error coefficients to be solved for.

$\bar{\varepsilon}$ is the vector of errors in the crossover differences due to measurement noise.

Table 3 shows Equation (13) written out explicitly for a two coefficient parameterization of orbit error and 4 North-South passes crossing 4 South-North passes.* For convenience, the

*The solution of the set of equations given in Table 3 is actually singular. Intuitively, it is obvious that whatever solution is obtained for the a_{i0}'s, the same constant added to each of these coefficients is an equally valid solution. The solution can be made non-singular through the use of a constraint on one or more of the passes. We will assume that such information is to be used, but will not explicitly include it in the equations shown here.

Table 3. Matrix Form of Linear Equations to be Solved for
4 Passes Crossing 4 Passes in the Opposite Direction and
2 Orbit Parameters Per Pass

$$
\begin{bmatrix} d_{15} \\ d_{16} \\ d_{17} \\ d_{18} \\ d_{25} \\ d_{26} \\ d_{27} \\ d_{28} \\ d_{35} \\ d_{36} \\ d_{37} \\ d_{38} \\ d_{45} \\ d_{46} \\ d_{47} \\ d_{48} \end{bmatrix}
=
\begin{bmatrix}
1 & t_{15} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{51} & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & t_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{61} & 0 & 0 & 0 & 0 \\
1 & t_{17} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{71} & 0 & 0 \\
1 & t_{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{81} \\
0 & 0 & 1 & t_{25} & 0 & 0 & 0 & 0 & -1 & -t_{52} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & t_{26} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{62} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & t_{27} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{72} & 0 & 0 \\
0 & 0 & 1 & t_{28} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{82} \\
0 & 0 & 0 & 0 & 1 & t_{35} & 0 & 0 & -1 & -t_{53} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & t_{36} & 0 & 0 & 0 & 0 & -1 & -t_{63} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & t_{37} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{73} & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & t_{38} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{83} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{45} & -1 & -t_{54} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{46} & 0 & 0 & -1 & -t_{64} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{47} & 0 & 0 & 0 & 0 & -1 & -t_{74} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & t_{48} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{84}
\end{bmatrix}
\begin{bmatrix} a_{10} \\ a_{11} \\ a_{20} \\ a_{21} \\ a_{30} \\ a_{31} \\ a_{40} \\ a_{41} \\ a_{50} \\ a_{51} \\ a_{60} \\ a_{61} \\ a_{70} \\ a_{71} \\ a_{80} \\ a_{81} \end{bmatrix}
+
\begin{bmatrix} n_{15}-n_{51} \\ n_{16}-n_{61} \\ n_{17}-n_{71} \\ n_{18}-n_{81} \\ n_{25}-n_{52} \\ n_{26}-n_{62} \\ n_{27}-n_{72} \\ n_{28}-n_{82} \\ n_{35}-n_{53} \\ n_{36}-n_{63} \\ n_{37}-n_{73} \\ n_{38}-n_{83} \\ n_{45}-n_{54} \\ n_{46}-n_{64} \\ n_{47}-n_{74} \\ n_{48}-n_{84} \end{bmatrix}
$$

North-South arcs have been denoted as arcs 1-4 and the South-North passes as arcs 5-8.

The weighted least squares solution of Equation (13) is [1]

$$\hat{Z} = (B^T W B)^{-1} B^T W \bar{Y} \tag{14}$$

where

$\hat{Z}$ is the estimated set of coefficients

$W^{-1} = E[\bar{\varepsilon}\ \varepsilon^T]$ = covariance of the measurement noise, with E the expectation operator.

That the solution allows partitioning can be seen by rewriting the equations given in Table 3 in the form:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} B_1 & 0 & 0 & 0 & B_{10} \\ 0 & B_1 & 0 & 0 & B_{20} \\ 0 & 0 & B_3 & 0 & B_{30} \\ 0 & 0 & 0 & B_4 & B_{40} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_0 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \tag{15}$$

where

$$Y_i = \begin{bmatrix} d_{i5} \\ d_{i6} \\ d_{i7} \\ d_{i8} \end{bmatrix}$$

$$B_i = \begin{bmatrix} 1 & t_{i5} \\ 1 & t_{i6} \\ 1 & t_{i7} \\ 1 & t_{i8} \end{bmatrix}$$

$$B_{i0} = \begin{bmatrix} -1 & -t_{5i} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -t_{6i} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -t_{7i} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -t_{8i} \end{bmatrix}$$

$$a_i = \begin{bmatrix} a_{i0} \\ a_{i1} \end{bmatrix}$$

$$a_0 = \begin{bmatrix} a_{50} \\ a_{51} \\ a_{60} \\ a_{61} \\ a_{70} \\ a_{71} \\ a_{80} \\ a_{81} \end{bmatrix}$$

$$n_i = \begin{bmatrix} n_{i5} - n_{5i} \\ n_{i6} - n_{6i} \\ n_{i7} - n_{7i} \\ n_{i8} - n_{8i} \end{bmatrix}$$

Assuming only that the n_i's are uncorrelated, and denoting

$$W_i^{-1} = E[n_i \; n_i^T], \tag{16}$$

Equation (14) for the 8 arc 16 parameter case of Table 3 becomes

$$\begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \\ \hat{a}_0 \end{bmatrix} = \begin{bmatrix} B_1^T W_1 B_1 & 0 & 0 & 0 & B_1^T W_1 B_{10} \\ 0 & B_2^T W_2 B_2 & 0 & 0 & B_2^T W_2 B_{20} \\ 0 & 0 & B_3^T W_3 B_3 & 0 & B_3^T W_3 B_{30} \\ 0 & 0 & 0 & B_4^T W_4 B_4 & B_4^T W_4 B_{40} \\ (B_1^T W_1 B_{10})^T & (B_2^T W_2 B_{20})^T & (B_3^T W_3 B_{30})^T & (B_4^T W_4 B_{40})^T & \sum_{i=1}^{4} B_{i0}^T W_i B_{i0} \end{bmatrix}^{-1}$$

$$X \begin{bmatrix} B_1^T W_1 Y_1 \\ B_2^T W_2 Y_2 \\ B_3^T W_3 Y_3 \\ B_4^T W_4 Y_4 \\ \sum_{i=1}^{4} B_{i0}^T W_i Y_i \end{bmatrix} \tag{17}$$

The matrix inverse in Equation (17) can be obtained via partitioning [1] with the resulting expressions for the coefficient sets found to be

$$\hat{a}_0 = M_0 \sum_{i=1}^{4} [B_{i0}^T W_i Y_i - B_{i0}^T W_i B_i (B_i^T W_i B_i)^{-1} \; B_i^T W_i Y_i] \tag{18}$$

$$\hat{a}_i = (B_i^T W_i B_i)^{-1} B_i^T W_i Y_i - (B_i^T W_i B_i)^{-1} B_i^T W_i B_{i0} \hat{a}_0 \qquad (19)$$

where

$$M_0^{-1} = \sum_{i=1}^{4} [B_{i0}^T W_i B_{i0} - B_{i0}^T W_i B_i (B_i^T W_i B_i)^{-1} (B_{i0}^T W_i B_i)^T] \qquad (20)$$

The extension of the solution given by Equations (18-20) to any number of arcs is achieved through two directions. Let us refer to the arcs for which the a_i parameters have been assigned as North-South arcs, and the set of arcs for which the a_0 set of parameters have been assigned as South-North arcs. To increase the number of North-South arcs, we simply increase the summation limit in Equations (18) and (10). To increase the number of South-North arcs, we must increase the number of a_0 parameters and thus the size of M_0 which is, in general, the matrix providing the major limitation on the number of passes which can be processed in one computer run.

The estimation scheme implemented in SEAHT is the partitioned least squares type solution described above, with the number of adjustable parameters per pass selectable as 1, 2, or 3. North-South arcs are partitioned out, so virtually any number of North-South passes can be processed simultaneously. South-North arcs provide the "common" parameter set, and the number of such passes is limited by available computer core.

SECTION 3.0
PROGRAM IMPLEMENTATION

3.1 PROGRAM ORGANIZATION

The SEAHT program utilizes overlays, and consists primarily of a main program (plus common block initialization), an input set of routines, and an estimation set of routines. The last set of routines, however, is divided into two groups, and there is an additional segment used for core allocation. Thus, there are a total of four program "LINKS" plus a main program segment.

The main program is responsible for the flow of the program, and for the computation of necessary array sizes. In LINK 1, optional input is read from cards and input data is read from tape(s), and the arc crossings are computed. The crossing data is stored on disk in the form to be used by the estimation routines. The order of the crossings is: all crossings of the first North-South arc, followed by all crossings of the second North-South arc, etc. This format allows the partitioning scheme discussed above to be implemented. LINK 1 contains routines with fixed array sizes, with dimensions set to accommodate the maximum amounts of data per arc and maximum number of arcs that are expected. However, these dimensions do not result in excessive core utilization.

At the conclusion of the computation of the arc crossings, the needed sizes for all arrays in the estimation process are calculated by the main program. LINK 1 is then overlaid by LINK 2 which contains a portion of the estimation routines. LINK 3 is then called to perform dynamic memory allocation for all arrays needed in the parameter estimation process. After this allocation, LINK 3 is overlaid by LINK 4 containing the remainder of the estimation routines. No further overlays are performed.

Control of the estimation process is from subroutine SOLVE, which is responsible for controlling the normal matrix set-up, the application of constraints, the parameter adjustments, and the application of the estimated parameters to the input data, with the corrected data then saved on tape.

3.2 PROGRAM INPUT

Input to the SEAHT program consists of one or more data tapes, and card input which requests certain program options. The latter category consists of a set of option cards, at least one of which is required.

3.2.1 Data Tape Input

Data input files to SEAHT consist of a header record followed by a set of data records, as specified on the following two pages. The header record identifies the time of the data by both the revolution (and segment) number and date. At the present time, the date is not used, while the revolution and segments numbers are used for the selection and deletion of particular data passes, as discussed below.

Input tapes may be prepared in several different ways. They may be prepared by extracting information contained on each frame of an ARC [2] tape, in which case sea surface heights are major frame ($\sim$2.0 or 3.2 seconds) averages. In addition, the ALTKAL program [3] produces a tape which is directly readable by SEAHT. This tape contains smoothed sea surface heights at the 1/second data rate.

In the current program implementation, only Words 1-5 of the header record and Words 1-4 of the data record are read by SEAHT, and the first 3 words of the header record are not used. Words 4-5 of the header record and the first 4 words of each data record are critical to program operation.

SEAHT Input Format
Header Record

Word No.	Type	Description
1	I	Year and Day of year (YYDDD)
2-3	DP	Time of day (seconds) for first data record
4	I	Segment number for altimeter data pass
5	I	Rev number for altimeter data pass
6	I	Measurement type 40 = Global mode 41 = Intensive mode
7-19	I	Zeroes (used for making all records same length)

SEAHT Input Format

Data Record

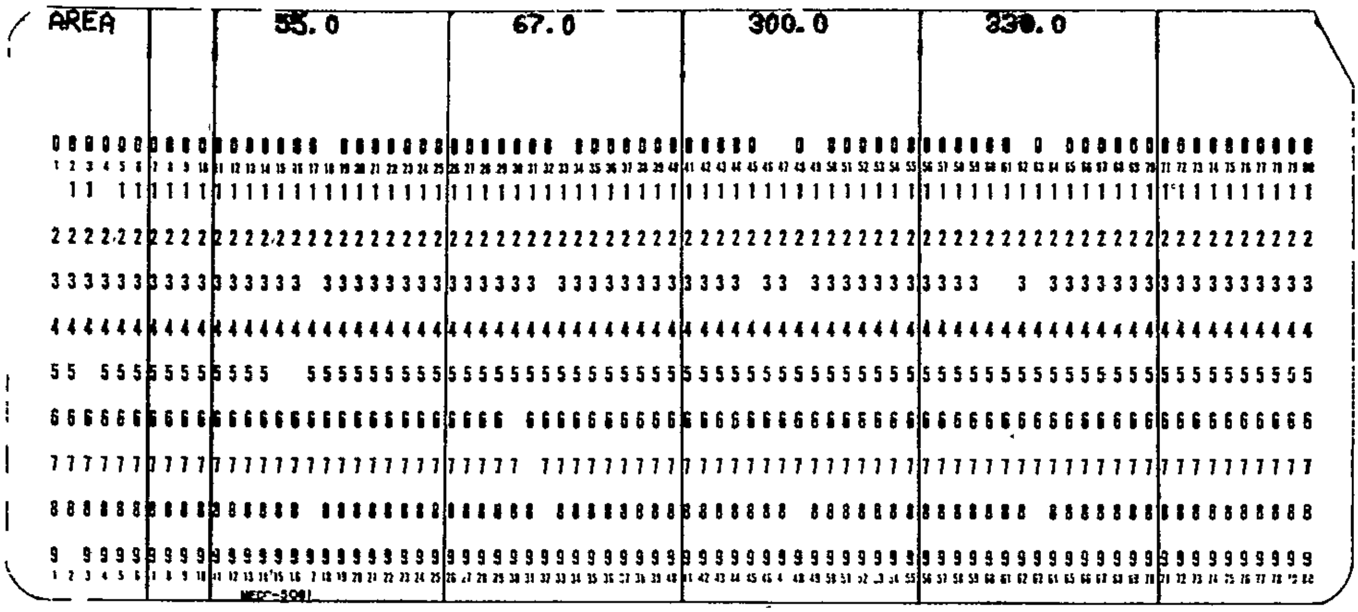
Word No.	Type	Description
1	R	Time from first data point (minutes)
2	R(LAT)	Latitude of sea surface height
3	R(LON)	Longitude of sea surface height
4	R	Sea surface height (meters)
5-13	R	Not used. (For tapes created by the ALTKAL smoother, words 4-13 contain sea surface heights at the 10/sec data rate.)
14	R(THITE)	Tide height (meters)
15	R(TREF)	Tropospheric refraction correction (meters)
16	R(RAGCAV)	Average AGC
17-19	R	Zeroes for future additions

Note: The "sea surface height" value(s) have been corrected using the tropospheric refraction correction and tide height on the data record. Strictly speaking, the sea surface height is the altimeter measurement of geoid height, although including the effects of currents and other non-stationary sea surface height effects.

3.2.2 Option Card Input

Through the use of one or more of nine different keyword option cards, various aspects of the operation of SEAHT can be altered from the default mode. These option cards and the purpose for which they are utilized are

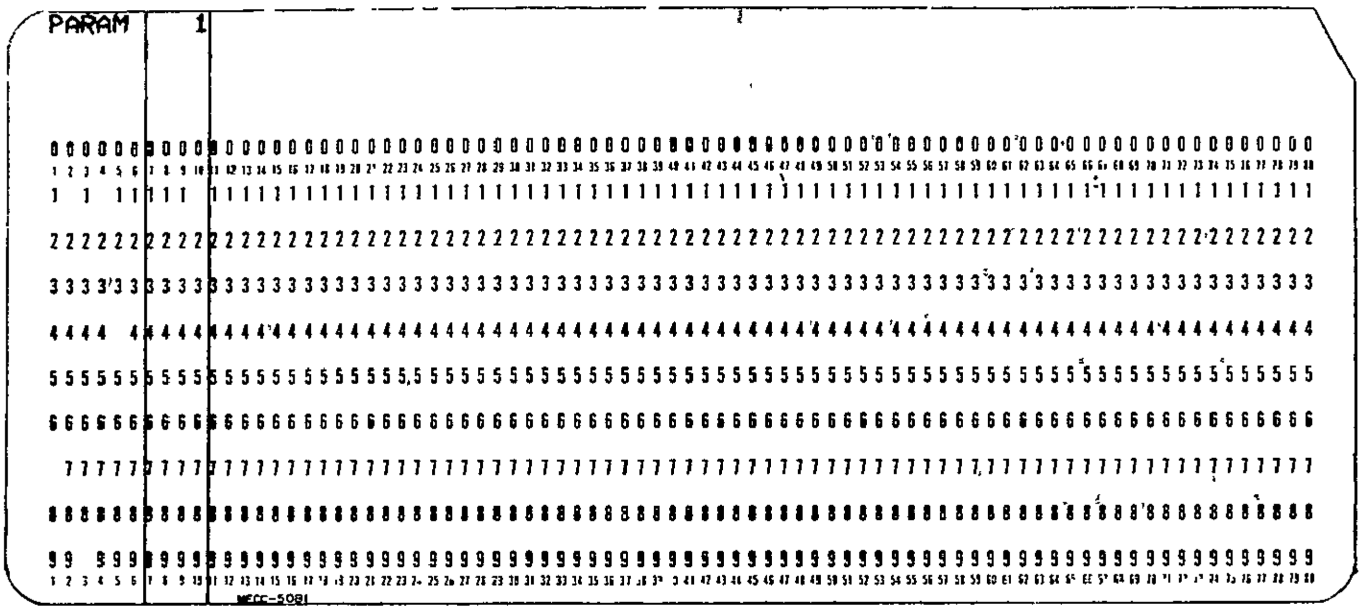
- AREA - defines the geographic area for which arc crossings are to be selected.
- PARAM - selects the number of coefficients to be used for each pass to parametrize orbit error.
- NODATA - specifies that arc crossings have been saved from a previous program execution and are to be input in place of the normal data tapes.
- CONSTR - specifies that a particular pass is to be treated as having a good orbit with the parameters constrained in the adjustment process.
- EDIT - specifies the level at which arc crossing differences are to be edited from the parameter estimation.
- SELECT - specifies that only particular passes are to be selected from a particular data tape.
- DELETE - specifies that all passes are to be selected from a particular tape except for those which are designated.



NAME	COLUMNS	FORMAT	DESCRIPTION
AREA	1-4	A4	Defines the area of interest
	11-25	E15.5	Minimum acceptable value of latitude
	26-40	E15.5	Maximum acceptable value of latitude
	41-55	E15.5	Minimum acceptable value of longitude
	56-70	E15.5	Maximum acceptable value of longitude

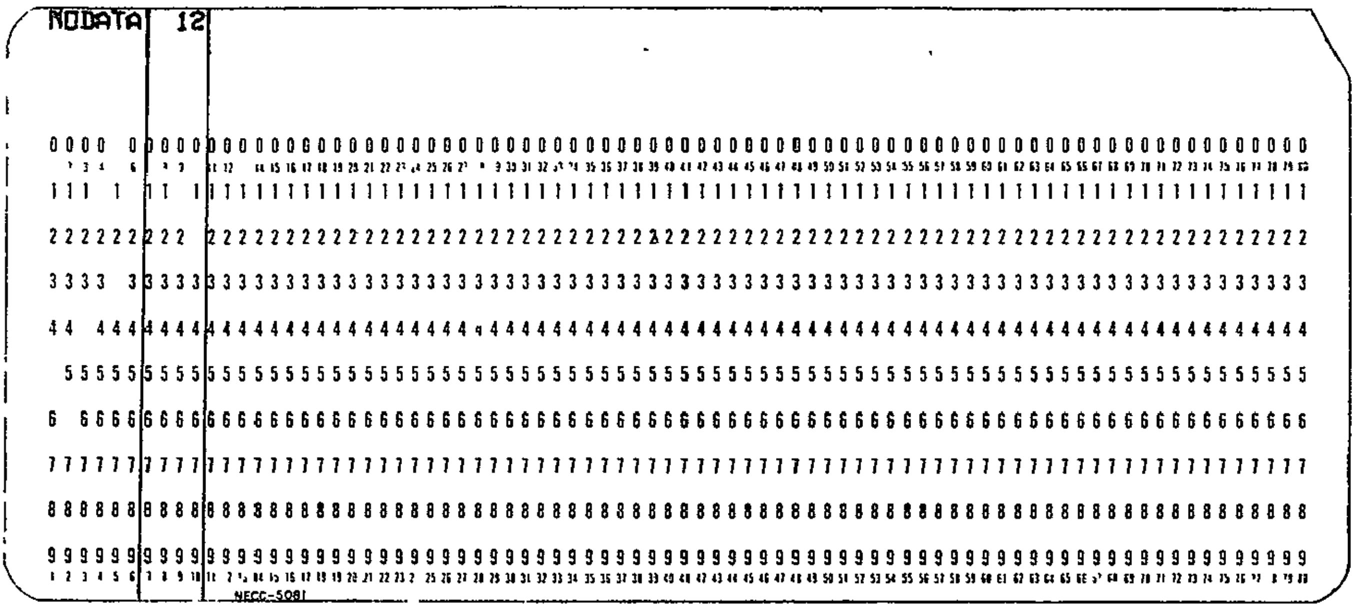
IF CARD OMMITTED: Default values are: minimum latitude, -90.0 degrees; maximum latitude, +90.0 degrees; minimum longitude, 0.0 degrees; maximum longitude, 360.0 degrees.

PARAM

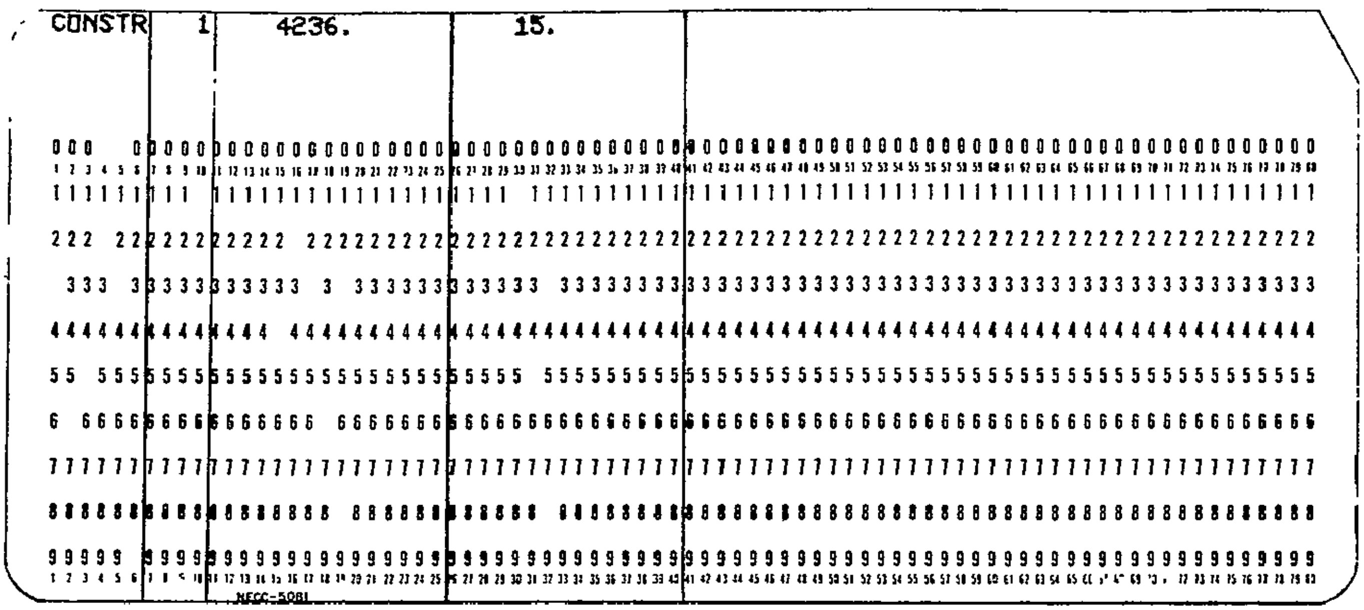


NAME	COLUMNS	FORMAT	DESCRIPTION
PARAM	1-5	A5	Indicates the number of parameters which are to be used for each pass.
	7-10	I4	The number of adjusted parameters for each pass.

NODATA



NAME	COLUMNS	FORMAT	DESCRIPTION
NODATA	1-6	A6	Indicates crossing data already exists
	7-10	I4	The logical unit number of the disk or tape where the crossing data is stored

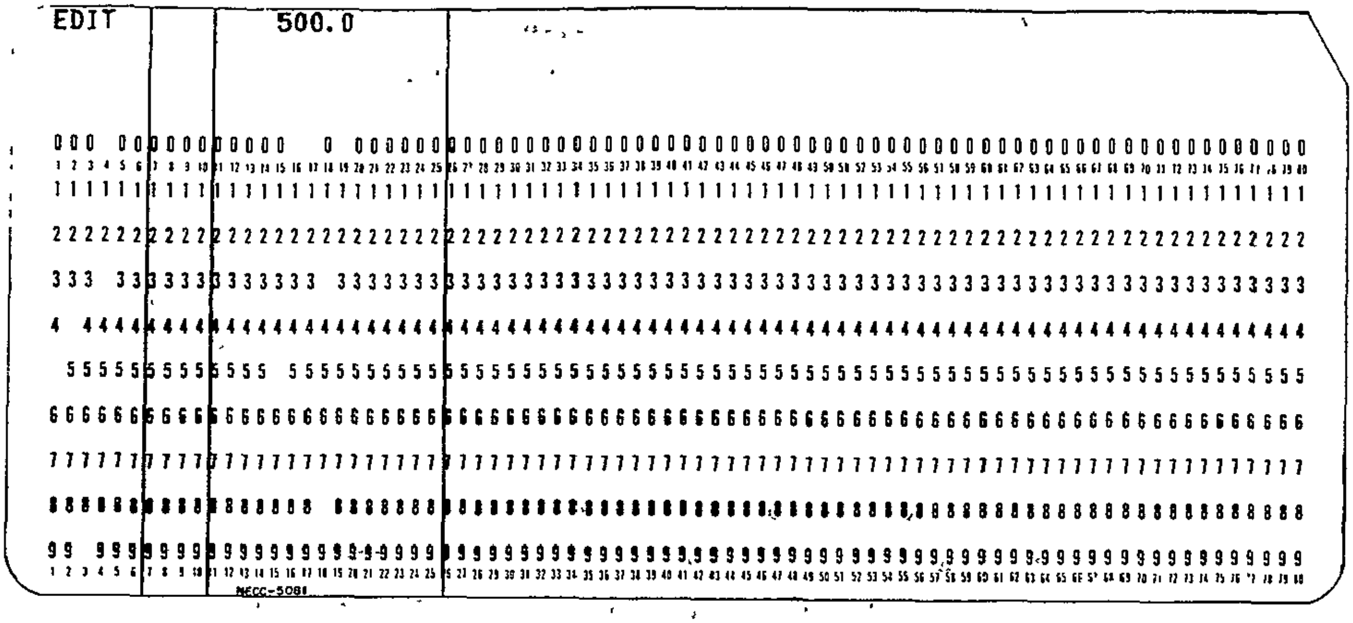


NAME	COLUMNS	FORMAT	DESCRIPTION
CONSTR	1-6	A6	Constrains selected revs (other revs are adjusted to fit constrained rev and constrained revs are not adjusted)
	7-10	I4	Type of constraint = 1, pass totally constrained (bias, slope and curvature) = 2, bias adjusted, slope and curvature constrained = 3, bias and slope adjusted, curvature constrained
	11-25	E15.6	Number of a revolution to be constrained

NAME	COLUMNS	FORMAT	DESCRIPTION
CONSTR (Cont.)	26-40	E15.6	Segment number associated with the rev number in columns 11-55

Note: The use of a constraint is meaningless unless the parameter has been otherwise requested to be adjusted. E.g., the constraint of curvature for a pass is meaningless if only bias and slope are being adjusted, and should therefore not be requested.

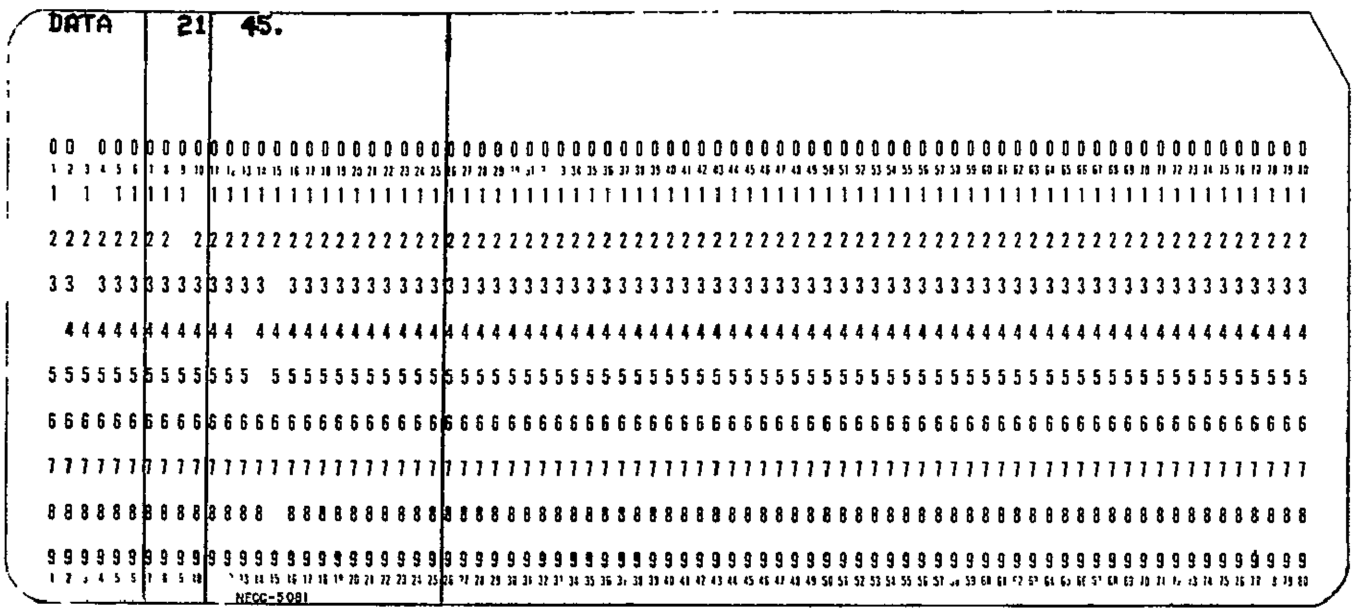
EDIT



NAME	COLUMNS	FORMAT	DESCRIPTION
EDIT	1-4	A4	Sets editing criteria.
	11-25	E15.6	Maximum acceptable difference in sea surface heights at any crossing point (meters).

IF CARD OMITTED: Default value is 100.0 meters.

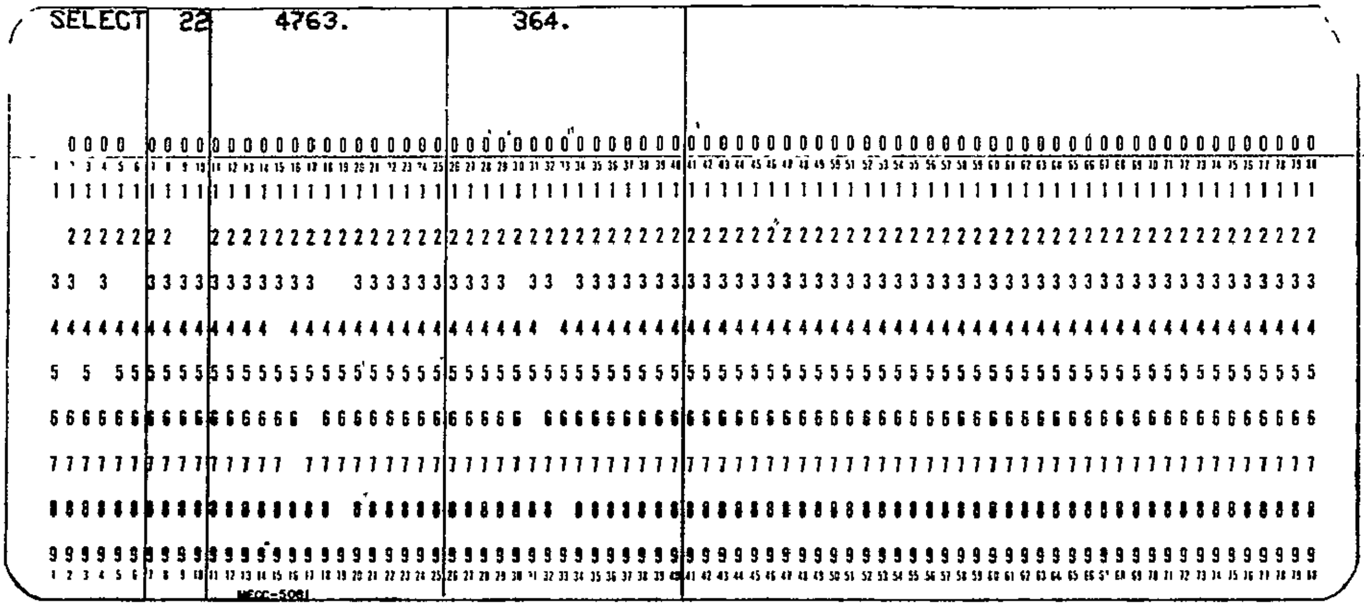
DATA



NAME	COLUMNS	FORMAT	DESCRIPTION
DATA	1-4	A4	Indicates the source of the data to be read in
	7-10	I4	Logical unit from which data is to be read
	11-25	E15.6	Number of files to be read from the logical unit designated in columns 7-10

As many as six of these cards are possible barring limitations set outside the program.

SELECT



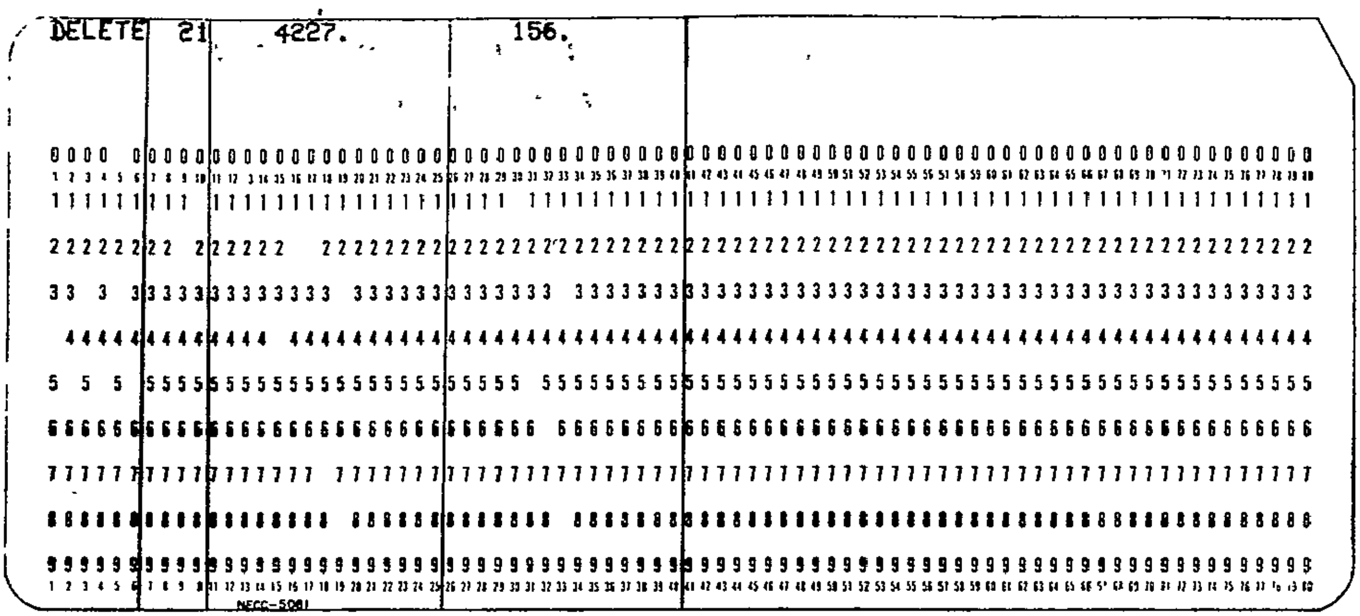
NAME	COLUMNS	FORMAT	DESCRIPTION
SELECT	1-6	A6	Selects revolutions from a specific tape
	7-10	I4	Logical unit number of the tape from which selection is to be made
	11-25	E15.6	Revolution to be selected
	26-40	E15.6	Segment number associated with revolution designated in Columns 11-25
	41-55	E15.6	Revolution to be selected*

* If not more than one revolution is selected, columns 41-80 should be blank.

NAME	COLUMN	FORMAT	DESCRIPTION
	56-70	E15.6	Segment number associated with the revolution designated in columns 41-55.

LIMITATION: Multiple SELECT cards are permitted, but no more than 50 selections may be made from the same logical unit number.

DELETE



NAME	COLUMNS	FORMAT	DESCRIPTION
DELETE	1-6	A6	Deletes revolutions not desired from a specific tape
	7-10	I4	Logical unit number of the tape from which deletion is to be made
	11-25	E15.6	Revolution to be deleted
	26-40	E15.6	Segment number associated with the revolution in columns
	41-55	E15.6	Revolution to be deleted*

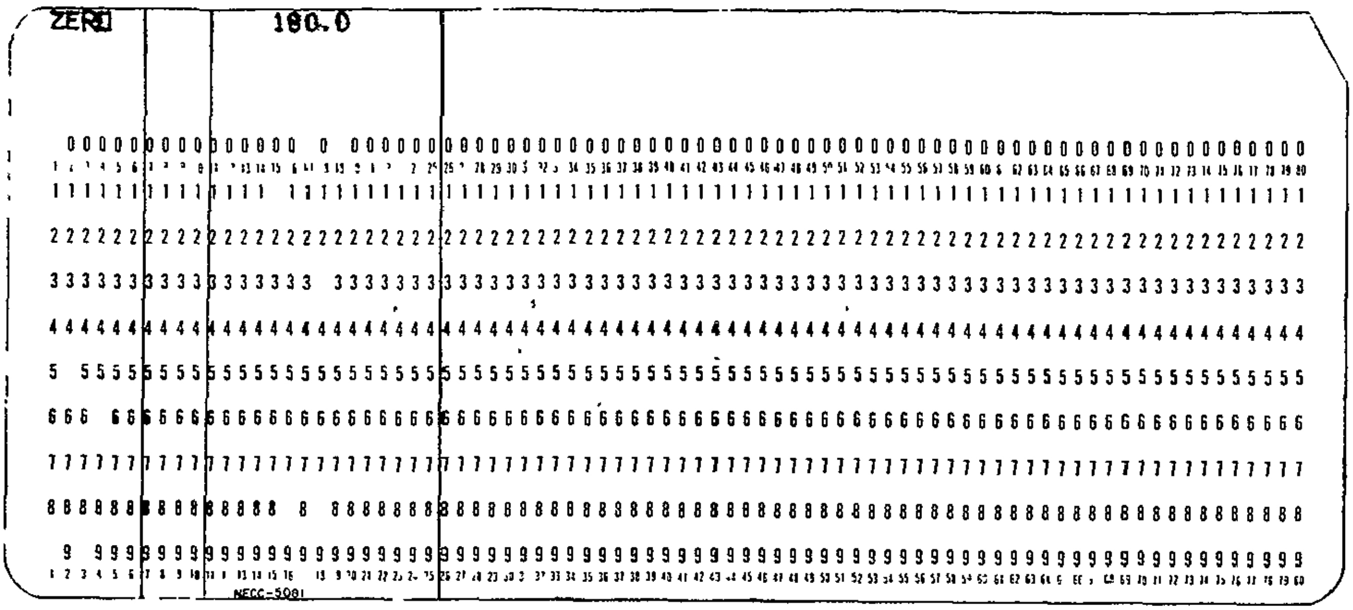
* If not more than one revolution is deleted columns 41-80 should be blank.

DELETE

NAME	COLUMN	FORMAT	DESCRIPTION
	56-70	E15.6	Segment number associated with the revolution designated in columns 41-55

LIMITATION: Multiple DELETE cards are permitted, but no more than 50 deletions may be made from the same logical unit number.

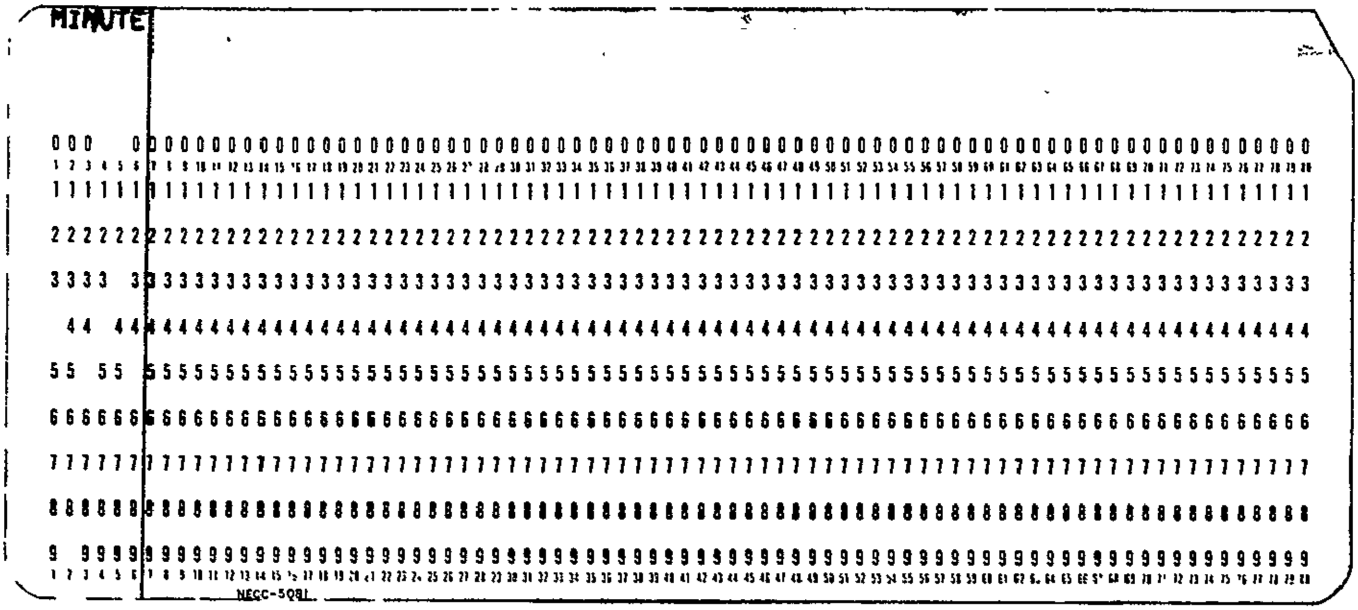
ZERO



NAME	COLUMNS	FORMAT	DESCRIPTION
ZERO	1-4	A4	Sets a longitude different than Greenwich as the longitude origin.
	11-25	E15.6	Longitude, relative to Greenwich, to be used as origin.

PURPOSE: Since the method of finding intersections of revolutions is dependent upon a decreasing longitude in the area of interest, it may be necessary to, temporarily, zero-set a certain longitude in the area of least interest. This is done within the program and requires no additional computation by the user.

MINUTE



NAME	COLUMNS	FORMAT	DESCRIPTION
MINUTE	1-6	A6	Permits the time to be input in minutes

IF CARD OMITTED: The units of time are assumed to be in days and are changed to minutes within the program.

3.3 DATA SELECTION AND PARAMETER ESTIMATION PROCESS

The estimation process begins with the acceptance of data that was collected in the area of interest. The revolutions are classified as being north-south or south-north. A revolution whose latitude increases to the maximum possible value (approximately 65.°) and, then, decreases is separated into a south-north segment and a north-south segment. This also applies to revolutions whose latitude decreases to the minimum value (approximately -65.°) and, then, increases.

Preliminary to finding crossing points, tests are made based on the fact that there will not be an intersection if the exiting latitude of a south-north revolution is less than the exiting latitude of the north-south revolution or if the entering latitude of the north-south revolution is less than the entering latitude of the south-north revolution. If the exiting longitude of the north-south revolution is greater than the entering longitude of the south-north revolution or the exiting longitude of the south-north revolution is greater than the entering longitude of the north-south revolution, there will not be a crossing.

Every south-north revolution is compared to every north-south revolution to determine crossing points by the following method:

1. Equations for lines connecting the entering and exiting points of the revolutions are determined, and the intersections of these lines computed.

2. The latitudes, corresponding to the longitude at the intersection, are found for each revolution, and the difference between these values is computed. If the absolute value of this difference is less than .0001 degrees, the average of the latitudes is taken, and a crossing point is declared at this average latitude and the longitude corresponding to the intersection point of the lines connecting entering and exiting points of the revolutions.

3. If the difference is too great and the latitude is larger for the north-south revolution, these points become the entering points of the revolutions. Otherwise, they become the exiting points. This process continues until a crossing point is found or the number of iterations exceeds ten.

The times into the revolutions and the sea surface heights at these crossing points are found by interpolation.

A process to determine whether each revolution has crossed other revolutions a sufficient number of times is performed. The process repeatedly deletes revolutions and checks for sufficient crossings until it is not necessary to delete any additional revolutions.

Examining one crossing point at a time, the difference in sea surface heights (sea surface height of the north-south revolution minus the sea surface height of the south-north revolution) at the crossing point is computed as well as the powers of time along the revolutions at this crossing point. The elements of the normal matrix B^TWB are computed and placed in their appropriate position as are the elements of the B^TWY matrix. The noise on each revolution is assumed to be the same.

Next, the north-south revolutions which are to be constrained are noted and elements of the B^TWB matrix are altered to ensure that the coefficients of the bias for this revolution will be small and that adjustments on other revolutions will, necessarily, be made relative to this revolution.

Handling all crossing point data for one north-south revolution at a time, inversion of the normal matrix is performed and biases are computed for the north-south revolutions based on the information from each particular north-south revolution under consideration with the south-north passes held fixed. After all the north-south passes have been processed, the south-north constraints (if any) are applied and the normal matrix for the south-north passes is calculated (Eqn. 20). The south-north biases are then computed (Eqn. 18) and the north-south biases updated (Eqn. 19).

All data are corrected for the biases found.

3.4 PROGRAM OPERATION

A source deck and binary decks of SEAHT are located at the Wolf Research and Development Group office in Pocomoke City, Maryland. The Honeywell 625/635 run setup consists of the appropriate control cards, the program on binary cards, the desired data cards, and a source of data input. The standard setup for a run consists of the following:

- Specification of the FORTRAN FILE units for both the disk files used by the program and the tapes used for data input.
- Program binary deck.
- The set of option cards required for the particular run.

A sample job setup is given in Section 3.4.2.

3.4.1 FORTRAN Logical Units

In addition to the normal card input and printer output files, SEAHT uses 5 scratch files, writes one output tape, writes 2 additional printer files, and normally uses 1-6 input tapes. All scratch files are sequentially accessed and logically could also be assigned to tapes. However, some of the files - particularly unit 12 - are rewound a number of times and could lead to inefficient program operation if assigned to tape.

The FORTRAN logical units used are as follows:

LOGICAL UNIT	USE
5	Option card input
6	Standard printer output
10	Output tape of corrected data
11	Disk storage during execution
12	Disk storage during execution
13	Disk storage during execution
14	Disk storage during execution
15	Disk storage during execution
18	Printed output
19	Printed output
*	Input tapes (normally multi-file)

*This set must be comprised of unique numbers different from the other units listed above. Multi-file tapes must be non-standard label. Single file tapes may be either standard label or non-standard label. A total of up to 6 input tapes may be used in a job.

3.4.2 Sample Deck Setup

The following is an example of a working deck setup. This particular setup would require that nineteen files be read from the tape with logical unit number 21 (in this case, tape #4572). A height bias would be found for all passes within the area: latitude; 45°-60°, longitude; 285°-300°, which are found to cross another pass. Rev 4476, segment 169 would not be adjusted and the biases determined for the other passes would be relative to this constrained pass.

COL.

```
1      8         16
$      IDENT     112293,SEAHT
$      OPTION    FORTRAN          )
        .                         |
        .                         |
        .                         |
        .                         |
$      LINK      LINK1            |
        .                         |
        .                         |
        .                         |
        .                         |        PROGRAM
$      LINK      LINK2,LINK1      }        BINARY
        .                         |        DECK
        .                         |
        .                         |
        .                         |
        .                         |
$      LINK      LINK3            |
        .                         |
        .                         |
        .                         |
        .                         |
$      LINK      LINK4,LINK3      )
        .
        .
        .
        .
$      EXECUTE   DUMP
$      LIMITS    20,47K,10K
$      TAPE9     10,X6D,,,,WJTAPE
$      FFILE     10,NSTDLB,NOSRLS,FIXLNG/5,BUFSIZ/5,MLTFIL
$      TAPE      21,X2D,,4572
$      FFILE     21,NSTDLB,NOSRLS,FIXLNG/10,BUFSIZ/320,MLTFIL
```

COL.

1	8	16	30	45	60
$	FILE	11,D1R,50S			
$	FILE	12,D2R,50S			
$	FILE	13,D3R,50S			
$	FILE	14,D4R,50S			
$	FILE	15,D5R,50S			
$	FILE	H*,F6R,144R			
$	SYSOUT	18			
$	SYSOUT	19			
$	INCODE	IBMF			
DATA	21	19.			
PARAM	01				
CONSTR	01	4476.	169.		
AREA		21.	45.	270.	315.
$	ENDJOB				

3.5 PROGRAM OUTPUT

The program output which would result from the setup described in Section 3.4.2 has been reproduced on the following pages. The printed output shows the area and the number of parameters used (Table 4). A tally of rev types and a list of revs for each type is printed (Table 5). The revs which were constrained are printed in Table 6. Tables 7 and 8 differ only when passes are edited because an insufficient number of crossings exists to solve for the requested set of parameters for each pass. E.g., it is not possible to estimate both bias and slope parameters for a pass with only one crossing, and the pass is accordingly deleted. The allocation of memory for matrices and their starting addresses are shown in Table 9. Table 10 shows the biases found (note that Rev 4476/Seg. 169 was constrained). Tables 11 and 12 show the corrections made at the crossing points (respective of the north-to-south in Table 11 and the south-to-north in Table 12).

Additional output has been included to show the effect that passes reaching maximum latitude have on the program; i.e., how the program handles passes that can be typed in more than one way (Tables 13-21).

The data which is corrected and output on 9-track tape, readable by the ECLIPSE S200, is described in Table 22.

SEAHT

LEAST SQUARES DETERMINATION OF SMOOTHED SEA SURFACE HEIGHTS

CALCULATED AT INTERSECTING POINTS OF CROSSING REVOLUTIONS

CALIBRATION AREA

LATITUDE 21.00 TO 45.00

LONGITUDE 270.00 TO 315.00

NO.OF PARAMETERS = 1

NO. OF SOUTH-TO-NORTH REVS = 7

NO. OF NORTH-TO-SOUTH REVS = 12

TABLE 4

NORTH-TO-SOUTH REVOLUTIONS

REV NO	SEG NO
4476	169
4334	82
4604	249
4462	160
4391	117
1974	294
1576	462
2102	387
1178	104
3552	879
1718	81
1988	303

SOUTH-TO-NORTH REVOLUTIONS

REV NO	SEG NO
4610	255
4624	278
4482	175
4553	218
2094	382
1710	73
2151	422

TABLE 5

**********CONSTRAINED REVS***********

REV NO.	SEG NO.	TYPE OF CONSTRAINT		
		HEIGHT	SLOPE	CURVATURE
4476	169	*		

TABLE 6

ARC CROSSING DATA INPUT FOR THIS RUN

REVOLUTION NO NORTH-SOUTH (REV/SEG)		SOUTH-NORTH (REV/SEG)		CROSSING POINTS LATITUDE (DEGREES)	LONGITUDE (DEGREES)	SMOOTHED SEA SURFACE HEIGHT NORTH-SOUTH (METERS)	SOUTH-NORTH (METERS)	TIME FROM EPOCH NORTH-SOUTH (MINUTES)	SOUTH-NORTH (MINUTES)
4476	169	4610	255	32.45963	291.84935	-53.41368	-53.21657	2144.82202	2094.13519
4476	169	4624	278	35.89903	294.58823	-44.48449	-44.83744	514.90916	2053.33249
4476	169	4482	175	33.33520	292.52428	-51.29365	-51.48305	1732.18018	2144.07568
4476	169	4553	218	34.63855	293.55638	-47.98283	-48.35899	1115.11148	1714.92035
4476	169	2094	382	27.67230	288.38274	-58.62893	-57.23110	4378.14001	2002.44388
4476	169	1710	73	30.58033	290.44606	-58.16207	-57.57306	3025.93799	1970.30901
4334	82	4610	255	35.07880	289.78633	-51.44165	-51.51245	1882.11766	3329.28116
4334	82	4624	278	38.29404	292.52493	-46.23445	-46.01506	340.31153	3200.98703
4334	82	4482	175	35.89852	290.46118	-49.84351	-49.84628	1491.41762	3357.28058
4334	82	4553	218	37.11724	291.49323	-48.38054	-48.47816	907.62082	2895.14792
4334	82	1710	73	33.31563	288.38353	-56.32589	-56.58433	2717.12689	3249.20496
4604	249	4610	255	33.33507	291.17394	-52.53151	-53.07022	1783.27779	2505.90063
4604	249	4624	278	36.70143	293.91276	-45.11979	-45.46606	182.41389	2436.21045
4604	249	4482	175	34.19322	291.84882	-50.79486	-51.44217	1377.85480	2548.61484
4604	249	4553	218	35.46865	292.88094	-47.71143	-48.47261	771.75292	2108.57404
4604	249	2094	382	28.64370	287.70745	-55.68188	-55.99393	3979.45212	2450.49774
4604	249	1710	73	31.49521	289.77079	-56.72089	-57.42927	2649.26337	2396.45407
4604	249	2151	422	26.14844	285.99549	-50.69884	-51.54139	5133.90637	2391.24704
4462	160	4553	218	37.89196	290.81810	-47.82329	-48.18358	535.48232	3267.14432
4391	117	4482	175	37.89222	288.75386	-48.02431	-48.74884	532.98451	4311.45142
1974	294	4610	255	33.37986	291.13981	-52.75601	-53.10049	1688.78401	2526.51672
1974	294	4624	278	36.74193	293.87828	-45.17331	-45.43630	91.68554	2455.56882
1974	294	4482	175	34.23619	291.81462	-50.86588	-51.42990	1284.28571	2568.91663
1974	294	4553	218	35.51037	292.84657	-48.20100	-48.53621	679.60391	2128.40247
1974	294	2094	382	28.69168	287.67375	-55.64155	-56.03182	3880.30493	2472.67120
1974	294	1710	73	31.54072	289.73684	-56.99460	-57.48225	2552.85614	2417.68823
1974	294	2151	422	26.19841	285.96199	-51.02344	-51.39341	5032.57422	2414.17776
1576	462	4610	255	34.23416	290.46656	-52.99675	-53.02664	1202.31419	2929.42444
1576	462	4482	175	35.07230	291.14130	-51.52079	-51.52854	805.12060	2964.67441
1576	462	4553	218	36.31889	292.17323	-48.26258	-49.19711	211.50903	2513.41693
1576	462	2094	382	29.64004	287.00087	-55.10346	-55.01818	3356.01572	2911.67218
1576	462	1710	73	32.43319	289.06383	-56.91784	-56.98485	2051.03781	2834.84293
1576	462	2151	422	27.19248	285.28925	-50.34713	-50.47452	4489.88635	2871.02502
2102	387	4610	255	34.25205	290.45231	-52.33857	-52.98447	1478.91104	2937.87918
2102	387	4482	175	35.08976	291.12711	-50.66233	-51.48080	1081.68381	2972.95511
2102	387	4553	218	36.33569	292.15907	-48.16755	-49.26047	488.02897	2521.43231
2102	387	2094	382	29.66020	286.98638	-55.05340	-55.05228	3632.77325	2921.04388
2102	387	1710	73	32.45202	289.04948	-56.81059	-56.96498	2327.69366	2843.66019
2102	387	2151	422	27.21384	285.27465	-51.23919	-50.52425	4766.73920	2880.85709
1178	104	4610	255	35.05774	289.80349	-51.47779	-51.57277	1606.56949	3319.29059

TABLE 7

1178	104	4624	278	38,27530	292,54158	-46,46430	-46,03265	67,42700	3191,98083
1178	104	4482	175	35,87807	290,47823	-50,02304	-49,83742	1216,52719	3347,54004
1178	104	4553	218	37,09769	291,51006	-48,14527	-48,50958	633,73615	2885,77856
1178	104	2094	332	30,55628	286,33811	-54,91659	-54,98199	3723,01672	3337,09067
1178	104	1710	73	33,29426	288,40095	-55,91979	-56,59882	2440,21732	3238,70004
1178	104	2151	422	28,15411	284,62661	-51,34346	-51,75623	4838,60590	3314,17050
3552	879	4610	255	33,39451	291,12839	-46,94230	-53,11638	1844,67471	2533,40982
3552	879	4624	278	36,75532	293,86686	-41,20212	-45,45241	246,14857	2461,97546
3552	879	4482	175	34,25052	291,80320	-45,19866	-51,45007	1439,82391	2575,68604
3552	879	4553	218	35,52424	292,83518	-43,01505	-48,55503	834,60037	2134,99152
3552	879	2094	332	28,70793	287,66233	-46,75507	-56,04250	4038,01172	2480,18259
3552	879	1710	73	31,55601	289,72542	-49,95378	-57,49687	2709,48355	2424,82361
3552	879	2151	422	26,21544	285,95057	-40,58103	-51,54337	5191,16931	2421,99570
1718	81	4610	255	31,57794	292,51498	-52,79746	-52,65171	1677,58382	1681,29201
1718	81	4624	278	35,09123	295,25541	-44,53096	-45,06153	16,48501	1669,41060
1718	81	4482	175	32,47186	293,18975	-50,64760	-50,70387	1253,44106	1738,45343
1718	81	4553	218	33,80306	294,22169	-47,88057	-48,04806	626,49030	1320,18997
1718	81	2094	332	26,69729	289,04887	-58,22546	-58,05201	3939,24167	1553,52240
1718	81	1710	73	29,66031	291,11182	-57,16647	-57,30741	2567,31482	1543,11702
1718	81	2151	422	24,11172	287,33712	-52,86048	-52,86401	5127,53009	1458,96642
1988	303	4610	255	29,71754	293,87719	-52,42044	-52,65150	1959,16600	814,55627
1988	303	4624	278	33,38137	296,61583	-44,07055	-44,15325	243,44985	861,44814
1988	303	4482	175	30,64854	294,55204	-50,80455	-51,70658	1525,30772	886,32440
1988	303	4553	218	32,03556	295,58401	-44,49924	-44,72337	875,87438	490,18060
1988	303	2094	332	24,64896	290,41068	-58,42932	-58,69779	4300,31201	614,65333
1988	303	1710	73	27,72268	292,47371	-55,49784	-55,92331	2884,51242	647,50870
1988	303	2151	422	21,97517	288,69878	-53,67839	-54,06509	5523,58356	485,65553

TABLE 7 (Cont.)

ARC CROSSING DATA USED FOR ADJUSTMENTS

REVOLUTION NO				CROSSING POINTS		SMOOTHED SEA SURFACE HEIGHT		TIME FROM EPOCH	
NORTH-SOUTH (REV/SEG)		SOUTH-NORTH (REV/SEG)		LATITUDE (DEGREES)	LONGITUDE (DEGREES)	NORTH-SOUTH (METERS)	SOUTH-NORTH (METERS)	NORTH-SOUTH (MINUTES)	SOUTH-NORTH (MINUTES)
4476	169	4610	255	32.45968	291.84935	-53.41368	-53.21657	2144.82202	2094.13519
4476	169	4624	278	35.89903	294.58823	-44.48449	-44.83744	514.90916	2053.33249
4476	169	4482	175	33.33520	292.52428	-51.29365	-51.48305	1732.18018	2144.07568
4476	169	4553	218	34.63856	293.55638	-47.98283	-48.35899	1115.11148	1714.92035
4476	169	2094	332	27.67280	288.38274	-58.62898	-57.23110	4378.14001	2002.44388
4476	169	1710	73	30.58033	290.44606	-58.16207	-57.57306	3025.93799	1970.30901
4334	82	4610	255	35.07880	289.78633	-51.44165	-51.51245	1882.11766	3329.28116
4334	82	4624	278	38.29404	292.52493	-46.23445	-46.01506	340.31153	3200.98703
4334	82	4482	175	35.89852	290.46118	-49.84351	-49.84628	1491.41762	3357.28058
4334	82	4553	218	37.11724	291.49323	-48.38054	-48.47816	907.62082	2895.14792
4334	82	1710	73	33.31663	288.38353	-56.32589	-56.58433	2717.12689	3249.20496
4604	249	4610	255	33.33507	291.17394	-52.53151	-53.07022	1783.27779	2505.90063
4604	249	4624	278	36.70145	293.91276	-45.11979	-45.40606	182.41389	2436.21045
4604	249	4482	175	34.19322	291.84882	-50.79486	-51.44217	1377.85480	2548.61484
4604	249	4553	218	35.46865	292.88094	-47.71143	-48.47261	771.75292	2108.57404
4604	249	2094	332	28.64370	287.70745	-55.68188	-55.99393	3979.45212	2450.49774
4604	249	1710	73	31.49521	289.77079	-56.72089	-57.42927	2649.26337	2396.45407
4604	249	2151	422	26.14844	285.99549	-50.69884	-51.54139	5133.90637	2391.24704
4462	160	4553	218	37.89196	290.81810	-47.82329	-48.18358	535.48232	3267.14432
4391	117	4482	175	37.89222	288.75386	-48.02431	-48.74884	532.98451	4311.45142
1974	294	4610	255	33.37986	291.16981	-52.75601	-53.10049	1688.78401	2526.51672
1974	294	4624	278	36.74193	293.87828	-45.17331	-45.45630	91.68554	2455.56882
1974	294	4482	175	34.23519	291.81462	-50.86586	-51.42990	1284.28571	2568.91663
1974	294	4553	218	35.51037	292.84657	-48.20100	-48.53621	679.60391	2128.40247
1974	294	2094	332	28.69168	287.67375	-55.64155	-56.06182	3880.30493	2472.67120
1974	294	1710	73	31.54072	289.73684	-56.99466	-57.48225	2552.85614	2417.68823
1974	294	2151	422	26.19841	285.96199	-51.02344	-51.39341	5032.57422	2414.17776
1576	462	4610	255	34.23416	290.46656	-52.99675	-53.02664	1202.31419	2929.42444
1576	462	4482	175	35.07230	291.14130	-51.52079	-51.52854	805.12060	2964.67441
1576	462	4553	218	36.31389	292.17623	-48.26258	-49.19711	211.50906	2513.41693
1576	462	2094	332	29.64004	287.00087	-55.10346	-55.01818	3356.01572	2911.67218
1576	462	1710	73	32.43319	289.06383	-56.91784	-56.98485	2051.03781	2834.84293
1576	462	2151	422	27.19248	285.28925	-50.34713	-50.47452	4489.88635	2871.02502
2102	387	4610	255	34.25205	290.45231	-52.33857	-52.98447	1478.91104	2937.87918
2102	387	4482	175	35.08976	291.12711	-50.66233	-51.48080	1081.68381	2972.95511
2102	387	4553	218	36.33569	292.15907	-48.16755	-49.26047	488.02897	2521.43231
2102	387	2094	332	29.66026	286.98638	-55.05340	-55.05228	3632.77325	2921.04388
2102	387	1710	73	32.45202	289.04948	-56.81059	-56.96498	2327.69366	2843.66019
2102	387	2151	422	27.21384	285.27465	-51.23919	-50.52425	4766.73920	2880.85709
1178	104	4610	255	35.05774	289.80349	-51.47779	-51.57277	1606.56949	3319.29059

TABLE 8

1178	104	4624	278	38,27536	292,54158	-46,46430	-46,03265	67,42700	3191,98083
1178	104	4482	175	35,87807	290,47823	-50,02304	-49,83742	1216,52719	3347,54004
1178	104	4553	218	37,09769	291,51006	-48,14527	-48,50958	633,73615	2885,77856
1178	104	2094	392	30,55528	286,33811	-54,91659	-54,98199	3723,01672	3337,09067
1178	104	1710	73	33,29428	288.40095	-55,91979	-56,59882	2440,21732	3238,70004
1178	104	2151	422	28,15411	284,62661	-51,34346	-51,75623	4838,60590	3314,17050
3552	879	4610	255	33,39451	291,12839	-46,94230	-53,11638	1844,67471	2533,40982
3552	879	4624	278	36,75532	293,86686	-41,20212	-45,43241	246,14857	2461,97546
3552	879	4482	175	34,25052	291,80520	-45,19866	-51,43007	1439,82391	2575,68604
3552	879	4553	218	35,52424	292,83518	-43,01505	-48,55303	834,60037	2134,99152
3552	879	2094	392	28,70793	287,66233	-46,75507	-56,04250	4038,01172	2480,18259
3552	879	1710	73	31,55601	289,72542	-49,95378	-57,49687	2709,48355	2424,82361
3552	879	2151	422	26,21544	285,95057	-40,58103	-51,34337	5191,16931	2421,99570
1718	81	4610	255	31,57794	292,51498	-52,79746	-52,65171	1672,58362	1681,29201
1718	81	4624	278	35,09123	295,25541	-44,53096	-45,06153	16,48501	1669,41060
1718	81	4482	175	32,47180	293,18975	-50,64760	-50,70387	1253,44106	1738,45343
1718	81	4553	218	33,80306	294,22169	-47,88057	-48,04806	626,49030	1320,18997
1718	81	2094	392	26,69729	289,04887	-58,22546	-58,05201	3939,24167	1553,52240
1718	81	1710	73	29,66031	291,11182	-57,16647	-57,30741	2567,31482	1543,11702
1718	81	2151	422	24,11172	287,33712	-52,86048	-52,86401	5127,53009	1458,96642
1988	303	4610	255	29,71754	293,87719	-52,42044	-52,65150	1959,16600	814,55627
1988	303	4624	278	33,38137	296,61583	-44,07055	-44,13325	243,44985	861,44814
1988	303	4482	175	30,64854	294,55204	-50,80455	-51,70658	1525,30772	886,32440
1988	303	4553	218	32,03558	295,58401	-44,49924	-44,72337	875,87438	490,18060
1968	303	2094	382	24,64896	290,41068	-58,42932	-58,69779	4300,31201	614,65333
1988	303	1710	73	27,72268	292,47371	-55,49784	-55,92331	2884,51242	647,50870
1988	303	2151	422	21,97517	288,69878	-53,67839	-54,06509	5523,58356	485,65553

TABLE 8 (Cont.)

```
CHARLY-DYNAMIC CORE ALLOCATOR-

  1089WORDS OF CORE HAVE BEEN
DISTRIBUTED AMONG THE FOLLOWING ARRAYS

          SNREV         9
          NSREV        12
          SUM1         72
          SUM2         16
          DELTA        16
          BBIAS         8
          KNSTR         2
          ARRAY       956


 STARTING ADDRESSES OF ARRAYS

    SNREV     126700000000
    NSREV     126710000000
    SUM1      126724000000
    SUM2      127034000000
    DELTA     127054000000
    BBIAS     127074000000
    KNSTR     127104000000
    ARRAY     127106000000
```


TABLE 9

*****BIASES*****

NORTH-TO-SOUTH REVS

REV NO	SEG NO	COEFFICIENTS
4476	169	0.2166438E-14
4434	82	0.2608515E 00
4604	249	0.7391857E 00
4462	160	0.4081491E 00
4391	117	0.6481592E 00
1974	294	0.5389130E 00
1576	462	0.2523978E 00
2102	387	0.4047830E 00
1178	104	0.2881538E 00
3552	879	0.7254924E 01
1718	81	0.2282054E 00
1988	303	0.5026312E 00

SOUTH-TO-NORTH REVS

REV NO	SEG NO	COEFFICIENTS
4610	255	0.2482993E 00
4624	278	0.5847675E 00
4482	175	0.1236266E 00
4553	218	0.5784947E-01
2094	382	0.1714784E 00
1730	73	0.5947062E-01
2121	422	0.2476388E 00

TABLE 10

REVOLUTIONS		CROSSING PTS.		BIASES		A PRIORI SEA SURFACE HEIGHTS		ADJUSTED SEA SURFACE HEIGHTS		AVERAGE	RESIDUALS
N/S	S/N	LATITUDE (DEGREES)	LONGITUDE (DEGREES)	N/S (METERS)	N/S (METERS)	S/N (METERS)	N/S (METERS)	S/N (METERS)	N/S (METERS)	(ADJUSTED) (METERS)	N/S (METERS)
4476	4610	32.459675	291.849354	-0.0000	0.2883	-53.4137	-53.2166	-53.4137	-53.4849	-53.4493	0.0356
4476	4624	35.399035	294.588226	-0.0000	0.5848	-44.4345	-44.8374	-44.4845	-45.4222	-44.9533	0.4689
4476	4482	33.335204	292.524251	-0.0000	0.1236	-51.2937	-51.4830	-51.2937	-51.6067	-51.4502	0.1565
4476	4553	34.638564	293.556377	-0.0000	0.0578	-47.9328	-48.3590	-47.9828	-48.4168	-48.1998	0.2170
4476	2094	27.672798	298.382736	-0.0000	0.1715	-58.6290	-57.2311	-58.6290	-57.4026	-58.0158	-0.6132
4476	1710	30.580333	290.446050	-0.0000	0.0595	-58.1621	-57.5731	-58.1621	-57.6325	-57.8973	-0.2648
4334	4610	35.078804	289.786335	0.2609	0.2883	-51.4417	-51.5124	-51.7025	-51.7807	-51.7416	0.0391
4334	4624	38.294036	292.524925	0.2609	0.5848	-46.2345	-46.0151	-46.4953	-46.5998	-46.5476	0.0523
4334	4482	35.898524	290.461178	0.2609	0.1236	-49.8435	-49.8463	-50.1044	-49.9699	-50.0371	-0.0672
4334	4553	37.117242	291.493225	0.2609	0.0578	-48.3805	-48.4782	-48.6414	-48.5360	-48.5887	-0.0527
4334	1710	33.316635	288.383526	0.2609	0.0595	-56.3259	-56.5843	-56.5867	-56.6438	-56.6153	0.0285
4604	4610	33.336068	291.273943	0.7392	0.2883	-52.5315	-53.0702	-53.2707	-53.3385	-53.3046	0.0339
4604	4624	36.701446	293.912762	0.7392	0.5848	-45.1198	-45.4661	-45.8590	-46.0508	-45.9549	0.0959
4604	4482	34.193219	291.848816	0.7392	0.1236	-50.7949	-51.4422	-51.5341	-51.5658	-51.5499	0.0159
4604	4553	35.468647	292.880936	0.7392	0.0578	-47.7114	-48.4726	-48.4506	-48.5305	-48.4905	0.0399
4604	2094	28.643783	287.707451	0.7392	0.1715	-55.6819	-55.9939	-56.4211	-56.1654	-56.2932	-0.1278
4604	1710	31.495209	289.770786	0.7392	0.0595	-56.7209	-57.4293	-57.4601	-57.4887	-57.4744	0.0143
4604	2151	26.148443	285.995491	0.7392	-0.2676	-50.6988	-51.5414	-51.4380	-51.2937	-51.3659	-0.0721
4462	4553	37.891960	290.818096	0.4181	0.0578	-47.8233	-48.1836	-48.2414	-48.2414	-48.2414	0.
4391	4482	37.892222	288.753857	0.8482	0.1236	-48.0243	-48.7488	-48.8725	-48.8725	-48.8725	0.

ORIGINAL PAGE IS OF POOR QUALITY

TABLE 11

1974	4610	33.379863	291.139809	0.5389	0.2583	-52.7560	-53.1005	-53.2949	-53.3688	-53.3319	0.0369
1974	4624	36.741929	293.878277	0.5389	0.5848	-45.1733	-45.4363	-45.7122	-46.0211	-45.8666	0.1544
1974	4482	34.236194	291.814617	0.5389	0.1236	-50.8659	-51.4299	-51.4048	-51.5535	-51.4792	0.0744
1974	4553	35.510372	292.846565	0.5389	0.0578	-48.2010	-48.5362	-48.7399	-48.5941	-48.6670	-0.0729
1974	2094	28.691679	287.673752	0.5389	0.1715	-55.6416	-56.0618	-56.1805	-56.2033	-56.1919	0.0114
1974	1710	31.540717	289.736835	0.5389	0.0595	-56.9947	-57.4823	-57.5336	-57.5417	-57.5376	0.0041
1974	2151	26.198408	285.961994	0.5389	0.2876	-51.0234	-51.3934	-51.5624	-51.1458	-51.3541	-0.2083
1976	4610	34.234157	290.466564	0.2524	0.2583	-52.9967	-53.0266	-53.2491	-53.2949	-53.2720	0.0229
1976	4482	35.072296	291.641300	0.2524	0.1236	-51.5208	-51.5285	-51.7732	-51.6522	-51.7127	-0.0605
1976	4553	36.318888	292.173225	0.2524	0.0578	-48.2626	-49.1971	-48.5150	-49.2550	-48.8850	0.3700
1976	2094	29.640040	287.600866	0.2524	0.1715	-55.1035	-55.0182	-55.3559	-55.1897	-55.2728	-0.0831
1976	1710	32.433189	289.663831	0.2524	0.0595	-56.9178	-56.9848	-57.1702	-57.0443	-57.1073	-0.0630
1976	2151	27.192479	285.289246	0.2524	0.2876	-50.3471	-50.4745	-50.5995	-50.2269	-50.4132	-0.1863
2102	4610	34.252049	290.452312	0.4048	0.2583	-52.3386	-52.9845	-52.7434	-53.2528	-52.9981	0.2547
2102	4482	35.089757	291.127306	0.4048	0.1236	-50.6623	-51.4808	-51.0671	-51.6044	-51.3358	0.2687
2102	4553	36.335685	292.159065	0.4048	0.0578	-48.1676	-49.2605	-48.5723	-49.3183	-48.9453	0.3730
2102	2094	29.660255	286.986382	0.4048	0.1715	-55.0534	-55.0523	-55.4582	-55.2238	-55.3410	-0.1172
2102	1710	32.452020	289.649480	0.4048	0.0595	-56.8106	-56.9650	-57.2154	-57.0245	-57.1199	-0.0955
2102	2151	27.213644	285.274654	0.4048	0.2876	-51.2392	-50.5242	-51.6440	-50.2766	-50.9603	-0.6837
1978	4610	35.057742	289.803490	0.2882	0.2583	-51.4778	-51.5728	-51.7659	-51.8411	-51.8035	0.0376
1978	4624	38.275364	292.541580	0.2882	0.5848	-46.4643	-46.0527	-46.7525	-46.6174	-46.6849	-0.0675
1978	4482	35.878066	290.478253	0.2882	0.1236	-50.0230	-49.8574	-50.3112	-49.9610	-50.1361	-0.1751

TABLE 11 (Cont.)

1178	4553	37.097687	291.510063	0.2882	0.0578	-48.1453	-48.5096	-48.4334	-48.5674	-48.5004	0.0670
1178	2094	30.556280	286.838108	0.2882	0.1715	-54.9166	-54.9820	-55.2047	-55.1535	-55.1791	-0.0256
1178	1710	33.294276	288.400948	0.2882	0.0595	-55.9198	-56.5988	-56.2079	-56.6583	-56.4331	0.2252
1178	2151	28.154114	284.626406	0.2882	-0.2876	-51.3435	-51.7562	-51.6316	-51.5086	-51.5701	-0.0615
3552	4610	33.394506	291.228391	7.2549	0.2883	-46.9423	-53.1164	-54.1972	-53.3847	-53.7909	-0.4063
3552	4624	36.755324	293.866856	7.2549	0.5848	-41.2021	-45.4324	-48.4570	-46.0172	-47.2371	-1.2199
3552	4482	34.250523	291.803204	7.2549	0.1236	-45.1987	-51.4301	-52.4536	-51.5537	-52.0036	-0.4499
3552	4553	35.524237	292.835178	7.2549	0.0578	-43.0151	-48.5530	-50.2700	-48.6109	-49.4404	-0.8295
3552	2094	28.707930	287.662331	7.2549	0.1715	-46.7551	-56.0425	-54.0100	-56.2140	-55.1120	1.1020
3552	1710	31.556008	289.725418	7.2549	0.0595	-49.9538	-57.4969	-57.2087	-57.5563	-57.3825	0.1738
3552	2151	26.215442	285.950565	7.2549	-0.2876	-40.5810	-51.3434	-47.8360	-51.0957	-49.4658	1.6299
1718	4610	31.577943	292.514776	0.2282	0.2883	-52.7975	-52.6517	-53.0257	-52.9200	-52.9728	-0.0528
1718	4624	35.091232	295.253107	0.2282	0.5848	-44.5310	-45.0615	-44.7592	-45.6463	-45.2027	0.4436
1718	4482	32.471801	293.189754	0.2282	0.1236	-50.6476	-50.7039	-50.8758	-50.8275	-50.8517	-0.0242
1718	4553	33.803059	294.221687	0.2282	0.0578	-47.8806	-48.0481	-48.1088	-48.1059	-48.1073	-0.0014
1718	2094	26.697291	289.648370	0.2282	0.1715	-58.2255	-58.0520	-58.4537	-58.2235	-58.3386	-0.1151
1718	1710	29.660306	291.111824	0.2282	0.0595	-57.1665	-57.3074	-57.3947	-57.3669	-57.3808	-0.0139
1718	2151	24.111718	287.837124	0.2282	-0.2876	-52.3605	-52.8640	-53.0887	-52.6164	-52.8525	-0.2362
1988	4610	29.717535	293.877193	0.5026	0.2883	-52.4204	-52.6515	-52.9231	-52.9198	-52.9214	-0.0016
1988	4624	33.381367	296.615329	0.5026	0.5848	-44.0706	-44.1333	-44.5732	-44.7180	-44.6456	0.0724
1988	4482	30.648543	294.552044	0.5026	0.1236	-50.8046	-51.7066	-51.3072	-51.8302	-51.5687	0.2615
1988	4553	32.036525	295.584011	0.5026	0.0578	-44.4992	-44.7234	-45.0019	-44.7812	-44.8915	-0.1103

TABLE 11 (Cont.)

1988	2094	24.648965	290.410675	0.5026	0.1715	-58.4293	-58.6978	-58.9320	-58.8693	-58.9006	-0.0313
1988	1710	27.722684	292.473713	0.5026	0.0595	-55.4978	-55.9233	-56.0005	-55.9828	-55.9916	-0.0088
1988	2151	21.975166	288.698776	0.5026	-0.2476	-53.6784	-54.0651	-54.1810	-53.8175	-53.9992	-0.1818

MEAN OF RESIDUALS -0.00000

STANDARD DEVIATION 0.87305

TABLE 11 (Cont.)

REVOLUTIONS S/N	REVOLUTIONS N/S	CROSSING PTS. LATITUDE (DEGREES)	CROSSING PTS. LONGITUDE (DEGREES)	BIASES S/N (METERS)	BIASES S/N (METERS)	A PRIORI SEA SURFACE HEIGHTS N/S (METERS)	A PRIORI SEA SURFACE HEIGHTS S/N (METERS)	ADJUSTED SEA SURFACE HEIGHTS N/S (METERS)	ADJUSTED SEA SURFACE HEIGHTS S/N (METERS)	AVERAGE (ADJUSTED) (METERS)	RESIDUALS S/N (METERS)
4610	4476	32.459675	291.849354	0.2683	0.0000	-53.2166	-53.4137	-53.4849	-53.4137	-53.4493	-0.0356
4610	4334	35.078804	289.786335	0.2683	0.2809	-51.5124	-51.4417	-51.7807	-51.7025	-51.7416	-0.0391
4610	4604	33.336068	291.173943	0.2683	0.7392	-53.0702	-52.5315	-53.3385	-53.2707	-53.3046	-0.0339
4610	1974	33.379863	291.139309	0.2683	0.5389	-53.1005	-52.7560	-53.3688	-53.2949	-53.3319	-0.0369
4610	1576	34.234157	290.866564	0.2683	0.2524	-53.0266	-52.9967	-53.2949	-53.2491	-53.2720	-0.0229
4610	2102	34.252049	290.852312	0.2683	0.4048	-52.9845	-52.3386	-53.2528	-52.7434	-52.9981	-0.2547
4610	1178	35.057742	289.803490	0.2683	0.2882	-51.5728	-51.4778	-51.8411	-51.7659	-51.8035	-0.0376
4610	3552	33.394506	291.128391	0.2683	7.2549	-53.1164	-46.9423	-53.3847	-54.1972	-53.7909	0.4063
4610	1718	31.577943	292.514976	0.2683	0.2282	-52.6517	-52.7975	-52.9200	-53.0257	-52.9728	0.0528
4610	1988	29.717535	293.877193	0.2683	0.5026	-52.6515	-52.4204	-52.9198	-52.9231	-52.9214	0.0016
4624	4476	35.899035	294.586226	0.5848	0.0000	-44.8374	-44.4845	-45.4222	-44.4845	-44.9533	-0.4689
4624	4334	36.294036	292.524925	0.5848	0.2809	-46.0151	-46.2345	-46.5998	-46.4953	-46.5476	-0.0523
4624	4604	36.701446	293.912762	0.5848	0.7392	-45.4661	-45.1198	-46.0508	-45.8590	-45.9549	-0.0959
4624	1974	36.741929	293.878277	0.5848	0.5389	-45.4363	-45.1733	-46.0211	-45.7122	-45.8666	-0.1544
4624	1178	36.275364	292.541580	0.5848	0.2882	-46.0327	-46.4643	-46.6174	-46.7525	-46.6849	0.0675
4624	3552	36.755324	293.866856	0.5848	7.2549	-45.4324	-41.2021	-46.0172	-48.4570	-47.2371	1.2199
4624	1718	35.091232	295.253407	0.5848	0.2282	-45.0615	-44.5310	-45.6463	-44.7592	-45.2027	-0.4436
4624	1988	33.381367	296.615829	0.5848	0.5026	-44.1333	-44.0706	-44.7180	-44.5732	-44.6456	-0.0724
4682	4476	33.335204	292.524281	0.1236	0.0000	-51.4830	-51.2937	-51.6067	-51.2937	-51.4502	-0.1565
4682	4334	35.898524	290.861178	0.1236	0.2809	-49.8463	-49.8435	-49.9699	-50.1044	-50.0371	0.0672

TABLE 12

4482	4604	34.193219	291.848816	0.1236	0.7392	-51.4422	-50.7949	-51.5658	-51.5341	-51.5499	-0.0159
4482	4391	37.892222	288.753857	0.1236	0.8882	-48.7488	-48.0243	-48.8725	-48.8725	-48.8725	0.
4482	1974	34.236194	291.814617	0.1236	0.5589	-51.4299	-50.8659	-51.5535	-51.4048	-51.4792	-0.0744
4482	1576	35.072296	291.141300	0.1236	0.2524	-51.5285	-51.5208	-51.6522	-51.7732	-51.7127	0.0605
4482	2102	35.089757	291.127106	0.1236	0.4048	-51.4808	-50.6623	-51.6044	-51.0671	-51.3358	-0.2687
4482	1178	35.873066	290.478233	0.1236	0.2882	-49.8374	-50.0230	-49.9610	-50.3112	-50.1361	0.1751
4482	3552	34.250523	291.802204	0.1236	7.2549	-51.4301	-45.1987	-51.5537	-52.4536	-52.0036	0.4499
4482	1718	32.471801	293.189754	0.1236	0.2282	-50.7039	-50.6476	-50.8275	-50.8758	-50.8517	0.0242
4482	1988	30.648543	294.552044	0.1236	0.5026	-51.7066	-50.8046	-51.8302	-51.3072	-51.5687	-0.2615
4553	4476	34.638564	293.556377	0.0578	*0.0000	-48.3590	-47.9828	-48.4168	-47.9828	-48.1998	-0.2170
4553	4334	37.117242	291.497225	0.0578	0.2609	-48.4782	-48.3805	-48.5360	-48.6414	-48.5887	0.0527
4553	4604	35.468647	292.880936	0.0578	0.7392	-48.4726	-47.7114	-48.5305	-48.4506	-48.4905	-0.0399
4553	4462	37.891960	290.818096	0.0578	0.4181	-48.1836	-47.8233	-48.2414	-48.2414	-48.2414	0.
4553	1974	35.510372	292.846565	0.0578	0.5589	-48.5362	-48.2010	-48.5941	-48.7399	-48.6670	0.0729
4553	1576	34.318888	292.173225	0.0578	0.2524	-49.1971	-48.2626	-49.2550	-48.5150	-48.8850	-0.3700
4553	2102	36.335685	292.159065	0.0578	0.4048	-49.2605	-48.1676	-49.3183	-48.5723	-48.9453	-0.3730
4553	1178	37.097687	291.510063	0.0578	0.2882	-48.5096	-48.1453	-48.5674	-48.4334	-48.5004	-0.0670
4553	3552	35.524237	292.835178	0.0578	7.2549	-48.5530	-43.0151	-48.6109	-50.2700	-49.4404	0.8295
4553	1718	33.803059	294.221687	0.0578	0.2282	-48.0481	-47.8806	-48.1059	-48.1088	-48.1073	0.0014
4553	1988	32.036575	295.584011	0.0578	0.5026	-44.7234	-44.4992	-44.7812	-45.0019	-44.8915	0.1103
2094	4476	27.672798	288.882736	0.1715	*0.0000	-57.2311	-58.6290	-57.4026	-58.6290	-58.0158	0.6132
2094	4604	28.643703	287.707451	0.1715	0.7392	-55.9939	-55.6819	-56.1654	-56.4211	-56.2932	0.1278

TABLE 12 (Cont.)

2094	1974	28,691679	287,673752	0.1715	0.5389	-56.0318	-55,6416	-56,2033	-56,1805	-56,1919	-0.0114
2094	1576	29,640040	287,900866	0.1715	0.2524	-55.0182	-55,1035	-55,1897	-55,3559	-55,2728	0.0831
2094	2102	29,660255	286,986382	0.1715	0.4848	-55,0523	-55,0534	-55,2238	-55,4582	-55,3410	0.1172
2094	1178	30,556280	286,839108	0.1715	0.2882	-54,9820	-54,9166	-55,1535	-55,2047	-55,1791	0.0256
2094	3552	28,707930	287,862331	0.1715	7.2549	-56,0425	-46,7551	-56,2140	-54,0100	-55,1120	-1.1020
2094	1718	26,697291	289,049870	0.1715	0.2282	-58,0520	-58,2255	-58,2235	-58,4537	-58,3386	0.1151
2094	1988	24.648965	290,410675	0.1715	0.5826	-58,6978	-58,4293	-58,8693	-58.9320	-58,9006	0.0313
1718	4476	30,580333	290,845060	0.0595	-0.0000	-57,5731	-58,1621	-57,6325	-58,1621	-57,8973	0.2648
1718	4334	33,316635	288,383526	0,0595	0.2609	-56,5843	-56,3259	-56,6438	-56,5867	-56,6153	-0.0285
1718	4604	31,495209	289,770786	0.0595	0.7892	-57.4293	-56,7209	-57,4887	-57,4601	-57,4744	-0.0143
1718	1974	31,540717	289,736835	0,0595	0.5389	-57,4823	-56,9947	-57,5417	-57,5336	-57,5376	-0.0041
1718	1576	32,433189	289,963831	0.0595	0.2524	-56,9848	-56,9178	-57,0443	-57,1702	-57,1073	0.0630
1718	2102	32,452020	289,049480	0,0595	0.4848	-56,9650	-56,8106	-57,0245	-57,2154	-57,1199	0.0955
1718	1178	33,294276	288,400948	0,0595	0.2882	-56,5988	-55,9198	-56,6583	-56.2079	-56,4331	-0.2252
1718	3552	31,556088	289,725418	0,0595	7,2549	-57,4969	-49,9538	-57,5563	-57,2087	-57,3825	-0.1738
1718	1718	27,660306	291,111824	0,0595	0.2282	-57.3074	-57,1665	-57,3669	-57,3947	-57,3808	0,0139
1718	1988	27,722684	292,473713	0.0595	0.5026	-55,9233	-55,4978	-55,9828	-56,0005	-55,9916	0.0088
2151	4604	26,148443	285,995491	-0,2476	0.7892	-51,5414	-50,6988	-51,2937	-51,4380	-51,3659	0.0721
2151	1974	26,198408	285,961994	-0,2476	0,5389	-51.3934	-51,0234	-51,1458	-51,5624	-51,3541	0.2083
2151	1576	27,192479	285,289246	-0,2476	0.2524	-50.4745	-50,3471	-50,2269	-50,5995	-50,4132	0.1863
2151	2102	27.213844	285,274654	-0,2476	0,4848	-50,5242	-51,2392	-50,2766	-51,6440	-50,9603	0.6837
2151	1178	28,154114	284,626606	-0,2476	0.2882	-51,7562	-51,3435	-51,5086	-51,6316	-51,5701	0.0615

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 12 (Cont.)

2151	3552	26.215442	285.950565	-0.2476	7.2249	-51.3434	-40.5810	-51.0957	-47.8360	-49.4658	-1.6299
2151	1718	24.111718	287.837124	-0.2476	0.2282	-52.8640	-52.8605	-52.6164	-53.0887	-52.8525	0.2362
2151	1988	21.975166	288.693776	-0.2476	0.5026	-54.0651	-53.6784	-53.8175	-54.1810	-53.9992	0.1818

TABLE 12 (Cont.)

SEAHT

LEAST SQUARES DETERMINATION OF SMOOTHED SEA SURFACE HEIGHTS

CALCULATED AT INTERSECTING POINTS OF CROSSING REVOLUTIONS

CALIBRATION AREA

LATITUDE	55.00	TO	67.00
LONGITUDE	300.00	TO	330.00

NO. OF PARAMETERS = 1

NO. OF SOUTH-TO-NORTH REVS = 9

NO. OF NORTH-TO-SOUTH REVS = 8

TABLE 13

NORTH-TO-SOUTH REVOLUTIONS

REV NO	SEG NO
640	563
2658	360
2829	479
4236	15
4577	232
4562	222
626	556
4548	213

SOUTH-TO-NORTH REVOLUTIONS

REV NO	SEG NO
640	563
2658	360
2673	376
2829	479
4236	15
626	556
683	591
641	564
1167	101

TABLE 14

***********CONSTRAINED REVS***********

REV NO	SEG NO	TYPE OF CONSTRAINT HEIGHT	SLOPE	CURVATURE
4236	15	*		
4548	213	*		

TABLE 15

ARC CROSSING DATA INPUT FOR THIS RUN

REVOLUTION NO NORTH-SOUTH (REV/SEG)		SOUTH-NORTH (REV/SEG)		CROSSING POINTS LATITUDE (DEGREES)	LONGITUDE (DEGREES)	SMOOTHED SEA SURFACE HEIGHT NORTH-SOUTH (METERS)	SOUTH-NORTH (METERS)	TIME FROM EPOCH NORTH-SOUTH (MINUTES)	SOUTH-NORTH (MINUTES)
640	563	640	563	65.13433	306.99883	22.42969	22.42969	-1.74089	-1.74089
2658	360	2658	360	65.12945	325.57243	60.85156	60.85156	8.37898	8.37898
2658	360	2829	479	65.04635	320.82355	50.11682	44.53872	8.93182	14.04771
4236	15	2829	479	65.04342	320.91197	43.00198	44.65421	5.74540	14.03740
4236	15	4236	15	65.12493	325.38430	51.92188	51.92188	5.22387	5.22387
4577	232	640	563	64.72919	319.71224	44.83345	33.16425	5.63083	-3.22911
4577	232	2829	479	64.04519	323.33159	52.49392	51.06031	5.20089	13.75370
4562	222	640	563	63.65492	329.57867	54.66899	56.53514	4.83760	-4.44844
4562	222	2673	376	63.49440	328.43145	52.87172	52.48216	4.98651	5.29311
4562	222	641	564	61.13342	316.89210	64.98653	41.40091	6.60434	9.03104
4562	222	1167	101	61.11954	316.83937	55.67219	64.53549	6.61237	8.64328
626	556	626	556	65.12077	300.44065	11.14063	11.14063	16.57750	16.57750

TABLE 16

ARC CROSSING DATA USED FOR ADJUSTMENTS

REVOLUTION NO NORTH-SOUTH (REV/SEG)		REVOLUTION NO SOUTH-NORTH (REV/SEG)		CROSSING POINTS LATITUDE (DEGREES)	CROSSING POINTS LONGITUDE (DEGREES)	SMOOTHED SEA SURFACE HEIGHT NORTH-SOUTH (METERS)	SMOOTHED SEA SURFACE HEIGHT SOUTH-NORTH (METERS)	TIME FROM EPOCH NORTH-SOUTH (MINUTES)	TIME FROM EPOCH SOUTH-NORTH (MINUTES)
640	563	640	563	65.13433	306.99883	22.42969	22.42969	-1.74089	-1.74089
2658	360	2658	360	65.12945	325.57243	60.85156	60.85156	8.37898	8.37898
2658	360	2829	479	65.04635	320.82355	50.11682	44.53872	8.93182	14.04771
4236	15	2829	479	65.04342	320.91197	43.00198	44.65421	5.74540	14.03740
4236	15	4236	15	65.12493	325.38430	51.92188	51.92188	5.22387	5.22387
4577	232	640	563	64.72919	319.71224	44.83345	33.16425	5.63083	-3.22911
4577	232	2829	479	64.94519	323.33159	52.49392	51.06031	5.20089	13.75370
4562	222	640	563	63.66482	329.57867	54.66899	56.53514	4.83760	-4.44844
4562	222	2673	376	63.49440	328.43145	52.87172	52.48216	4.98651	5.29311
4562	222	641	564	61.13342	316.89210	64.98653	41.40091	6.60434	9.03104
4562	222	1167	101	61.11954	316.83937	55.67219	64.53549	6.61237	8.64328
626	556	626	556	65.12077	300.44065	11.14063	11.14063	16.57750	16.57750

TABLE 17

```
    CHARLY-DYNAMIC CORE ALLOCATOR-
   599WORDS OF CORE HAVE BEEN
DISTRIBUTED AMONG THE FOLLOWING ARRAYS

           SNREV          8
           NSREV          6
           SUM1          90
           SUM2          18
           DELTA         18
           BBIAS         10
           KNSTR          2
           ARRAY        448



  STARTING ADDRESSES OF ARRAYS

   SNREV     127426000000
   NSREV     127436000000
   SUM1      127444000000
   SUM2      127576000000
   DELTA     127620000000
   BBIAS     127642000000
   KNSTR     127654000000
   ARRAY     127656000000
       DIV CHECK    AT LOCATION     024507
```

TABLE 18

*****BIASES*****

NORTH-TO-SOUTH REVS

REV NO	SEG NO	COEFFICIENTS
640	563	-0.8583354E 01
2658	360	0.7230325E 01
4236	15	-0.9595514E-15
4577	232	0.3065838E 01
4562	222	-0.1044951E 02
626	556	0.

SOUTH-TO-NORTH REVS

REV NO	SEG NO	COEFFICIENTS
640	563	-0.8583354E 01
2658	360	0.7230326E 01
2673	376	-0.1083907E 02
2829	479	0.1652231E 01
4236	15	0.
626	556	0.
641	564	-0.3403513E 02
1167	101	-0.1586218E 01

TABLE 19

REVOLUTIONS N/S	S/N	CROSSING PTS, LATITUDE (DEGREES)	LONGITUDE (DEGREES)	BIASES N/S (METERS)	S/N (METERS)	ORIGINAL SEA SURFACE HEIGHTS N/S (METERS)	S/N (METERS)	ADJUSTED SEA SURFACE HEIGHTS N/S (METERS)	S/N (METERS)	AVERAGE (ADJUSTED) (METERS)	RESIDUALS N/S (METERS)
640	640	65.134333	306.998838	-8.5834	-8.5834	22.4297	22.4297	31.0130	31.0130	31.0130	0.
2658	2658	65.129446	325.572426	7.2303	7.2303	60.8516	60.8516	53.6212	53.6212	53.6212	0.
2658	2829	65.046348	320.823555	7.2303	1.6522	50.1168	44.5387	42.8865	42.8865	42.8865	0.
4236	2829	65.043416	320.911968	-0.0000	1.6522	43.0020	44.6542	43.0020	43.0020	43.0020	0.
4236	4236	65.124931	325.384300	-0.0000	0.	51.9219	51.9219	51.9219	51.9219	51.9219	0.
4577	640	64.729195	319.712238	3.0858	-8.5834	44.8334	33.1643	41.7476	41.7476	41.7476	0.
4577	2829	64.945189	323.331593	3.0858	1.6522	52.4939	51.0603	49.4081	49.4081	49.4081	0.
4562	640	63.664815	329.578667	-10.4495	-8.5834	54.6690	56.5351	65.1185	65.1185	65.1185	0.
4562	2673	63.494395	328.431454	-10.4495	-10.8391	52.8717	52.4822	63.3212	63.3212	63.3212	0.
4562	641	61.133418	316.892101	-10.4495	-34.0351	64.9865	41.4009	75.4360	75.4360	75.4360	0.
4562	1167	61.119545	316.839371	-10.4495	-1.5862	55.6722	64.5355	66.1217	66.1217	66.1217	0.
626	626	65.120769	300.440651	0.	0.	11.1406	11.1406	11.1406	11.1406	11.1406	0.

MEAN OF RESIDUALS 0.

STANDARD DEVIATION 0.

TABLE 20

REVOLUTIONS S/N	REVOLUTIONS N/S	CROSSING PTS. LATITUDE (DEGREES)	CROSSING PTS. LONGITUDE (DEGREES)	BIASES S/N (METERS)	BIASES N/S (METERS)	A PRIORI SEA SURFACE HEIGHTS S/N (METERS)	A PRIORI SEA SURFACE HEIGHTS N/S (METERS)	ADJUSTED SEA SURFACE HEIGHTS S/N (METERS)	ADJUSTED SEA SURFACE HEIGHTS N/S (METERS)	AVERAGE (ADJUSTED) (METERS)	RESIDUALS S/N (METERS)
640	640	65.134333	306.998833	-8.5834	-8.5834	22.4297	22.4297	31.0130	31.0130	31.0130	0.
640	4577	64.729195	319.712238	-8.5834	3.0858	33.1643	44.8334	41.7476	41.7476	41.7476	0.
640	4562	63.664815	329.578667	-8.5834	-10.4495	56.5351	54.6690	65.1185	65.1185	65.1185	0.
2658	2658	65.129448	325.572426	7.2303	7.2303	60.8516	60.8516	53.6212	53.6212	53.6212	0.
2673	4562	63.494395	328.431454	-10.8391	-10.4495	52.4822	52.8717	63.3212	63.3212	63.3212	0.
2829	2658	65.046348	320.823555	1.6522	7.2303	44.5387	50.1168	42.8865	42.8865	42.8865	0.
2829	4236	65.043416	320.911968	1.6522	-0.0000	44.6542	43.0020	43.0020	43.0020	43.0020	0.
2829	4577	64.945139	323.331593	1.6522	3.0858	51.0603	52.4939	49.4081	49.4081	49.4081	0.
4236	4236	65.124931	325.384300	0.	-0.0000	51.9219	51.9219	51.9219	51.9219	51.9219	0.
626	626	65.120769	300.440651	0.	0.	11.1406	11.1406	11.1406	11.1406	11.1406	0.
641	4562	61.133418	316.892101	-34.0351	-10.4495	41.4009	64.9865	75.4360	75.4360	75.4360	0.
1167	4562	61.119545	316.839371	-1.5862	-10.4495	64.5355	55.6722	66.1217	66.1217	66.1217	0.

TABLE 21

FORMAT OF SEAHT OUTPUT TAPE

IEADER RECORD

WORD NO.	FORMAT	DESCRIPTION
1	I	Dummy (0)
2	I	Dummy (0)
3	I	No. of Data Records
4	I	Segment No.
5	I	Revolution No.

ATA RECORD

WORD NO.	FORMAT	DESCRIPTION
1	I	Time (milliseconds)
2	I	Latitude (milliseconds)
3	I	Longitude (milliseconds)
4	I	Sea Surface Height (micrometers)
5	I	Dummy (0)

TABLE 22

3.6 LIMITATIONS

Several dimensions have been set in SEAHT to values sufficiently high to accommodate all runs that are expected to be made by the program. The dimensions set produce only the following limitations:

- Number of North-South passes $\leq$ 200
- Number of South-North passes $\leq$ 200
- Number of data records per pass $\leq$ 2000 (≈33 minutes at the 1/second data rate)
- Number of Selects/Deletes from one data tape $\leq$ 50
- Total number of input data tapes $\leq$ 6.

Within limits, all the above dimensions could be significantly increased if needed, without increasing the total core used by the program. The arrays used in the parameter estimation portion of the program are all dynamically allocated, and so the number of passes that can be processed depends upon the number of parameters estimated per pass, and the amount of core that is available.

SEAHT will not function properly if:

- Some revolutions do not cross either a constrained revolution, or a revolution which crosses a constrained revolution.

There are no error terminations in SEAHT, and improperly setup jobs may be terminated either via system abort, or by meaningless program output.

SECTION 4.0
PROGRAM LISTING

A listing of the working version of the program follows. The placement of LINK cards is shown.

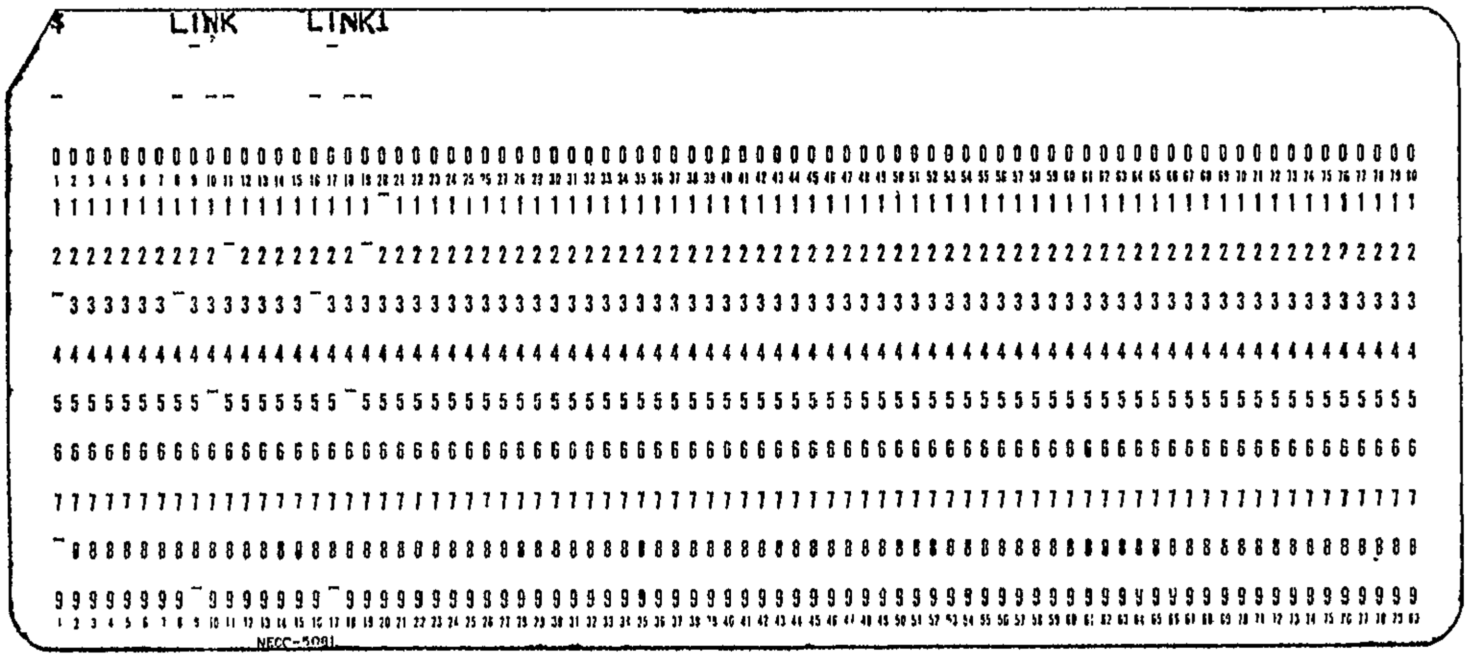
57345 01 03-16-77 16.995

```
      OSEAHT
            DIMENSION ICORE(8)
            COMMON/TAPES/INTP,IOUT,GTAP,NSTAPE,SNTAPE,SCRB,SCRC,DATAPE,DOUT
            COMMON/KONTRL/NOPAR,NONS,NOSN,IKNSTR,EDIT,MAX
            EQUIVALENCE(ICORE(1),ICSN),(ICORE(2),ICNS),(ICORE(3),NSUM1),
           .    (ICORE(4),NSUM2),(ICORE(5),NDEL),(ICORE(6),NP),
           .(ICORE(7),NKNSTR),(ICORE(8),IMAX)
            DATA NCORE/8/,IMORE/0/
            CALL LLINK(5HLINK1 )
            CALL SEARCH
            NP=NOPAR*(NOSN+1)
            NSUM1=NP*(NP+1)
            NSUM2=NP*2
            NDEL=NSUM2
            ICSN=NOSN
            ICNS=NONS
            NKNSTR=IKNSTR*2
            IMAX=MAX*4
            DO 10 I=1,NCORE
            IMORE=IMORE+ICORE(I)
         10 IF(MOD(ICORE(I),2).NE.0) ICORE(I)=ICORE(I)+1
            KOREB=320
            WRITE(6,5) IMORE,(ICORE(K),K=1,8)
          5 FORMAT('1',4X,'CHARLY-DYNAMIC CORE ALLOCATOR-',//,I6,
           .'WORDS OF CORE HAVE BEEN',/,'DISTRIBUTED AMONG THE FOLLOWING',
           .' ARRAYS',//,10X,'SNREV',5X,I6,/,10X,'NSREV',5X,I6,/,10X,'SUM1',
           .6X,I6,/,10X,'SUM2',6X,I6,/,10X,'DELTA',5X,I6,/,10X,'BBIAS',5X,I6,
           ./,10X,'KNSTR',5X,I6,/,10X,'ARRAY',5X,I6)
            CALL LLINK(5HLINK2 )
            CALL LLINK(5HLINK3 )
            CALL CHARLY(KOREB,IMORE,NCORE,ICORE,KOREG,$100)
            GO TO 102
        100 WRITE(5,101)
        101 FORMAT('1','CHARLY HAS BEEN UNSUCCESSFUL IN ALLOCATING CORE.')
        102 CALL LLINK(5HLINK4 )
            CALL SOLVE
            STOP
            END
```

57345 01 03-16-77 16.997

```
CBDATA
      BLOCK DATA
      INTEGER GTAP,DOUT
      COMMON/KONTRL/NPARC,NONS,NOSN,IKNSTR,EDIT,MAX
      COMMON/TAPES/INTP,IOUT,GTAP,ISCR(5),DOUT
      DATA INTP/5/,IOUT/6/,ISCR/11,12,13,14,15/,GTAP/18/,DOUT/10/
      DATA NPARC/3/,NONS/0/,NOSN/0/
      DATA IKNSTR/0/
      DATA EDIT/100./
      DATA MAX/0/
      END
```

```
      CNUMBER
            INTEGER FUNCTION NUMBER(K,IA,ID)
            DIMENSION IA(ID)
            NUMBER=0
            IF(ID.LT.1) RETURN
            DO 10 I=1,ID
            IF(K.EQ.IA(I)) GO TO 20
         10 CONTINUE
            RETURN
         20 NUMBER=I
            RETURN
            END
```


LINK LINK1
NECC-5081

```
      CSEARCH
            SUBROUTINE SEARCH
            EXTERNAL USERR
            LOGICAL MINUTE
            LOGICAL NODATA
            INTEGER DATA(8),SNTAPE,SCRB,REVNO,CDNAME(11),CARDID,TAPLIM,UNIQ
            INTEGER SNREV(200),NSREV(200),CRSNGS(5000,2),NONLEG(200,2),
               NDELI(2),NDELT(2),INTNOS(200,2)
            INTEGER DATAPE
            INTEGER SELECT(50,6),DELETE(50,6),KNO(6),KNA(6),ITAPE(6,2)
            REAL MINLAT,MAXLAT,MINLON,MAXLON
            DIMENSION WORDS(4),DT1(4),SSHT(2)
            DIMENSION KNSTR(5000,2)
            COMMON/KONTRL/NOPAR,NONS,NOSN,IKNSTR,EDIT,MAX
            COMMON/ALT/NI,ARRAY(4,1000),NJ,ARRY(4,1000)
            COMMON/TAPES/INTP,IOUT,GTAP,NSTAPE,SNTAPE,SCRB,SCRC,DATAPE,DOUT
            EQUIVALENCE (XCROSS,DATA(3)),(YCROSS,DATA(4)),(GJ,DATA(6)),
           ,    (TI,DATA(7)),(TJ,DATA(8)),(GI,DATA(5)),SSHT(1))
            EQUIVALENCE  (UNIQ,WORDS(4))
            DATA NODATA/.FALSE./
            DATA MINUTE/.TRUE./
            DATA MINUS/-1/
            DATA NDELT/2*0/,NCARD/11/,TAPLIM/6/,ITP1/0/,ITAPE/12*0/
            DATA CDNAME/6HAREA  ,6HPARAM ,6HNODATA,6HCONSTR,6HEDIT  ,6HDATA  ,
           ,6HSELECT,6HDELETE,6HVERTCL,6HZERO  ,6HMINUTE/
            DATA KNO/6*0/,KNA/6*0/
            DATA MINLAT/-90.0/
           ,        MAXLAT/90./,
           ,        MINLON/000./,
           ,        MAXLON/360./
            DATA ZERO/0.0/
            DATA IKNSTR/0/
            DATA EDIT1/1000./
            INTEGER R,C
         20 READ(INTR,1002,END=21) CARDID,IARRAY,DT1
       1002 FORMAT(A6,I4,4E15.5)
            I=NUMBER(CARDID,CDNAME,NCARD)
            IF(I.NE.6) GO TO 20
            ITP1=ITP1+1
            ITAPE(ITP1,1)=IARRAY
            K=DT1(1)+0.1
            ITAPE(ITP1,2)=MAX0(K,1)
```

```
          GO TO 20
       21 REWIND INTP
        6 READ(INTP,1002,END=50) CARDID,IARRAY,DT1
          I=NUMBER(CARDID,CDNAME,NCARD)
          GO TO (1,2,3,4,5,6,7,8,9,10,11),I
        1 MINLAT=DT1(1)
          MAXLAT=DT1(2)
          MINLON=DT1(3)
          MAXLON=DT1(4)
          GO TO 6
        2 NOPAR=IARRAY
          GO TO 6
        3 NODATA=.TRUE.
          IF(IARRAY.GT.0) SCRB=IARRAY
          READ(SCRB) NONS,NOSN
          GO TO 6
        4 IKNSTR=IKNSTR+1
          KNSTR(IKNSTR,1)=DT1(1)*100000.0+DT1(2)*10.0
          KNSTR(IKNSTR,2)=IARRAY
          GO TO 6
        5 EDIT=DT1(1)
          GO TO 6
        7 CALL SELDEL(ITP1,ITAPE,IARRAY,DT1,SELECT,KNO)
          GO TO 6
        8 CALL SELDEL(ITP1,ITAPE,IARRAY,DT1,DELETE,KNA)
          GO TO 6
        9 ISETV=111
          GO TO 6
       10 ZERO=DT1(1)
          WRITE(6,1003) ZERO
     1003 FORMAT(1X,'A ZERO OPTION USED WITH',E13.5,'LONGITUDE ZERO-SET')
          GO TO 6
       11 MINUTE=.FALSE.
          GO TO 6
       50 IF(NODATA) RETURN
          DO 55 I=1,ITP1
          IUNIT=ITAPE(I,1)
          NTPASS=ITAPE(I,2)
          CALL FLGRRC(IUNIT,USERR)
          DO 90 K=1,NTPASS
          READ(IUNIT,END=90) WORDS,REVNO
          REVNO=REVNO*100000+UNIQ*10
```

```
      IF(NUMBER(REVNO,SELECT(1,I),KNO(1)).GT.0) GO TO 60
      IF(KNO(I).NE.0) GO TO 61
      IF(NUMBER(REVNO,DELETE(1,I),KNA(1)).EQ.0) GO TO 60
   61 READ(IUNIT,END=90) WORDS
      GO TO 61
   60 NR=0
      IPRTL=0
      M=1
   65 READ(IUNIT,END=78) WORDS
      IF(WORDS(2).LT.MINLAT.OR.WORDS(2).GT.MAXLAT) GO TO 65
      IF(WORDS(3).LT.MINLON.OR.WORDS(3).GT.MAXLON) GO TO 65
      IF(MINUTE) WORDS(1)=WORDS(1)*1440.
      ARRAY(1,M)=WORDS(1)
      ARRAY(2,M)=WORDS(2)
      ARRAY(3,M)=WORDS(3)
      ARRAY(4,M)=WORDS(4)
      IF(M.LE.2) GO TO 74
      IF(DIFF*(ARRAY(2,M)-ARRAY(2,M-1)).GE.0.0) GO TO 75
      NR=NR+1
      BACKSPACE IUNIT
      IPRTL=1
      GO TO 79
   74 IF(M.EQ.2) DIFF=ARRAY(2,2)-ARRAY(2,1)
   75 NR=NR+1
      M=M+1
      IF(M.LE.1000) GO TO 65
   78 IF(NR.LE.0) GO TO 90
   79 IF(NR.LE.1.AND.IPRTL.GT.0) GO TO 60
      IF(NR.GT.MAX) MAX=NR
      IF(ISETV.EQ.111.AND.(ARRAY(3,2)-ARRAY(3,1)).GT.0.0000001) GO TO 80
      XS=ARRAY(2,2)-ARRAY(2,1)
      IF(XS.GT.0.0) GO TO 80
      NM=NM+1
      NSREV(NM)=REVNO
      WRITE(NSTAPE) REVNO,NR,ARRAY
      IF(IPRTL.GT.0) GO TO 60
      GO TO 90
   80 NN=NN+1
      SNREV(NN)=REVNO
      WRITE(SNTAPE) REVNO,NR,ARRAY
      IF(IPRTL.GT.0) GO TO 60
   90 CONTINUE
```

```
   55 CONTINUE
      ENDFILE NSTAPE
      ENDFILE SNTAPE
      REWIND NSTAPE
      WRITE(SCRB) NM,NN,NSREV,SNREV
      DO 200 M=1,NM
      READ(NSTAPE,END=200) REVNO,NI,ARRAY
      IF(ZERO.EQ.0.0) GO TO 56
      DO 57 L=1,NI
      IF((ARRAY(3,L)-ZERO).GE.0.0) GO TO 58
      ARRAY(3,L)=ARRAY(3,L)+360.0-ZERO
      GO TO 57
   58 ARRAY(3,L)=ARRAY(3,L)-ZERO
   57 CONTINUE
   56 REWIND SNTAPE
      DATA(2)=M
      EXLNGM=ARRAY(3,1)
      EXLATM=ARRAY(2,1)
      ENLNGM=ARRAY(3,NI)
      ENLATM=ARRAY(2,NI)
      DO 150 N=1,NN
      READ(SNTAPE,END=150) REVNO,NJ,ARRY
      IF(ZERO.EQ.0.0) GO TO 59
      DO 70 L=1,NJ
      IF((ARRY(3,L)-ZERO).GE.0.0) GO TO 69
      ARRY(3,L)=ARRY(3,L)+360.0-ZERO
      GO TO 70
   69 ARRY(3,L)=ARRY(3,L)-ZERO
   70 CONTINUE
   59 DATA(1)=N
      ENLATN=ARRY(2,1)
      ENLNGN=ARRY(3,1)
      EXLATN=ARRY(2,NJ)
      EXLNGN=ARRY(3,NJ)
      D7=(ENLNGM-ENLNGN)*(EXLNGM-EXLNGN)
      D8=(EXLATN-ENLATM)*(ENLATN-EXLATM)
      IF(D7.GT.0.0.OR.D8.GT.0.0) GO TO 150
      CALL CROSS(XCROSS,YCROSS,ENLATM,ENLNGM,EXLATM,EXLNGM,ENLATN,
         ENLNGN,EXLATN,EXLNGN,INDC)
      IF(INDC.EQ.1) GO TO 150
      CALL GHTIM(XCROSS,YCROSS,GI,GJ,TI,TJ)
      IF(ZERO.EQ.0.0) GO TO 72
```

```
      IF((YCROSS+ZERO).GT.360.0) GO TO 73
      YCROSS=YCROSS+ZERO
      GO TO 72
   73 YCROSS=YCROSS-360.0+ZERO
   72 IF(ABS(SSHT(1)).GT.EDIT1) GO TO 150
      RESID=ABS(SSHT(2)-SSHT(1))
      IF(RESID.GT.EDIT) GO TO 150
      WRITE(SCRB) DATA
      NDATA=NDATA+1
      CRSNGS(NDATA,1)=DATA(2)
      CRSNGS(NDATA,2)=DATA(1)
  150 CONTINUE
      DATA(1)=-1
      WRITE(SCRB) DATA
  200 CONTINUE
      ENDFILE SCRB
      WRITE(6,2000)
 2000 FORMAT('1',63X,'SEAHT',////,36X,
     1'LEAST SQUARES DETERMINATION OF SMOOTHED SEA SURFACE HEIGHTS',//,
     237X,'CALCULATED AT INTERSECTING POINTS OF CROSSING REVOLUTIONS')
      WRITE(6,3000) MINLAT,MAXLAT,MINLON,MAXLON
 3000 FORMAT(////,57X,'CALIBRATION AREA',/,51X,'LATITUDE ',F8.2,
     1'  TO',F8.2,/,51X,'LONGITUDE ',F8.2,'  TO',F8.2)
      WRITE(6,3001) NOPAR
 3001 FORMAT('0',53X,'NO.OF PARAMETERS   =   ',I1)
      WRITE(6,3003) NN
 3003 FORMAT(///,50X,'NO. OF SOUTH-TO-NORTH REVS   =   ',I2)
      WRITE(6,3002) NM
 3002 FORMAT('0',49X,'NO. OF NORTH-TO-SOUTH REVS   =   ',I2)
      WRITE(IOUT,1005)
      DO 67 I=1,NM
      INS=NSREV(I)/100000
      INSSEG=(NSREV(I)-INS*100000)/10
   67 WRITE(IOUT,1006) INS,INSSEG
      WRITE(IOUT,1007)
      DO 68 I=1,NN
      ISN=SNREV(I)/100000
      ISNSEG=(SNREV(I)-ISN*100000)/10
   68 WRITE(IOUT,1006) ISN,ISNSEG
 1005 FORMAT('1',50X,'NORTH-TO-SOUTH REVOLUTIONS',/,54X,
     ,'REV NO      SEG NO',/)
 1006 FORMAT(' ',54X,I5,6X,I5)
```

```
      1007 FORMAT('0',50X,'SOUTH-TO-NORTH REVOLUTIONS',/,54X,
          ,'REV NO        SEG NO',/)
           WRITE(6,2999)
           DO 3007 K=1,IKNSTR
           KNREV=KNSTR(K,1)/100000
           KNSEG=(KNSTR(K,1)-KNREV*100000)/10
           IF(KNSTR(K,2).EQ.1) WRITE(6,3004) KNREV,KNSEG
           IF(KNSTR(K,2).EQ.2) WRITE(6,3005) KNREV,KNSEG
           IF(KNSTR(K,2).EQ.3) WRITE(6,3006) KNREV,KNSEG
      3007 CONTINUE
      2999 FORMAT('1',47X,'***********CONSTRAINED REVS***********',///,43X,
          ,'REV NO',5X,'SEG NO',10X,'TYPE OF CONSTRAINT',/,65X,'HEIGHT',3X,
          ,'SLOPE',3X,'CURVATURE',//)
      3006 FORMAT(43X,I5,6X,I5,10X,'*',7X,'*',9X,'*')
      3005 FORMAT(43X,I5,6X,I5,10X,'*',7X,'*')
      3004 FORMAT(43X,I5,6X,I5,10X,'*')
           WRITE(IOUT,7000)
       210 WRITE(IOUT,7001)
           WRITE(IOUT,7002)
           WRITE(IOUT,7005)
           REWIND SORB
           READ(SORB) T1
       220 READ(SORB,END=240) DATA
           IF(DATA(1).EQ.MINUS) GO TO 220
           I1=DATA(2)
           I2=DATA(1)
           DATA(1)=NSREV(I1)
           DATA(2)=SNREV(I2)
           INS=DATA(1)/100000
           INSSEG=(DATA(1)-INS*100000)/10
           ISN=DATA(2)/100000
           ISNSEG=(DATA(2)-ISN*100000)/10
           WRITE(IOUT,7003)INS,INSSEG,ISN,ISNSEG,DATA(3),DATA(4),DATA(5),
          ,DATA(6),DATA(7),DATA(8)
           GO TO 220
       240 NPRINT=NPRINT+1
           IF(NPRINT.GT.1) RETURN
       300 NDELI(1)=0
           NDELI(2)=0
           II=1
           I1=2
           NLIM=NM
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      330 DO 350 I=1,NLIM
          IF(NUMBER(I,NONLEG(1,II),NDELT(II)).NE.0) GO TO 350
****W  1457 DO LOOP INDEX I MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
    C DETERMINE THE NUMBER OF CROSSINGS OF ITH NS(SN) ARC
          NCROSS=0
          DO 340 J=1,NDATA
          IF(CRSNGS(J,II).NE.I) GO TO 340
    C TEST TO SEE IF SN(NS) ARC DELETED.
          IF(NUMBER(CRSNGS(J,I1),NONLEG(1,I1),NDELT(I1)).EQ.0)
         ,NCROSS=NCROSS+1
      340 IF(NCROSS.GE.NJPAR) GO TO 350
          IF(NCROSS.EQ.0) GO TO 345
          IT=NSREV(I)
          IF(IT.EQ.2) IT=SNREV(I)
          IT=NUMBER(IT,KNSTR,IKNSTR)
          IF(IT.EQ.0) GO TO 345
          IT=(NCROSS+1)-KNSTR(IT,2)
          IF(IT.GE.0) GO TO 350
      345 NDELI(II)=NDELI(II)+1
          NDELT(II)=NDELT(II)+1
          J=NDELT(II)
          NONLEG(J,II)=I
      350 CONTINUE
          II=II+1
          I1=1
          NLIM=NN
          IF(II.LE.2) GO TO 330
          IF(NDELI(1).GT.0.OR.NDELI(2).GT.0) GO TO 300
          I2=MAX0(NM,NN)
          DO 410 I=1,I2
          DO 410 J=1,2
      410 INTNOS(I,J) = 1
          NLIM=NM
          DO 420 II=1,2
          ILIM=NDELT(II)
          IF(ILIM.EQ.0) GO TO 420
          DO 415 I=1,NLIM
      415 IF(NUMBER(I,NONLEG(1,II),ILIM).NE.0) INTNOS(I,II)=0
****W  1457 DO LOOP INDEX I MAY NOT BE REDEFINED IN CALL OR ABNORMAL FUNCTION
      420 NLIM=NN
          DO 430 I=1,NM
          IF(INTNOS(I,1).EQ.0) GO TO 430
```

```
            NONS=NONS+1
            NSREV(NONS)=NSREV(I)
            INTNOS(NONS,1)=INTNOS(I,1)
        430 CONTINUE
            DO 435 I=1,NN
            IF(INTNOS(I,2).EQ.0) GO TO 435
            NOSN=NOSN+1
            SNREV(NOSN)=SNREV(I)
            INTNOS(NOSN,2)=INTNOS(I,2)
        435 CONTINUE
            NOKNST=0
            DO 480 I=1,IKNSTR
            K=KNSTR(I,1)
            IJ=NUMBER(K,NSREV,NONS)
            IF(IJ.EQ.0) IJ=NUMBER(K,SNREV,NOSN)
            IF(IJ.EQ.0) GO TO 480
            NOKNST=NOKNST+1
            DO 475 J=1,2
        475 KNSTR(NOKNST,J)=KNSTR(I,J)
        480 CONTINUE
            IKNSTR=NOKNST
            REWIND SCRB
            WRITE(DATAPE) NONS,NOSN,(NSREV(I),I=1,NONS),(SNREV(I),I=1,NOSN),
           .((KNSTR(I,J),I=1,NOKNST),J=1,2)
            READ(SCRB) T
        510 NDATA=0
        550 READ(SCRB,END=600) DATA
            IF(DATA(1).EQ.MINUS) GO TO 590
            DATA(1)=NUMBER(DATA(1),INTNOS(1,2),NOSN)
            DATA(2)=NUMBER(DATA(2),INTNOS,NONS)
            IF(ABS(SSHT(1)).GT.EDIT1) GO TO 550
            NDATA=NDATA+1
            WRITE(DATAPE) DATA
            GO TO 550
        590 IF(NDATA.GT.0) WRITE(DATAPE)DATA
            GO TO 510
        600 ENDFILE DATAPE
            SCRB=DATAPE
            WRITE(IOUT,7004)
            GO TO 210
       6000 FORMAT('1',15X,'THE CONSTRAINED ARC',I7,'HAS BEEN DELETED')
       7000 FORMAT(1H1,30X,'ARC CROSSING DATA INPUT FOR THIS RUN')
```

```
      7001 FORMAT('0',7X,'REVOLUTION NO',12X,'CROSSING POINTS',6X,
          1'SMOOTHED SEA SURFACE HEIGHT',8X,'TIME FROM EPOCH')
      7002 FORMAT(' ',2X,'NORTH-SOUTH  SOUTH-NORTH',5X,'LATITUDE  LONGITUDE',
          24X,'NORTH-SOUTH  SOUTH-NORTH',6X,'NORTH-SOUTH  SOUTH-NORTH')
      7003 FORMAT(' ',2X,I5,' ',I5,2X,I5,' ',I5,5X,F10.5,1X,F10.5,3X,F10.5,
          ,3X,F10.5,5X,F10.5,4X,F10.5)
      7004 FORMAT(1H1,30X,'ARC CROSSING DATA USED FOR ADJUSTMENTS')
      7005 FORMAT(' ',3X,'(REV/SEG)',4X,'(REV/SEG)',6X,'(DEGREES)',1X,
          ,'(DEGREES)',6X,'(METERS)',5X,'(METERS)',8X,'(MINUTES)',4X,
          ,'(MINUTES)')
           END
****W  1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS
****W     7 MEMORY EXPANDED, USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```

57345 01 03-16-77   17,004

```
      CSELDEL
            SUBROUTINE SELDEL(NOTPS,ITAPS,N,ARRAY,SELECT,NSEL)
            INTEGER SELECT(50,1),ITAPS(1),NSEL(1)
            DIMENSION ARRAY(1)
            IST=1
            ILIM=NOTPS
            IF(N.EQ.0) GO TO 30
            IST=NUMBER(N,ITAPS,NOTPS)
            IF(IST.EQ.0) RETURN
         25 ILIM=IST
         30 DO 50 IT=IST,ILIM
            K=NSEL(IT)
            DO 40 J=1,3,2
            ISEL=ARRAY(J)*100000.0+ARRAY(J+1)*1000.0+0.1
            IF(ISEL.EQ.0) GO TO 40
            K=K+1
            SELECT(K,IT)=ISEL
         40 CONTINUE
            NSEL(IT)=K
         50 CONTINUE
            RETURN
            END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GCROSS
          SUBROUTINE BROSS(XCROSS,YCROSS,ANLATM,ANLNGM,AXLATM,AXLNGM,ENLATN,
         ,ENLNGN,EXLATN,EXLNGN,INDC)
          COMMON/ALT/ NI,ARRAY(4,1000),NU,ARRY(4,1000)
          ENLATM=AXLATM
          ENLNGM=AXLNGM
          EXLATM=ANLATM
          EXLNGM=ANLNGM
          INDC=0
          IF(EXLATM.EQ.ENLATN.AND.EXLNGM.EQ.ENLNGN) GO TO 501
          IF(EXLATN.EQ.ENLATM.AND.EXLNGN.EQ.ENLNGM) GO TO 500
          IF(ENLATM.LT.EXLATN) GO TO 10
          GO TO 20
        5 IF(EXLATM.GT.ENLATN) GO TO 30
          GO TO 40
       10 DO 11 I=1,NU-1
       11 IF(ENLATM.GE.ARRY(2,I).AND.ENLATM.LE.ARRY(2,I+1))
         ,GO TO 15
          GO TO 99
       15 CALL YFIND(ENLATM,ARRY(2,I),ARRY(2,I+1),ARRY(3,I),ARRY(3,I+1),Y)
          IF(Y.GT.ENLNGM) GO TO 99
          GO TO 5
       20 DO 21 I=1,NI-1
       21 IF(EXLATN.LE.ARRAY(2,I).AND.EXLATN.GE.ARRAY(2,I+1))
         ,GO TO 25
          GO TO 99
       25 CALL YFIND(EXLATN,ARRAY(2,I),ARRAY(2,I+1),ARRAY(3,I),ARRAY(3,I+1),
         ,Y)
          IF(Y.LT.EXLNGN) GO TO 99
          GO TO 5
       30 DO 31 I=1,NU-1
       31 IF(EXLATM.GE.ARRY(2,I).AND.EXLATM.LE.ARRY(2,I+1)) GO TO 35
          GO TO 99
       35 CALL YFIND(EXLATM,ARRY(2,I),ARRY(2,I+1),ARRY(3,I),ARRY(3,I+1),Y)
          IF(Y.LT.EXLNGM) GO TO 99
          GO TO 6
       40 DO 41 I=1,NI-1
       41 IF(ENLATN.LE.ARRAY(2,I).AND.ENLATN.GE.ARRAY(2,I+1)) GO TO 45
          GO TO 99
       45 CALL YFIND(ENLATN,ARRAY(2,I),ARRAY(2,I+1),ARRAY(3,I),ARRAY(3,I+1),
         ,Y)
          IF(Y.GT.ENLNGN) GO TO 99
```

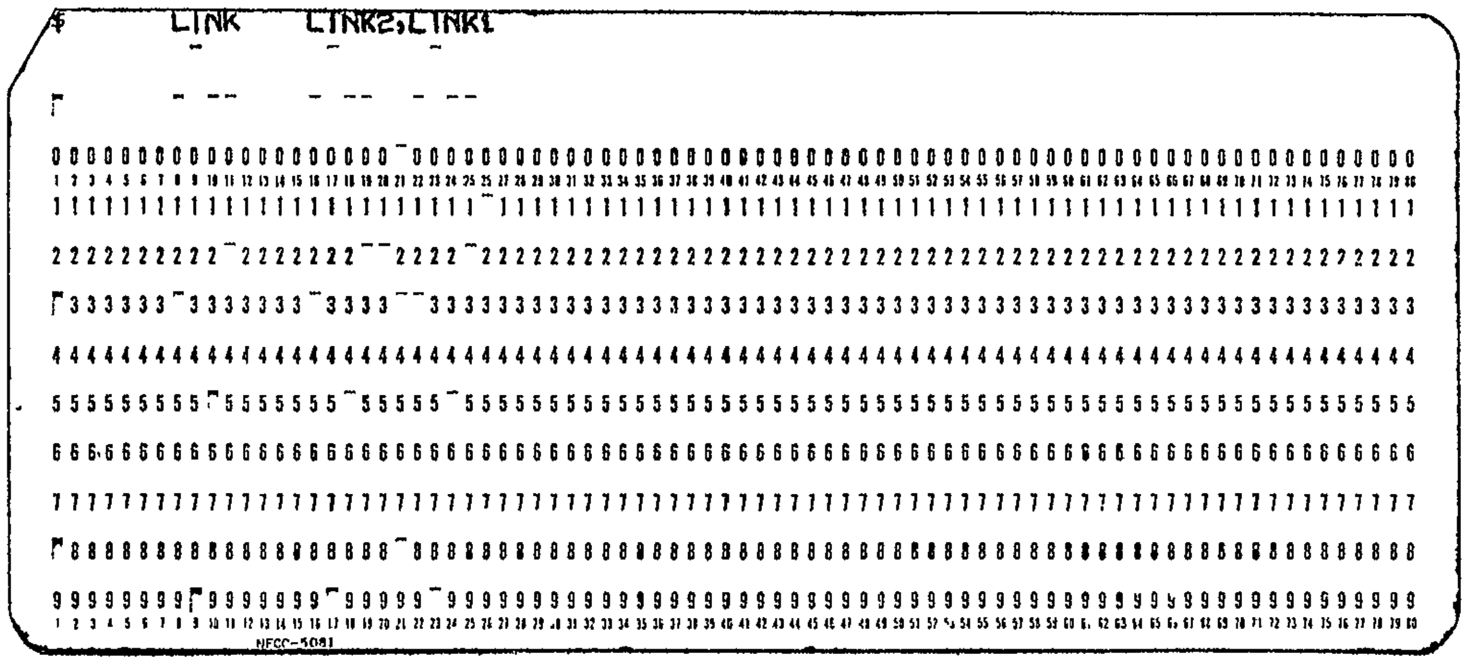
```
      6 KOUNT=0
      1 SLPNS=(ENLNGM-EXLNGM)/(ENLATM-EXLATM)
        SLPSN=(ENLNGN-EXLNGN)/(ENLATN-EXLATN)
        CNS=(ENLNGM-SLPNS*ENLATM)
        CSN=(ENLNGN-SLPSN*ENLATN)
        XINT=(CSN-CNS)/(SLPNS-SLPSN)
        YINT=(SLPSN*CNS-SLPNS*CSN)/(SLPSN-SLPNS)
        NOLAT=0
        KOUNT=KOUNT+1
        IF(KOUNT.GT.10) GO TO 99
        DO 100 I=1,NI-1
    100 IF(YINT.LE.ARRAY(3,I).AND.YINT.GE.ARRAY(3,I+1)) GO TO 101
        IF(KOUNT.GT.10) GO TO 99
        NOLAT=1
        GO TO 199
    101 CALL YFIND(YINT,ARRAY(3,I),ARRAY(3,I+1),ARRAY(2,I),ARRAY(2,I+1),
       .XNS)
    199 DO 200 I=1,NJ-1
    200 IF(YINT.LE.ARRY(3,I).AND.YINT.GE.ARRY(3,I+1)) GO TO 201
        IF(KOUNT.GT.10) GO TO 99
        NOLAT=2
        GO TO 299
    201 CALL YFIND(YINT,ARRY(3,I),ARRY(3,I+1),ARRY(2,I),ARRY(2,I+1),XSN)
    299 IF(NOLAT.EQ.1) GO TO 305
        IF(NOLAT.EQ.2) GO TO 310
        IF(ABS(XNS-XSN).LE.0.0001) GO TO 400
        IF(XNS.LT.XSN) GO TO 300
        ENLATN=XSN
        ENLNGN=YINT
        ENLATM=XNS
        ENLNGM=YINT
        GO TO 1
    300 EXLATN=XSN
        EXLNGN=YINT
        EXLATM=XNS
        EXLNGM=YINT
        GO TO 1
    305 IF(XSN.GE.XINT) GO TO 307
        ENLATN=XSN
        ENLNGN=YINT
        GO TO 1
    307 EXLATN=XSN
```

```
                EYLNGN=YINT
                GO TO 1
            310 IF(XNS.GE.XINT) GO TO 311
                EYLATM=XNS
                EXLNGM=YINT
                GO TO 1
            311 ENLATM=XNS
                ENLNGM=YINT
                GO TO 1
            400 XCROSS=(XNS+XSN)/2.0
                YCROSS=YINT
                RETURN
             99 INDC=1
                RETURN
            500 XCROSS=EXLATN
                YCROSS=EXLNGN
                RETURN
            501 XCROSS=EXLATM
                YCROSS=EXLNGM
                RETURN
                END
*****W  1475 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS
```

57345 01 03-16-77 17.007

```
      SUBROUTINE YFIND(X,TWOI,TWOI1,THREEI,THREEI1,THREE)
      DX=(X-TWOI)/(TWOI1-TWOI)
      DY=THREEI1-THREEI
      ADDTOY=DX*DY
      THREE=THREEI+ADDTOY
      RETURN
      END
```

```
      CGHTIM
            SUBROUTINE GHTIM(XCROSS,YCROSS,GI,GJ,TI,TJ)
            COMMON/ALT/NI,ARRAY(4,1000),NJ,ARRY(4,1000)
            IF(ARRAY(2,NI).EQ.XCROSS) GO TO 57
            IF(ARRAY(2,1).EQ.XCROSS) GO TO 58
            NE = NI-1
            DO 49 IN=1,NE
            IF((ARRAY(2,IN)-XCROSS)*(ARRAY(2,IN+1)-XCROSS).LT.0.) GO TO 55
   49       CONTINUE
   55       A=XCROSS-ARRAY(2,IN)
            B=ARRAY(2,IN+1)-ARRAY(2,IN)
            C=ARRAY(4,IN+1)-ARRAY(4,IN)
            D=ARRAY(1,IN+1)-ARRAY(1,IN)
            GI=((A*C)/B)+ARRAY(4,IN)
            TI=((A*D)/B)+ARRAY(1,IN)
            NA = NJ-1
            DO 53 IN=1,NA
            IF((ARRY(2,IN)-XCROSS)*(ARRY(2,IN+1)-XCROSS).LT.0.) GO TO 42
   53       CONTINUE
   42       A=XCROSS-ARRY(2,IN)
            B=ARRY(2,IN+1)-ARRY(2,IN)
            C=ARRY(4,IN+1)-ARRY(4,IN)
            D=ARRY(1,IN+1)-ARRY(1,IN)
            GJ= (A*C/B)+ARRY(4,IN)
            TJ=((A*D)/B)+ARRY(1,IN)
            RETURN
        57  GI=ARRAY(4,NI)
            TI=ARRAY(1,NI)
            GJ=GI
            TJ=TI
            RETURN
        58  GI=ARRAY(4,1)
            TI=ARRAY(1,1)
            GJ=GI
            TJ=TI
            RETURN
            END
*****W  1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS
```


$ LINK LINK2,LINK1

```
CSOLVE
      SUBROUTINE SOLVEI(IADDR,SNREV,NSREV,SUM1,SUM2,DELTA,BBIAS,KNSTR,
     .                   ARRAY)
      DIMENSION ARRAY(4,1)
      DIMENSION SIGMA(2),BBIAS(1),DATA(8),SSHT(5)
C     COMMON= SOUTH - NORTH
      DOUBLE PRECISION SUM1(1),SUM2(1),DELTA(1),ARCPAR(6),COMPAR(3)
      INTEGER SNTAPE
      INTEGER ARCNOS(6),GTAP,SCRB,SNREV(1),NSREV(1),IADDR(1),KNSTR(1)
      INTEGER Q,R
      INTEGER REVNO,DATAPE,DOUT
      INTEGER SCRB
      INTEGER DUM1,DUM2,DUM3
      DATA DUM1/0/,DUM2/0/,DUM3/0/
      COMMON/TAPES/INTP,IOUT,GTAP,NSTAPE,SNTAPE,SCRB,SCRC,DATAPE,DOUT
      COMMON/KONTRL/NOPAR,NONS,NOSN,IKNSTR,EDIT,MAX
      EQUIVALENCE (ISN,DATA(1)),(INS,DATA(2))
      DATA ARCNOS/1,2,3,3*0/
      DATA SIGMA/2*1.0/
      DATA AVE/0.0/,SDEV/0.0/,KOUNT/0/
      DATA INTPOS/0/
      WRITE(IOUT,2001) (IADDR(I),I=1,8)
 2001 FORMAT(/////,1X,'STARTING ADDRESSES OF ARRAYS',//,3X,'SNREV',5X,
     ,O12,/,3X,'NSREV',5X,O12,/,3X,'SUM1',6X,O12,/,3X,'SUM2',6X,O12,/,
     ,3X,'DELTA',5X,O12,/,3X,'BBIAS',5X,O12,/,3X,'KNSTR',5X,O12,
     ,/,3X,'ARRAY',5X,O12)
      RETURN
      ENTRY SOLVE
      JTAP=19
      REWIND SCRB
      IJK=2*IKNSTR
      READ(SCRB) I1,I2,(NSREV(I),I=1,NONS),(SNREV(I),I=1,NOSN),
     ,(KNSTR(I),I=1,IJK)
      NDIM=NOPAR*(NOSN+1)
      CALL CLEAR(BBIAS,NDIM)
      CALL ESTIM1(SUM1,SUM2,DELTA,BBIAS,ARCNOS,ARCPAR,NOPAR,NOSN)
      ARCPAR(1)=1.0D0
      COMPAR(1)=-1.0D0
  100 READ(SCRB,END=200) DATA
      IF(ISN.LT.0) GO TO 150
      IARC=INS
      ARCPAR(2)=DATA(7)
```

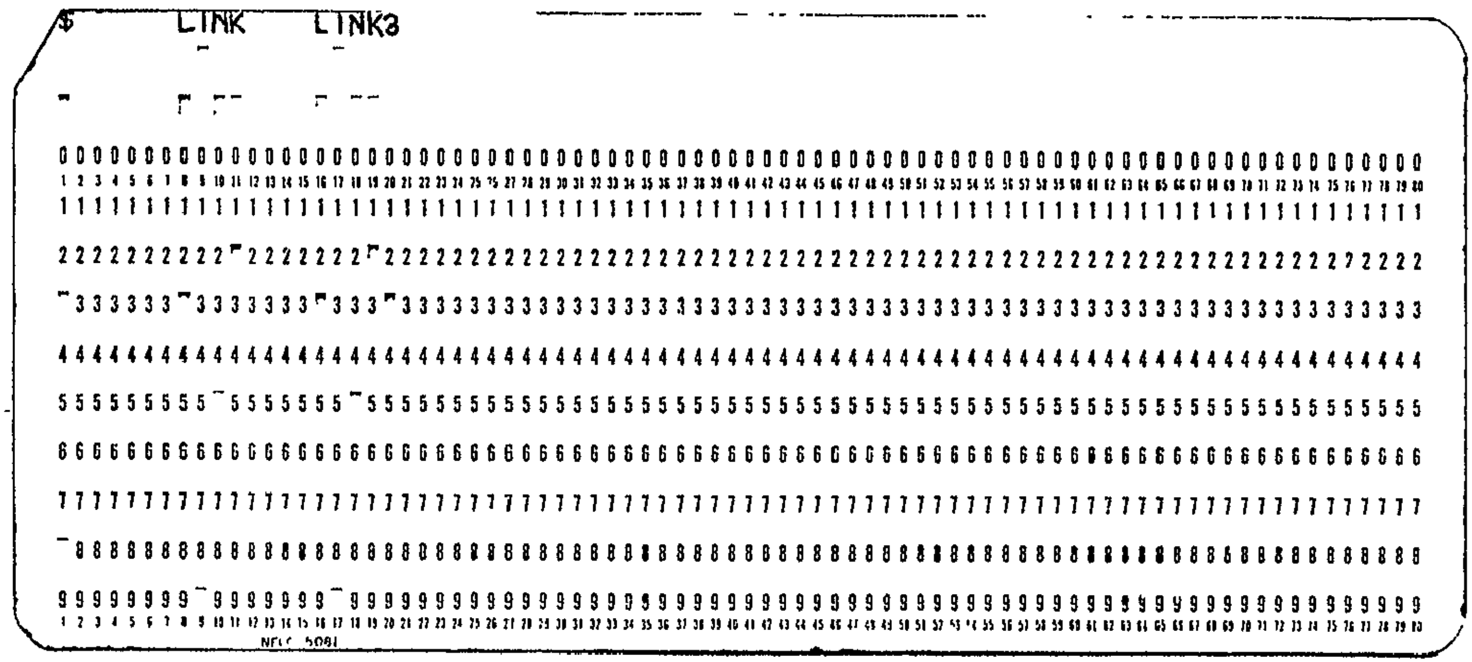
C-2

```
      ARCPAR(3)=DATA(7)**2
      COMPAR(2)=-DATA(8)
      COMPAR(3)=-DATA(8)**2
      RESID=DATA(5)-DATA(6)
      IF(ABS(RESID).LE.EDIT)
     .CALL ESTIM(NOPAR+ISN,COMPAR,SIGMA,RESID)
      GO TO 100
  150 ICON=0
      IJK=NUMBER(NSREV(IARC),KNSTR,IKNSTR)
      IF(IJK.GT.0) ICON=KNSTR(IJK+IKNSTR)
      CALL ESTIMA(ICON)
      CALL CLEAR(BBIAS,NOPAR)
      GO TO 100
  200 CALL ESTIMC(SNREV,KNSTR,IKNSTR)
      DO 500 KSN=1,NOSN
      REWIND SORB
      READ(SORB) T1
      IBC=NOPAR+1
      IF(KSN.NE.1) GO TO 501
      WRITE(IOUT,11)
      WRITE(GTAP,15)
      WRITE(JTAP,15)
      WRITE(GTAP,16)
      WRITE(JTAP,16)
      WRITE(GTAP,17)
      WRITE(JTAP,20)
      WRITE(GTAP,19)
      WRITE(JTAP,19)
      REWIND NSTAPE
  501 CONTINUE
      DO 400 IARC=1,NONS
      CALL UPDATE
      IF(KSN.NE.1) GO TO 502
      IREV=NSREV(IARC)/100000
      ISEG=(NSREV(IARC)-IREV*100000)/10
      WRITE(IOUT,1010) IREV,ISEG,(BBIAS(I),I=1,NOPAR)
  205 READ(NSTAPE) REVNO,NI,((ARRAY(C,R),C=1,4),R=1,NI)
      IF(REVNO.NE.NSREV(IARC)) GO TO 205
      WRITE(DOUT) DUM1,DUM2,NI,ISEG,IREV
      DO 207 K=1,NI
      ARRAY(1,K)=ARRAY(1,K)*60000.
      ARRAY(2,K)=ARRAY(2,K)*3600000.
```

```
      ARRAY(3,K)=ARRAY(3,K)*3600000.
      ITIME=ARRAY(1,K)
      ILAT=ARRAY(2,K)
      ILON=ARRAY(3,K)
      GO TO (208,209,210),NOPAR
  208 ARRAY(4,K)=ARRAY(4,K)-BBIAS(1)
      GO TO 206
  209 ARRAY(4,K)=ARRAY(4,K)-(BBIAS(2)*ARRAY(1,K)+BBIAS(1))
      GO TO 206
  210 XBBIAS=BBIAS(3)*ARRAY(1,K)**2.+BBIAS(2)*ARRAY(1,K)+BBIAS(1)
      ARRAY(4,K)=ARRAY(4,K)-XBBIAS
  206 ARRAY(4,K)=ARRAY(4,K)*1000000.
      ISEAHT=ARRAY(4,K)
  207 WRITE(DOUT) ITIME,ILAT,ILON,ISEAHT,DUM3
      ENDFILE DOUT
  502 CONTINUE
  300 READ(SCRB,END=410) DATA
      IF(ISN.EQ.-1) GO TO 400
      IF(KSN.NE.1.AND.ISN.NE.KSN) GO TO 300
      I2=IBC+NOPAR*(ISN-1)
      SSHT(1)=DATA(5)
      SSHT(2)=DATA(6)
      DO 310 I=1,NOPAR
      SSHT(1)=SSHT(1)-BBIAS(I)*DATA(7)**(I-1)
      SSHT(2)=SSHT(2)-BBIAS(I2)*DATA(8)**(I-1)
      I2=I2+1
  310 CONTINUE
      SSHT(3)=(SSHT(1)+SSHT(2))/2.0
      SSHT(4)=SSHT(1)-SSHT(3)
      SSHT(5)=SSHT(2)-SSHT(3)
      IF(ABS(DATA(5)-DATA(6)).GT.EDIT) GO TO 330
      AVE=AVE+SSHT(4)
      SDEV=SDEV+SSHT(4)**2
      KOUNT=KOUNT+1
  330 IF(KSN.EQ.1) WRITE(GTAP,1000)NSREV(INS)/100000,SNREV(ISN)/100000,
     ,DATA(3),DATA(4),BBIAS(1),BBIAS(I2-NOPAR),DATA(5),DATA(6),
     ,(SSHT(K),K=1,4)
      IF(ISN.EQ.KSN)WRITE(JTAP,1000)SNREV(ISN)/100000,NSREV(INS)/100000,
     ,DATA(3),DATA(4),BBIAS(I2-NOPAR),BBIAS(1),DATA(6),DATA(5),SSHT(2),
     ,SSHT(1),SSHT(3),SSHT(5)
      GO TO 300
  400 CONTINUE
```

```
      410 CONTINUE
      500 REWIND SORC
          WRITE(IOUT,28)
          REWIND SNTAPE
          DO 420 I=1,NOSN
          IE=IBC+NOPAR-1
          IREV=SNREV(I)/100000
          ISEG=(SNREV(I)-IREV*100000)/10
          WRITE(IOUT,1010) IREV,ISEG,(BBIAS(J),J=IBC,IE)
      405 READ(SNTAPE) REVNO,NJ,((ARRAY(C,R),C=1,4),R=1,NJ)
          IF(REVNO.NE.SNREV(I)) GO TO 405
          WRITE(DOUT) DUM1,DUM2,NJ,ISEG,IREV
          DO 407 K=1,NJ
          ARRAY(1,K)=ARRAY(1,K)*60000.
          ARRAY(2,K)=ARRAY(2,K)*3600000.
          ARRAY(3,K)=ARRAY(3,K)*3600000.
          ITIME=ARRAY(1,K)
          ILAT=ARRAY(2,K)
          ILON=ARRAY(3,K)
          GO TO (408,409,411),NOPAR
      408 ARRAY(4,K)=ARRAY(4,K)-BBIAS(IBC)
          GO TO 406
      409 ARRAY(4,K)=ARRAY(4,K)-(BBIAS(IBC+1)*ARRAY(1,K)+BBIAS(IBC))
          GO TO 406
      411 YBIAS=BBIAS(IBC+2)*ARRAY(1,K)**2.+BBIAS(IBC+1)*ARRAY(1,K)
          YBIAS=YBIAS+BBIAS(IBC)
          ARRAY(4,K)=ARRAY(4,K)-YBIAS
      406 ARRAY(4,K)=ARRAY(4,K)*1000000.
          ISEAHT=ARRAY(4,K)
      407 WRITE(DOUT) ITIME,ILAT,ILON,ISEAHT,DUM3
          ENDFILE DOUT
          IBC=IBC+NOPAR
      420 CONTINUE
          AVE=AVE/KOUNT
          SDEV=SDEV-AVE**2
          SDEV=SQRT(SDEV/KOUNT)
          WRITE(GTAP,28) AVE
       28 FORMAT('0','MEAN OF RESIDUALS',5X,F10.5)
          WRITE(GTAP,29) SDEV
       29 FORMAT('0','STANDARD DEVIATION',5X,F10.5)
       11 FORMAT('1',17X,'*****BIASES*****',/,16X,'NORTH-TO-SOUTH REVS',/,
         ,'REV NO',5X,'SEG NO',11X,'COEFFICIENTS')
```

```
         15 FORMAT('1',53X,'A PRIORI',13X,'ADJUSTED')
         16 FORMAT(1X,'REVOLUTIONS',6X,'CROSSING PTS',12X,'BIASES',9X,
           ,'SEA SURFACE HEIGHTS',2X,'SEA SURFACE HEIGHTS',3X,'AVERAGE',3X,
           ,'RESIDUALS')
         17 FORMAT(2X,'N/S',3X,'S/N',4X,'LATITUDE  LONGITUDE',6X,'N/S',7X,
           ,'N/S',7X,'S/N',7X,'N/S',8X,'S/N',7X,'N/S',5X,'(ADJUSTED)',4X,
           ,'N/S')
         18 FORMAT('0',15X,'SOUTH-TO-NORTH REVS',/,'REV NO',5X,'SEG NO',11X,
           ,'COEFFICIENTS')
         19 FORMAT(14X,'(DEGREES)',2X,'(DEGREES)',4X,'(METERS)',2X,
           ,'(METERS)',3X,'(METERS)',2X,'(METERS)',3X,'(METERS)',2X,
           ,'(METERS)',3X,'(METERS)',3X,'(METERS)')
         20 FORMAT(2X,'S/N',3X,'N/S',4X,'LATITUDE  LONGITUDE',6X,'S/N',7X,
           ,'S/N',7X,'N/S',7X,'S/N',8X,'N/S',7X,'S/N',5X,'(ADJUSTED)',4X,
           ,'S/N')
       1000 FORMAT(/,2(1X,I5),1X,2(1X,F10.6),2X,2(1X,F9.4),3(1X,2(1X,F9.4)))
       1010 FORMAT(' ',I4,7X,I4,10X,E17.7,5X,E17.7,5X,E17.7)
            STOP
            END
86 ****W    7 MEMORY EXPANDED, USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```


$ LINK LINK3

```
        LBL     CHARLY,SEAHT SUBROUTINE CHARLY                                CHARLY01
        TTL     SUBROUTINE CHARLY - DYNAMIC MEMORY ALLOCATOR                  CHARLY02
****** START OF DOCUMENTATION COMMENT CARDS ******************************CHARLY03
*                                                                        *CHARLY04
*           PROGRAM TITLE - SUBROUTINE CHARLY                            *CHARLY05
*           PROGRAM NO.   - 112293                                       *CHARLY06
*       WRITTEN BY DAN CHIN (WOLF RESEARCH AND DEVELOPMENT)              *CHARLY07
*           REWRITTEN BY TOM HARMON (NASA WALLOPS-AMS) 1/74              *CHARLY08
*           MODIFIED BY TOM NORTHAM (WRDC) 5/74, 6/74                    *CHARLY09
*                                                                        *CHARLY10
*       HARDWARE/SOFTWARE SUMMARY                                        *CHARLY11
*                                                                        *CHARLY12
*           COMPUTER REQUIRED - HONEYWELL 600/6000 SERIES                *CHARLY13
*           SYSTEM EXECUTIVE  - GECOS III                                *CHARLY14
*           PROGRAM LANGUAGE  - GMAP                                     *CHARLY15
*           PERIPHERALS       - NONE                                     *CHARLY16
*                                                                        *CHARLY17
*       PURPOSE                                                          *CHARLY18
*                                                                        *CHARLY19
*           THIS SUBROUTINE DYNAMICALLY ALLOCATES ARRAY STORAGE FOR USE *CHARLY20
*           WITH THE SEAHT PROGRAM.                                      *CHARLY21
*                                                                        *CHARLY22
*       METHOD                                                           *CHARLY23
*                                                                        *CHARLY24
*           WORD 31 (DECIMAL) IN THE SLAVE PROGRAM PREFIX IS SCANNED     *CHARLY25
*           TO FIND THE LOCATION OF AVAILABLE MEMORY FOR ARRAY ALLOCATIONCHARLY26
*           IF SUFFICIENT MEMORY ISN'T AVAILABLE, THEN AN ATTEMPT TO     *CHARLY27
*           GEMORE MORE MEMORY IS MADE.  IF GCOS DENIES THE REQUEST,     *CHARLY28
*           THEN CHARLY RETURNS TO THE NON STANDARD ERROR EXIT,          *CHARLY29
*           OTHERWISE, THE ARRAY LOCATIONS ARE STORED INTO THE SOLVEI    *CHARLY30
*           CALLING SEQUENCE AND A CALL TO SOLVEI IS EXECUTED.           *CHARLY31
*                                                                        *CHARLY32
*       CALLING SEQUENCE                                                 *CHARLY33
*                                                                        *CHARLY34
*           CALL CHARLY(KOREB,IMORE,NCORE,ICORE,KOREG,$NNN)              *CHARLY35
*                                                                        *CHARLY36
*               KOREB = THE AMOUNT OF CORE THAT MUST BE AVAILABLE FOR    *CHARLY37
*                       BUFFERS AFTER CHARLY HAS BEEN CALLED.            *CHARLY38
*               IMORE = THE SUM OF ARRAY SIZES,                          *CHARLY39
*               NCORE = NUMBER OF ARRAYS FOR WHICH CORE MUST BE          *CHARLY40
*                       ALLOCATED.                                       *CHARLY41
*               ICORE = NUMBER OF WORDS OF CORE REQUIRED BY EACH ARRAY, *CHARLY42
```

```
*            KOREG = THE AMOUNT OF CORE LEFT IN USER JOB SEGMENT      *CHARLY43
*                    IF POSITIVE, THE AMOUNT OF CORE TO BE GEMORE-D   *CHARLY44
*                    IF NEGATIVE,                                     *CHARLY45
*            $NNN  = RETURN WHEN CORE ALLOCATION NOT SUCESSFUL        *CHARLY46
*                                                                     *CHARLY47
*                                                                     *CHARLY48
*      RETURNS                                                        *CHARLY49
*                                                                     *CHARLY50
*         NORMAL ONLY (TERMINATION OCCURS IN CONSOL                   *CHARLY51
*         ON INSUFFICIENT MEMORY AVAILABLE)                           *CHARLY52
*                                                                     *CHARLY53
*      INPUT                                                          *CHARLY54
*                                                                     *CHARLY55
*         VIA CALLING SEQUENCE                                        *CHARLY56
*                                                                     *CHARLY57
*      OUTPUT                                                         *CHARLY58
*                                                                     *CHARLY59
*         CHARLY GENERATES A SIMULATED CALL SOLVEI(ARG1,ARG2,...)     *CHARLY60
*         WHERE ARG1 POINTS TO THE ADDRESS OF ARG2, AND ARG2 THRU ARGN*CHARLY61
*         CONTAIN THE ADDRESSES OF THE DYNAMICALLY ALLOCATED ARRAYS.  *CHARLY62
*                                                                     *CHARLY63
*      SUBPROGRAMS REQUIRED                                           *CHARLY64
*                                                                     *CHARLY65
*         SOLVEI                                                      *CHARLY66
*                                                                     *CHARLY67
*      CALLED BY                                                      *CHARLY68
*                                                                     *CHARLY69
*         SEAMT MAIN PROGRAM                                          *CHARLY70
*                                                                     *CHARLY71
*      EXTERNAL REFERENCES                                            *CHARLY72
*                                                                     *CHARLY73
*         NONE                                                        *CHARLY74
*      BIBLIOGRAPHY                                                   *CHARLY75
*                                                                     *CHARLY76
*                                                                     *CHARLY77
*                                                                     *CHARLY78
*                                                                     *CHARLY79
****** END OF DOCUMENTATION COMMENT CARDS *****************************CHARLY80
```

```
          000008      82 CHARLY SAVE                                                                  CHARLY82
000002710000     010
000217630000     010
000217754000     010
000217741000     010
000002 7200 31   000     83              LXLO    2,1*             ACCESS KOREF                        CHARLY83
000044 7400 00   010     84              STXO    X                STORE IN UPPER HALF OF INSTR AT LOC X CHARLY84
000007 5500 00   010     85              SBAR    *+1                                                  CHARLY85
000000 2350 07   000     86              LDA     **,DL              GET BAR IN AL                     CHARLY86
000777 3750 07   000     87              ANA     =0777,DL           NO. OF 512 WORD BLOCKS            CHARLY87
000011 7350 00   000     88              ALS     9                  MULTIPLY BY 512                   CHARLY88
000001 1750 07   000     89              SBA     1,DL               UPPER CORE ADDRESS                CHARLY89
000037 7510 07   000     90              STCA    31,07                                                CHARLY90
000004 2350 31   000     91              LDA     4,1*             NO. OF ELEMENTS                     CHARLY91
000006 7350 00   000     92              ALS     6                                                    CHARLY92
000153 7510 06   010     93              STCA    OTTALY+1,06                                          CHARLY93
000153 2350 00   010     94              LDA     OTTALY+1           INITIALIZE WORKING TALLY WORD     CHARLY94
000152 7550 00   010     95              STA     OTTALY                                               CHARLY95
000005 6200 31   000     96              EAXO    5,1*             ARRAY ADDRESS                       CHARLY96
00 151 7400 00   010     97              STXO    INTALY                                               CHARLY97
Y 102 RD CHARLY02
00  37 2350 00   000     98              LDA     31                                                   CHARLY98
777777 3750 03   000     99              ANA     =1,DU              GET LOWER MEMORY ADDRESS IN AU    CHARLY99
000001 0350 03   000    100 AGAIN        ADLA    1,DU               MAKE ADDRESS EVEN                 CHARLY00
777776 3750 03   000    101              ANA     =0777776,DU                                          CHARLY01
000152 7550 56   010    102              STA     OTTALY,ID          AND STORE INTO CALLING SEQUENCE   CHARLY02
000002 6070 04   000    103              TTF     2,IC               CHECK FOR TALLY RUNOUT            CHARLY03
000036 7100 00   010    104              TRA     CKMEM              YES                               CHARLY04
000151 7200 56   010    105              LXLU    INTALY,ID          SIZE OF ARRAY                     CHARLY05
000034 7400 00   010    106              STXO    *+1                                                  CHARLY06
000000 0350 03   000    107              ADLA    **,DU                                                CHARLY07
000025 7100 00   010    108              TRA     AGAIN                                                CHARLY08
000151 7200 56   010    109 CKMEM        LXLO    INTALY,ID          SIZE OF LAST ARRAY                CHARLY09
000040 7400 00   010    110              STXU    *+1                                                  CHARLY10
000000 0350 03   000    111              ADLA    **,DU                                                CHARLY11
000037 7510 70   000    112              STCA    31,70              UPDATE LOWER LIMIT                CHARLY12
000037 7200 00   000    113              LXLO    31                 UPPER MEMORY LIMIT                CHARLY13
000037 1600 00   000    114              SBXU    31                 LOWER MEMORY LIMIT                CHARLY14
000000 1600 03   000    115 X            SBXO    **,DU                                                CHARLY15
000000 6350 10   000    116              EAA     0,0              GET X0 IN AU                        CHARLY16
Y CARD CHARLY03
000022 7310 00   000    117              ARS     18               SHIFT AU TO AL W SIGN SET           CHARLY17
```

```
00006 7550 31   000   118          STA    6,1*                                                    CHARLY18
00154 6050 00   010   119          TPL    FCALL            WE HAVE ENOUGH MEMORY                  CHARLY19
00000 5310 00   000   120 GMORE    NEG                    MAKE DIFFERENCE POSITIVE                CHARLY20
01777 0350 07   000   121          ADLA   1023,DL         ROUND TO NEAREST 1K                     CHARLY21
00012 7710 00   000   122          ARL    10                                                      CHARLY22
00150 7550 00   010   123          STA    MEMSIZ                                                  CHARLY23
00061 7510 07   010   124          STCA   MME+1,07        STORE K REQD IN MME ZERO WORD           CHARLY24
00012 7350 00   000   125          ALS    10                                                      CHARLY25
00037 0550 00   000   126          ASA    31               UPDATE UPPER MEMORY LIMIT              CHARLY26
00011 0010 00   000   127 MME      MME    GEMORE          ASK FOR MORE MEMORY                     CHARLY27
00000 000000    000   128          ZERO   0,**                                                    CHARLY28
00002 7100 04   000   129          TRA    2,IC             DENIAL                                 CHARLY29
00135 7100 00   010   130          TRA    FCALL1                                                  CHARLY30
00065 7100 00   010   131 DENIAL   TRA    SMESG1                                                  CHARLY31
10000701000     030   132 SMESG1   CALL   CONSOL(MEMSIZ,IREPLY)DENRET                             CHARLY32
00073710000     010
00217000204     010
00150000000     010
 CARD CHARLY04
00103 000000    010
00103 710000    010
001   2340 00   010   133          SZN    IREPLY                                                  CHARLY33
00131 6000 00   010   134          TZE    DENRET                                                  CHARLY34
00143 0540 00   010   135          AOS    GMRIND                                                  CHARLY35
10000701000     030   136          CALL   CLKTIM(TIMZRO)                                          CHARLY36
00102710000     010
00217000210     010
00145000000     010
00130 2350 00   010   137          LDA    DENAL1                                                  CHARLY37
00064 7550 00   010   138          STA    DENIAL                                                  CHARLY38
00060 7100 00   010   139          TRA    MME                                                     CHARLY39
10000701000     030   140 SMESG2   CALL   CLKTIM(TIME)                                            CHARLY40
00111710000     010
00217000214     010
00146000000     010
00146 2350 00   010   141          LDA    TIME                                                    CHARLY41
00145 1750 00   010   142          SBA    TIMZRO                                                  CHARLY42
00036 1150 07   000   143          CMPA   30,DL                                                   CHARLY43
 CARD CHARLY05
00060 7100 00   010   144          TRA    MME                                                     CHARLY44
30000701000     030   145          CALL   CONMES                                                  CHARLY45
00120710000     010
```

```
00217000221      010
10000701000      031   146           CALL     CLKTIM(TIMZRO)                                    CHARLY46
00124710000      010
00217000222      010
00145000000      010
00144 0540 00    010   147           AOS      CLKCNT                                            CHARLY47
00144 2340 00    010   148           SZN      CLKCNT                                            CHARLY48
00131 6000 00    010   149           TZE      DENRET                                            CHARLY49
00060 7100 00    010   150           TRA      MME                                               CHARLY50
00105 7100 00    010   151  DENAL1   TRA      SMESG2                                            CHARLY51
        000132         152  DENRET   RETURN   CHARLY,1                                          CHARLY52
00217221051      010
00001161003      000
00217741000      010
00001710000      010
00143 2340 00    010   153  FCALL1   SZN      GMRIND                                            CHARLY53
00154 6000 00    010   154           TZE      FCALL                                             CHARLY54
 CARD CHARLY06
20 ‥0701003      030   155           CALL     CONGRT                                            CHARLY55
00 104 2710000   010
00 4 7000233     010
00154 7100 00    010   156           TRA      FCALL                                             CHARLY56
00000000000      000   157  GMRIND   DEC      0                                                 CHARLY57
77777777766      000   158  CLKCNT   DEC      -10                                               CHARLY58
        000145         159  TIMZRO   BSS      1                                                 CHARLY59
        000146         160  TIME     BSS      1                                                 CHARLY60
        000147         161  IREPLY   BSS      1                                                 CHARLY61
        000150         162  MEMSIZ   BSS      1                                                 CHARLY62
00000 0000 00    000   163  INTALY   TALLY    **,0             INPUT TALLY WORD                 CHARLY63
00000 0000 00    000   164  OTTALY   TALLY    **,**            WORKING TALLY WORD               CHARLY64
00160 0000 00    010   165           TALLY    FCALL+4,**                                        CHARLY65
                       166           SYMREF   SOLVEI                                            CHARLY66
50000 7010 00    030   167  FCALL    TSX1     SOLVEI                                            CHARLY67
00216 7100 00    010   168           TRA      NCALL                                             CHARLY68
00217 000000     010   169           ZERO     ?E,L..                                            CHARLY69
00160 0000 00    010   170           ARG      *+1                                               CHARLY70
        000165         171           BSS      30               MAXIMUM OF 30 ARRAYS             CHARLY71
00001710000      010   172  NCALL    RETURN   CHARLY                                            CHARLY72

AG?

00000000000      000
```

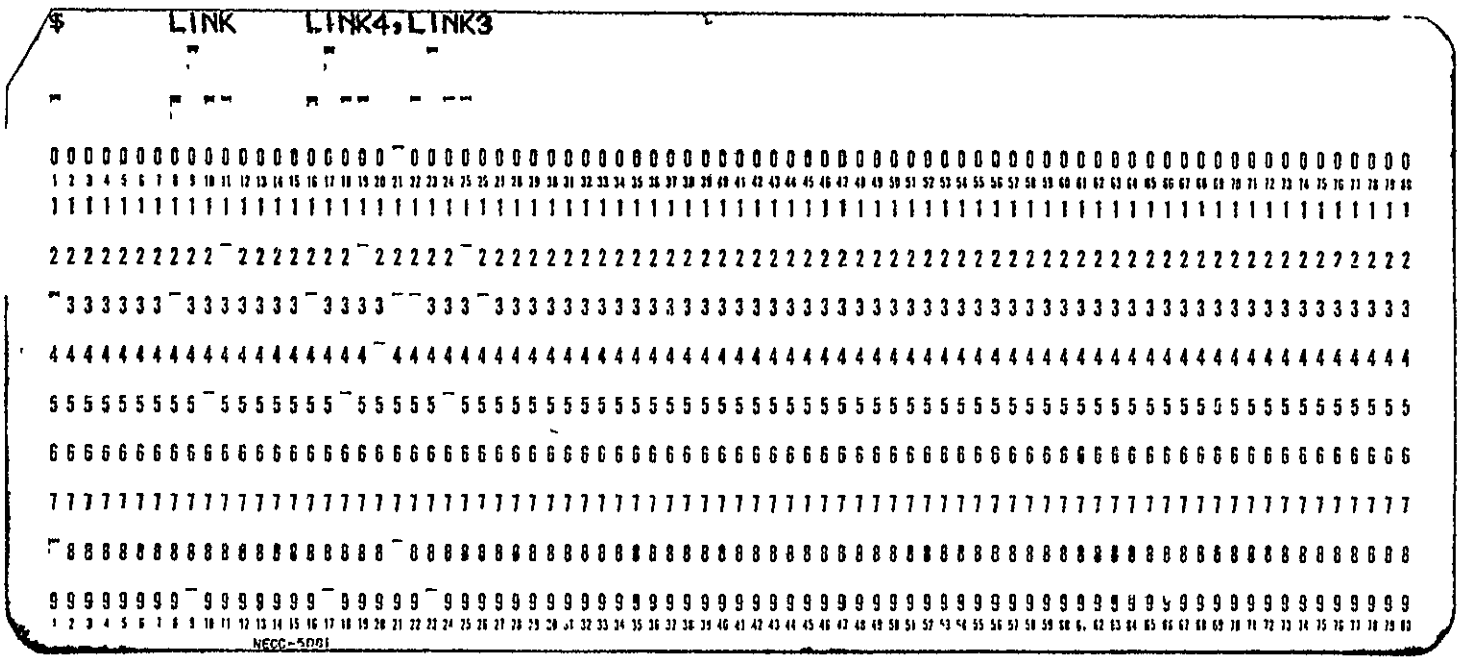
```
57345 03  03-16-77   17.017      SUBROUTINE CHARLY - DYNAMIC MEMORY ALLOCATOR

      000220  233021514370    000
  END OF BINARY CARD CHARLY07
                                         173         END
   222 IS THE NEXT AVAILABLE LOCATION.
 GMAP VERSION/ASSEMBLY DATES   JMPA 730601/052373      JMPB 730601/052373      JMPC 730601/052373
 THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

```
      CCONSOL
            SUBROUTINE CONSOL(MEMSZE,IREPLY,*)                                CONSOL01
            DIMENSION MES(10),MES0(11),MES1(11),MES2(8),MES21(9),MES3(9),     CONSOL02
           ,    MES5(9)                                                       CONSOL03
            CHARACTER*6 MES,MES0,MES1,MES2,MES21,MES3,MES5                    CONSOL04
            CHARACTER*6 CONSLW,NANS1,NANS2,NANS3,NANS4                        CONSOL05
            CHARACTER*6 NANS5,NANS6,NANS7,NANS8                               CONSOL06
            CHARACTER *6 IZERO                                                CONSOL07
            CHARACTER*6 NREPLY                                                CONSOL08
            DATA MES/60HPLEASE SET OTHER JOBS URGC TO 00 SO THEY WILL SWAP OUT CONSOL09
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST                CONSOL10
           ,...   /                                                           CONSOL11
            DATA MES0/66HTYPE ** YES ** AND SET ALL OTHER ** URGC ** TO 00 OR CONSOL12
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
           ,** NO **      /                                                   CONSOL13
            DATA MES1/66HGEODYN NEEDS      K ADDITIONAL CORE FOR A TOTAL OF   CONSOL14
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
           ,        K...   /                                                  CONSOL15
            DATA MES2/48HTYPE YES AND TEMPORARILY SWAP OTHER JOBS OUT,,, /    CONSOL16
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
            DATA MES21/54H    BY SETTING ALL OTHER URGENCIES TO ZERO...   OR.. CONSOL17
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
           ,. /                                                               CONSOL18
            DATA MES3/54HTYPE NO AND GEODYN WILL STOP EXECUTION AND ENDJOB.   CONSOL19
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
           , /                                                                CONSOL20
            DATA MES5/54H**GEODYN** GOT THE CORE IT NEEDED...   THANK YOU,    CONSOL21
*****W  402 DATA STATEMENT: VARIABLE LIST EXCEEDS LITERAL LIST
           , /                                                                CONSOL22
            DATA CONSLW/6H0000T//,NANS1/6HYES   /,NANS2/6HY                   CONSOL23
           ,NANS3/6HNO    /,NANS4/6HN     /                                   CONSOL24
            DATA NANS5/6HYES000/,NANS6/6HY00000/,NANS7/6HNO0000/,NANS8/6HN0000 CONSOL25
           ,0/                                                                CONSOL26
            DATA IZERO/6H000000/                                              CONSOL27
            NREPLY=IZERO                                                      CONSOL28
            CALL MEMSIZ(MTSIZE)                                               CONSOL29
            MTSIZE=MTSIZE+MEMSZE                                              CONSOL30
            PRINT 100,MEMSZE,MTSIZE                                           CONSOL31
        100 FORMAT(1X,O12,1X,O12)                                             CONSOL32
            IF(MTSIZE.GT.80.OR.MEMSZE.GT.25) GO TO 15                         CONSOL33
            ENCODE (MES1(3),10) MEMSZE                                        CONSOL34
            ENCODE (MES1(10),10) MTSIZE                                       CONSOL35
```

```
      10 FORMAT(I6)                                                    CONSOL36
         CALL CNSLIO(CONSLW,MES1,11)                                   CONSOL37
      11 CALL CNSLIO(CONSLW,MES2,8)                                    CONSOL38
         CALL CNSLIO(CONSLW,MES21,9)                                   CONSOL39
         CALL CNSLIO(CONSLW,MES3,9)                                    CONSOL40
         CALL CNSLIO(CONSLW,MES0,11,NREPLY)                            CONSOL41
         IF(NREPLY.EQ.NANS1.OR.NREPLY.EQ.NANS2) GO TO 12               CONSOL42
         IF(NREPLY.EQ.NANS5.OR.NREPLY.EQ.NANS6) GO TO 12               CONSOL43
         IF(NREPLY.EQ.NANS3.OR.NREPLY.EQ.NANS4) GO TO 13               CONSOL44
         IF(NREPLY.EQ.NANS7.OR.NREPLY.EQ.NANS8) GO TO 13               CONSOL45
         NREPLY=IZERO                                                  CONSOL46
         GO TO 11                                                      CONSOL47
      12 IREPLY=1                                                      CONSOL48
         GO TO 14                                                      CONSOL49
      13 IREPLY=0                                                      CONSOL50
      14 RETURN                                                        CONSOL51
      15 RETURN 1                                                      CONSOL52
         ENTRY GONMES                                                  CONSOL53
         CALL CNSLIO(CONSLW,MES,10)                                    CONSOL54
         RETURN                                                        CONSOL55
         ENTRY GONGRT                                                  CONSOL56
         CALL CNSLIO(CONSLW,MES5,9)                                    CONSOL57
         RETURN                                                        CONSOL58
         END                                                           CONSOL59
```

```
CCLKTIM
          SUBROUTINE CLKTIM(ITIME)
          CALL PTIME(ATIME)
          ITIME=ATIME*3600.
          RETURN
          END
```


$ LINK LINK4,LINK3
NECC-5081

```
        CESTIM
              SUBROUTINE ESTIM1(SUM1,SUM2,DELTA,BBIAS,PARNOI,PARTLI,NDX1,NDX2)
              DOUBLE PRECISION SUM1(1),SUM2(1),DELTA(1),PARTLI(1),PARTLC(1),WT
              DIMENSION BBIAS(1),BIASSG(3),SIGMA(1)
              INTEGER PARNOI(1),PARNOC,SCRC,SNREV(1),KNSTR(1)
              LOGICAL NEWARC
              COMMON/TAPES/INTP,IOUT,GTAP,NSTAPE,SNTAPE,SCRB,SCRC
              DATA NEWARC/.TRUE./
              DATA BIASSG/0.001,.2,10./
              INDXNO(I)=NDIM*(I-1)-(I*(I-1))/2
              NPARI=NDX1
              NPCOM=NPARI*NDX2
              NDIM=NPARI+NPCOM
              NSTART=NPARI+1
              IEND=NPARI*2
              I1=INDXNO(NDIM)+NDIM
              CALL CLEAR(SUM1,2*I1)
              CALL CLEAR(SUM2,2*NDIM)
              RETURN
              ENTRY ESTIM(PARNOC,PARTLC,SIGMA,RESID)
              IF(SIGMA(1).LE.0.0.OR.SIGMA(2).LE.0.0) RETURN
              IF(.NOT.NEWARC) GO TO 25
              NSTORE=INDXNO(NPARI)+NDIM
              CALL CLEAR(SUM1,2*NSTORE)
              CALL CLEAR(SUM2,2*NPARI)
              NEWARC=.FALSE.
           25 DO 50 I=1,NPARI
              I1=I+NPARI
              PARTLI(I1)=PARTLC(I)
           50 PARNOI(I1)=PARNOC+I
              WT=1.0/(SIGMA(1)**2+SIGMA(2)**2)
              DO 75 I=1,IEND
              I1=PARNOI(I)
              T=PARTLI(I)*WT
              SUM2(I1)=SUM2(I1)+T*RESID
              DO 65 J=I,IEND
              IST=INDXNO(I1)+PARNOI(J)
           65 SUM1(IST)=SUM1(IST)+T*PARTLI(J)
           75 CONTINUE
              RETURN
              ENTRY ESTIMA(ICON)
              IF(ICON.EQ.0) GO TO 100
```

57345 04 03-16-77   17.025

```
      DO 85 I=ICON,NPARI
      I1=INDXNQ(I)+I
   85 SUM1(I1)=SUM1(I1)+1.0D0/BIASSG(I)**2
  100 CALL SYMINV(SUM1,NDIM,NPARI,DELTA)
      I1=INDXNQ(NSTART)
      DO 200 I=NSTART,NDIM
      L1=0
      DO 160 L=1,NPARI
      DELTA(L)=0.0D0
      IST=0
      DO 140 J=1,L
      J1=IST+L
      J2=IST+I
      DELTA(L)=DELTA(L)+SUM1(J1)*SUM1(J2)
  140 IST=IST+NDIM-J
      IF(L.EQ.NPARI) GO TO 150
      LP1=L+1
      DO 145 J=LP1,NPARI
      J1=L1+J
      J2=IST+I
      DELTA(L)=DELTA(L)+SUM1(J1)*SUM1(J2)
  145 IST=IST+NDIM-J
  150 SUM2(I)=SUM2(I)-SUM2(L)*DELTA(L)
      DO 155 K=I,NDIM
      K1=I1+K
      K2=L1+K
  155 SUM1(K1)=SUM1(K1)-DELTA(L)*SUM1(K2)
  160 L1=L1+NDIM-L
      J1=I
      DO 175 J=1,NPARI
      SUM1(J1)=DELTA(J)
  175 J1=J1+NDIM-J
  200 I1=I1+NDIM-I
      IST=0
      DO 250 I=1,NPARI
      DELTA(I)=0.0D0
      I1=I
      DO 240 J=1,I
      DELTA(I)=DELTA(I)+SUM1(I1)*SUM2(J)
  240 I1=I1+NDIM-J
      IF(I.EQ.NPARI) GO TO 248
      IP1=I+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
            DO 245 J=IP1,NPARI
            I1=IST+J
        245 DELTA(I)=DELTA(I)+SUM1(I1)*SUM2(J)
        246 BBIAS(I)=BBIAS(I)+DELTA(I)
        250 IST=IST+NDIM-I
            WRITE(SCRC) (SUM1(I),I=1,NSTORE),(SUM2(I),I=1,NPARI),
           ,   (BBIAS(I),I=1,NPARI)
            NEWARC=.TRUE.
            RETURN
            ENTRY ESTIMB(SNREV,KNSTR,IKNSTR)
            DO 300 I=1,NDX2
            IJK=NUMBER(SNREV(I),KNSTR,IKNSTR)
            IF(IJK.EQ.0) GO TO 300
            L1=KNSTR(IJK+IKNSTR)
            IF(L1.GT.NPARI) GO TO 300
            DO 275 J=L1,NPARI
            I1=NPARI*I+J
            I1=INDXNO(I1)+I1
        275 SUM1(I1)=SUM1(I1)+1.0D0/BIASSG(J)**2
        300 CONTINUE
        310 IST=INDXNO(NSTART)
            I1=IST+NSTART
            CALL SYMINV(SUM1(I1),NPCOM,NPCOM,DELTA(NSTART))
            I2=IST
            DO 350 I=NSTART,NDIM
            DELTA(I)=0.0D0
            I1=I2+I
            DO 340 J=NSTART,I
            DELTA(I)=DELTA(I)+SUM1(I1)*SUM2(J)
        340 I1=I1+NDIM-J
            IF(I.EQ.NDIM) GO TO 348
            IP1=I+1
            DO 345 J=IP1,NDIM
            I1=IST+J
        345 DELTA(I)=DELTA(I)+SUM1(I1)*SUM2(J)
        348 BBIAS(I)=BBIAS(I)+DELTA(I)
        350 IST=IST+NDIM-I
            ENDFILE SCRC
            REWIND SCRC
            RETURN
            ENTRY UPDATE
            READ(SCRC) (SUM1(I),I=1,NSTORE),(SUM2(I),I=1,NPARI),
```

```
     .    (BBIAS(I),I=1,NPARI)
          I1=0
          DO 500 I=1,NPARI
          DELTA(I)=0.0D0
          DO 450 L=NSTART,NDIM
          L1=I1+L
      450 DELTA(I)=DELTA(I)-SUM1(L1)*DELTA(L)
          BBIAS(I)=DELTA(I)+BBIAS(I)
      500 I1=I1+NDIM-I
          RETURN
          END
****W     7 MEMORY EXPANDED, USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```

```
CSYMINV                                                                 SYMINV01
C NAME               SYMINV                                             SYMINV02
C                                                                       SYMINV03
C PURPOSE            TO RECURSIVELY FIND INVERSE OF SYMMETRIC MATRIX    SYMINV04
C                                                                       SYMINV05
C CALLING SEQUENCE   CALL SYMINV(SUM1,NDIM,NLIM,DELTA)                  SYMINV06
C                                                                       SYMINV07
C    SYMBOL  TYPE    DESCRIPTION                                        SYMINV08
C                                                                       SYMINV09
C    SUM1    DP      INPUT - LOWER RECTANGULAR PART OF MATRIX           SYMINV10
C    (1)                     TO BE INVERTED                             SYMINV11
C                    OUTPUT - LOWER RECTANGULAR PART OF INVERTED MATRIX SYMINV12
C                                                                       SYMINV13
C    NDIM    I       INPUT - DIMENSION OF MATRIX                        SYMINV14
C                                                                       SYMINV15
C    NLIM    I       INPUT - DIMENSION OF PARTITION TO BE INVERTED      SYMINV16
C                                                                       SYMINV17
C    DELTA   DP      SCRATCH                                            SYMINV18
C    (1)                                                                SYMINV19
C                                                                       SYMINV20
C SUBROUTINES USED   NONE                                               SYMINV21
C                                                                       SYMINV22
C COMMON BLOCKS      NONE                                               SYMINV23
C                                                                       SYMINV24
C INPUT FILES        NONE                                               SYMINV25
C                                                                       SYMINV26
C OUTPUT FILES       NONE                                               SYMINV27
C                                                                       SYMINV28
C REFERENCES         'GEODYN SYSTEMS DESCRIPTION'                       SYMINV29
C                    VOLUME 1 - GEODYN DOCUMENTATION                    SYMINV30
C                                                                       SYMINV31
C                                                                       SYMINV32
C                                                                       SYMINV33
C                                                                       SYMINV34
      SUBROUTINE SYMINV(SUM1,NDIM,NLIM,DELTA)                           SYMINV35
      DOUBLE PRECISION SUM1(1),DELTA(1)                                 SYMINV36
C INITIALIZE BY FINDING INVERSE OF 1X1                                  SYMINV37
      SUM1(1)=1.0D0/SUM1(1)                                             SYMINV38
      IF(NLIM.EQ.1) RETURN                                              SYMINV39
      N1=NDIM-1                                                         SYMINV40
C RECURSIVELY FIND INVERSE OF NXN KNOWING INVERSE OF (N-1)X(N-1) UNTIL  SYMINV41
C THE INVERSE OF AN NLIM X NLIM SQUARE PARTITION IS FOUND               SYMINV42
```

```
      DO 400 N=2,NLIM                                   SYMINV43
      NM1=N-1                                           SYMINV44
      L1=0                                              SYMINV45
      DO 100 L=1,NM1                                    SYMINV46
      J1=0                                              SYMINV47
      DELTA(L)=0.0D0                                    SYMINV48
      DO 60 J=1,L                                       SYMINV49
      JL=J1+L                                           SYMINV50
      JN=J1+N                                           SYMINV51
      DELTA(L)=DELTA(L)+SUM1(JL)*SUM1(JN)               SYMINV52
   60 J1=J1+NDIM-J                                      SYMINV53
      IF(L.EQ.NM1) GO TO 100                            SYMINV54
      LP1=L+1                                           SYMINV55
      DO 80 J=LP1,NM1                                   SYMINV56
      JN=J1+N                                           SYMINV57
      JL=L1+J                                           SYMINV58
      DELTA(L)=DELTA(L)+SUM1(JL)*SUM1(JN)               SYMINV59
   80 J1=J1+NDIM-J                                      SYMINV60
  100 L1=L1+NDIM-L                                      SYMINV61
      J1=N                                              SYMINV62
      NN=N1+N                                           SYMINV63
      DO 150 J=1,NM1                                    SYMINV64
      SUM1(NN)=SUM1(NN)-DELTA(J)*SUM1(J1)               SYMINV65
  150 J1=J1+NDIM-J                                      SYMINV66
      SUM1(NN)=1.0D0/SUM1(NN)                           SYMINV67
      J1=N                                              SYMINV68
      DO 200 J=1,NM1                                    SYMINV69
      SUM1(J1)=-DELTA(J)*SUM1(NN)                       SYMINV70
  200 J1=J1+NDIM-J                                      SYMINV71
      I1=N                                              SYMINV72
      DO 300 I=1,NM1                                    SYMINV73
      J1=I                                              SYMINV74
      DO 250 J=1,I                                      SYMINV75
      SUM1(J1)=SUM1(J1)-SUM1(I1)*DELTA(J)               SYMINV76
  250 J1=J1+NDIM-J                                      SYMINV77
  300 I1=I1+NDIM-I                                      SYMINV78
  400 N1=N1+NDIM-N                                      SYMINV79
      RETURN                                            SYMINV80
      END                                               SYMINV81
```

REFERENCES

1. Martin, T.V., W.F. Eddy, I.H. Oh, J.A. Kogut, "GEODYN System Description," prepared for NASA/GSFC under Contract No. NAS 5-22849, August 1976.

2. McMillan, J.D., "ARC Program Documentation," Wolf Research and Development Group, report prepared under Contract No. NAS 6-2173 for NASA/Wallops Flight Center, March 1976.

3. Amann, D.W., "ALTKAL Operations Manual," Wolf Research and Development Group, prepared under Contract No. NAS 6-2639 for NASA/Wallops Flight Center, January 1977.