(4)

THE DETERMINATION OF NAVSPASUR PHASE DIFFERENCE ERRORS

Technical Report
for
Contract N00014-87-C-2547

Prepared by:

Dr. E. James Wadiak

Dr. Michael D. Andrews

INTERFEROMETRICS INC.
8150 Leesburg Pike
Vienna, Virginia 22180

16 March 1988

EXECUTIVE SUMMARY

This paper describes the methodology and results of a project to determine the magnitude and amplitude dependence of the errors inherent in the full-Doppler mode NAVSPASUR phase difference data. The purpose of this work was to generate an accurate model of these errors for subsequent use in simulation programs designed to predict the performance capabilities of both the current NAVSPASUR system and various proposed upgrades. We report here on the preliminary results obtained from an analysis of full-Doppler mode data collected by the NAVSPASUR system during one four hour period. We plan in the near future to process a much more comprehensive data base in order to refine the error model and to allow us to address the possible sources of these errors. No attempt has been made in the current work to identify the source of the errors or to address whether (and/or which) operational improvements could be made to reduce their magnitude.

The main results of this project, described in detail in this report, are as follows:

1. For a given satellite pass data set (hereafter scan) the phase difference errors for each antenna array have both random and systematic components. For signals which are more than about 6 dB above the noise level the systematic errors are about 50% larger than the purely random errors. These systematic phase difference offsets are relatively constant from scan to scan, at least for time intervals up to 4 hours. Their behavior mimics (and they may in fact be) phase calibration errors. For strong signals the random errors are comparable in size to the phase data quantization size (5.6 degrees). However, since systematic errors dominate for high signal to noise, the phase quantization does not appear to limit overall accuracy at this time.

2. The random phase difference errors appear to be independent of actual or projected baseline length or orientation (N-S vs. E-W). The systematic errors show a bias towards larger errors on the shorter baselines. The net (both systematic and random) phase difference errors are normally distributed with a dispersion characterized by a single parameter - the received signal strength. No significant station-to-station variations are apparent, and there do not appear to be any temporal (day/night) variations of significance.

The RMS phase difference error as a function of received signal strength is well-behaved, and its dependence has been modelled with a fourth-order polynomial (see Figures 2 and 3).

3. An effective noise floor of approximately -152 dBm is confirmed as representative of the current system.

I. System Description and Analytic Approach

The fundamental NAVSPASUR measured quantity is the received signal phase at each antenna array at a given receiver site. The raw phases from each of the *n* antenna arrays at a receiver site are combined to produce *n-1* phase differences, each taken with respect to the designated reference antenna. The deployment of antenna arrays along each of two orthogonal directions, together with the use of antenna pairs (hereafter baselines) with different spatial separations (baseline lengths), allows for the simultaneous determination of two direction cosines (cos α, where α is the angle between the baseline vector and the satellite position vector in station coordinates) at each receiving site. Further, time-sequence observations of the phase differences associated with each baseline allow for the calculation of two direction cosine rates, d(cos α)/dt, which are functionally related to the satellite's velocity.

Because the direction angles and angle rates are derived directly from the phase *difference* data rather than the individual antenna phases, we focus here on determining the errors on the phase differences as opposed to the raw phases. The RMS phase difference errors due to random raw phase errors are expected to be $\sqrt{2}$ larger than the RMS raw phase errors. Further, the phase differences will also include systematic errors due to phase biases, or offsets, which would not appear in an analysis of raw phase data.

The phase difference data from a given receiver site allows for the calculation of four parameters which provide information on the satellite position and velocity. In order to fully specify the position and velocity of the satellite at some given time t_0, a total of six parameters are required. The NAVSPASUR system meets this requirement by combining measurements of four parameters from each of two or more spatially separated receiving sites. It is important to note here, however, that because of the particular geometry of the NAVSPASUR system, the parameters as determined from the different receiving stations are to varying degrees linearly dependent on one another, so that the satellite position and velocity are not necessarily well determined in all cases. For this reason the relationship between single station performance and system performance is a complex issue, and its treatment is deferred to a subsequent report, which will build on the results contained herein.

The phase difference data from which the direction cosines and cosine rates are obtained are contaminated by random noise and systematic measurement and calibration errors. In the absence of any noise or systematic errors, the direction cosines and cosine rates are overdetermined by the available data.[1] In the presence of noise and systematic errors, the problem becomes one of obtaining a best estimate of the desired parameters given the available data. While a variety of estimation techniques exist, perhaps the most common and straightforward technique is the method of least squares. With this method one adopts as the best estimators those values of the independent parameters which minimize the sum of the squares of the residual errors. A substantial body of mathematics exists in support of the least squares method, and it is extensively discussed in many statistical textbooks. For completeness, a description of the methodology is included here, following the discussions contained in Lewis and Odell (1971) and Mathews and Walker (1970).

II. The Method of Least Squares

a). Basic formulation of the nonlinear least squares problem

The method of least squares provides a means of obtaining a best estimate of a set of parameters, represented by a parameter vector Ψ of dimension m, given an associated data vector y of dimension $n > m$ such that the vector $f(\Psi)$ gives an approximate correspondence between Ψ and y. In general, the vector f is such that the parameters Ψ may be expressed nonlinearly in f. Where the functional form of f is known, the problem reduces to one of determining the parameter vector Ψ which most closely describes the n relationships expressed by the vector equation $y = f(\Psi)$, in the sense that the sum of the squares of the residuals,

$$Q(\Psi) = [y - f(\Psi)]^T[y - f(\Psi)] = \sum_{j=1}^{n}[y_j - f_j(\Psi)]^2 \tag{1}$$

is minimized. This requirement is satisfied when the partial derivatives of Q with respect to the parameters Ψ_i equal zero, *i.e.*,

$$Q_i(\Psi) \equiv \frac{\partial Q}{\partial \Psi_i} = -2[y - f(\Psi)]^T[\phi_i(\Psi)] = 0, \quad i = 1, 2, \ldots, m. \tag{2}$$

1. For each receiving station there are k determinations of each direction cosine and k-1 determinations of each direction cosine rate where, typically, $20 < k < 55$.

Here, $\phi_i(\Psi)$ is an n-dimensional vector given by

$$\phi_i(\Psi) = \frac{\partial f(\Psi)}{\partial \Psi_i}. \tag{3}$$

In the case where the output vector y is expressed linearly in Ψ, the conditions given by equations (2) yield m linear equations in m unknowns, which can be solved directly for the Ψ_i. When the function f is nonlinear in Ψ, the usual approach to minimizing equation (1) is to linearize equations (2) by choosing a suitable first estimate Ψ_0 of Ψ and replacing $f(\Psi)$ in equation (1) with its first-order Taylor series expansion about Ψ_0,

$$f(\Psi) \cong p(\Psi) = f(\Psi_0) + \sum_{k=1}^{m} (\Psi_k - \Psi_{0k})\phi_k(\Psi_0) \quad . \tag{4}$$

Substituting $p(\Psi)$ for $f(\Psi)$ in equation (1) and setting the partial derivatives $Q_i(\Psi)$ equal to zero then yields the following system of linear equations:

$$\begin{bmatrix} \phi_1^T\phi_1 & \phi_1^T\phi_2 & \cdots & \phi_1^T\phi_m \\ \phi_2^T\phi_1 & \phi_2^T\phi_2 & \cdots & \phi_2^T\phi_m \\ \vdots & \vdots & & \vdots \\ \phi_m^T\phi_1 & \phi_m^T\phi_2 & \cdots & \phi_m^T\phi_m \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{bmatrix} \tag{5}$$

where $\phi_i = \phi_i(\Psi_0)$ and $Q_i = Q_i(\Psi_0)$. Equation (5) can be written as the matrix equation

$$F\,d = -\tfrac{1}{2}\,Q \tag{6}$$

with the solution

$$d = -\tfrac{1}{2}\,F^{-1}Q \quad . \tag{7}$$

The elements $d_i = (\Psi_i - \Psi_{0i})$, $i = 1, 2, \ldots, m$ represent corrections to the first estimate Ψ_0 of Ψ. Equation (7) may be solved for the vector d using standard matrix inversion techniques. From the corrections d, one obtains an improved estimate $\Psi_1 = \Psi_0 + d$ for the parameter vector Ψ. Replacing Ψ_0 in equation (4) with Ψ_1, the above process is repeated to obtain a new estimate Ψ_2 for Ψ. This iterative procedure continues until one obtains sufficiently accurate estimates for each of the parameters Ψ_i.

b). Weighting data of nonuniform precision

In the above analysis we have assumed that all data points have the same relative precision, so that all points contribute equally to obtaining the least squares solution. When the data are not all of equal precision the least squares best estimate is obtained when the ***weighted*** sum of the squares of residuals, *i.e.*,

$$Q(\Psi) = \sum_{j=1}^{n} w_j [y_j - f_j(\Psi)]^2 \qquad (8)$$

are minimized. Here the w_j represent the weights for the individual data points. If the errors for all measurements of a given precision are assumed to be normally distributed, then the relative weight of those data points is inversely proportional to their variance, so that $w_j = 1/\sigma_j^2$. If the variances σ_j^2 are not known *a priori*, one may proceed by making a reasonable first estimate of their values, perform the least squares fitting, calculate refined values of the variances from the results, and use these newly obtained values as input to a second running of the least squares routine.

c). Error estimation and correlation of errors

Once a least squares estimate of the parameter vector Ψ has been obtained, it is useful to investigate the probable errors in the determined parameters. If the experiment under investigation were repeated a large number of times in exactly the same fashion and with the same error distribution, we could calculate the mean-square deviation of Ψ for any individual experiment from the grand average $\bar{\Psi}$ taken over all experiments. To find the errors in Ψ, then, we need to calculate the mean-square deviations $\langle(\Psi_i - \bar{\Psi}_i)^2\rangle$. Since $(\Psi_i - \bar{\Psi}_i) = (d_i - \bar{d}_i)$, we have, from equations (7), (8), and (3),

$$(\Psi_i - \bar{\Psi}_i) = \sum_{k=1}^{m} (F^{-1})_{ik} (Q_k - \bar{Q}_k) = \sum_{k=1}^{m}\sum_{j=1}^{n} (F^{-1})_{ik} w_j \Phi_{kj} (y_j - \bar{y}_j) \quad , \qquad (9)$$

where $\Phi_{kj} \equiv \partial f_j(\Psi)/\partial\Psi_k$. Since the individual measurements y_j are assumed to be statistically independent, the expectation values of the errors must satisfy

$$\Big\langle (y_j - \bar{y}_j)(y_q - \bar{y}_q) \Big\rangle = \sigma_j^2 \, \delta_{jq} = \frac{1}{w_j} \delta_{jj} \quad , \qquad (10)$$

so that, in general,

$$\Big\langle (\Psi_i - \bar{\Psi}_i)(\Psi_l - \bar{\Psi}_l) \Big\rangle = \sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{n} (F^{-1})_{ik} (F^{-1})_{lp} \Phi_{kj} \Phi_{pq} w_j w_q \frac{1}{w_j} \delta_{jq}$$

$$= \sum_k\sum_p (F^{-1})_{ik} (F^{-1})_{lp} \sum_j \frac{\Phi_{kj}\Phi_{pj}}{\sigma_j^2}$$

$$= \sum_k\sum_p (F^{-1})_{ik} (F^{-1})_{lp} F_{pk}$$

$$= (F^{-1})_{il} \quad . \qquad (11)$$

The standard error $\Delta\Psi_i$ in Ψ_i is therefore

$$\Delta\Psi_i = \Big\langle (\Psi_i - \bar{\Psi}_i)^2 \Big\rangle^{1/2} = \sqrt{(F^{-1})_{ii}} \quad . \qquad (12)$$

The matrix F^{-1} represents an error matrix whose diagonal elements are the squares of the standard errors in the parameters Ψ_i. The fact that the cross terms $\langle(\Psi_i - \bar{\Psi}_i)(\Psi_l - \bar{\Psi}_l)\rangle$, with $i \neq l$ are not in general equal to zero implies that the parameters Ψ_i are not statistically independent and that the errors are correlated. The expected error in any quantity which is a function of the parameter vector Ψ may be determined from the error matrix. For a given function $g(\Psi)$, the standard error in $g(\Psi)$ due to the uncertainties in the parameters Ψ may be found from the relation $\langle[g(\Psi) - \bar{g}(\Psi)]^2\rangle^{1/2} = \langle[g(\Psi - \bar{\Psi})]^2\rangle^{1/2}$ by direct substitution.

III. The Application of Least Squares to the NAVSPASUR System

We have applied a nonlinear least squares analysis with nonuniform data weighting to full-Doppler mode NAVSPASUR phase4 data in order to determine the distribution of the residual phase difference errors. For each data scan (defined as the data collected by one receiving station from a single pass of a target satellite illuminated by a single transmitter) we have obtained a best fit to the E-W and N-S direction angles (hereafter θ and Φ, respectively) and direction angle rates (hereafter $\dot{\theta}$ and $\dot{\Phi}$) at the epoch of fence crossing. From these, we calculate the expected value of the differential phase for each datum and derive the residual phase difference errors. We then analyze these residual errors to obtain their dependence on received signal amplitude. The details of this process are outlined in the following sections.

a). Coordinate system and basic equations

For the single-station fits to the data, the most natural coordinate system in which to work is the local station coordinates, where the reference antenna is at the origin and the orthogonal axes are local "north" and "east", both tangent to the geoid, and local height, or up, which is perpendicular to the surface of the geoid. Here "east" is eastward in the plane of the NAVSPASUR great circle, and "north" is normal to the great circle in a northward direction. Figure 1 depicts the orientation of a local station coordinate system on the earth's surface. For this coordinate system, all antenna offsets are to the "east" and/or "north", with no vertical component (*i.e.*, the antennas all lie in the tangent plane). The offsets from the coordinate origin can be obtained directly from the station survey plats. Further, the station location in geocentric coordinates and the rotation vectors needed to transform the satellite geocentric position and velocity to local station coordinates are known and can be checked for consistency against those currently used by NAVSPASUR. In this coordinate system the direction angles Θ and Φ and the distance to the satellite r_{sat} can be expressed as

$$\Theta(t) = \tan^{-1}\left[\frac{r_E(t)}{r_H(t)} \right]$$

$$\Phi(t) = \sin^{-1}\left[\frac{r_N(t)}{r_{sat}(t)} \right] \qquad (13)$$

$$r_{sat}(t) = \left(r_N(t)^2 + r_E(t)^2 + r_H(t)^2 \right)^{1/2}$$

where

$$r_N(t) = r_N(t_0) + v_N(t_0)(t-t_0) + \frac{GM_{\oplus}}{2|\vec{R}|^2}(t-t_0)^2 (\hat{N} \cdot \hat{R})$$

$$r_E(t) = r_E(t_0) + v_E(t_0)(t-t_0) + \frac{GM_{\oplus}}{2|\vec{R}|^2}(t-t_0)^2 (\hat{E} \cdot \hat{R}) \qquad (14)$$

$$r_H(t) = r_H(t_0) + v_H(t_0)(t-t_0) + \frac{GM_{\oplus}}{2|\vec{R}|^2}(t-t_0)^2 (\hat{H} \cdot \hat{R}) \quad .$$

Here t_0 is the epoch of fence crossing, r_i and v_i are the satellite position and velocity in local station coordinates, G is the gravitational constant, $M_{\oplus}$ is the mass of the earth, $\vec{R}$ is the satellite's position in geocentric coordinates, and the $\hat{N}$, $\hat{E}$, $\hat{H}$, are the unit direction vectors of the three local station coordinate axes. The angles used here, together with the satellite range, represent an orthogonal polar coordinate system in which θ measures the rotation angle in the great circle plane and Φ measures the angle normal to the great circle. As such, the angles θ and Φ do not exactly correspond to the angles measured by the two direction cosines at the station location. However, for the particular geometry of the NAVSPASUR system, where the satellite is illuminated only within a few arcminutes of the great circle plane, these angles do closely correspond to those measured by the direction cosines. For computational simplicity it is preferable to work with these angles as opposed to working directly with the direction cosine angles. This choice of angle definitions has no effect on the results of our analysis.

Over the short duration of the satellite's illumination by the transmitter beam (≤ 1 sec), the last (acceleration) terms in equations (14) can safely be ignored. The direction angle rates and range rate may be obtained by taking the time derivatives of equations (13), where again the acceleration terms can safely be ignored, so that we have

$$\dot{\theta}(t) = \frac{v_E(t_0)\cos\{\theta(t)\} - v_H(t_0)\sin\{\theta(t)\}}{\rho(t)}$$

$$\dot{\Phi}(t) = \frac{v_N(t_0) - v_E(t_0)\sin\{\theta(t)\}\sin\{\Phi(t)\} - v_H(t_0)\cos\{\theta(t)\}\sin\{\Phi(t)\}}{\rho(t)} \qquad (15)$$

$$\dot{r}_{sat}(t) = \frac{v_N(t_0)\,r_N(t) + v_E(t_0)\,r_E(t) + v_H(t_0)\,r_H(t)}{r_{sat}(t)}$$

where

$$\rho(t) = [\,r_H^2(t) + r_E^2(t)\,]^{1/2} .$$

For the least squares fitting procedure, equations (13) and (15) provide a means of obtaining a first estimate of the angles and angle rates from the predicted satellite position and velocity at the epoch of fence crossing. Because the single station data can effectively constrain only these four parameters, we must fix the two remaining parameters, r_{sat} and $\dot{r}_{sat}$. The logical choice for these parameters is to fix them to their *a priori* values based on the ephemeris prediction. Since the phase differences are insensitive to range and range rate, this approximation introduces no significant error into our calculations. With this assumption, then, the satellite position at time t on subsequent iterations is given in terms of the current estimates of the direction angles and direction angle rates by

$$r_N(t) = r_{sat}(t) \sin\{\Phi_n(t)\}$$

$$r_E(t) = r_{sat}(t) \cos\{\Phi_n(t)\} \sin\{\theta_n(t)\} \qquad (16)$$

$$r_H(t) = r_{sat}(t) \cos\{\Phi_n(t)\} \cos\{\theta_n(t)\}$$

where

$$r_{sat}(t) = r_{sat}(t_0) + \dot{r}_{sat}(t_0)(t - t_0)$$

$$\Phi_n(t) = \Phi_n(t_0) + \dot{\Phi}_n(t_0)(t - t_0) \qquad (17)$$

$$\theta_n(t) = \theta_n(t_0) + \dot{\theta}_n(t_0)(t - t_0) \quad .$$

Here, the subscript n has been added to the angles and angle rates to indicate their value after the nth iteration. Equations (16) and (17) are the basic equations from which the satellite position (and the expected phase differences) are calculated for all least squares iterations after the first.

b). Derivation of the "ideal" phase difference

For a satellite of known position and velocity, the phase difference (in rotations) between two receiving antennas at known positions is simply the difference in the respective path lengths from the satellite, expressed in wavelengths. For the satellite-receiver distances typical for the NAVSPASUR system, one cannot use a plane wave approximation over the longest baselines, and the respective path lengths must be explicitly calculated. Since the reference antenna has been taken as the origin of the local station coordinate system used, this phase difference is simply

$$\xi_k = \frac{|\vec{r}_{sat} - \vec{r}_k| - |\vec{r}_{sat}|}{\lambda_{rec}} \tag{18}$$

where ξ_k is the phase difference at the kth antenna relative to the reference antenna, $\vec{r}_k$ is the position vector of the kth antenna, and λ_{rec} is the wavelength of the received (*i.e.*, Doppler shifted) signal. In actuality, the NAVSPASUR system only measures antenna phases modulo 1 (*i.e.*, 0 to 1 rotations, or 0 to 360 degrees). In the least squares fitting program we perform the appropriate modular arithmetic and express all phase differences, both observed and ideal, in the range -0.5 to +0.5 rotations.

At the beginning of each iteration of the least squares routine we calculate the ideal, or expected, phase differences based on the satellite position as derived from equations (16) and (17).

c). Calculation of the phase difference partial derivatives

In addition to the the expected phase differences, we also need for the least squares routine the partial derivatives of the phase differences with respect to the parameters to be solved for (Φ, θ, $\dot{\Phi}$, and $\dot{\theta}$). We obtain these using a finite difference scheme, where, for example, the partial derivative with respect to Φ of the phase difference for the baseline associated with the kth antenna is given by

$$\frac{\partial \xi_k}{\partial \Phi} = \frac{\xi_k(\theta,\ \Phi+\delta\Phi,\ r_{sat},\ \dot{\theta},\ \dot{\Phi},\ \dot{r}_{sat}) - \xi_k(\theta,\ \Phi-\delta\Phi,\ r_{sat},\ \dot{\theta},\ \dot{\Phi},\ \dot{r}_{sat})}{2\delta\Phi} \tag{19}$$

where we take $\delta\Phi = \Phi/10^5$. The partials for the three remaining parameters are calculated in an analagous manner.

d). Data calibration

For each data tape received from Dahlgren we extract the half-hourly phase calibration data scans. For each calibration scan we compute the differenced phases with respect to the reference antenna for each antenna and timeline, and average all timelines. Prior to the averaging process, the data are screened to remove any spurious data points which occasionally occur at the end of the cal scans. Given the good phase stability of the system, these points are easily identified and removed. If the data tape spans more than one calibration scan for a given receiver site, the differenced phases for each baseline are averaged across all available calibration scans, which again is possible owing to the excellent phase stability of the system.

Once the phase calibration values for all the receiving stations have been established, these calibration values are subtracted from the individual satellite data scans. Finally, the phase differences for each baseline are computed for each data scan. Only data scans for which the target satellite has been identified as a known catalog object are retained for processing, since the nonlinear least squares analysis requires reasonably good initial estimates of the position and velocity to achieve convergence. The calibrated phase difference data, together with the corresponding received signal amplitude, time of observation, Doppler frequency, catalog position and velocity, and epoch of fence crossing are written to a calibrated data file which is used as input to the least squares fitting program.

e). Data weighting

Since the purpose of this project was to derive an accurate error model for the full-Doppler mode NAVSPASUR data, we did not have an initial error model available to use in the data weighting. From an examination of the calibrated data scans we produced an initial rough estimate of the error model, fitting the estimated errors to a piecewise linear function of signal strength. This initial model was used as the error weighting model for the first pass least squares fitting of the data, and from the first pass we generated a fourth-order polynomial error model. This new model was then used as the input error model for data weighting in a subsequent pass of the least squares routine.

In order to test the sensitivity of the results to the assumed error model we also ran the data base with no (*i.e.*, uniform) weighting of the input data. A comparision of the final error models for each of these three cases demonstrated that the form of the final error model was not very sensitive to the form of the input error model used for the data weighting.

f). Least squares fitting to the data

Initial estimates of the angles, angle rates, range, and range rate are calculated from the *a priori* (catalog) satellite position and velocity contained in the data scan header using the relationships given in equations (13) and (15). In the least squares fitting process we fix the range and range rate to their initial values, since data from a single receiving station can constrain only the remaining four parameters. The angles and angle rates form a four dimensional parameter vector Ψ which we wish to estimate in the least squares sense. From the initial estimate Ψ_0 of Ψ we calculate the expected phase differences $p(\Psi_0)$ and the partial derivatives of the phase differences $\phi(\Psi)$. For each datum we also calculate the data weight $w_i = 1/\sigma_i^2$ based on a piecewise linear initial estimate of the error model. We then construct the normal equations as described in §II, yielding four equations in the four unknown parameters, *i.e.*, the four corrections d_i to the initial estimates Ψ_0. From these we compute new estimates Ψ_1 which are used as the starting point for the second iteration. These new estimates are used to obtain the satellite position via equations (16) and (17), and the remainder of the process continues as for the initial iteration.

We continue iterating in this fashion until the solution converges or until a specified maximum number of iterations are completed. The convergence criterion we adopted was that the magnitude of each of the corrections d_i at the end of an iteration be less than or equal to the estimated errors in the corrections $\Delta d_i = \Delta\Psi_i$. If this convergence test is not met within three iterations the fitting process is abandoned and we proceed to the next data scan. The three iteration limit was selected empirically based on our experience in generating fits to NAVSPASUR data. When the satellite signal is sufficiently above the noise level and the identification in the scan header is correct, the least squares routine typically converges in one or two iterations. When the routine is unable to converge within three iterations, our experience shows that it is unlikely to converge at all. The failure to achieve convergence is due in almost all cases to poor signal-to-noise. Therefore, to prevent the waste of excessive amounts of computer time, we set the convergence limit to three iterations. All data scans for which the program failed to reach convergence were retained for future analysis. The non-convergent scans represent approximately 8% of the total number of scans processed.

Once the least squares fits to the observational data were complete, the residual phase difference errors for each datum were computed. These residuals, together with the corresponding received signal amplitudes and the data scan headers, were saved to disk files for further analysis.

IV. Analysis of the Phase Difference Errors

The analysis we present here is based on a four hour NAVSPASUR phase4 data tape containing all observations for the period 20AUG87 01:50 UT - 06:00 UT. Since the nonlinear least squares fitting routine requires a good initial estimate of the satellite position and velocity, only those scans for which a catalog identification was assigned were processed. This amounted to a total of 4,212 data scans. Prior to least-squares processing of the data, the calibration scans from the data tape were carefully checked to ensure that no calibration errors were present which would propagate into the data analysis. The calibration data were found to be well-behaved over the full time interval spanned by the data.

During the time period spanned by the data tape the Hawkinsville station was operating in the 'J1' mode instead of its normal configuration. Therefore, we have excluded the Hawkinsville data (925 scans) from our analysis. This exclusion is not expected to significantly affect our results, since we have analyzed Hawkinsville data from other shorter time period data tapes and find that the error distribution is similar to that which we report here for the remaining stations.

Approximately 17% of the data scans which nominally met the convergence criteria produced clearly anomalous results, in that the mean residuals for the majority of the baselines were excessively large (typically 70 - 150 degrees), while the dispersion in the residuals was normal (typically 9 degrees). We believe these scans to be cases for which the assigned catalog identification is incorrect. In the event of an incorrect satellite identification, it is possible (and, in fact, likely) that the nonlinear least squares routine could converge on incorrect values of the parameters. This occurs because the nonlinear least squares method requires a reasonably good first estimate of the parameters in order to converge on the correct answer. Given a poor first estimate of the parameters, the algorithm can converge on a *local*, rather than global minimum of the sum of the squares of the residuals.

We have examined several of the anomalous scans and have been able to verify in many cases that the originally assigned identification was incorrect. While we have not been able to verify that this was the case for all the anomalous scans, it does appear that they may all be due to the least squares routine converging on the wrong minimum. To avoid contamination of our results be these anomalous scans, we have eliminated them from our data analysis.

After filtering the data to remove the nonconvergent and anomalous scans, and all scans for Hawkinsville, our data set consisted of 3,055 scans. We have analyzed the remaining data at three levels and intercompared our results from each level, as described below.

First, we examine the data on an individual scan basis. At this level of examination, there is insufficient data to determine the amplitude dependence of the errors. We therefore have combined all data whose amplitudes exceed the noise level by more than 6 dB on a baseline-by-baseline basis and perform a statistical analysis. We calculate for these data the mean, the dispersion about the mean, and the RMS magnitude of the residuals. If the noise were purely random, the mean of the residuals for each baseline should be zero within the statistical errors, and the calculated dispersions and RMS deviations should be approximately equal. We find that, in general, this is not the case.

Typically, the absolute value of the mean of the residuals for any baseline is about 14 degrees, and the dispersion about the mean is about 9 degrees. The deviation of the means from zero is statistically significant given the number of data points. Further, the means for any given baseline correlate well from scan to scan, even over time intervals of up to four hours (the longest time period sampled to date). This suggests that there are in fact significant systematic effects contributing to the phase difference errors, a point we will return to below. The magnitude of the combined systematic and random errors is significantly larger than the phase quantization size in the raw data ($5\overset{\circ}{.}6$), suggesting that for the current operation phase quantization effects are not a limiting factor to the system performance.

The second level of data examination occurs at the station level, where data for the whole time period for a given receiving station is combined on both a baseline-by-baseline basis and on an overall station basis. At this level we are able to bin the data by received signal amplitude and determine the amplitude dependence of the errors. For data whose amplitude exceeds the system noise level by more than 6 dB, we find that, for any given baseline, the RMS errors are constant with signal strength and are typically 11-29 degrees. The significant variations we see between different baselines is due primarily to the fact that the means are not zero within statistical errors, presumably due to systematic errors. The standard deviation about the mean, typically 9 degrees, remains fairly constant from baseline to baseline, however.

We have not searched in detail for correlations in the apparent systematic errors, but a visual inspection of the data indicates that the magnitude in the systematic component anti-correlates with baseline length. That is, the shortest baselines exhibit the largest mean deviation from zero. The interpretation of this trend may be misleading, however, since systematic errors on the longest baselines can be compensated for by relatively small changes to the best estimates of the direction angles. In other words, systematic errors on the long baselines, if present, may result in biased solutions for the direction angles! For data which is less than about 6 dB above the noise, we find that both the RMS deviations and the dispersions about the mean increase as signal strength decreases. For these amplitudes the random component of the errors dominates the systematic errors.

At the station level of analysis we have investigated whether there are differences in the RMS errors for E-W baselines as opposed to N-S baselines. We binned the residuals for each individual station on the basis of baseline orientation and produced polynomial fits to the data. While we did find some differences between these populations at each station (see Figure 2, which is representative), we detected no repeatable trend from station to station. For each station we have plotted the RMS errors vs. amplitude for all baselines combined. An inter-comparison of these plots showed that, in a statistical sense, the error patterns at all stations were identical.

The third level of analysis which we performed was at the system level. In this case we combined all the residuals from all stations (except Hawkinsville) and binned the data by signal strength. A plot of the errors vs. amplitude (see Figure 3) shows that there are no significant N-S vs. E-W differences, and that the amplitude dependence is well behaved. In the Figure, the amplitude is expressed in dB above the assumed noise floor of -152 dBm. The curves show that this is in fact a reasonable estimate of the overall noise level. For signals which are well above the noise, we find that the RMS errors are typically 18 degrees and are nearly independent of amplitude.

The dotted lines on the graph show the best fourth-order polynomial fit to the data. The fit to the ensemble of all data for the four hour period investigated is expressed as

$$R = - 3.67\times10^{-5} S^4 - 1.61\times10^{-3} S^3 + 0.195 S^2 - 4.44 S + 47.2 \quad (20)$$

where R is the RMS magnitude of the residuals, in degrees, and S is the received signal strength in dB above the assumed noise level. In addition to examining the dependence of the errors on the amplitude, it is instructive to inspect the distribution of the phase difference errors at any given amplitude. Because the RMS error is approximately independent of signal strength for signals greater than -140 dBm, we have combined these data to generate the histogram in Figure 4. Superimposed on the histogram is a Guassian distribution with a standard deviation of 18 degrees. Figure 4 demonstrates that the ensemble of residuals is nearly normally distributed, although the skew to positive error values is real and statistically significant. Similar curves for lower signal stregths also display a normal distribution and also show significant skewing to positive error values. The reason for this skewing is not presently understood.

As part of our continuing effort to understand the nature and magnitude of the errors in the NAVSPASUR data, we are currently analyzing a larger data base. With this larger data base we should be able to identify correlations in the error distribution as well as refine our error model. Preliminary results with the larger data base indicate that, for example, the errors associated with the Kickapoo transmitter are larger than those associated with the other two transmitters. We will present the results of our further analysis in a future report when available.

VI. Summary of Results

We have performed a nonlinear least squares analysis on a four hour segment of full-Doppler mode NAVSPASUR phase4 data. This analysis has provided direct estimates of the phase difference errors present in full-Doppler mode data and the dependence of these errors on received signal strength.

We find that for signals whose amplitudes exceed ~ -145 dBm, systematic errors, which are typically 14 degrees, dominate the random errors, which are nearly constant over this amplitude range and are typically 9 degrees. The systematic errors for any given baseline remain fairly constant over the time interval we have investigated. There is a tendency towards larger systematic errors on the shortest baselines. However, this tendency may be an artifact of the least squares fitting process, since phase difference offsets on the longest baselines result in biased solutions to the direction angles.

At amplitudes below about -145 dBm, the random component of the phase difference errors dominate the systematic errors, and the RMS errors grow quickly with decreasing amplitude. This finding is in agreement with our assumption that the effective noise floor is approximately -152 dBm.

The total (systematic plus random) phase difference errors are, when analyzed as an ensemble, normally distributed with a dispersion which is well characterized as a function of received signal strength. The dependence on signal strength is accurately modelled by a fourth-order polynomial between the lowest and highest amplitudes normally encountered by the system.

The random component of the phase difference errors are comparable in size to the uncertainties due to the 6-bit quantization of the phase data. However, since the total phase difference errors always dominate the contribution due to phase quantization, the 6-bit quantization of phase data is not a limiting factor in system performance.

REFERENCES

Lewis, T. O., and Odell, P. L. 1971, *Estimation in Linear Models*, (Englewood Cliffs, NJ: Prentice-Hall).

Mathews, J. and Walker, R. L. 1970, *Mathematical Methods of Physics*, (Menlo Park, CA: Benjamin Cummings).

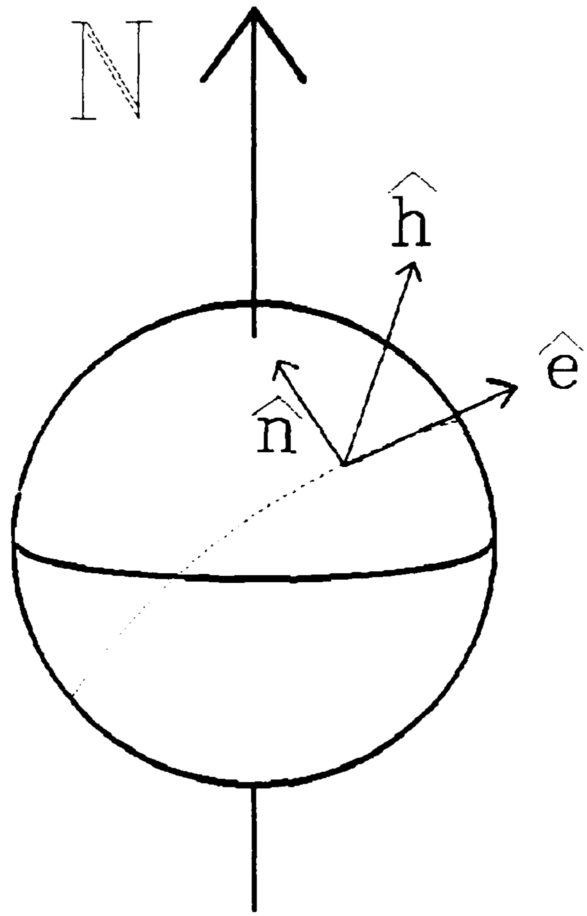

FIGURE 1: Diagram showing the orientation of a local station coordinate system on the surface of the geoidal earth. The **h** and **e** axes lie in the plane of the NAVSPASUR great circle, with the **n** axis perpendicular to the great circle plane.

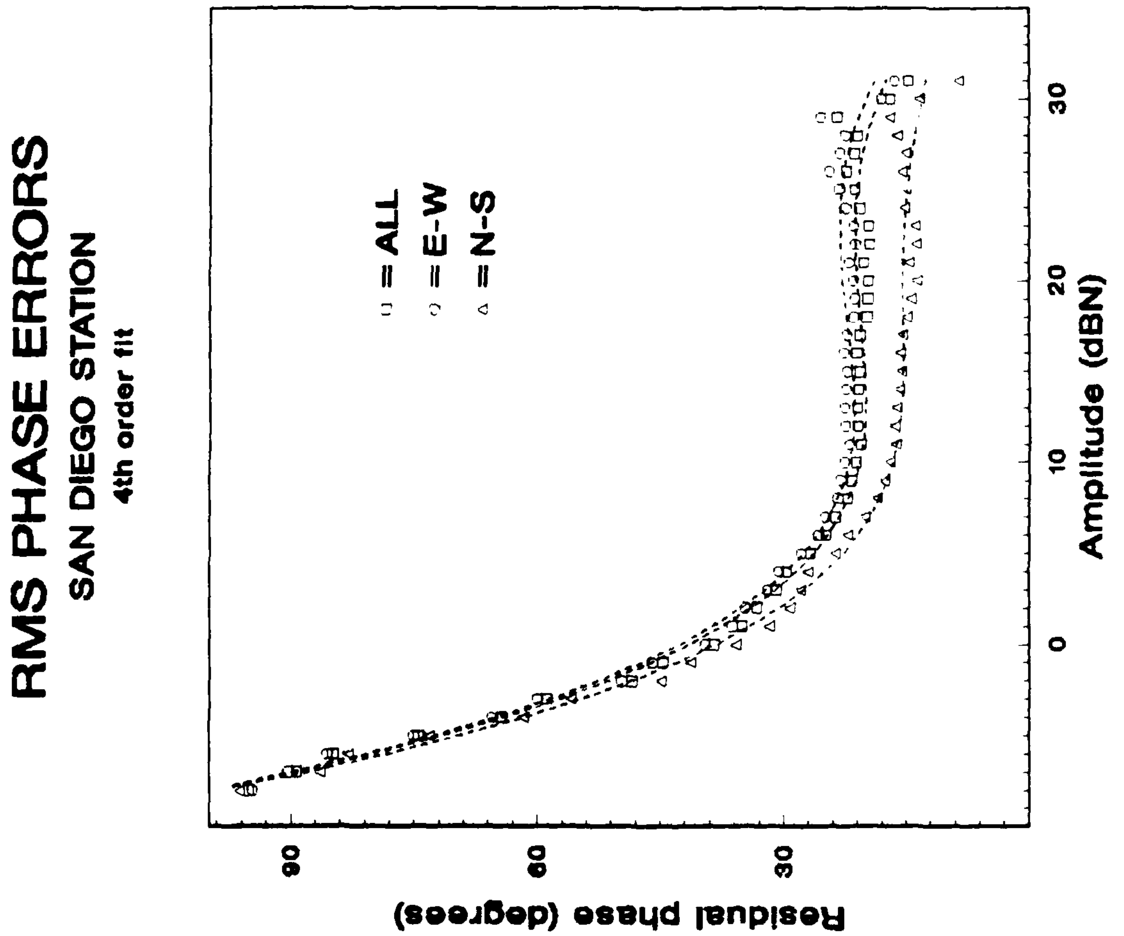

FIGURE 2: Plot of the distribution of the RMS phase difference errors vs. amplitude for the San Diego receiving station as determined from a least squares fit to four hours of NAVSPASUR data. The dashed lines indicate fourth order polynomial fits to the data.

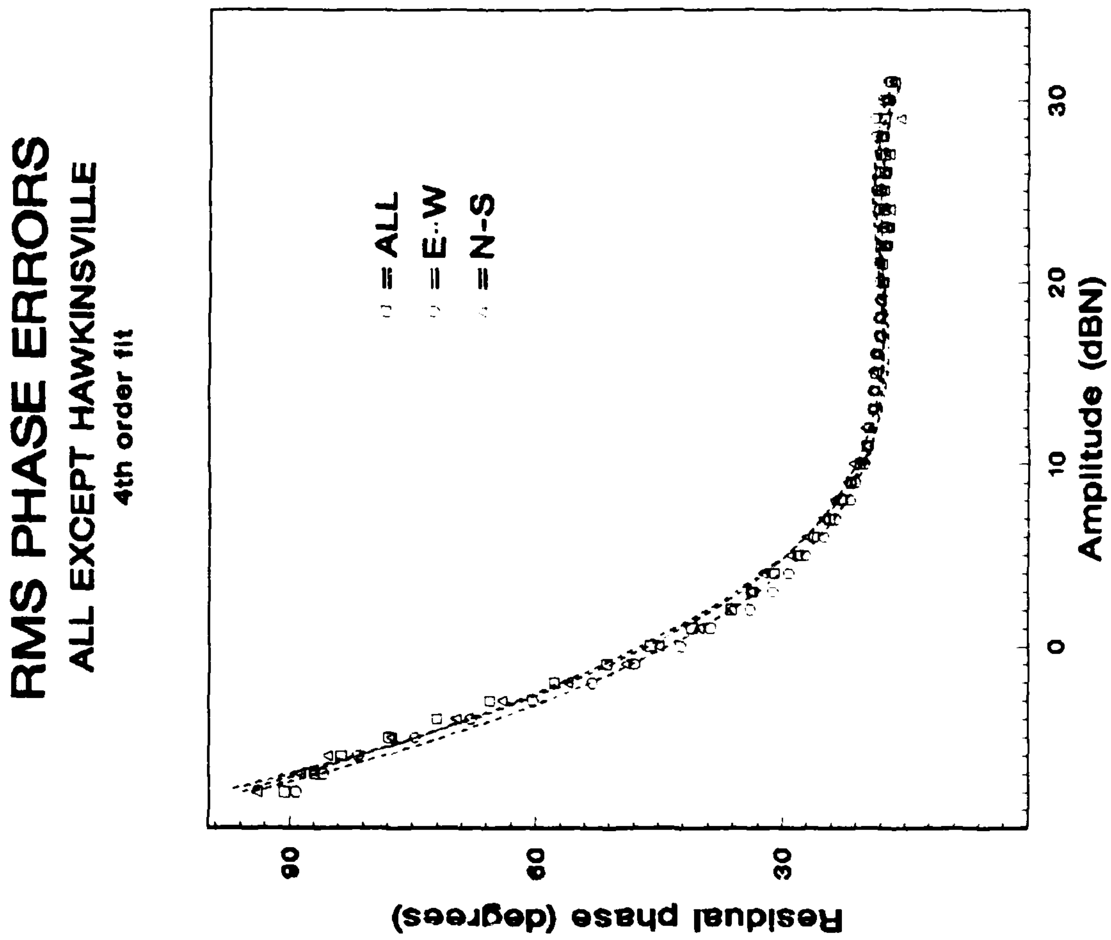

FIGURE 3: Plot of the RMS phase difference error distribution vs. amplitude for all receiving stations combined, except Hawkinsville, based on a least squares fit to four hours of NAVSPASUR data. The dashed lines indicate fourth order polynomial fits to the data.

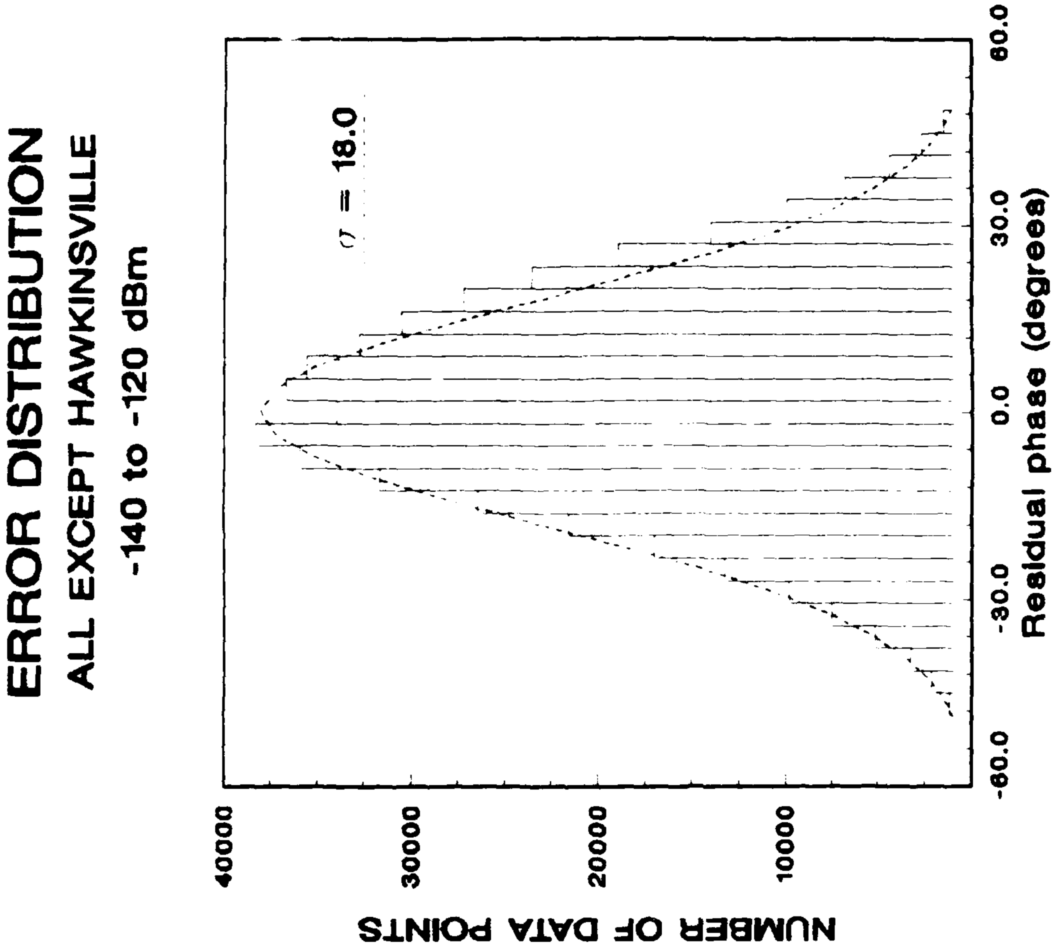

FIGURE 4: The distribution of the residual phase difference errors for data in the range -145 to -120 dBm from all receiving stations except Hawkinsville. The data were derived from least squares fits to four hours of NAVSPASUR data. The dashed curve shows a Gaussian distribution of dispersion 9.0 degrees and mean 0.0 degrees.

```
$!                     COMMAND PROCEDURE NEWLSQ
$!
$!     This command procedure executes the single station least squares fitting
$!  routine NEWLSQ.  The I/O assignment statements are contained in the COM file
$!  FORASS.  The parameter P1 specifies the input data set name (such as T5321,
$!  etc.) which is to be processed.  The control flags for the program are in
$!  NEWLSQ.INP, which should be edited prior to running.
$!
$ @space:[wadiak.lsq.data]forass 'p1'
$ assign space:[wadiak.lsq.data]'p1'.log for006
$ r space:[wadiak.lsq]newlsq
```

```
C
C ***********************************************************************
C *                                                                     *
C *                       PROGRAM NEWLSQ                                *
C *                                                                     *
C ***********************************************************************
C
C  AUTHORS:    Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:       19-FEB-1988
C  LANGUAGE:   FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:       VX7770::SPACE:[WADIAK.LSQ]NEWLSQ.FOR
C
C  SUBROUTINES CALLED:  CALC  - Calculates the expected phase difference
C                               for an antenna pair at the time of the
C                               current data line.
C                       DERIV - Calculates the phase partial derivative
C                               for an antenna pair at the time of the
C                               current data line.
C                       NRMEQ - Increments the normal equations for each
C                               datum.
C                       SYMIN - Solves the normal equations via matrix
C                               inversion.
C                       MATPR - Prints the covariance matrix to the
C                               output file.
C
C  USER-DEFINED         ERR - Computes the data weights based on
C  FUNCTIONS CALLED:          the received amplitude and the
C                             assumed error model.
C                       IEL - Calculates the one dimensional
C                             array index corresponding to each
C                             matrix element.
C
C
C
C  COMPILE INSTRUCTIONS:        $ FORTRAN NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:      $ LINK NEWLSQ
C
C  PROGRAM DESCRIPTION:
C
C   This is a specialized version of a general recursive nonlinear Least
C   Squares fitting routine which does a five parameter fit to NAVSPASUR
C   observations at each station.
C
C   The "observed" quantity will be differenced phase.  For this
C   application, the reference antenna used is the same as that used by
C   NAVSPASUR in their data reduction (either #4 or #5 depending on the
C   receiving station.  The phase differences (in units of rotations)
C   for the non-reference antennas are pre-computed in DIF.FOR from the
C   NAVSPASUR data.
C
C   The five parameters which can be fit are the satellite  range, the
C   angle in the plane of the great circle (with + to the East), the
C   angle normal to the great circle plane (with + to the North), and
C   the rates of change of these angles.
C
C   (Almost) everything will be passed to and from subroutines in the
C   common block, which also contains the array dimensions and which
C   sets all real vaiables as double precision.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C  A full discussion of the algorithm is contained in the report THE
C  DETERMINATION OF NAVSPASUR PHASE ERRORS, dated 16 March 1988.  An
```

```
C  outline of the major steps in the algorithm is given below.
C
C    1. Input the station locations and antenna array coordinates.
C
C    2. Read in control information which specifies which parmeters
C       are to be varied, maximum number of iterations, convergence
C       limit, etc.
C
C    3. Read in the data scan header and phase difference data.
C
C    4. Calculate initial estimates for R0, THTA0, PHI0, RDOT, PDOT,
C       and TDOT (range and direction angles and their respective time
C       derivatives - hereafter called the parameters).  These estimates
C       will be the starting point for the first least squares iteration.
C
C    5. Loop through the data, first by data time line (II), then by
C       antenna (I).
C
C       5a. Calculate the expected value of the differential phase from
C           the antenna positions and the current estimates of the
C           parameters (subroutine CALC).
C
C       5b. Calculate the residual phase for each data point by sub-
C           traction.
C
C       5c. Calculate the partial derivatives of the expected phase with
C           respect to each of the parameters (subroutine DERIV).
C
C       5d. Increment the normal equations using data weighting and the
C           error model given in function ERR.  See the above-referenced
C           report, equations (5) and (8) for the form of the normal
C           equations.
C
C       5e. END OF BOTH LOOPS
C
C    6. Solve the normal equations via standard matrix inversion
C       techniques.  The quantities obtained from the solution are the
C       corrections to the first estimates of the parameters.
C
C    7. Check whether the solution meets the convergence test or if the
C       maximum number of iterations have been performed.
C
C    8. IF no convergence AND IF not maximum number of iterations,
C       calculate new estimates of the parameters to be used for the
C       next least squares iteration.  Return to step 5 and do another
C       iteration.  ELSE, proceed to step 9.
C
C    9. IF maximum number of iterations AND IF no convergence, write
C       information to a dump file for later analysis.  Proceed to
C       step 3 and begin a new data scan.
C
C   10. IF convergence, calculate the final residuals on the data and
C       write residuals and summary info to output files for further
C       analysis.  Proceed to step 3 and begin a new data scan.
C
C
C  INPUTS (EXPLICIT):
C
C              FOR008 - scan header and observational data (differenced phase)
C                       in file *.DIF [file created as DIF.FOR output].
C                       Specific variables read in are:
C
C                         ISAT   - satellite number.
C                         ISTATN - receiving station number.
C                         ITRAN  - transmitter number.
C                         IDOP   - Doppler frequency in Hz.
```

```
C                   RSEC   - data scan start time.
C                   TPRED  - predicted fence crossing time.
C                   NLINES - number of phase difference data lines.
C
C                   X :  predicted geocentric x, y, z
C                   Y :  coordinates of the satellite
C                   Z :  at time TPRED.
C
C                   VX :  predicted satellite velocity components
C                   VY :  in geocentric x, y, z coordinate system
C                   VZ :  at time TPRED.
C
C                   DATA(I,J) - differenced phase for the Jth antenna
C                               at the time of the Ith data line.
C
C            FOR013 - antenna positions and station coordinates in file
C                     ANTPOS.INP.  Specific variables read in are:
C
C                   POS(KK,II,1) - east offset of the IIth antenna at
C                                  the KKth receiver site, in feet.
C                   POS(KK,II,2) - north offset of the IIth antenna at
C                                  at the KKth receiver site, in feet.
C
C                   STRX(KK) :  x, y, z coordinates of the reference
C                   STRY(KK) :  antenna at the KKth receiver station
C                   STRZ(KK) :  in units of earth radii.
C
C                   STHX(KK) :  projection of local "up" unit vector
C                   STHY(KK) :  of the KKth receiver station into the
C                   STHZ(KK) :  geocentric x, y, z coordinate system.
C
C                   STEX(KK) :  projection of local "east" unit vector
C                   STEY(KK) :  of the KKth receiver station into the
C                   STEZ(KK) :  geocentric x, y, z coordinate system.
C
C                   STNX(KK) :  projection of local "north" unit vector
C                   STNY(KK) :  of the KKth receiver station into the
C                   STNZ(KK) :  geocentric x, y, z coordinate system.
C
C        FOR014 - parameter flags, etc. in NEWLSQ.INP.  Specific variables
C                 read in are:
C
C                   IFLAG(1-6) - flags to control which parameters are
C                                solved for. 1 => solve for. Choices are:
C                                (1) => R0 (range)
C                                (2) => THTA0 (N-S angle)
C                                (3) => PHI0 (E-W angle)
C                                (4) => TDOT (N-S angle rate)
C                                (5) => PDOT (E-W angle rate)
C                                (6) => RDOT (range rate)
C
C                   ITMAX  - maximum number of iterations.
C                   FRACTC - convergence test limit.
C                   NSCANS - number of scans to process (0 => all).
C                   NSNGL  - receiver station # to process (0 => all).
C
C  INPUTS (IMPLICIT):
C
C            FTTOM  - conversion factor from feet to meters.
C            REARTH - radius of the earth in meters.
C
C  OUTPUTS  EXPLICIT:   FOR016-21 - indvidual station results in *.ST1-6
C                       FOR022-27 - final residuals in *.RS1-6
C                       FOR028 - residuals for non-convergent scans in *.DMP
C
C          IMPLICIT:   NONE
```

```
C
C  MAJOR VARIABLES:
C
C                ADISP(11) - final residual dispersions for each antenna.
C                ALHS(15) - vector containing the matrix elements of left hand
C                        side of the normal equations. (function IEL converts).
C                ARMEAN(11) - final residual means for each antenna.
C                ARRMS(11) - final residual rms's for each antenna.
C                CORR(15) - vector containing elements of the covariance matrix.
C                DATA(605) - differenced phase data for all antennas/timelines.
C                DELTA - RMS error of the data point currently being processed.
C                ERROR(605) - RMS error associated with each datum.
C                IAMP(55) - received amplitude for each data line, in dBm.
C                IDOP - received Doppler shift (+ => blueshift).
C                ISAT - satellite catalog number.
C                ISTATN - receiving station number (1-6).
C                ITRAN - transmitter number (7-9).
C                IFLAG(6) - flags to specify which parameters are solved for.
C                IVP(6) - array holding control flags.
C                LU - fortran unit number of the *.ST* output file.
C                NANT - number of antenna pairs at the current receiver site.
C                NLINES - number of data lines in the current scan.
C                NSNGL - ID # of receiver station to process (1-6 or 0 => all).
C                NSCANS - number of data scans to process (0 => all).
C                NVP - number of parameters to be varied.
C                PAR(6) - current values of the 6 parameters at time RSEC.
C                PARTL(6) - current values of the partial derivatives.
C                PDOT - a priori value of N-S direction cosine rate.
C                PHIO - a priori value of N-S direction cosine.
C
C                PN:
C                PE: satellite position at time RSEC in local station coords.
C                PH:
C
C                POS(6,12,2) - antenna position offsets from station center.
C                RO - a priori value of satellite range.
C                RDOT - apriori value of satellite range rate.
C                REARTH - equatorial earth radius, in meters.
C                RESID(605) - phase residuals (real-ideal) for each datum.
C                RHS(6) - right-hand-side of the normal equations. (after sol'n
C                         of the normal equations, contains the corrections)
C                RSEC - data scan start time.
C                S(6) - square roots of the L.H.S. of the normal equations.
C                SIGM(6) - expected errors in the parameters.
C
C                STAX:
C                STAY:  receiver geocentric x, y, z coordinates.
C                STAZ:
C
C                TDOT - a priori value of E-W direction cosine rate.
C                THTAO - a priori value of E-W direction cosine.
C                TPRED - satellite predicted fence crossing time.
C
C                VN:
C                VE: satellite velocity at time RSEC in local station coords.
C                VH:
C
C                VX:
C                VY:  satellite velocity at time TPRED in geocentric coordinates.
C                VZ:
C
C                WVLN - wavelength of the received signal, in meters.
C
C                X:
C                Y:  satellite position at time TPRED in geocentric coordinates.
C                Z:
```

```
C
C
C  COMMON BLOCKS UTILIZED:
C
C              COMMON - contained in VX7770::SPACE:[WADIAK.LSQ]NEWLSQ.CMN
C
C  MODIFIED:
C
C
C***************************************************************************
C
C
        INCLUDE 'SPACE:[WADIAK.LSQ]NEWLSQ.CMN/LIST'
C
C   Input the antenna positions as a 6 X 12 X 2 array.  The first arguement
C   is the station number, the second argument the antenna number, and the
C   third argument the antenna coordinate in the local frame (1 => "EAST"
C   along the great circle; 2 => "NORTH" perpendicular to the great circle).
C   This array contains the displacements, in feet, from the reference
C   antenna.  Convert from feet to meters.
C
C   Read in the station coordinates (in earth radii), and the XYZ
C   projections of the unit "UP", "EAST", and "NORTH" vectors.
C
        FTTOM= .3048D0
          DO KK=1,6
            IF(KK.EQ.2.OR.KK.EQ.5) THEN
              NANT = 10
             ELSE
              NANT = 11
            ENDIF
            DO II=1,NANT
              READ(13,*) (POS(KK,II,K),K=1,2)
              POS(KK,II,1) = FTTOM*POS(KK,II,1)
              POS(KK,II,2) = FTTOM*POS(KK,II,2)
            ENDDO
            READ(13,*) STRX(KK),STRY(KK),STRZ(KK)
            READ(13,*) STHX(KK),STHY(KK),STHZ(KK)
            READ(13,*) STEX(KK),STEY(KK),STEZ(KK)
            READ(13,*) STNX(KK),STNY(KK),STNZ(KK)
            READ(13,*)
          ENDDO
        CLOSE (UNIT=13)
C
C   Read in integer flags to identify which parameters will be varied.
C
        READ(14,*) ( IFLAG(K),K=1,5)
C
C   Read in the maximum number of iterations and the convergence limit.
C   Iteration ends when ITMAX iterations completed or all changes are
C   less than FRACT times the error.
C
        READ(14,*)ITMAX,FRACTC
C
C   Read in the number of scans to process (NSCANS) and the station
C   number of the station to process (NSNGL).
C              NSCANS=0 => do all scans in the data base.
C               NSNGL=0 => do all stations.
C
        READ(14,*) NSCANS,NSNGL
        CLOSE (UNIT=14)
        IF(NSCANS.EQ.0) NSCANS = 10000
C
C   Do some bookeeping stuff so we can keep track of things.
C   NVP and the array IVP will be used to control things.
C
```

```
      NVP=0
      NPAR=5
        DO I=1,NPAR
          NVP=NVP + IFLAG(I)
         ENDDO
        DO I=1,NVP
          IVP(I)=0
         ENDDO
      JJJ=0
        DO I=1,NPAR
         IF(IFLAG(I).EQ.1) JJJ=JJJ+1
         IF(IFLAG(I).EQ.1) IVP(JJJ)=I
        ENDDO
C
C   Read in the data.  There will be NLINES of calibrated differenced
C   phases in units of rotations.  Load up the DATA and ERROR arrays.
C   The errors will be  determined by the amplitude and calculated in
C   function ERR.
C
     DO WHILE (NSCANS.GT.0)
301   READ(8,*,END=302)ISAT,ISTATN,ITRAN,IDOP,RSEC,TPRED,NLINES
      READ(8,*) X,Y,Z,VX,VY,VZ
        IF(ISTATN.EQ.2.OR.ISTATN.EQ.5) THEN
            NANT = 10
          ELSE
            NANT = 11
        ENDIF
        DO I=1,NLINES
          III = NANT * (I - 1)
          READ(8,*) IAMP(I),(DATA(J),J=III+1,III+NANT)
          DIVN=ERR(IAMP(I) )
            DO J=III+1,III+NANT
              ERROR(J)=DIVN
            ENDDO
        ENDDO
      READ(8,*)
C
C  Check whether the scan is to be processed.  If so, decrement NSCANS.
C
      IF(NSNGL.NE.0.AND.NSNGL.NE.ISTATN) GOTO 666
      NSCANS = NSCANS - 1
C
C  Set the output file designator (LU).  Here, we're set up to write
C  each station's output to a different file.
C
      LU = 15 + ISTATN
      WRITE(LU,406) ISAT,TPRED,NLINES,ISTATN,ITRAN
 406  FORMAT(' SATELLITE #',I6,6X,'TPRED',F10.3,I6,' DATA LINES',/,
     +          ' RECEIVING STATION #',I2,9X,'TRANSMITTER #',I2)
C
C   Calculate the station position in geocentric coordinates.
C   Here, REARTH is the equatorial radius of the earth in meters.
C
      REARTH = 6378135.0
      STAX = STRX(ISTATN) * REARTH
      STAY = STRY(ISTATN) * REARTH
      STAZ = STRZ(ISTATN) * REARTH
C
C  Calculate the wavelength of the received radiation.  The following
C  adopt the NAVSPASUR doppler convention of + => BLUESHIFT.
C
      WVLN = 2.997925D8/(216.98D6 + IDOP)
C
C   Use the a priori position and velocity to calculate the rotation angles
C   and distance at time RSEC, which is the time of the first data line.
C   SX, SY, and SZ give the satellite position relative to the receiving
```

```
C   station in geocentric (XYZ) coordinates.  PN, PE, and PZ give the
C   satellite position relative to the station in station (NEH) coordinates.
C
       TDIF = RSEC - TPRED
       X0 = X + TDIF*VX
       Y0 = Y + TDIF*VY
       Z0 = Z + TDIF*VZ
       SX = X0 - STAX
       SY = Y0 - STAY
       SZ = Z0 - STAZ
       PN = STNX(ISTATN)*SX + STNY(ISTATN)*SY + STNZ(ISTATN)*SZ
       PE = STEX(ISTATN)*SX + STEY(ISTATN)*SY + STEZ(ISTATN)*SZ
       PH = STHX(ISTATN)*SX + STHY(ISTATN)*SY + STHZ(ISTATN)*SZ
       VN = STNX(ISTATN)*VX + STNY(ISTATN)*VY + STNZ(ISTATN)*VZ
       VE = STEX(ISTATN)*VX + STEY(ISTATN)*VY + STEZ(ISTATN)*VZ
       VH = STHX(ISTATN)*VX + STHY(ISTATN)*VY + STHZ(ISTATN)*VZ
       RHO = DSQRT( PE*PE + PH*PH )
       R0 = DSQRT( SX*SX + SY*SY + SZ*SZ )
       PHI0 = ASIN(PN/R0)
       THTA0 = ATAN2(PE,PH)
       RDOT = ( SX*VX + SY*VY + SZ*VZ ) / R0
       PDOT = ( VN - VE*SIN(THTA0)*SIN(PHI0) -
     +                   VH*COS(THTA0)*SIN(PHI0) ) / R0
       TDOT = ( VE*COS(THTA0) - VH*SIN(THTA0) ) / RHO
C
C   Convert distances to kilometers for output.
C
       SXKM = SX / 1.D3
       SYKM = SY / 1.D3
       SZKM = SZ / 1.D3
       PNKM = PN / 1.D3
       PEKM = PE / 1.D3
       PHKM = PH / 1.D3
       R0KM = R0 / 1.D3
C
C  Set the parameters equal to their predicted values based on the
C  catalog data.  This will be the initial estimate of the parameters
C  for the nonlinear least squares fitting.
C
       PAR(1) = R0
       PAR(2) = THTA0
       PAR(3) = PHI0
       PAR(4) = TDOT
       PAR(5) = PDOT
       PAR(6) = RDOT
C
C   Start the Least Squares fitting.
C
       ITER=0
 303   ITER=ITER + 1
C
C  This is an iterative process so it is NECESSARY to initialize some
C  arrays on each pass.
C
       K=IEL(NVP,NVP)
         DO I=1,K
           IF(I.LT.NVP+1) THEN
             RHS(I)=0D0
             PARTL(I)=0D0
            END IF
           ALHS(I)=0D0
          END DO
C
C   The work is all done in the subroutines.  The subroutines CALC
C   and DERIV are problem specific i.e., the nature of the observables
C   must be considered.  The remaining subroutines are general.  They
```

```
C   are used to set-up and invert the matrices and display the results.
C
      ICNT=0
      DO 105 II=1,NLINES
      DO 106 I=1,NANT
      ICNT=ICNT + 1
C
C   Call CALC to calculate the "ideal" value of the phase difference,
C   based on the "current" estimate of the satellite position.  This
C   difference is returned as CLC.  Set the residual to (real - ideal).
C
      CALL CALC(II,I)
      VRBL=DATA(ICNT) - CLC
 30   IF(VRBL.GT. .5D0) VRBL= VRBL -1D0
      IF(VRBL.LT.-0.5D0)VRBL=1D0+VRBL
      IF( DABS(VRBL).GT. 0.5D0) GOTO 30
      RESID(ICNT) = VRBL
      IF( DABS( RESID(ICNT) ).GT. 0.5D0 ) THEN
        WRITE(6,*) ICNT,' IS A BIG RESIDUAL'
        END IF
C
C   Get the partial derivatives and increment the normal equations.
C   Write out the residual array before updating the parameter values.
C
      CALL DERIV(II,I)
      DELTA = ERROR(ICNT)
      CALL NRMEQ(ICNT)
 106  CONTINUE
      JJ = (II-1) * NANT
 105  CONTINUE
      NDATA=ICNT
C
C   Get the square roots of the diagonal elements and use those
C   values to scale the array which will insure numerical stability.
C
      DO I=1,NVP
      JJ = IEL(I,I)
      S(I) = DSQRT( ALHS(JJ) )
      IF(S(I).EQ.0D0) S(I)=1D0
      ENDDO
      DO 107 I=1,NVP
      RHS(I) = RHS(I)/S(I)
      DO 108 J=1,I
      K=IEL(I,J)
      ALHS(K) = ALHS(K)/( S(I)*S(J) )
 108  CONTINUE
 107  CONTINUE
C
C   Solve the normal equations in subroutine SYMIN.
C
      CALL SYMIN
C
C   Rescale the result to get back to the correct units.
C
      DO 109 I=1,NVP
      RHS(I) = RHS(I)/S(I)
      DO 110 J=1,I
      K=IEL(I,J)
      ALHS(K) = ALHS(K)/( S(I)*S(J) )
 110  CONTINUE
 109  CONTINUE
C
C   Calculate the correction matrix.
C
      DO I=1,NVP
      J=IEL(I,I)
```

```
        SIGM(I) = DSQRT( ALHS(J) )
        ENDDO
        DO 111 I=1,NVP
        DO 112 J=1,I
        IA = IEL(I,I)
        IB = IEL(J,J)
        A = DSQRT( ALHS(IA) )
        B = DSQRT( ALHS(IB) )
        K = IEL(I,J)
        CORR(K) = ALHS(K)/( A*B )
 112    CONTINUE
 111    CONTINUE
C
C     ANALYZE THE RESIDUALS BY FINDING THE LARGEST, SMALLEST, MEAN,
C     AND THE RMS.
C
      RMAX=-1.E30
        RMIN=1.E30
        RMEAN=0.D0
        RRMS=0.D0
      DO 113 I=1,NDATA
      AR=RESID(I)
      IF(AR.GT.RMAX) RMAX=AR
      IF(AR.LT.RMIN) RMIN=AR
      RMEAN=RMEAN + AR
  113 CONTINUE
      RMEAN=RMEAN/NDATA
      DO 114 I=1,NDATA
      AR=RESID(I) - RMEAN
      RRMS=RRMS + AR*AR
  114 CONTINUE
      RRMS=RRMS/NDATA
      RRMS=DSQRT(RRMS)
C
C     CALCULATE UPDATED PARAMETERS AND PRINT OUT THE RESULTS.
C
      NPARNC=0
        JFLG=0
      DO 115 I=1,NVP
      J=IVP(I)
      A=PAR(J) + RHS(I)
      PAR(J)=A
      FRACT=RHS(I)/SIGM(I)
      FRACT=DABS(FRACT)
      IF(FRACT.LT.FRACTC) GOTO 115
      JFLG=1
      NPARNC=NPARNC + 1
  115 CONTINUE
      IF(JFLG.EQ.0.OR.ITER.EQ.ITMAX) GOTO 304
      GOTO 303
  304 WRITE(LU,214) JFLG,FRACTC,ITER,ITMAX,NPARNC
  214 FORMAT(' JFLG = ',I1,' FRACTC = ',F4.2,' ITER = ',I2,
     +          ' ITMAX = ',I2,1X,I2,' parameters not converged')
C
C  If convergence is achieved, write out the covariance matrix and
C  print the final means and RMS's.  Also, write the final residuals
C  (with headers) to the individual station *.RS1-6 files.  If no
C  convergence, write the residuals to the *.DMP file (dump file).
C  LR is the output file designator for the residuals.
C
      IF(JFLG.NE.0) THEN
        LR = 28
        WRITE(LU,*)
       ELSE
        LR = 21 + ISTATN
        WRITE(LU,215)
```

```
 215    FORMAT(' The correlation matrix is:')
        CALL MATPR(CORR,1,NVP,LU)
        ICNT=0
          DO II=1,NLINES
            DO I=1,NANT
              ICNT=ICNT + 1
              CALL CALC(II,I)
              VRBL=DATA(ICNT) - CLC
 20           IF(VRBL. GT. .5D0) VRBL = VRBL -1D0
              IF(VRBL. LT. -.5D0) VRBL = VRBL + 1D0
              IF( DABS(VRBL) . GT. .5D0) GOTO 20
              RESID(ICNT)=VRBL
             ENDDO
           ENDDO
C
C     Fnd and print the largest, smallest, mean, and RMS residuals.
C
        RMAX=-1.E30
        RMIN=1.E30
        RMEAN=0.D0
        RRMS=0.D0
          DO I=1,NDATA
            AR=RESID(I)
            IF(AR.GT.RMAX) RMAX=AR
            IF(AR.LT.RMIN) RMIN=AR
            RMEAN=RMEAN + AR
          ENDDO
        RMEAN=RMEAN/NDATA
          DO I=1,NDATA
            AR=RESID(I) - RMEAN
            RRMS=RRMS + AR*AR
          ENDDO
        RRMS=RRMS/NDATA
        RRMS=DSQRT(RRMS)
C
C   Write out the overall mean and rms.
C
        WRITE(LU,281) RMAX,RMIN,RMEAN,RRMS
  281 FORMAT(' Residuals after convergece:',/,' MAX  = ',
     +      F7.4,' MIN = ',F7.4,/,' MEAN = ',F7.4,' RMS = ',F7.4)
C
C  Write out the initial and final parameter values and the error
C  estimates for each parameter.  Before doing this, we must "unpack"
C  the error array.
C
        IMARK = 0
        DO IJ=5,1,-1
          IF(IFLAG(IJ).EQ.0) THEN
              SIGM(IJ) = 0.
            ELSE
              SIGM(IJ) = SIGM(NVP-IMARK)
              IMARK = IMARK + 1
          ENDIF
        ENDDO
        WRITE(LU,418)
  418  FORMAT(' Initial, final, and error values for each parameter:',
     +   /,7X,'RANGE',8X,'THETA',9X,'PHI',7X,'dTHETA/dt',5X,'dPHI/dt',
     +   5x,'dRANGE/dt')
        WRITE(LU,414) R0,THTA0,PHI0,TDOT,PDOT,RDOT
        WRITE(LU,414) PAR
        WRITE(LU,414) SIGM
C
C   Calculate and write out the individual antenna means,
C   RMS's, and dispersions.
C
        DO L=1,NANT
```

```
          ARMEAN(L) = 0.D0
          ARRMS(L) = 0.D0
          ADISP(L) = 0.D0
        ENDDO
        NUSED = 0
        DO I=1,NLINES
          IF(IAMP(I).LE.144) THEN
            NUSED = NUSED + 1
            DO L=1,NANT
              ARMEAN(L) = ARMEAN(L) + RESID(NANT*(I-1)+L)
              ARRMS(L) = ARRMS(L) + RESID(NANT*(I-1)+L)**2
            ENDDO
          ENDIF
        ENDDO
        DO L=1,NANT
            ARMEAN(L) = ARMEAN(L) / NUSED
            ARRMS(L) = DSQRT( ARRMS(L) / NUSED )
        ENDDO
        DO I=1,NLINES
          IF(IAMP(I).LE.144) THEN
            DO L=1,NANT
              ADISP(L) = ADISP(L) + (RESID(NANT*(I-1)+L)-ARMEAN(L))**2
            ENDDO
          ENDIF
        ENDDO
        DO L=1,NANT
          ADISP(L) = DSQRT( ADISP(L) / NUSED)
        ENDDO
        WRITE(LU,404) NUSED,(ARMEAN(J),J=1,NANT)
 404    FORMAT(' Means for',I3,' amplitudes > -145dbmW:',/,4X,11F7.3)
        WRITE(LU,407) NUSED,(ARRMS(J),J=1,NANT)
 407    FORMAT(' RMS for',I3,' amplitudes > -145 dbmW:',/,4X,11F7.3)
        WRITE(LU,411) NUSED,(ADISP(J),J=1,NANT)
 411    FORMAT(' Disp for',I3,' amplitudes > -145 dbmW:',/,4X,11F7.3)
        WRITE(LU,408)
 408    FORMAT(/)
      ENDIF
C
C  Write out the final residuals to either the *.RS* file or *.DMP file.
C  If any one antenna mean exceeds +-0.2, dump the scan.
C
        DO JJ=1,NANT
          IF(DABS(ARMEAN(JJ)).GE.0.2) LR = 28
        ENDDO
        WRITE(LR,412) ISAT,ISTATN,ITRAN,IDOP,RSEC,TPRED,NLINES
 412    FORMAT(1X,4I7,2F10.3,3X,I4)
        WRITE(LR,413) X,Y,Z,VX,VY,VZ
 413    FORMAT(1X,6F13.2)
        WRITE(LR,414) R0,THTA0,PHI0,TDOT,PDOT
        WRITE(LR,414) PAR
 414    FORMAT(1X,F13.0,4F13.8,F13.2)
        WRITE(LR,417) SIGM
 417    FORMAT(1X,5E13.4)
          DO L=1,NLINES
            LL= (L-1) * NANT
            WRITE(LR,282) IAMP(L), ( RESID(J),J=LL+1,LL+NANT )
 282        FORMAT(1X,I3,11F7.3)
           ENDDO
        WRITE(LR,*)
 666    CONTINUE
      ENDDO
 999    STOP
 302    WRITE(6,*) 'EOF ENCOUNTERED AT START OF NEW READ'
        STOP
        END
        INCLUDE 'SPACE:[WADIAK.LSQ]CALC.FOR/LIST'
```

```
INCLUDE 'SPACE:[WADIAK.LSQ]DERIV.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ]ERR.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ]IEL.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ]MATPR.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ]NRMEQ.FOR/LIST'
INCLUDE 'SPACE:[WADIAK.LSQ]SYMIN.FOR/LIST'
```

```
      SUBROUTINE  CALC(II,J)
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE  CALC                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHORS:      Dr. Michael D. Andrews, Dr. E  James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ]CALC.FOR
C
C  CALLING ROUTINES:             NEWLSQ.FOR, DERIV.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:               NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This program will calculate the theoretical value of the phase difference
C   between the reference antenna and any other antenna at a given NAVSPASUR
C   observing station, based on the satellite's current position.  The coor-
C   dinate system used here is the local station coordinate system with the
C   reference antenna at the origin.  The axes are local east and north, both
C   tangent to the geoid, and local height, or "up", normal to the geoid.
C
C   All the necessary data is passed in the common block with the exception
C   of II, the data line number, and J, the antenna number.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Obtain the X, Y coordinates of the Jth antenna with respect to the
C       reference antenna at the current receiver site.
C
C    2. Obtain the current estimates of the satellite position and velocity
C       at time RSEC in a station-based polar coordinate system.
C
C    3. Calculate the time difference TINCR between RSEC and the time of the
C       IIth data line.
C
C    4. Calculate the satellite position at the time of the IIth data line
C       in local station cartesian coordinates.
C
C    5. Calculate the distance D to the satellite from the Jth antenna.
C
C    6. Calculate the differential path length between the satellite and the
C       Jth antenna and reference antenna, respectively.  Convert from meters
C       to wavelengths, or rotations.  Express the result modulo 1, since
C       the phase measurement is ambiguous as to the number of integral phase
C       rotations present.  Express the result in the range -0.5 to +0.5
C       rotations.
C
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C                          II - data line number.
C                          J  - antenna number.
C
C           IMPLICIT (via COMMON block):
```

```
C
C                   ISTATN - ID number of the receiver site.
C                   PAR(1-6) - satellite position and velocity at RSEC.
C                   POS(ISTATN,J,1-2) - X, Y coordinates of Jth antenna
C                                       at receiver number ISTATN.
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (via COMMON block):
C
C                   CLC - calculated theoretical phase difference
C
C  MAJOR VARIABLES:
C
C               ANGP - N-S angle at time of the data line.
C               ANGT - E-W angle at time of the data line.
C               D - distance to the satellite from the Jth antenna
C                   at the time of the data line.
C
C               DX:  X, Y, Z distances to the satellite
C               DY:  relative to the Jth antenna at the
C               DZ:  time of the data line.
C
C               ISTATN - ID number of the receiver site.
C               PAR(1-6) - satellite position, velocity at time RSEC (expressed
C                          in polar coordinates - passed in COMMON block).
C               POS(ISTATN,J,1-2) - X, Y coordinates of the Jth antenna at
C                                   receiver number ISTATN.
C               RANGE - distance to satellite at time of the data line.
C               R0 - distance to the satellite at time RSEC.
C               RDOT - range rate at time RSEC
C               PHI - N-S angle to the satellite at time RSEC.
C               PHIDOT - N-S angle rate at time RSEC.
C
C               SATX:
C               SATY:  position if the satellite at the time of the data line.
C               SATZ:
C
C               T - time of the data line relative to RSEC.
C               TINCR - time increment between data lines (nominally 1/55 sec).
C               THDOT - E-W angle rate at time RSEC.
C               THETA - E-W angle to the satellite at time RSEC.
C               X - E-W coordinate of the Jth antenna.
C               Y - N-S coordinate of the Jth antenna.
C
C
C  MODIFIED:
C
C
C*****************************************************************************
C
C
C  Include the common block file.
C
      INCLUDE 'SPACE:[WADIAK.LSQ]NEWLSQ.CMN/LIST'
C
C  Get the X and Y coordinates of the Jth antenna at station ISTATN.
C
      X=POS(ISTATN,J,1)
      Y=POS(ISTATN,J,2)
C
C  Get the values of the variables out of the PAR array.
C
      RO = PAR(1)
      THETA = PAR(2)
      THDOT = PAR(4)
```

```
      PHI = PAR(3)
      PHIDOT = PAR(5)
      RDOT = PAR(6)
C
C  Calculate time since the start of the data scan and increment the
C  values of the direction angles and the range.
C
      TINCR = 1.0 / 54.98
      T = (II-.5) * TINCR
      ANGT = THETA + T * THDOT
      ANGP = PHI + T * PHIDOT
      RANGE = R0 + T * RDOT
C
C   Calculate the distance from the antenna to the satellite.
C
      SATX = RANGE * SIN(ANGT) * COS(ANGP)
      SATY = RANGE * 1D0       * SIN(ANGP)
      SATZ = RANGE * COS(ANGT) * COS(ANGP)
      DX = SATX - X
      DY = SATY - Y
      DZ = SATZ
      D = DSQRT( DX*DX + DY*DY + DZ*DZ )
C
C  The path difference is just RANGE-D.  Convert to rotations.
C  Adjust so the result is in the range -.5 to + .5.
C
      PATH =  RANGE - D
      CLC = DMOD( (PATH/WVLN),1D0 )
      IF(CLC.GT. 0.5D0) CLC = CLC - 1D0
      RETURN
      END
```

```
      SUBROUTINE DERIV(II,J)
C
C ************************************************************************
C *                                                                      *
C *                           SUBROUTINE DERIV                           *
C *                                                                      *
C ************************************************************************
C
C  AUTHORS:      Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ]DERIV.FOR
C
C  CALLING ROUTINE:               NEWLSQ.FOR
C
C  SUBROUTINES CALLED:            CALC.FOR
C
C  COMPILE INSTRUCTIONS:          compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:        linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:                NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This program uses a finite difference method to calculate the partial
C   derivatives of the phase difference between the Jth antenna and the
C   reference antenna.  Partial derivatives are calculated with respect
C   to each of the parameters being varied in the least-squares fitting.
C   The procedure is to vary one of the six position/velocity components
C   by + and - deltaPAR while holding the remaining parameters constant.
C   The change in the calculated phase difference between these two values
C   of the varied parameter, divided by twice deltaPAR, is the partial
C   derivative.
C
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1.  DO, for each parameter being varied in the least-squares fit;
C
C       1a. Get the current value of the parameter.  Save this value in a
C           holding variable B.  B will be used at the end of the routine
C           to reset the parameter to its original value.
C
C       1b. IF the parameter is currently equal to zero, set it equal to a
C           small number in order to avoid divide by zero errors later on.
C
C       1c. Increment and decrement the parameter by 0.001% and calculate
C           the theoretical phase differences for each value.
C
C       1d. Difference these two phase difference values and divide by
C           twice the incremental value.  Set the partial derivative to
C           this value.
C
C       1e. END of the DO loop.
C
C    2.  RETURN the values of the partial derivatives to the main program
C        via the COMMON statement.
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C               II - data line number.
C               J  - antenna number.
C
C           IMPLICIT (via COMMON block):
C
```

```
C              IVP(1-6) - parameter ID number for each parameter being varied.
C              NVP - total number of parameters being varied in the least
C                    squares fitting routine.
C              PAR(1-6) - satellite position and velocity.
C              PLS - phase difference for PAR+deltaPAR.
C
C
C OUTPUTS  EXPLICIT:   NONE
C
C          IMPLICIT (via COMMON block):
C
C              PARTL(1-6) - partial derivatives of the phase difference
C                           with respect to the varied parameters.
C
C MAJOR VARIABLES:
C
C              B - original parameter value.  Used to reset the parameter
C                  after the partial derivative is calculated.
C              PAR(1-6) - satellite position and velocity.
C              PARTL(1-6) - phase difference partials with respect to
C                           each of the varied parameters.
C
C
C MODIFIED:
C
C
C*****************************************************************************
              INCLUDE 'SPACE:[WADIAK.LSQ]NEWLSQ.CMN/LIST'
C
C  Calculate the partials.  Loop through and get the phase difference
C  partial with respect to each of the parameters which are being varied
C  in the least squares program.  The array IVP contains the parameter ID
C  numbers (1-6) corresponding to the PAR array elements of the varied
C  parameters.  NVP is the number of parameters being varied in the
C  least squares program.
C
        DO I=1,NVP
          K = IVP(I)
C
C  Hold the original value of the parameter in variable B.  Calculating
C  the partial derivatives requires varying the value of PAR.  B will be
C  used to restore PAR to its original value once we're done.
C
          B = PAR(K)
C
C  If the parameter whose partial is to be calculated equals zero, set its
C  value to 0.01 (meters or meters/sec as appropriate) to avoid zero-divide
C  errors later on.
C
          IF(PAR(K).EQ.0D0) PAR(K)=1D-2
          C = PAR(K)
C
C  Get the phase difference for PAR + deltaPAR.
C
          PAR(K) = C + C*1D-5
          CALL CALC(II,J)
          PLS = CLC
C
C  Get the phase difference for PAR - deltaPAR.
C
          PAR(K) = C - C*1D-5
          CALL CALC(II,J)
C
C  Calculate the partial derivative via finite differences.
C
          PARTL(K) = (PLS - CLC)/( C*2D-5 )
```

```
  PAR(K) = B
ENDDO
RETURN
END
```

```
      REAL*8 FUNCTION ERR(INT)
C
C ***********************************************************************
C *                                                                     *
C*                              FUNCTION ERR                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHORS:        Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:           23-NOV-1987
C  LANGUAGE:       FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:           VX7770::SPACE:[WADIAK.LSQ]ERR.FOR
C
C  CALLING ROUTINE:              NEWLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:               NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This function calculates the expected RMS error on a phase difference
C   datum based on the received signal strength of the datum.  The RMS errors
C   have been modelled as a fourth order polynomial in dB above the noise
C   floor.
C
C  INPUTS   EXPLICIT (arguments to FUNCTION statement):
C
C                INT - amplitude of the received signal.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (returned as FUNCTION value):
C
C                ERR(INT) - RMS error associated with the data point amplitude.
C
C  MAJOR VARIABLES:
C
C
C                NDB - signal strength in dB above assumed -152dBm noise floor.
C                A0 - zeroth order coefficient of polynomial.
C                A1 - first order coefficient of polynomial.
C                A2 - second order coefficient of polynomial.
C                A3 - third order coefficient of polynomial.
C                A4 - fourth order coefficient of polynomial.
C
C  MODIFIED:
C
C
C************************************************************************
                IMPLICIT REAL*8 (A-H,O-Z)
C
C  The following error model is based on the four-hour NAVSPASUR data
C  tape T5321.  The model was generated from the combined data for
C  all stations except station #5, with N-S and E-W baselines combined.
C
      NDB = 152 - INT
      A0 =  1.31D-1
      A1 = -1.23D-2
      A2 =  5.43D-4
      A3 = -4.48D-6
```

```
      A4 = -1.02D-7
      ERR = A4*NDB**4 + A3*NDB**3 + A2*NDB**2 + A1*NDB + A0
      RETURN
      END
```

```
      FUNCTION IEL(I,J)
C
C ***************************************************************************
C *                                                                         *
C *                             FUNCTION IEL                                *
C *                                                                         *
C ***************************************************************************
C
C  AUTHOR:        Dr. Michael D. Andrews
C  DATE:          23-NOV-1987
C  LANGUAGE:      FORTRAN ANSI-77  ( 'X/VMS operating system)
C  FILE:          VX7770::SPACE:[WADIAK.LSQ]IEL.FOR
C
C  CALLING ROUTINES:            NEWLSQ.FOR
C                               NRMEQ.FOR
C
C  SUBROUTINES CALLED:          NONE
C
C  COMPILE INSTRUCTIONS:        compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:      linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:              NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This function is a mapping routine designed to allow storage of
C   an inherently 2-dimensional matrix into a 1-dimensional vector.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C  INPUTS   EXPLICIT (arguments to FUNCTION statement):
C
C                I - row number of the matrix element.
C                J - column number of the matrix element.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT:   NONE
C
C           IMPLICIT (returned as FUNCTION value):
C
C                IEL(I,J) - vector index corresponding to matrix element (I,J)
C
C  MODIFIED:
C
C
C****************************************************************************
        IF(J.LE.I) THEN
            K = I
            L = J
          ELSE
            K = J
            L = I
        ENDIF
        IEL = (( K*K - K ) / 2 ) + L
        RETURN
        END
```

```
      SUBROUTINE MATPR(A,N,M,LU)
C
C ***********************************************************************
C *                                                                     *
C *                          SUBROUTINE MATPR                           *
C *                                                                     *
C ***********************************************************************
C
C  AUTHORS:      Dr. Michael D. Andrews, Dr. E. James Wadiak
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ]MATPR.FOR
C
C  CALLING ROUTINE:              NEWLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:               NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine prints out a lower triangular matrix which has been
C   previously stored in a vector array using subroutine IEL.
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C                A - vector array in which the matrix elements are stored.
C                LU - FORTRAN unit number to which output is directed.
C                M - dimension of the lower triangular matrix.
C                N - flag specifying which format to use on output.
C
C           IMPLICIT:   NONE
C
C  OUTPUTS  EXPLICIT:   matrix stored in array A is printed to output unit.
C
C           IMPLICIT:   NONE
C
C  MODIFIED:
C
C
C************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(78)
      DO  I=1,M
        IL=IEL(I,I)
        IG=IEL(I,1)
        IF(N.NE.1) THEN
          WRITE(LU,100) (A(J),J=IG,IL)
100       FORMAT(5(2X,E11.3),/)
        ELSE
          WRITE(LU,101) (A(J),J=IG,IL)
101       FORMAT(5(2X,F7.4))
        ENDIF
      ENDDO
      RETURN
      END
```

```
      SUBROUTINE NRMEQ(IOBS)
C
C ****************************************************************************
C *                                                                          *
C *                           SUBROUTINE NRMEQ                               *
C *                                                                          *
C ****************************************************************************
C
C  AUTHOR:       Dr. Michael D. Andrews
C  DATE:         23-NOV-1987
C  LANGUAGE:     FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:         VX7770::SPACE:[WADIAK.LSQ]NRMEQ.FOR
C
C  CALLING ROUTINE:              NEWLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  USER-DEFINED FUNCTIONS:       IEL (in IEL.FOR)
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:               NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine increments the normal equations of the nonlinear least
C   squares program NEWLSQ.  Each call to NRMEQ increments the normal
C   equations for one data residual.
C
C  PROGRAM ALGORITHM (PSEUDOCODE):
C
C    1. Get the error associated with the datum through the COMMON block.
C       Set the data weight GG equal to 1./(error)**2.
C
C    2. DO, for each varied parameter (i.e., NVP normal equations),
C
C       2a. Increment the right-hand-side by the weighted data residual
C           times the partial derivative with respect to the parameter.
C
C       2b. DO, for each lower triangular matrix element on the left-
C           hand-side of the normal equations,
C
C           2b(1). Increment the left-hand side by the weighted product
C                  of the partial derivatives associated with the respective
C                  row and column numbers.
C
C       2c. END of both loops.
C
C    3. RETURN to the main program.
C
C  INPUTS   EXPLICIT (arguments to CALL statement):
C
C               IOBS - number of the data residual being processed.
C
C           IMPLICIT (via COMMON block):
C
C               ALHS(1-16) - current left-hand-side of the normal equations.
C               DELTA - RMS error associated with the IOBSth data residual.
C               IVP(1-6) - array containing the parameter number of each of
C                          the NVP parameters being varied.
C               NVP - number of parameters being varied in least squares fit.
C               PARTL(1-6) - partial derivatives of the phase difference
C                            with respect to each of the varied parameters.
C               RHS(1-6) - current right-hand-side of the normal equations.
```

```
C
C  OUTPUTS  EXPLICIT: NONE
C
C           IMPLICIT (via COMMON block):
C
C               ALHS(1-16) - incremented left-hand-side of the normal equations.
C               RHS(1-6) - incremented right-hand-side of the normal equations.
C
C  MAJOR VARIABLES:
C
C               ALHS(1-16) - current left-hand-side of the normal equations.
C               DELTA - RMS error associated with the IOBSth data residual.
C               IOBS - ID number of the data residual being processed.
C               IVP(1-6) - array containing the parameter number of each of
C                          the NVP parameters being varied.
C               NVP - number of parameters being varied in least squares fit.
C               PARTL(1-6) - partial derivatives of the phase difference
C                            with respect to each of the varied parameters.
C               RHS(1-6) - current right-hand-side of the normal equations.

C
C  MODIFIED:
C
C
C*******************************************************************************
      INCLUDE 'SPACE:[WADIAK.LSQ]NEWLSQ.CMN/LIST'
      GG=1. / ( DELTA * DELTA )
      DO I=1,NVP
        IA = IVP(I)
        RHS(I) = RHS(I) + GG * RESID(IOBS) * PARTL(IA)
        DO J=1,I
          K = IEL(I,J)
          IB = IVP(J)
          ALHS(K) = ALHS(K) + GG * PARTL(IA) * PARTL(IB)
        ENDDO
      ENDDO
      RETURN
      END
```

```
      SUBROUTINE SYMIN
C
C ***********************************************************************
C *                                                                     *
C *                         SUBROUTINE SYMIN                            *
C *                                                                     *
C ***********************************************************************
C
C  AUTHOR:         Unknown
C  DATE:           09-JUL-1987 (last modification)
C  LANGUAGE:       FORTRAN ANSI-77  (VAX/VMS operating system)
C  FILE:           VX7770::SPACE:[WADIAK.LSQ]SYMIN.FOR
C
C  CALLING ROUTINE:              NEWLSQ.FOR
C
C  SUBROUTINES CALLED:           NONE
C
C  COMPILE INSTRUCTIONS:         compiled via INCLUDE statement in NEWLSQ
C
C  LINK/LOAD INSTRUCTIONS:       linked via INCLUDE statement in NEWLSQ
C
C  PARENT PROGRAM:               NEWLSQ.FOR
C
C  PROGRAM DESCRIPTION:
C
C   This subroutine is a working matrix inversion algorithm which dates
C   back to the early days of electronic computers.  We plan to replace
C   this program element someday with a modernized version.  Extensive
C   documentation of this element is therefore unwarranted, and it is
C   submitted as is.
C
C************************************************************************
        INCLUDE 'SPACE:[WADIAK.LSQ]NEWLSQ.CMN/LIST'
      DIMENSION PVCOL(27), PVROW(27),IUSE(27)
C
C     INITIALLY ALL ROWS ARE UNUSED
C
      DO 10 I=1,NVP
10    IUSE(I)=0
      DO 130 ITER=1,NVP
C     SEARCH FOR THE LARGEST DIAGONAL ELEMENT FOR USE AS A PIVOT
      DMAX=0.D0
      II=1
      DO 40 I=1,NVP
      DIAG=DABS(ALHS(II))
      IF(DIAG-DMAX) 40,40,20
20    IF(IUSE(I)) 30,30,40
30    DMAX=DIAG
      K=I
      PIVOT=ALHS(II)
40    II=II+I+1
      IF(DMAX)200,200,50
 50   IUSE(K)=1
C     GET THE ELEMENTS OF THE PIVOT ROW AND COLUMN
C     WITH THE PROPER SIGN
      IK=1+(K*(K-1))/2
      DO 120 I=1,NVP
      IF(I-K) 60,80,90
   60 CONTINUE
      PVROW(I)=ALHS(IK)
      PVCOL(I)=ALHS(IK)/PIVOT
      ALHS(IK)=0.D0
      IK=IK+1
      IF(IUSE(I)) 120,120,70
70    PVCOL(I)=-PVCOL(I)
      GO TO 120
```

```
80    PVROW(I)=1.D0
      PVCOL(I)=-1.D0/PIVOT
      ALHS(IK)=0.D0
      PVRWB=RHS(I)
      RHS(I)=0.D0
      GO TO 110
   90 CONTINUE
      PVROW(I)=ALHS(IK)
      PVCOL(I)=ALHS(IK)/PIVOT
      ALHS(IK)=0.D0
      IF(IUSE(I)) 110,110,100
100   PVROW(I)=-PVROW(I)
  110 IK=IK+I
120   CONTINUE
C     PERFORM THE PIVOT STEP
      IJ=1
      DO 130 I=1,NVP
      RHS(I)=RHS(I)-PVCOL(I)*PVRWB
      DO 130 J=1,I
      ALHS(IJ)=ALHS(IJ)-PVCOL(I)*PVROW(J)
130   IJ=IJ+1
      RETURN
200   WRITE(LU,210) ITER,N, (IUSE(I),I=1,N)
  210 FORMAT(' FAILURE TO FIND NON-ZERO DIAGONAL ELEMENT IN SYMIN ON
     .ITERATION  ',I3,' FOR N= ',I3,' IUSE = ',/,(1X,50I2))
      RETURN
      END
```

ATE

LMED

8