SPLINE SMOOTHING OF TWO-DIMENSIONAL DATA SERIES WITH PRECISION ESTIMATION APPLIED TO SATELLITE NAVIGATION

SEE HEAN QUEK

March 1983

GEODESY AND GEOMATICS ENGINEERING
UNB

TECHNICAL REPORT NO. 101

PREFACE

In order to make our extensive series of technical reports more readily available, we have scanned the old master copies and produced electronic versions in Portable Document Format. The quality of the images varies depending on the quality of the originals. The images have not been converted to searchable text.

SPLINE SMOOTHING OF TWO-DIMENSIONAL DATA SERIES WITH PRECISION ESTIMATION, APPLIED TO SATELLITE NAVIGATION

by

See Hean Quek

March, 1983

Reprinted July 1985

ABSTRACT

Noisy two-dimensional data series are a common occurrence in the geodetic field. As a result of this, numerous algorithms have been formulated to separate the systematic components from the noise within the observed data series. These algorithms, however, are not applicable to all types of data series. In this thesis, the performance of the piecewise cubic function, as a means of smoothing such data series, is investigated.

Cubic splines have, in the past, been used as smoothing algorithms. However, they proved ineffective in dealing with two-dimensional data series as they were developed with a weighting scheme for only one of the two variables. Hence they cannot accept a fully populated covariance matrix associated with each observed data point. The spline algorithm developed in this thesis uses both parameterized cubic splines and the method of least-squares to formulate a weighting scheme which allows the incorporation of the two-dimensional covariance matrices of the observations.

The resulting spline approximation technique is then used to smooth the navigation data sets of the three ice camps of the Lomonosov Ridge Experiment (LOREX) in the vicinity of the North Pole in 1979. The Navy Navigation Satellite

System (Transit) was used as the primary positioning system. Error models evaluating the accuracy of the position fixes using Transit satellites at high latitudes are developed. Smoothed positions and velocities for the three ice camps at one hour intervals are computed for the duration of the expedition.

To evaluate the performance of the spline algorithm, the smoothed data series produced by the spline algorithm (DSPLIN) and the real-time smoothing technique (SMOBS) used during LOREX are compared with those generated by the precise dynamic package (GEODOP), i.e. DSPLIN versus GEODOP and SMOBS versus GEODOP. The smoothed data series produced by the precise dynamic technique (GEODOP) are hence used as a reference standard. In the comparisons between the smoothed data series of positions and velocities for the three ice camps, a reduction of about 56 and 47 percent in the root mean square of the differences in position and velocity respectively, is achieved by DSPLIN over SMOBS. In the same comparisons, the maximum discrepancy between individual smoothed positions is reduced by about one-half (i.e. from 3758 m to 1564 m). The computed standardised position differences between the smoothed positions produced by the spline alogrithm and the precise dynamic technique shows that the precision estimates computed by the spline algorithm are consistent with accuracy only over certain periods of time in the LOREX data spans.

TABLE OF CONTENTS

TABLE OF CONTENTS - cont'd

TABLE OF CONTENTS - cont'd

LIST OF FIGURES

LIST OF FIGURES - cont'd

LIST OF FIGURES - cont'd

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis has been supervised by Professor David E. Wells, who not only provided the impetus for this research, but also aided in the acquisition of the LOREX navigation data sets and information pertaining to the three ice stations in the Arctic. I am very grateful for the many enlightening discussions held with Professor Wells during the course of this research.

In addition, I would like to thank Professor Petr Vanicek for his constructive and thought-provoking criticisms during the early stages of formulating the algorithm, Mr. Yong Qi Chen for all the interesting late night discussions, and Mr. Tom Inzinga and Mr. Derek Davidson for their assistance in preparing the written thesis. Continued assistance has been rendered by the faculty, staff and fellow graduates in the geodesy group here at U.N.B.

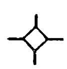
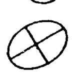
Financial assistance for this research has been made available through the teaching and research assistantships from the Department of Surveying Engineering, and from an operating grant from the Natural Sciences and Engineering Research Council of Canada, entitled "Arctic Marine Navigation Aids", held by Professor Wells.

Finally, I would like to dedicate this thesis to my parents. Words cannot express my gratitude for all that they have done.

NOTATION

:=		equal by definition
ϵ	:=	belongs to
$\subset$	:=	subset
$\cap$	:=	intersection
$s(x)$	:=	polynomial function
s^n	:=	n th derivative of function s
$\langle a,b \rangle$	:=	closed interval: $a \leq x \leq b$
S_k	:=	Spline Space of degree k
P_k	:=	Polynomial Space of degree k
$C^n\langle a,b \rangle$	:=	Space of n th differentially continuous functions

SYMBOLS

	:=	data point
	:=	knot point
	:=	confidence ellipse
	:=	spline

In general

1. Matrices and vectors are underlined (e.g. $\underline{A}$ and $\underline{x}$).
2. Matrices are denoted by upper case letters (e.g. $\underline{A}$).
3. Vectors are denoted by in lower case letters (e.g. $\underline{x}$).

Chapter 1

INTRODUCTION

The purpose of this investigation is to develop a smoothing algorithm applicable to noisy two-dimensional data series. The resulting algorithm incorporates the full covariance matrix depicting the accuracy of the data in its smoothing process.

Such data sets are a common occurrence in the geodetic field. They can be found in marine navigation; for example, the time series of position determinations using modern electronic devices to monitor the motion of a ship at sea. A specific example, which is later used to test the smoothing algorithm, is the Lomonosov Ridge Experiment (code-named LOREX) station navigation Doppler data set. The positioning of the LOREX ice stations in the vicinity of the North Pole was achieved using the Navy Navigation Satellite System (NNSS). Due to proximity to the pole, position determinations were greatly influenced by poor geometry and inadequate modelling of the ice motion. It was felt that an algorithm which is able to utilize all the information contained in the position fixes will greatly increase the accuracy of the predicted locations of the observing stations between position determinations.

A recomputation of all position fixes using better station velocity estimates will remove the errors caused by inadequate modelling of the ice motion. In this thesis, error models are used to account for these errors in the LOREX position fixes. A recomputation of all position fixes with better velocity estimates is outside the scope of this research.

1.1 MAIN CONTRIBUTIONS

The following are the main contributions of this thesis:

1. The development of a two-dimensional least-squares cubic spline approximation algorithm.
2. The evaluation of the performance of the spline algorithm using simulated and actual data sets.
3. The application of the spline algorithm to the navigation data sets of the three LOREX ice camps.
4. The modelling of positioning errors, caused by satellite orbital, station velocity and station height errors, through the a priori covariance matrix of the position computations.

1.2 OUTLINE OF THESIS

This thesis is divided into the following chapters.

Chapter 2 contains a brief description of the LOREX-79 expedition, the navigation equipment used and the accuracy of position determinations. This information provides a background for the discussion and assessment of the results from the processing of the navigation data sets using the developed smoothing algorithm.

Chapter 3 presents a brief overview of the scope of the investigation and details several of the options that were considered when designing the smoothing algorithm.

Chapter 4 gives a basic review of the concept of a spline and its similarities with the mechanical spline.

Chapters 5 and 6 are devoted to the formulation of the working equations of the least-squares parametric cubic spline approximation.

Chapter 7 discusses the various possible methods of adjustment; the merits and implication of each in relation to the spline algorithm.

Chapter 8 deals with the aspect of precision estimation of the least-squares estimates, the covariance law and computational accuracy.

Chapter 9 gives an overview into the various tests done to assess the performance of the spline algorithm with regard to the simulated and LOREX test data sets.

Chapter 10 discusses and evaluates the LOREX Doppler data sets and the application of the spline algorithm to the complete navigation data series of the three ice stations on LOREX. Various error models, based on the formal covariance matrix of the position determinations, are used to develop a weighting scheme suitable for high latitude positioning by NNSS. The real-time application of the spline algorithm is also addressed.

Finally, in Chapter 11 a summary of the discussions are given, conclusions drawn and recommendations made.

Chapter 2

DESCRIPTION OF LOREX-79 NAVIGATION DOPPLER DATA SET

In this chapter, a brief description of the LOREX-79 expedition together with the navigation equipment used, and the characteristics of the collected Doppler data are given. The on-site and post-processing of the data series are discussed and inadequacies noted.

2.1 LOREX DATA CHARACTERISTICS

In the spring of 1979, the Earth Physics Branch of the Federal Department of Energy, Mines and Resources in conjunction with the Polar Continental Shelf Project organised an expedition to the geographical North Pole. The Lomonosov Ridge Experiment was designed to explore the nature and origin of the major submarine mountain ridge running across the Arctic Ocean between Greenland and Siberia.

During the expedition, three camps were established on the Arctic ice. Main Camp (S0), Snowsnake (S1) and Iceman (S2) were laid out in an approximate isosceles triangle with sides of about 60 kilometres long and a base of about 100 kilometres.

Canadian Marconi 722B Transit receivers were deployed for positioning the camps. The receiver at the Main Camp was interfaced to a HP 2100 minicomputer, under the RTE II operating system with a HP 7900A Megabyte disc drive, 7-track magnetic tape, plotter and two terminal peripherals. In the off-line mode, it was linked to a CMA 749 magnetic tape cassette unit which allowed the recording of Doppler counts to a resolution of 0.01 of a Doppler count. This, due to the inherent characteristics of the navigation system, is two orders of magnitude better that the on-line tracking mode. Receivers at Snowsnake and Iceman operated unattended, in the automatic data acquisition mode. Gaps, which were later found in the collected data at these two stations, were attributed to failure of the operator to change the data cassettes when full, and signal loss at point of closest approach of the satellite because of the tracking antenna's gain pattern [Popelar et al. 1981].

The monitoring of Arctic ice drift using polar orbiting NNSS satellites has several drawbacks. Firstly, due to the high latitudes, there are many more passes than at southern latitudes and with an omni-directional antenna, severe interference problems arise. The effect of interference results in acquisition of fewer passes with the complete set of Doppler information unless the receiver is programmed to reject some passes in favour of others. Programmed response was not even possible at the Main Camp due to hardware restrictions [Wells and Popelar 1979] and the 20 percent higher tracking efficiency at the Main Camp was attributed to manual intervention [Popelar et al. 1981].

Secondly, with polar orbiting satellites, passes at high latitudes tend to be nearly overhead. Overhead passes produce geometrically poorer cross track position fixes than non-overhead passes. Position determinations using high elevation satellites hence have very elongated error ellipses. The semi-major axis of the error ellipses are in the cross track direction of the passing satellite.

These inherent system difficulties are further complicated by the erratic behaviour of the motion of the ice platform. The ice motion changes both speed and direction, depending on the combined effects of the wind, temperature, and sea currents.

The amassed Doppler data from the three stations were processed on-site and later reprocessed in Ottawa [Popelar et al. 1981].

The satellite fix software package BIONAV [Wells and Grant 1981] was deployed on LOREX to provide position fixes for the ice stations in real time. The positions were computed in the Stereographic Coordinate System and on the International Ellipsoid. An arbitrary rejection criterion, as opposed to a statistical one, of 1000 metres on the formal standard deviations of the position determinations was used. Accepted passes were computed and stored for further processing.

2.2 LOREX DATA POST-PROCESSING

The LOREX data has been post-processed using four different methods, one of which is the subject of this thesis. The first two techniques mentioned below were used on-site at LOREX to smooth the raw fixes.

A one-dimensional least-squares cubic spline approximation (program SMOBS), using the inverse of the trace of the formal covariance matrix derived from the fix computation as weights (equation 4.3), was performed on moving sets of fixes over time periods depending on the character of the station's motion and the number of satellite fixes available. The weighting scheme used is further discussed in Section 4.2. Of a moving set of 70 passes, 50 smoothed central position fixes were accepted. The Lagrange polynomial interpolant was then used to predict the position of the station between smoothed fixes [Wells and Popelar 1979] as SMOBS does not compute the cubic coefficients needed for interpolation. The velocities and associated standard deviations of the ice stations were computed from six preceeding one-hour smoothed coordinate differences. A list of positions and velocities produced by SMOBS can be found in Appendix A of Popelar et al. [1981].

In addition to the on-site cubic spline smoothing, a Kalman filter routine using covariance matrices to model measurement errors and perturbations in the ice motion, was used to sequentially process the Doppler data [Wells and Popelar 1979].

The two on-site smoothing techniques that were used had their disadvantages. The cubic spline approximation technique used weighted both coordinates equally and totally ignored the correlation between errors in the coordinates (which was above 0.9 for 80 percent of the passes). The Kalman filter algorithm could be successfully applied only if the filter could be tuned in accordance to the ice motion noise [Wells and Popelar 1979].

In an effort to improve position determinations in the post-processing of the data, a precise geodetic positioning software package (GEODOP) was extensively modified to accommodate sequential simultaneous positioning of slowly moving stations with constant velocity vectors over a three-hour time period. It uses both higher order modelling of environmental and instrumental effects to reflect the time and space correlation of model parameters during simultaneous satellite tracking from several stations and post-fitted precise satellite orbits. The mean station positions were obtained from a series of satellite position fixes, weighted by their formal covariance matrix, within the three-hour time interval. Station velocities were derived from consecutive mean positions [Popelar et al. 1981].

In this thesis, the acronym "GEODOP" will henceforth be used to denote the modified precise dynamic positioning software package. Of the four post-processing techniques, GEODOP is the only one which does a recomputation of the individual satellite position fixes using better station velocity estimates.

Unlike the spline algorithm used on LOREX, the spline algorithm developed in this thesis (program DSPLIN) is capable of using the full covariance matrix associated with each position fix. This technique is hence able to extract all the information contained in the position determinations. In order to assess the performance of DSPLIN, the LOREX navigation data sets are used and the GEODOP solution adopted as reference standard. The agreement between DSPLIN and GEODOP smoothed values is evaluated and is shown to be better than the agreement between SMOBS and GEODOP.

Chapter 3

CONSIDERED ALTERNATIVE ALGORITHMS

In this chapter, the characteristics of a noisy data set are outlined, the area of investigation given, the alternatives discussed and the reasons for using the spline approach stated. The discussions of the considered possible alternatives for the design of the new smoothing algorithm are a minor digression from the main theme of this thesis. However, they are given here to indicate the scope and area of investigation from which the new smoothing spline algorithm evolved.

The approximating function to be developed must be able to smooth data series with the following characteristics:

1. Unequal data intervals.

   The data points within the given data series are unequally spaced in either time or space.

2. Non-predictive (or non-analytic) data series.

   The external factors governing or generating the data series cannot be effectively modelled or are rapidly changing in time or space.

3. Fully populated data covariance matrices.

   Each data point within the data series has an associated covariance matrix depicting the two-dimensional accuracy of the observed data point.

In addition to the above, the technique must be able to provide a data series of smoothed data points with reliability estimates.

Research into the new smoothing algorithm centered around the concept of extracting existing ideas in the physical world and trying to relate or apply them as smoothing algorithms. The following were examined in this thesis.

3.1 TWO-DIMENSIONAL LEAST-SQUARES CUBIC SPLINE

The mathematical spline which "imitates" the mechanical spline has several properties conducive to its use as a smoothing device. Current development [Spath 1974; Boor 1978], of the cubic spline only allows the cubic spline to smooth data along one axis at a time and totally ignores any correlation between variables.

However, by incorporating the least-squares principle, the covariance law, and by simultaneously solving two parametric cubic splines, the new spline algorithm is able to satisfy all the stated requirements mentioned above. This concept is developed in greater detail, together with a short review of splines, in Chapter 4.

3.2 HAMILTONIAN PRINCIPLE OF LEAST ACTION

The proposed concept here is the utilization of the Lagrangian equations of motion with the Hamiltonian Principle of Least Action [Landau and Lifshitz 1976] to solve for the motion of a particle moving in a non-symmetric cluster of finite particles. Disregarding the mass of the sensing particle, the attracted particle moves according to the sum of all forces generated by the other forcing particles in that cluster [Vanicek 1973]. The interrelationship between the law of gravitational attraction and the conservation of momentum leads to the fact that both velocity and displacement of the particle are smooth functions of time. Under the kinematic approach, the generated force field can be thought of as a non-symmetric potential field and by maintaining a closed system with non-time varying elements of potential, a stationary or conservative potential or force field is said to exist.

By portraying the data series as that of the motion of the particle and the observed data points as gravity-generating mass bodies, a smooth path depicting the data series can hence be generated. By solving the equations of motion of the massless particle, the velocity and position of the particle can be obtained at any time provided that the velocity and position vectors of the particle at the beginning of the trajectory are defined.

In celestial mechanics, this is termed as the solution of a restricted n-body problem; restricted in that all attracting mass bodies are rigidly fixed and held motionless in Eculidean two-space. Further, the use of "relativistic directional masses" as the mass bodies allows the incorporation of error ellipses.

3.3 HEISENBERG PRINCIPLE OF UNCERTAINTY

The concepts of velocity and energy which stem from simple observations of common objects have been applied to certain fundamental experiments in atomic physics.

The position of an electron is known only to a certain degree of accuracy and at any time can be visualized as a wave packet in the proper position with an approximate extension.

The term "wave packet" [Heisenberg 1930] is used to denote a wave-like disturbance whose amplitude differs appreciably from zero only in a bounded region. This region, in general, is in motion and changes in shape and size. The laws of optics and that of the conservation of momentum, can be used to obtain a relationship between the uncertainty in velocity and position; the product is bounded by a certain (Planck's) constant.

This fact bears a striking resemblance to the position of a data point within a data series. The uncertainity of the data point can only be represented by a covariance matrix, much like the wave packet. The probability of the wave packet grows larger from the last observed data point until the next observation.

Using such a representation, the motion of the data point can be represented and reliability estimates obtained for predicted points.

3.4 INTERLACING WIRE AND SLOT REPRESENTATION

A time series of two-dimensional data can be thought of as points in a geometric space defined by three orthogonal vectors. In relation to navigational data, two of the vectors represent the two-dimensional coordinate system of the position while the third is the time axis.

A sequence of time-tagged positions, for example, appears as a series of spatially suspended dots within that space. If the motion is circular, all the dots will lie on the path of a helix.

By interlacing the points with a wire of variable stiffness, points between observations can be predicted. Error ellipses are introduced by replacing the dots or holes with variable size slots. A wire so chosen in its natural state, tries to achieve minimum strain or stress within itself.

For the transformation of the wire to a mathematical tool Frenet formulae and curves are used.

3.5 SELECTION OF TECHNIQUE ADOPTED

An examination of these four proposed smoothing algorithms, revealed that the spline representation seems to be the most promising, easiest to implement, has few variables to contend with (so as not to introduce complex smoothing parameters) and from a practical application point of view satisfies all the requirements of the desired smoothing algorithm.

Chapter 4

BRIEF REVIEW OF SPLINES IN GENERAL

In this chapter, an introduction to the concept of splines is given. The transition from the mechanical spline to its mathematical form is outlined. The topic of smoothing splines contained in the present literature is addressed and differences between those and the smoothing spline algorithm used in this thesis mentioned.

4.1 INTRODUCTION

Polynomials have always played a central role in approximation theory and numerical analysis. Piecewise polynomials, however, had a very limited application due to the existance of discontinuities between polynomial functions. During the past twenty years, piecewise polynomial functions experienced tremendous theoretical advances (e.g. Boor [1978], Shumaker [1981] and Prenter [1975]). Poirer [1973] in his attempt to apply spline functions to econometric data mentions the large gap between theoretical development and practical applications of spline functions.

Currently, spline functions are being applied to smooth noisy geodetic data sets. Their use, however, has been limited to the smoothing of raw data series without obtaining accuracy estimates of either the smoothed data points or the predicted values. A brief exposé on the origin of splines allows better comprehension of the so-called spline functions .

Scheonberg [Shumaker 1981] in 1946 introduced the label spline" when he observed similarities between piecewise polynomial interpolation and a certain mechanical device called the "spline".

A "spline" is a thin beam of some elastic material equipped with a groove and a set of weights (called either ducks or rats), with attached arms designed to fit into the grooves (see Figure 4-1).

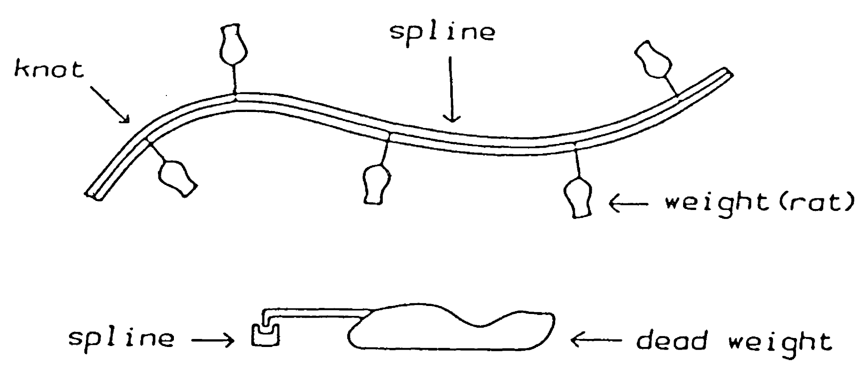


FIGURE 4-1 The Mechanical Spline

These devices are used by architects to draw smooth curves through a set of prescribed points (called in the context of this thesis, data points). To accomplish this, the flexible "spline" is bent in such a manner that it passes through all the points and is held in place by strategically located ducks. Back in the mid-1700's, Euler and the Bernoulli brothers [Shumaker 1981] discovered that the centre-line of the "spline" approximates a given mathematical function (s) which has the following properties:

1. The function (s) is a series of piecewise cubic polynomials between the first and last junction point or knot (knots being analogous to the position of the ducks on the mechanical "spline").
2. The function (s) is linear before the first and after the last knot.
3. The function (s) has continuous first and second derivatives everywhere.
4. The function (s) takes on the value of the data points (i.e. the mechanical spline is an interpolation device).

Given the location of the knots and the data points, these properties define a unique function (s). No assumption, at this stage, is made with regard to the location of the knots amongst the data points. However it should be noted that the knots may or may not be coincident with the position of the data points. This simple but effective analogue of the mathematical spline underlies the fundamental concept of splines in general.

4.2 THE MATHEMATICAL SPLINE

In this section, the properties of the mathematical spline are stated. Reasons for using the piecewise cubic function, along with the current limitations of splines are given. An extension of the use of splines from the current literature is made.

The mathematical spline (s) of degree k is an element of the Spline Space (S) which on a closed interval <a,b> can be written as:

$$s_k \in S_k \subset [p_k \cap c^n \langle a,b \rangle] \qquad - (4.1)$$

where P_k := Polynomial Space containing k th degree polynomials; a subspace of the total Polynomial Space,

$C^n \langle a,b \rangle$:= Space of n th differentially continuous functions on the closed interval <a,b>,

and $n \leq k - 1$.

Any function (s) so defined on a set of points (u_j, v_j), $j = 1, \ldots, m$ will hence have the following properties:

1. It is continuous at $u_1, u_2, \ldots, u_m$ and with $s(u_j) = v_j$.

2. The first k-1 derivatives of the spline functions, (i.e. $s'(u), s''(u), \ldots, s^n(u)$) exist and are continuous within the closed interval $\langle a,b \rangle$ on which the spline function is defined.

It becomes apparent that the mathematical spline defined in Section 4.1 is of the form:

$$s_3 \in S_{3,t} \subset [P_{3,t} \cap C^2 \langle a,b \rangle] \qquad - (4.2)$$

with t_j, $j = 1, \ldots, m$ being the knot sequence.

Cubic splines as in (4.2) are the most popular splines for two reasons. The first is that the human eye has increasing difficulty in detecting jumps in the derivatives of a function as the order of the derivative increases. Discontinuities in the first and second derivatives can be detected, but the eye has great difficulty in detecting jumps in derivatives of order greater than two. Hence a function with only second order derivative continuity will appear smooth to the human eye. The second reason is that there exist distinct disadvantages in using higher degree polynomial functions. The higher the degree, the greater is the evaluation cost of the parameters defining the spline. In addition, individual oscillations of the polynomials generally increase with the degree. Hence the cubic spline combines the advantages of smaller oscillations, modest evaluation cost and second derivative continuity.

The simple one-dimensional interpolating cubic spline, in the context of this research, has several major defects:

1. At least one of the variables has to increase monotonically.
2. It is confined to point estimation between knots or data points.
3. It does not allow for the incorporation of differently weighted data points.
4. No precision estimation of predicted points between data points is possible.

To overcome the first problem, the spline is computed against a third variable (t) which has the property of being always monotonically increasing. For a given set of points (x,y), the parametric spline [Spath 1974] is computed on data sets (x,t) and (y,t); with t monotonically increasing and is related functionally or otherwise to the data points (x,y). A discussion of possible types of variable (t) can be found in Spath [1974]: page 60.

No assumptions have yet been made about the number of knots with respect to a given set of data points. When the number of knots is equal to the number of data points, the spline interpolates between the data points. The transition (which is further discussed in Chapter 5) from an interpolating to an approximating spline is through the incorporation of additional data points in the solution of the parameters defining the spline function. However, additional information is required if a unique approximating spline function is to exist. This additional information takes the form of a weighting scheme amongst the data points; some of which are of an arbitrary nature (e.g. Spath [1974]: page 106) whilst others, being more rigorous (e.g. Boor [1978]: page 235) rely on the minimisation of a specified variational function.

Wells and Popelar [1979] on the LOREX expedition utilized, in near real-time, parametric cubic splines minimising the differences between smoothed and observed positions to smooth the satellite positioning data (x,y). The cubic spline, mentioned earlier in Section 2.2, was computed first in x against time and then in y against time, using the inverse of the trace of the formal covariance matrix obtained from the satellite fix computations, as weights for both splines, i.e.

$$w_i := 1/(\sigma_1^2 + \sigma_2^2)^{\frac{1}{2}} \qquad - (4.3)$$

where i := computed fix number,

w_i := designated weight for the i th computed coordinates,

σ_1^2 and σ_2^2 := diagonal elements of the formal covariance matrix associated with the i th computed position.

The formal covariance matrix associated with each fix is of the form

$$\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

and it portrays the two-dimensional precision of the computed coordinates. The variances of the individual coordinates are denoted by σ_1^2 and σ_2^2 , and the correlation (or covariance) between the two determined coordinates by σ_{12} .

The weighting scheme used on LOREX was rather arbitrary as an operator-specified "coefficient of roughness" was used to scale the assigned weights computed via equation (4.3) to achieve different degrees of smoothness. The high correlation between the two position coordinates was totally ignored. In addition no precision estimates for the individual predicted points between observed positions were available.

The spline algorithm developed in this thesis has been primarily designed to overcome these deficiencies of previous smoothing splines. The full covariance matrices of the data points are utilized in the simultaneous computation of two parametric cubic splines against a third variable, time. Through the method of least squares, it is possible to obtain least-squares estimates and an associated covariance matrix for the parameters defining the spline. The covariance law is also used to derive precision estimates for the predicted data points.

Chapter 5

DEVELOPMENT OF THE PARAMETRIC CUBIC SPLINE

In this chapter, the aspect of base functions for the spline is addressed, the transition from the simple cubic spline to the parametric cubic spline is described and the working equations are given.

Using the concepts embodied in Chapter 4, the cubic spline (s) is hence defined on a set of points (x_i, y_i, t_i), $i = 1, \ldots, m$ for which $s_x(t_i) = x_i$ and $s_y(t_i) = y_i$, and is a composite function with 2(m-1) cubic polynomials. Further, the function (s) is twice differentiable with respect to the variable (t), for $t \in \langle t_o, t_m \rangle$.

5.1 DEFINITION OF BASE FUNCTIONS

The precise definition of the base functions ϕ_ℓ, $\ell = 1, \ldots, 4$ in the piecewise polynomial function

$$s(t) := \sum_{\ell=1}^{4} \alpha_\ell \phi_\ell(t)$$

of the cubic spline has yet to be mentioned. In actuality, there are numerous possible definitions and alternatives. The so-called "generalised cubic splines" [Spath 1974: page

124] utilizes complicated base functions. The two conditions [Spath 1974: page 58] that the base functions of the cubic spline have to satisfy are:

1. The resulting polynomial is a cubic function.
2. The transformation between coefficients and knot vectors must exist (i.e. the equations in Section 5.2 can be formulated).

The use of more complex bases than those given below complicates the formulation and increases computational cost. This is in addition to the fact that smoother splines are obtained from the usual cubic spline than other more complex cubic splines without constraints [Spath 1974]. Hence in this research, a simple set of base functions are used (i.e. 1, Δt, $(\Delta t)^2$ and $(\Delta t)^3$: Δt being the time interval between knots).

5.2 PARAMETRIC CUBIC SPLINE

On a given set of knots (x_j, y_j, t_j), $j = 1, \ldots, m$ the j th piecewise cubic polynomial is thus defined:

$$s_{xj}(t) := a_{j1} + \sum_{\ell=2}^{4} a_{j\ell}(t-t_j)^{\ell-1}$$

$$s_{yj}(t) := b_{j1} + \sum_{\ell=2}^{4} b_{j\ell}(t-t_j)^{\ell-1} \qquad - (5.1)$$

with $s_{xj}(t_j) = x_j$

and $s_{yj}(t_j) = y_j$

and $t_j \leq t \leq t_{j+1}$

Note that $a_{j\ell}$ and $b_{j\ell}$ are the elements of the coefficient vectors $(\underline{a}_j, \underline{b}_j)$ of the two cubic polynomials of the spline.

The functional relationship between the given knots and the cubic coefficients, i.e. the boundary values, can be expressed as follows:

$$x_j := s_{xj}(t_j) = a_{j1} \qquad \text{- (5.2a)}$$

$$y_j := s_{yj}(t_j) = b_{j1} \qquad \text{- (5.2b)}$$

$$x_{j+1} := s_{xj}(t_j+1) = a_{j1} + \sum_{\ell=2}^{4} a_{j\ell}(t_{j+1}-t_j)^{\ell-1} \qquad \text{- (5.2c)}$$

$$y_{j+1} := s_{yj}(t_{j+1}) = b_{j1} + \sum_{\ell=2}^{4} b_{j\ell}(t_{j+1}-t_j)^{\ell-1} \qquad \text{- (5.2d)}$$

$$x'_j := s'_{xj}(t_j) = a_{j2} \qquad \text{- (5.2e)}$$

$$y'_j := s'_{yj}(t_j) = b_{j2} \qquad \text{- (5.2f)}$$

$$x'_{j+1} := s'_{xj}(t_{j+1}) = a_{j2} + \sum_{\ell=3}^{4} (\ell-1)a_{j\ell}(t_{j+1}-t_j)^{\ell-2} \qquad \text{- (5.2g)}$$

$$y'_{j+1} := s'_{yj}(t_{j+1}) = b_{j2} + \sum_{\ell=3}^{4} (\ell-1)b_{j\ell}(t_{j+1}-t_j)^{\ell-2} \qquad \text{- (5.2h)}$$

$$x''_j := s''_{xj} = 2a_{j3} \qquad \text{- (5.2i)}$$

$$y''_j := s''_{yj} = 2b_{j3} \qquad \text{- (5.2j)}$$

$$x''_{j+1} := s''_{xj} = 2a_{j3} + 6a_{j4}(t_{j+1}-t_j) \qquad \text{- (5.2k)}$$

$$y''_{j+1} := s''_{yj} = 2b_{j3} + 6b_{j4}(t_{j+1}-t_j) \qquad \text{- (5.2l)}$$

There is a choice of expressions with regard to the explicit form of the coefficient vectors a and b. They can be expressed either in terms of their functional values (i.e. equations 5.2a to 5.2d) and their first derivatives (i.e. equations 5.2e to 5.2h) or their functional values (i.e. equations 5.2a to 5.2d) and their second derivatives (i.e. equations 5.2e to 5.2l) [Spath 1974; Boor 1978].

Parameters of functional values and first derivatives are used here to represent the knots on the assumption that the positions and velocities, taking navigational data as an example, are easier to visualize than positions and accelerations.

Inverting equations 5.2a to 5.2h,

$$a_{j1} = x_j \qquad \text{- (5.3a)}$$

$$a_{j2} = x'_j \qquad \text{- (5.3b)}$$

$$a_{j3} = 3(x_{j+1}-x_j)/\Delta t_j^2 - (2x'_j + x'_{j+1})/\Delta t_j \qquad \text{- (5.3c)}$$

$$a_{j4} = [-2(x_{j+1}-x_j)/\Delta t_j + x'_j + x''_{j+1}]/\Delta t_j^2 \qquad \text{- (5.3d)}$$

$$b_{j1} = y_j \qquad \text{- (5.3e)}$$

$$b_{j2} = y'_j \qquad \text{- (5.3f)}$$

$$b_{j3} = 3(y_{j+1}-y_j)/\Delta t_j^2 - (2y'_j + y'_{j+1})/\Delta t_j \qquad \text{- (5.3g)}$$

$$b_{j4} = [-2(y_{j+1}-y_j)/\Delta t_j + y'_j + y'_{j+1}]/\Delta t_j^2 \qquad \text{- (5.3h)}$$

with $\Delta t_j := t_{j+1}-t_j$

Hence there exist a total of eight elements, four in each of the two coefficient vectors, $\underline{a}_j$ and $\underline{b}_j$, for each interval $\langle t_j, t_{j+1} \rangle$ defining the cubics.

To ensure that adjacent piecewise cubics are continuous, constraint relationships are enforced. Considering the junction point between the j th and (j+1) th interval, second derivative continuity is achieved by the following equations:

From equations 5.2i to 5.21,

$$6a_{j4}\,\Delta t_j + 2a_{j3} = 2a_{j+1,3}$$

and $$6b_{j4}\,\Delta t_j + 2b_{j3} = 2b_{j+1,3}$$

- (5.4)

Substituting the respective coefficients for their functional values and derivatives, the following relationships are obtained [Spath 1974: page 46]:

$$(1/\Delta t_j)x'_j + 2(1/\Delta t_{j+1} + 1/\Delta t_j)x'_{j+1} + (1/\Delta t_{j+1})x'_{j+2}$$

$$= 3(x_{j+1}-x_j)/\Delta t_j^2 + 3(x_{j+2}-x_{j+1})/\Delta t_{j+1}^2$$

and similarly,

$$(1/\Delta t_j)y'_j + 2(1/\Delta t_{j+1} + 1/\Delta t_j)y'_{j+1} + (1/\Delta t_{j+1})y'_{j+2}$$

$$= 3(y_{j+1}-y_j)/\Delta t_j^2 + 3(y_{j+2}-y_{j+1})/\Delta t_{j+1}^2$$

- (5.5)

The working equations developed up to the present chapter are the usual parametric cubic spline equations and they can be used as an interpolating function for any given set of data points.

For m points, there are 2(m-1) equations with 2 m first derivative unknowns. Further information are required to make the set of equations non-singular. These take the form of "boundary" (or "end") conditions. They specify the form which the outer knots will take with respect to themselves or other inner knots.

The choice and type of end conditions affects the performance of the spline. The effect, however, diminishes towards the centre of the data span. For a relatively large data series, a large proportion of the spline remains invariant to the type of boundary conditions chosen.

The transition from the parametric interpolating cubic spline to an approximating spline is discussed in the following chapter.

Chapter 6

LEAST-SQUARES CUBIC SPLINE APPROXIMATION

In this chapter, the cubic spline is transformed to an approximation tool by the incorporation of additional data points between knots. The equations representing an additional point are developed.

Consider an "extra" data point (x^o, y^o) at time t^o within the j th piecewise cubic interval. Since the point lies on the cubic,

$$x^o := s_{xj}(t^o) = a_{j1} + \sum_{\ell=2}^{4} a_{j\ell}(t^o - t_j)^{\ell-1}$$

and $$y^o := s_{yj}(t^o) = b_{j1} + \sum_{\ell=2}^{4} b_{j\ell}(t^o - t_j)^{\ell-1}$$

with $t_j \le t^o \le t_{j+1}$ - (6.1)

By substituting the explicit values of the coefficients in terms of the position of the knots (equations 5.3) and first derivatives, the equations (6.1) are transformed to

$$[1 + \overline{\Delta t}^2(2\overline{\Delta t}-3)]x_j + \Delta t^o(1-\overline{\Delta t})^2 x'_j$$

$$- \overline{\Delta t}^2(2\overline{\Delta t}-3)x_{j+1} - \Delta t^o\overline{\Delta t}(1-\overline{\Delta t})x'_{j+1}$$

$$- x^o = 0$$

and $$[1 + \overline{\Delta t}^2(2\overline{\Delta t}-3)]y_j + \Delta t^o(1-\overline{\Delta t})^2 y'_j$$

$$- \overline{\Delta t}^2(2\overline{\Delta t}-3)y_{j+1} - \Delta t^o\overline{\Delta t}(1-\overline{\Delta t})y'_{j+1}$$

$$- y^o = 0 \qquad - (6.2)$$

with $\overline{\Delta t} := \Delta t^o/\Delta t_j$

$\Delta t^o := t^o - t_j$

and $\Delta t_j := t_{j+1} - t_j$

Using equations (6.2) (which are termed as observation equations) and those enforcing second derivative continuity (equations 5.4 - termed as constraint equations), the parameters of the least-squares spline can now be determined.

In the transformation from an interpolating to an approximating spline, the value of the knots themselves are treated as unknowns and instead of 2 m first derivative unknowns, the solution vector is expanded to include the functional value of the knots, making a total of 4 m unknowns; m being the number of knots.

There exist several methods of solving this type of least-squares problem with added constraints. The merits of each of the three techniques will be discussed in the following chapter.

Chapter 7

LEAST-SQUARES ADJUSTMENT

In this chapter, three methods of performing a least-squares adjustment with added constraints are evaluated. These are the unified approach, least-squares adjustment through the elimination of constraints, and the method of least squares with constraints. The explicit equations of the latter technique are given.

A least-squares adjustment is performed to derive the best estimates for the unknown parameters. The development of the least-squares method of solution is via the Lagrange method with the covariance law being applied to obtain precision estimates of the computed parameters [Wells and Krakiwsky 1971].

Let the unknown parameters be represented by the vector $\underline{x}$, and the observed data points by the vector $\underline{l}$. The two mathematical models that have to be simultaneously satisfied are:

1. The observation (or primary) model (equations 6.1),

$$\underline{f}_1(\underline{x},\underline{l}) = \underline{0}.$$

2. The constraint (or secondary) model (equations 5.4),

$$\underline{f}_2(\underline{x}) = \underline{0} .$$

There exist several approaches to solving combined models of this nature; the unified approach, the solution via the elimination of constraints, and the functionally constrained least-squares technique.

7.1 UNIFIED APPROACH

With the unified approach [Mikhail 1976], the unknown parameters are also regarded as "observations" or pseudo-observations and the solution here is by the summation of normal equations from the two models above. The constraint pseudo-observables are differently weighted against the observations. By varying the weights for the constraint pseudo-observables, various degrees of satisfaction of the constraint equations are achieved. This seemingly easy treatment has several severe pitfalls [Lawson and Hanson 1974: page 149].

To ensure that the constraint equations are adequately satisfied, heavier weights (i.e. heavier in relation to those of the observations) are placed on the constraint pseudo-observables. However, the constraint equations are, by themselves, singular and the implementation of large weights causes the combined set of normal equations to be ill-conditioned.

Ill-conditioned solutions are undesirable as a loss in precision occurs when computing the inverse and later the solution vector. One of the prerequisites of the smoothing algorithm is the ability to provide reliability estimates for the estimated parameters (Chapter 3). The assignment of arbitrarily large weights on the constraint pseudo-observables affects the a posteriori covariance matrix for the parameters. The "correct" ratios between the input covariances linked together by a common a priori variance factor presents a difficulty to any one using this method as there exists no physical or geometrical criteria for selecting these ratios.

This technique was used here to verify the feasibility of the new spline algorithm. However it was later discarded as it could not meet the requirements of providing precision estimates for the parameters and hence predicted points (as specified in Chapter 3).

7.2 SOLUTION VIA THE ELIMINATION OF CONSTRAINTS

The constraint equations can be used to solve for, functionally, as many parameters as there are constraints [Mikhail 1976: page 217]. These parameters are then eliminated from the observation model. Hence the total number of unknown parameters is reduced. The remaining parameters are solved for directly by least-squares adjustment. The eliminated parameters are later computed by back substitution.

As this is a rigorous technique, formal covariance matrices are derived for all estimated parameters. A development of this method of solution is included in Appendix I.

The two primary advantages of this method are:

1. a reduced set of unknown parameters are present in the least-squares solution, and
2. the constraint equations are removed from the least-squares solution.

However to successfully apply this method of least-squares estimation, the constraint equations have to be made non-singular; as shown in Appendix I. As mentioned in Chapter 5, the constraint equations can be made non-singular by specifying the end conditions of the spline. The question of developing an algorithm to select the independent parameters, a difficulty expressed by Mikhail [1976]: page 217, does not exist as it will always be the first derivatives that are eliminated.

The predefinition of the end conditions poses as a minor drawback in the use of this technique. The merits of such a step is questionable in light of the fact that, for large data series, it will be required to "join pieces" of splines together. This is due to the length of the data span used for each spline computation being limited by the user's computer memory capacity. The capability of passing on the junction (or knot) vector values is hence desirable if the data series is to be continuous at the joints. However, with this method of least squares, the end conditions have to be predefined within the algorithm and remain unaltered. To generate a continuous smoothed data series, an overlap of data points between adjacent splines within a data series is required. This uses the property that the effect of the end knot condition diminishes towards the centre of the spline (Section 5.2).

7.3 METHOD OF LEAST SQUARES WITH CONSTRAINTS

Finally, there is a method of least squares with added constraints between unknown parameters [Wells and Krakiwsky 1971: page 142].

The constraints are imbedded into the variational function and strictly (or absolutely) enforced. Primary drawbacks are of a computational nature. For m knots there are two matrices of sizes 4 m and 2(m-2) to be inverted. In addition, the observation model rather than the constraint model must be non-singular (the reverse of that required by the technique described in Section 7.2). In practical terms, this means that there will be a lower limit on the number of data points that must be contained in each cubic interval.

As the spline algorithm is used here as an approximation tool, adequate redundancy to meet this lower limit will be the usual case. This condition can hence be easily adhered to or algorithmically checked and enforced (e.g. by subroutine CHECK in Appendix VII).

7.3.1 Derivation of the Functionally Constrained Model

In this section, a brief development of the method of least squares used in the algorithm is described and the working equations are given.

There are, as mentioned earlier in this chapter, two mathematical models present in the adjustment:

the observation model,

$$\underline{f}_1(\underline{x},\underline{l}) = \underline{0}$$

and the constraint model,

$$\underline{f}_2(\underline{x}) = \underline{0}$$

$$- (7.1)$$

with $\underline{x}$ vector representing the value and first derivative unknowns of the knots,

$\underline{l}$ vector containing the observed data points,

and $\underline{x} = \underline{x}^o + \underline{\delta}$

$\underline{l} = \underline{l}^o + \underline{v}$.

The observation vector, $\underline{l}^o$, has a full weight matrix, $\underline{P} := \sigma_o^2 \underline{\Sigma}_{\underline{l}^o}$ ($\underline{\Sigma}_{\underline{l}^o}$ being the a priori covariance matrix of the observations).

Using Taylor's expansion, equations (7.1) are expanded about the initial approximation, $\underline{x}^o$ and the observed values, $\underline{l}^o$, i.e.

$$\underline{f}_1(\underline{x},\underline{l}) \doteq \underline{f}_1(\underline{x}^o,\underline{l}^o) + \left.\frac{\partial \underline{f}_1}{\partial \underline{x}}\right|_{\underline{x}^o,\underline{l}^o} \underline{\delta} + \left.\frac{\partial \underline{f}_1}{\partial \underline{l}}\right|_{\underline{x}^o,\underline{l}^o} \underline{v} = \underline{0}$$

and

$$\underline{f}_2(\underline{x}) \doteq \underline{f}_2(\underline{x}^o) + \left.\frac{\partial \underline{f}_2}{\partial \underline{x}}\right|_{\underline{x}^o} \underline{\delta} = \underline{0}$$

- (7.2)

where $\underline{f}_1(\underline{x}^o,\underline{l}^o)$ and $\underline{f}_2(\underline{x}^o)$ are the misclosure vectors. Rewriting,

$$\underline{w}_1 + \underline{A}_1\underline{\delta} + \underline{Bv} = \underline{0}$$

$$\underline{w}_2 + \underline{A}_2\underline{\delta} = \underline{0}$$

where $\underline{w}_1 := \underline{f}_1(\underline{x}^o,\underline{l}^o)$

$\underline{w}_2 := \underline{f}_2(\underline{x}^o)$ - (7.3)

and the design matrices,

$$\underline{A}_1 := \left.\frac{\partial \underline{f}_1}{\partial \underline{x}}\right|_{\underline{x}^o,\underline{l}^o}, \quad \underline{A}_2 := \left.\frac{\partial \underline{f}_2}{\partial \underline{x}}\right|_{\underline{x}^o} \quad \text{and } \underline{B} := \left.\frac{\partial \underline{f}_1}{\partial \underline{l}}\right|_{\underline{x}^o,\underline{l}^o}$$

In hyper-matrix notation,

$$\begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} + \begin{bmatrix} \underline{A}_1 \\ \underline{A}_2 \end{bmatrix} \underline{\delta} + \begin{bmatrix} \underline{B} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \underline{v} = \underline{0}$$

- (7.4)

(Equation (7.4) is equivalent to the partitioning of the design matrix $\underline{A}$, and the misclosure vector $\underline{w}$, in the direct least-squares model of $\underline{A\delta} + \underline{Bv} + \underline{w} = \underline{0}$).

The least-squares solution is then derived by minimising the variational function $\emptyset$, where

$$\emptyset := \underline{\hat{v}}^T \underline{P} \underline{\hat{v}} + \partial \underline{\hat{k}}_1^T (\underline{w}_1 + \underline{A}_1 \underline{\delta} + \underline{Bv}) + \partial \underline{\hat{k}}_2^T (\underline{w}_2 + \underline{A}_2 \underline{\delta}) \qquad - (7.5)$$

with $\underline{\hat{k}}_1$ and $\underline{\hat{k}}_2$ being the estimates for the Lagrange multipliers. (The '^' symbol denotes least-squares estimates of their true values).

For minima, $\dfrac{\partial \emptyset}{\partial \underline{\hat{v}}} = \dfrac{\partial \emptyset}{\partial \underline{\hat{\delta}}} = 0.$

or $\quad \underline{P}\underline{\hat{v}} + \underline{B}^T \underline{\hat{k}}_1 = \underline{0}$

and $\quad \underline{A}_1^T \underline{\hat{k}}_1 + \underline{A}_2^T \underline{\hat{k}}_1 = \underline{0}$

$$- (7.6)$$

Combining equations (7.4) and (7.6), the set of normal equations is thus formulated.

$$\begin{bmatrix} \underline{P} & \underline{B}^T & \underline{0} & \underline{0} \\ \underline{B} & \underline{0} & \underline{A}_1 & \underline{0} \\ \underline{0} & \underline{A}_1^T & \underline{0} & \underline{A}_2^T \\ \underline{0} & \underline{0} & \underline{A}_2 & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\hat{v}} \\ \underline{\hat{k}}_1 \\ \underline{\hat{\delta}} \\ \underline{\hat{k}}_2 \end{bmatrix} + \begin{bmatrix} \underline{0} \\ \underline{w}_1 \\ \underline{0} \\ \underline{w}_2 \end{bmatrix} = \underline{0} \qquad - (7.7)$$

By the process of elimination of variables, followed by back substitution of functionally solved parameters, the following explicit expressions are derived.

$$\hat{\underline{\delta}} = -(\underline{A}_1^T\underline{M}^{-1}\underline{A}_1)^{-1}(\underline{A}_2^T\hat{k}_2 + \underline{A}_1^T\underline{M}^{-1}\underline{w}_1)$$

$$\hat{\underline{k}}_2 = (A_2(\underline{A}_1^T\underline{M}^{-1}\underline{A}_1)^{-1}A_2^T)^{-1}[\underline{w}_2 - \underline{A}_2(\underline{A}_1^T\underline{M}_1^{-1}\underline{A}_1)^{-1}\underline{A}_1^T\underline{M}^{-1}\underline{w}_1]$$

$$\hat{\underline{k}}_1 = \underline{M}^{-1}(\underline{A}_1\hat{\underline{\delta}}+\underline{w}_1)$$

$$\hat{\underline{v}} = -\underline{P}^{-1}\underline{B}^T\hat{\underline{k}}_1$$

$$\hat{\sigma}_o = \hat{\underline{v}}^T\underline{P}\hat{\underline{v}}/\gamma$$

where $\underline{M} = \underline{BP}^{-1}\underline{B}^T$

and γ = degrees of freedom. - (7.8)

The compatibility between the spline model and the observed data points can be evaluated through the Chi-squared (χ^2) statistical test on the a posteriori variance factor ($\hat{\sigma}_o^2$). The a priori variance factor (σ_o^2) of the spline model is set to unity (i.e. the weight matrix $\underline{P} := \underline{\Sigma}_\ell^{-1}$). The validity of the hypothesis that both variance factors are compatible with each other can be tested using the knowledge that the statistic

$$y := \gamma\, \hat{\sigma}_o^2/\sigma_o^2 \qquad \text{(with } \gamma \text{ degrees of freedom)}$$

has a $\chi^2(\xi;\gamma)$ probablity density distribution [Vanicek and Krakiwsky 1982]. Incompatibility exists between the a posteriori variance factor ($\hat{\sigma}_o^2$) and a priori variance factor (σ_o^2) (i.e. the Chi-squared test fails) when the assumed model for the observations is incorrect or when the a priori covariance matrix of the observations is in error [Vanicek and Krakiwsky 1982: page 237].

7.3.2 Simplification of Functionally Constrained Model

Two basic facts regarding the spline model (equation 7.1) that can be taken advantage of are:

1. the model is linear, and
2. the model is parametric (i.e. $\underline{B} = -\underline{I}$, with $\underline{I}$ being the identity matrix).

As the model is linear, no iterations are required. The incorporation of an a priori solution vector (subroutine APRORI in Appendix VII) is seen as just a method of increasing the numerical accuracy of the solution vector.

By being a parametric model, the equations (7.8) given for the general case are thus reduced to

$$\hat{\underline{\delta}} = -(\underline{A}_1^T \underline{P} \underline{A}_1)^{-1} (\underline{A}_2^T \hat{\underline{k}}_2 + \underline{A}_1^T \underline{P} \underline{w}_1)$$

$$\hat{\underline{k}}_2 = [\underline{A}_2 (\underline{A}_1^T \underline{P} \underline{A}_1)^{-1} \underline{A}_2^T]^{-1} [\underline{w}_2 - \underline{A}_2 (\underline{A}_1^T \underline{P} \underline{A}_1)^{-1} \underline{A}_1^T \underline{P} \underline{w}_1]$$

$$\hat{\underline{v}} = -(\underline{A}_1 \hat{\underline{\delta}} + \underline{w}_1)$$

$$\hat{\sigma} = \hat{\underline{v}}^T \underline{P} \hat{\underline{v}} / \gamma$$

$$- (7.9)$$

Chapter 8

PRECISION ESTIMATION

In this chapter, the covariance law is described, an expression for the covariance matrix for the least-squares estimates is given and the computational accuracy of the algorithm is addressed.

Precision estimation of the least-squares estimates and all other subsequently derived quantities are obtained by using the covariance law.

8.1 COVARIANCE LAW

The covariance law, which is also called the law of covariances and propagation of covariances [Wells and Krakiwsky 1971: page 20], is stated as follows:

Given a variate $\underline{y}$ linearly related to another variate $\underline{x}$ by the equation

$$\underline{y} = \underline{Gx} \;, \quad \underline{G} \text{ being the transformation matrix from } \underline{x} \text{ to } \underline{y}$$

and with

$$\begin{aligned} E[\underline{y}] &= E[\underline{Gx}] \\ &= \underline{G}E[\underline{x}], \quad E \text{ being the expectation operator.} \end{aligned} \qquad - (8.1)$$

The covariance matrix of $\underline{y}$ (i.e. $\underline{\Sigma}_{\underline{y}}$), given the covariance matrix of $\underline{x}$ (i.e. $\underline{\Sigma}_{\underline{x}}$), is hence derived as follows:

$$
\begin{aligned}
\underline{\Sigma}_{\underline{y}} &= E[(\underline{y} - E[\underline{y}])(\underline{y} - E[\underline{y}]) \;] \\
&= E[(\underline{G}\underline{x} - \underline{G}E[\underline{x}])(\underline{G}\underline{x} - \underline{G}E[\underline{x}]) \;] \\
&= E[\underline{G}(\underline{x} - E[\underline{x}])(\underline{x} - E[\underline{x}]) \; \underline{G} \;] \\
&= \underline{G}E[(\underline{x} - E[\underline{x}])(\underline{x} - E[\underline{x}]) \;]\underline{G} \\
&= \underline{G} \, \underline{\Sigma}_{\underline{x}} \, \underline{G}
\end{aligned}
\qquad - (8.2)
$$

The propagation of variances and covariances with linear functions is independent of the density functions and is valid for any probability distribution [Mikhail 1976: page 78].

Incidentally, the $\underline{G}$ matrix essentially represents the Jacobian of $\underline{y}$ with respect to $\underline{x}$, i.e.

$$\underline{\Sigma}_{\underline{y}} = \underline{J}_{\underline{yx}} \, \underline{\Sigma}_{\underline{x}} \, \underline{J}^{T}_{\underline{yx}} \qquad - (8.3)$$

with $\underline{J}_{\underline{yx}} := \dfrac{\partial \underline{y}}{\partial \underline{x}}$

8.2 PRECISION OF LEAST-SQUARES ESTIMATES

Using the above covariance law (equation 8.3) and equations (7.9), the a posteriori covariance matrix of the unknown solution vector($\underline{\hat{\delta}}$) is of the form:

$$\underline{\Sigma}_{\hat{\underline{\delta}}} = \underline{N}^{-1}(\underline{I} - \underline{A}_2^T\underline{M}^{-1}\underline{A}_2\underline{N}^{-1}) \qquad - (8.4)$$

where $\underline{N} = \underline{A}_1^T\underline{P}\underline{A}_1$

and $\underline{M} = \underline{A}_2\underline{N}^{-1}\underline{A}_2^T$

8.3 PRECISION OF COMPUTATIONS

The spline algorithm approaches instability under either of the following conditions:

1. the number of data points in the knot intervals is insufficient to define the spline (Section 7.3), or
2. the observed data points within an interval have large variances (or very low weights).

Computational accuracy is degraded when either of the above occurs. Both conditions cause the normal equation matrix $\underline{N}$, of the observation model $\underline{f}_1$ to be ill-conditioned. If the violations are severe enough, the matrix becomes singular. Between singularity and a well-conditioned matrix, there lies many shades of ill-conditioning, some of which are not detected in the calculations, but give apparently good results through round-off.

In an effort to detect such "in-betweens" the program DSPLIN utilizes two different techniques. Firstly, condition numbers (via subroutine COND in Appendix VII) are computed for all inverses computed in DSPLIN and secondly, a check on the internal computational accuracy is performed (in subroutine LS in Appendix VII). Equations (II-4 and II-2) used by routines COND and LS in Appendix VII are described in Appendix II.

As a preventive measure, a count (by subroutine CHECK) of the number of data points in each knot interval is made prior to computations and superfluous knots dropped if necessary.

The aspect of numerical computational accuracy and the detection of precision loss in computing the least-squares estimates is further addressed in Appendix II.

Chapter 9

DISCUSSION AND EVALUATION OF THE SPLINE ALGORITHM

In this chapter, the various areas of investigation into the performance of the spline algorithm are discussed. Selected results of the processing of the simulated and LOREX data sets are presented.

The following are the areas investigated:

a) Distribution of data points within a data series.

b) Selection of knots.

c) Completeness of spline model.

d) Boundary (outer) knots.

e) Joining two separate adjacent splines.

9.1 DISTRIBUTION OF DATA POINTS

The simulated data set (Appendix IV), is created, amongst other reasons, to evaluate the spline algorithm under the following distributions of data points within a data series:

1. Equal distribution of points along a curve.
2. Dense distribution of points along a straight line.

3. Irregular distribution of points along a straight line.
4. Sparse distribution of points along straight line.
5. No data points along a curve.

The simulated data points were generated at equal time intervals by equation IV-1. Of the 50 simulated data points, 21 were chosen as the simulated data set. The selection of the points is such that all the above distributions appeared in subsets of the simulated data series.

From the multitude of tests that were carried out (Table 9-1 forms but a small sample), no detrimental effects on the spline were observed with any of the above mentioned data point distributions.

9.2 SELECTION OF KNOTS

In this section, the topic of knot selection is addressed. The design constraints of the curve fitting algorithm are given. Performance of the various possible knot selection schemes are discussed and conclusions drawn.

The computation of the spline, given the location of the knots (e.g. through the specification of the third variable time, in navigation data sets), has been discussed in the Chapters 5 and 6. The number and location of the knots

TABLE 9-1a Results from the DSPLIN Program Using Simulated Data

TEST NO.	KNOT SELECTION SCHEME USED	RUN NO.	NO. OF KNOTS	R.M.S. RESIDUALS	VARIANCE FACTOR	CHI. SQ. TEST*	ADDITIONAL INFORMATION (SEE BELOW): Knot Times[1]	ADDITIONAL INFORMATION (SEE BELOW): Curve Fitting Options[2]	REMARKS
1	Equal interval	1	3	30.07	24.24	FAIL			*Two of the knots have been dropped by CHECK routine. i.e. only 8 knots were used.
		2	5	7.94	1.69	PASS			
		3	10*	5.57	1.00	PASS			
2	Visual inspection	1	3	20.42	10.57	FAIL	0,38,98		
		2	5	6.03	0.96	PASS*	0,8,43,78,98		
								(Parameter No.) (1 2 3 4 5 6 7)	
3	Curve fitting routine (with residual error bar tests at α % level as the rejection criteria	1	7	5.89	1.00	PASS	0,17,27,39,56,45,98	6 100 10 2 1 0 0	α = 50%
		2	7	5.66	0.96	PASS	0,17,27,39,63,80.5,98	6 100 10 2 0 0 0	α = 50%
		3	6	5.20	0.76	PASS	0,22,34,56,71,98	6 100 10 2 1 0 0	α = 90%
								(Parameter No.) (1 2 3 4 5 6 7)	
4	Curve fitting routine (with the apos. variance factor at α % confidence level as the rejection criteria).	1	4	7.04	1.29	PASS	0,47,72.5,98	6 1 10 10 2 0 1	α = 99%
		2	5	7.21	1.39	PASS	0,25,47,80,98	6 1 10 10 2 1 0	α = 50%
		3	6	5.46	0.84	PASS	0,22,29,47,84.5,98	6 1 10 10 2 1 1	α = 38%

* at the 99% confidence level

[1] Knot Times - These are the times of the knots chosen to represent the simulated data series. A complete listing of the times and location of the input positions of the data points is given in Appendix IV.

[2] Curve Fitting Options - These are the parameters used with the curve fitting algorithm (Subroutine FIT in Appendix VII). A condensed description of the set of parameters (or options) is given in the following Table 9-1b.

TABLE 9-1b

DESCRIPTION OF THE PARAMETERS (OR OPTIONS) OF THE CURVE FITTING ALGORITHM (FIT)

Parameter Number	Description
1	: Minimum number of data points contained in the first cubic polynomial.
2	: Location of the designated knot times: 1 : midway between two data point times. 0 : coincident with the data point time.
3	: Number of points, in terms of a percentage of the current interval, to be placed in the extended interval (see Figure 9-1).
4	: Weighting factor(X) for the former slope vector based on the estimated covariance matrix, i.e. $(\text{weight matrix})^{-1}$:=formal covariance matrix * (10/X) < 1 : unweighted, i.e. X=0. 10 : formal covariance matrix, i.e. X=1. 10> : formal covariance matrix to be divided by (10/X).
5	: Minimum number of data points per interval.
6	: Inclusion of data points in the extended interval in the rejection tests.
7	: Minimum number of data points in the extended interval.

The last three parameters deal with the level of output desired from the subroutine FIT.

8	: All information requested, e.g. current knot vector, residuals, a posteriori variance factor and status of curve fitting.
9	: Print computed knot vectors only.
10	: Print formal or estimated covariance matrix of computed knot vector.

Note: Generally, '1' denotes 'yes' and '0' denotes 'no' influence the resulting splined (smoothed) values. At one extreme, a lower number of knots decreases the computational load and heavily smooths the data set. At the other extreme, the spline approaches the interpolation of all the data points; including the noise in them.

This difficulty is present in all smoothing algorithms. The desired degree of smoothness of the resulting curve must be determined by external evidence; this is more of an art than a science.

Four approaches to the problem of knot selection are incorporated (through subroutine SELKNT in Appendix VII) in the spline algorithm.

They are:

1. equal time intervals (i.e. equal time span per cubic interval),
2. equal point intervals (i.e. equal number of points per cubic interval),
3. knots chosen by visual inspection, and
4. knots chosen via a "trend evaluating" algorithm.

The first and second knot selection schemes are easy to implement and the spline model can be considered as "single number criterion" type of smoothing algorithm (i.e. different degrees of smoothing are achieved by varying only one parameter).

With unequally spaced or irregular data sets, a reduction in the number of knots is achieved by strategically locating each knot within the data span. This can be done either by visual inspection or through some predefined trend evaluating (or curve fitting) algorithm.

Basically, what is needed is a process to determine the length of time within a data span in which the data series can be aptly represented by a cubic curve.

With reference to the LOREX navigation data sets, the choice of the minimum time interval between knots can be obtained by evaluating the length of the response time of the ice platform to the maximum possible force that can act on the ice sheet. Unfortunately, due to the complexity of the ice sheet, wind stress effects or that of undercurrents, a single figure cannot be easily obtained. In addition, if the ice is subjected to a force less than the maximum, it may be more appropriate to smooth over a longer time span.

A plausible alternative is the implementation of an algorithm designed specifically to automatically select "optimum" knot locations based on some predefined criteria. For the design of the trend evaluating routine, numerous papers (e.g. Ellis and McLain [1977], Ichida and Kiyono [1977] and Chung [1980]) on curve fitting have been drawn upon to create the "weighted one pass (left to right) local cubic polynomial" curve fitting algorithm (subroutine FIT in Appendix VII).

Several considerations must be taken into account in designing such an algorithm. Firstly, the spline function cannot be computed solely using local data, i.e. it is of a global nature [Ellis and McLain 1977]. An alternative has to be found which still uses the attractive properties of both piecewise cubics and the least-squares norm as the basis for the curve fitting approximation.

Ichida and Kiyono [1977] through eigenvalue analysis (based on simple assumptions) and Chung [1980] using more complex error techniques, demonstrated that the C^2 functions, (i.e. space containing all polynomials with second derivative continuity), cannot be used as they exhibit inherent instabilities. The authors, Ichida and Kiyono, and Chung, mentioned above, used C^1 functions and the extended interval concept to resolve the numerical instabilities in the curve fitting procedure.

The "extended interval" concept (Figure 9-1) entails extending the current interval, Δt , of the j th piecewise function by $\delta \Delta t$ to incorporate additional observed data points in the computation of the right end knot vector (i.e. the (j+1) th knot vector in Figure 9-1). The position and slope of the right end knot is hence adjusted to allow for the next curve segment [Chung 1980]. Very briefly, the non-inclusion of future data through the extended interval concept results in instabilities because of the amplification of propagation and truncation errors [Chung 1980].

A curve fitting algorithm with variable controlling parameters (or options - Table 9-1b) has been developed

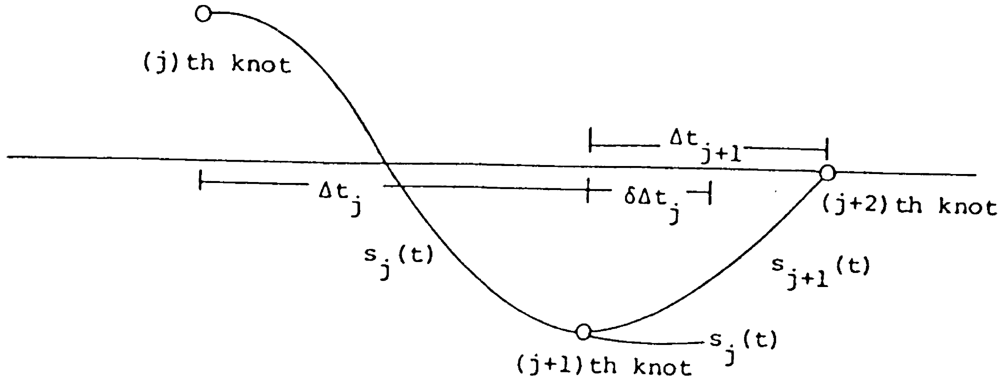


FIGURE 9-1 Extended Interval Concept in Curve Fitting

(subroutine FIT in Appendix VII). It utilizes both weighted knot and slope information from the previous interval to simultaneously compute two least-squares cubic polynomials, x against t and y against t (see Section 5.2, equation 5.3), in the current and extended intervals, and iteratively extends both intervals until the chosen error tolerances are exceeded.

Hence for the Δt_j interval, the two least-squares cubic polynomial coefficients are computed using data points from the interval $\Delta t_j + \delta\Delta t_j$. The extension, $\delta\Delta t_j$, is taken as a fixed percentage of the interval Δt_j. Both intervals are increased until the chosen error tolerances are exceeded or when the end of the data series is reached. These error tolerances can be either simple error bar tests on the residuals (i.e. the distance between smoothed and observed data points) or the Chi-squared statistical test on the a posteriori variance factor of each subsequent fit (Section 7.3.1).

The performance of the four knot selection schemes were evaluated using both the simulated and LOREX data sets. Due to the different characteristics of the simulated and LOREX test data sets, each is discussed in turn.

9.2.1 Knot Selection Algorithm Using Simulated Data

The following are the general conclusions for the simulated data set:

1. All knot selection schemes, with an adequate density of knots, are applicable (see Table 9-1).
2. Equal interval knot schemes, as a rule, do not always give the best distribution of knots. However, with a sufficient density of knots, the underlying motion or curve can be aptly fitted (see Table 9-1, Test No. 2).
3. Visual knot selection, which at times do lead to better knot distributions (Table 9-1, Test No. 2) than equal interval knot schemes, are difficult to implement.
4. The curve fitting algorithm (Table 9-1, Test No. 3 and Test No. 4) gives the best results; best in terms of lowest number of knots and with an equivalent variance factor (i.e. compare results of Test No. 3, Run No. 3 with that of Test No. 4, Run No. 2 or Run No. 3).

5. The use of the Chi-squared statistical test on the a posteriori variance factor in each of the subsequently fitted curves by the curve fitting routine (FIT) as the rejection criterion seems remarkably better than the simple error bar residual (or outlier) tests (i.e. result of Table 9-1, Test No. 3 is better than Test No. 4).
6. The interplay between the 10 option parameters (Table 9-1b) on the curve fitting routine requires further investigation and at this point no guidelines to the use of the curve fitting algorithm can be given.

9.2.2 Knot Selection Algorithm using LOREX Test Data Sets

Different knot selection schemes are used for the LOREX data sets (Table 9-2, Tests Nos. 2, 3 and 4). The curve fitting algorithm could not optimally locate the knots in the LOREX data sets. The two possible reasons for this are:

1. incorrect interplay between the parameters on the fitting routine, or
2. presence of outliers (i.e. badly determined data points) in the observed data sets.

Different combinations of the parameters of the curve fitting routine were attempted and results indicated the second reason to be the most probable cause.

Although visual inspection of Figures V-1 to V-9 in Appendix V of the LOREX test data sets suggest the presence of outliers, examination of the statistics of the recorded passes revealed no apparent reason for the discrepancies in the computed positions. In an effort to detect outliers, a linear filter routine was devised (subroutine FILTER in Appendix VII). The routine fits a straight line, via the method of of least-squares, to a subset of the data series and shifting the location of the subset in increments until the total data series is sequenced. Interpretation of the results which take the form of a time series of ratios of the residuals to its input standard deviations proved difficult. Its use was later discontinued.

Due to the great variability in terms of noise on the test data sets, the visual knot selection scheme was not used. For the two remaining knot selection schemes, no significant differences between them were detected. Equal points per interval knots (Table 9-2, Test No. 4) performed just as well as equal time interval knots (Table 9-2, Test No. 2 and Test No. 3).

For the purposes of investigating the completeness of the spline model and the inadequacies of the formal covariance matrices of the position fixes, the equal time span per cubic interval knot selection scheme was used. For each of the three test data sets, the knot time intervals were chosen at one-third, one-fifth, one-tenth and one-twentieth of the total time span of the test data series used (e.g. if the test data time span was 72 hours, then the knot time

TABLE 9-2 Results of the various test runs by DSPLIN on the 3 LOREX test data sets under different conditions.

TEST NO.	RUN NO.	NO. OF KNOTS	R.M.S. ERROR (m) TEST DATA SET NO. 1	2	3	VARIANCE FACTOR TEST DATA SET NO. 1	2	3	CHI-SQ.TEST TEST SET NO. 1	2	3	D.O.F. TEST SET NO. 1	2	3	COMMENTS CONDITIONS OF RUN
1	1	3	429.70	1,068.31	908.82	1,819.91	5,290.20	110.87	F	F	F	194	194	194	Formal covariance matrices
	2	5	387.85	692.63	909.55	482.69	389.73	97.44	F	F	F	190	190	190	from position fixes are
	3	10	385.30	579.32	912.82	289.55	89.39	64.64	F	F	F	180	180	180	used as weights.
	4	20	402.70	588.47	911.61	155.44	49.10	55.27	F	F	F	160	160	160	
2	1	3	411.10	1,039.88	821.15	5.29	78.65	2.39	F	F	F	194	194	192	Minimum semi-minor axis (B)
	2	5	380.38	657.13	819.10	1.70	5.51	2.17	F	F	F	190	190	188	of 35 m enforced. Altitude
	3	10	381.42	584.23	121.00	1.24	2.32	1.59	P	F	F	180	180	178	and velocity errors modelled.
	4	20	385.92	588.89	820.66	1.03	1.37	1.61	P	F	F	160	160	158	
3	1	8	386.19		817.29	1.35		1.82	F		F	190		188	Minimum B = 35 m. Alt and
	2	8	379.60		818.11	1.26		1.61	P		F	184		182	velocity errors modelled.
	3	4		658.09			10.69			F	F		192		Knot times at 1/2 day (12 hrs)
	4	7		618.32			3.96			F			186		and 1/4 day intervals.
4	1	3	409.89	1,083.64	821.17	4.12	92.78	2.40	F	F	F	194	194	192	Equal no. of point intervals.
	2	5	381.30	704.31	819.67	1.41	9.64	2.17	F	F	F	190	190	188	Minimum B = 35 m. Alt and
	3	10	383.96	581.77	821.00	1.05	2.12	1.50	P	F	F	180	180	178	velocity errors modelled.
	4	21	386.88	584.16	817.42	0.93	1.56	1.64	P	F	F	158	158	156	Pts per interval = [50,25,11,5] pts.

TABLE 9-2 (Continued)

TEST NO.	RUN NO.	NO. OF KNOTS	R.M.S. ERROR (m) TEST DATA SET NO. 1	2	3	VARIANCE FACTOR TEST DATA SET NO. 1	2	3	CHI-SQ. TEST TEST SET NO. 1	2	3	D.O.F. TEST SET NO. 1	2	3	COMMENTS AND CONDITIONS OF RUN
5	1	3	416.55	1,094.30	911.52	57.31	1,472.74	35.41	F	F	F	194	194	194	A minimum of 30 m std deviation
	2	5	380.15	662.83	910.28	22.50	111.34	27.17	F	F	F	190	190	190	is imposed on all formal error
	3	10	387.48	581.60	911.40	12.25	30.37	16.25	F	F	F	180	180	180	ellipses of position fixes.
	4	20	386.77	586.81	918.19	9.25	19.34	14.95	F	F	F	160	160	160	
6	1	3			820.49			33.67			F			192	Investigating the effect of the
	2	5			818.46			27.17			F			188	deletion of pt. 37 (bad pt.)
	3	10			820.67			16.13			F			176	in Test Data Set 3. All para-
	4	20			823.81			14.84			F			158	meters are as in Test No. 5.

intervals would be set at 24, 14.4, 7.2 and 3.6 hours respectively). Three knots represent the absolute minimum number of knots one can apply to any data set. The resulting data set is heavily smoothed. Twenty knots, on the other hand, ensures a sufficient density of knots to aptly portray any of the LOREX test data sets regardless of knot placements.

9.3 COMPLETENESS OF THE SPLINE MODEL

The completeness of the spline model relates directly to the choice of the number of knots for each data span. The results, in Table 9-1 and Table 9-2, show that the root mean square error (rms) and the a posteriori variance factor are effective in determining whether a sufficient number of knots have been used. When the model has an inadequate knot density, an increase in the number of knots significantly reduces the value of the root mean square of the differences between smoothed and observed positions. This improvement in the rms diminishes as the number of knots continue to rise (Table 9-1, Test No. 1). The failure of the Chi-squared statistical test (Section 7.3.1) on the variance factor can be attributed to either:

1. an incorrect or incomplete model, or
2. an incorrect a priori covariance matrix for the observed data set [Vanicek and Krakiwsky 1982: page 237].

Under a "steady state" in the rms (i.e. no appreciable change in rms follows any increase in the number of knots, e.g. Table 9-2, Test No. 1, test data set No. 3, Run nos. 1 to 3) the latter reason is given for its failure to pass the statistical Chi-squared test.

Deletion of a badly determined position fix (e.g. point 37 in test data set three was computed using severely unbalanced Dopplers about the point of closest approach) affects the rms more than the variance factor (see Table 9-2, Test Nos. 5 and 6). This suggests that the "erroneous" fix is correctly weighted within the data span.

It should be noted that the Chi-squared statistical test is only a global indication of knot sufficiency (Section 7.3.1). Knot selection at the local levels can be evaluated through the analysis of a time series of standardised residuals (i.e. r_x and r_y). The standardised residuals are defined as follows:

$$r_x := (E_p - E_s)/ \sigma_{Ep}$$

and

$$r_y := (N_p - N_s)/ \sigma_{Np}$$

where p - subscript denoting observed position coordinates (E for eastings and N for northings),

σ - formal standard deviations associated with each observed position coordinate,

and s - subscript denoting smoothed position coordinates produced by the spline (E for eastings and N for northings).

9.4 BOUNDARY (OR OUTER) KNOTS

Two boundary knot vectors, along with their a priori covariance matrices, are read together with the observed data points, into the least-squares process. For a single separate spline computation, the outer data points are usually chosen as knots.

Weighting the end knot positions as heavily as the outer data points is found to bias the final location of the outer knots; more so when the extreme points have small formal covariance matrices and are unrepresentative of the true precision of the points. In light of this, low weights are recommended in the absence of information about the end knot vectors.

9.5 SPLICING SPLINES

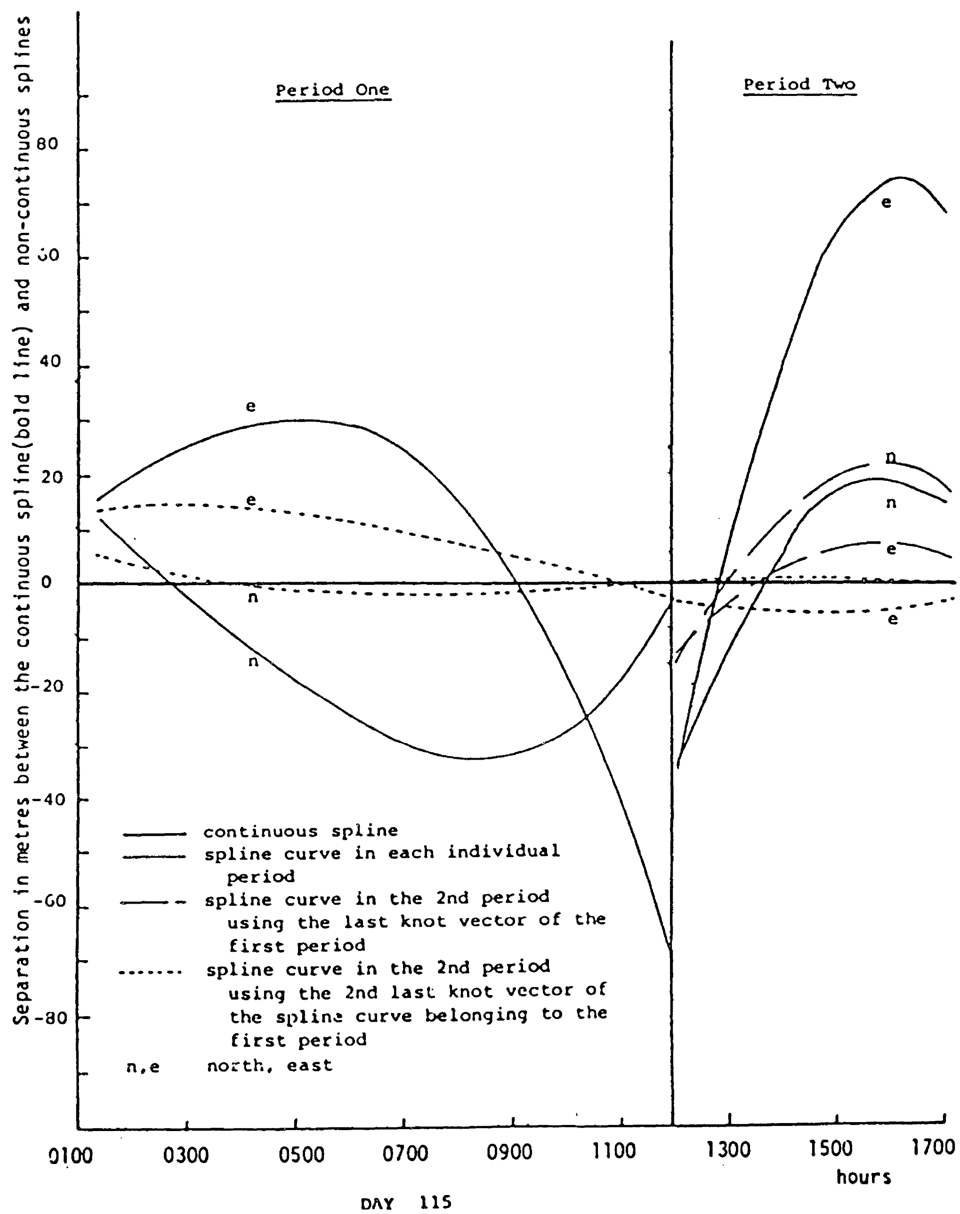
The primary purpose of the inclusion of outer knot vectors into the program is to facilitate the joining of splines from separate adjacent data spans. This is desirable when dealing with very large data series. The minimum data point overlap between adjacent splines is investigated by comparing two separate but adjacent splines computed under different conditions against a continuous spline over the whole period.

The experiments are conducted only with LOREX test data set one (Appendix V). Test data set one (which consist of position fixes for station SO from day 113 hr 14 to day 116 hr 19) is subdivided into two periods. Period one is from day 113 hr 14 to day 115 hr 12 and period two is from day 115 hr 12 to day 116 hr 19.

The following are the splines computed using DSPLIN:

1. A continuous spline over the whole period (i.e. from day 113 hr 14 to day 116 hr 19). In Figure 9-2, the continuous spline is represented by the straight bold line.
2. Separate and unspliced splines for period one (i.e. from day 113 hr 14 to day 115 hr 12) and period two (i.e. from day 115 hr 12 to day 116 hr 19). These splines are represented by curved unbroken lines in Figure 9-2.
3. Second spline in period two computed with the estimated end knot vector and its covariance matrix, from the first spline in period one, as its beginning knot vector. This is represented by the long dashed lines in Figure 9-2.
4. Second spline is computed using the second last estimated knot vector from the first spline together with a common overlap of data points (i.e. the spline is computed using data points from the period day 115 hr 0 to day 116 hr 19; with day 115 hr 0 being the

FIGURE 9-2 Comparisons between separate and continuous splines. The effect of supplying a beginning knot vector to the second spline is clearly illustrated by the plot below.



location of the second last knot vector of the first spline). This is portrayed in Figure 9-2 by the short dashed lines.

Refering to Figure 9-2, the use of the end knot vector (and its a posteriori covariance matrix) of the first spline as the beginning knot vector of the second spline, allows the latter spline to 'settle down' faster. The discrepancy between the second spline and the continuous spline, in the first knot interval, diminishes further when, instead of the last knot vector, the second last knot vector of the former spline, along with a common overlap of data points, is used.

Apparently the second spline, with a weighted outer knot vector, demands a one knot interval overlap to damp out the oscillations in its front segment. In light of this, all adjacent splines should contain at least two common knot intervals and points, with the first splined values read up to the second last knot and the second set of values starting from the second knot (illustrated in Figure 9-3).

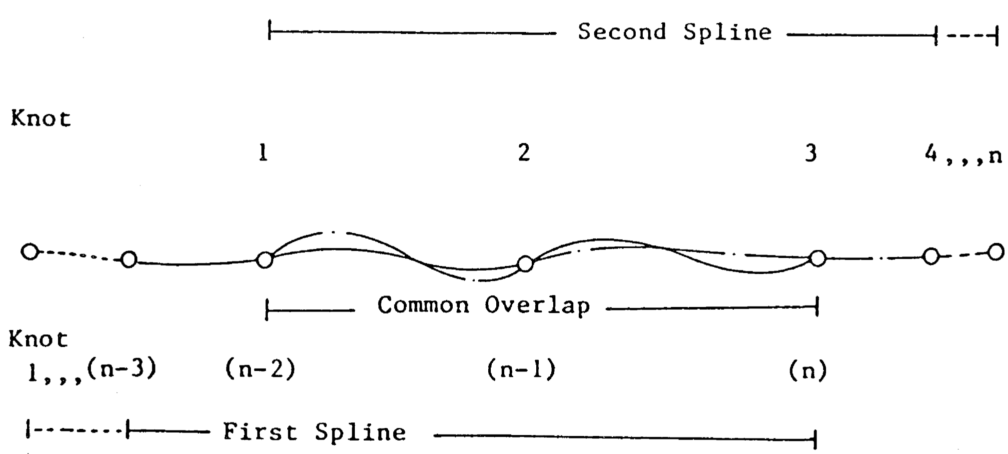


FIGURE 9-3 Minimum Overlap Between Splines

Chapter 10

DISCUSSION, EVALUATION AND APPLICATION OF THE SPLINE ALGORITHM TO THE LOREX DATA SETS

In this chapter, the orbital separation between the NNSS satellites during the period of the expedition are given, the processing sequence of the LOREX data sets are outlined, the error models used to modify the formal covariance matrix of the position fixes are developed, the real-time application of the spline algorithm is discussed, results of the comparisons between DSPLIN and GEODOP, and SMOBS and GEODOP are given, and the results from the processing of the three LOREX data sets from the ice stations are given and the difficulties investigated.

The general outline of the processing of the three LOREX data sets is given in Figure 10-1.

10.1 SEPARATION BETWEEN TRANSIT SATELLITE ORBITS

The three LOREX test data sets (Appendix V) were visually examined prior to the processing to determine if there exists any periodic degradation, as indicated by a larger scatter of position fixes about the mean, within the data series. Visual inspection of the plots of the test data

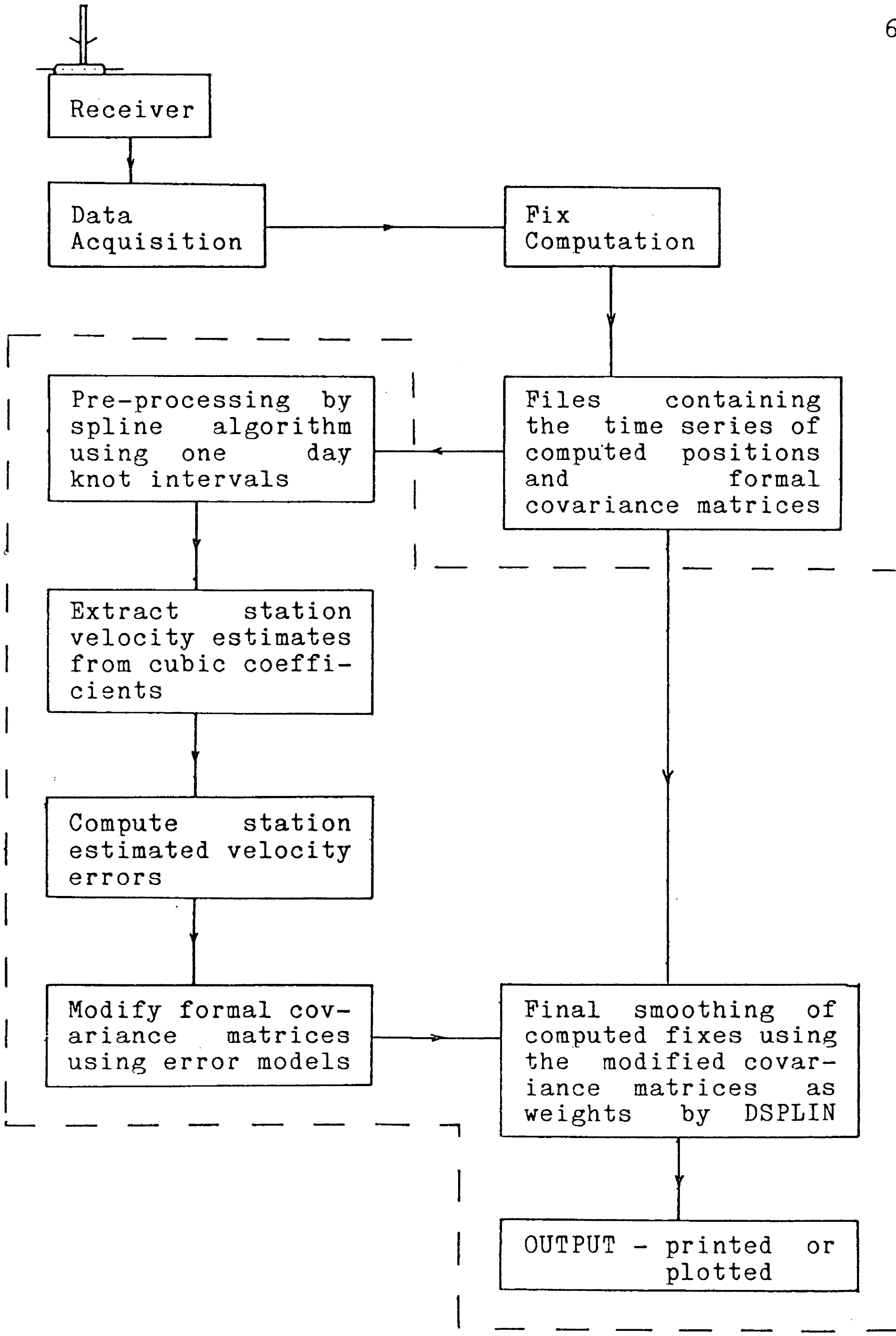


FIGURE 10-1 LOREX Satellite Data Processing Flow-chart. Enclosed within the broken line is the processing procedure used in this thesis.

sets reveals an apparent periodic degradation at 12 hour intervals, i.e. at 0 and 1200 hours each day (see Figures V-1 to V-9). This could have been caused by the orbital configuration of the Transit satellites during the period of the expedition. Figure 10-2, drawn from the orbit parameters of the operating satellites published by USNO [1979], shows the orbit of the satellites to be within a 90 degree quadrant. The closeness of the orbital planes and direction of motion of the satellites may have contributed to the observed periodic degradation.

10.2 STATION VELOCITY ESTIMATES

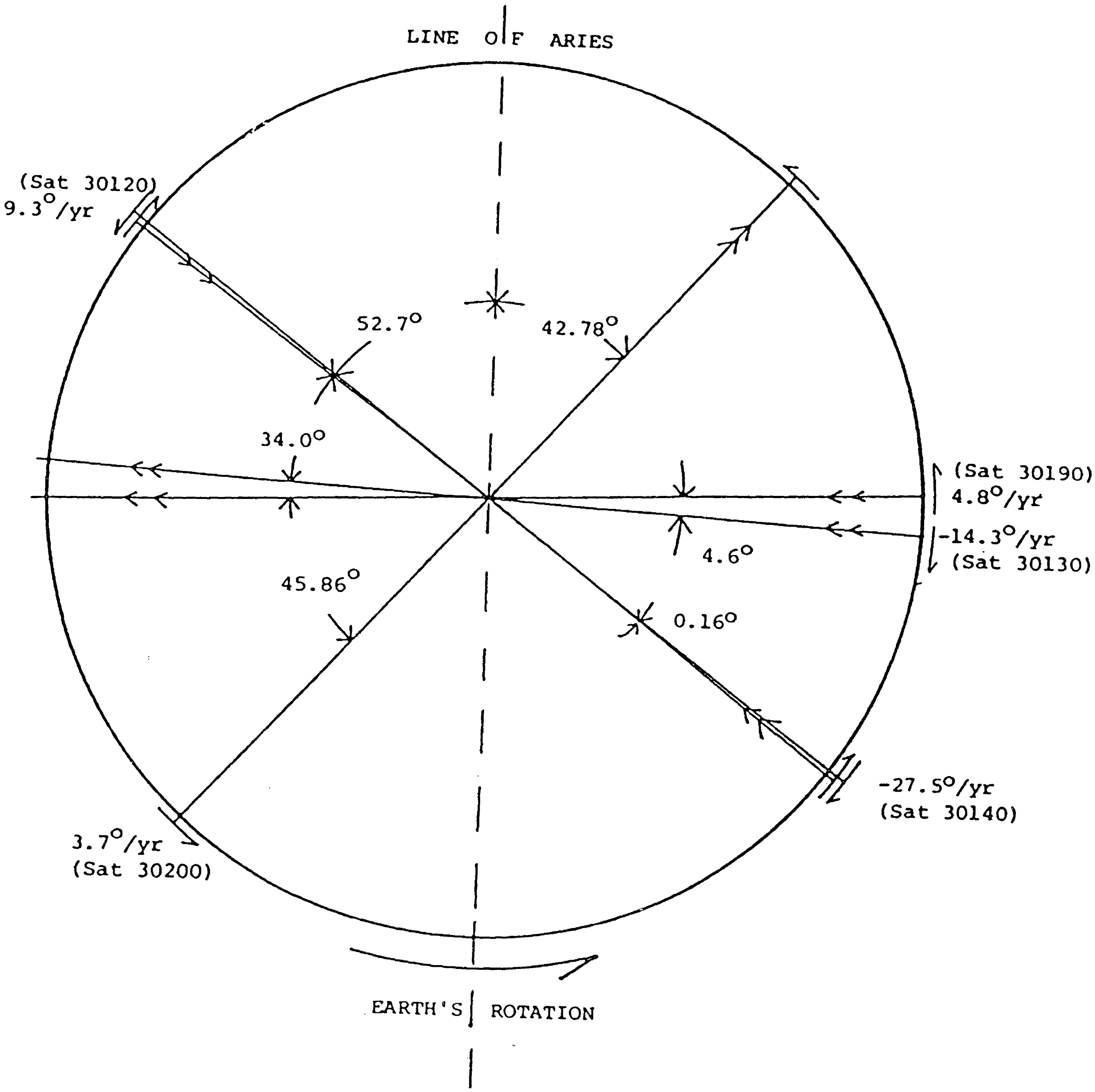
For each of the three test data sets there exist three sets of velocity vectors:

1. Real-time predicted speeds and azimuths.

   These are the speeds and azimuths used in the real-time computation of the position fixes.

2. Pre-processed splined speeds and azimuths.

   The spline algorithm is used to generate a set of smoothed velocities and azimuths for the stations. To extract the long term trends in the motion of the station, the spline is computed using one day knot intervals over the full data series.



NORTH POLE VIEW

FIGURE 10-2 ORBITAL SEPARATION OF 5 OPERATIONAL TRANSIT SATELLITES ON MAY 1ST 1979

3. Final splined velocities and azimuths.

   These are the final set of speeds and azimuths produced by program DSPLIN incorporating all the modifications described in this chapter.

The error in the estimated velocity of the receiver, as will be described in Section 10.3.2, is the difference between final splined velocities and real-time predicted values. An approximate velocity error is the difference between the pre-processed splined values and the real-time predicted velocities. In this thesis, however, the real-time predicted velocity estimates are assumed to be nonexistant and the receiver velocity error is that computed by the pre-processing spline. The real-time predicted velocity estimates could not be easily extracted from the existing Doppler data.

10.3 A PRIORI COVARIANCE MATRICES

The spline algorithm requires that the a priori covariance matrix adequately reflect the accuracy of the position determinations. The formal covariance matrices obtained from the fix computations appears incorrect from the results of the Chi-squared statistical tests (Table 9-2, Test No. 1). In this section, a description and justification of the procedures used to modify the formal covariances of the computed position fixes are given. The results of each modification to the a priori covariances are evaluated.

Hoar [1982] and Stansell [1978] stated that a single pass point positioning accuracy for a stationary receiver is in the vicinity of 37 metres (at one sigma level) and between 27 to 37 metres rms respectively. The imposition of a minimum of 20 metres (Table 10-1, Test No. 1 and Figure V-1a) or 30 metres (Table 9-5, Test No. 5) on the standard deviations of the passes (i.e. all fixes with a standard deviation of less than the minimum are scaled up equally in eastings and northings) results in a dramatic drop in the magnitude of the variance factor. This further supports the hypothesis that the a priori covariance matrices, resulting from the fix computation (based on the residuals of the Doppler observations) are unrepresentative of the accuracy of the position determinations. The effect of errors in the satellite coordinates and those due to receiver velocity errors are not represented in these formal covariance matrices.

10.3.1 Modelling Orbital Errors

Wells [1974] mentioned that the broadcast predicted satellite orbits have, approximately, 26, 11 and 5 metres standard deviations in the along, radial and cross track components respectively. The uncertainity in the computed positions can be no smaller than the uncertainities in the position of the satellite. A value of 30 metres, based on the approximate precision of the along track position of the satellite, was chosen as the mimimum allowable semi-minor axis for the error ellipse of the fixes. In order to assess the impact of setting this constraint on the formal covariance matrices of the position fixes, this figure was increased to 35 metres and the test runs (done under the 30 metres minimum semi-minor axis) repeated. Formal error ellipses which had semi-minor axes of less than the minimum were scaled up accordingly. The scale factors thus range from 1 to 30 (or 35). The imposition of a minimum of one metre on the semi-minor axis of the formal error ellipse prior to scaling limits the applied scale factor to within the said range. The scaled error ellipses are then translated via equations (VI-5) developed in Appendix VI to their equivalent standard deviations and covariances (subroutine ELLSIG in Appendix VII).

Test No. 2 and Test No. 6 (Table 10-1) performed under the above mentioned conditions show an improvement in the variance factor; with test data set one passing the statistical Chi-squared test. Test data sets two and three, however, still fail to pass the Chi-squared test even with the maximum number of knots. This suggests the presence of yet unmodelled errors (Section 9.3).

TABLE 10-1 Results of the various test runs by DSPLIN on the 3 LOREX Test Data Sets under different conditions.

TEST NO.	RUN NO.	NO. OF KNOTS	R.M.S. ERROR (METRES) TEST DATA SET NO. 1	2	3	VARIANCE FACTOR TEST DATA SET NO. 1	2	3	CHI-SQ. TEST TEST SET NO. 1	2	3	D.O.F. TEST SET NO. 1	2	3	COMMENTS AND CONDITIONS OF RUN
1	1	3	415.30	408.27	911.64	103.65	2,204.77	48.21	F	F	F	194	194	194	A minimum of 20 m std. deviation is
	2	5	379.79	668.01	910.60	40.73	147.29	38.73	F	F	F	190	190	190	imposed on all formal errors of
	3	10	387.38	582.25	914.76	21.06	43.65	23.13	F	F	F	180	180	180	position fixes.
	4	20	385.90	587.24	918.17	15.05	25.11	20.64	F	F	F	160	160	160	
2	1	3	409.47	1,016.33	821.55	5.51	138.92	3.12	F	F	F	194	194	192	A minimum semi-minor axis (B) of
	2	5	379.87	647.99	819.21	2.09	9.69	2.82	F	F	F	190	190	188	30 m is enforced on all formal
	3	10	383.05	582.44	820.96	1.56	3.69	2.05	F	F	F	180	180	178	error ellipses.
	4	20	386.49	585.40	820.77	1.21	2.40	2.06	F	F	F	160	160	158	
3	1	3	409.49	1,015.70	821.15	4.05	102.84	2.39	F	F	F	194	194	192	A minimum semi-minor axis (B) of
	2	5	379.88	648.21	819.11	1.54	7.21	2.17	F	F	F	190	190	188	35 m is enforced on all formal
	3	10	383.06	582.34	821.00	1.15	2.74	1.59	F	F	F	180	180	178	error ellipses.
	4	20	386.45	585.52	820.67	0.89	1.79	1.61	P	F	F	160	160	158	
4	1	3	409.42	1,050.31	821.25	5.50	96.12	3.12	F	F	F	194	194	192	A minimum B of 30 m is enforced;
	2	5	379.84	658.64	819.21	2.09	6.77	2.82	F	F	F	190	190	188	with altitude, X-track and along
	3	10	383.08	584.67	820.96	1.55	2.83	2.05	F	F	F	180	180	178	track velocity errors modelled.
	4	20	386.50	589.42	820.76	1.21	1.67	2.06	F	F	F	160	160	158	

TABLE 10-1 (Continued)

TEST NO.	RUN NO.	NO. OF KNOTS	R.M.S. ERROR (m) TEST DATA SET NO. 1	2	3	VARIANCE FACTOR TEST DATA SET NO. 1	2	3	CHI-SQ.TEST TEST SET NO. 1	2	3	D.O.F. TEST SET NO. 1	2	3	COMMENTS AND CONDITIONS OF RUN
5	1	3	410.26	1,050.32	821.43	6.06	95.97	2.39	F	F	F	194	194	192	Minimum B = 30 m. Alt and velocity errors modelled. If satellite elev. > 88°, semi-major axis (A) is set to 3 km.
	2	5	380.08	658.73	819.43	2.17	6.69	2.08	F	F	F	190	190	188	
	3	10	383.37	584.86	821.43	1.57	2.76	1.24	F	F	P	180	180	178	
	4	20	386.54	589.88	822.14	1.19	1.59	1.17	F	F	P	160	160	158	
6	1	3	410.77	1,039.94	821.35	4.58	78.53	1.80	F	F	F	194	194	192	As in the above but with minimum B = 35 m.
	2	5	380.21	657.25	819.34	1.61	5.45	0.57	F	F	F	190	190	188	
	3	10	383.46	584.44	821.49	1.16	2.27	0.95	P	F	P	180	180	178	
	4	20	386.60	589.33	822.11	0.87	1.30	0.89	P	P	P	160	160	158	

10.3.2 Modelling Station Velocity Errors

Stansell [1978] gives the two possible components of a Transit fix error as being the inherent system errors (e.g. orbit and propagation errors) and errors caused by unknown receiver velocity during a satellite pass. The effect of such a velocity error remains as a complex function of satellite pass geometry and the structure of the error in velocity. The magnitude of velocity-caused errors is portrayed graphically in Figures 10-3 and 10-4. Under normal Transit use, the maximum satellite elevation acceptable for a fix computation is limited by this error.

However, in the vicinity of the North Pole, almost all passes have high elevation angles and unknown velocity-caused errors have hence to be contended with and accounted for in the a priori covariance matrix of the computed positions. The procedure used in an attempt to better model the influence of velocity errors, by modifying the a priori covariance matrix, is described below.

It is assumed that all Transit satellites have a 90 degree orbit inclination. This introduces a small error into the computed elevation angle. Using USNO [1979] data (April and July 1979), it can be shown that the maximum error caused by this assumption is within one to three degrees for all satellites except Satellite No. 30140. For Satellite No. 30140, due to its much larger departure from a 90 degree orbital plane inclination, the maximum possible error on the computed elevation angle is about five degrees.

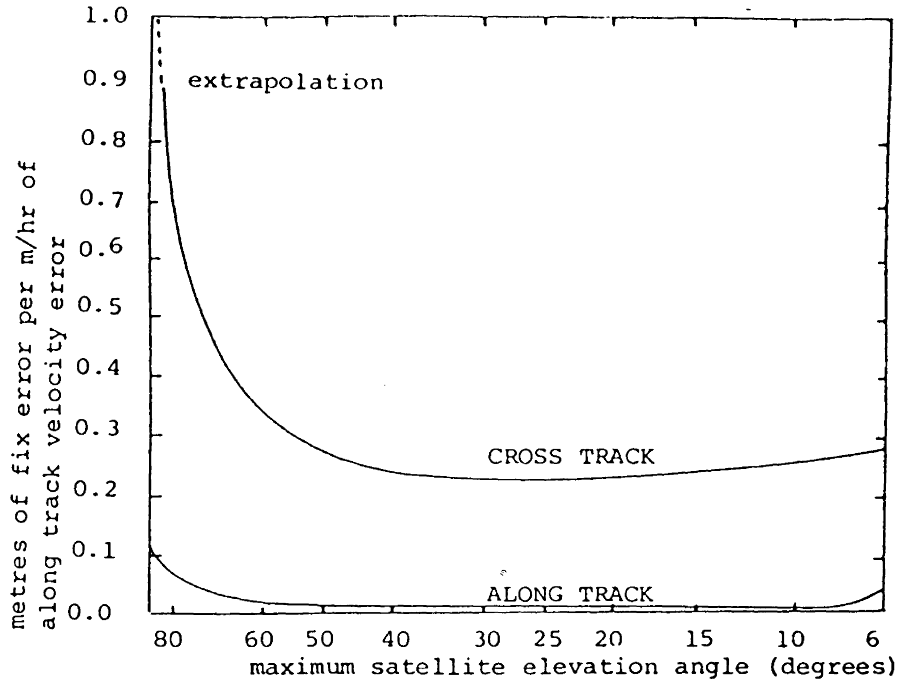


FIGURE 10-3 Sensitivity of satellite fix to a one metre per hour error in velocity in the along-track direction.

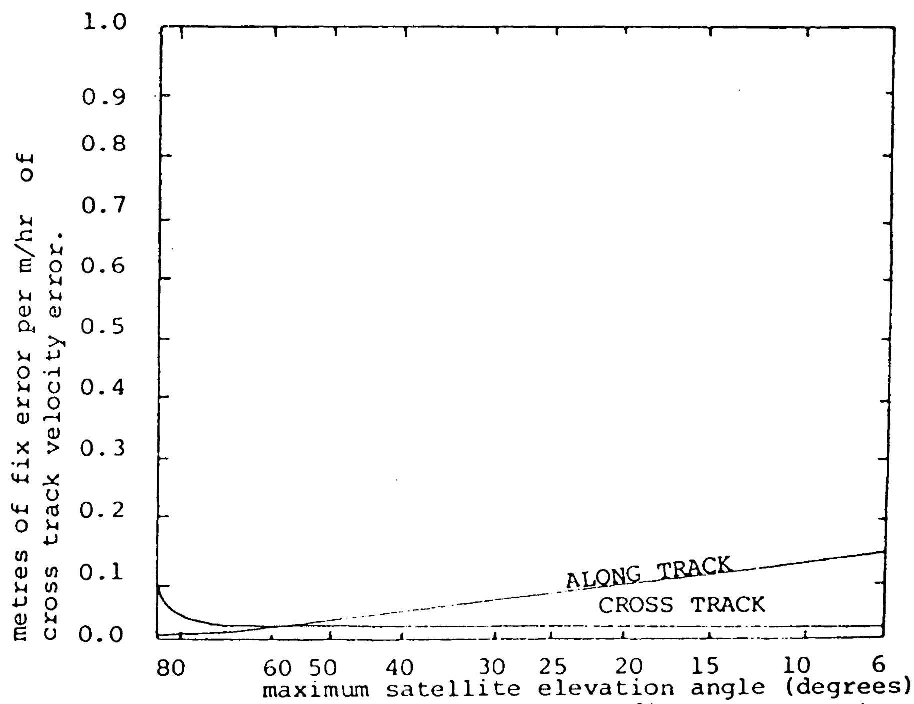


FIGURE 10-4 Sensitivity of satellite fix to a one metre per hour error in velocity in the cross-track direction.

The elevation angle for each fix is computed using the algorithm described in Appendix III. The direction of motion of the satellite at the time of closest approach (correct to within a few degrees) is derived from the formal error ellipse of the computed position. The semi-minor axis of the error ellipse lies in the along track direction of the passing satellite because the position fix is more precisely determined in the along track direction of the satellite than in the cross track direction.

The position fixes used in the test runs were computed on LOREX in real-time using predicted velocity vectors. Errors in the values of the predicted velocity vector (Figures 10-3 and 10-4) propagate into the computed position determinations.

At this point, the analytical expression underlying the curves in Figures 10-3 and 10-4 could be derived and used. However a simpler procedure is adopted here. The curves shown in Figures 10-3 and 10-4 are approximated by step functions and extrapolated towards the 90 degree elevation angle (as described in Table 10-2). The fact that the extrapolated figures might prove inadequate cannot be ignored. However, can a reliable estimate of the cross track position be estimated with satellites having elevation angles of greater than 88 degrees (termed here as "high elevation" satellites) ? A move towards answering this question is made by treating all passes with a maximum

TABLE 10-2

STEP FUNCTIONS APPROXIMATING THE FIX ERROR DUE TO

VELOCITY AND HEIGHT ERRORS.

The following are the step functions used in approximating the Figures 10-3, 10-4 and 10-5, divided into cross track and along track directions of the observed satellite.

I) Cross Track Error Components

Satellite elevation angle (e in degrees)	Fix error in metres

a) Errors due to an estimated velocity north error (Vn)

$e \leq 75$	0.45*Vn
$75 < e \leq 80$	0.56*Vn
$80 < e \leq 84$	0.62*Vn
$84 < e \leq 85$	1.00*Vn
$85 < e \leq 88$	1.20*Vn
$e \geq 88$	1.50*Vn

b) Errors due to an estimated velocity east error (Ve)

$e \leq 80$	0.03*Ve
$80 < e \leq 85$	0.05*Ve
$85 < e \leq 87$	0.09*Ve
$e > 87$	0.12*Ve

c) Errors due to an estimated height error (H)

$e \leq 75$	3.30*H
$75 < e \leq 80$	4.20*H
$80 < e \leq 83$	5.60*H
$83 < e \leq 85$	7.20*H
$85 < e \leq 86$	10.00*H
$e \geq 86$	15.00*H

II) Along Track Error Components

Satellite elevation angle (e in degrees)	Fix error in metres
a) Errors due to an estimated velocity north error (Vn)	
$e \leq 75$	0.03*Vn
$75 < e \leq 80$	0.04*Vn
$80 < e \leq 83$	0.08*Vn
$83 < e \leq 85$	0.09*Vn
$85 \leq e \leq 87$	0.10*Vn
$e > 87$	0.13*Vn
b) Errors due to an estimate velocity east error (Ve)	
$e < 80$	0.02*Ve
$80 \leq e < 85$	0.15*Ve
$e \geq 85$	0.10*Ve
c) Errors due to an estimate height error (H)	
$e < 75$	0.25*H
$75 \leq e < 80$	0.35*H
$80 \leq e < 85$	0.50*H
$85 \leq e < 87$	0.75*H
$e \geq 87$	1.00*H

The root sum square (rss) fix errors, in both cross track and along track directions, are defined as being the square root of the sum of squares of the computed fix errors from the three error sources described above. For example,

$$\text{rss(cross track)} := \sqrt{a^2 + b^2 + c^2}$$

where a, b and c are the estimated fix errors computed using the step functions Ia, Ib and Ic respectively.

[Reference: Stansell 1978: pages 72, 73 and 68] satellite elevation angle of greater than 88 degrees as line fixes (or position fixes with very elongated ellipses). The term "line fix" is used to denote a computed position with a formal error ellipse having a semi-major axis of greater than 3000 metres.

10.3.3 Modelling Height Errors

Also considered are the errors in the computed position due to an error in the assumed station height. When using high elevation satellites (as defined in Section 10.3.2), height errors, as shown in Figure 10-5, can cause a substantial error in the computed position. Popelar et al. [1981] in their post-processing of the LOREX navigation data sets arrived at an average figure of around eight metres for the station ellipsoidal heights.
This is different from the assumed height of the stations used in the fix computations on LOREX by 12 metres.

10.3.4 Combined Effects on the Formal Covariance Matrix

Using the above information, the contribution from each of the error sources, described in Section 10.3.1, 10.3.2 and 10.3.3, are evaluated. The resulting root sum square error (rss) (obtained from the position errors due to velocity and height errors - Table 10-2), divided into along and cross track components, replaced the semi-minor and semi-major axes of the formal error ellipse from the fix

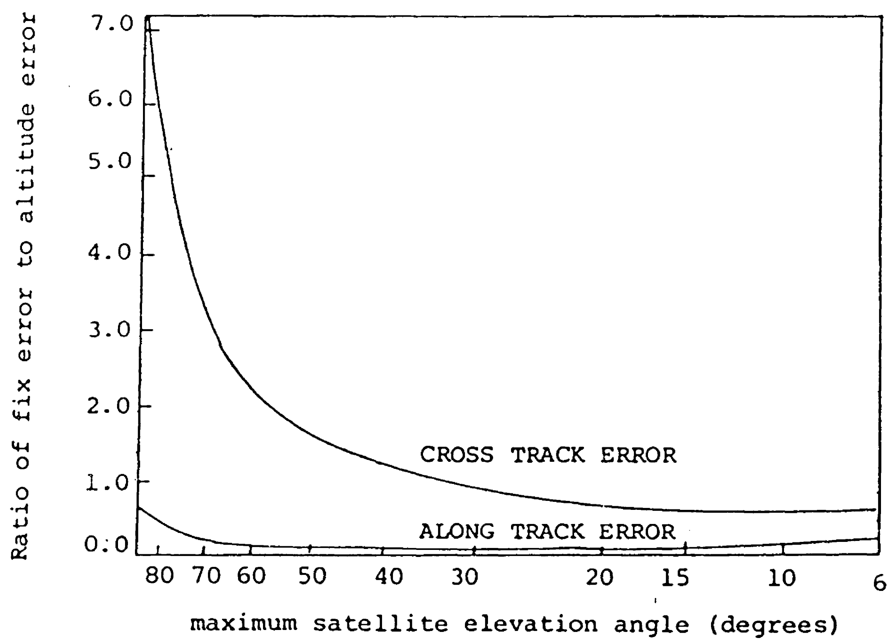


FIGURE 10-5 Sensitivity of satellite fix to station altitude estimate error

computation (subroutine ERRORS in Appendix VII) whenever these were below the rss figures. Test No. 4 of Table 10-1 used this modification to achieve a further improvement in the variance factor. The maximum effect, as expected, is shown by the test data set describing the fastest motion of station (i.e. test data set two).

To evaluate the effect of treating position fixes computed from high elevation satellites as line fixes (defined in 10.3.2), all position fixes computed with high elevation satellites had the semi-major axis of their error ellipses set to 3000 metres. The outcome of this experiment (Table 10-1, Test Runs Nos. 5 and 6) agrees with the notion that the cross track position determinations using high elevation satellites are extremely poor. The improvement in the variance factor increases as the station get nearer to the pole. This can be expected as test data set three has more position fixes with high elevation satellites (55 percent of test data points) than test data set two (14 percent) or test data set one (7 percent). In addition, the examination of the predicted covariance matrices for the smoothed points (in test data set three) revealed an average drop around 22 percent in the variances of the position coordinates (the actual figures range from -14 percent to 25 percent).

Since the spline model for all the three test data sets passes the Chi-squared tests when the revised a priori covariance matrices is used (Table 10-1, Test No. 6), these revised matrices can be assumed to aptly reflect the accuracy of the observed position fixes. For the spline algorithm and indeed any weighted smoothing technique, this condition must be fulfilled.

The time series of predicted positions for each of the three test data sets (Table 10-1, Test No.6) are given in Figures 10-6 to 10-14.

10.4 REAL-TIME APPLICATION

The near real-time application of the algorithm developed in this thesis would involve the processing of the observed data series in consecutive data spans. As described in Section 9.5, a minimum overlap of two knot intervals is required to ensure continuity. The estimated second last knot vector of the former spline, along with its covariance matrix, is used as the weighted a priori beginning knot vector for the latter spline.

The processing of the Doppler data from the three ice camps on the LOREX-79 expedition proceeded in a manner similiar to that described above. Real-time position determinations using the Transit satellite system over variable time spans (depending on the motion of the station and the resulting number of knots) are used by DSPLIN to produce smoothed estimated paths of the drifting ice

FIGURE 10-6 SMOOTHED THREE HOUR POSITIONS WITH ESTIMATED ERROR ERROR ELLIPSES AT THE 99% CONFIDENCE LEVEL

TEST DATA SET ONE

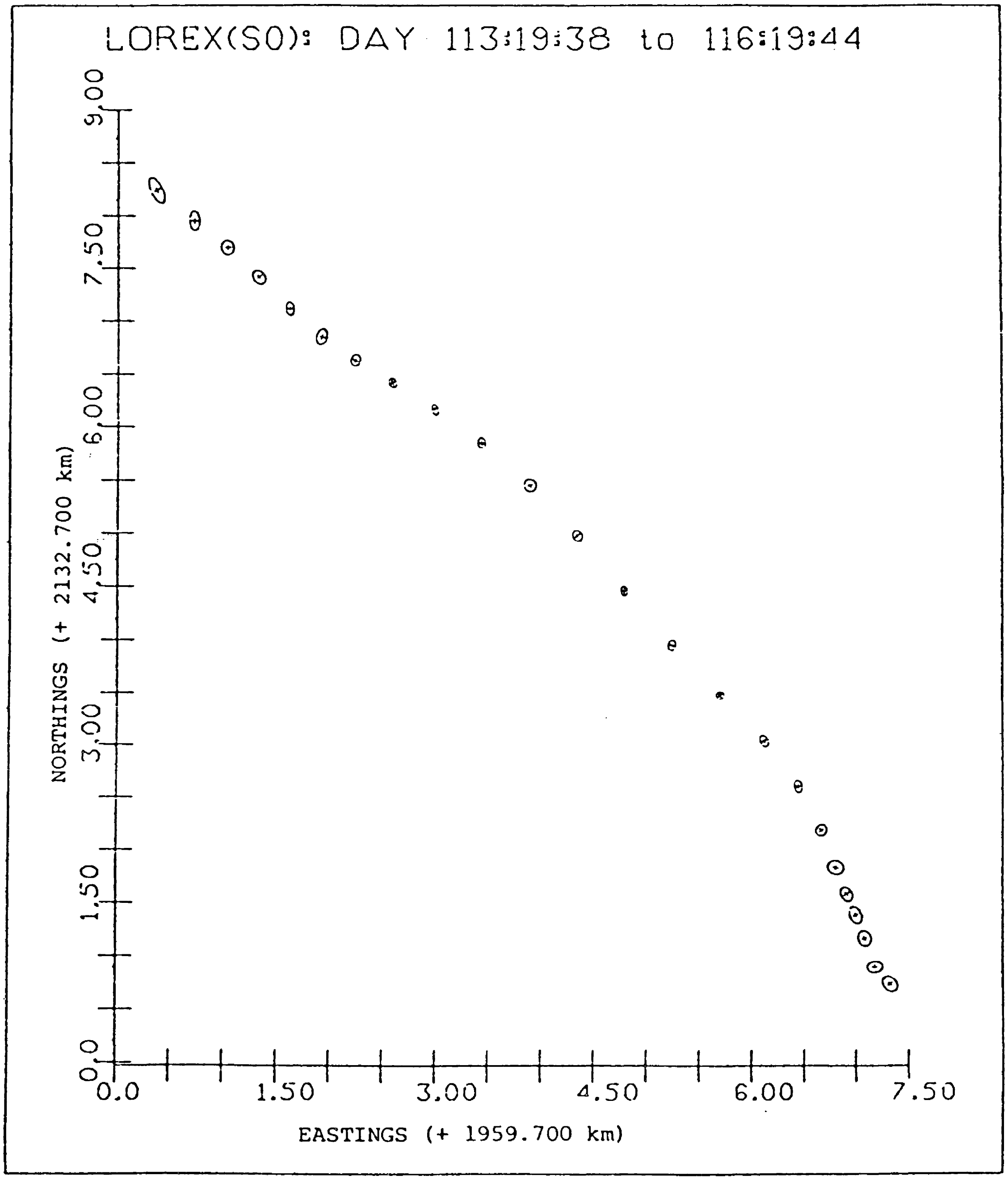


FIGURE 10-7 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET ONE

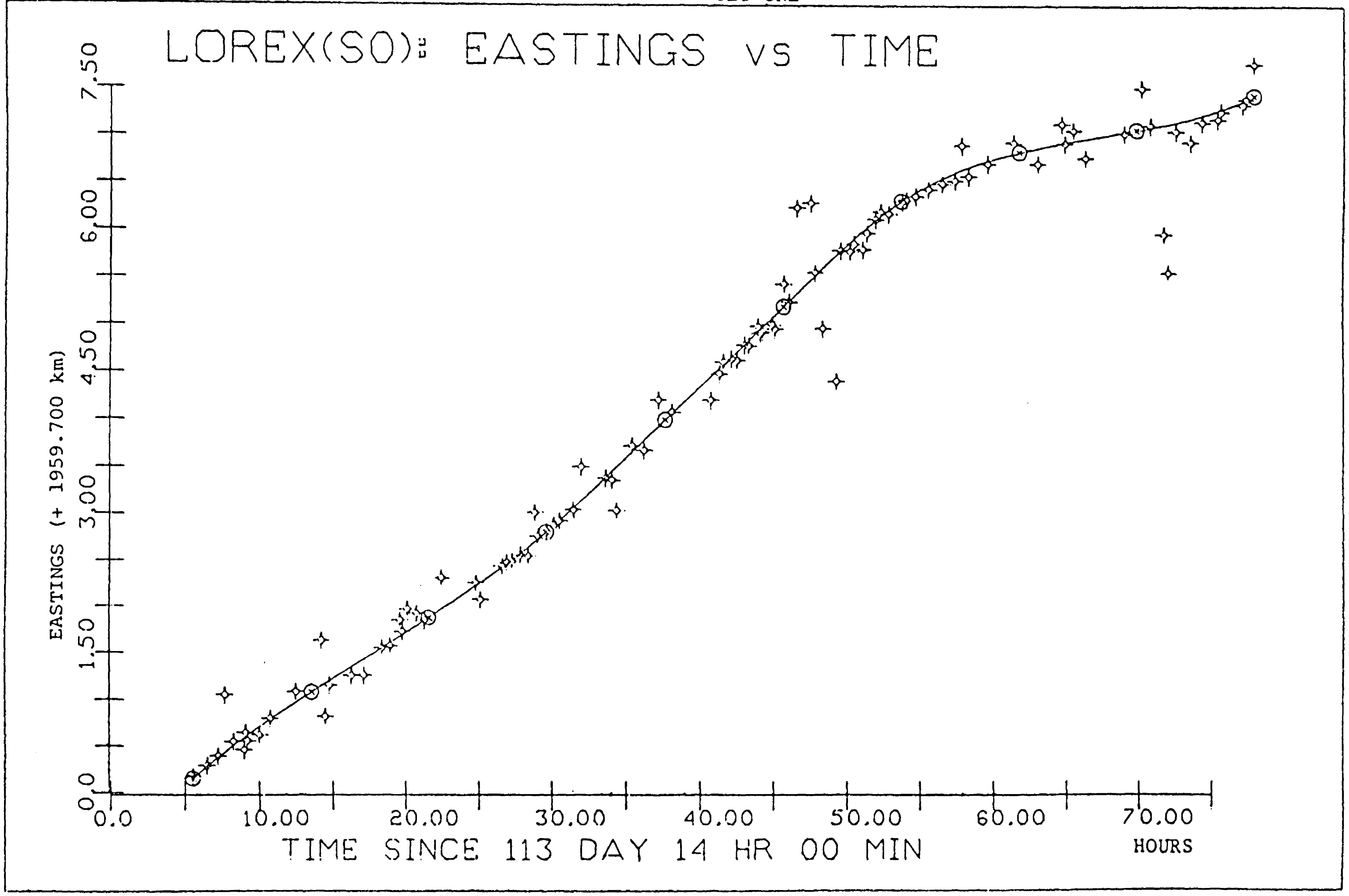


FIGURE 10-8 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET ONE

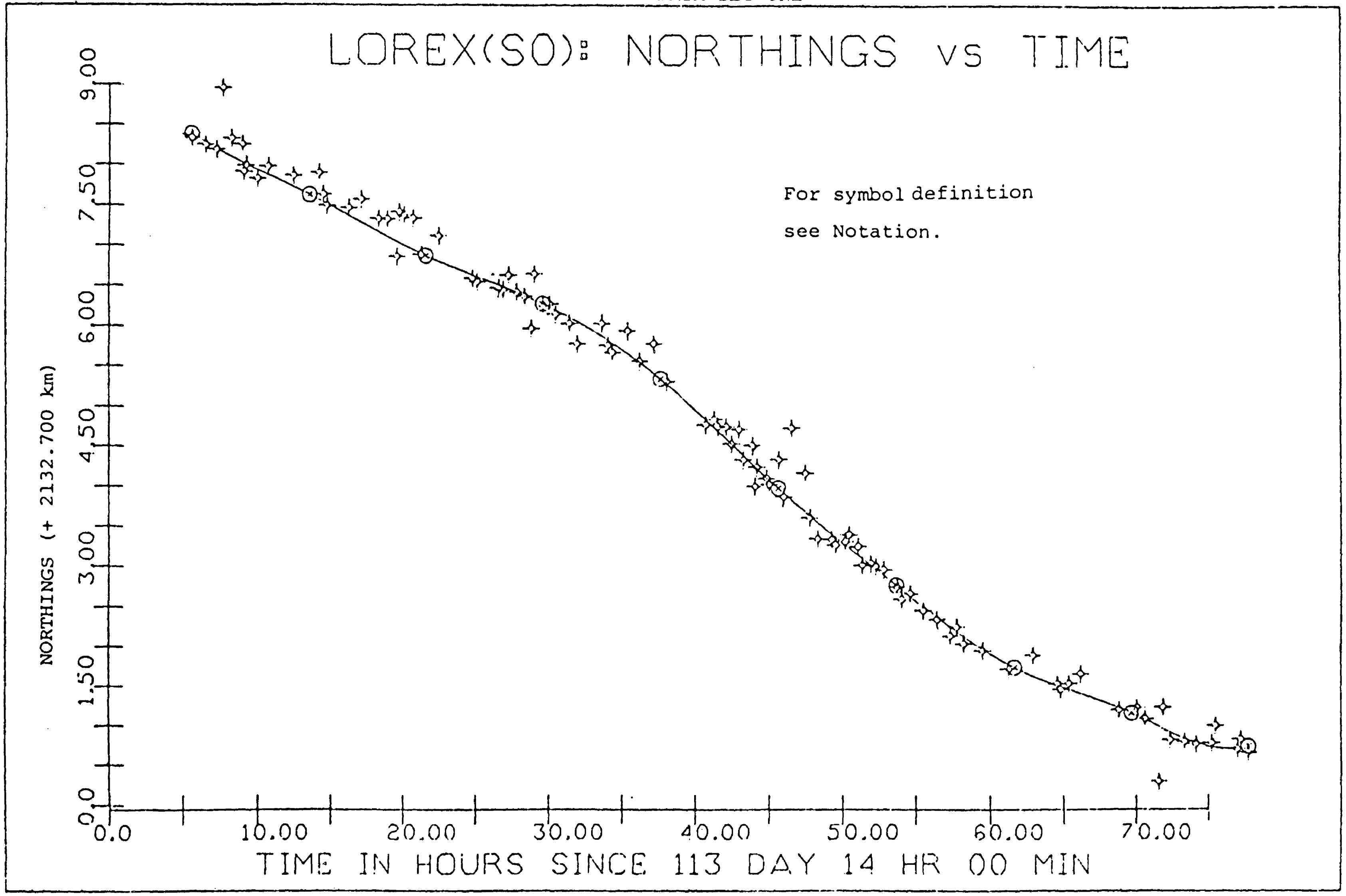


FIGURE 10-9 SMOOTHED THREE HOUR POSITIONS WITH ESTIMATED ERROR ELLIPSES AT THE 99 % CONFIDENCE LEVEL

TEST DATA SET TWO

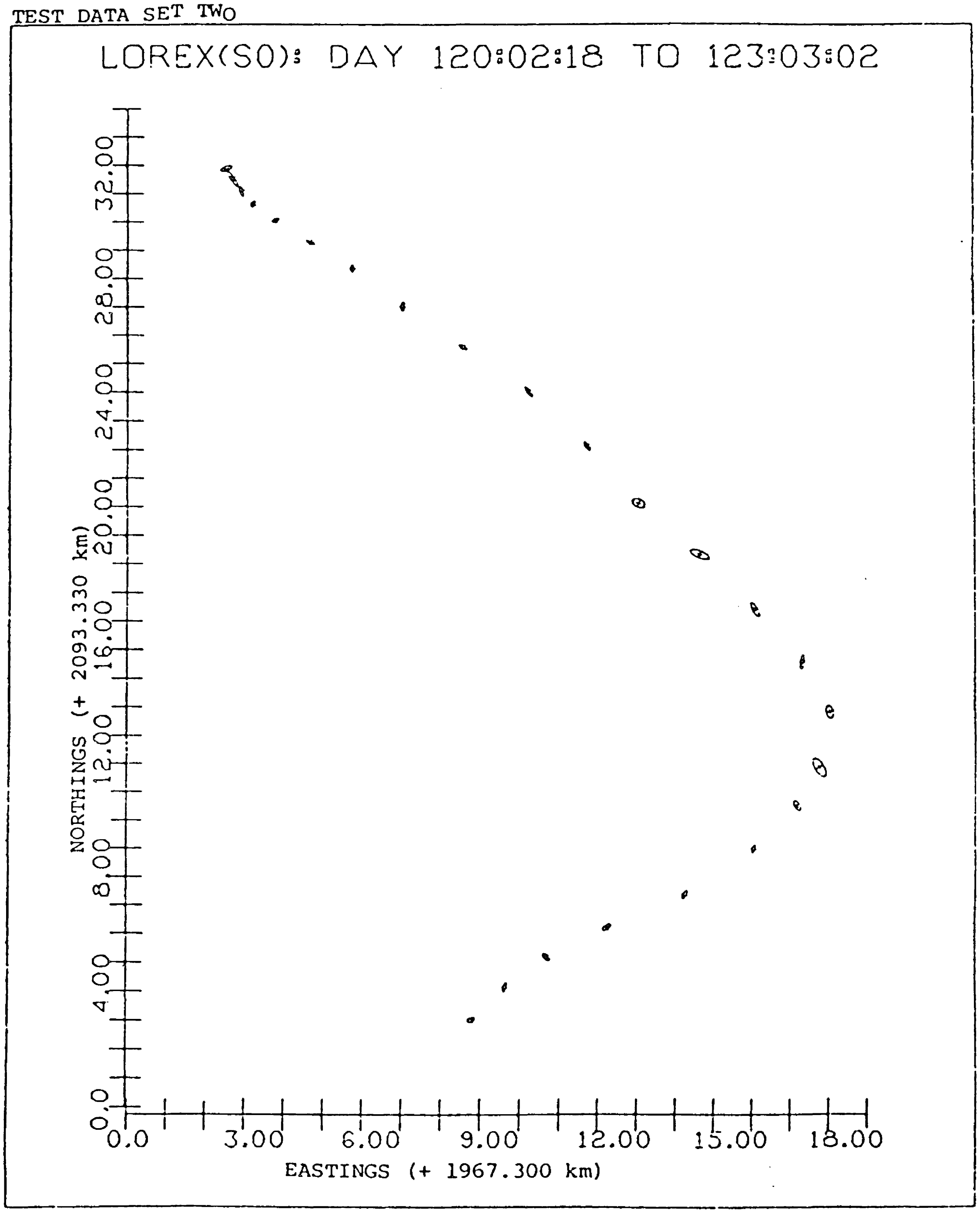


FIGURE 10-10 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET TWO

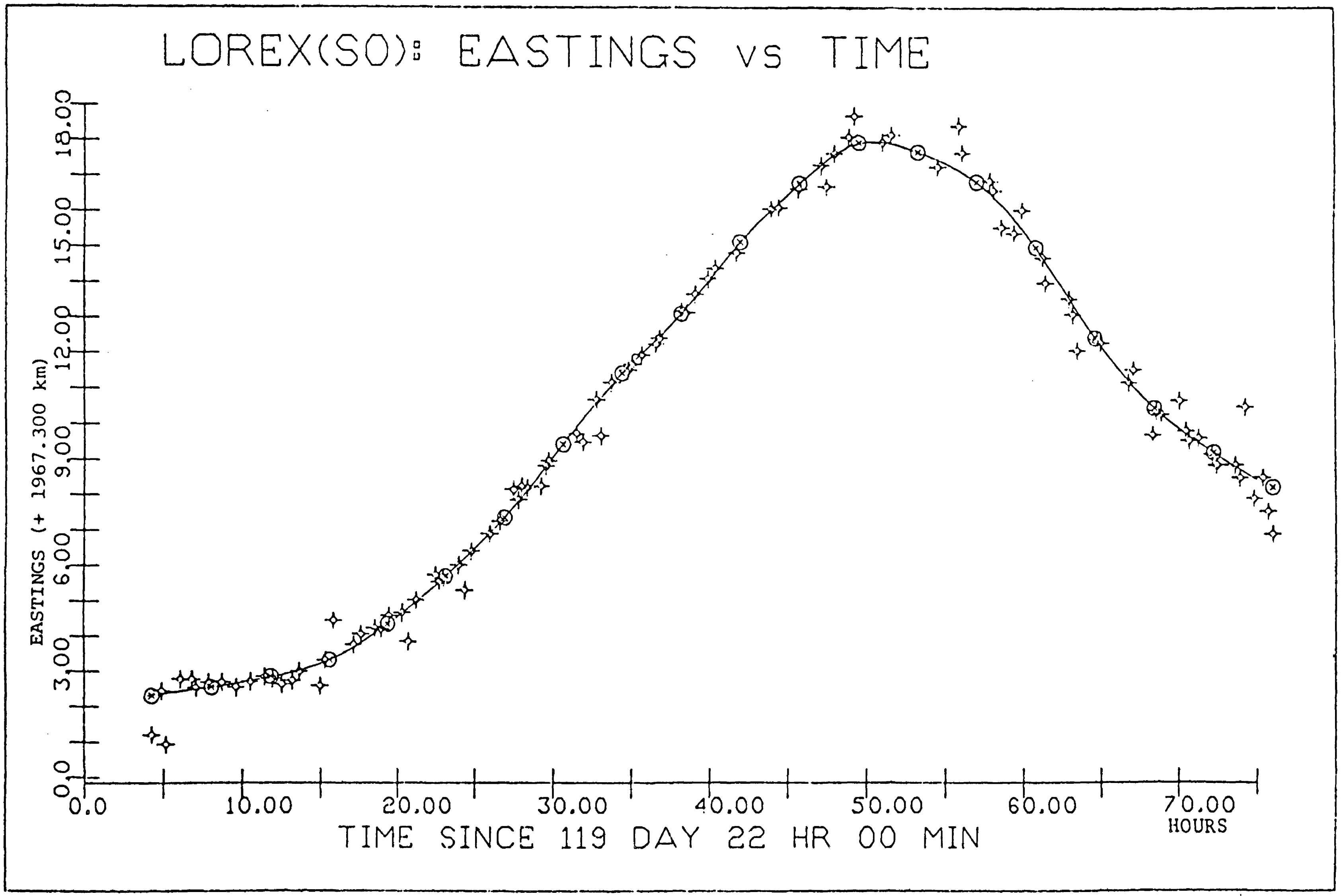


FIGURE 10-11 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET TWO

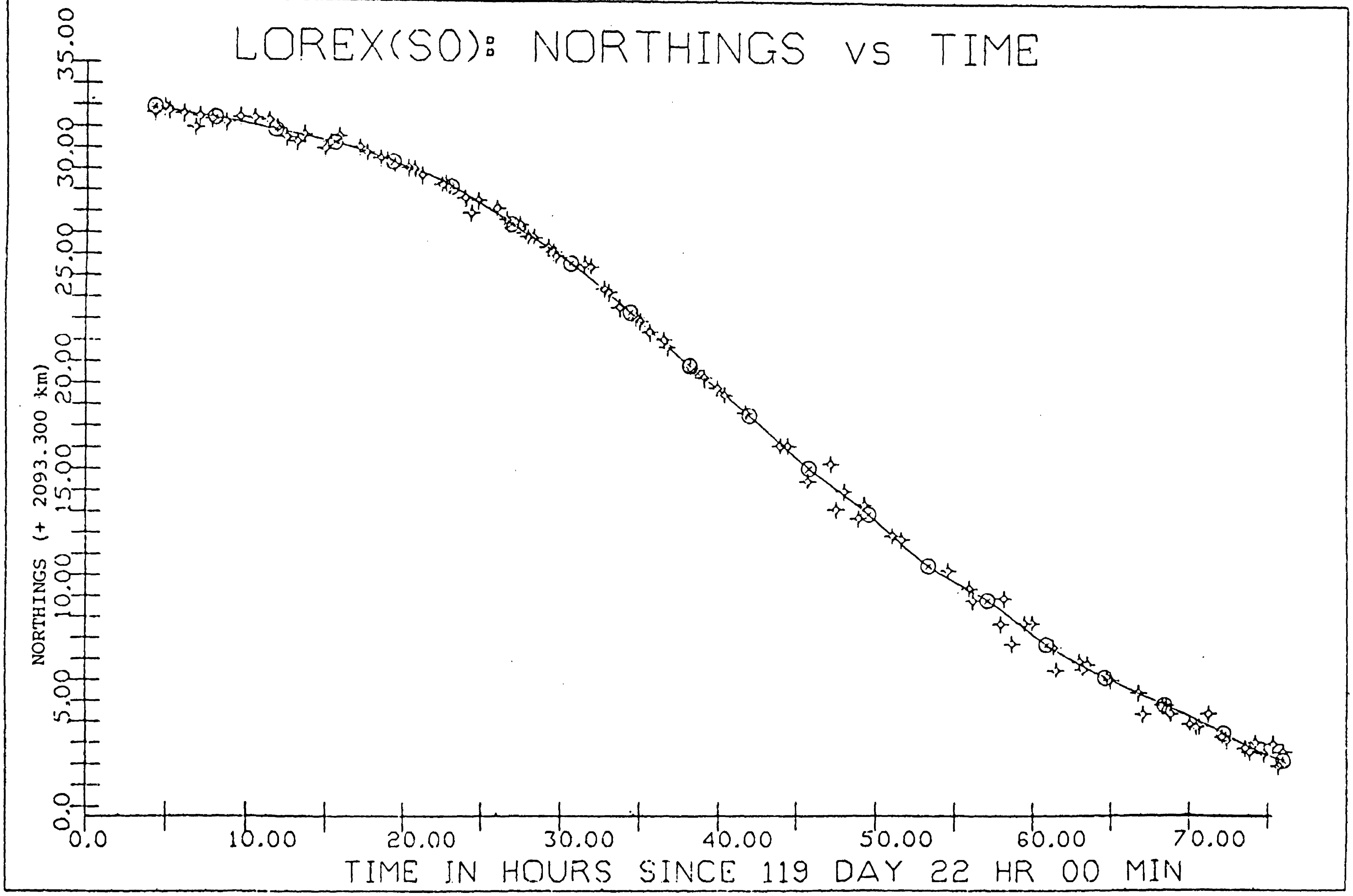


FIGURE 10-12 SMOOTHED THREE HOUR POSITIONS WITH ESTIMATED ERROR ELLIPSES AT THE 99 % CONFIDENCE LEVEL

TEST DATA SET THREE

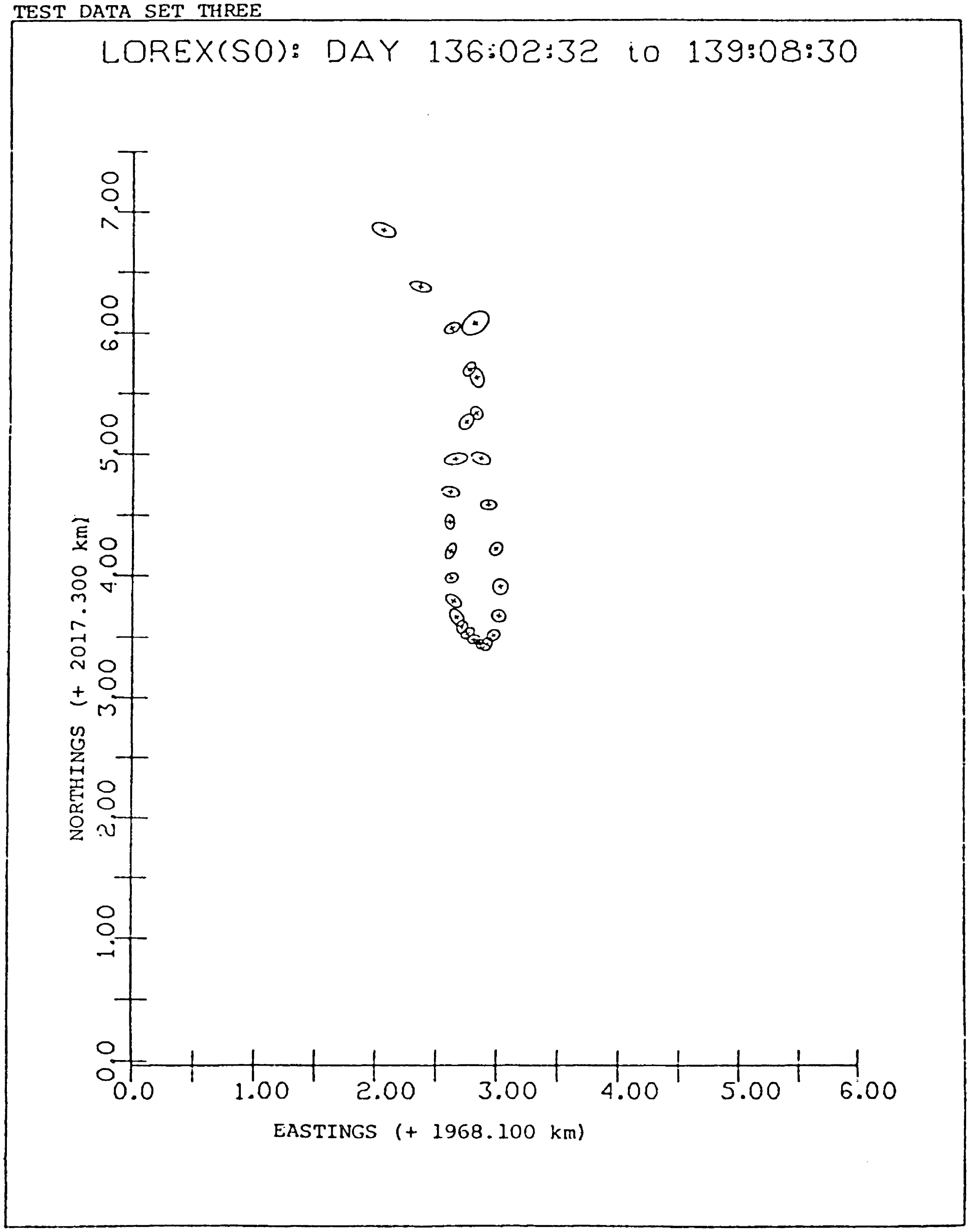


FIGURE 10-13 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET THREE

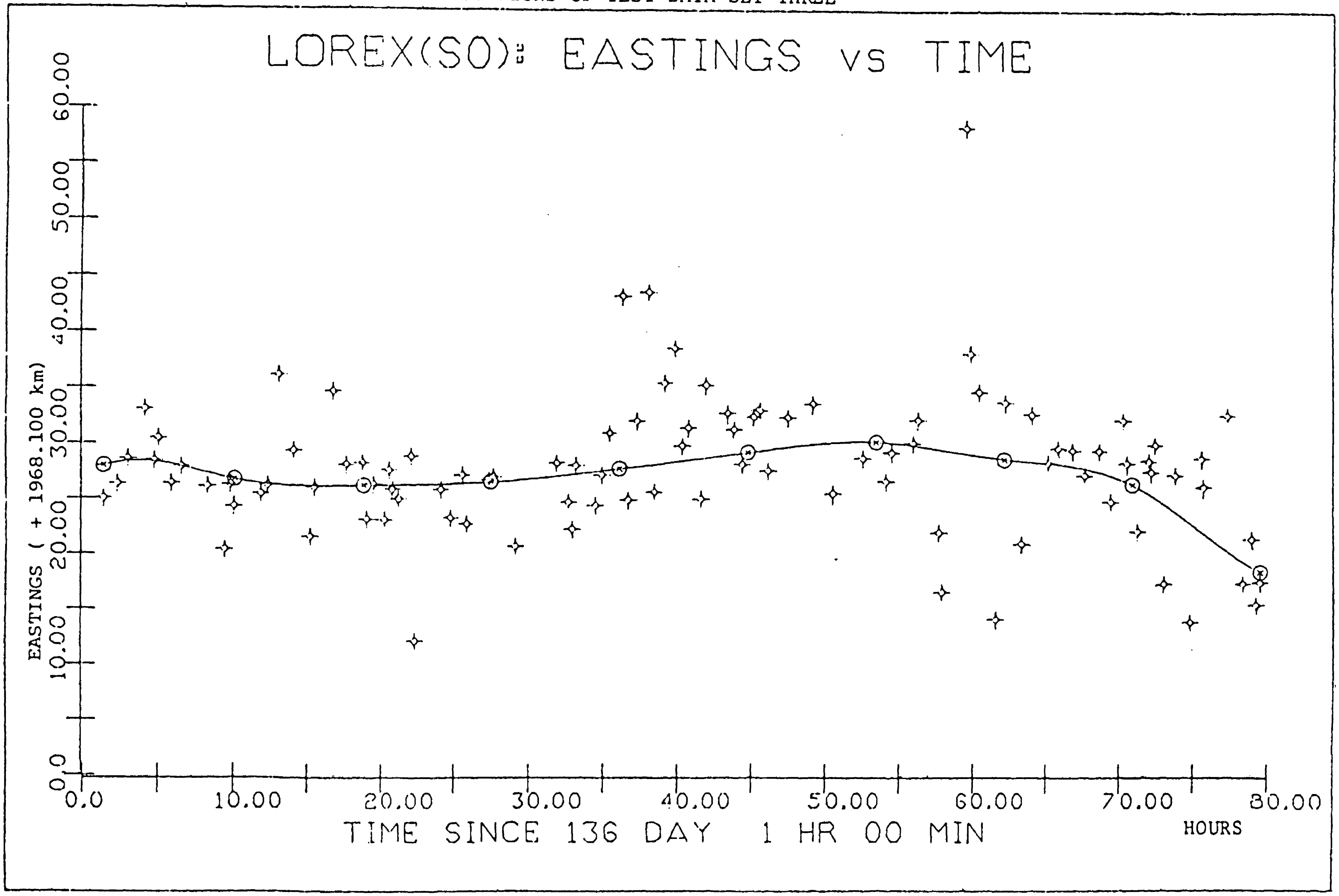
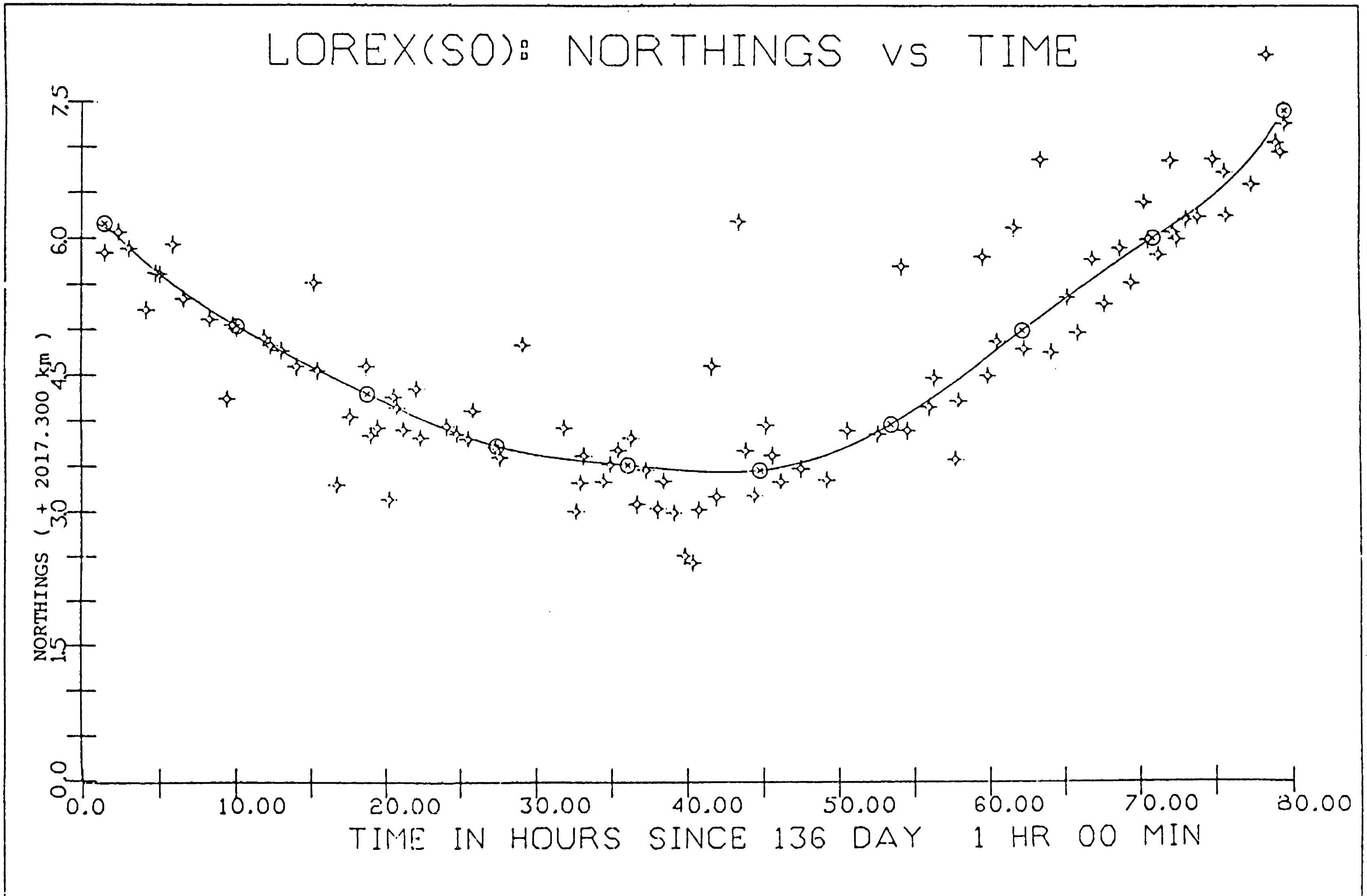


FIGURE 10-14 SMOOTHED AND OBSERVED POSITIONS OF TEST DATA SET THREE



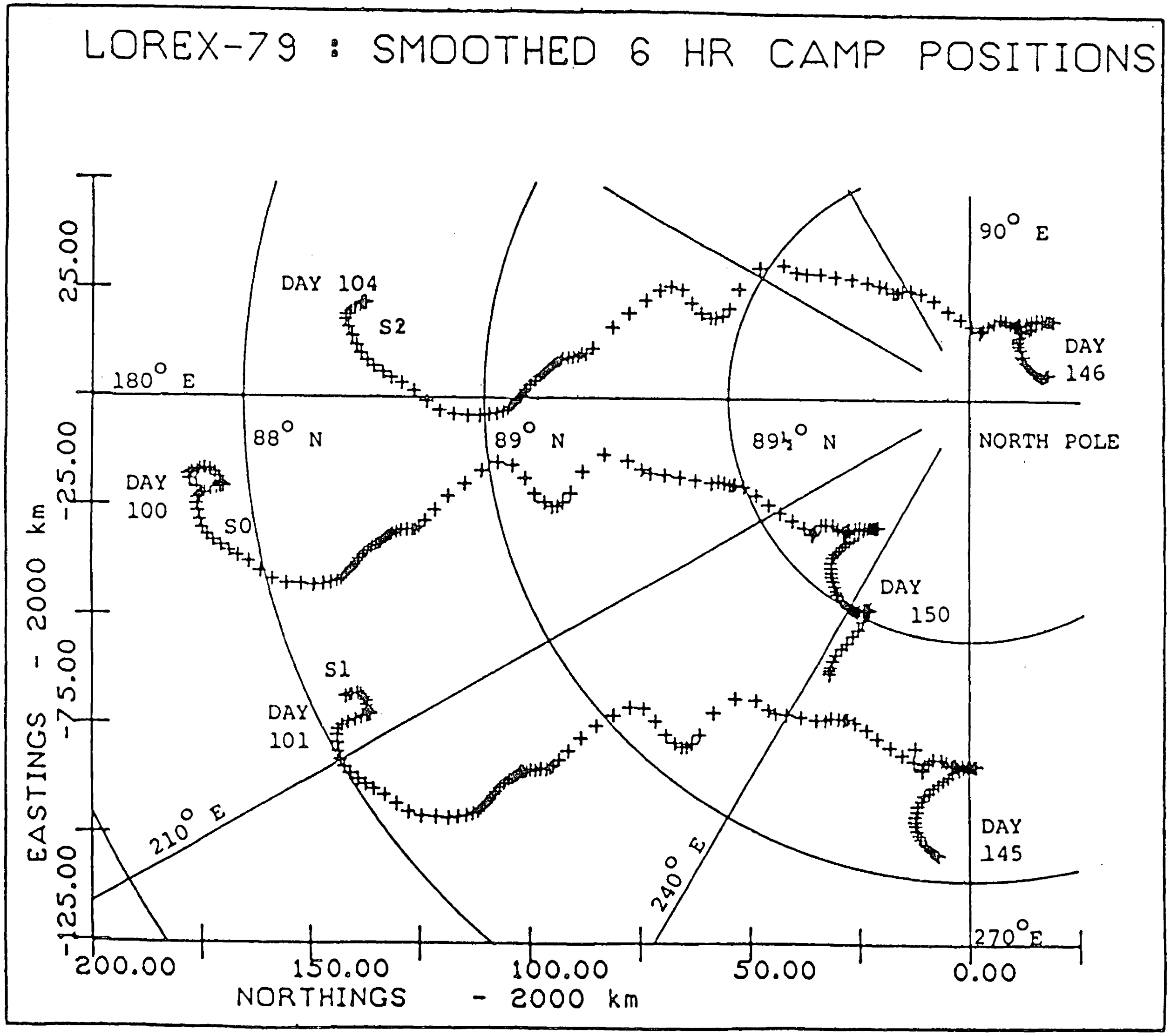
stations. The length of the data series of positions used in the computation of each spline is dependent upon the maximum number of knots that the computer program can handle at any one time. Implications of increasing the upper limit of the number of knots is discussed in Section 10.5.

10.5 PROCESSING OF LOREX-79 DOPPLER POSITIONING DATA

In this section, the complete processing of the positioning data series from the three camps on the LOREX expedition using DSPLIN is described and discussed.

The series of computed real-time position determinations, using the Transit system, for the camps are used as input to the spline algorithm (Figure 10-1) to generate smooth paths for the stations during the period of the expedition (Figure 10-15). The smoothed velocity vectors of the three stations are extracted from the estimated knot vectors of the spline (Figures 10-16 to 10-19). Each of the station's navigation data series is divided into several subsets as there exists a limit to the number of knots the computer program can handle for any given series. Currently, DSPLIN (Appendix VII) has an upper limit of 32 knots. Increasing the maximum number of knots within the program is not desirable as it increases both the random access memory array storage requirements of the program and the execution time in computing the much larger inverses.

FIGURE 10 - 15



LOREX-79 : SMOOTHED 6 HR CAMP POSITIONS
DAY 104
S2
180° E
90° E
DAY 146
88° N
89° N
89½° N
NORTH POLE
DAY 100
S0
DAY 150
S1
DAY 101
DAY 145
210° E
240° E
270° E
25.00
-25.00
-75.00
-125.00
EASTINGS - 2000 km
200.00
150.00
100.00
50.00
0.00
NORTHINGS - 2000 km

LOREX Station Velocity Plots

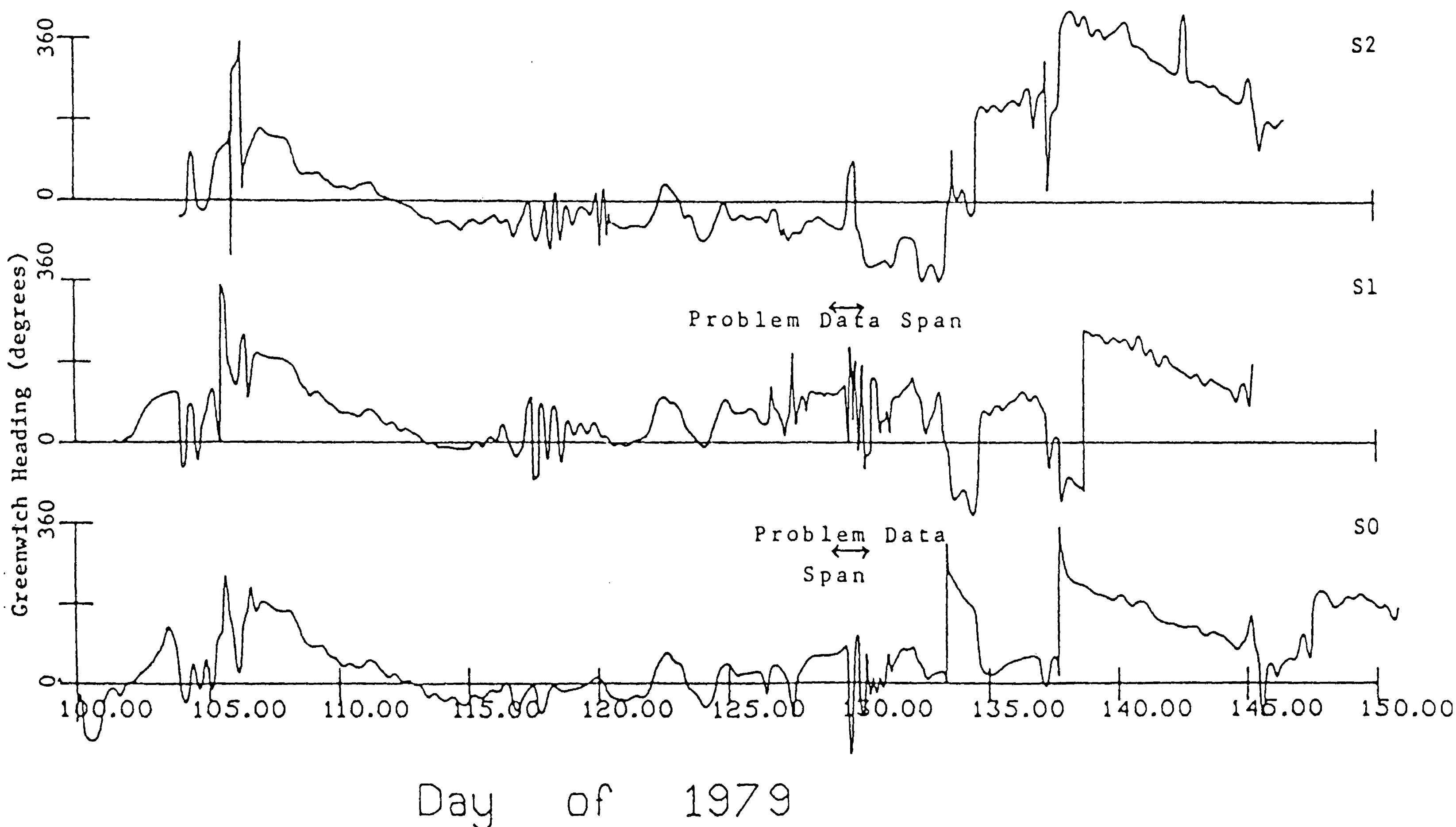


FIGURE 10-16

LOREX Station Velocity Plots

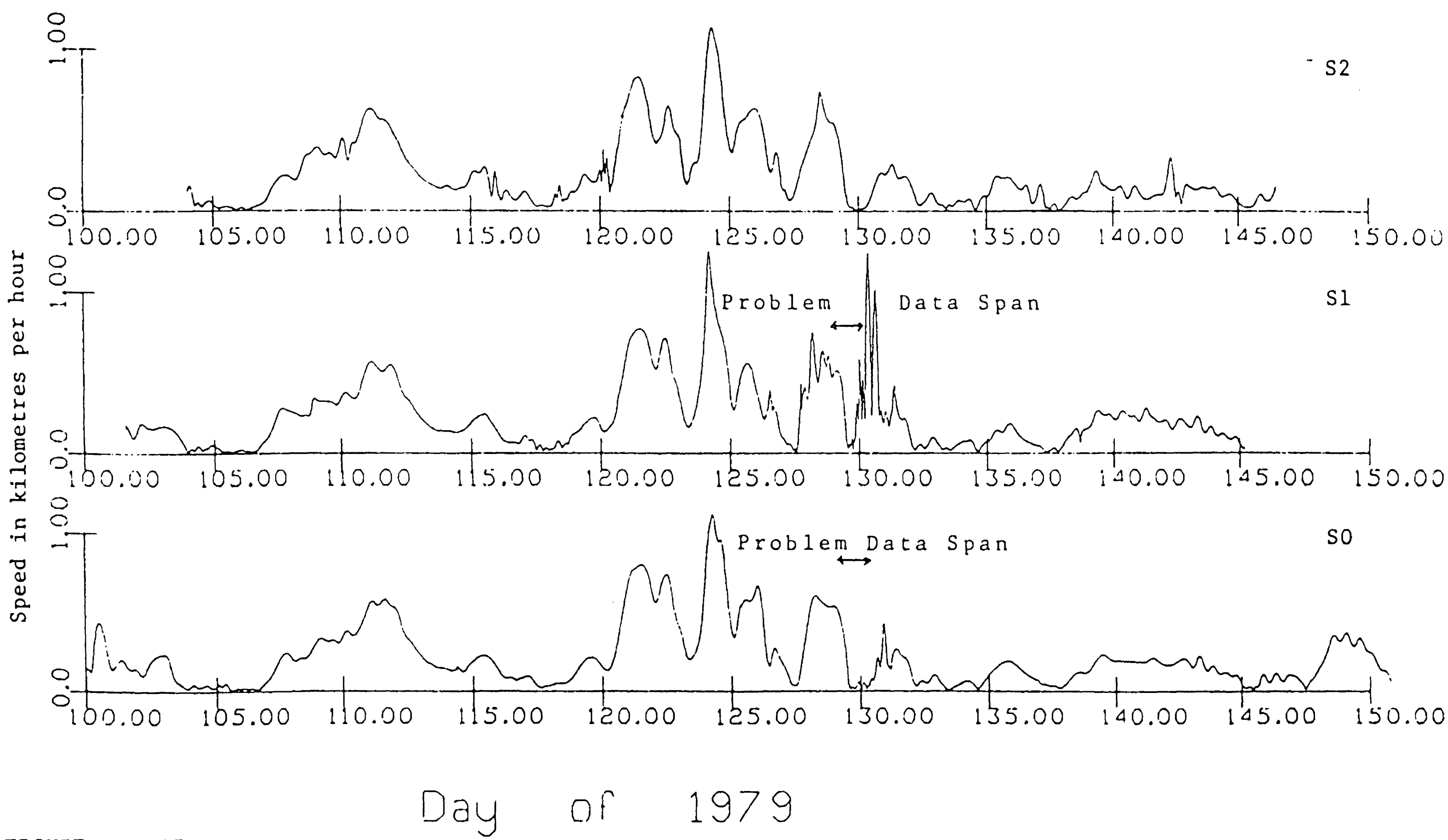
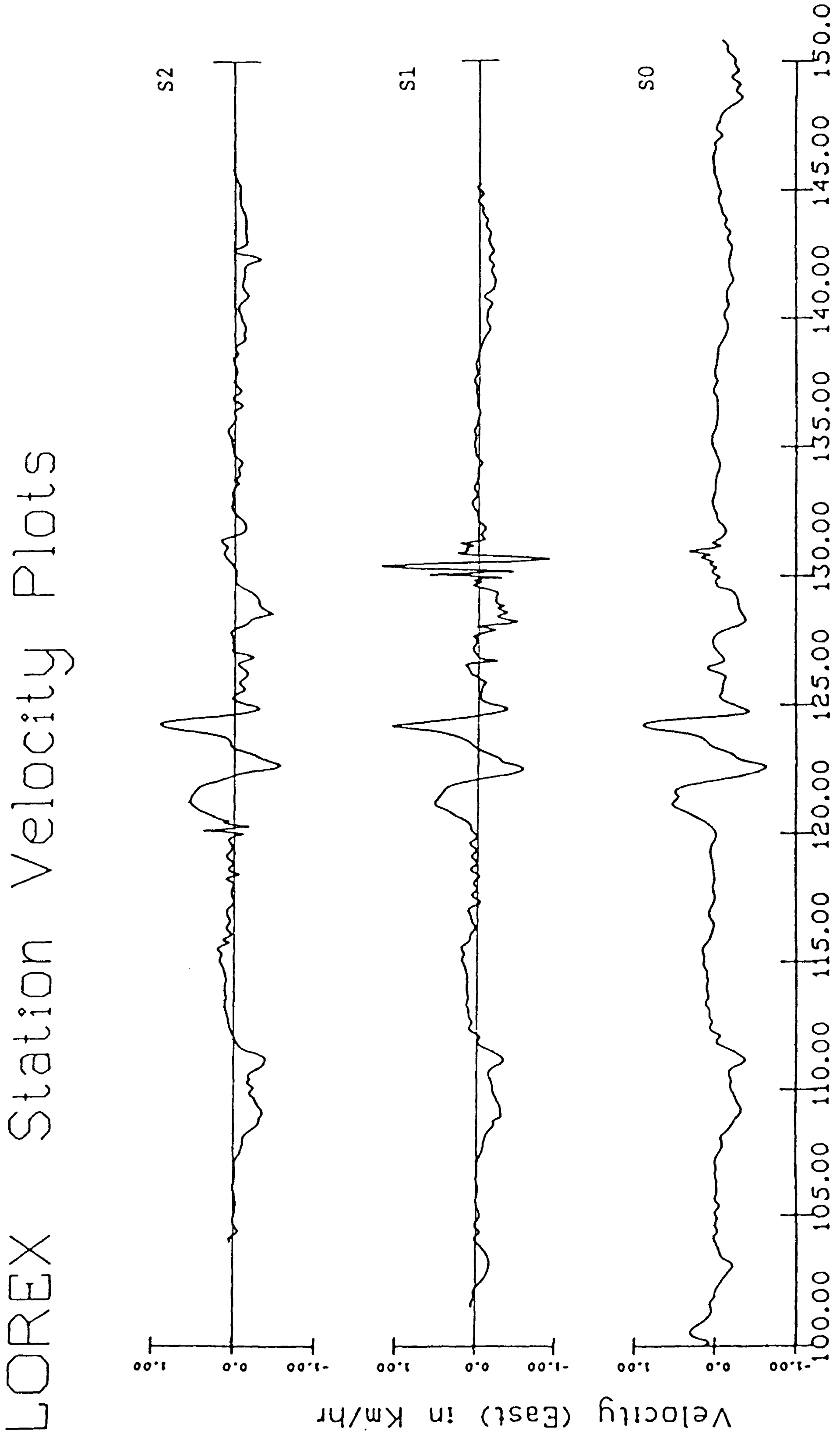


FIGURE 10-17



LOREX Station Velocity Plots
S2
S1
S0
1.00
0.0
-1.00
Velocity (East) in Km/hr
100.00
105.00
110.00
115.00
120.00
125.00
130.00
135.00
140.00
145.00
150.00
Day of 1979

FIGURE 10-18

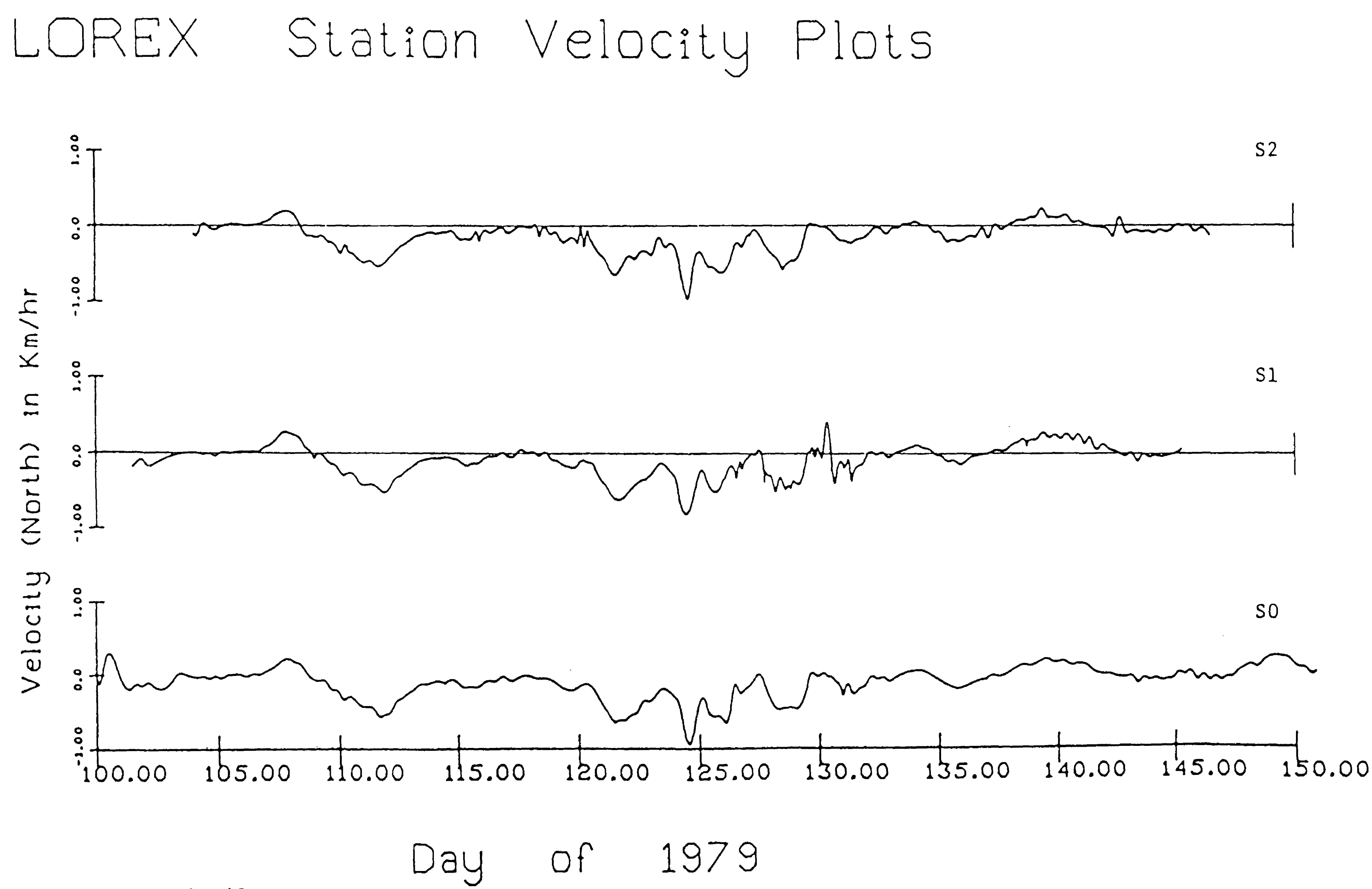

LOREX Station Velocity Plots
S2
S1
S0
1.00
0.0
-1.00
Velocity (North) in Km/hr
100.00
105.00
110.00
115.00
120.00
125.00
130.00
135.00
140.00
145.00
150.00
Day of 1979

FIGURE 10-19

The LOREX Doppler positioning data sets for ice camps SO, S1 and S2 were divided into 9, 8 and 7 subsets respectively.

10.5.1 Problem Data Spans

All subsequent splined subsets, with the exception of two, have a posteriori variance factors which pass the Chi-squared statistical test (Section 7.3.1). The exceptions occur in the LOREX data series from station SO and station S1, and are both around the same time period (i.e. between day 129 to day 131). This phemonenon is absent in the data series from station S2, possibly due to the much lower sampling period of position fixes at station S2.

One of the subsets, as identified above, a data span in the data series from station SO, was subjected to a number of tests in an effort to determine the reason for the failure of the spline model (computed from the data contained in the subset of the data series) to pass the Chi-squared test. The following were investigated:

1. Error in the value of the minimum semi-minor axis of the error ellipses adopted. The selected value of 35 metres (see Section 10.3.1) was gradually increased and the subset model passed the test when a value of 80 metres was reached. The value of 80 metres as the minimum value for the semi-minor axis of the error ellipses greatly reduces the assigned accuracy of all position determinations; including good position fixes.

2. Error in either the value of the cutoff angle for the elevation (i.e. 88 degrees - Table 10-1) of the satellite in the treatment of the position fix as that of a line fix (as defined in Section 10.3.2) or the assigned value of the semi-major axis of the elongated error ellipse constituting a line fix. Results show that changes in the value of the cutoff elevation angle (from 88 to 85 degrees) or the assigned cross track error (increased from 3000 m to 5000 m) did not make any significant change in the a posteriori variance factor.

From the results above, the following are the conclusions drawn about the inability of the a posteriori variance factor of the two data subsets to pass the Chi-squared statistical test.

Firstly, the sampling period, the average time between position fixes, may be inadequate to define the motion of the ice platform during periods of rapid ice movements. Secondly, the fact that here, the effect of velocity-caused errors in the computed positions are modelled simply by expanding the a priori covariance matrix. This may have two defects: inadequacy of the error models to expand the covariance matrices, and the problem that the velocity-caused errors (which are essentially position biases) are modelled as a random effect. The observation period for each satellite pass is about 18 minutes. During this period, any variation in speed and direction of the motion of the receiver will affect the observed Doppler counts and the resulting position computations. The effect of short term variations in speed and direction on the position determinations have not been investigated and can be safely neglected because changes in the velocity vector of the receivers on LOREX, due to the inertia of the ice sheet, would be negligible over the short 18 minute interval.

Predicted speeds and directions, constant over a certain period of time, were used in computing the position fixes. Figure 10-20, which compares the real-time predicted velocity vectors with the final smoothed values from DSPLIN, highlights the errors that can exist in the estimated real-time velocity vectors used in the fix computation. A reduction of these errors would have been possible through a pre-processing algorithm which produces better velocity vector estimates (as illustrated in Figure 10-1 with the spline being used as the pre-processing algorithm) for the recomputation of the position fixes.

The final smoothed path of the three ice stations are portrayed in Figure 10-15. Figures 10-16 to 10-19 give the smoothed velocity vectors of the three ice stations for the duration of the expedition. The differences between the smoothed positions, velocities and directions produced by the spline algorithm and those obtained using GEODOP (see Section 2.2) are given in the following section.

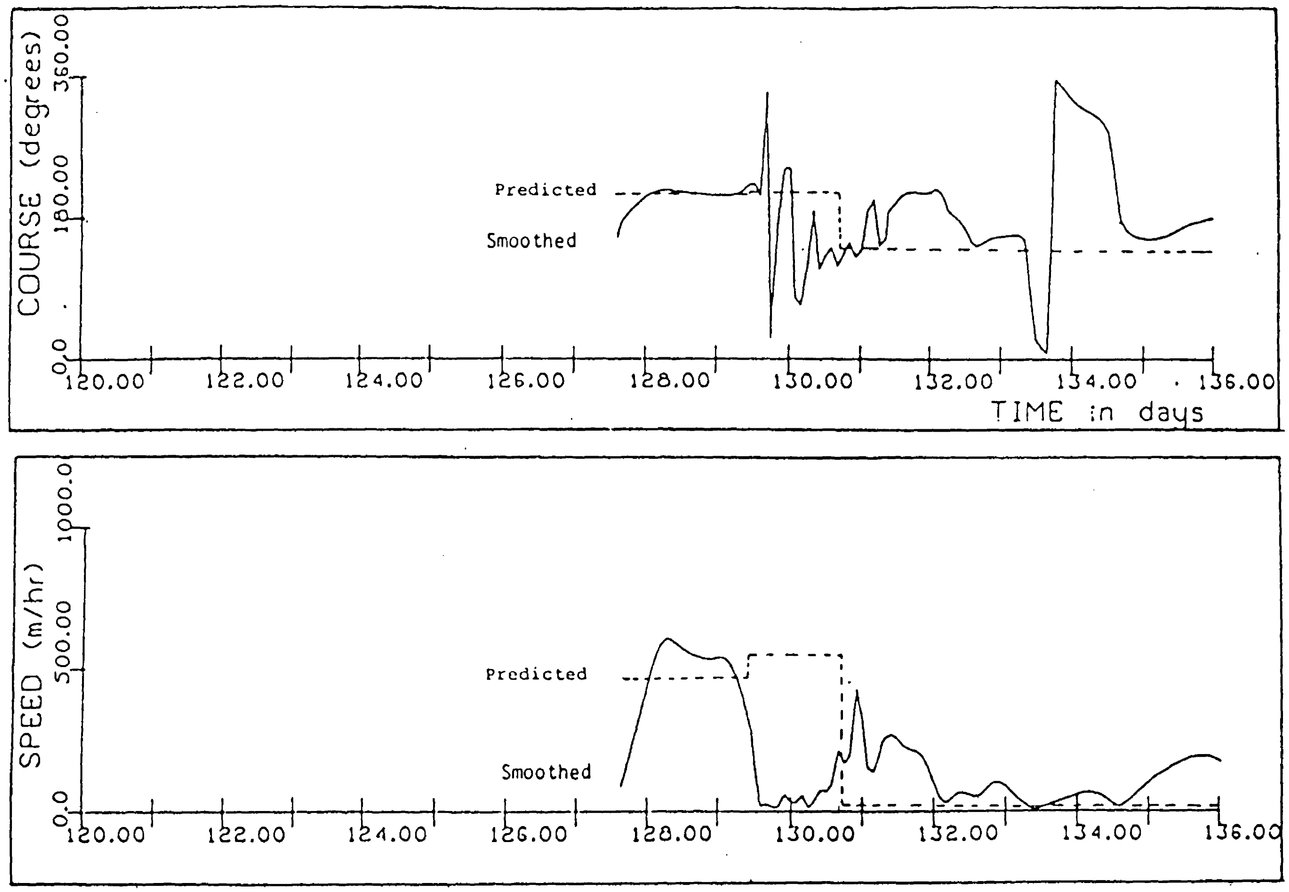


FIGURE 10-20 The real-time predicted and smoothed velocity vectors for station S0.

10.6 COMPARISON WITH OTHER TECHNIQUES

In this section, the six-hour smoothed positions and velocities for the three ice camps produced by DSPLIN, SMOBS and GEODOP are compared. A brief description of SMOBS and GEODOP can be found in Section 2.2. The procedure followed for LOREX data processing by DSPLIN is given in Figure 10-1.

The position and velocity information produced by GEODOP, computed from better orbital and environmental models and recomputed position fixes based on precise satellite orbits in the multi-station three-dimensional adjustment mode (see Section 2.2), are taken as the reference standard for the comparisons. The comparsion between DSPLIN and GEODOP was made to determine the following:

1. Accuracy of the recovered positions and velocities.
2. Consistency of recovered positions and velocities with time.
3. Consistency of estimated precision of position and velocity estimates with accuracy.

Differences between SMOBS and GEODOP are also evaluated and compared with DSPLIN versus GEODOP differences. The following are computed in the comparison between the technique and the adopted reference (GEODOP).

1. Position deviations (i.e. Δr_i).

$$\Delta r_i := [(\phi_i^R - \phi_i)^2 + (\lambda_i^R - \lambda_i)^2]^{1/2}$$

– (10-1)

2. Velocity deviations (i.e. Δv_i).

$$\Delta v_i := [(v_{Ni}^R - v_{Ni})^2 + (v_{Ei}^R - v_{Ei})^2]^{1/2}$$

– (10-2)

where ϕ, λ := geodetic latitude and longitude E,
v_N, v_E := velocity north and velocity east,
i := data point number,
and R := reference standard.

3. Standard deviations of position ($\sigma_{\Delta r_i}$) and velocity($\sigma_{\Delta v_i}$) differences computed using the covariance law (Section 8.1), i.e.

$$\sigma^2_{\Delta r_i} := [\Delta\phi_i^2(\sigma^2_{\phi i} + \sigma^{R2}_{\phi i}) + 2\Delta\phi_i\Delta\lambda_i(\sigma_{\phi_i\lambda_i} + \sigma^R_{\phi_i\lambda_i})$$

$$+ \Delta\lambda_i^2(\sigma^2_{\lambda_i} + \sigma^{R2}_{\lambda_i})]/\Delta r_i^2$$

$$\sigma^2_{\Delta v_i} := [\Delta v^2_{N_i}(\sigma^2_{vN_i} + \sigma^{R2}_{vN_i}) + 2\Delta_{vE_i}v_{N_i}(\sigma_{vNE_i} + \sigma^R_{vNE_i})$$

$$+ \Delta v^2_{E_i}(\sigma^2_{vE_i} + \sigma^{R2}_{vE_i})]/\Delta v_i^2$$

– (10-3)

where $\Delta\phi$, $\Delta\lambda$:= differences in latitude and longitude,
σ_ϕ, σ_λ := std. deviations of position coordinates,
$\sigma_{\phi\lambda}$:= covariance between coordinates,

Δ_{vN}, Δ_{vE} := differences in velocity north and east,
σ_{vN}, σ_{vE} := std. deviations of velocity north and east,
σ_{vNE} := covariance between velocity components.

4. Standardised deviations of positions and velocities.

$$\tilde{\Delta r}_i := \Delta r_i / \sigma_{\Delta r_i}$$

$$\tilde{\Delta v}_i := \Delta v_i / \sigma_{\Delta v_i} \qquad - (10\text{-}4)$$

Ideally, all the above parameters (defined by equations 10-1 to 10-4) should be computed and plotted against time to evaluate their consistency in time. However, as shown in Table 10-3, much of the required data is absent. This is partly due to the technique itself (e.g. SMOBS does not have individual precision estimates for mean positions), and partly by the omission of information in the program outputs (e.g. the correlation coefficients in GEODOP positions are not printed).

The following comparisons were those that could be readily done with the available information.

1. Computation of absolute position and velocity differences.

   These differences are computed in both comparisons, i.e. DSPLIN versus GEODOP and SMOBS versus GEODOP. The set of differences (or deviations) are plotted against time (Figures 10-21 and 10-22). The cumulative distributions, obtained from reordering the position and velocity deviations (equations 10.1 and 10.2), are also plotted (Figure 10-23).

TABLE 10-3

SUMMARY OF POSITION AND VELOCITY INFORMATION PRODUCED BY GEODOP, SMOBS AND DSPLIN

I) Position Estimates

Program Name	Latitude or Northing	Longitude or Easting	Standard Deviations and covariances		
			σ_1	σ_2	σ_{12}
GEODOP	Both	Both	σ_{lat}	σ_{long}	No
SMOBS	Northing	Easting	No	No	No
DSPLIN	Northing	Easting	σ_N	σ_E	σ_{NE}

II) Velocity Estimates

Program Name	Speed	Direction	Standard Deviations and covariances		
			σ_1	σ_2	σ_{12}
GEODOP	Yes	Yes	No	No	No
SMOBS	Yes	Yes	σ_{ve}	σ_{vn}	No
DSPLIN	Yes	Yes	σ_{ve}	σ_{vn}	σ_{ven}

where ve - Velocity East
and vn - Velocity North

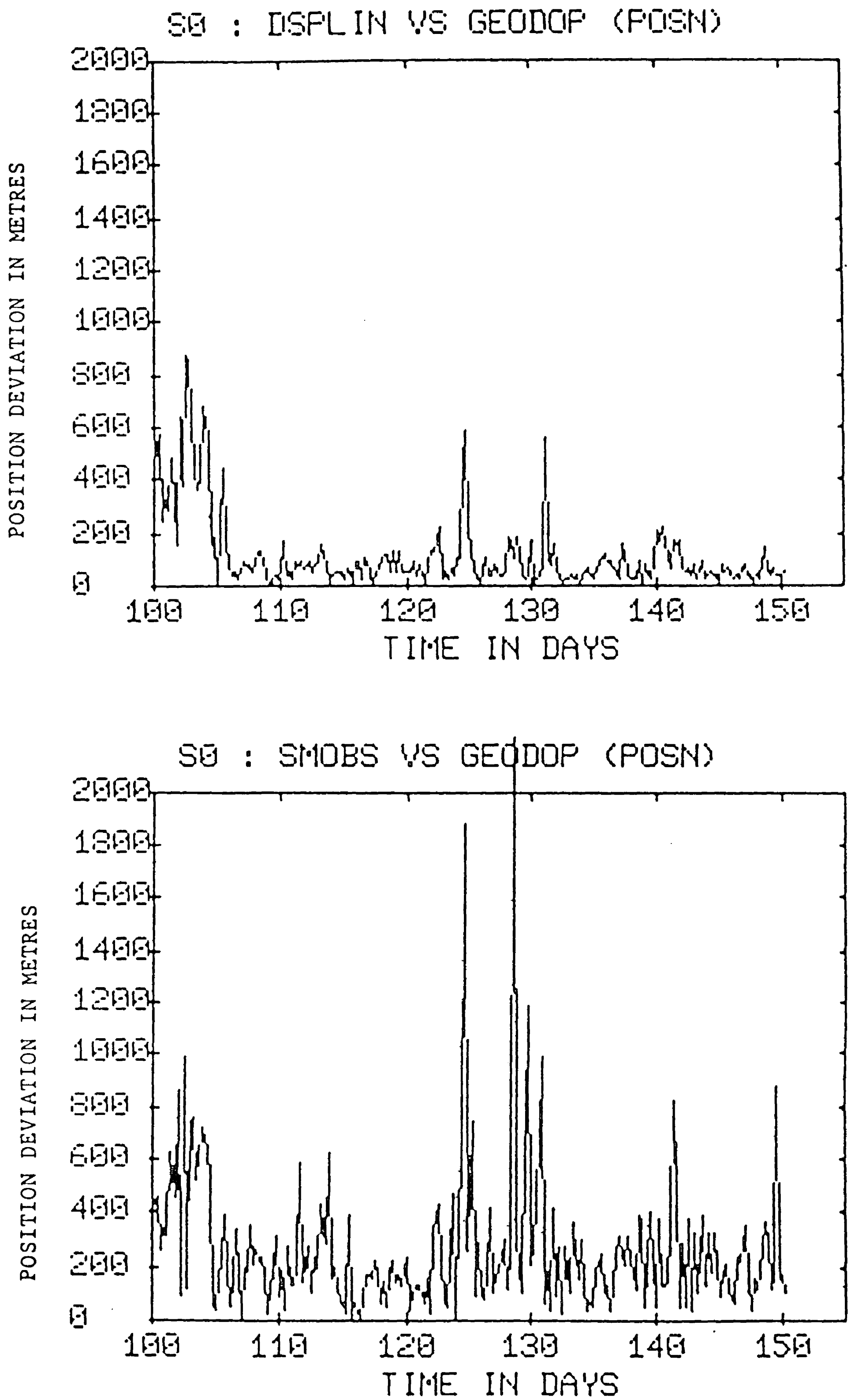
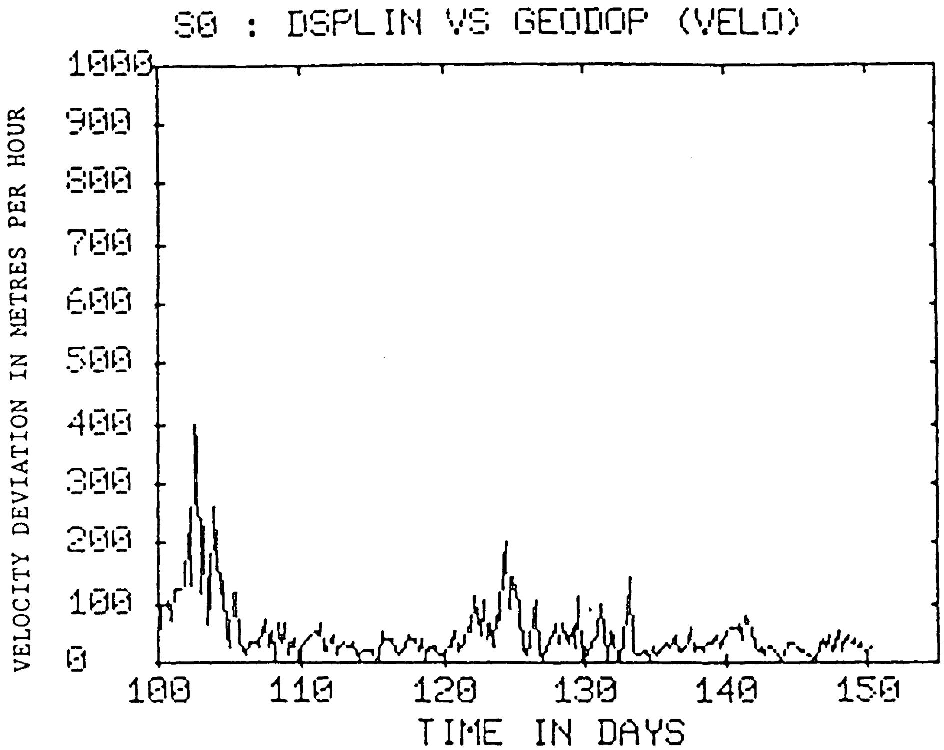


FIGURE 10-21 Time Series of Position Differences



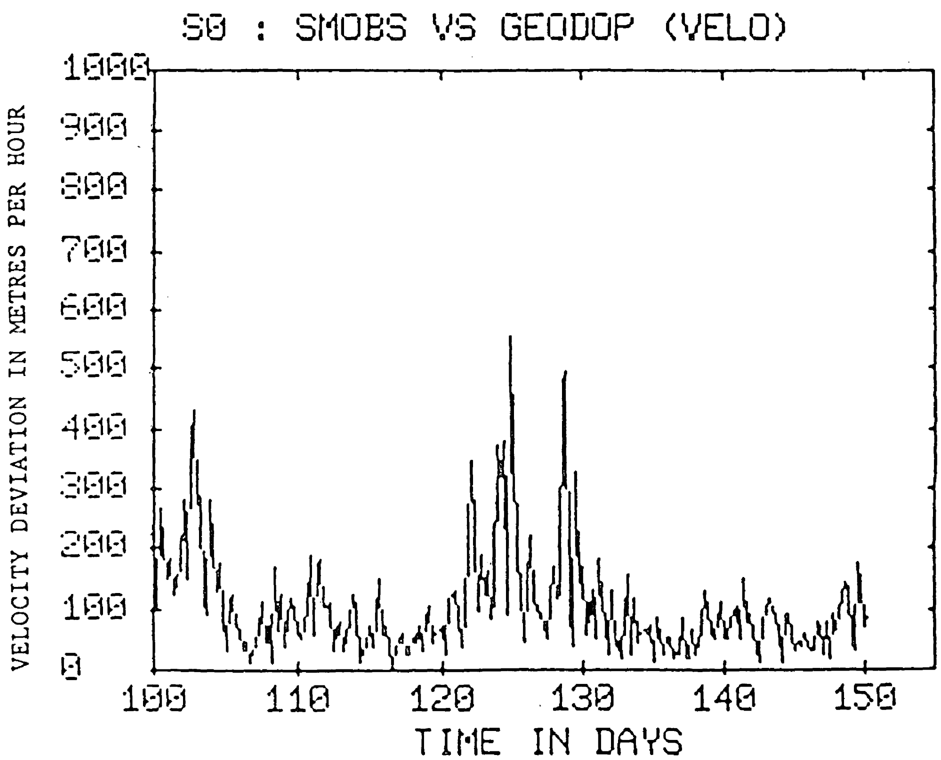
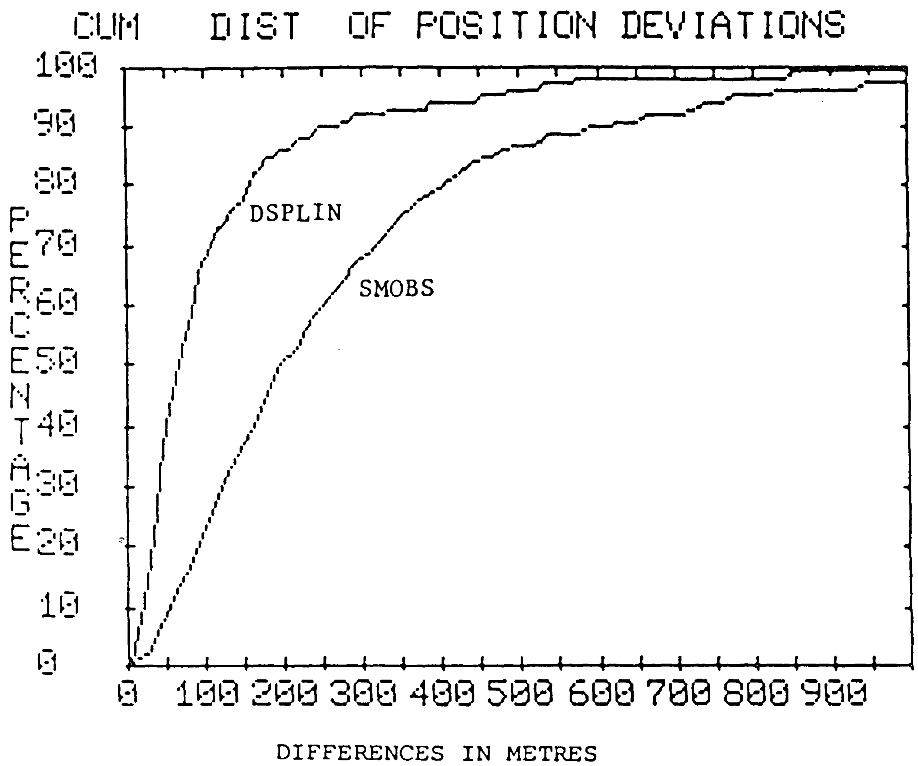


FIGURE 10-22 Time Series of Velocity Differences



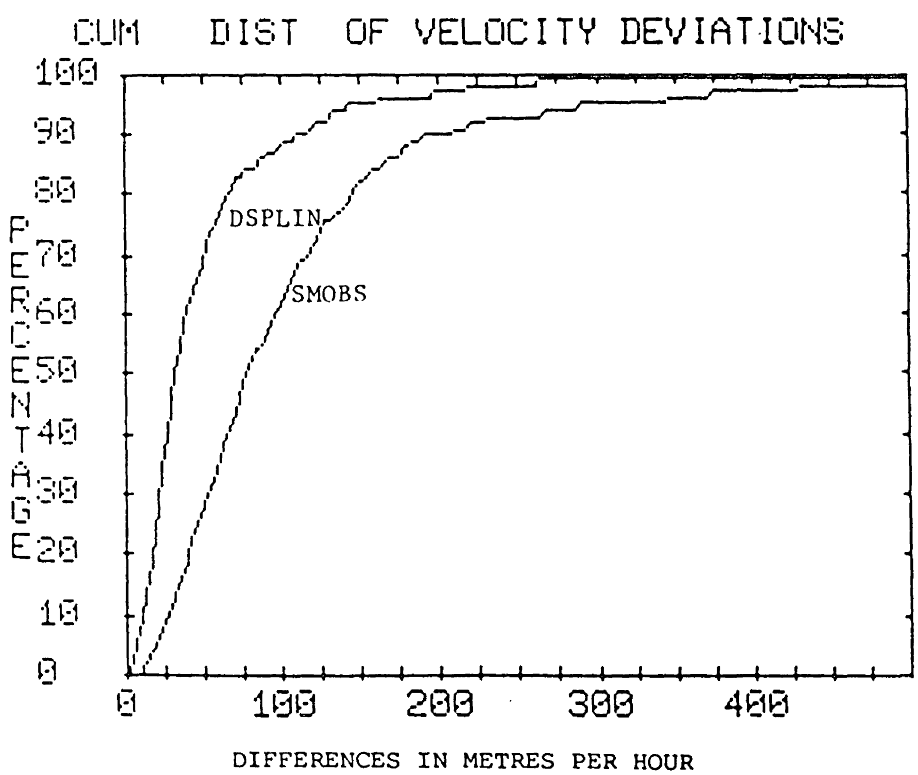


FIGURE 10-23 Cumulative Distribution of Position and Velocity Deviations

2. Computation of standardised position deviations.

This is only possible in the comparison between DSPLIN and GEODOP, and even so, the assumption that the coordinates produced by GEODOP are uncorrelated (Table 10-3) had to be made (Figure 10-24).

520 positions and velocities from S0, S1 and S2, common to the smoothed LOREX data series produced by GEODOP, SMOBS and DSPLIN, were chosen for the comparisons. The resulting root mean square of the position and velocity differences are 194 metres and 70 metres per hour, respectively from the DSPLIN versus GEODOP comparison, and 437 metres and 144 metres per hour, respectively from the SMOBS versus GEODOP comparison. Hence DSPLIN position and velocity estimates are 56 and 47 percent better than SMOBS. The time series plots of position and velocity deviations (Figure 10-21 and 10-22) show the improvement for station S0 graphically. This improvement in position and velocity estimation is again demonstrated by the cumulative distribution plots of the position and velocity deviations (Figure 10-23). The coincidence in the peaks of position and velocity deviations in Figure 10-21 and 10-22 can be explained by the fact that both DSPLIN and SMOBS use the same computed position data series and hence were influenced by the same "bad" data points. The effect of an erroneous data point however, is more pronounced with SMOBS. As demonstrated in Figure 10-21 and 10-22, smoothed position errors propagate into smoothed

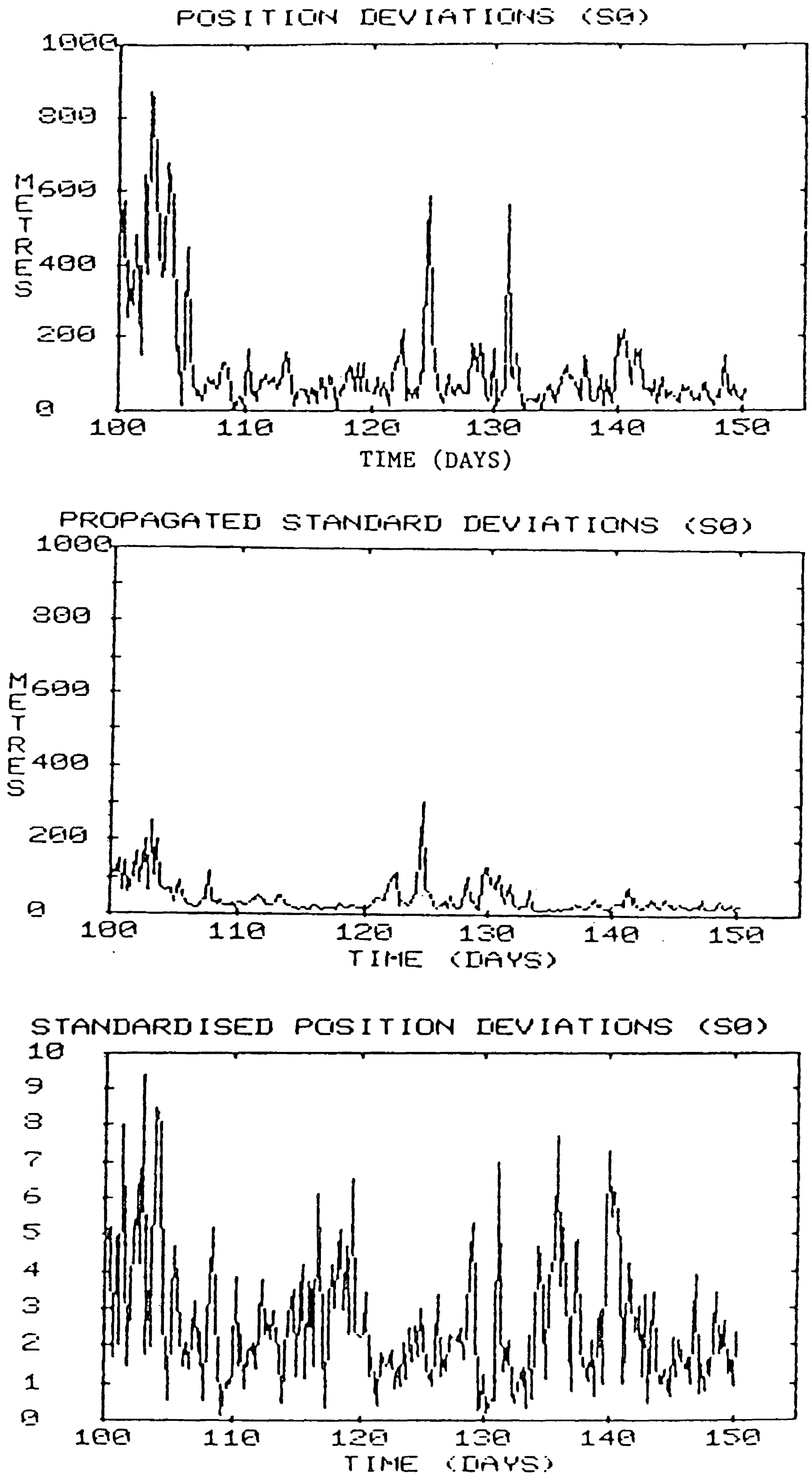


FIGURE 10-24 Time Series of Position Deviations, Propagated Standard Deviations and Standardised Position Deviations.

velocity errors. A larger discrepancy between positions produced by SMOBS or DSPLIN against GEODOP is observed to occur between day 100 and day 105. The reason for this discrepancy is not apparent. One plausible cause could be oscillator instabilities in the satellite receivers as they were all cold started upon arrival on LOREX.

As to the assessment of the consistency of the precision estimates, the ratio of the deviation to its propagated standard deviation is ideally equal to one. The plots of the time series of position deviations (equation 10-1), propagated standard deviations (equation 10-3) and standardised position deviations (equation 10-4) for station SO are given in Figure 10-24. Examination of Figure 10-24 reveals that the precision estimates (accepting GEODOP as being true) are not consistent in time with the accuracy. Generally, the position deviations are correlated with the propagated standard deviations. There are several places where they are different. The larger discrepancies between day 100 to day 105 are not adequately accounted for by the standard deviations. Consistency in the precision estimates occurs only between day 120 to day 130. The motion of the station during this period is characterized by high velocities and larger velocity changes (Figure 10-17). The highest absolute velocities reached by the three stations are recorded in this time period. Maximum velocity east (Figure 10-18) and velocity north (Figure 10-19) of the three stations also occur during this time period.

The inconsistencies are caused by the covariance matrix associated with each position being optimistic. This optimism may exist in either GEODOP or DSPLIN values. The contribution of the neglected correlation between GEODOP smoothed coordinates to the periods of inconsistency cannot be evaluated.

The overall conclusion that can be drawn is that DSPLIN (with GEODOP as reference standard) produces more accurate position and velocity estimates which are consistent with time than SMOBS and that the precision estimates are inconsistent in time with the derived measures of accuracy.

Chapter 11

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, the contributions of this thesis are summarized. Conclusions and recommendations are made on smoothing with the developed spline algorithm, application of the spline algorithm to the LOREX navigation data sets and the treatment of high latitude position fixes using NNSS.

11.1 TWO-DIMENSIONAL LEAST-SQUARES CUBIC SPLINE ALGORITHM

An algorithm to smooth two-dimensional data series has been successfully developed (Chapters 5 and 6) and proven feasible with the LOREX positioning data (Chapters 9 and 10). From the numerous studies done for this thesis, the following are the conclusions and recommendations.

11.1.1 Number and Placements of Knots

The spline algorithm requires a sufficient density of knots to aptly portray the underlying nature of the data series in question. As discussed in Section 9.2, an indication of the adequacy of the number of knots can be arrived at through the evaluation of the root mean square of the residuals and the a posteriori variance factor of the subsequently fitted spline curve (equation 7.8).

This, however, remains only a global indication of knot sufficiency. Knot selection at local intervals can be obtained through the analysis of the time series of standardised residuals of the observed data points (Section 9.3). Where the knot intervals can be linked to some physical evidence pertaining to the data series, the knot selection process is very simple. This is the minimum data period in which the process can be aptly portrayed as being cubic.

The simplistic nature of selecting the desired degree of smoothness within the data series (i.e. through defining the knot intervals) is a unique feature of this technique.

11.1.2 Joining Splines

The aspect of joining splines within a data series has been successfully explored (Section 9.5) and a minimum of two common knot intervals is recommended.

11.1.3 Predicted Data Points

The developed program (Appendix VII) computes, on request, the smoothed data points and first derivatives at any time within the data series. Estimated covariance matrices are derived for these quantities using the covariance law (Section 8.1). For a complete model, i.e.

with a sufficient density of knots, the estimated covariance matrix reflects the accuracy of the predicted data points (Section 9.3). Covariance matrices can be derived for all subsequently computed quantities using the covariance law outlined in Chapter 8.

11.1.4 The DSPLIN Program

A computer program (DSPLIN), based on the spline algorithm developed in this thesis, has been created (Appendix VII). All computations involving the spline algorithm are done with the DSPLIN Program. Various optimising procedures aid in the handling of large matrices and inverses. These, along with a short description of the 64 subroutines in DSPLIN, are given in Appendix VII: Table VII-1. Currently, DSPLIN can handle up to 32 knots and a maximum of 500 data points in a data span. The program requires about 512 kilobytes of memory to run. The programming language used is Fortran IV. All software development was done using the Fortran H Extended Compiler on the IBM 3032 computer at the University of New Brunswick.

11.1.5 Comparison with SMOBS and GEODOP

The DSPLIN program smoothed values are compared with SMOBS's (Section 2.2) using GEODOP as the reference standard (Section 10.6). In the comparisons of position and velocity differences (equations 10-1 and 10-2) of DSPLIN versus

GEODOP and SMOBS versus GEODOP, the root mean square of the differences computed from DSPLIN versus GEODOP are about 50 percent smaller than those of SMOBS versus GEODOP. Figures 10-21 to 10-23 portray the improvement of DSPLIN over SMOBS as time series plots of differences and cumulative distributions of differences. The standardised position differences in the DSPLIN and GEODOP comparison shows that the precision estimates produced by DSPLIN are not consistent in time with the accuracy. There exist however, short periods where they are consistent.

11.2 LOREX NAVIGATION DATA SETS

The following conclusions and recommendations are arrived at from the analysis and processing of the LOREX navigation data sets (Chapter 10). Conclusions drawn are applicable to high latitude navigation using the NNSS system.

The final smoothed positions of the LOREX three ice stations are given in Figure 10-15. Plots of the time series of coordinates, input positions and their formal error ellipses, and the final smoothed positions of the subsets of the three data sets from stations S0, S1 and S2 can be found in Quek [1983]. In the processing of the LOREX navigation data sets, the effects of real-time predicted velocity errors (Section 10.3.2) with high elevation satellites had to be taken into account. Their treatment involved the modification of the a priori covariance matrix of the position fixes (Section 10.3.4).

11.2.1 A Priori Covariance Matrix of Observed Data Points

For a correct interpretation of the results of the Chi-squared test (Section 7.3.1) on the a posteriori variance factor of the computed spline curve, a reliable a priori covariance matrix has to be assigned to the observations (Section 10.3). With the LOREX data sets, extensive modifications to the formal error ellipses are made to account for errors affecting position determinations using the Transit system at high latitudes. These include the modelling of predicted velocity (Section 10.3.2) and assumed height errors (Section 10.3.3), and the accuracy of the satellite broadcast ephemerides (Section 10.3.1). An algorithm for determining the elevation angle of the observed satellite from the formal error ellipses has been developed (Appendix III: equation III-3). The conservative formal error ellipses recorded, primarily result from the small sample size of the Doppler counts used in computing the fixes.

11.2.2 Unbalanced Doppler Counts

A great majority of the computed satellite passes were observed to have Doppler counts that are severely unbalanced about the point of closest approach. As the NNSS system was used in its weakest single pass point positioning mode, the computed positions are displaced in the along track direction of the satellite. Outright rejection of such passes is not being recommended as they do contain some useful information. Unbalanced Doppler counts within a fix computation, as mentioned above, propagate as a bias in the position of the receiver in the along track direction of the satellite. By using known accuracy estimates of the Transit position determinations and modelling all other possible errors (Section 10.3) useful information from these observations can be drawn.

11.2.3 Station Motion

The computation of position determinations with the NNSS system requires the monitoring of the motion of the receiver during the satellite pass. Any errors in the assumed velocity vector reduces the accuracy of the computed position. This degradation is especially severe with high elevation satellites. In order to minimise this phemonenon, an on-site reprocessing of the position determinations using iteratively improved velocity vectors is recommended (Section 10.5.1). The improved velocity vectors can be obtained from, for example, the spline algorithm developed in this thesis (Figure 10-1).

Computed velocity vectors of this nature, however, represent the long term variations in the motion of the receiver. Under short term velocity vector changes, the position fixes should be computed with zero velocity vectors unless the rapid changes in velocity and direction are determined by other external means. The presence of the predicted velocity vector, when in error, will only downgrade the precision of the computed positions.

11.2.4 High Elevation Satellites

From the limited experiments made to study the behaviour of high elevation satellite passes with the spline algorithm (Section 10.3.4), it is concluded that for a non-stationary observer, the position coordinate in the cross satellite track is extremely poor. The current treatment of those fixes as line fixes (defined in Section 10.3.2) has met with some success.

11.3 SUGGESTIONS FOR FUTURE WORK

The following areas are recommended for further investigation:

For the spline algorithm:

1. The evaluation of the performance of the curve fitting routine with less noisy data.
2. The effect of different base functions for the cubic spline (e.g. those given by Spath [1974]: page 58).

3. Extension of the spline algorithm, developed here, to smooth multi-dimensional data series.

   The technique, as described in this thesis, can be easily extended to smooth a multi-dimensional data series. Here, more than two parametric cubic splines are simultaneously solved using the method of least-squares.

For the computer program:

4. The incorporation of a banded inverse routine which operates on a banded normal matrix and produces only a banded a posteriori covariance matrix.

   The processing time in computing the inverses can be reduced by a factor of b^2/n^2; with n being the order of the matrix and b being the bandwidth (Appendix VII).

5. Additions to the program to allow the processing of a full data series without any manual intervention.

   Currently, the subsets of a data series are individually splined and the technique of splicing splines (described in Section 9.5) used to achieve continuity. The proposed modifications here involve automating data management of the data series, which currently is done manually through interactive files.

For the LOREX navigation data series:

6. Further investigation into the effect of velocity-caused errors on position fixes with high elevation satellites.
7. Reprocessing of the position fixes, using velocities derived from the spline-smoothed trajectories, and reprocessing using the spline algorithm on these improved position fixes.
8. Real-time implementation of the spline algorithm.

   The real-time application of the spline algorithm is described in Section 10.3. The DSPLIN program with minor alterations, easily done due to its modular structure, and a new driver routine, can be adapted to operate in real-time mode. It has to be used together with a satellite fix computation routine and in the pre-processing mode supply better velocity estimates for a recomputation of the position fixes.

REFERENCES

Ahlberg, Nilson and Walsh (1977) The Theory of Splines and Their Applications, Mathematics in Science and Engineering, Vol 38, Academic Press, New York.

Akima, H. (1977) A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, Journal of the A.C.M., Vol 17, No.4, Oct 1977, pages 589-602.

Boor, C. de (1978) A Practical Guide to Splines, Applied Mathematical Sciences, Vol 27, Springer-Verlag, New York.

Chung, W.L. (1980) Automatic Curve Fitting Using An Adaptive Local Algorithm, A.C.M. Trans. on Mathematical Software, Vol 6, No.1, March 1980, pages 45-57.

Danby, J.M.A. (1962) Fundamentals of Celestial Mechanics, Brett-Macmillan Ltd, Galt, Ontario.

Ellis, T.M.R. and McLain, D.H. (1977) A New Method of Cubic Curve Fitting Using Local Data, A.C.M. Trans. on Mathematical Software, Vol 3, No.2, June 1977, pages 175-178.

Ferguson, J. and Staley, P.A. (1973) Least Squares Piecewise Cubic Curve Fitting, Communications of the A.C.M., Vol 16, No.6, June 1973, pages 380-382.

Forsythe, G.E., Malcolm, M.A. and Moler, C.B. (1977) Computer Methods For Mathematical Computations, Series in Automatic Computation, Prentice-Hall, Inc., New Jersey.

Goldstein, H. (1950) Classical Mechanics, Addison-Wesley Publishing Co., Inc., London, England.

Gujar, U.G. (1981) Computer Plotting, Users Guide, Vol. 11 Computer Centre, University of New Brunswick, N.B., Canada.

Heisenberg, W. (1930) The Physical Principles of the Quantum Theory, The University of Chicago Science Series, University of of Chicago Press, Illinois.

Heisenberg, W. (1958) Physics and Philosophy, Harper Torchbooks, Harper and Row Publishers, New York.

Hoar, G.J. (1982) Satellite Surveying, Magnavox Advanced Products and Systems Company, California, U.S.A.

IBM (1974) IBM System/360 and System/370 FORTRAN IV Language, IBM Corporation, Programmimg Publications, 1271 Avenue of the Americas, New York, New York, U.S.A.

Ichida, K. and Kiyono, T. (1977) Curve Fitting by a One Pass Method With a Piecewise Cubic Polynominal, A.C.M. Trans. on Mathematical Software, Vol 3, No.2, June 1977, pages 164-174.

IMSL (1975) Library 1 : Reference Manual, Vol. I and II, sixth floor, GNB Building, 7500 Bellaire Boulevard, Houston, Texas 77036.

Landau, L.D. and Lifshitz, E.M. (1976) Mechanics, Third Edition, Course in Theoretical Physics, Vol 1, Pergamon Press. Oxford.

Lawson, C.L. and Hanson, R.J (1974) Solving Least Squares Problem, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

McCuskey, S.W. (1962) Introduction To Celestial Mechanics, Addison-Wesley Publishing Company, Inc., London.

Mikhail, E.M. (1970) Parameter Constraints on Least Squares, Photogrammetry Engineering, Dec., pages 1277-1291.

Mikhail, E.M. (1976) Observations and Least Squares, IEP Series in Civil Engineering, Harper and Row Publishers, New York.

O'Neill, B. (1966) Elementary Differential Geometry, Academic Press, New York.

Poirer, D.J. (1973) Piecewise Regression Using Cubic Splines Journal of the American Statistical Ass., Vol 68, No. 343, Sept 1973, pages 515-523.

Pollard, H. (1966) Mathematical Introduction to Celestial Mechanics, Prentice Hall, Inc., New Jersey.

Popelar, J., Kouba, J. and Wells, D. (1981) LOREX-79 Satellite Positioning, Earth Physics Branch, Department of Energy, Mines and Resources, Ottawa.

Prenter, P.M. (1975) Splines and Variational Methods, Pure and Applied Mathematics, John Wiley and Sons, New York.

Quek, S.H. (1983) LOREX-79 - Ice Stations Positions and Velocities using Two-Dimensional Cubic Spline Approximation, Technical Report - in prep , Department of Surveying Engineering, U. of New Brunswick, New Brunswick.

Shumaker, L.L. (1981) Spline Functions : Basic Theory, Pure and Applied Mathematics, John Wiley and Sons, New York.

Spath, H. (1974) Spline Algorithms For Curves And Surfaces, Utilitas Mathematics Publishing Inc., Winnipeg.

Stansell, T.A. (1978) The Transit Navigation Satellite System, Magnavox Government and Industrial Electronics Company, California, U.S.A.

Steeves, P.A. (1974) Least Squares Adjustment of Control on The Mapping Plane M. Sc. E Thesis, Department of Surveying Engineering, U. of New Brunswick, New Brunswick.

USNO (1979) Quarterly Keplerian Parameters, Navy navigation Satellite System, April 1979.

Vanicek, P. (1973) Gravimetric Satellite Geodesy, Lecture Notes No. 32, Department of Surveying Engineering, U. of New Brunswick, New Brunswick.

Vanicek, P. and Krakiwsky, E. (1982) Geodesy : The Concepts, North-Holland Publishing Co., Amsterdam, Holland.

Vondrak, J. (1969) A Contribution To The Problem of Smoothing Observational Data, Bull. of the Astronomical Institute of Czechoslovakia, Vol 20, No.6, page 349-355.

Wells, D.E. (1974) Doppler Satellite Control, Technical Report No. 29 , Department of Surveying Engineering, U. of New Brunswick.

Wells, D.E. and Grant, S.T. (1981) An Adaptable Integrated Navigation System : BIONAV, Presented at the Canadian Petroleum Association Colloquium III, Banff, Alberta, Oct 1981.

Wells, D.E. and Popelar, J. (1979) Preliminary Satellite Navigation Results on LOREX 79, (unpublished).

Wells, D.E. and Krakiwsky, E. (1971) The Method of Least Squares, Lecture Notes No. 18, Department of Surveying Engineering, U. of New Brunswick.

Westlake, J.R. (1968) Numerical Matrix Inversion and Solution of Linear Equations, John Wiley and Sons, Inc., New York.

Appendix I

LEAST-SQUARES ADJUSTMENT VIA PARAMETER ELIMINATION

In this appendix, the least-squares adjustment with constraints through the elimination of constraint equations is presented. The working equations are derived and the compatibility of implementing these equations in the current algorithm discussed.

The basic models, primary and secondary, were given in Chapter 7. Reference should be made to that chapter for clarification of model definitions and variable names used in this appendix.

Constraint equations relate only the functional relationship between parameters (or unknowns). The functional dependence of the parameters leads to the fact that there are as many dependent parameters as there are constraint equations. In the spline algorithm developed in Chapter 5, the constraint equations functionally relate the first derivative (or slope) at the current knot with the first derivatives and positions of fore and aft knots (equations 5.4).

Using the method of parameter elimination [Mikhail 1976: page 217], the constraint equations are used to functionally solve for as many parameters as there are constraints.

With Taylor's expansion and Lagrange multipliers using the constrained minima technique, the following sets of equations can be obtained (as in Section 7.3).

$$\begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} + \begin{bmatrix} \underline{A}_1 \\ \underline{A}_2 \end{bmatrix} \underline{\delta} + \begin{bmatrix} \underline{B} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \underline{v} = \underline{0} \qquad \text{- (I-1)}$$

where $\underline{w}_1 := \partial \underline{f}_1(\underline{x}^o, \underline{l}^o)$

$\underline{w}_2 := \partial \underline{f}_2(\underline{x}^o)$

$\underline{A}_1 := \left.\dfrac{\partial \underline{f}_1}{\partial \underline{x}}\right|_{\underline{x}^o, \underline{l}^o}$, $\underline{A}_2 := \left.\dfrac{\partial \underline{f}_2}{\partial \underline{x}}\right|_{\underline{x}^o}$

$\underline{B} := \left.\dfrac{\partial \underline{f}_1}{\partial \underline{l}}\right|_{x^o, l^o}$

and with $\underline{P} := \sigma_o \underline{\Sigma}_{\underline{l}}^{-1}$

($\underline{\Sigma}_{\underline{l}}$ is the covariance matrix of observables, $\underline{l}$)

Now, let the subvector $\underline{\delta}_2$ contain the parameters to be eliminated

i.e. $\underline{\delta} := [\underline{\delta}_1 \mid \underline{\delta}_2]$

Similarly, by partitioning the design matrices,

i.e. $\underline{A}_1 := [\underline{C} \mid \underline{D}]$

and $\underline{A}_2 := [\underline{G} \mid \underline{H}]$

equation (I-1) is transformed to

$$[\underline{C} \quad \underline{D}]\begin{bmatrix}\underline{\delta}_1\\ \underline{\delta}_2\end{bmatrix} + \underline{Bv} + \underline{w}_1 = \underline{0} \qquad \text{- (I-2)}$$

$$[\underline{G} \quad \underline{H}]\begin{bmatrix}\underline{\delta}_1\\ \underline{\delta}_2\end{bmatrix} + \underline{w}_2 = \underline{0} \qquad \text{- (I-3)}$$

Solving equation (I-3),

$$\underline{\delta}_2 = -\underline{H}^{-1}(\underline{w}_2 + \underline{G\delta}_1) \qquad \text{- (I-4)}$$

Note that the $\underline{H}^{-1}$ matrix must exist for this technique to be feasible. A discussion relating this "constraint" to the spline algorithm was presented under Section 7.2.

Eliminating $\underline{\delta}_2$ in equation (I-2),

$$\underline{C\delta}_1 - \underline{DH}^{-1}(\underline{w}_2 + \underline{G\delta}_1) + \underline{Bv} + \underline{w}_1 = \underline{0} \qquad \text{- (I-5)}$$

Rearranging equation (I-5),

$$[\underline{C} - \underline{DH}^{-1}\underline{G}]\underline{\delta}_1 + [\underline{w}_1 - \underline{DH}^{-1}w_2] + \underline{Bv} = \underline{0} \qquad \text{- (I-6)}$$

By letting

$$\underline{\bar{A}} := [\underline{C} - \underline{DH}^{-1}\underline{G}]$$

and $\underline{\bar{w}} := [\underline{w}_1 - \underline{DH}^{-1}\underline{w}_2]$

and with the assumption that the model is of a parametric nature, the direct least-squares treatment of equation (I-6) yields:

$$\underline{\hat{\delta}}_1 = -(\underline{\bar{A}}^T \underline{P}\underline{\bar{A}})^{-1} \underline{\bar{A}}^T \underline{P}\underline{\bar{w}} := -\underline{\bar{N}}^{-1}\underline{\bar{A}}^T \underline{P}\underline{\bar{w}} \qquad \text{- (I-7)}$$

Through back substitution,

$$\underline{\hat{\delta}}_2 = -\underline{H}^{-1}[\underline{w}_2 - \underline{G}(\underline{\bar{A}}^T \underline{P}\underline{\bar{A}})^{-1} \underline{\bar{A}}^T \underline{P}\underline{\bar{w}}] \qquad \text{- (I-8)}$$

Using the covariance law (see Section 8.1), the covariance of the subvectors of the least-squares estimates are as follows:

$$\Sigma_{\underline{\hat{\delta}}_1} := \underline{J}\underline{C}_{\underline{\bar{w}}}\underline{J}^T \quad , \underline{J} := \frac{\partial \underline{\hat{\delta}}_1}{\partial \underline{\bar{w}}}$$

$$= (\underline{\bar{N}}^{-1}\underline{\bar{A}}^T \underline{P}) \; \underline{\Sigma}_{\ell} \; (\underline{P}\underline{\bar{A}}\underline{\bar{N}}^{-1}) \qquad \text{- (I-9)}$$

Simplifying,

$$\underline{\Sigma}_{\hat{\delta}_1} = \underline{\bar{N}}^{-1} \qquad \text{- (I-10)}$$

In a similar manner,

$$\underline{\Sigma}_{\hat{\delta}_2} = (\underline{H}^{-1}\underline{G})\underline{\bar{N}}^{-1}(\underline{H}^{-1}\underline{G})^T$$

and $$\underline{\Sigma}_{\hat{\delta}_1\hat{\delta}_2} = -\underline{\bar{N}}^{-1}(\underline{H}^{-1}\underline{G})^T$$

Finally, in summary,

$$\underline{\hat{\delta}} = [\underline{\hat{\delta}}_1 \;\vdots\; \underline{\hat{\delta}}_2]$$

and

$$\underline{\Sigma}_{\hat{\delta}} = \left[\begin{array}{c|c} \underline{\bar{N}}^{-1} & -\underline{\bar{N}}^{-1}(\underline{H}^{-1}\underline{G})^T \\ \hline -(\underline{H}^{-1}\underline{G}\underline{\bar{N}}^{-1}) & \underline{H}^{-1}\underline{G}\underline{\bar{N}}^{-1}(\underline{H}^{-1}\underline{G}^T)^T \end{array}\right] \quad \text{- (I-11)}$$

Hence for the spline algorithm, the parameter solution through the elimination of constraints is possible only if $\underline{H}^{-1}$ matrix exists (equation I-4).

Three routines (LS2, TRNSFM and PERM) have been formulated and listed in Appendix VII. Subroutine TRNSFM reformats the primary and secondary design matrices into their various submatrices. Subroutine LS2 performs the least-squares adjustment using the method described above. Finally subroutine PERM unscrambles the least-squares estimates and covariances to conform to the rest of the DSPLIN program. This has to be done because the DSPLIN program expects the solution vector and the covariance matrix to be of the form produced by the functionally constrained least-squares method.

Appendix II

COMPUTATIONAL ACCURACY

In this appendix, the aspect of computational accuracy is discussed. Primarily there are two possible conditions which can lead to a degradation in the computational precision of the solution vector in the linear system of equations, $\underline{N}\ \underline{x} = \underline{b}$, that is, the use of floating-point arithmetic and the presence of an unstable algorithm.

The accuracy of floating-point arithmetic [Forsythe et al. 1977: page 13] can be characterized at run time by the so-called "machine epsilon"; the smallest floating point number ε such that

$$1 \oplus \varepsilon > 1 \qquad \text{- (II-1)}$$

Round-off error, through a phenomenon called "catastrophic cancellation" [Forsythe et al. 1977: page 15] in badly conceived computation sequences can result in a near total loss of significant digits. Although it is possible to carry more digits to avoid this phenomenon, it is always more costly in terms of execution time and storage space. A better approach is to keep a good track of the arithmetic process and the expected sizes of the numbers involved in the computations.

An estimate of the internal computational accuracy can be arrived at through the manipulation of the results such that a null vector is formulated. In the direct least-squares treatment of the system of equations, $\underline{N}\ \underline{x} = \underline{b}$, where $\underline{N} = (\underline{A}^T \underline{PA})$ and $\underline{b} = -(\underline{A}^T \underline{Pw})$, the null vector, $\underline{e}$ can be expressed as $\underline{e} = \underline{A}^T\ \underline{P}(\underline{A\hat{x}} + \underline{w})$. For the functionally constrained least-squares model, the vector $\underline{e}$ takes the form:

$$\underline{e} = \underline{\Sigma}_{\underline{x}}^{T}\ \underline{A} \qquad \text{- (II-2)}$$

with $\underline{\Sigma}_{\underline{x}}$ and $\underline{A}$ matrices as defined in Section 8.2. Theoretically, the vector $\underline{e}$ is a null vector. However, with the use of finite precision arithmetic, the maximum $|e|$, $(e \in \underline{e})$ so obtained, gives a good estimate of the computational accuracy of the algorithm.

For certain problems, good answers cannot be obtained no matter how well the algorithm is conceived. This aspect of numerical analysis is independent of the floating-point number system or the algorithm used.

When small errors in the right hand side of a system of linear equations causes a large effect on the solution vector, the problem is termed "<u>ill-conditioned</u>", i.e. it is very sensitive to the values on the right-hand side. Ill-conditioning occurs when the transformation (or coefficient) matrix $\underline{N}$ is nearly singular. An example of such an effect is illustrated in Westlake [1968]: page 89. If $\underline{N}$ is a singular matrix, then for some $\underline{b}$'s a solution of $\underline{x}$ does not exist. If $\underline{N}$ is nearly singular, then small changes in $\underline{b}$ would result in disproportionally large changes in $\underline{x}$.

To have a more precise and comprehensible measure of nearness to singularity, there exist the "condition numbers". Determinants of unnormalised transformation matrices are ineffective as a yardstick for conditioning as the equations can be premultiplied by any constant to obtain any value of det($\underline{N}$).

Considering the system of equations $\underline{N}\ \underline{x} = \underline{b}$, introducing an error of $\Delta\underline{b}$ in $\underline{b}$ results in a change of $\Delta\underline{x}$ in $\underline{x}$. The condition number of $\underline{N}$ is then defined as the ratio between the relative change in the right-hand side and the relative error caused by this change [Forsythe et al. 1977: page 43], i.e.

$$\mathrm{cond}(\underline{N}) \geq \frac{\|\Delta\underline{x}\| \,/\, \|\underline{x}\|}{\|\Delta\underline{b}\| \,/\, \|\underline{b}\|} \qquad \text{- (II-3)}$$

where $\|\ \|$ denotes the norm of a vector.

The condition number, as mentioned earlier, is a measure of the nearness to singularity and can be thought of as the reciprocal of the relative distance of the matrix to a set of singular matrices [Forsythe et al. 1977: page 43]. If cond($\underline{N}$) is large, then $\underline{N}$ is close to being singular.

The actual computation of cond($\underline{N}$) involves knowing the inverse of $\underline{N}$. If $\underline{n}_j$ and $\underline{n}'_j$ are the column vectors of $\underline{N}$ and $\underline{N}^{-1}$ respectively, then in terms of the Euclidean vector norm ($\|x\|^2 := \sum_{j=1}^{m} x_j^2$).

$$\text{cond}(N) = \max_j \left\| \underline{n}_j \right\| \cdot \max_j \left\| \underline{n}'_j \right\| \qquad - \text{(II-4)}$$

Other norms, like the sum of absolute values can be used as an alternative to the expensive-to-evaluate Euclidean norm.

This definition represents but only one of the family of possible condition numbers; Turing's m-condition number, Von Neuman and Goldstine's P-condition number, are but a few [Westlake 1968: page 90].

In the realm of computational accuracy estimates in performing the inverse, the power of the condition number is an approximate estimate as to the number of significant digits lost during the inversion process.

A third method which assesses the cumulative loss in significant figures due to round-off and conditioning of the transformation matrix, is the multiplication of the inverse with the uninverted matrix, i.e. $\underline{N}^{-1}\underline{N} = [\underline{I}]$. The technique requires a massive amount of array space when large inverses are contemplated. Hence it is seldom used.

In relation to the program DSPLIN, both the condition numbers (computed via equation II-4) and the internal computational accuracy estimates (computed via equation II-2) are evaluated to assess the conditioning of the inverses and the computational accuracy of the least-squares estimates (in equations 7.9).

Appendix III

ALGORITHM FOR COMPUTING MAXIMUM SATELLITE ELEVATION

The maximum elevation of the Transit satellite can be evaluated solely from geometrical considerations given the position of observer and the velocity vector of the satellite at point of closest approach (obtained from the formal error ellipse associated with each position determination - see Section 10.2.2).

Consider Figure III-1, we have

$$(y + Re)/Rs = \cos \gamma \qquad \text{- (III-1)}$$

Rearranging

$$y = Rs \cos \gamma - Re$$

and

$$x = Rs \sin \gamma \qquad \text{- (III-2)}$$

with Rs = radius of the satellite at the center of the earth

and Re = radius of the observer at the center of the earth

Now, the elevation angle of the satellite (E) can be expressed as

$$E = \tan^{-1} [y/x]$$

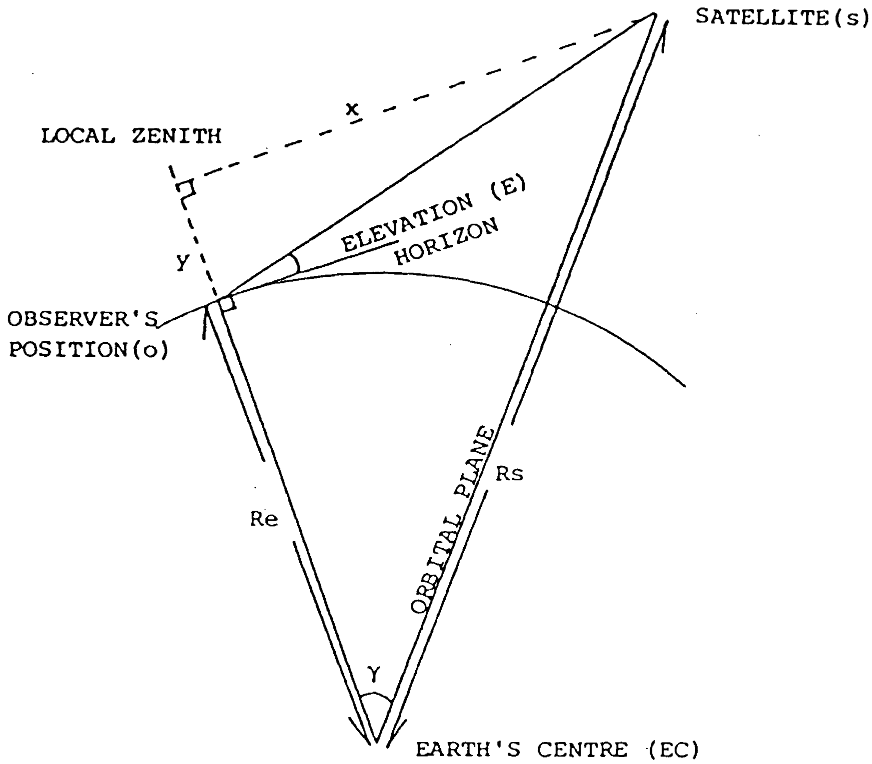


FIGURE III-1 Plane defined by the Earth's centre, the observer's position and the point of closest approach of the satellite.

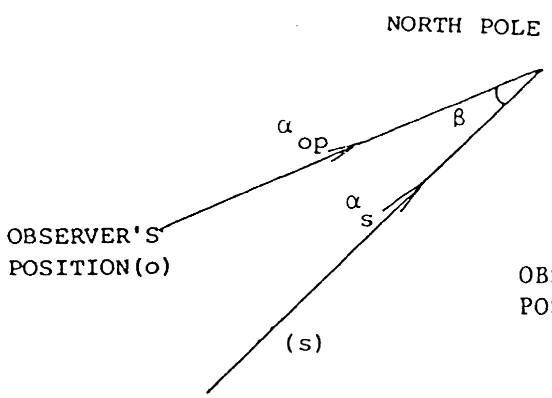


Figure III-3 North Pole view in the Polar Stereographic Coordinate System.

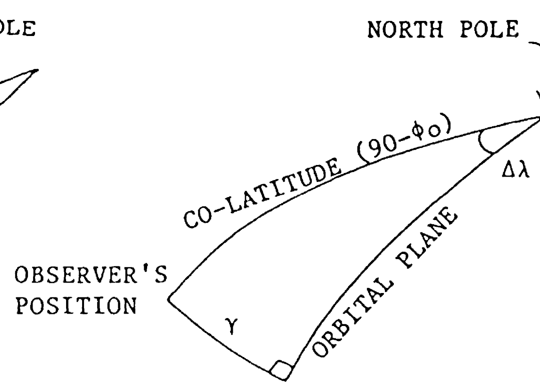


FIGURE III-2 Spherical triangle defined by the observer's position, point of closest approach of the satellite and the North Pole.

Substituting for x and y,

$$E = \tan^{-1} \frac{\cos\gamma - Re/Rs}{\sin\gamma} \qquad \text{- (III-3)}$$

Assuming that all satellites pass directly over the North Pole, the angle γ can be expressed as being (see Figure III-2):

$$\sin\gamma = \cos\phi_o \sin\Delta\lambda \qquad \text{- (III-4)}$$

where ϕ_o := observer's latitude,

and $\Delta\lambda$:= difference in longitude between observer and satellite orbital plane.

However, the difference in longitude (i.e. $\Delta\lambda$) between the observer and the ground track of the satellite is the difference between the direction vector from the observer to the pole (α_{op}) and the velocity vector of the satellite at point of closest approach (α_s) (Figure III-3).

Hence given the position of the observer and the velocity vector of the satellite at the point of closest approach, the maximum elevation of the satellite can be computed (i.e. via equation III-3).

[Reference : Wells 1974: page 131]

Appendix IV

FORMULATION OF SIMULATED DATA SET

In this appendix, the explicit equations used to generate the simulated data points are given along with the data points chosen as being the simulated data set. There are no specific reasons behind chosing this form to portray the simulated data set. The data set need only be smooth, not necessarily periodic, and with some superimposed "noise".

The following are the formulas used:

$$(x,y) := (\ a \sin(w_x t + \phi_x) + R_x\ ,\quad b \sin(w_y t + \phi_y) + 3t + 250 + R_y\ ) \qquad \text{- (IV-1)}$$

with

$$a := 1400 \text{ m}$$

$$b := 250 \text{ m}$$

$$w_x := 2*\pi/440$$

$$w_y := 2*\pi/70$$

$$\phi_x := 2^o$$

$$\phi_y := 40^o$$

$$R_x := 5 \sin(2*\pi t/1.5)$$

$$R_y := -5 \sin(2*\pi t/1.5 - \pi)$$

and

$$t := 0,2,4,\ldots,100\ .$$

Basically, the x component has only a periodic term and a phase offset, whereas the y component incorporates, in addition, a linear trend. Both coordinates have a high frequency term (i.e. R_x and R_y) which acts as noise on the simulated data set.

TABLE IV-1

SIMULATED DATA SET

Time	Easting (x)	Northing (y)	Time	Easting (x)	Northing (y)
0	48.859	410.697	36	730.550	180.783
4	124.325	475.412	42	830.070	133.425
10	243.151	525.592	52	977.734	203.142
14	330.073	528.126	60	1089.400	380.462
20	445.395	465.281	66	1160.651	531.156
24	516.200	393.303	70	1199.122	616.366
26	577.481	359.702	72	1223.388	658.309
28	585.321	312.231	88	1341.437	693.779
30	625.670	278.311	92	1369.932	643.273
32	665.509	246.374	94	1369.522	598.973
			98	1391.317	530.892

Note : All points are assigned a variance of 25 square metres in x and y, and a covariance of 0.0001 square metres. The selection of the points constituting the simulated data set is described in Section 9.1.

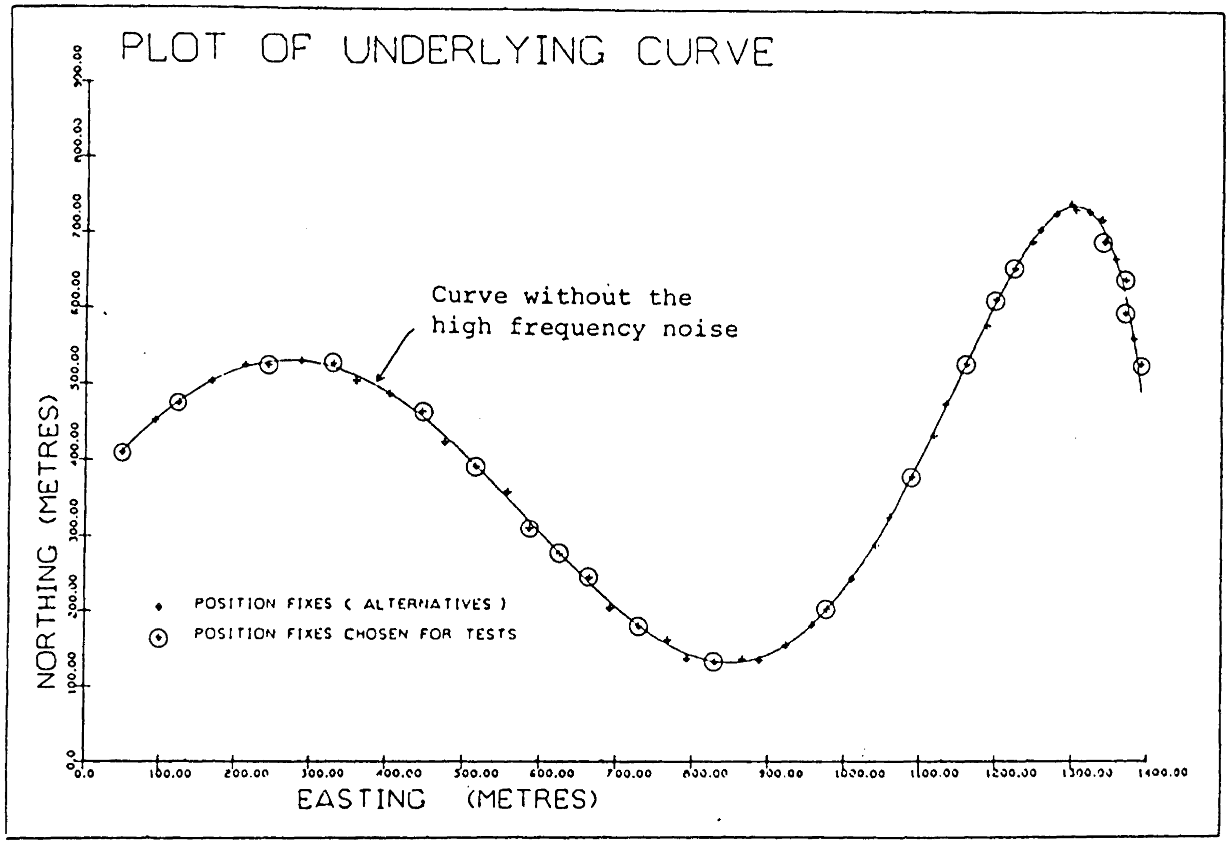


FIGURE IV-1 Location of the simulated data points chosen to test the spline algorithm

Appendix V

DESCRIPTION OF LOREX-79 TEST DATA SETS

In this appendix, a brief description and nature of the selected three disjointed stretches of the LOREX Main Camp positioning data used in evaluating the performance of the spline algorithm are given.

To aptly sample the LOREX Doppler data, three seperate segments of the satellite fix data series were chosen. This is done to allow the evaluation of the algorithm (see Chapter 9) under a variety of different conditions at the time of the observation of the Transit satellites.

V.1 LOREX TEST DATA SET ONE

Period : From day 113 hr 19 min 38
to day 116 hr 19 min 44

Number of points : 100

Receiver's motion : Fairly constant; velocity northwards was about 100 metres per hour and velocity eastwards was about 120 metres per hour.

Data Plots : Figures V-1 (or V-1a), V-2 and V-3

V.2 LOREX TEST DATA SET TWO

Period : From day 120 hr 2 min 18
to day 123 hr 3 min 2

Number of points : 100

Receiver's motion : Accelerating; velocity north changes from +50 to -50 metres per hour and velocity east changes from -110 to -620 and finally to -300 metres per hour.

Data Plots : Figures V-4, V-5 and V-6

V.3 LOREX TEST DATA SET THREE

Period : From day 130 hr 2 min 30
to day 139 hr 8 min 30

Number of points : 100

Receiver's motion : Decelerating; ice platform slows down from 90 metres per hour, stops and moves off in a new direction at 140 metres per hour.

Data Plots : Figures V-7, V-8 and V-9

Note : All plotted ellipses are drawn using the formal covariance matrix of the satellite fixes at the 99 percent confidence level.

FIGURE V-1 OBSERVED POSITION FIXES OF TEST DATA SET ONE. ALL ERROR ELLIPSES ARE AT THE 99 % CONFIDENCE LEVEL

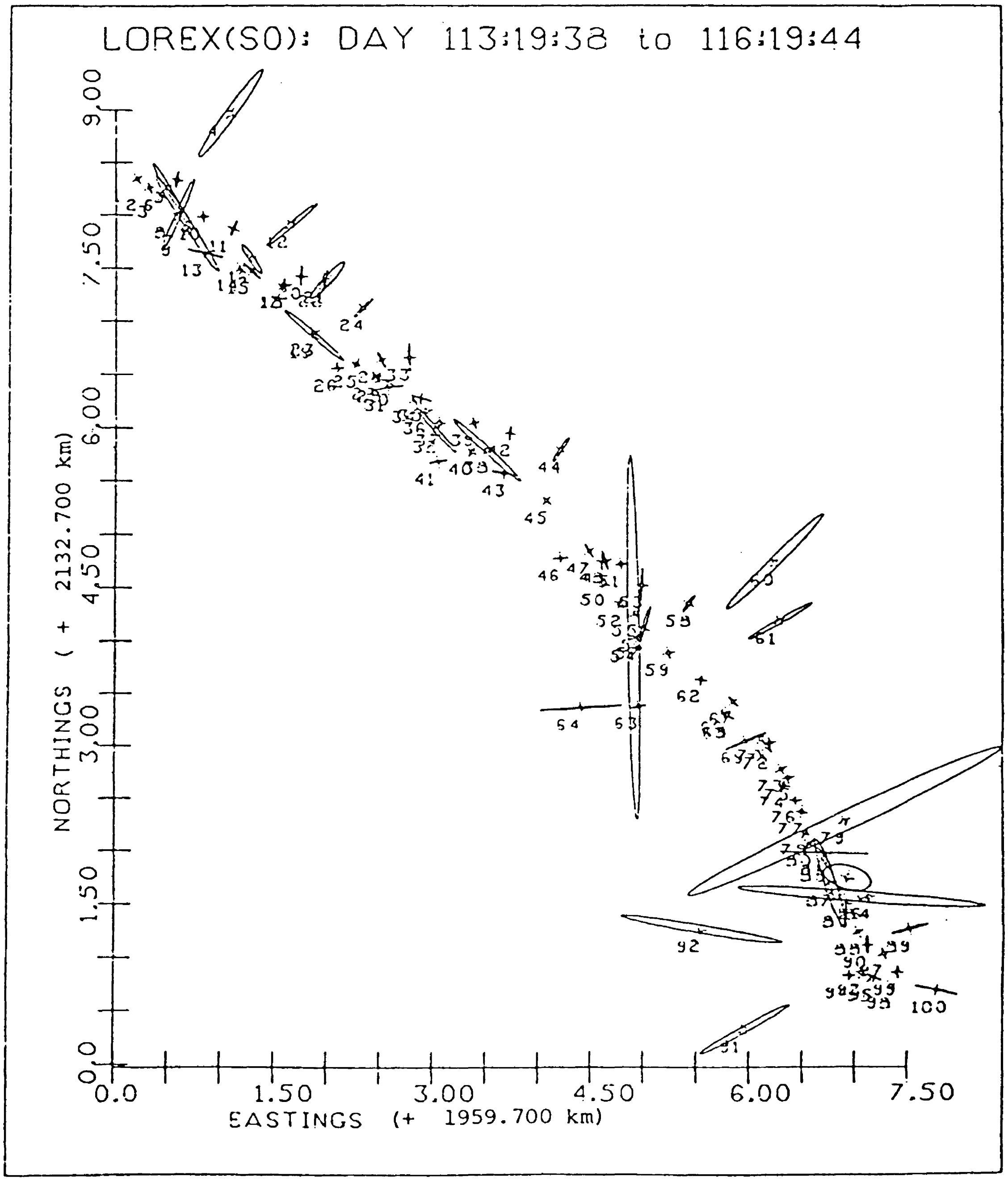


FIGURE V-1a OBSERVED POSITION FIXES OF TEST DATA SET ONE. ALL ERROR ELLIPSES WITH STD DEV. OF LESS THAN 20 m HAVE BEEN SCALED UP EQUALLY IN EASTINGS AND NORTHINGS

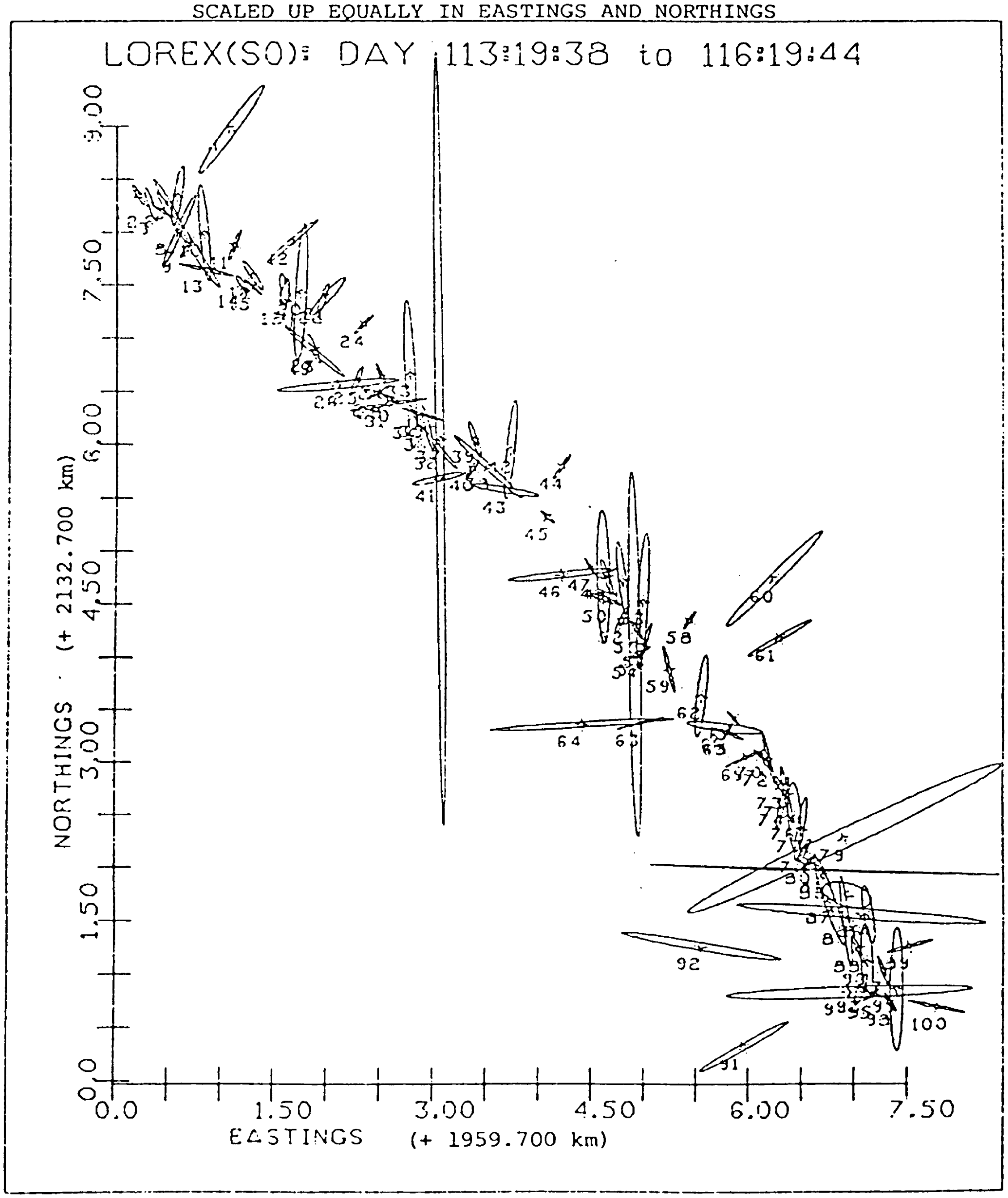


FIGURE V-2 OBSERVED POSITIONS OF TEST DATA SET ONE PORTRAYED IN THE TIME DOMAIN

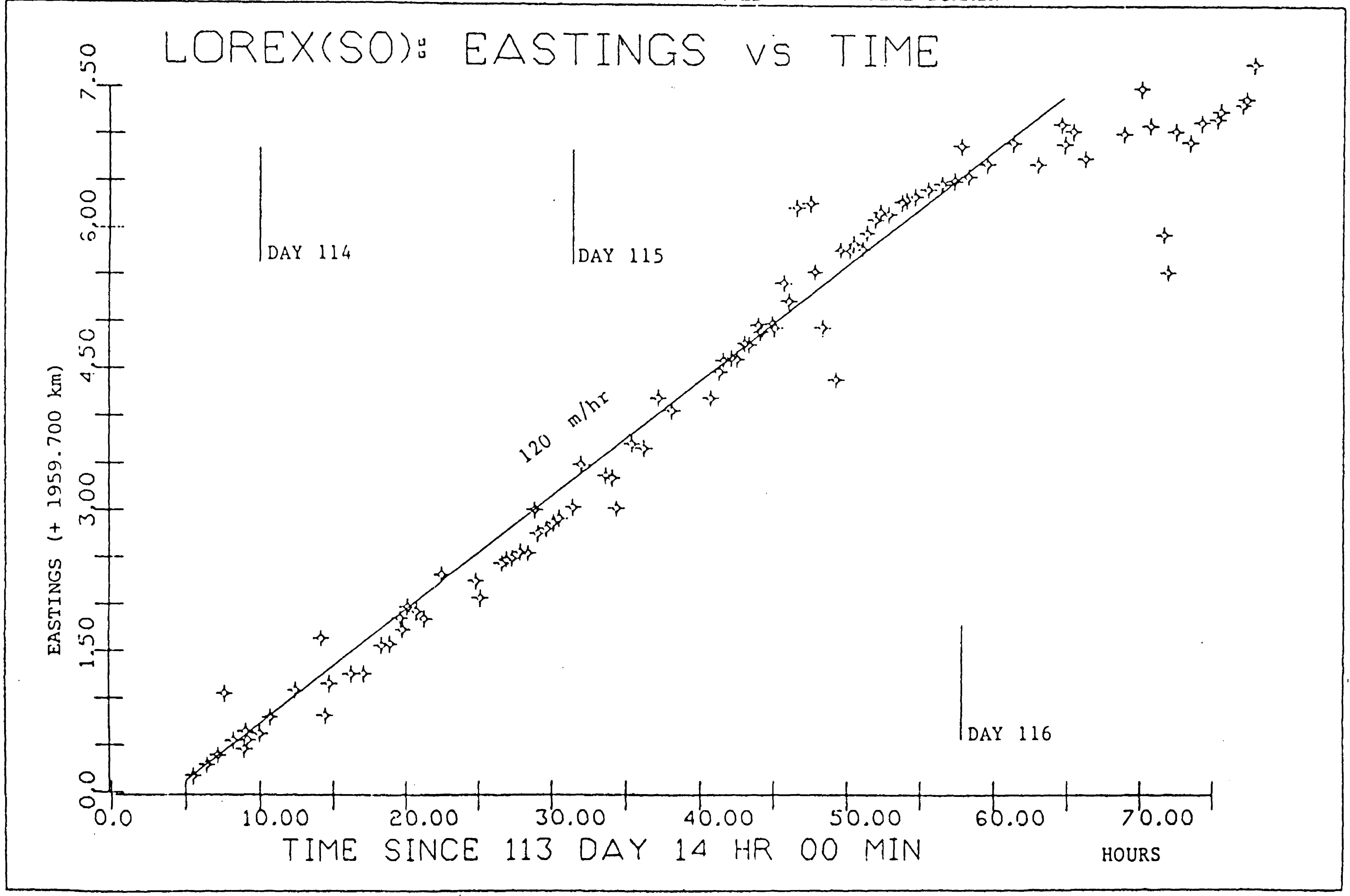


FIGURE V-3 OBSERVED POSITIONS OF TEST DATA SET ONE PORTRAYED IN THE TIME DOMAIN

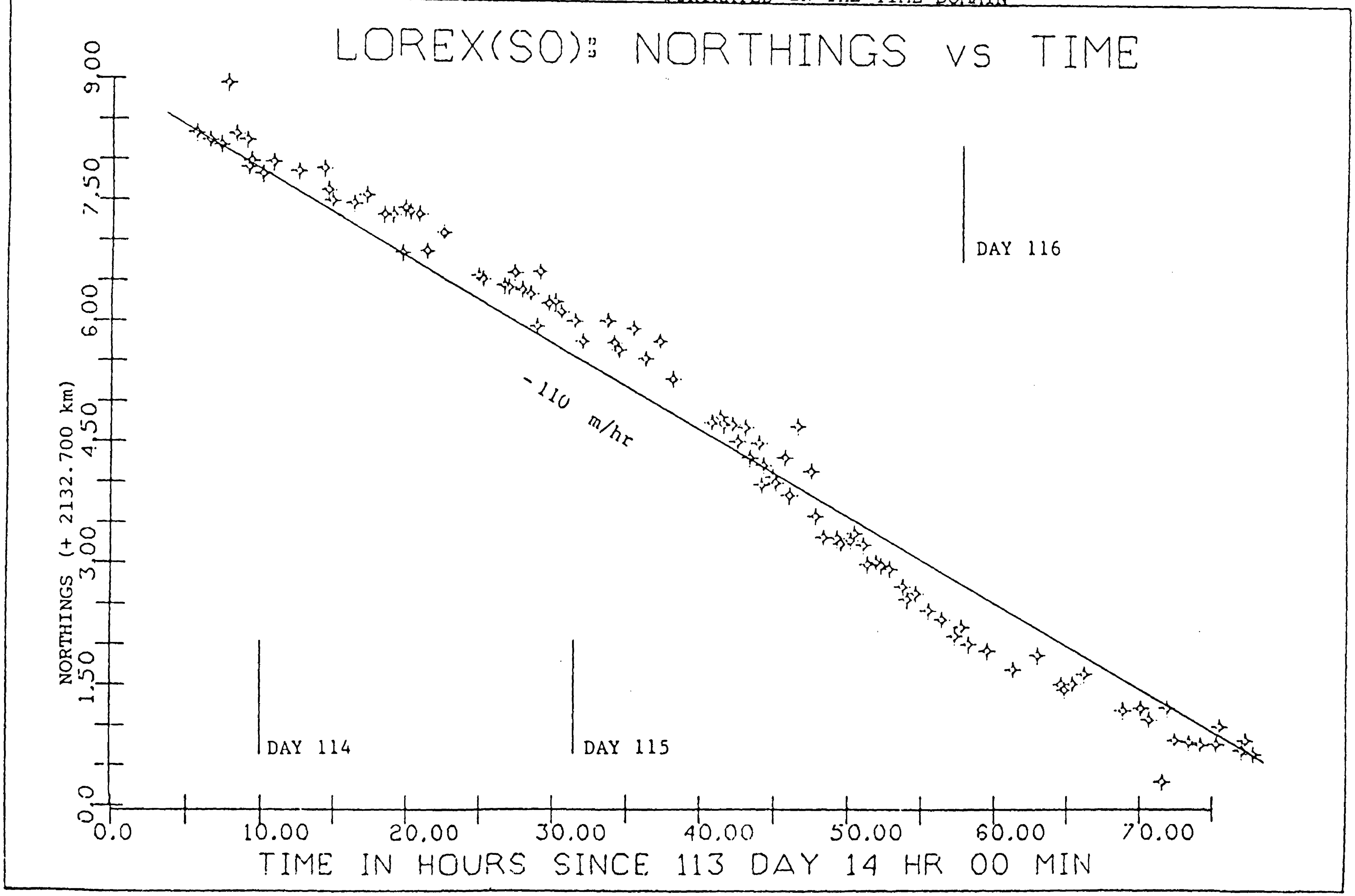


FIGURE V-4 OBSERVED POSITION FIXES OF TEST DATA SET TWO

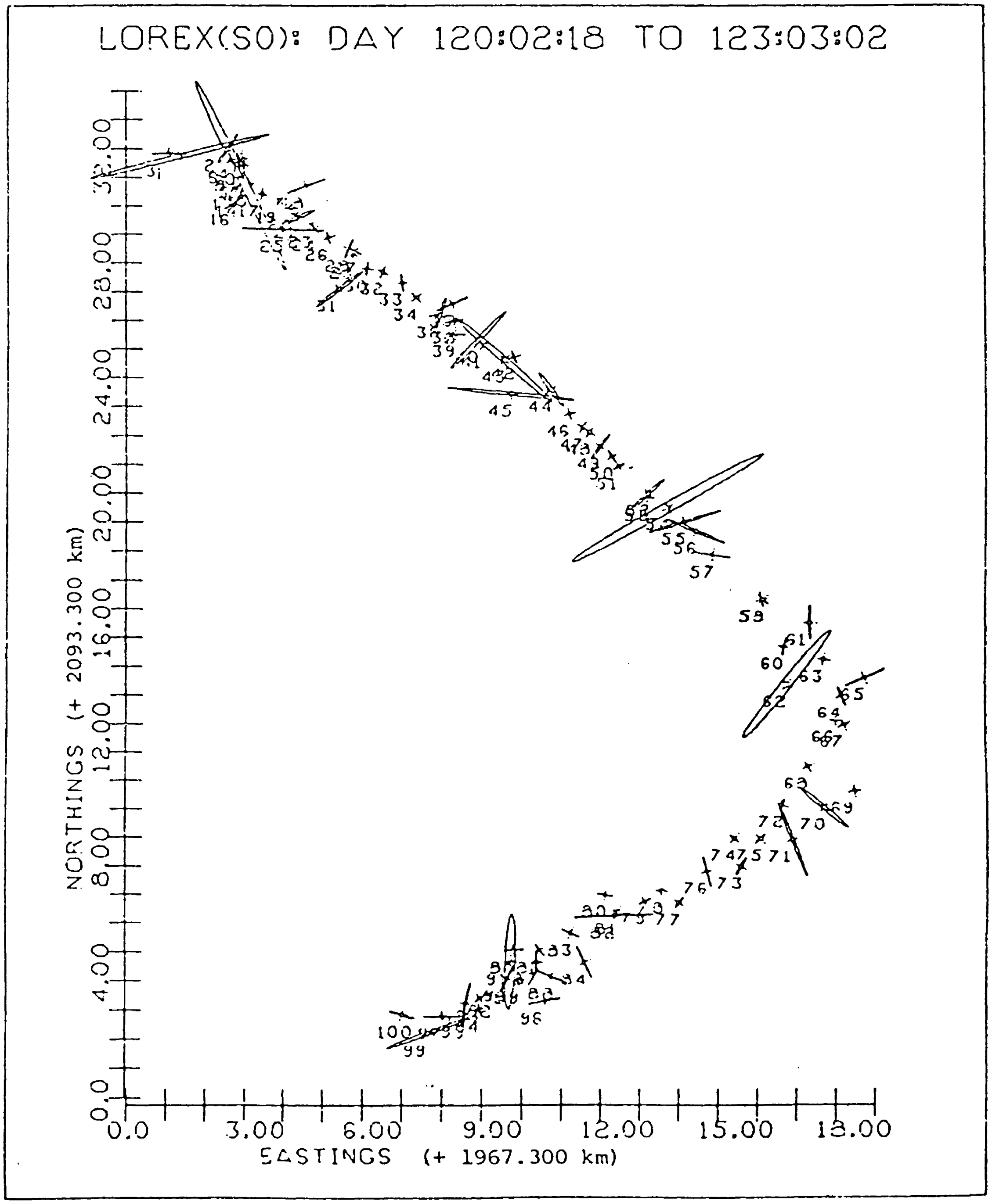


FIGURE V-5 OBSERVED POSITIONS OF TEST DATA SET TWO PORTRAYED IN THE TIME DOMAIN

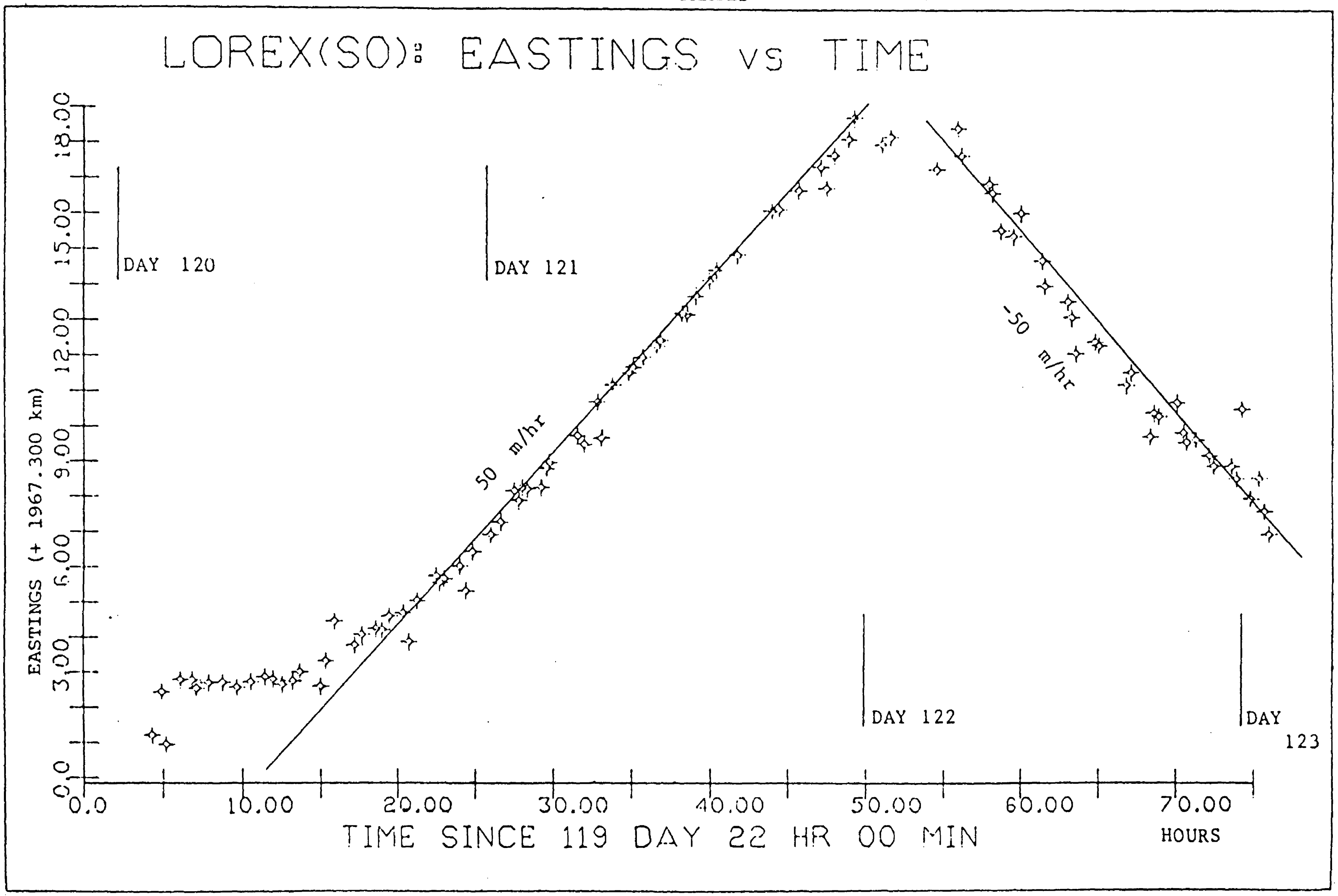


FIGURE V-6 OBSERVED POSITIONS OF TEST DATA SET TWO PORTRAYED IN THE TIME DOMAIN

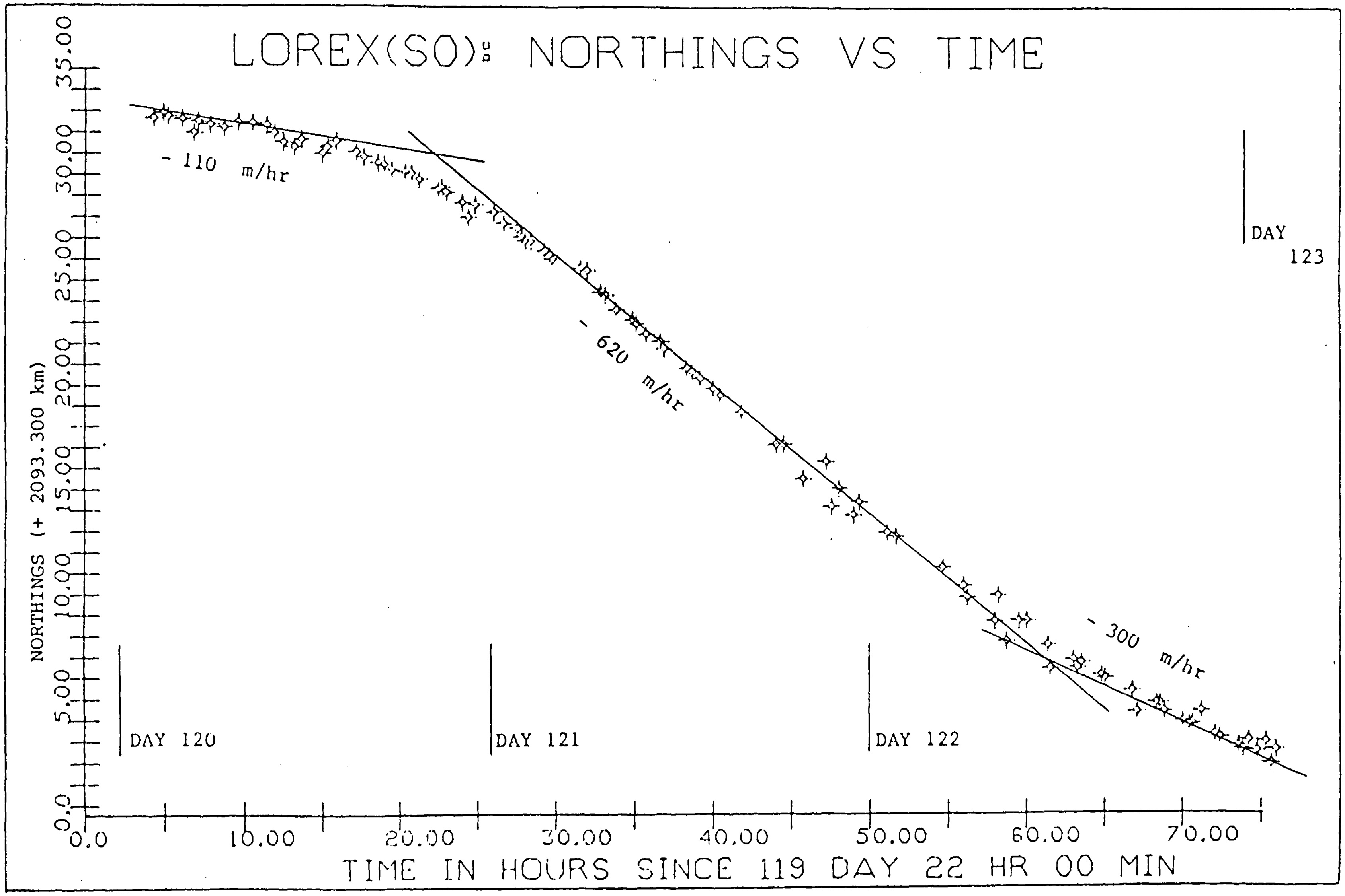


FIGURE V-7 OBSERVED POSITION FIXES OF TEST DATA SET THREE

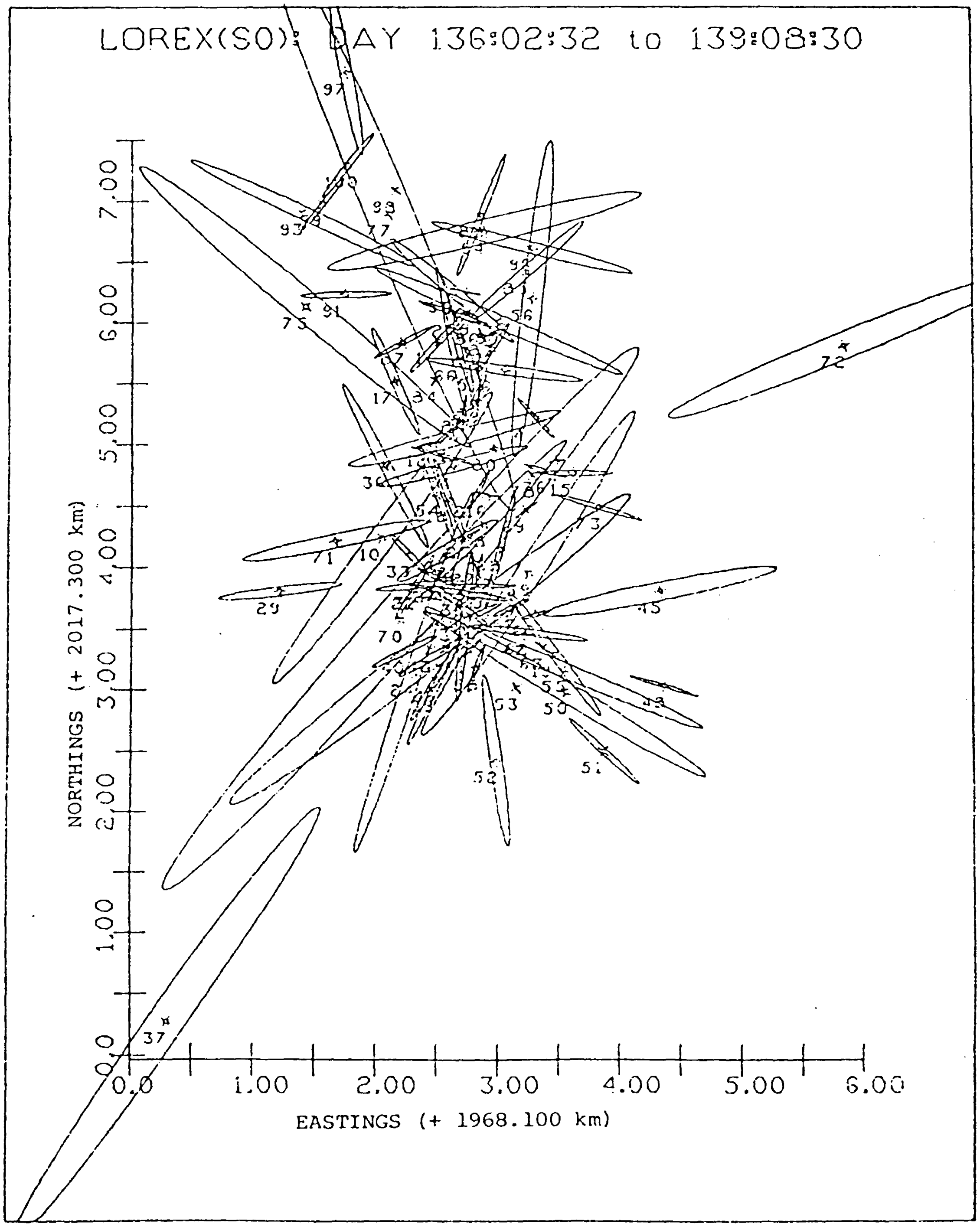


FIGURE V-8 OBSERVED POSITIONS OF TEST DATA SET THREE PORTRAYED IN THE TIME DOMAIN

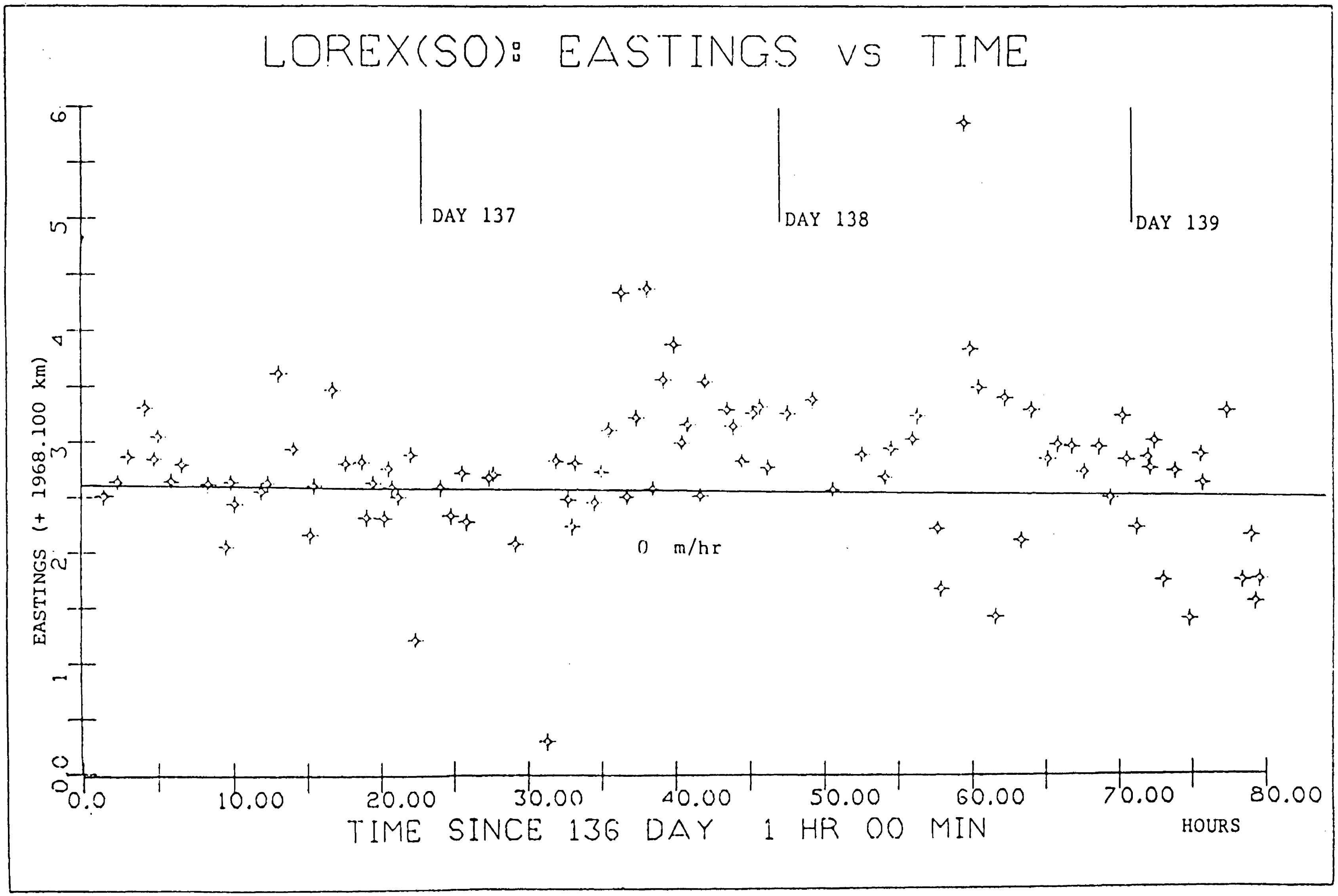
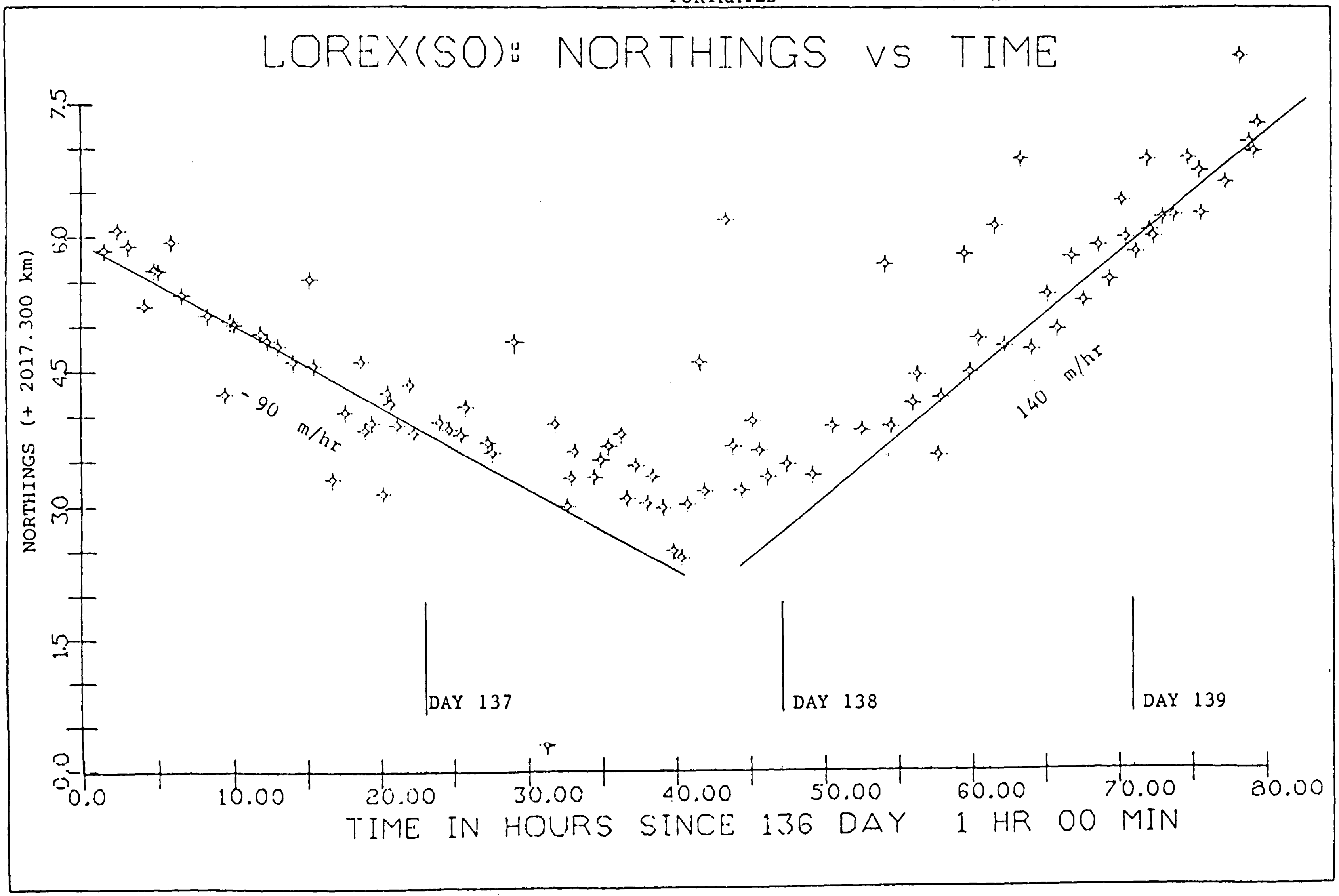


FIGURE V-9 OBSERVED POSITIONS OF TEST DATA SET THREE PORTRAYED IN THE TIME DOMAIN



APPENDIX VI

TRANSFORMATION FROM ERROR ELLIPSES TO STANDARD DEVIATIONS

In this appendix, the algorithm used to transform the parameters of an error ellipse to its corresponding standard deviation is described.

The following are the assumed knowns and unknowns (Figure VI-1)

Known

- a - semi-major axis of the error ellipse
- b - semi-minor axis of the error ellipse
- γ - direction of semi-major axis of error ellipse with respect to the x-axis (or $90^\circ - \Theta$; with Θ being the azimuth of the semi-major axis of the error ellipse).

Unknown

- σ_x - standard deviation along the x - axis
- σ_y - standard deviation along the y - axis
- σ_{xy} - covariance between coordinates.

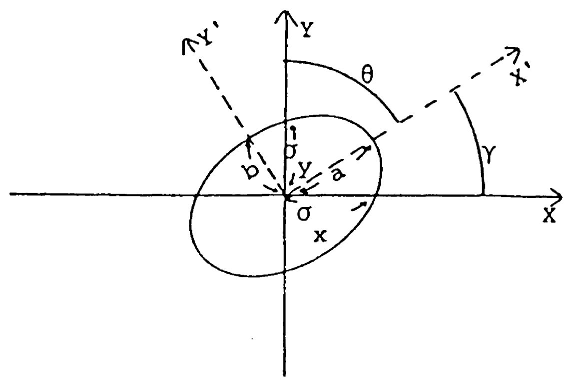


FIGURE VI-1 Relationship Between Error Ellipse and Covariance Matrix.

The transformation from the error ellipse (i.e. a, b and γ) to the covariance matrix (i.e. σ_x, σ_y and σ_{xy}) is achieved using rotation matrices, i.e.

$$\begin{bmatrix} x \\ y \end{bmatrix} = R(-\gamma) \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad \text{- (VI-1)}$$

where

$$R(-\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix}$$

is the negative rotation matrix for the coordinate axes.

Now, the covariance matrix in the x' y' coordinate system can be written as:

$$C_{x'y'} = \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix}$$

Therefore, using the covariance law (as outlined in Chapter 8):

$$\begin{aligned} C_{xy} &= R(-\gamma)\, C_{x'y'}\, R(-\gamma)^T \\ &= R(-\gamma)\, C_{x'y'}\, R(\gamma) \qquad \text{- (VI-2)} \end{aligned}$$

or

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \quad \text{- (VI-3)}$$

In terms of θ, we have

$$C_{xy} = R(\theta - 90^\circ)\, C_{x'y'}\, R(90^\circ - \theta) \quad \text{- (VI-4)}$$

Rearranging equations (VI-3),

$$\sigma_x = [\tfrac{1}{2}(a^2 + b^2 + 2\sigma_{xy}\cot 2\gamma)]^{\frac{1}{2}}$$

$$\sigma_y = [\tfrac{1}{2}(a^2 + b^2 - 2\sigma_{xy}\cot 2\gamma)]^{\frac{1}{2}}$$

and $$\sigma_{xy} = \tfrac{1}{2}(a^2 - b^2)\sin 2\gamma \quad \text{- (VI-5)}$$

(Note: If the azimuth (θ) is given instead of γ, replace γ by ($90^\circ - \Theta$) in equations (VI-5)).

Appendix VII

ALGORITHM DESIGN, IMPLEMENTATION AND COMPUTER LISTINGS

In this appendix, the general flow of the algorithm, together with any special considerations made are mentioned.

The basic structure of the program is given in Figure VII-1. The routines are classified into "levels"; with the following definitions:

Level I - routines called only once to perform a specific task.

Level II - routines called more than once, but of a specialised nature.

Level III - routines called extensively by other routines.

Displayed in Figure VII-1 are all level I subroutines called during the processing of a data set and with the order of execution being from the left to right. A complete description of the function of each routine, along with the input and output parameters and external routines called, are given in their computer listings. The list of 64 subroutines developed for this thesis, together with a short description and external routines called are given in Table VII-1.

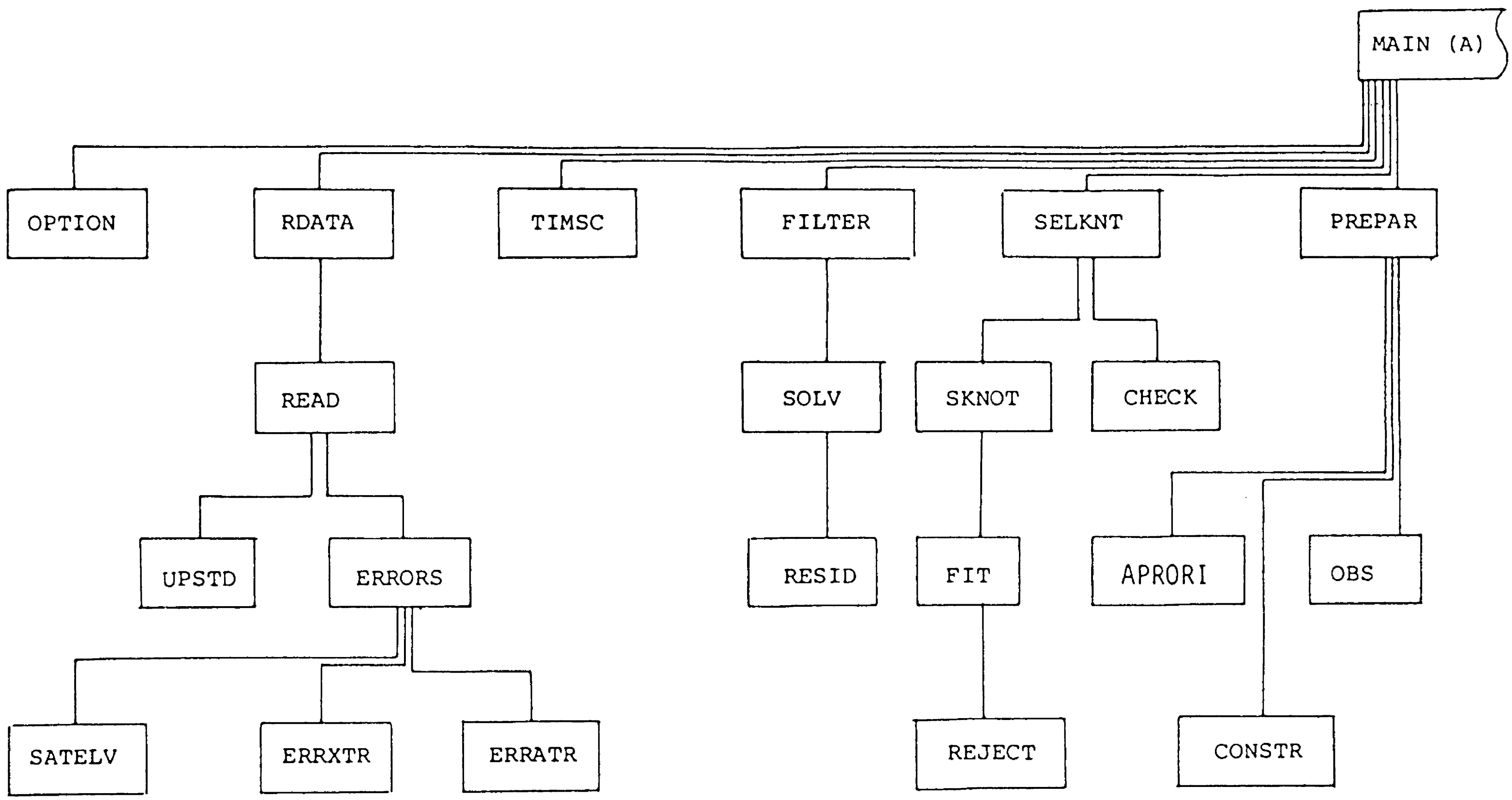


FIGURE VII - 1a Basic Structure of the DSPLIN Program

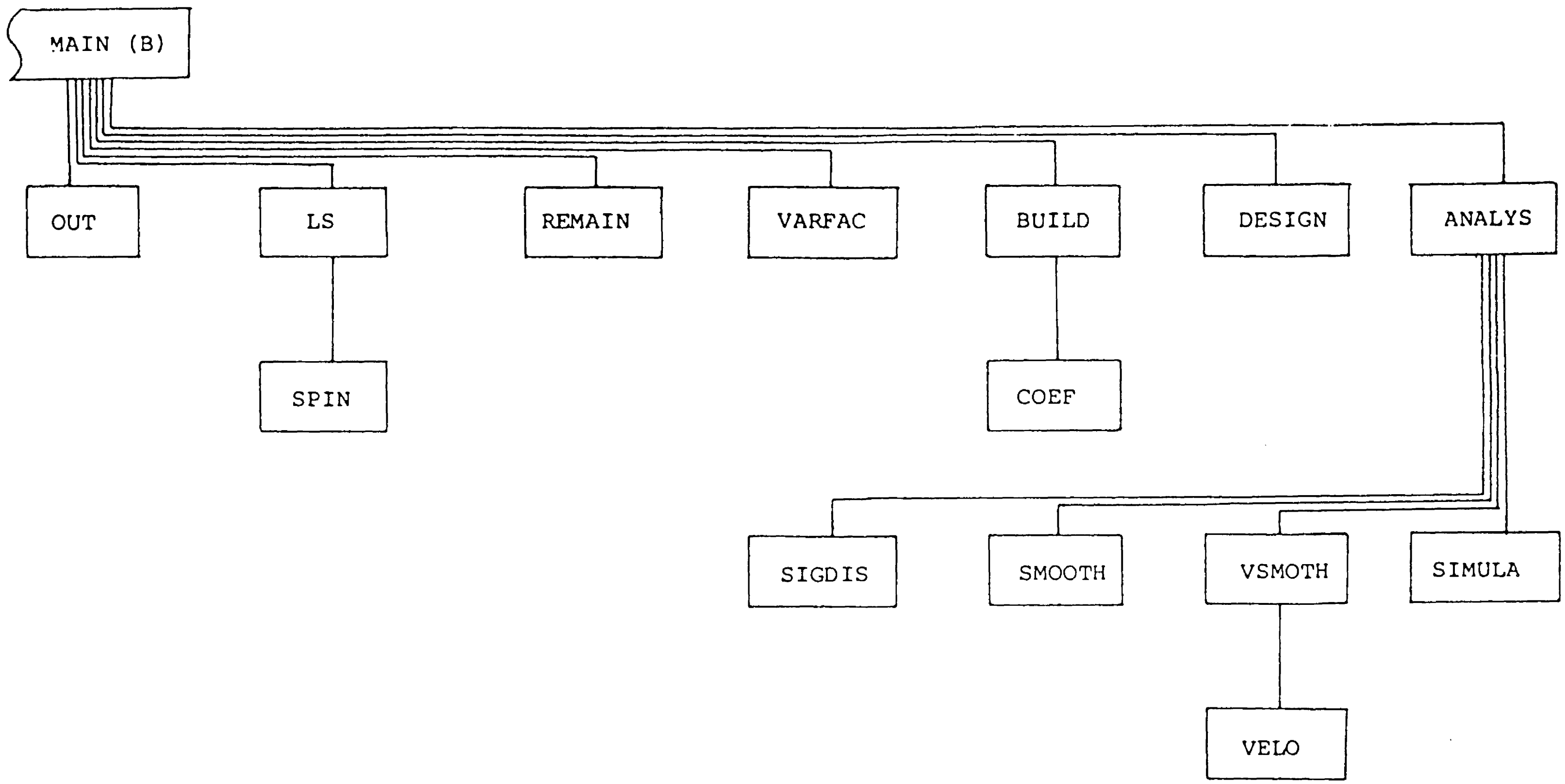


FIGURE VII - 1b Basic Structure of the DSPLIN Program

TABLE VII-1

List and Description of Subroutines

I) Level I subroutines

Name		Description and external routines called
ANALYS	:	Plots the different components of the least-squares residuals and printing of smoothed data series at equal time intervals. Calls - CTIME,DSHLNS,DSQRT,ELIPSE,ENDPLT, NOWPLT,PCOPY,PLOT,POINT,PRNTCH,SCMULT, SIGDIS,SIMULA,SMOOTH,VSMOTH,
APRORI	:	Reads or generates an a priori solution vector for the least-squares adjustment. Calls - none
BUILD	:	Transforms position and velocity knot vectors into cubic coefficients. Driver routine for COEF. Calls - COEF,DPRINT,PCOPY
CHECK	:	Checks the number of data points per cubic knot interval and drops superfluous knots. Calls - INTERV
COEF	:	Transforms a set of adjacent knot vectors into cubic coefficients defining the interval between the two knots. Calls - DMULT
CONSTR	:	Generate the coefficients of a constraint equation. Calls - none
DESIGN	:	Display of knots, data points and smoothed points at time series plots, individually or overlayed in the Polar Stereographic Coordinate system. Calls - CIRCLE,DSHLNS,DSQRT,ELIPSE,ELLIPS, ENDPLT,NMBR,NOWPLT,PLOT,POINT,PRNTCH, RADD,SPSYMB
ERRATR	:	Computes along track fix error due to estimated receiver motion and height errors. Calls - DSQRT

TABLE VII-1 - cont'd

ERRXTR	:	Computes cross track fix error due to estimated receiver motion and height errors. Calls - DSQRT
ERRORS	:	Modify input formal covariance matrices based on computed fix errors in the along and cross track directions of the passing satellite. Calls - DCOS,DSIN,ELIPSE,ELLSIG,ERRATR,ERRXTR, SATELV
FILTER	:	Linear filter routine for detection of outliers. Calls - DSQRT,DABS,SOLV
FIT	:	Cubic curve fitting using weighted beginning position and slope vectors. Calls - CHOLD,OBS,REJECT
LS	:	Least-squares adjustment of the functionally constrained least-squares model. Calls - COND,DABS,SPIN
LS2	:	Least-squares adjustment using the method of elimination of constraints. Calls - CHOLD,COND,DINV,DMAG,DMULT,DPRINT, PCOPY,SCMULT
OBS	:	Generate the coefficients of an observation equation. Calls - none
OPTION	:	Allow user changes to any or all of the default options, parameters or constants. Calls - ELSFAC
OUT	:	Print banded observation or constraint design matrix, formal covariance matrices of data points, banded weight matrix, knot times and knot time intervals. Calls - DPRINT
PREM	:	Permutation of the a posteriori covariance matrix computed using LS2 to as if it is produced by LS. Calls - none

TABLE VII-1 - cont'd

PREPAR	:	Formulate design matrices and misclosure vectors for the functionally constrained least-squares model. Calls - APRORI,CONSTR,DCLEAR,INTERV,OBS
RDATA	:	Read input data points and end knot vectors. Constructs the banded weight matrix. Driver routine for READ. Calls - CHOLD,READ
READ	:	Read in one line of input data. Modify formal covariances if requested. Calls - DABS,ERRORS,UPSTD
REJECT	:	Perform tests on the estimated residuals based on selected curve fitting rejecting criteria. Calls - CHITES
RESID	:	Compute the residual vector in least-squares adjustment with banded design matrix. Calls - none
REMAIN	:	Compute observation and constraint residuals of the functionally constrained least-squares model. Calls - DMAG
SATELV	:	Compute maximum satellite elevation. Calls - DATAN2,DCOS,DSIN,RTOP
SELKNT	:	Selection of knot generation scheme and prints number of data points in each knot interval. Calls - CHECK,SKNOT
SIGDIS	:	Evalute if the input data point falls to the right or left of the smoothed series of positions. Calls - DATAN2,DSQRT,POINT
SIMULA	:	Generate simulated data series (without any superimposed noise). Calls - DSIN
SKNOT	:	Driver routine for FIT. Calls - FIT
SMOOTH	:	Computes smoothed positions and precision estimates from the cubic coefficients and its covariance matrix. Calls - ELIPSE,DSQRT,POINT,RADD

TABLE VII-1 - cont'd

SPIN	:	Positive-definate symmetrical matrix inversion Calls - DABS,DLOG10
SOLV	:	Least-squares adjustment with a banded design matrix. Calls - BMULT1,CHOLD,COND,DMAG,RESID,SCMULT
TIMSC	:	Scaling of data point and knot times. Calls - SCMULT
TRNSFM	:	Rearrangment of design matrices produced for the functionally constrained least-squares technique to the least-squares method using the elimination of constraints model. Calls - DCLEAR
UPSTD	:	Scaling of a formal covariance matrix according to the minimum allowable semi-minor axis of error ellipse. Calls - ELIPSE,ELLSIG
VARFAC	:	Compute and performs the Chi-squared test on the a posteriori variance factor of the functionally constrained least-squares model. Calls - DABS,CHITES,BMULT1
VELO	:	Compute the smoothed velocity and associated covariance matrix from the cubic coefficients. Calls - DABS,DATAN2,DMULT,DSQRT,INTERV
VSMOTH	:	Compute a data series of smoothed velocities at input data times. Driver routine for VELO. Calls - RADD,VELO

II) Level II subroutines

Name		Description and external routines called
CHITES	:	Perform the Chi-squared statistical test on the a posteriori variance factor. Calls - MDCHI
CHOLD	:	Matrix inversion using the Choleski alogrithm. Calls - DABS,DSQRT,TRAPS

TABLE VII-1 - cont'd

COND	:	Compute the condition number of a matrix. Calls - none
CTIME	:	Convert time in days and decimals of a day to days, hours and minutes. Calls - none
DINV	:	Inversion routine for a square non-symmetric matrix. Calls - DBLE
ELIPSE	:	Compute the parameters of an error ellipse from its covariance matrix. Calls - DSQRT,DATAN2
ELLSIG	:	Compute the covariance matrix from the error ellipse. Calls - DTAN,DSQRT
ELSFAC	:	Compute the scale factor for non-standard error ellipses. Calls - MDCHI
FORM	:	Formulate the transformation matrix to convert cubic coefficients to smoothed positions. Calls - none
INTERV	:	Locate the interval in which a data point lies within the knot structure. Calls - DABS
PLOT	:	Set up plot specifications, draw axes and titles. Calls - AREA,AXS,CHRPRT,RECT
POINT	:	Compute smoothed position and associated co-variance matrix at any given time. Calls - FORM,INTERV,DMULT,TRAPS
PRNKNT	:	Print the position and slope vectors, and co-variance matrix for any specified knot time. Calls - DSQRT

TABLE VII-1 - cont'd

RADD	:	Convert radians to degrees. Calls - none
RTOP	:	Convert differences in plane coordinates to polar coordinates.

III) Level III subroutines

Name		Description and external routines called
BMULT1	:	Multiply a banded matrix with a full matrix. Calls - none
DADD	:	Add two matrices. Calls - none
DCLEAR	:	Set all elements in a matrix to zero. Calls - none
DCOPY	:	Copy a matrix. Calls - none
DMAG	:	Seek the largest or smallest element in a matrix. Calls - DABS
DMULT	:	Multiply two matrices. Calls - none
DPRINT	:	Print the elements of a matrix with a user specified title. Calls - SPACE
DSUBT	:	Subtract two matrices. Calls - none.
DTRAN	:	Transpose a matrix. Calls - none
PCOPY	:	Copy one part of a matrix into another part of a matrix. Calls - none
SPACE	:	Create blank lines. Calls - none

TABLE VII-1 - cont'd

IV) Other library routines used.

Name		Description of routines
A) FORTRAN library with plotting routines		
AREA	:	Physical plot area specification.
AXS	:	Draw plot axes.
CHRPRT	:	Draw a sequence of characters.
CIRCLE	:	Draw a circle.
DABS	:	Absolute value of a variable.
DATAN2	:	Arc tangent of an angle based on sides of a right-angle traingle.
DBLE	:	Conversion from single to double precision.
DCOS	:	Cosine of an angle.
DLOG10	:	Logarithm to the base 10.
DSIN	:	Sine of an angle.
DSQRT	:	Square root.
DTAN	:	Tangent of an angle.
ELLIPS	:	Draw an ellipse.
ENDPLT	:	Terminate plotting.
NMBR	:	Draw numbers on plot.
NOMPLT	:	Plotter pen movement commands.
PRNTCH	:	Line printer character to be used for plotting.
RECT	:	Draw a rectangle.
TRAPS	:	Error trap routine.

[Reference : Gujar, U.G. 1981 and IBM 1974]

B) IMSL library routines

MDCHI	:	Inverse Chi-squared probability density function.

[Reference : IMSL 1975]

The program is designed to be as modular as possible. This allows modifications to any of the routines to be done with relative ease. Certain routines are incorporated into the algorithm to take advantage of the uniqueness of the computations. They fall mainly into the catagory of reducing the enormous array storage requirements of the technique.

The design and weight matrices are kept as banded matrices. The design matrices ($\underline{A}_1$ and $\underline{A}_2$ - see Section 7.3.1) are stored using variable profile banding techniques. A further reduction is obtained (i.e. only one-half of the total rows of the design matrices are actually kept) by exploiting the similarities in the coefficients between adjacent rows. The weight matrix, on the other hand, uses a fixed bandwidth scheme to optimise storage. The extensive use of pointers reduces the computational cost (i.e. removing all multiplications outside the bandwidth) and allows the location of the desired elements in a matrix.

DSPLIN uses the SPIN routine for matrix inversion in the functionally constrained least-squares algorithm. This is a full matrix inversion routine. The use of a banded matrix inversion routine cuts down the processing time used in computing the inverses by a factor of b^2/n^2 ; where n is the order of the matrix and b is the bandwidth [Steeves 1974]. More studies are needed before a banded matrix inversion routine can be used. Unlike the normal least-squares adjustment, the functionally constrained least-squares algorithm requires the inversion of two matrices. The interrelationship between these inverses and the solution vector has to be investigated before a banded inverison routine can be used.

Figure VII-2 shows the processing time of DSPLIN against the number of knots used in the data span. The processing time, however, varies with the options for a particular run.

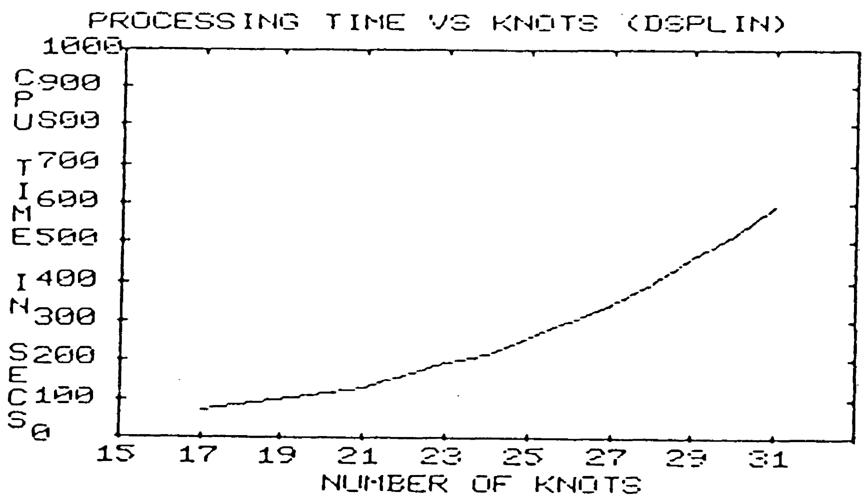


FIGURE VII-2 Processing Times of DSPLIN

Finally, given below, is the general outline of the processing sequence within the algorithm (see Figure VII-1).

1. Selection of options.

   (Routine: OPTION)

2. Input and pre-processing of data for smoothing.

   (Routines: RDATA, TIMSC and FILTER)

3. Selection of knot scheme.
   (Routine: SELKNT)
4. Formulation and printing of design matrices and misclosure vectors
   (Routines: PREPAR and OUT)
5. Computation of least-squares estimates using the functionally constrained least-squares model.
   (Routines: LS, REMAIN and VARFAC)
6. Generation of time-tagged coefficients and associated covariance matrices for each knot interval.
   (Routine: BUILD)
7. Generation of smoothed points, velocities, plotting of smoothed and raw data points or the analysis of residuals.
   (Routines: DESIGN and ANALYS)

The displays of raw and smoothed data points are made possible by several plot routines. A complete description of the possible options is included in the computer listings (i.e. in routines ANALYS and DESIGN).

Input to the program is described in the program listings and free format is used to read all variables.

```
      IMPLICIT INTEGER*4(I),REAL*8(A-H,M-Y),REAL*4(Z)                   DSPLIN
C*   #-------------------------------------------------------------#    DSPLIN
C*   #    PROGRAM         : DSPLIN                                 #    DSPLIN
C*   #    VERSION DATED : 27 TH NOVEMBER  1982                     #    DSPLIN
C*   #    AUTHOR          : QUEK SEE HEAN.    U.N.B. 1982          #    DSPLIN
C*   #    REFFERENCE      : M.SC.E THESIS 1982                     #    DSPLIN
C*   #-------------------------------------------------------------#    DSPLIN
C*                                                                      DSPLIN
C*         #####################################################        DSPLIN
C*         #                INPUT FORMAT TO PROGRAM             #       DSPLIN
C*         #                -----------------------             #       DSPLIN
C*         #                                                    #       DSPLIN
C*         #   DO YOU WISH TO CHANGE THE OPTIONS OF THE PROGRAM?#       DSPLIN
C*         #     IF YES - 1, TYPE IN ALL OPTIONS ON NEXT LINE   #       DSPLIN
C*         #                FOR DETAILED CODES SEE SUBR OPTION  #       DSPLIN
C*         #     IF NO  - 0, PROCEED TO THE *                   #       DSPLIN
C*         # * READ IN THE TITLE OF TEST DONE                   #       DSPLIN
C*         #   READ IN THE NO OF DATA SETS TO BE PROCESSED      #       DSPLIN
C*         #                                                    #       DSPLIN
C*         # 1 READ IN THE NUMBER OF KNOTS AND KNOT TYPE CODE- TKRD #   DSPLIN
C*         #     IF   TKRD - 0,EQUAL KNOT OPTION ; PROCEED TO 2 #       DSPLIN
C*         #                1,TYPE IN KNOT TIMES ON NEXT LINE   #       DSPLIN
C*         #                2,READ IN 10 OPTIONS (A-J)          #       DSPLIN
C*         #                  AND IRJET & ALPKNT(K-L) - SEE SUBR FIT #  DSPLIN
C*         #                3,PTS PER INTERVAL SAVED IN NO OF KNOTS #   DSPLIN
C*         # 2 READ IN THE NUMBER OF DATA POINTS                #       DSPLIN
C*         # 3 READ IN DATA PTS - FOR INPUT FORMAT SEE SUBR READ #      DSPLIN
C*         # 4 IF SPLIN >= 1 ; REPEAT SPLIN TIMES               #       DSPLIN
C*         #                     A)INPUTS (A-D) OF SUBR DESIGN  #       DSPLIN
C*         #                   & B)INPUTS (A-F) OF SUBR PLOT    #       DSPLIN
C*         # 5 IF IANLYS >=1 ; REPEAT IANLYS TIMES              #       DSPLIN
C*         #                     A) INPUTS (A-L) OF SUBR ANALYS #       DSPLIN
C*         #                   & B) INPUTS (A-F) OF SUBR PLOT IF PLOTS# DSPLIN
C*         #                        REQUESTED                   #       DSPLIN
C*         #----------------------------------------------------#       DSPLIN
C*         # FOR DATA SETS GREATER THAN ONE                     #       DSPLIN
C*         #   UNDER CHANGE = 0 ; REPEAT INPUTS 1, 4 AND 5      #       DSPLIN
C*         #   UNDER CHANGE = 1 ; REPEAT INPUTS 1 TO 5          #       DSPLIN
C*         # NOTE                                               #       DSPLIN
C*         # FOR DETAILS OF VARIABLES SPLIN,IANLYS AND CHANGE   #       DSPLIN
C*         #         SEE SUBR OPTION                            #       DSPLIN
C*         #####################################################        DSPLIN
C**********************************************************************DSPLIN
C*                             *** NOTE ***                           *DSPLIN
C* WARNING:                                                           *DSPLIN
C*   THE 1ST AND LAST 'DATA POINT' IN THE INPUT SEQUENCE ARE ACTUALLY *DSPLIN
C*   THE FIRST AND LAST KNOT VECTOR OF VALUES WHICH WILL DEFINE       *DSPLIN
C*   INTERVAL SIZE, AFTER EACH INPUT OF THE KNOT VECTOR, THE NEXT     *DSPLIN
```

```
C*   LINE SHOULD CONTAIN THE 1ST DERIVATIVES AND THEIR VARIANCES              *DSPLIN
C*    AND CORRELATION IN THE FORMAT 'X1 Y1 SIGX1 SIGY1 CORRXY1.               *DSPLIN
C* CURRENT SETTING:                                                           *DSPLIN
C*    MAXIMUM NUMBER OF DATA POINTS PROGRAM CAN TAKE                          *DSPLIN
C*                (DETERMINED BY ARRAY DECLARATION) = 500                     *DSPLIN
C*    MAXIMUM NUMBER OF KNOTS THAT THE PROGRAM CAN                            *DSPLIN
C*                (DETERMINED BY ARRAY DECLARATION) = 32                      *DSPLIN
C* GENERALLY:                                                                 *DSPLIN
C*    VARIABLES BEGINNING WITH 'ID' OR 'JD' DENOTES THE INTEGER               *DSPLIN
C*     DIMENSIONS OF THE RESP MATRIX.                                         *DSPLIN
C*    VARIABLES WITH  PREFIX 'PR' ARE PRINTING CODES FOR RESP MATRICES.       *DSPLIN
C* PLOTTING:                                                                  *DSPLIN
C*    PROGRAMS SUPPORTS ONE DATA SET WITH OVERLAYING PLOTS OR MULTIPLE        *DSPLIN
C*     DATA SETS AND SEPERATE PLOTS FOR EACH DATA SET. TO UPGRADE             *DSPLIN
C*     PROGRAM TO HANDLE MULTIPLE DATA SETS WITH OVERLAYING CAPABILITIES*DSPLIN
C*     MODIFY CALL ROUTINES PLOT AND ENDPLT IN MAIN, SUBR ANALYS AND          *DSPLIN
C*     SUBR DESIGN.                                                           *DSPLIN
C*                                                                            *DSPLIN
C*********************************************************************DSPLIN
C  VARIABLE NAMES : DESCRIPTION                                                DSPLIN
C                                                                              DSPLIN
C   M     :  NUMBER OF UNKNOWNS                                                DSPLIN
C   N     :  NUMBER OF OBSERVATION EQUATIONS                                   DSPLIN
C   P     :  FIXED BANDWIDTH WEIGHT MATRIX                                     DSPLIN
C   R     :  NUMBER OF CONSTRAINT EQUATIONS IN SECONDARY MODEL                 DSPLIN
C   V     :  RESIDUAL VECTOR OF OBSERVABLES                                    DSPLIN
C   X     :  LS SOLUTION VECTOR                                                DSPLIN
C   A1    :  VARIABLE PROFILE BANDWIDTH OBSERVATION (PRIMARY) DESIGN MATRIDSPLIN
C   A2    :  VARIABLE PROFILE BANDWIDTH CONSTRAINT (SECONDARY)DESIGN MATRIDSPLIN
C   NO    :  NUMBER OF DATA POINTS                                             DSPLIN
C   SC    :  INVERSE OF THE SCALE FACTOR FOR THE TIME DOMAIN                   DSPLIN
C   SO    :  APOSTERIORI VARIANCE FACTOR                                       DSPLIN
C   TK    :  VECTOR OF KNOT TIMES                                              DSPLIN
C   TX    :  VECTOR OF DATA TIMES                                              DSPLIN
C   VX    :  COVARIANCE MATRIX OF LS ESTIMATES                                 DSPLIN
C   V2    :  RESIDUAL VECTOR OF CONSTRAINTS                                    DSPLIN
C   W1    :  OBSERVATION MISCLOSURE VECTOR                                     DSPLIN
C   W2    :  CONSTRAINT MISCLOSURE VECTOR                                      DSPLIN
C   XO    :  APRIORI SOLUTION VECTOR                                           DSPLIN
C   COF   :  MATRIX OF  COEFFICIENTS DEFINING CUBICS                           DSPLIN
C   DTK   :  VECTOR CONTAINING KNOT TIME INTERVALS                             DSPLIN
C   ESC   :  SCALE FACTOR AT ALPHA PERCENT CONFIDENCE INTERVAL                 DSPLIN
C   IDP   :  DIM DEC  FOR P MATRIX                                             DSPLIN
C   IDV   :  DIM DEC  FOR V VECTOR                                             DSPLIN
C   IDX   :  DIM DEC  FOR X VECTOR                                             DSPLIN
C   IER   :  ERROR CODE USUALLY ASSOCIATED WITH INVERSES                       DSPLIN
C   IRD   :  READ UNIT                                                         DSPLIN
C   IWR   :  WRITE UNIT                                                        DSPLIN
```

```
C   KNO   :  NUMBER OF KNOTS                                                  DSPLIN
C   VAR   :  HYPER MATRIX OF INPUT DATA POINT COVARIANCE MATRICES             DSPLIN
C   ACON   :  WORK ARRAY OF SUBR CONSTR                                       DSPLIN
C   ALPH   :  LEVEL OF SIGNIFICANCE TESTS - SEE SUBR FILTER                   DSPLIN
C   AOBS   :  WORK ARRAY OF SUBR OBS                                          DSPLIN
C   DINC   :  INTERVAL INCREMENT - SEE SUBR FILTER                            DSPLIN
C   DINT   :  INTERVAL WIDTH - SEE SUBR FILTER                                DSPLIN
C   DRAW   :  SPLINE DRAWING - SEE SUBR DESIGN                                DSPLIN
C   ECHO   :  INPUT ECHO OPTION                                               DSPLIN
C   FESC   :  SCALE FACTOR RELATED TO VARIABLE ALPH                           DSPLIN
C   GINT   :  INTERVAL BETWEEN SMOOTHED PARAMETERS - SEE SUBR ANALYS          DSPLIN
C   ICOL   :  FUNCTION - COLUMN POINTERS FOR A1 AND A2 MATRICES               DSPLIN
C   IDA1   :  ROW DIM DEC  OF A1 MATRIX                                       DSPLIN
C   IDA2   :  ROW DIM DEC  OF A2 MATRIX                                       DSPLIN
C   IDTK   :  DIM DEC  FOR TK VECTOR                                          DSPLIN
C   IDTX   :  DIM DEC  FOR TX VECTOR                                          DSPLIN
C   IDVX   :  DIM DEC  FOR VX MATRIX                                          DSPLIN
C   IDV2   :  DIM DEC  FOR V2 VECTOR                                          DSPLIN
C   IDW1   :  DIM DEC  FOR W1 VECTOR                                          DSPLIN
C   IDW2   :  DIM DEC  FOR W2 VECTOR                                          DSPLIN
C   IDXO   :  DIM DEC  FOR XO VECTOR                                          DSPLIN
C   IFLT   :  CALL CODE FOR FILTER ROUTINE - SEE SUBR OPTION                  DSPLIN
C   IROW   :  FUNCTION - ROW POINTERS FOR A1 AND A2 MATRICES                  DSPLIN
C   ISCT   :  CALL CODE FOR TIME SCALING - SEE SUBR OPTION                    DSPLIN
C  LOBS    :  VECTOR CONTAINING ALL OBSERVABLES                               DSPLIN
C   PRTK   :  PRINT CODE FOR TK VECTOR                                        DSPLIN
C   PRTX   :  PRINT CODE FOR TX VECTOR                                        DSPLIN
C   TKRD   :  CODE ON KNOT TYPE                                               DSPLIN
C   VCOF   :  COVARIANCE MATRIX FOR COEFFICIENTS (COF)                        DSPLIN
C   VTWV   :  QUADRATIC  NORM OF RESIDUALS                                    DSPLIN
C   XINT   :  PLOTTING SPLINE IN XINT INCREMENTS - SEE SUBR DESIGN            DSPLIN
C   ADDA1   :  ADDRESS SEQUENCE FOR A1 MATRIX                                 DSPLIN
C   ADDA2   :  ADDRESS SEQUENCE FOR A2 MATRIX                                 DSPLIN
C   ALPHA   :  PERCENTAGE CONFIDENCE LEVEL OF DRAWN ELLIPSES                  DSPLIN
C   DEGOF   :  DEGREES OF FREEDOM IN CURRENT MODEL                            DSPLIN
C   GDATE   :  SUBR - GETS DATE FOR RUN  ( FORTRAN LIBRARY )                  DSPLIN
C   IAPRI   :  CALL CODE ON SUBR APRORI- SEE SUBR APRIORI                     DSPLIN
C   IDCOF   :  ROW DIM DEC  FOR COF MATRIX                                    DSPLIN
C   IDDTK   :  DIM DEC  FOR DTK VECTOR                                        DSPLIN
C   IDVAR   :  ROW DIM DEC  FOR VAR MATRIX                                    DSPLIN
C   IPLOT   :  PLOTTING OPTION CODE - SEE SUBR OPTION                         DSPLIN
C   IREAD   :  READ FORMAT CODE - SEE SUBR READ                               DSPLIN
C   IRJET   :  TOTAL NO OF FAILURES B4 REJECTION - SEE SUBR REJECT            DSPLIN
C   ITYPE   :  PRINTER PLOT OR GRAPHIC  PLOT                                  DSPLIN
C   NINTY   :  90 DEG IN RADIANS                                              DSPLIN
C   PRCOV   :  PRINT CODE FOR INPUT COVARIANCES OF DATA POINTS                DSPLIN
C   PRDTK   :  PRINT CODE FOR KNOT INTERVAL TIMES                             DSPLIN
C   PRGEN   :  PRINT CODE FOR SMOOTHED POINTS USED IN DRAWING SPLINE          DSPLIN
```

```
C   PRKNT   :   SELECTIVE PRINTING OF KNOT INFORMATION                           DSPLIN
C   PRPTS   :   PLOTTING CODE CN DATA POINTS                                     DSPLIN
C   SPLIN   :   CALL CODE ON SUBR DESIGN                                         DSPLIN
C   ALPKNT  :   LEVEL OF SIGNIFICANCE TO BE USED IN REJECT ROUTINE               DSPLIN
C   CALPHA  :   PERCENTAGE LEVEL OF SIGNIFICANCE TO BE USED IN SUDR VARFAC DSPLIN
C   CHANGE  :   DIFFERENT DATA SETS - SEE SUBR OPTION                           DSPLIN
C   FORFIV  :   45 DEGREES IN RADIANS                                            DSPLIN
C   GSTART  :   TIME TO BEGIN GENERATION OF SMOOTH PARAMETERS - SUBR ANALYSDSPLIN
C   IANLYS  :   CALL CODE FOR ANALYS SUBROUTINE                                  DSPLIN
C   ICANYS  :   OPTION VECTCR CF CODES FOR ANALYS RCUTINE                        DSPLIN
C   ICHECK  :   CODE FOR CHECK ROUTINE - SEE CHECK SUBROUTINE                    DSPLIN
C   ICKNOT  :   CPTION CODE ON KNOT GENERATION ROUTINE - SEE SKNOT ROUTINE DSPLIN
C   ICOUNT  :   NUMBER OF PROCESSED DATA SETS                                    DSPLIN
C   IDKNOT  :   MAXIMUM NUMBER OF KNOTS THE PROGRAM CAN HANDLE                   DSPLIN
C   IDLOBS  :   DIM DEC  FCR LCBS VECTOR                                         DSPLIN
C   IDVCOF  :   ROW DIM DEC  FOR VCOF MATRIX                                     DSPLIN
C   IELIPS  :   CODE ON PLOTTED ELLIPSES - SEE OPTION SUBR                       DSPLIN
C   IOBSNO  :   MAX NUMBER OF OBSERVED SATA POINTS THE PROGRAM CAN MANAGE    DSPLIN
C   ITOTAL  :   TOTAL NUMBER OF DATA SETS TO BE PROCESSED                        DSPLIN
C   PRAMAT  :   PRINT CODE FOR A1 AND A2 MATRICES                                DSPLIN
C   PRCOFF  :   PRINT CODE FOR COEFFICIENTS OF SPLINE                            DSPLIN
C   PRCOFV  :   PRINT CODE FOR COVARIANCES CF CCEFFICIENTS                       DSPLIN
C   PRKNOT  :   PLOTTIG CODE ON KNOTS - SEE DESIGN SUBR                          DSPLIN
C   PRLOBS  :   PRINT CODE CN LCBS ROUTINE                                       DSPLIN
C   PRPMAT  :   PRINT CODE CN WEIGHT MATRIX                                      DSPLIN
C   PRRESD  :   PRITN CODE ON RESIDUALS                                          DSPLIN
C   PRVARX  :   PRINT CODE ON COVARIANCE OF LS ESTIMATES                         DSPLIN
C   PRXMAT  :   PRINT CODE ON LS ESTIMATES - X VECTCR                            DSPLIN
C   SCTIME  :   SCALE FACTOR ON TIME DOMAIN                                      DSPLIN
C   XSTART  :   TIME TO BEGIN DRAWING SPLINE,CO OR SPD - SUBR DESIGN/ANALYSDSPLIN
C                                                                                DSPLIN
C CALLS    - GDATE,CPUTIM      (EXTERNAL ROUTINES)                               DSPLIN
C            OPTION,DCLEAR,RCATA,TIMSC,FILTER,SELKNT,PREPAR,OUT,                 DSPLIN
C            LS,VARFAC,SCMULT,DADD,DPRINT,PRNKNT,BUILD,DESIGN,ANALYS             DSPLIN
C                                                                                DSPLIN
C                                                                                DSPLIN
C*PROGRAM BEGINS                                                                 DSPLIN
C* COMMON BLK CONTAINING DECLARED MATRIX DIMENSIONS                              DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV               DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,               DSPLIN
     @       IDCOF,IDVCOF                                                        DSPLIN
C* DIMENSION BLK CONTAINING  ARRAY DECLARATIONS                                  DSPLIN
      REAL*8 A1(504,7),W1(1008,1),TX(502,1),P(1008,3),PV(1008,1),                DSPLIN
     .       LOBS(1008,1),VAR(502,2,2),V(1008,1)                                 DSPLIN
      REAL*8 A2(30,11),TK(32,1),DTK(31,1),X(128,1),                              DSPLIN
     .   VX(128,128),W2(60,1),V2(60,1),XO(128,1),COF(31,8),VCOF(31,8,8)  DSPLIN
      REAL*8 AOBS(7,1),ACON(11,1),MAT(2,2),VTWV,                                 DSPLIN
     .       CX(8,1),VCX(8,8),CO(8,1),VCO(8,8)                                   DSPLIN
```

```
C* WORK ARRAY DIMENSIONS WK1(IDX,IDA2*2) AND WK2(IDA2*2,IDA2*2)                 DSPLIN
      REAL*8 WK1(128,60),WK2(60,60)                                             DSPLIN
C* WORK ARRAYS                                                                  DSPLIN
C* AS DIMENSIONED IN FIT ROUTINE ;SEE FIT RCUTINE                                DSPLIN
      REAL*8 FWK1(102,7),FWK2(204,3),FWK3(100,2,2),FWK4(8,8),FWK5(8,8),          DSPLIN
     @      FWK6(200,1),FWK7(8,204)                                             DSPLIN
C* AS DIMENSIONED IN FILTER RCUTINE                                             DSPLIN
      REAL*8 FLWK1(20,3),FLWK2(40,3),FLWK3(40,1),FLWK4(40,1),FLWK5(4,4)          DSPLIN
C* AS DIMENSIONED IN ANALYS ROUTINE                                             DSPLIN
      REAL*8 AWK1(500,1),AWK2(500,1),AWK3(500,1),AWK4(500,1),AWK5(500,1)DSPLIN
C*                                                                              DSPLIN
      INTEGER*4 NC,KNO,R,M,N,L,K,SPLIN,DRAW,                                    DSPLIN
     @          ADDA1(504),ADDA2(30),ICANYS(10),ICKNOT(10)                      DSPLIN
      INTEGER*4 PRTX,PRTK,PRDTK,PRLOBS,PRCOV,PRCOFF,PRCOFV,ECHO,                DSPLIN
     @     PRPTS,PRKNCT,PRAMAT,PRPMAT,PRXMAT,PRVARX,PRRESD,PRGEN,               DSPLIN
     @     CHANGE,IAPRI,TKRD,PRKNT,DEGOF                                        DSPLIN
C*                                                                              DSPLIN
      LOGICAL*1 ITITLE(80),DATE(18),TIME(6)                                     DSPLIN
C* SPACE ECOMCNIZING VIA EQUIVALENCE STATEMENTS                                  DSPLIN
      EQUIVALENCE (A2(1),VCOF(1)),                                              DSPLIN
     @            (V(1),W1(1)),                                                 DSPLIN
     @            (V2(1),W2(1))                                                 DSPLIN
      EQUIVALENCE (A1(1),CX(1)),                                                DSPLIN
     @            (A1(9),VCX(1)),                                               DSPLIN
     @            (A1(73),MAT(1)),                                              DSPLIN
     @            (A1(77),CO(1)),                                               DSPLIN
     @            (A1(85),VCO(1))                                               DSPLIN
      EQUIVALENCE (A1(149),VTWV),                                               DSPLIN
     @            (A1(150),XX),                                                 DSPLIN
     @            (A1(151),YY),                                                 DSPLIN
     @            (A1(152),CXY)                                                 DSPLIN
      EQUIVALENCE (WK1(1),DX1),                                                 DSPLIN
     @            (WK1(2),DX2),                                                 DSPLIN
     @            (WK1(3),TUME),                                                DSPLIN
     @            (WK1(4),DD)                                                   DSPLIN
      EQUIVALENCE (WK1(5),XLR),                                                 DSPLIN
     @            (WK1(6),SO),                                                  DSPLIN
     @            (WK1(7),AOBS(1)),                                             DSPLIN
     @            (WK1(15),PV(1))                                               DSPLIN
      EQUIVALENCE (WK1(1100),FWK1(1)),                                          DSPLIN
     @            (WK1(1815),FWK2(1)),                                          DSPLIN
     @            (WK1(2428),FWK3(1)),                                          DSPLIN
     @            (WK1(2829),FWK4(1)),                                          DSPLIN
     @            (WK1(2894),FWK5(1)),                                          DSPLIN
     @            (WK1(2959),FWK6(1)),                                          DSPLIN
     @            (WK1(3160),FWK7(1))                                           DSPLIN
      EQUIVALENCE (WK1(4800),FLWK1(1)),                                         DSPLIN
     @            (WK1(4861),FLWK2(1)),                                         DSPLIN
```

```
     @                (WK1(4982),FLWK3(1)),                               DSPLIN
     @                (WK1(5023),FLWK4(1)),                               DSPLIN
     @                (WK1(5064),FLWK5(1))                                DSPLIN
      EQUIVALENCE (WK1(5100),AWK1(1)),                                    DSPLIN
     @                (WK1(5601),AWK2(1)),                                DSPLIN
     @                (WK1(6102),AWK3(1)),                                DSPLIN
     @                (WK1(6603),AWK4(1)),                                DSPLIN
     @                (WK1(7104),AWK5(1))                                 DSPLIN
      EQUIVALENCE (WK2(1),VCOF(331))                                      DSPLIN
C* DATA ASSIGMNENT                                                        DSPLIN
      DATA ITITLE/80*' '/,ACON/11*0.0D00/,ICANYS/10*0/                    DSPLIN
      DATA FORFIV/0.785398164D0/,NINTY/1.570796327D0/                     DSPLIN
C* FUNCTIONS                                                              DSPLIN
       IROW(I) = (I+1)/2                                                  DSPLIN
       ICOL(I) = (2 - ((I+1)/2 - (I/2)))                                  DSPLIN
C* READ UNIT                                                              DSPLIN
      IRD = 5                                                             DSPLIN
C* WRITE UNIT                                                             DSPLIN
      IWR = 6                                                             DSPLIN
C* EQUALITIES GOVERNING THE MATRIX SIZES                                  DSPLIN
C* MAXIMUM NUMBER OF OBSERVATIONS THAT CAN BE TAKEN INTO THE PROGRAM      DSPLIN
      IOBSNO = 500                                                        DSPLIN
C* MATRIX SIZES ARE AS FOLLOWS                                            DSPLIN
      IDA1 = IOBSNO   + 4                                                 DSPLIN
      IDW1 = IOBSNO*2 + 8                                                 DSPLIN
      IDTX = IOBSNO   + 2                                                 DSPLIN
      IDP  = IOBSNO*2 + 8                                                 DSPLIN
      IDPV = IOBSNO*2 + 8                                                 DSPLIN
      IDLOBS=IOBSNO*2 + 8                                                 DSPLIN
      IDVAR= IOBSNO   + 2                                                 DSPLIN
      IDV  = IOBSNO*2 + 8                                                 DSPLIN
C* MAXIMUM NUMBER OF KNOTS THAT CAN BE TAKEN INTO THE PROGRAM             DSPLIN
      IDKNOT = 32                                                         DSPLIN
C* MATRIX SIZES ARE AS FOLLOWS                                            DSPLIN
      IDA2  = IDKNOT   - 2                                                DSPLIN
      IDTK  = IDKNOT                                                      DSPLIN
      IDDTK = IDKNOT   - 1                                                DSPLIN
      IDX   = IDKNOT*4                                                    DSPLIN
      IDVX  = IDKNOT*4                                                    DSPLIN
      IDW2  = IDKNOT*2 - 4                                                DSPLIN
      IDV2  = IDKNOT*2 - 4                                                DSPLIN
      IDXO  = IDKNOT*4                                                    DSPLIN
      IDCOF = IDKNOT   - 1                                                DSPLIN
      IDVCOF= IDKNOT   - 1                                                DSPLIN
C* ADDRESS SEQUENCE OF A1 AND A2 MATRICES FOLLOW THEIR RESP MATRICES      DSPLIN
C  *** SUBR LS1FCF02 REQS MODIFICATION TO DIM IF NO. OF KNOT CHANGES ***DSPLIN
C* PRINTS THE DATE AND TIME OF RUN                                        DSPLIN
      CALL GDATE(DATE,TIME)                                               DSPLIN
```

```
      WRITE(IWR,9040)DATE,TIME
C* READ IN TITLE OF JOB : MAX 80 CHARACTERS
      READ(IRD,9020)(ITITLE(I),I = 1,80)
      WRITE(IWR,9030)(ITITLE(I),I = 1,80)
C* OPTION CHANGE
      CALL OPTION(CHANGE,IREAD,IFLT,IAPRI,ISCT,IELIPS,ECHO,
     @           PRCCV,PRPMAT,PRTX,PRTK,PRDTK,PRLOBS,ICHECK,
     @           PRAMAT,PRXMAT,PRVARX,PRKNT,PRRESD,PRCOFF,PRCOFV,
     @           ITYPE,SPLIN,PRPTS,PRKNOT,DRAW,PRGEN,IANLYS,IPLOT,
     @           ICANYS,SCTIME,DINT,DINC,ALPH,ALPHA,CALPHA,
     @           FESC,ESC,XSTART,XINT,GSTART,GINT)
C* RESET PROCESSING CPU TIME COUNTER FOR NEW DATA SET
      CALL CPUTIM(IELAPS,IREM)
      ZELAPS = IELAPS/10.0
      ZREM = IREM/10.0
      WRITE(IWR,9050)ZELAPS,ZREM
C*
C* READ IN THE NUMBER OF DATA SETS
      READ(IRD,*)ITOTAL
C* RESET DATA SET COUNTER
      ICOUNT = 0
 8000 READ(IRD,*)KNO,TKRD
C* READ IN THE TKNOT TIMES
      IF(TKRD.NE.1)GO TO 6
      READ(IRD,*)(TK(I,1),I = 1,KNO)
      IF(ECHO.EQ.1)WRITE(IWR,7520)(TK(I,1),I = 1,KNO)
C* ALTERNATIVE TO READING KNOT TIMES
    6 IF(TKRD.NE.2)GO TO 7
      READ(IRD,*)(ICKNOT(I),I=1,10)
      READ(IRD,*)IRJET,ALPKNT
      IF(ECHO.EQ.1)WRITE(IWR,7530)(ICKNOT(I),I=1,10)
      IF(ECHO.EQ.1)WRITE(IWR,7550)IRJET,ALPKNT
C* READS DATA SET ONCE ONLY IF IT REMAINS THE SAME.
    7 IF(CHANGE.EQ.0.AND.ICOUNT.EQ.0)READ(IRD,*)NO
      IF(CHANGE.EQ.1)READ(IRD,*)NO
      ICOUNT = ICOUNT + 1
      IF(ICOUNT.GT.1)WRITE(IWR,9060)ICOUNT
      IF(ECHO.EQ.1)WRITE(IWR,7510)KNO,NO
C* AVOIDING THE REFORMULATION AND READING IN OF THE DATA
C* SET. READS ONLY DURING THE FIRST RUN
      IF(CHANGE.EQ.0.AND.ICOUNT.GT.1)GO TO 13
      CALL DCLEAR(LOBS,IDLOBS,1)
      CALL RDATA(NO,TX,VAR,LOBS,P,MAT,
     @           IREAD,ECHO,XX,YY,CXY,TUME)
C* TIME SCALING
   13 CALL TIMSC(KNO,NO,ISCT,SCTIME,TX,TK,IDTX,IDTK,TKRD)
C* LINEAR FILTER IMPLEMENTATION
      IF(IFLT.EQ.1)
```

```
     @        CALL   FILTER(NO,TX,LOBS,P,VAR,DINT,DINC,ALPH,FESC,SCTIME,    DSPLIN
     @                     FLWK1,FLWK2,FLWK3,FLWK4,FLWK5,FWK6)                DSPLIN
C* FORMULATION OF KNOTS                                                      DSPLIN
      CALL SELKNT(TKRD,KNO,NO,ICHECK,TX,TK,LOBS,VAR,ICKNOT,                  DSPLIN
     @      IRJET,ALPKNT,DTK,DD,FWK1,FWK2,FWK3,FWK4,FWK5,FWK6,FWK7)          DSPLIN
C* SET UP MATRICES FOR LS SOLUTION                                           DSPLIN
      CALL PREPAR(KNO,NO,R,M,N, TX,TK,DTK,AOBS,ACON,                         DSPLIN
     @               A1,ADDA1,A2,ADDA2,                                      DSPLIN
     @               XO,LOBS,W1,W2,IAPRI)                                    DSPLIN
C* PROGRAM OUTPUT SECTION PRIOR TO LS SOLUTION                               DSPLIN
      CALL OUT(KNO,NO,R,M,N,                                                 DSPLIN
     @            PRTX,PRTK,PRLOBS,PRDTK,PRCOV,PRPMAT,PRAMAT,                DSPLIN
     @            TX,TK,LOBS,DTK,VAR,P,A1,A2,ADDA1,ADDA2)                    DSPLIN
C* LEAST SQUARES PROCESS                                                     DSPLIN
      CALL LS(A1,ADDA1,A2,ADDA2,P,W1,W2,X,VX,R,N,M,                          DSPLIN
     @           IDA1,IDA2,IDP,IDW1,IDW2,IDX,IER,WK1,WK2)                    DSPLIN
C* TERMINATE DATA SET PROCESSING IF INVERSE FAILS                            DSPLIN
      IF(IER.NE.0)GO TO 7000                                                 DSPLIN
C* COMPUTE RESIDUALS                                                         DSPLIN
      CALL REMAIN(R,M,N,A1,A2,ADDA1,ADDA2,W1,W2, X,V,V2)                     DSPLIN
C* COMPUTE VARIANCE FACTOR AND DEGREES OF FREEDOM                            DSPLIN
      CALL VARFAC(SO,DEGOF,NO,R,M,N,CALPHA,TX,P,V,PV,VTWV,DX1,DX2)           DSPLIN
C* SCALING OF THE COVARINACE MATRIX = NOTICE THE APOSTERIORI VAR USED.       DSPLIN
      CALL SCMULT(SO,VX,VX,M,M,IDVX,IDVX)                                    DSPLIN
C* FORMS FINAL LEAST SQUARES ESTIMATES                                       DSPLIN
      CALL CADD(XO,X,M,1,X,IDX,1)                                            DSPLIN
C* RESULTS OUTPUT SECTION                                                    DSPLIN
      IF(PRXMAT.EQ.1)CALL DPRINT(X,M,1,IDX,1,'FINAL EST X-VECTOR',9)         DSPLIN
      IF(PRVARX.EQ.1)CALL DPRINT(VX,M,M,IDVX,IDVX,'COVARIANCE AF SC',8)      DSPLIN
C* PRINT END SLOPE AND KNOT INFORMATION                                      DSPLIN
      IF(PRVARX.NE.1.AND.PRKNT.NE.0)                                         DSPLIN
     @                   CALL PRNKNT(X,M,IDX,VX,IDVX,PRKNT)                  DSPLIN
      IF(PRRESD.EQ.1)CALL DPRINT(V,N,1,IDV,1,'OBS VECTOR RESIDUALS',10)      DSPLIN
C* TRANSFORM KNOT VECTORS TO CUBIC COEFFICIENTS                              DSPLIN
      CALL BUILD(KNO,X,VX,COF,VCOF,DTK,                                      DSPLIN
     @               CX,VCX,CO,VCO,PRCOFV,PRCOFF)                            DSPLIN
C* PLOTTING OF KNOTS,POINTS AND SPLINE                                       DSPLIN
C*    CONFIDENCE ELLIPSE SET AT THE ALPHA PERCENTAGE LEVEL                   DSPLIN
 7000 IF(IELIPS.EQ.1)WRITE(IWR,7540)ALPHA,ESC                                DSPLIN
      IF(IELIPS.EQ.0)WRITE(IWR,7570)                                         DSPLIN
      IF(SPLIN.EQ.0)GO TO 8020                                               DSPLIN
      DO 7010 I = 1,SPLIN                                                    DSPLIN
      CALL DESIGN(NO,KNO,ICOUNT,ITYPE,SPLIN,PRGEN,PRPTS,PRKNOT,              DSPLIN
     @                            DRAW,IER,ESC,SCTIME,                       DSPLIN
     @                            XSTART,XINT,FORFIV,NINTY,                  DSPLIN
     @                            TK,TX,COF,VCOF,DTK,VAR,LOBS)               DSPLIN
      IF(SPLIN.GT.1.AND.I.NE.SPLIN)READ(IRD,*)PRPTS,PRKNOT,DRAW,PRGEN        DSPLIN
      IF(SPLIN.GT.1.AND.I.NE.SPLIN.AND.DRAW.NE.0)READ(5,*)XSTART,XINT        DSPLIN
```

```
      WRITE(IWR,9070)                                                   DSPLIN
 7010 CONTINUE                                                          DSPLIN
C* ANALYSIS OF RESIDUALS VIA PRINTED NUMBERS OR PLOTS                   DSPLIN
 8020 IF(IANLYS.EQ.0)GO TO 8030                                         DSPLIN
      DO 8040 I = 1,IANLYS                                              DSPLIN
      CALL ANALYS(IANLYS,ESC,ICANYS,KNO,NO,SCTIME,XSTART,XINT,GSTART,   DSPLIN
     &     GINT,SO,DEGOF,LOBS,VAR,TX,TK,COF,VCOF,V,ICOUNT,IPLOT,IER,    DSPLIN
     &     AWK1,AWK2,AWK3,AWK4,AWK5)                                    DSPLIN
      IF(IANLYS.EQ.0.OR.I.EQ.IANLYS)GO TO 8025                          DSPLIN
            READ(IRD,*)(ICANYS(II),II=1,10)                             DSPLIN
            IF(ECHO.EQ.1)WRITE(IWR,7530)(ICANYS(II),II=1,10)            DSPLIN
            IF(ICANYS(8).EQ.1)READ(IRD,*)GSTART,GINT                    DSPLIN
            IF(ECHO.EQ.1.AND.ICANYS(8).EQ.1)WRITE(IWR,7520)GSTART,GINT  DSPLIN
            IF(ICANYS(9).NE.0)READ(IRD,*)XSTART,XINT                    DSPLIN
            IF(ECHO.EQ.1.AND.ICANYS(9).NE.0)WRITE(IWR,7520)XSTART,XINT  DSPLIN
 8025 WRITE(IWR,9070)                                                   DSPLIN
 8040 CONTINUE                                                          DSPLIN
C* EVALUATE CPU TIME USED AND REMAINING                                 DSPLIN
 8030 CALL CPUTIM(I,J)                                                  DSPLIN
      ZELAPS = I/10.0                                                   DSPLIN
      ZREM = J/10.0                                                     DSPLIN
      WRITE(IWR,9050)ZELAPS,ZREM                                        DSPLIN
      IF(ICOUNT.EQ.ITOTAL)GO TO 9999                                    DSPLIN
C* CORRECT THE SCALED TX VECTOR IN ORDER TO TAKE IN NEW SCTIME          DSPLIN
      SC = 1.0D00/SCTIME                                                DSPLIN
      L = NO + 2                                                        DSPLIN
      CALL SCMULT(SC,TX,TX,L,1,IDTX,1)                                  DSPLIN
      GO TO 8000                                                        DSPLIN
C* FORMAT BLOCK                                                         DSPLIN
 7510 FORMAT(///,3X,'ECHO : ',I4,I4)                                    DSPLIN
 7520 FORMAT(/,3X,'ECHO : ',6(1PD12.5,3X))                              DSPLIN
 7530 FORMAT(/,3X,'OPTION LIST ',100(35I3,/,15X))                       DSPLIN
 7540 FORMAT(/,3X,'SCALE FACTOR FOR ',F6.1,' PERCENT CONFIDENCE',       DSPLIN
     @ ' ELLIPSES = ',F10.4,/)                                          DSPLIN
 7550 FORMAT(/3X,'ECHO : ',I4,2X,F6.1)                                  DSPLIN
 7570 FORMAT(/,3X,'STANDARD ELLIPSE REQUESTED',/)                       DSPLIN
C*                                                                      DSPLIN
 9020 FORMAT(100A1)                                                     DSPLIN
 9030 FORMAT(//,3X,100A1,//)                                            DSPLIN
 9040 FORMAT(1H1,//,3X,'DATE OF RUN : ',18A1,5X,'TIME : ',2A1,          DSPLIN
     @          2(' : ',2A1))                                           DSPLIN
 9050 FORMAT(//,'    ELAPSED TIME : ',F9.2,'  TIME LEFT : ',F9.2,       DSPLIN
     @ ' MILLISECONDS')                                                 DSPLIN
 9060 FORMAT(1H1,////,' DATA SET NUMBER ',I4,/,1X,20('='),///)          DSPLIN
 9070 FORMAT(1H1)                                                       DSPLIN
 9999 WRITE(IWR,9070)                                                   DSPLIN
      IF((IPLOT.EQ.1.AND.IANLYS.EQ.1).OR.SPLIN.EQ.1)CALL ENDPLT         DSPLIN
      STOP                                                              DSPLIN
```

```
      END                                                               DSPLIN
      SUBROUTINE ANALYS(IANLYS,ESC,ICANYS,KNO,NO,SCTIME,XSTART,XINT,    DSPLIN
     @          GSTART,GINT,SO,DEGOF,LOBS,                              DSPLIN
     @          VAR,TX,TK,COF,VCOF,V,IC,IPLOT,IER,TIM,DIST,XX,YY,SDIST) DSPLIN
      IMPLICIT REAL*8(A-H,O-Y),REAL*4(Z)                                DSPLIN
C ***************************************************************      DSPLIN
C * VERSION     : 23 SEPTEMBER 1982                               *     DSPLIN
C * DESCRIPTION : ANALYSIS AND DISPLAY OF RESIDUALS               *     DSPLIN
C ***************************************************************      DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  10 OPTIONS CONTAINED IN ICANYS CONTROL THE PERFORMANCE               DSPLIN
C  OF THIS SUBROUTINE.                                                  DSPLIN
C  ICANYS(10) :                                                         DSPLIN
C       A) ICANYS(1) - BREAKDOWN OF RESIDUALS - ESSENTIAL FOR ICANYS(2) DSPLIN
C                                               AND ICANLYS(5)          DSPLIN
C       B) ICANYS(2) - PLOTTING OF RESIDUALS                            DSPLIN
C                          0 - NO PLOT DESIRED                          DSPLIN
C                          1 - RESIDUALS ONLY                           DSPLIN
C                          2 - INCLUDED SMOOTHED STANDARD DEVIATIONS    DSPLIN
C                          3 - INCLUDED ALSO RAW STANDARD DEVIATIONS    DSPLIN
C       C) ICANYS(3) - COMPUTE AND PRINT SMOOTH DATA POINTS             DSPLIN
C       D) ICANYS(4) - COMPUTE AND PRINT SMOOTH VELOCITIES AT DATA TIMES DSPLIN
C       E) ICANYS(5) - PLOT OF ______ RESIDUALS VS TIME DESIRED.        DSPLIN
C                          1 - ABSOLUTE DISTANCE                        DSPLIN
C                          2 - EASTING                                  DSPLIN
C                          3 - NORTHING                                 DSPLIN
C                          4 - SIGNED DISTANCE                          DSPLIN
C                          5 - APP. STD RESIDUALS IN EASTINGS           DSPLIN
C                          6 - APP. STD RESIDUALS IN NORTHINGS          DSPLIN
C       F) ICANYS(6) - ACTIVATE SIMULATED DATA COMPARISON               DSPLIN
C       G) ICANYS(7) - PRINT PLOTTING VALUES OF OPTION (E)              DSPLIN
C       H) ICANYS(8) - LOREX PRINT FORMAT                               DSPLIN
C       I) ICANYS(9) - PLOT _______ VS TIME                             DSPLIN
C                           1 - COURSE                                  DSPLIN
C                           2 - SPEED                                   DSPLIN
C       J) ICANYS(10)- NOT USED                                         DSPLIN
C                                                                       DSPLIN
C  IF   ICANYS(8) = 1  THEN INPUT                                       DSPLIN
C       K) GSTART    - TIME TO BEGIN SMOOTHED VALUES                    DSPLIN
C       L) GINT      - INCREMENT BETWEEN SMOOTHED PARAMETERS            DSPLIN
C  FOR FIRST CALL TO ANALYS SUBROUTINE                                  DSPLIN
C        INPUT OF VARIABLES XSTART AND XINT IS VIA THE OPTION SUBR      DSPLIN
C  FOR THE SECOND AND SUBSEQ CALLS TO ANALYS SUBROUTINE,I.E. IANLYS >1  DSPLIN
C        IF ICANYS(9) = 1 OR = 2  INPUT:                                DSPLIN
C             M) XSTART - TIME TO START SPEED/COURSE PLOT               DSPLIN
C             N) XINT   - PLOTTING INCREMENTS                           DSPLIN
C  IC     - CURRENTLY PROCESSING DATA SET NUMBERED 'IC'                 DSPLIN
C  IANLYS - NO OF TIMES THE ROUTINE IS TO CALL IN THIS DATA SET         DSPLIN
```

```
C  ESC,KNO,NO,SCTIME,SO,DEGOF,LOBS,VAR,TX,TK,COF,                   DSPLIN
C  VCOF,V,IPLOT,IER                                                 DSPLIN
C           - SEE MAIN FOR DESCRIPTION                              DSPLIN
C                                                                   DSPLIN
C OUTPUT PARAMETERS                                                 DSPLIN
C  PROGRAM NOT DESIGN TO RETURN ANY PARAMETERS                      DSPLIN
C                                                                   DSPLIN
C WORK ARRAYS                                                       DSPLIN
C     TIM,DIST,XX,YY,SDIST                                          DSPLIN
C                                                                   DSPLIN
C CALLS - PRNTCH,NOWPLT,DSHLNS,ENDPLT ( PLOTTING LIB ROUTINES )     DSPLIN
C         PLOT,PCOPY,POINT,SCMULT,SMOOTH,VSMOTH,SIMULA,ELIPSE,      DSPLIN
C         SIGDIS,CTIME                                              DSPLIN
C       - DSQRT                                                     DSPLIN
C*                                                                  DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV  DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,  DSPLIN
     @       IDCOF,IDVCOF                                           DSPLIN
      REAL*8 LOBS(IDLOBS,1),VAR(IDVAR,2,2),TX(IDTX,1),TK(IDTK,1),   DSPLIN
     @       COF(IDCOF,8),VCOF(IDVCOF,8,8),V(IDV,1)                 DSPLIN
C*                                                                  DSPLIN
C* DIMENSIONS OF LOCAL MATRICES - IDCOM = IOBSNO                    DSPLIN
C*                              - DIM TKO = IDKNOT                  DSPLIN
      REAL*8  XX(500,1),YY(500,1),DIST(500,1),TIM(500,1),TKO(32,1)  DSPLIN
      REAL*8  SDIST(500,1)                                          DSPLIN
      REAL*4 ZDASH(2)                                               DSPLIN
      INTEGER ICANYS(10),DEGOF                                      DSPLIN
C*                                                                  DSPLIN
      IDCOM = IOBSNO                                                DSPLIN
      IDTKO = IDKNOT                                                DSPLIN
      SC = 1.0D00/SCTIME                                            DSPLIN
C* IGNORE ROUTINE IF SUBR LS FAILED                                 DSPLIN
      IF(IANLYS.EQ.1.AND.IPLOT.EQ.1)CALL PLOT(ZA,ZB,ZC,ZG,ZS,ZD,ZF,ZE) DSPLIN
      IF(IER.NE.0)RETURN                                            DSPLIN
      IF((IANLYS.GT.1).AND.(  (ICANYS(2).NE.0)                      DSPLIN
     @.OR.(ICANYS(9).NE.0) ) )CALL PLOT(ZA,ZB,ZC,ZG,ZS,ZD,ZF,ZE)    DSPLIN
      IF(ICANYS(1).NE.1)GO TO 50                                    DSPLIN
C* TRANSFER AND COMPUTATION OF THE TIME , RMS AND DISTANCES         DSPLIN
      IL = 2*NO + 4                                                 DSPLIN
      J = 0                                                         DSPLIN
      RMS = 0.0D00                                                  DSPLIN
      DO 10 I = 5,IL,2                                              DSPLIN
      J = J + 1                                                     DSPLIN
      I2 = I + 1                                                    DSPLIN
      XX(J,1) = V(I,1)                                              DSPLIN
      YY(J,1) = V(I2,1)                                             DSPLIN
      X2 = XX(J,1)*XX(J,1)                                          DSPLIN
      Y2 = YY(J,1)*YY(J,1)                                          DSPLIN
```

```
      DIST(J,1) = X2 + Y2                                                DSPLIN
      RMS = RMS + DIST(J,1)                                              DSPLIN
      DIST(J,1) = DSQRT(DIST(J,1))                                       DSPLIN
   10 CONTINUE                                                           DSPLIN
C* TRANSFER THE TIME OF FIXES FOR LATER USE                              DSPLIN
      IL = NO + 1                                                        DSPLIN
      CALL PCOPY(TX,2,1,IL,1,TIM,1,1,IDTX,1,IDCOM,1)                     DSPLIN
      RMS = DSQRT(RMS/NO)                                                DSPLIN
C*                                                                       DSPLIN
C* COMPUTE SIGNED DISTANCES                                              DSPLIN
      CALL SIGDIS(KNO,NO,TIM,SDIST,TK,COF,VCOF,LOBS,IDCOM)               DSPLIN
C*                                                                       DSPLIN
C*  SUBPROGRAM OUTPUT                                                    DSPLIN
C*                                                                       DSPLIN
      WRITE(6,9000)KNO                                                   DSPLIN
      CALL SCMULT(SC,TK,TKO,KNO,1,IDTKO,1)                               DSPLIN
      CALL SCMULT(SC,TIM,TIM,NO,1,IDCOM,1)                               DSPLIN
      WRITE(6,9005)(TKO(I,1),I = 1,KNO)                                  DSPLIN
      WRITE(6,9010)(TIM(I,1),I = 1,NO)                                   DSPLIN
      WRITE(6,9020)(XX(I,1),I = 1,NO)                                    DSPLIN
      WRITE(6,9030)(YY(I,1),I = 1,NO)                                    DSPLIN
      WRITE(6,9040)(DIST(I,1),I = 1,NO)                                  DSPLIN
      WRITE(6,9045)(SDIST(I,1),I = 1,NO)                                 DSPLIN
      WRITE(6,9050)RMS                                                   DSPLIN
C*                                                                       DSPLIN
C* PLOTTING OF THE SEPARATION BETWEEN THE SMOOTH AND OBS DATA POINTS     DSPLIN
 50   IF(ICANYS(2).EQ.0)GO TO 100                                        DSPLIN
      FSC = (DSQRT(DEGCF/(2*NO+8.0D0)))                                  DSPLIN
      IF(ICANYS(7).EQ.1)WRITE(6,9120)                                    DSPLIN
C* DRAW THE CENTRE LINE FIRST ONLY FOR EAST DIFF AND NORTH DIFFERENCES   DSPLIN
      IF(ICANYS(2).EQ.1.OR.ICANYS(5).EQ.1)GO TO 90                       DSPLIN
      CALL PRNTCH('-')                                                   DSPLIN
      ZX = TIM(1,1)                                                      DSPLIN
      CALL NOWPLT(0,ZX,0.0)                                              DSPLIN
      ZX = TIM(NO,1)                                                     DSPLIN
      CALL NOWPLT(1,ZX,0.0)                                              DSPLIN
C* SET THE TYPE OF LINE TO BE DRAWN                                      DSPLIN
 90   ZDASH(1) = IC*0.05                                                 DSPLIN
      ZDASH(2) = 0.1                                                     DSPLIN
C* POSITIONS THE PEN AT THE FIRST POINT                                  DSPLIN
      ZX = TIM(1,1)                                                      DSPLIN
      IF(ICANYS(5).EQ.1)ZY = DIST(1,1)                                   DSPLIN
      IF(ICANYS(5).EQ.2)ZY = XX(1,1)                                     DSPLIN
      IF(ICANYS(5).EQ.3)ZY = YY(1,1)                                     DSPLIN
      IF(ICANYS(5).EQ.4)ZY = SDIST(1,1)                                  DSPLIN
      IF(ICANYS(5).EQ.5)ZY = XX(1,1)/(FSC*DSQRT(VAR(2,1,1)))             DSPLIN
      IF(ICANYS(5).EQ.6)ZY = YY(1,1)/(FSC*DSQRT(VAR(2,2,2)))             DSPLIN
      CALL NOWPLT(0,ZX,ZY)                                               DSPLIN
```

```
C* PLOTTING OF THE POINTS                                              DSPLIN
      CALL PRNTCH('.')                                                 DSPLIN
      DO 60 I = 1,NO                                                   DSPLIN
      II = I + 1                                                       DSPLIN
      ZX = TIM(I,1)                                                    DSPLIN
      IF(ICANYS(5).EQ.1)ZY = DIST(I,1)                                 DSPLIN
      IF(ICANYS(5).EQ.2)ZY = XX(I,1)                                   DSPLIN
      IF(ICANYS(5).EQ.3)ZY = YY(I,1)                                   DSPLIN
      IF(ICANYS(5).EQ.4)ZY = SDIST(I,1)                                DSPLIN
      IF(ICANYS(5).EQ.5)ZY = XX(I,1)/(FSC*DSQRT(VAR(II,1,1)))          DSPLIN
      IF(ICANYS(5).EQ.6)ZY = YY(I,1)/(FSC*DSQRT(VAR(II,2,2)))          DSPLIN
      CALL NOWPLT(0,ZX,0.0)                                            DSPLIN
      CALL NOWPLT(1,ZX,ZY)                                             DSPLIN
C* WRITE OUT VALUES                                                    DSPLIN
      IF(((ICANYS(5).EQ.5).OR.(ICANYS(5).EQ.6) )                       DSPLIN
     *                     .AND.(ICANYS(7).EQ.1))WRITE(6,9110)ZX,ZY    DSPLIN
   60 CONTINUE                                                         DSPLIN
C* AVOID FOR DISTANCES                                                 DSPLIN
      IF(ICANYS(2).EQ.1.OR.ICANYS(5).EQ.1)GO TO 100                    DSPLIN
      CALL PRNTCH('+')                                                 DSPLIN
      DO 70 I = 1,NO                                                   DSPLIN
          CALL POINT(TIM(I,1),X,Y,VX,VY,VXY,ICO,KNO,TK,COF,VCOF)       DSPLIN
          SX = DSQRT(VX)                                               DSPLIN
          SY = DSQRT(VY)                                               DSPLIN
          ZX = TIM(I,1)                                                DSPLIN
          IF( ICANYS(5).EQ.2)ZY = SX                                   DSPLIN
          IF( ICANYS(5).EQ.3)ZY = SY                                   DSPLIN
          CALL NOWPLT(0,ZX,ZY)                                         DSPLIN
      IF(ICANYS(2).EQ.2)GO TO 70                                       DSPLIN
          II = I + 1                                                   DSPLIN
          SX = DSQRT(VAR(II,1,1))                                      DSPLIN
          SY = DSQRT(VAR(II,2,2))                                      DSPLIN
          IF( ICANYS(5).EQ.2)ZY = SX                                   DSPLIN
          IF( ICANYS(5).EQ.3)ZY = SY                                   DSPLIN
          CALL PRNTCH('*')                                             DSPLIN
          CALL NOWPLT(0,ZX,ZY)                                         DSPLIN
          CALL PRNTCH('+')                                             DSPLIN
   70 CONTINUE                                                         DSPLIN
C*                                                                     DSPLIN
C* PRINTS THE SMOOTHED DATA POINTS,COVARIANCE ETC.                      DSPLIN
100   IF(ICANYS(3).NE.1)GO TO 200                                      DSPLIN
      CALL SMOOTH(NO,KNO,SC,ESC,TX,TK,COF,VCOF)                        DSPLIN
C*                                                                     DSPLIN
C* PRINTS THE SMOOTHED VELOCITIES AT THE DATA INPUT TIMES              DSPLIN
 200  IF(ICANYS(4).NE.1)GO TO 300                                      DSPLIN
      CALL VSMOTH(SCTIME,NO,KNO,TX,TK,COF,VCOF)                        DSPLIN
C*                                                                     DSPLIN
C* SPECIAL -  DETERMINES SEPARATION OF SPLINE FORM SIMULATED DATA      DSPLIN
```

```
  300   IF(ICANYS(6).NE.1)GO TO 400                                        OSPLIN
C* GENERATOR DEACTIVATED IF ICANYS = 0                                     OSPLIN
C*                                                                         OSPLIN
C* TEST POINTS                                                             OSPLIN
C*                                                                         OSPLIN
        WRITE(6,9090)                                                      OSPLIN
C* INTERVAL BETWEEN COMPUTED FIXES                                         OSPLIN
        CINT = 2.0000*SCTIME                                               OSPLIN
        SUMT = 0.0000                                                      OSPLIN
        XLARGE = 0.0000                                                    OSPLIN
        INO = (TK(KNO,1)-TK(1,1))/CINT  + 1                                OSPLIN
        DO 20 I = 1,INO                                                    OSPLIN
        TFIX = TK(1,1) + (I-1)*CINT                                        OSPLIN
C* COMPUTING SIMULATED VALUES OF  X AND Y AT TIME TFIX                     OSPLIN
        TFIX2 = TFIX*SC                                                    OSPLIN
        CALL SIMULA(TFIX2,GX,GY)                                           OSPLIN
        CALL POINT(TFIX,X,Y,VX,VY,VXY,ICO,KNO,TK,COF,VCOF)                 OSPLIN
        CALL ELIPSE(VX,VY,VXY,A,B,ESC,DIR)                                 OSPLIN
        SX = DSQRT(VX)                                                     OSPLIN
        SY = DSQRT(VY)                                                     OSPLIN
        DR = RADD(DIR)                                                     OSPLIN
C* COMPUTE SEPERATION                                                      OSPLIN
        DX = X - GX                                                        OSPLIN
        DY = Y - GY                                                        OSPLIN
        DIS = DSQRT(DX*DX + DY*DY)                                         OSPLIN
        SUMT = SUMT + DIS                                                  OSPLIN
        IF(XLARGE.LT.DIS)XLARGE = DIS                                      OSPLIN
        TRU = TFIX*SC                                                      OSPLIN
   40   WRITE(6,9070)I,TRU,X,Y,SX,SY,VXY,DX,DY,DIS                         OSPLIN
  20    CONTINUE                                                           OSPLIN
        WRITE(6,9100)SUMT,XLARGE                                           OSPLIN
C*                                                                         OSPLIN
  400   IF(ICANYS(8).NE.1)GO TO 500                                        OSPLIN
C*                                                                         OSPLIN
C* PRINT SMOOTH LOREX  PARAMETERS AT GINT INTERVALS                        OSPLIN
C*                                                                         OSPLIN
        YSTART = GSTART*SCTIME                                             OSPLIN
        YINT = GINT*SCTIME                                                 OSPLIN
        INO = ((TK(KNO,1) - YSTART)/YINT) + 1                              OSPLIN
C* SET COUNTER                                                             OSPLIN
        IST = 0                                                            OSPLIN
        IK = 1                                                             OSPLIN
        DO 410 I = 1,INO                                                   OSPLIN
        TFIX = YSTART + (I-1)*YINT                                         OSPLIN
        CALL POINT(TFIX,X,Y,VX,VY,VXY,ICO,KNO,TK,COF,VCOF)                 OSPLIN
        IF(ICO.EQ.0)GO TO 410                                              OSPLIN
        IST = IST + 1                                                      OSPLIN
        CALL ELIPSE(VX,VY,VXY,A,B,ESC,DIR)                                 OSPLIN
```

```
          DR = RADD(DIR)                                                  DSPLIN
          CALL VELO(TFIX,SPEED,DIR,VE,VN,SVE,SVN,CVEN,ICO,KNO,TK,COF,VCOF) DSPLIN
          SDR = RADD(DIR)                                                 DSPLIN
          VX = SVE*SVE                                                    DSPLIN
          VY = SVN*SVN                                                    DSPLIN
          CALL ELIPSE(VX,VY,CVEN,VA,VB,ESC,DIR)                           DSPLIN
          VDR = RADD(DIR)                                                 DSPLIN
          TRU = TFIX/SCTIME                                               DSPLIN
          CALL CTIME(TRU,IDAY,IHR,IMIN)                                   DSPLIN
C* CONVERT TO INTEGERS                                                    DSPLIN
          IX   = X                                                        DSPLIN
          IY   = Y                                                        DSPLIN
          IA   = A                                                        DSPLIN
          IB   = B                                                        DSPLIN
          IDR  = DR                                                       DSPLIN
          IVE  = VE                                                       DSPLIN
          IVN  = VN                                                       DSPLIN
          ISDR = SDR                                                      DSPLIN
          ISPD = SPEED                                                    DSPLIN
          IVA  = VA                                                       DSPLIN
          IVB  = VB                                                       DSPLIN
          IVDR = VDR                                                      DSPLIN
C*                                                                        DSPLIN
C* WRITE HEADING                                                          DSPLIN
          IF (IST.NE.IK)GO TO 405                                         DSPLIN
          IK = IK + 45                                                    DSPLIN
          WRITE(6,9130)                                                   DSPLIN
  405     WRITE(6,9140)IDAY,IHR,IX,IY,IA,IB,IDR,                          DSPLIN
     @                 IVE,IVN,ISPD,ISDR,IVA,IVB,IVDR                     DSPLIN
  410     CONTINUE                                                        DSPLIN
C*                                                                        DSPLIN
  500     IF(ICANYS(9).NE.1.AND.ICANYS(9).NE.2)GO TO 999                  DSPLIN
C*                                                                        DSPLIN
C* DRAW COURSE VS TIME OR SPEED VS TIME CHARTS                            DSPLIN
          CALL PRNTCH('.')                                                DSPLIN
          YSTART = XSTART*SCTIME                                          DSPLIN
          YINT = XINT*SCTIME                                              DSPLIN
          INO = ((TK(KNO,1)-YSTART)/YINT) + 1                             DSPLIN
          IK = 0                                                          DSPLIN
          DO 510 I = 1,INO                                                DSPLIN
          TFIX = YSTART + (I-1)*YINT                                      DSPLIN
          CALL VELO(TFIX,SPEED,DIR,VE,VN,SVE,SVN,CVEN,ICO,KNO,TK,COF,VCOF) DSPLIN
          IF(ICO.EQ.0)GO TO 510                                           DSPLIN
          IK = IK + 1                                                     DSPLIN
          SDR = RADD(DIR)                                                 DSPLIN
          ZX = TFIX/SCTIME                                                DSPLIN
          ZY = SPEED                                                      DSPLIN
          IF(ICANYS(9).EQ.1)ZY = SDR                                      DSPLIN
```

```
      IF(IK.EQ.1)CALL NOWPLT(0,ZX,ZY)                                   DSPLIN
      IF(IK.GT.1)CALL NOWPLT(1,ZX,ZY)                                   DSPLIN
  510 CONTINUE                                                          DSPLIN
C*                                                                      DSPLIN
  999 IF((IANLYS.NE.1).AND.(   (ICANYS(2).NE.0)                         DSPLIN
     @          .OR.(ICANYS(9).NE.0) ) )CALL ENDPLT                     DSPLIN
C*                                                                      DSPLIN
      RETURN                                                            DSPLIN
 9000 FORMAT(///,'   NUMBER OF KNOTS = ',I4)                            DSPLIN
 9005 FORMAT(//,'   TIME OF KNOTS',//,                                  DSPLIN
     @ 100(2X,10(1PD12.5,1X),/))                                        DSPLIN
 9010 FORMAT(//,'   TIME OF DATA POINTS',//,                            DSPLIN
     @ 100(2X,8(1PD12.5,1X),/))                                         DSPLIN
 9020 FORMAT(///,'   SEPERATION BETWEEN THE SPLINE AND DATA POINT',//,  DSPLIN
     @ '   EASTING',/,                                                  DSPLIN
     @ 100(2X,8(1PD12.5,1X),/))                                         DSPLIN
 9030 FORMAT(//,'   NORTHINGS',/,                                       DSPLIN
     @ 100(2X,8(1PD12.5,1X),/))                                         DSPLIN
 9040 FORMAT(//,'   ABSOLUTE DISTANCE',/,100(2X,8(1PD12.5,1X),/))       DSPLIN
 9045 FORMAT(//,'   SIGNED   DISTANCE',/,100(2X,8(1PD12.5,1X),/))       DSPLIN
 9050 FORMAT(///,'   R.M.S. VALUE = ',1PD12.5,//)                       DSPLIN
 9070 FORMAT(2X,I4,2X,F10.4,10(F12.3,2X))                               DSPLIN
 9090 FORMAT(//,10X,'PREDICTED FIXES AT SPECIFIED TIMES',/10X,          DSPLIN
     .17('=='),//,2X,'NO',6X,'TIME',12X,                                DSPLIN
     .'POSITION(E,N)           SIGMA POSN(E,N)',13X,' COVARIANCE',4X,   DSPLIN
     .'SEP-X (M)   SEP-Y (M)    DISTANCE(M) ',//)                       DSPLIN
 9100 FORMAT(//,3X,'SUMMATION OF SEPERATION DISTANCES ',1PD12.5,/,      DSPLIN
     @           3X,'LARGEST SEPERATION BETWEEN POINTS',1PD12.5)        DSPLIN
 9110 FORMAT(3X,'TIME : ',F9.4,'   STD RESIDUAL : ',F9.2)               DSPLIN
 9120 FORMAT(//,3X,'PLOTTING VALUES',/,/)                               DSPLIN
 9130 FORMAT(1H1,12(/),12X,'DAY HR  EASTING NORTHING    E1  E2  DIR',   DSPLIN
     @           '  VEAST VNORTH  SPEED COU    E1  E2  DIR',/)          DSPLIN
 9140 FORMAT(12X,I3,1X,I2,2X,I7,1X,I8,2X,I4,1X,I3,1X,I4,                DSPLIN
     @          2X,I5,1X,I6,2X,I5,1X,I3,2X,I4,1X,I3,1X,I4)              DSPLIN
      END                                                               DSPLIN
      SUBROUTINE APRORI(X,M,L,ICODE)                                    DSPLIN
      IMPLICIT REAL*8(A-H,P-Z)                                          DSPLIN
      REAL*8 X(L,1)                                                     DSPLIN
C ******************************************************************** DSPLIN
C * VERSION     : JUNE 1982                                          * DSPLIN
C * DESCRIPTION : INCORPORATES A READ-IN-APRIORI SOLUTION VECTOR     * DSPLIN
C *               FOR THE LEAST SQUARES ESTIMATION                   * DSPLIN
C ******************************************************************** DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C     ICODE =  0    NO APRIORI SOLUTION VECTOR IS CONSIDERED            DSPLIN
C              1    THE APRIORI SOLN VECTOR IS READ                     DSPLIN
C              2    THE APRIORI SOLN VECTOR IS GENERATED BY THE PROGRAM DSPLIN
C                                                                       DSPLIN
```

```
C OUTPUT PARAMETERS                                                          DSPLIN
C     X(M,1)=  APRIORI SOLUTION VECTOR OF DIMENSION (M,1)                    DSPLIN
C              BUT WITH DECLARED DIMENSION (L,1)                             DSPLIN
C                                                                            DSPLIN
C* READ/WRITE UNITS                                                          DSPLIN
      IRD = 5                                                                DSPLIN
      IWR = 6                                                                DSPLIN
      IF(ICODE.LT.0.AND.ICODE.GT.2)RETURN                                    DSPLIN
      IF(ICODE.EQ.1)GO TO 200                                                DSPLIN
      IF(ICODE.EQ.2)GO TO 300                                                DSPLIN
C* OPTION ZERO                                                               DSPLIN
 100  DO 10 I = 1,M                                                          DSPLIN
  10  X(I,1) = 0.0D00                                                        DSPLIN
      RETURN                                                                 DSPLIN
C* OPTION ONE                                                                DSPLIN
 200  READ(IRD,*)(X(I,1),I = 1,M)                                            DSPLIN
      RETURN                                                                 DSPLIN
C* OPTION TWO                                                                DSPLIN
 300  WRITE(IWR,99)                                                          DSPLIN
 99   FORMAT(//,'  THIS OPTION IS NOT AVAILIABLE YET',//)                    DSPLIN
      IF(ICODE.EQ.2)GO TO 100                                                DSPLIN
C* DEVELOPED ALGO SHOULD BE FEED IN HERE                                     DSPLIN
C*                                                                           DSPLIN
      RETURN                                                                 DSPLIN
      END                                                                    DSPLIN
      SUBROUTINE BMULT1(F,I,J,IBAND,S,K,L,R,NF,NS,MS,NR,MR)                  DSPLIN
      REAL*8 F(NF,IBAND),S(NS,MS),R(NR,MR)                                   DSPLIN
C ****************************************************************           DSPLIN
C * VERSION     : JAN 1981                                         *         DSPLIN
C * DESCRIPTION : TO MULTIPLY A BANDED F MATRIX WITH S TO GIVE R   *         DSPLIN
C ****************************************************************           DSPLIN
C INPUT PARAMETERS                                                           DSPLIN
C    F      - PSEUDO DIMENSIONS OF F ARE (I,J)                               DSPLIN
C             THE ACTUAL DIMENSICNS OF F ARE (NF,IBAND)                      DSPLIN
C    IBAND - EANCWIDTH OF MATRIX F                                           DSPLIN
C    S      - THE DECLARED SIZE OF S IS (NS,MS)                              DSPLIN
C                                                                            DSPLIN
C OUTPUT PARAMETERS                                                          DSPLIN
C    R    - DECLARED SIZE IS (NR,MR)                                         DSPLIN
C                                                                            DSPLIN
C ADDITIONAL INFORMATICN                                                     DSPLIN
C  VARIABLE IELE DENOTES THE NUMBER OF ELEMENTS UP TO AND INCLUDING THE      DSPLIN
C  DIAGONAL ELEMENT IN ANY ONE ROW -  NO. OF CODIAGONALS  + 1                DSPLIN
C*BEGIN                                                                      DSPLIN
      IELE = (IBAND+1)/2                                                     DSPLIN
      DO 10 II = 1,I                                                         DSPLIN
      IPOS = II - IELE                                                       DSPLIN
      DO 10 IJ = 1,L                                                         DSPLIN
```

```
      R(II,IJ) = 0.0D00
      DO 10 JAC = 1,IBAND
C* JAC DENOTES THE ACTUAL POSITION OF THE ELEMENT IN THE F MATRIX
      IK = JAC + IPOS
      IF(IK.LT.1.OR.IK.GT.K)GO TO 10
   20 R(II,IJ) = R(II,IJ) + F(II,JAC)*S(IK,IJ)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE BUILD(KNO,X,VX,COF,VCOF,DTK,
     @                 CX,VCX,CO,VCO,PRCOFV,PRCOFF)
C *****************************************************************
C * VERSION         : 7 JUNE 1982                                 *
C * DESCRIPTION   : TRANSFORM POSITION AND SLOPE INTO COEFFICIENTS *
C *****************************************************************
C INPUT PARAMETERS
C      KNO    - NUMBER OF KNOTS
C      X,VX   - SOLUTION AND COVARIANCE MATRICES (DIM PASS IN BY COM BLK)
C      DTK    - INTERVAL IN TIME BETWEEN KNOTS(SEE MAIN )
C      PRCOFF,PRCOFV - CODES FOR PRINTING OF COEFFICIENTS AND VARIANCES
C                     (SEE OPTION ROUTINE )
C OUTPUT PARAMTERS
C      COF,VCOF- COEFFICIENTS AND COVARIANCES (DIM PASSED IN BY COM BLK)
C
C WORK ARRAYS
C      CX,VCX,CO,VCO AS DIMENSIONED IN THE ROUTINE
C
C CALLS        - PCOPY,COEF,DPRINT
C                     ***************
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,
     @         IDCOF,IDVCOF
C* ARRAYS
      REAL*8 X(IDX,1),VX(IDVX,IDVX),COF(IDCOF,8),VCOF(IDVCOF,8,8),
     @       CO(8,1),VCO(8,8),CX(8,1),VCX(8,8),DTK(IDDTK,1)
      INTEGER KNO,PRCOFV,PRCOFF
C*BEGINS
      L = 4*KNO - 4
      J = 0
      DO 10 I = 1,L,4
      J = J + 1
      IL = I + 7
      CALL PCOPY(X,I,1,IL,1,CX,1,1,IDX,1,8,1)
      CALL PCOPY(VX,I,I,IL,IL,VCX,1,1,IDVX,IDVX,8,8)
      CALL COEF(CO,VCO,DTK(J,1),CX,VCX)
      DO 20 IK = 1,8
      COF(J,IK) = CO(IK,1)
      DO 20 IJ = 1,8
```

```
   20  VCOF(J,IK,IJ) = VCO(IK,IJ)                                            DSPLIN
       IF(PRCOFV.EQ.0)GO TO 10                                               DSPLIN
       CALL DPRINT(CO,8,1,8,1,'COEFFICIENTS',6)                              DSPLIN
       CALL DPRINT(VCO,8,8,8,8,'COVARIANCE',5)                               DSPLIN
   10  CONTINUE                                                              DSPLIN
C* OUTPUT OF COEFFICIENTS                                                     DSPLIN
       IF(PRCOFF.EQ.0)RETURN                                                 DSPLIN
       IF(PRCOFF.EQ.1.AND.PRCOFV.EQ.1)RETURN                                 DSPLIN
       CALL DPRINT(COF,J,8,IDCOF,8,'COEFFICIENTS PER  INTERVAL',13)          DSPLIN
       RETURN                                                                DSPLIN
       END                                                                   DSPLIN
       SUBROUTINE CHECK(KNO,NO,TX,TK,ICODE)                                  DSPLIN
       IMPLICIT REAL*8(A-H,O-Z)                                              DSPLIN
C   ***************************************************************          DSPLIN
C   *  VERSION      : 22ND MARCH 1982                                *       DSPLIN
C   *  DESCRIPTION : CHECKING AND OPTIONAL DROPPING OF KNOTS         *       DSPLIN
C   ***************************************************************          DSPLIN
C INPUT PARAMETERS                                                            DSPLIN
C    KNO    - NUMBER OF KNOTS                                                DSPLIN
C    NO     - NUMBER OF POINTS                                               DSPLIN
C    TX     - ARRAY CONTAINING DATA POINT TIMES                              DSPLIN
C    TK     - ARRAY CONTAINING KNOT TIMES                                    DSPLIN
C    ICODE - 0  TKNOT VS DATA POINT SEQUENCE IS NOT CHECKED                  DSPLIN
C            1  CHECKING AND DROP SCHEME KNOT ACTIVATED                      DSPLIN
C                                                                            DSPLIN
C OUTPUT PARAMETERS                                                           DSPLIN
C    TK     - MODIFIED AS INSTRUCTED BY CODES                                DSPLIN
C                                                                            DSPLIN
C CALLS    - INTERV                                                          DSPLIN
C ADDITIONAL NOTES                                                            DSPLIN
C     CRTERA DENOTES THE PRESET MINIMUM NUMBER OF DATA POINTS ALLOWED        DSPLIN
C  PER INTERVAL BEFORE THE 'DROP KNOT SCHEME' IS IMPLEMENTED.                DSPLIN
C                                                                            DSPLIN
C     KNOTS ARE DROPPED IN SUCH A MANNER AS TO INCREASE THE                  DSPLIN
C  MORE DEFICIENT ADJACENT INTERVAL.                                         DSPLIN
C  EG.  DATA PLACEMENT  6 2 1 4 5   UNDER CRTERA = 2                         DSPLIN
C        BECOMES        6 3 4 5                                              DSPLIN
C  WHEREAS                                                                   DSPLIN
C       DATA PLACEMENT  6 4 1 2 5   UNDER CRTERA  = 2                        DSPLIN
C       BECOMES         6 4 3 5                                              DSPLIN
C                   ****************                                         DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV            DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,            DSPLIN
     @      IDCOF,IDVCOF                                                     DSPLIN
      REAL*8 TX(IDTX,1),TK(IDTK,1)                                           DSPLIN
      INTEGER*4 NO,KNO,NUM(40),NOK,ITOTAL,I84,IAF,IDROP,CRTERA               DSPLIN
      DATA CRTERA/1/                                                         DSPLIN
C*BEGINS                                                                      DSPLIN
```

```
      ITOTAL = 0
  999 NOK = KNO - 1                                                         DSPLIN
      DO 10 I = 1,NOK                                                       DSPLIN
   10 NUM(I) = 0                                                            DSPLIN
      L = NO + 1                                                            DSPLIN
      DO 20 I = 2,L                                                         DSPLIN
      CALL INTERV(TX(I,1),TK,KNO,INT,IDTK,IC)                               DSPLIN
      IF(IC.EQ.0)WRITE(6,50)TX(I,1)                                         DSPLIN
      IF(IC.EQ.0)GO TO 20                                                   DSPLIN
      NUM(INT) = NUM(INT) + 1                                               DSPLIN
   20 CONTINUE                                                              DSPLIN
      WRITE(6,40)(NUM(I),I = 1,NOK)                                         DSPLIN
   40 FORMAT(//,3X,'PLACEMENT OF DATA POINTS IN THE KNOT SEQUENCE',         DSPLIN
     @ //,3X,5(10I4,/,3X),//)                                               DSPLIN
   50 FORMAT(/,'  TIME ',1PD12.5,' BEYOND THAT OF OUTER KNOTS',//)          DSPLIN
C*                                                                          DSPLIN
C* IF ICODE = 0 THE CHECKING IS IGNORED                                     DSPLIN
      IF(ICODE.EQ.0)RETURN                                                  DSPLIN
C*                                                                          DSPLIN
C* BEGIN *** LOCATE THE FIRST DEFICIENT INTERVAL WITH LESS THAN             DSPLIN
C*             CRTERA NUMBER OF DATA POINTS                                 DSPLIN
      DO 100 I = 1,NOK                                                      DSPLIN
      IF(NUM(I).LT.CRTERA)GO TO 110                                         DSPLIN
  100 CONTINUE                                                              DSPLIN
C* IF NONE ARE TO BE FOUND ANY MORE                                         DSPLIN
      IF(ITOTAL.NE.0)WRITE(6,200)ITOTAL                                     DSPLIN
      GO TO 9999                                                            DSPLIN
C* NOW IDENTIFY WHICH KNOT TO DROP(IDROP)                                   DSPLIN
C*   SPECIAL CASE I = 1 OR I = NOK                                          DSPLIN
  110 IF(I.EQ.1)IDROP = 2                                                   DSPLIN
      IF(I.EQ.NOK)IDROP = NOK                                               DSPLIN
      IF(I.EQ.1.OR.I.EQ.NOK)GO TO 1000                                      DSPLIN
C* NO SPECIAL CASE       1 < I < NOK                                        DSPLIN
      IB4 = I-1                                                             DSPLIN
      IAF = I + 1                                                           DSPLIN
C* THE KNOT IS DROPPED IN SUCH A WAY AS TO INCREASE THE MORE DEFICIENT      DSPLIN
C* ADJACENT KNOT INTERVAL                                                   DSPLIN
C* SPECIAL CASE - EQUAL DATA POINTS IN ADJACENT INTERVALS                   DSPLIN
      IF(NUM(IB4).EQ.NUM(IAF))GO TO 120                                     DSPLIN
C* NO SPECIAL CASE                                                          DSPLIN
      IF(NUM(IAF).LT.NUM(IB4))IDROP = I + 1                                 DSPLIN
      IF(NUM(IAF).GT.NUM(IB4))IDROP = I                                     DSPLIN
      GO TO 1000                                                            DSPLIN
C* IF THEY ARE EQUAL THAN THE NEXT OUTER TWO KNOTS IS EXAMINED              DSPLIN
C* SPECIAL CASE I=2 OR I = NOK-1                                            DSPLIN
  120 IB4 = NOK - 1                                                         DSPLIN
      IF(I.EQ.2)IDROP = 3                                                   DSPLIN
      IF(I.EQ.IB4)IDROP = KNO -2                                            DSPLIN
```

```
      IF(I.EQ.2.OR.I.EQ.IB4)GO TO 1000                                   DSPLIN
C* NO SPECIAL CONSIDERATION        2 < I < NCK-1                         DSPLIN
      IB4 = I - 2                                                        DSPLIN
      IAF = I + 2                                                        DSPLIN
C* IF THEY ARE ALSO EQUAL THEN THE LATER KNCT IS DRCPPED                 DSPLIN
      IDROP = I + 1                                                      DSPLIN
      IF(NUM(IAF).GT.NUM(IB4))IDROP = I                                  DSPLIN
C* DELETION OF THE UNWANTED KNOT                                         DSPLIN
 1000 DC 130 J = IDRCP,NCK                                               DSPLIN
      IJ = J + 1                                                         DSPLIN
  130 TK(J,1) = TK(IJ,1)                                                 DSPLIN
C* READJUSTMENT OF THE TOTAL NUMBER OF KNOTS                             DSPLIN
      KNO = KNC - 1                                                      DSPLIN
C* TOTAL UP THE NUMBER OF DROPPED KNOTS                                  DSPLIN
      ITCTAL = ITOTAL + 1                                                DSPLIN
      GO TO 999                                                          DSPLIN
  200 FORMAT(/,3X,'NUMBER OF KNOTS DELETED : ',I4)                       DSPLIN
9999  RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE CHITES(SO,DEGOF,ALPHA,ICODE)                            DSPLIN
      IMPLICIT REAL*8(A-H,O-Y)                                           DSPLIN
C ****************************************************************       DSPLIN
C * VERSION      : 4 JULY  1982                                  *       DSPLIN
C * DESCRIPTION : DETERMINES IF THE VARIANCE FACTOR (SO)         *       DSPLIN
C *               PASSES THE CHI SQUARE TEST ON THE VARIANCE     *       DSPLIN
C *               FACTORS WITH DEGOF DEGREES OF FREEDOM.         *       DSPLIN
C ****************************************************************       DSPLIN
C INPUT PARAMETERS                                                       DSPLIN
C  SO    - VARIANCE FACTOR TO BE TESTED                                  DSPLIN
C  DEGOF - DEGREES OF FREEDOM ASSOCIATED WITH THE ABOVE APOSTERIORI      DSPLIN
C          VARIANCE FACTCR.                                              DSPLIN
C  ALPHA - PERCENTAGE CONFIDENCE INTERVAL                                DSPLIN
C          0 < ALPHA < 100                                               DSPLIN
C                                                                        DSPLIN
C OUTPUT PARAMETERS                                                      DSPLIN
C  ICODE  - 0  PASSED THE TEST                                           DSPLIN
C           1  FAILED THE TEST - EXCEEDED LOWER LIMIT                    DSPLIN
C           2  FAILED THE TEST - EXCEEDED UPPER LIMIT                    DSPLIN
C                                                                        DSPLIN
C CALLS - MDCHI ( IMSL ROUTINE )                                         DSPLIN
C                *********                                               DSPLIN
      REAL*8 ALPHA,SO                                                    DSPLIN
      INTEGER DEGOF                                                      DSPLIN
      ZUL = 1.0D0 - ((1.0D0 - (ALPHA/100.0D0))/2.0D0)                    DSPLIN
      ZLL = (1.0D0 - (ALPHA/100.0D0))/2.0D0                              DSPLIN
      ZDEGOF = DEGOF                                                     DSPLIN
      CALL MDCHI(ZLL,ZDEGOF,ZX,IER)                                      DSPLIN
      IF(IER.NE.0)RETURN                                                 DSPLIN
```

```
      CALL MDCHI(ZUL,ZDEGOF,ZY,IER)                                     DSPLIN
      IF(IER.NE.0)RETURN                                                DSPLIN
      ICODE = 0                                                         DSPLIN
      ZS0 = S0*DEGOF                                                    DSPLIN
      IF(ZS0.LT.ZX)ICODE = 1                                            DSPLIN
      IF(ZS0.GT.ZY)ICODE = 2                                            DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE CHOLD(A,M,L,ICO,DET,IER)                               DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                          DSPLIN
C *******************************************************************  DSPLIN
C * VERSION DATES - 25TH APRIL 1982                                  *  DSPLIN
C * NAME CHOLD                                                       *  DSPLIN
C * AUTHOR          - D.WELLS                                        *  DSPLIN
C * MODIFICATIONS - SH QUEK                                          *  DSPLIN
C * DESCRIPTION    - MATRIX INVERSION                                *  DSPLIN
C *******************************************************************  DSPLIN
C INPUT ARGUMENTS                                                       DSPLIN
C                 A - MATRIX TO BE INVERTED                             DSPLIN
C                 M - UPPER LEFT CORNER OF MATRIX                       DSPLIN
C                 L - DELCARED SIZE OF A IN THE CALLING PROGRAM         DSPLIN
C               ICO - NO. OF CODIAGONALS IN A                           DSPLIN
C               DET - DETERMINANT OF MATRIX A                           DSPLIN
C               IER - ERROR CODES                                       DSPLIN
C                   = 0 SUCCESSFUL RETURN                               DSPLIN
C                     1 SINGULAR INVERSE                                DSPLIN
C                     2 INPUT ARGUMENT ERROR                            DSPLIN
C                     3 INPUT MATRIX NOT DIAGINAL DOMINANT              DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C                 A  - INVERTED MATRIX                                  DSPLIN
C                                                                       DSPLIN
C CALLS  - TRAPS,DSQRT,DABS                                             DSPLIN
C* BEGIN                                                                DSPLIN
      REAL*8 A(L,L),TDET                                                DSPLIN
      REAL*8 SUM,DET                                                    DSPLIN
C* SINGULAR DETECTOR SETTING                                            DSPLIN
      SING = 1.0D-20                                                    DSPLIN
C* HALF BAND WIDTH - CODIAGONALS PLUS ONE                               DSPLIN
      IK = ICO + 1                                                      DSPLIN
C* DIMENSION CHECK                                                      DSPLIN
      IER = 2                                                           DSPLIN
      IF(M.LT.1)GO TO 999                                               DSPLIN
      IER = 0                                                           DSPLIN
C* TEST NON-NEGATINENESS OF FIRST ELEMENT                               DSPLIN
      IF(A(1,1).LE.0.0D0)GO TO 5                                        DSPLIN
C* SCALER INVERSION                                                     DSPLIN
      IF(M.GT.1)GO TO 10                                                DSPLIN
```

```
      DET = A(1,1)                                                      DSPLIN
      A(1,1) = 1.0D0/A(1,1)                                             DSPLIN
      RETURN                                                            DSPLIN
    5 IER = 3                                                           DSPLIN
      RETURN                                                            DSPLIN
C* CHOLESKI DECOMPOSITION OF INPUT MATRIX INTO TRIANGULAR MATRIX         DSPLIN
C** NOTE THE END OF LOOPS - ACCOUNTING FOR BANDNESS OF MATRIX A          DSPLIN
   10 DET = A(1,1)                                                      DSPLIN
      A(1,1) = DSQRT(A(1,1))                                            DSPLIN
      IF(IK.GT.M)IK = M                                                 DSPLIN
      DO 20 I = 2,IK                                                    DSPLIN
   20    A(I,1) = A(I,1)/A(1,1)                                         DSPLIN
      DO 60 J = 2,M                                                     DSPLIN
         SUM = 0.0D0                                                    DSPLIN
         J1 = J - 1                                                     DSPLIN
         DO 30 K = 1,J1                                                 DSPLIN
   30       SUM = SUM + A(J,K)**2                                       DSPLIN
         X = (A(J,J)-SUM)                                               DSPLIN
C* SINGULARITY CHECK                                                    DSPLIN
         IER = 3                                                        DSPLIN
         IF(X.LT.0.0D0)GO TO 999                                        DSPLIN
C* UNDERFLOW TRAP ACTIVATED WHEN DET => 0                               DSPLIN
         CALL TRAPS(0,0,1,0,0)                                          DSPLIN
         IER = 4                                                        DSPLIN
         DET = DET*X                                                    DSPLIN
         TDET = DET                                                     DSPLIN
         IF(DABS(TDET).LT.1.0D-50)GO TO 999                             DSPLIN
         IER = 1                                                        DSPLIN
         X = DSQRT(X)                                                   DSPLIN
         XX = X/A(J,J)                                                  DSPLIN
         IF(DABS(XX).LE.SING)GO TO 999                                  DSPLIN
         IER = 0                                                        DSPLIN
         A(J,J) = X                                                     DSPLIN
        IF(J.EQ.M)GO TO 60                                              DSPLIN
        J2 = J + 1                                                      DSPLIN
C*                                                                      DSPLIN
        IE = J2 + ICO                                                   DSPLIN
        IF(IE.GT.M)IE = M                                               DSPLIN
        DO 50 I = J2,IE                                                 DSPLIN
        SUM = 0.0D0                                                     DSPLIN
          DO 40 K = 1,J1                                                DSPLIN
   40        SUM = SUM + A(I,K)*A(J,K)                                  DSPLIN
   50    A(I,J) = (A(I,J)- SUM)/A(J,J)                                  DSPLIN
   60 CONTINUE                                                          DSPLIN
C*                                                                      DSPLIN
C* INVERSION OF LOWER TRIANGULAR MATRIX                                  DSPLIN
C*                                                                      DSPLIN
      DO 70 I = 1,M                                                     DSPLIN
```

```
   70      A(I,I) = 1.0D0/A(I,I)                                          DSPLIN
        M1 = M - 1                                                        DSPLIN
        DO 90 J = 1,M1                                                    DSPLIN
           J2 = J + 1                                                     DSPLIN
           DO 90 I = J2,M                                                 DSPLIN
              SUM = 0.0D0                                                 DSPLIN
              I1 = I - 1                                                  DSPLIN
              DO 80 K = J,I1                                              DSPLIN
   80            SUM = SUM + A(I,K)*A(K,J)                                DSPLIN
   90      A(I,J) = -A(I,I)*SUM                                           DSPLIN
C*                                                                        DSPLIN
C* CONSTRUCTION OF THE INVERSE OF INPUT MATRIX                            DSPLIN
C*                                                                        DSPLIN
        DO 140 J = 1,M                                                    DSPLIN
           IF(J.EQ.1)GO TO 110                                            DSPLIN
           J1 = J - 1                                                     DSPLIN
           DO 100 I = 1,J1                                                DSPLIN
  100         A(I,J) = A(J,I)                                             DSPLIN
  110      DO 130 I = J,M                                                 DSPLIN
              SUM = 0.0D0                                                 DSPLIN
              DO 120 K = I,M                                              DSPLIN
  120            SUM = SUM + A(K,I)*A(K,J)                                DSPLIN
  130         A(I,J) = SUM                                                DSPLIN
  140       CONTINUE                                                      DSPLIN
        RETURN                                                            DSPLIN
  999   WRITE(6,9999)IER                                                  DSPLIN
 9999   FORMAT(/,3X,'**TERMINAL ERROR** CHOLD FAILURE ; CODE = ',I2,/,/)  DSPLIN
        RETURN                                                            DSPLIN
        END                                                               DSPLIN
        SUBROUTINE COEF(COF,VCOF,DT,X,VX)                                 DSPLIN
C   *****************************************************************    DSPLIN
C   * VERSION     : JAN 1982                                        *    DSPLIN
C   * DESCRIPTION : COMPUTES THE COEFFICIENTS OF A 3RD DEGREE       *    DSPLIN
C   *               POLYNOMIAL WITH AN ASSOCIATED COVARIANCE MATRIX *    DSPLIN
C   *****************************************************************    DSPLIN
C   INPUT PARAMETERS                                                      DSPLIN
C    DT    - TIME INTERVAL BETWEEN KNOTS                                  DSPLIN
C    X     - VECTOR CONTAINING (XK YK X1K Y1K XK+1 YK+1 X1K+1 Y1K+1)     DSPLIN
C            I.E. POSITON AND SLOPE OF KNOTS DEFINING INTERVAL            DSPLIN
C    VX    - THE FULL COVARIANCE MATRIX OF THE ABOVE X VECTOR             DSPLIN
C                                                                         DSPLIN
C   OUTPUT PARAMETERS                                                     DSPLIN
C    COF   - VECTOR OF 8 COEFFICIENTS DEFINING THE KTH CUBIC INTERVAL     DSPLIN
C    VCOF  - COVARIANCE MATRIX FOR THE COEFFICIENTS                       DSPLIN
C                                                                         DSPLIN
C   CALLS - DMULT                                                         DSPLIN
C                                                                         DSPLIN
C* PROGRAM BEGINS                                                         DSPLIN
```

```
      REAL*8 COF(8,1),VCOF(8,8),X(8,1),VX(8,8),                          DSPLIN
     .C(8,8),DT,D,DD,DDD                                                 DSPLIN
      DATA C/1.0D0,8*0.0D0,1.0D0,8*0.0D0,1.0D0,8*0.0D0,1.0D0,36*0.0D0/   DSPLIN
      D   = 1.0D0/DT                                                     DSPLIN
      DD  = D*D                                                          DSPLIN
      DDD = DD*D                                                         DSPLIN
      C(5,1) = -3.0D00*DD                                                DSPLIN
      C(5,3) = -2.0D0*D                                                  DSPLIN
      C(5,5) = 3.0D00*DD                                                 DSPLIN
      C(5,7) = -D                                                        DSPLIN
      C(7,1) = 2.0D0*DDD                                                 DSPLIN
      C(7,3) = DD                                                        DSPLIN
      C(7,5) = -2.0D00*DDD                                               DSPLIN
      C(7,7) = DD                                                        DSPLIN
C*                                                                       DSPLIN
      DO 10 I = 6,8,2                                                    DSPLIN
      II = I - 1                                                         DSPLIN
      DO 10 J = 2,8,2                                                    DSPLIN
      JJ = J - 1                                                         DSPLIN
   10 C(I,J) = C(II,JJ)                                                  DSPLIN
C* FORMULATION OF COEFFICIENT VECTOR                                     DSPLIN
      CALL DMULT(C,8,8,X,8,1,COF,8,8,8,1,8,1)                            DSPLIN
C* FORMULATION OF COEFFICIENT COVARIANCE MATRIX                          DSPLIN
      DO 20 I = 1,8                                                      DSPLIN
      DO 20 L = 1,8                                                      DSPLIN
      VCOF(I,L) = 0.0D0                                                  DSPLIN
      DO 20 J = 1,8                                                      DSPLIN
      DO 20 K = 1,8                                                      DSPLIN
   20 VCOF(I,L) = VCOF(I,L) + C(I,K)*VX(K,J)*C(L,J)                      DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE COND(A,M,L,CONDNC,ICODE)                                DSPLIN
C ******************************************************************    DSPLIN
C * VERSION     : JAN 1982                                          *    DSPLIN
C * DESCRIPTION : COMPUTES THE CONDITION NUMBER OF A MATRIX         *    DSPLIN
C ******************************************************************    DSPLIN
C INPUT PARAMETERS                                                       DSPLIN
C  A,M,L = MATRIX A OF ROW DIMENSION M AND DECLARED DIMENSION L          DSPLIN
C  ICODE = 1 THE MATRIX PRIOR TC THE INVERSE.                            DSPLIN
C            COMPUTES THE MAX NORM VECTOR.                               DSPLIN
C          2 OR 4 THE MATRIX AFTER INVERSE.                              DSPLIN
C            COMPUTE THE MAX NORM VECTOR AND MULTIPLY  WITH THE          DSPLIN
C              FIRST TO GET CONDITION NUMBER.                            DSPLIN
C           IF   2  - CONDITION NUMBER PRINTED BY SUBROUTINE             DSPLIN
C                4  - CONDITION NUMBER RETURN WITHOUT PRINTING IT        DSPLIN
C                                                                        DSPLIN
C OUTPUT PARAMETERS                                                      DSPLIN
C  CONDNO = CONDITION NUMBER OF MATRIX A(M,M)                            DSPLIN
```

```
C                                                                         DSPLIN
      REAL*8 A(L,L),NORM,NORM1,SUM,XLR,CONDNO                             DSPLIN
      IWR = 6                                                             DSPLIN
      XLR = 0.0D0                                                         DSPLIN
      DO 10 J = 1,M                                                       DSPLIN
      SUM = 0.0D0                                                         DSPLIN
      DO 20 I = 1,M                                                       DSPLIN
C* CHIOCE OF NORMS - ECULIDEAN OR SUM OF ABSOLUTE VALUES                   DSPLIN
   20 SUM = SUM + A(I,J)*A(I,J)                                           DSPLIN
CS20  SUM = SUM + DABS(A(I,J))                                            DSPLIN
      NORM = DSQRT(SUM)                                                   DSPLIN
      NORM = SUM                                                          DSPLIN
      IF(NORM.GT.XLR)XLR = NORM                                           DSPLIN
   10 CONTINUE                                                            DSPLIN
      IF(ICODE.EQ.4.OR.ICODE.EQ.2)GO TO 100                               DSPLIN
      IF(ICODE.NE.1)GO TO 999                                             DSPLIN
      NORM1 = NORM                                                        DSPLIN
      RETURN                                                              DSPLIN
  100 IF(NORM1.EQ.0)GO TO 999                                             DSPLIN
      CONDNO = NORM1*NORM                                                 DSPLIN
      IF(ICODE.EQ.2)WRITE(IWR,90)CONDNO                                   DSPLIN
      NORM1 = 0.0D0                                                       DSPLIN
      RETURN                                                              DSPLIN
  999 WRITE(IWR,99)                                                       DSPLIN
      RETURN                                                              DSPLIN
   99 FORMAT('   CODING ERROR ON SUBR COND')                              DSPLIN
   90 FORMAT('   CONDITION NUMBER OF MATRIX ; ',1PD12.5)                  DSPLIN
      END                                                                 DSPLIN
      SUBROUTINE CONSTR(DTKBF,DTKAF,A)                                    DSPLIN
      IMPLICIT REAL*8(A-H,M-Z)                                            DSPLIN
C *************************************************************          DSPLIN
C * VERSION      :  JAN 1981                                   *          DSPLIN
C * DESCRIPTION  : TO GENERATE THE COEFFICIENTS OF A CONSTRAINT *         DSPLIN
C *                EQUATION.                                    *         DSPLIN
C *************************************************************          DSPLIN
C  INPUT PARAMETERS                                                       DSPLIN
C        DTKBF - LENGTH OF TIME INTERVAL PRIOR TO CENTRAL KNOT            DSPLIN
C        DTKAF - LENGTH OF TIME INTERVAL AFTER CENTRAL KNOT               DSPLIN
C          I.E.     DTKBF = KNOT(2) - KNOT(1)                             DSPLIN
C                   DTKAF = KNOT(3) - KNOT(2)                             DSPLIN
C OUTPUT PARAMETERS                                                       DSPLIN
C        A        - VECTOR OF 11 COEFFICIENTS                             DSPLIN
C                   ********                                              DSPLIN
      REAL*8 DTKBF,DTKAF,A(11,1),X,XX,Y,YY                                DSPLIN
      X = 1.0D00/DTKBF                                                    DSPLIN
      Y = 1.0D00/DTKAF                                                    DSPLIN
      XX = X*X                                                            DSPLIN
      YY = Y*Y                                                            DSPLIN
```

```
      A(1,1) = 3.0D00*XX                                                DSPLIN
      A(3,1) = X                                                        DSPLIN
      A(5,1) = 3.0D00*(YY-XX)                                           DSPLIN
      A(7,1) = 2.0D00*(X+Y)                                             DSPLIN
      A(9,1) = -3.0D00*YY                                               DSPLIN
      A(11,1) = Y                                                       DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE DESIGN(NO,KNO,ICOUNT,ITYPE,                            DSPLIN
     @                  SPLIN,PRGEN,PRPTS,PRKNOT,DRAW,                  DSPLIN
     @                  IER,ESC,SCTIME,XSTART,XINT,FORFIV,NINTY,        DSPLIN
     @                  TK,TX,CCF,VCOF,DTK,VAR,LOBS)                    DSPLIN
      IMPLICIT REAL*8(A-H,O-Y)                                          DSPLIN
C ******************************************************************** DSPLIN
C * VERSION     : 14 JUNE 1982                                        * DSPLIN
C * DESCRIPTION : PERFORMS THE SETTING UP OF THE PLOTTING AREA.       * DSPLIN
C *               PLOTS THE REQUIRED SPECIFIED POINTS, KNOTS,         * DSPLIN
C *               SMOOTHED POINTS, DO OVERLAYS ETC.                   * DSPLIN
C ******************************************************************** DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C   NO,KNO      - NO. OF DATA POINTS AND KNOTS                          DSPLIN
C   ICOUNT      - NUMBER OF TIMES THIS ROUTINE IS CALLED(PLOT CNTRL)    DSPLIN
C   ITYPE       - PRINT/GRAPHICS PLOT(SEE OPTION ROUTINE)               DSPLIN
C                 0 - GRAPHICS PLOTTER                                  DSPLIN
C                 1 - LINE PRINTER                                      DSPLIN
C                                                                       DSPLIN
C   SPLIN,PRGEN,PRPTS,PRKNOT,DRAW - PRINT/DRAWING CODES - SEE MAIN      DSPLIN
C                                                                       DSPLIN
C      A) PRPTS   : CODES  0 - NO PLOT OF POINTS                        DSPLIN
C                          1 - PLOT OF POINTS - N VS E                  DSPLIN
C                          2 -                - E VS TIME               DSPLIN
C                          3 -                - N VS TIME               DSPLIN
C                          4 -                - N VS E WITH ELLIPSE     DSPLIN
C      B) PRKNOTS: CODES   0 - NO PLOTTING OF KNOT POSITIONS            DSPLIN
C                          1 - PLOT OF KNOTS  - N VS E                  DSPLIN
C                          2 -                - E VS TIME               DSPLIN
C                          3 -                - N VS TIME               DSPLIN
C      C) DRAW    : CODES  0 - NO PLOT OF COMPUTED SPLINE POSITIONS     DSPLIN
C                          1 - PLOT OF SPLINED PTS                      DSPLIN
C                                             - N VS E (DASHED LINES)   DSPLIN
C                          2 -                - E VS TIME               DSPLIN
C                          3 -                - N VS TIME               DSPLIN
C                          4 -                - N VS E (DISCRETE ELLIPSE)DSPLIN
C      D) PRGEN   : CODES  0 - NO PRINTING OF COMPUTED SPLINED PTS      DSPLIN
C                          1 - INFORMATION USED TO DRAW SPLINE PRINTED  DSPLIN
C                                                                       DSPLIN
C   IER         - ERROR CODE TO HALT PLOT OF POINTS IF LS ROUTINE FAILS DSPLIN
C   ESC         - SCALE FACTOR FOR ERROR ELLIPSES                       DSPLIN
```

```
C    SCTIME       - SCALE FACTOR ON TIME DOMAIN                               DSPLIN
C    XSTART,XINT- TIME TO BEGIN DRAWING SPLINE AT  XINT INCREMENTS           DSPLIN
C    TK,TX,COF,VCOF,DTK,VAR,LOBS                                              DSPLIN
C                 - DATA ARRAYS AS DESCRIBED IN THE MAIN                      DSPLIN
C    FORFIV,NINTY - 45 DEG AND 90 DEG IN RADIANS                              DSPLIN
C                                                                             DSPLIN
C CALLS - PRNTCH,NMBR,ELLIPS,SPSYMB,NOWPLT,CIRCLE,DSHLNS,                     DSPLIN
C          ENDPLT ( FORTPLOT LIBRARY ROUTINES )                               DSPLIN
C       - PLOT,ELIPSE,POINT,RADD                                              DSPLIN
C       - DSORT                                                               DSPLIN
C ADDITIONAL NOTES                                                            DSPLIN
C    THE INPUT OF THE DATA POINTS ARE STORED IN THE LOBS                      DSPLIN
C    MATRIX BET AND INCLUDING THE 5ND ELEMENT  AND (2*NO+4)                  DSPLIN
C    ELEMENT . ARRAYS SIZES ARE TRANSFER THROUGH COMMON BLOCKS               DSPLIN
C                            *******                                          DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV           DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,           DSPLIN
     @         IDCOF,IDVCOF                                                   DSPLIN
C* ARRAYS                                                                     DSPLIN
      REAL*8 TK(IDTK,1),TX(IDTX,1),COF(IDCOF,8),VCOF(IDVCOF,8,8),            DSPLIN
     @       DTK(IDDTK,1),VAR(IDVAR,2,2),LOBS(IDLOBS,1),                     DSPLIN
     @       NINTY                                                            DSPLIN
      REAL*4 ZDASHG(2)                                                        DSPLIN
      INTEGER SPLIN,PRGEN,PRPTS,PRKNOT,DRAW                                   DSPLIN
C* APPLY SCALE FACTOR ON PLOTTED ELLIPSES                                      DSPLIN
      DATA SCALEL/1.0D0/                                                      DSPLIN
C*                                                                            DSPLIN
C* PROGRAM BEGINS                                                             DSPLIN
C* SET UP PLOT AREA FOR ONCE ONLY                                             DSPLIN
      IF(ICOUNT.EQ.1.AND.SPLIN.EQ.1)                                          DSPLIN
     @            CALL PLOT(ZIHF,ZIHC,ZIDPX,ZIDPY,ZX1,ZY1,ZX2,ZY2)           DSPLIN
      IF(SPLIN.NE.1)                                                          DSPLIN
     @            CALL PLOT(ZIHF,ZIHC,ZIDPX,ZIDPY,ZX1,ZY1,ZX2,ZY2)           DSPLIN
      IF(PRPTS.EQ.0)GO TO 100                                                 DSPLIN
      CALL PRNTCH('X')                                                        DSPLIN
      L = NO + 1                                                              DSPLIN
      J = 3                                                                   DSPLIN
      INUMP = 1                                                               DSPLIN
      DO 10 I = 2,L                                                           DSPLIN
      INUM = I-1                                                              DSPLIN
      J = J + 2                                                               DSPLIN
      J1 = J + 1                                                              DSPLIN
C* SELECT OPTION                                                              DSPLIN
      ZX = LOBS(J,1)                                                          DSPLIN
      ZY = LOBS(J1,1)                                                         DSPLIN
      IF(PRPTS.EQ.2)ZY = ZX                                                   DSPLIN
      IF(PRPTS.EQ.2.OR.PRPTS.EQ.3)ZX = TX(I,1)/SCTIME                        DSPLIN
C* PRINT NUMBER                                                               DSPLIN
```

```
      IF(PRPTS.EQ.2.OR.PRPTS.EQ.3)GO TO 30                            DSPLIN
      ZRX = ZX - ZIDPX                                                DSPLIN
      ZRY = ZY - ZIDPY                                                DSPLIN
      IF(INUM.NE.INUMP)GO TO 30                                       DSPLIN
      IF(INUM.EQ.1)INUMP = INUMP + 19                                 DSPLIN
      IF(INUM.NE.1)INUMP = INUMP + 20                                 DSPLIN
      CALL NMBR(ZRX,ZRY,INUM,'(I3)',0.0,ZIHF)                         DSPLIN
C* OMIT ELLIPSE ETC IF ON LINE PRINTER                                DSPLIN
 30   IF(ITYPE.NE.0.OR.(PRPTS.NE.1.AND.PRPTS.NE.4))GO TO 20           DSPLIN
      CALL PRNTCH('.')                                                DSPLIN
      CALL ELIPSE(VAR(I,1,1),VAR(I,2,2),VAR(I,1,2),A,B,               DSPLIN
     @            ESC,DIR)                                            DSPLIN
      ZD = 3*FORFIV - DIR                                             DSPLIN
      IF(PRPTS.EQ.1.OR.PRPTS.EQ.4)CALL SPSYMB(ZX,ZY,-11,ZD,ZIHF)      DSPLIN
      ZDR = NINTY - DIR                                               DSPLIN
      ZA = A*SCALEL                                                   DSPLIN
      ZB = B*SCALEL                                                   DSPLIN
      IF(PRPTS.EQ.4)CALL ELLIPS(ZX,ZY,-ZA,-ZB,ZDR)                    DSPLIN
   20 IF(ITYPE.EQ.0.AND.(PRPTS.NE.1.AND.PRPTS.NE.4))                  DSPLIN
     @              CALL SPSYMB(ZX,ZY,-11,FORFIV,ZIHF)                DSPLIN
      IF(ITYPE.EQ.1)CALL NOWPLT(0,ZX,ZY)                              DSPLIN
   10 CONTINUE                                                        DSPLIN
C*                                                                    DSPLIN
C* PLOTTER CONTROL -PRINTING OF THE KNOTS                             DSPLIN
C*                                                                    DSPLIN
 100  CALL PRNTCH('*')                                                DSPLIN
      IF(PRKNOT.EQ.0)GO TO 200                                        DSPLIN
      DO 110 II = 1,KNO                                               DSPLIN
      IF(II.NE.KNO)GO TO 130                                          DSPLIN
      L = KNO-1                                                       DSPLIN
      ZX=COF(L,7)*(DTK(L,1)**3)+COF(L,5)*(DTK(L,1)**2) +              DSPLIN
     .COF(L,3)*DTK(L,1) + COF(L,1)                                    DSPLIN
      ZY = COF(L,8)*(DTK(L,1)**3) + COF(L,6)*(DTK(L,1)**2) +          DSPLIN
     .COF(L,4)*DTK(L,1) + COF(L,2)                                    DSPLIN
      GO TO 120                                                       DSPLIN
  130 ZX = COF(II,1)                                                  DSPLIN
      ZY = COF(II,2)                                                  DSPLIN
  120 IF(PRKNOT.EQ.2)ZY = ZX                                          DSPLIN
      IF(PRKNOT.EQ.2.OR.PRKNOT.EQ.3)ZX = TK(II,1)/SCTIME              DSPLIN
      IF(ITYPE.EQ.0)CALL CIRCLE(ZX,ZY,ZIHC)                           DSPLIN
      IF(ITYPE.EQ.0)CALL SPSYMB(ZX,ZY,-4,-0.785,ZIHF)                 DSPLIN
      IF(ITYPE.EQ.1)CALL NOWPLT(0,ZX,ZY)                              DSPLIN
  110 CONTINUE                                                        DSPLIN
C*                                                                    DSPLIN
C* DRAW THE SPLINE                                                    DSPLIN
C*                                                                    DSPLIN
 200  IF(DRAW.EQ.0)GO TO 999                                          DSPLIN
      IF(IER.NE.0)GO TO 999                                           DSPLIN
```

```
C* CORRECTION APPLIED TO THE TIME INTERVAL AND INPUT TIME                DSPLIN
      YINT = XINT*SCTIME                                                 DSPLIN
      YSTART = XSTART*SCTIME                                             DSPLIN
C* POSITIONS THE PEN AT THE FIRST KNOT                                   DSPLIN
      ZAX = COF(1,1)                                                     DSPLIN
      ZAY = COF(1,2)                                                     DSPLIN
             IF(DRAW.EQ.2)ZAY = ZAX                                      DSPLIN
             IF(DRAW.EQ.2.OR.DRAW.EQ.3) ZAX = YSTART/SCTIME              DSPLIN
      CALL NOWPLT(0,ZAX,ZAY)                                             DSPLIN
      CALL PRNTCH('.')                                                   DSPLIN
      INO = (TK(KNO,1)-YSTART)/YINT + 1                                  DSPLIN
      WRITE(6,9000)                                                      DSPLIN
C* BEGIN DRAWING                                                         DSPLIN
      DO 210 I = 1,INO                                                   DSPLIN
      TFIX = YSTART + (I-1)*YINT                                         DSPLIN
      CALL POINT(TFIX,XN,YN,VXN,VYN,VXY,ICODE,KNO,TK,COF,VCOF)           DSPLIN
      IF(ICODE.EQ.0)GO TO 210                                            DSPLIN
      SE = DSQRT(VXN)                                                    DSPLIN
      SN = DSQRT(VYN)                                                    DSPLIN
C* ESC IS SET TO PRODUCE THE ALPHA PERCENT ELLIPSE                        DSPLIN
      CALL ELIPSE(VXN,VYN,VXY,A,B,ESC,DIR)                               DSPLIN
      ZDR = NINTY - DIR                                                  DSPLIN
      ZA = A                                                             DSPLIN
      ZB = B                                                             DSPLIN
      ZAX = XN                                                           DSPLIN
      ZAY = YN                                                           DSPLIN
             IF(DRAW.EQ.2)ZAY = ZAX                                      DSPLIN
             IF(DRAW.EQ.2.OR.DRAW.EQ.3)ZAX = TFIX/SCTIME                 DSPLIN
      ZDASHG(1) = ICOUNT*0.05                                            DSPLIN
      ZDASHG(2) = 0.1                                                    DSPLIN
C* CONFINE PLOTTING TO WITHIN GRID AREA                                  DSPLIN
      IF(ZAX.GT.ZX2.OR.ZAX.LT.ZX1)GO TO 220                              DSPLIN
      IF(ZAY.GT.ZY2.OR.ZAY.LT.ZY1)GO TO 220                              DSPLIN
C* CHIOCES                                                               DSPLIN
      IF(ITYPE.EQ.1)GO TO 230                                            DSPLIN
      IF(DRAW.EQ.1)CALL DSHLNS(ZAX,ZAY,ZDASHG,-2)                        DSPLIN
      IF(DRAW.EQ.4)CALL ELLIPS(ZAX,ZAY,-ZA,-ZB,ZDR)                      DSPLIN
      IF(DRAW.EQ.4)CALL SPSYMB(ZAX,ZAY,-4,ZDR,ZIHF)                      DSPLIN
      IF(DRAW.EQ.2.OR.DRAW.EQ.3)CALL NOWPLT(1,ZAX,ZAY)                   DSPLIN
  230 IF(ITYPE.EQ.1)CALL NOWPLT(1,ZAX,ZAY)                               DSPLIN
  220 TFIX = TFIX/SCTIME                                                 DSPLIN
      DR = RADD(DIR)                                                     DSPLIN
      IF(PRGEN.EQ.1)WRITE(6,9010)I,TFIX,XN,YN,SE,SN,VXY,A,B,DR           DSPLIN
  210 CONTINUE                                                           DSPLIN
  999 IF(SPLIN.NE.1)CALL ENDPLT                                          DSPLIN
      RETURN                                                             DSPLIN
 9000 FORMAT(//,10X,'PREDICTED FIXES AT SPECIFIED TIMES',/10X,           DSPLIN
     .17('=='),//,3X,'NO',6X,'TIME',12X,                                 DSPLIN
```

```
     .'POSITION(E,N)                SIGMA POSN(E,N)',13X,' COVARIANCE',4X,     DSPLIN
     .'SEMI-MAJOR  SEMI-MINOR   ORIENT(DEG) ',//)                              DSPLIN
 9010 FORMAT(2X,I4,2X,F10.4,10(F12.3,2X))                                      DSPLIN
      END                                                                      DSPLIN
      SUBROUTINE ELIPSE(VXX,VYY,VXY,A,B,SC,PHI)                                DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                                 DSPLIN
C ******************************************************************          DSPLIN
C * VERSION     : JANUARY 1982                                       *          DSPLIN
C * DESCRIPTION : COMPUTE THE PARAMETERS OF AN ELLIPSE GIVEN         *          DSPLIN
C *               THE COVARIANCE MATRIX PARAMETERS.                  *          DSPLIN
C ******************************************************************          DSPLIN
C INPUT PARAMETERS                                                              DSPLIN
C     VXX - VARIANCE OF X                                                       DSPLIN
C     VYY - VARIANCE OF Y                                                       DSPLIN
C     VXY - COVARIANCE OF X AND Y                                               DSPLIN
C     SC  - THE FACTOR  (FOR ELLIPSES OTHER THAN STD ELLIPSES)                  DSPLIN
C                                                                               DSPLIN
C OUTPUT PARAMETERS                                                             DSPLIN
C     A,B - THE SEMI-MAJOR AND SEMI-MINOR AXES OF ELLIPSE                       DSPLIN
C     PHI - AZIMUTH OF THE MAJOR AXIS(IN RADIANS)                               DSPLIN
C                                                                               DSPLIN
C CALLS  - DSQRT,DATAN2                                                         DSPLIN
C                                                                               DSPLIN
      DATA PI/3.14159265358979300D0/                                           DSPLIN
      D1 = (VXX+VYY)/2.0D00                                                    DSPLIN
      D2 = DSQRT((VXX-VYY)**2/4.0D00 + VXY**2)                                 DSPLIN
      A = DSQRT(D1+D2)*SC                                                      DSPLIN
      B = DSQRT(D1-D2)*SC                                                      DSPLIN
      IF(D2.LT.1.0D-20)PHI = 0.0D00                                            DSPLIN
      IF(D2.LT.1.0D-20)GO TO 10                                                DSPLIN
      IF(VXX.LT.1.0D-20.AND.VYY.LT.1.0D-20)PHI = 0.0D00                        DSPLIN
      IF(VXX.LT.1.0D-20.AND.VYY.LT.1.0D-20)GO TO 10                            DSPLIN
      PHI = -0.5*DATAN2(-2.0*VXY,VYY-VXX)                                      DSPLIN
      IF(PHI.LT.0.0D00)PHI = PHI + 2.0D00*PI                                   DSPLIN
   10 RETURN                                                                   DSPLIN
      END                                                                      DSPLIN
      SUBROUTINE ELLSIG(A,B,DIR,SE,SN,SEN,ICODE)                               DSPLIN
C *****************************************************                       DSPLIN
C * VERSION     : 27 JUNE 1982                          *                       DSPLIN
C * DESCRIPTION : TRANSFORMS FROM ELLIPSE TO STANDARD   *                       DSPLIN
C *               DEVIATIONS AND COVARIANCE             *                       DSPLIN
C *****************************************************                       DSPLIN
C INPUT PARAMTERS                                                               DSPLIN
C  A,B     - SEMI-MAJOR AND SEMI MINOR AXES OF ELLIPSE                          DSPLIN
C  DIR     - AZIMUTH OF SEMI MAJOR AXIS                                         DSPLIN
C  ICODE   - 0 : AZIMUTH IS IN DEGREES                                          DSPLIN
C            1 : AZIMUTH IS IN RADIANS                                          DSPLIN
C                                                                               DSPLIN
```

```
C OUTPUT PARAMTERS
C   SE      - STANDARD DEVIATION IN THE EASTINGS                    DSPLIN
C   SN      - STANDARD DEVIATIONS IN THE NORTHINGS                  DSPLIN
C   SEN     - COVARIANCE BETWEEN EASTINGS AND NORTHINGS             DSPLIN
C                                                                   DSPLIN
C CALLS    - DSQRT,DTAN                                             DSPLIN
C                          **************                           DSPLIN
      REAL*8 A,B,DIR,SE,SN,SEN,DR,K,C,GAMMA,PHI                     DSPLIN
      DATA PHI/3.141592654D00/                                      DSPLIN
      DR = DIR                                                      DSPLIN
      IF(ICODE.EQ.0)DR = (DIR/180.0D0)*PHI                          DSPLIN
      GAMMA = 2.0D0*(PHI/2.0D0 - DR)                                DSPLIN
      K = PHI*2.0D0                                                 DSPLIN
      IF(GAMMA.GT.K)GAMMA = GAMMA - K                               DSPLIN
      SEN = 0.5D0*(A*A - B*B)*DSIN(GAMMA)                           DSPLIN
      K = A*A + B*B                                                 DSPLIN
      C = 2.0D0*SEN/DTAN(GAMMA)                                     DSPLIN
      SE = DSQRT((K+C)/2.0D0)                                       DSPLIN
      SN = DSQRT((K-C)/2.0D0)                                       DSPLIN
      RETURN                                                        DSPLIN
      END                                                           DSPLIN
      REAL FUNCTION ELSFAC*8(ALPHA)                                 DSPLIN
      REAL*8 ALPHA                                                  DSPLIN
C ****************************************************************** DSPLIN
C * VERSION     : JUNE 1982                                       * DSPLIN
C * DESCRIPTION : TO COMPUTE THE SCALE FACTOR FOR THE CONFIDENCE  * DSPLIN
C *               ELLIPSES AT THE ALPHA PERCENTAGE LEVEL.         * DSPLIN
C *               IF ERROR DETECTED, SCALE FACTOR IS SET TO 1.0D0 * DSPLIN
C ****************************************************************** DSPLIN
C CALLS  - MDCHI (IMSL ROUTINE)                                     DSPLIN
C        - DSQRT,DBLE                                               DSPLIN
C                                                                   DSPLIN
      ZX = ALPHA/100.0D0                                            DSPLIN
      CALL MDCHI(ZX,2.0,Y,IER)                                      DSPLIN
      IF(IER.EQ.0)ELSFAC = DSQRT(DBLE(Y))                           DSPLIN
      IF(IER.NE.0)ELSFAC = 1.0D0                                    DSPLIN
C*                                                                  DSPLIN
      RETURN                                                        DSPLIN
      END                                                           DSPLIN
      SUBROUTINE ERRATR(XVELO,AVELO,ELEV,XTER)                      DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                      DSPLIN
C *********************************************************        DSPLIN
C * VERSION     : AUGUST 1982                            *          DSPLIN
C * DESCRIPTION : COMPUTES THE ALONG TRACK ERROR         *          DSPLIN
C *               DUE AN  ERROR IN RECEIVER MOTION       *          DSPLIN
C *               BASED ON THE MAGNITUDE OF THE          *          DSPLIN
C *               MOTION AND THE ELEVATION OF THE        *          DSPLIN
C *               SATELLITE.                             *          DSPLIN
```

```
C ************************************************************           DSPLIN
C INPUT PARAMETERS                                                       DSPLIN
C  AVELO  - ERROR IN VELOCITY OF THE RECEIVER ALONG-TRACK DIRECTION      DSPLIN
C  XVELO  - ERROR IN VELOCITY OF THE RECEIVER CROSS-TRACK DIRECTION      DSPLIN
C  ELEV   - ELEVATION OF THE TRACKED SATELLITE                           DSPLIN
C                                                                        DSPLIN
C OUTPUT PARAMETERS                                                      DSPLIN
C  XTER   - COMPUTED RSS ALONG TRACK ERROR                                DSPLIN
C                                                                        DSPLIN
C CALLS  - DSQRT                                                         DSPLIN
C                                                                        DSPLIN
      ELEVAT = (ELEV/3.141592654)*180.0D0                                DSPLIN
C ERROR DUE TO VELOCITY NORTH ERRORS                                     DSPLIN
      XTER1 =  0.03D0*AVELO                                              DSPLIN
      IF(ELEVAT.GT.75 )XTER1 =  0.04D0*AVELO                             DSPLIN
      IF(ELEVAT.GT.80 )XTER1 =  0.075D0*AVELO                            DSPLIN
      IF(ELEVAT.GT.83 )XTER1 =  0.09D0*AVELO                             DSPLIN
      IF(ELEVAT.GT.85 )XTER1 =  0.10D0*AVELO                             DSPLIN
      IF(ELEVAT.GT.87 )XTER1 =  0.13D0*AVELO                             DSPLIN
C ERROR DUE TO VELOCITY EAST ERRORS                                      DSPLIN
      XTER2 = 0.02D0*XVELO                                               DSPLIN
      IF(ELEVAT.GE.75)XTER2 = 0.02D0*XVELO                               DSPLIN
      IF(ELEVAT.GE.80)XTER2 = 0.015D0*XVELO                              DSPLIN
      IF(ELEVAT.GE.85)XTER2 = 0.01D0*XVELO                               DSPLIN
C ERROR DUE TO INCORRECT ALTITUDE                                        DSPLIN
      XTER3 = 12*0.25                                                    DSPLIN
      IF(ELEVAT.GE.75)XTER3 = 12*0.35                                    DSPLIN
      IF(ELEVAT.GE.80)XTER3 = 12*0.50                                    DSPLIN
      IF(ELEVAT.GE.85)XTER3 = 12*0.75                                    DSPLIN
      IF(ELEVAT.GE.87)XTER3 = 12*1.00                                    DSPLIN
C CUMMULATIVE ERROR                                                      DSPLIN
      XTER = DSQRT(XTER1*XTER1 + XTER2*XTER2 + XTER3*XTER3)              DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE ERRORS(E0,N0,SE,SN,SEN,VELO,DIR,ICODE)                  DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                           DSPLIN
C ***********************************************************************DSPLIN
C * VERSION      : 05 JULY 1982                                         *DSPLIN
C * DESCRIPTION  : IF THE UNMODELLED ERRORS DUE TO VELOCITY AND         *DSPLIN
C *                ALTITUDE ERRORS EXCEED THAT OF THE FORMAL ERROR      *DSPLIN
C *                ESTIMATES, THE STD DEVIATIONS ARE ADJUSTED ACCORDING.*DSPLIN
C ***********************************************************************DSPLIN
C INPUT PARAMETERS                                                       DSPLIN
C  E0,N0    -  POSITION AS DETERMINED BY THE METHOD OF LEAST SQUARES     DSPLIN
C  SE,SN    -  STANDARD DEVIATIONS IN EASTING AND NORTHING RESP.         DSPLIN
C  SEN      -  COVARIANCE                                                DSPLIN
C  VELO     -  ESTIMATED VELOCITY OF THE RECEIVER                        DSPLIN
C  DIR      -  AZIMUTH(IN RADIANS) OF THE MOTION OF RECEIVER             DSPLIN
```

```
      C
      C OUTPUT PARAMETERS                                                      DSPLIN
      C   E0,N0,                                                               DSPLIN
      C   VELO,DIR   -   UNCHANGED                                             DSPLIN
      C   SE,SN,                                                               DSPLIN
      C   SEN          - MODIFIED BY A FACTOR DETERMINED BY THE ROUTINE.      DSPLIN
      C                                                                        DSPLIN
      C CALLS - ELIPSE,SATELV,ERRXTR,ERRATR,ELLSIG                             DSPLIN
      C       - DCOS,DSIN                                                      DSPLIN
      C                       ***********                                      DSPLIN
            REAL*8 E0,N0,NINTY,FULC                                            DSPLIN
            DATA PHI/3.14159265400/,NINTY/1.570796327D0/,FULC/6.283185308D0/   DSPLIN
            VX = SE*SE                                                         DSPLIN
            VY = SN*SN                                                         DSPLIN
            CALL ELIPSE(VX,VY,SEN,A,B,1.0D0,DRELL)                             DSPLIN
      C* COMPUTE ANGLE BETWEEN SATELLITE DIRECTION AND MOTION                  DSPLIN
            DRS = DRELL + NINTY                                                DSPLIN
            IF(DRS.GT.FULC)DRS = DRS - FULC                                    DSPLIN
            DR = DRS                                                           DSPLIN
            IF(DRS.LT.DIR)DR = DRS + FULC                                      DSPLIN
            ADR = DR - DIR                                                     DSPLIN
            IF(ADR.GE.PHI)ADR    = ADR - PHI                                   DSPLIN
            IF(ADR.GT.NINTY)ADR = PHI - ADR                                    DSPLIN
      C* VELOCITY COMPONENT OF RECEIVER IN THE CROSS TRACK AND ALONG TRACK     DSPLIN
      C* DIRECTION OF THE SATELLITE                                            DSPLIN
            AVELO = VELO*DCOS(ADR)                                             DSPLIN
            XVELO = VELO*DSIN(ADR)                                             DSPLIN
            CALL SATELV(E0,N0,2.0D6,2.0D6,DRS,ELEV)                            DSPLIN
            CALL ERRXTR(XVELO,AVELO,ELEV,AB)                                   DSPLIN
            CALL ERRATR(XVELO,AVELO,ELEV,BA)                                   DSPLIN
            DR = RADD(DRS)                                                     DSPLIN
            ELEV = RADD(ELEV)                                                  DSPLIN
            IF(ICODE.EQ.1)WRITE(6,9000)DR,ELEV,A,B,AB,BA                       DSPLIN
            IF((AB.LT.A.AND.BA.LT.B).AND.ICODE.EQ.1)WRITE(6,9010)              DSPLIN
            IF(AB.LT.A.AND.BA.LT.B)RETURN                                      DSPLIN
      C* ERROR ELLIPSES MODIFIED ACCORDING TO RSS ERRORS                       DSPLIN
            IF(AB.GT.A)A = AB                                                  DSPLIN
            IF(BA.GT.B)B = BA                                                  DSPLIN
            CALL ELLSIG(A,B,DRELL,SE,SN,SEN,1)                                 DSPLIN
            IF(ICODE.EQ.1)WRITE(6,9020)SN,SE,SEN,A,B                           DSPLIN
            RETURN                                                             DSPLIN
       9000 FORMAT(/,3X,'AZIMUTH OF SATELLITE AT TCA ',F6.1,' DEGS',/,         DSPLIN
           @          3X,'ELEVATION OF SATELLITE      ',F6.1,' DEGS',/,        DSPLIN
           @          3X,'INPUT ERROR ELLIPSE;A AND B ',D12.5,2X,D12.5,' M',   DSPLIN
           @        /,3X,'RSS  TRACK ERROR; XTR & ALTR',D12.5,2X,D12.5,' M')   DSPLIN
       9010 FORMAT(3X,'RSS ERRORS WITHIN ERROR ELLIPSE',/)                     DSPLIN
       9020 FORMAT(3X,'RSS ERRORS EXCEEDS ERROR ELLIPSE',                      DSPLIN
           @     /,3X,'NEW STD DEV(N,E) AND COV ',3(F10.1,3X),                 DSPLIN
```

```
     @       /,3X,'NEW ERROR ELLIPSE ;A & B ',3(F10.1,3X),//)               DSPLIN
      END                                                                   DSPLIN
      SUBROUTINE ERRXTR(XVELO,AVELO,ELEV,XTER)                              DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                              DSPLIN
C ***************************************************************           DSPLIN
C * VERSION       : 15 JULY 1982                                *           DSPLIN
C * DESCRIPTION : COMPUTES THE CROSS TRACK ERROR                *           DSPLIN
C *               DUE AN  ERROR IN RECEIVER MOTION              *           DSPLIN
C *               BASED ON THE MAGNITUDE OF THE                 *           DSPLIN
C *               MOTION AND THE ELEVATION OF THE               *           DSPLIN
C *               SATELLITE.                                    *           DSPLIN
C ***************************************************************           DSPLIN
C INPUT PARAMETERS                                                          DSPLIN
C  AVELO  - ERROR IN VELOCITY OF THE RECEIVER ALONG TRACK DIRECTION         DSPLIN
C  XVELO  - ERROR IN VELOCITY OF THE RECEIVER CROSS TRACK DIRECTION         DSPLIN
C  ELEV   - ELEVATION OF THE TRACKED SATELLITE                              DSPLIN
C                                                                           DSPLIN
C OUTPUT PARAMETERS                                                         DSPLIN
C  XTER  - COMPUTED RSS CROSS TRACK ERROR                                   DSPLIN
C                                                                           DSPLIN
C CALLS  - DSQRT                                                            DSPLIN
C                                                                           DSPLIN
       ELEVAT = (ELEV/3.141592654)*180.0D0                                  DSPLIN
C* ERROR DUE TO VELOCITY NORTH ERRORS                                       DSPLIN
       XTER1 =  0.45D0*AVELO                                                DSPLIN
       IF(ELEVAT.GT.75)XTER1 =  0.56D0*AVELO                                DSPLIN
       IF(ELEVAT.GT.80)XTER1 =  0.72D0*AVELO                                DSPLIN
       IF(ELEVAT.GT.84)XTER1 =  1.00D0*AVELO                                DSPLIN
       IF(ELEVAT.GT.85)XTER1 =  1.20D0*AVELO                                DSPLIN
C* GRAPH OR LINE FIX TREATMENT IF > 88 DEGREES IN SAT ELEVATION ?           DSPLIN
CS     IF(ELEVAT.GE.88)XTER1 =  1.50D0*AVELO                                DSPLIN
       IF(ELEVAT.GE.88)XTER1 =  3.00D3                                      DSPLIN
C* ERROR DUE TO VELOCITY EAST ERRORS                                        DSPLIN
       XTER2 = 0.03D0*XVELO                                                 DSPLIN
       IF(ELEVAT.GT.75  )XTER2 = 0.03D0*XVELO                               DSPLIN
       IF(ELEVAT.GT.80  )XTER2 = 0.05D0*XVELO                               DSPLIN
       IF(ELEVAT.GT.85  )XTER2 = 0.09D0*XVELO                               DSPLIN
       IF(ELEVAT.GT.87  )XTER2 = 0.12D0*XVELO                               DSPLIN
C* ERRORS DUE TO INCORRECT ALTITUDE                                         DSPLIN
       XTER3 = 3.30D0*12                                                    DSPLIN
       IF(ELEVAT.GT.75  )XTER3 = 4.20D0*12                                  DSPLIN
       IF(ELEVAT.GT.80  )XTER3 = 5.60D0*12                                  DSPLIN
       IF(ELEVAT.GT.83  )XTER3 = 7.20D0*12                                  DSPLIN
       IF(ELEVAT.GT.85  )XTER3 = 10.00D0*12                                 DSPLIN
       IF(ELEVAT.GT.86  )XTER3 = 15.00D0*12                                 DSPLIN
       XTER = DSQRT(XTER1*XTER1 + XTER2*XTER2 + XTER3*XTER3)                DSPLIN
       RETURN                                                               DSPLIN
       END                                                                  DSPLIN
```

```
      SUBROUTINE FILTER(NO,TX,LOBS,P,VAR,FINT,FINC,ALPHA,XSC,SCTIME,      DSPLIN
     @                  A,PF,W,V,VX,WK)                                    DSPLIN
      IMPLICIT REAL*8 (A-H,P-Z)                                            DSPLIN
C ***************************************************************         DSPLIN
C * VERSION     : 23 APRIL 1982                                   *         DSPLIN
C * DESCRIPTION : DETECTION OF UNDESIRABLES USING A MOVING        *         DSPLIN
C *               LINEAR FILTER                                   *         DSPLIN
C ***************************************************************         DSPLIN
C INFUT PARAMETERS                                                         DSPLIN
C  NO      - NO OF DATA POINTS TO BE PROCESSED                             DSPLIN
C  TX      - DATA TIMES                                                    DSPLIN
C  LOBS,P  - OBSERVATION VECTOR AND WEIGHTS                                DSPLIN
C  VAR     - INPUT VARIANCES OF DATA POINTS                                DSPLIN
C                                                                          DSPLIN
C  A) FINT  - FILTER INTERVAL WIDTH                                        DSPLIN
C  B) FINC  - FILTER INTERVAL WIDTH INCREMENT                              DSPLIN
C  C) ALPHA - PERCENTAGE TEST LEVEL                                        DSPLIN
C                                                                          DSPLIN
C  XSC     - SCALE FACTOR AT ALPHA % CONFIDENCE INTERVAL                   DSPLIN
C                                                                          DSPLIN
C OUTPUT PARAMETERS                                                        DSPLIN
C  NONE                                                                    DSPLIN
C                                                                          DSPLIN
C WORK ARRAYS                                                              DSPLIN
C  A,PF,W,V,VX - AS DIMENSIONED BELOW                                      DSPLIN
C  WK          - WORK ARRAY AS REQUESTED BY SOLV ROUTINE                   DSPLIN
C                                                                          DSPLIN
C CALLS - SOLV                                                             DSPLIN
C       - DSQRT,DABS                                                       DSPLIN
C                         *************                                    DSPLIN
      COMMON /DIM1/ICBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV         DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,         DSPLIN
     @           IDCOF,IDVCOF                                              DSPLIN
      REAL*8 TX(IDTX,1),LOBS(IDLOBS,1),P(IDP,3),VAR(IDVAR,2,2),            DSPLIN
     @       WK(8,56),                                                     DSPLIN
     @       A(20,3),PF(40,3),W(40,1),V(40,1),X(4,1),VX(4,4)               DSPLIN
      INTEGER PCODE                                                        DSPLIN
C*BEGINS                                                                   DSPLIN
C* SET DIMENSIONS OF FILTER ARRAYS                                         DSPLIN
C* MAXIMUM NUMBER OF POINTS EQUAL TO 20                                    DSPLIN
      IMAX = 20                                                            DSPLIN
      IA = IMAX                                                            DSPLIN
      IPF = IMAX*2                                                         DSPLIN
      IW = IMAX*2                                                          DSPLIN
      IV = IMAX*2                                                          DSPLIN
      IX = 4                                                               DSPLIN
      IVX = 4                                                              DSPLIN
C* STORE                                                                   DSPLIN
```

```
      ISOLD = 0                                                         DSPLIN
      IEOLD = 0                                                         DSPLIN
C* CORRECT INTERVALS FOR TIME SCALING                                   DSPLIN
      DINT = FINT*SCTIME                                                DSPLIN
      DINC = FINC*SCTIME                                                DSPLIN
C* OUTPUT INFORMATION                                                   DSPLIN
      WRITE(6,750)ALPHA,XSC                                             DSPLIN
      WRITE(6,700)DINT,DINC                                             DSPLIN
C* BEGIN FILTER                                                         DSPLIN
      TBG = TX(1,1)                                                     DSPLIN
      ISTOP = 0                                                         DSPLIN
      TED = TBG + DINT                                                  DSPLIN
 1000 L = NO + 1                                                        DSPLIN
      IF(TED.GT.TX(L,1))ISTOP = 1                                       DSPLIN
C* SET SWITCHES                                                         DSPLIN
      ISWT = 0                                                          DSPLIN
      ISWT2 = 1                                                         DSPLIN
      L = NO + 1                                                        DSPLIN
      DO 40 I = 2,L                                                     DSPLIN
      IF(TBG.LE.TX(I,1).AND.TED.GE.TX(I,1))GO TO 10                     DSPLIN
      IF(ISWT2.EQ.1)GO TO 40                                            DSPLIN
      IEND = I                                                          DSPLIN
      GO TO 50                                                          DSPLIN
   10 IF(ISWT.EQ.0)ISTART = I                                           DSPLIN
      ISWT = 1                                                          DSPLIN
      ISWT2 = 0                                                         DSPLIN
   40 CONTINUE                                                          DSPLIN
      IEND = NO + 2                                                     DSPLIN
C* CORRECT LAST POINTER                                                 DSPLIN
 50   IEND = IEND - 1                                                   DSPLIN
C* GENERATE MATRICES                                                    DSPLIN
      J = -1                                                            DSPLIN
      L = 0                                                             DSPLIN
      IF(ISOLD.EQ.ISTART.AND.IEOLD.EQ.IEND)GO TO 3000                   DSPLIN
      DO 100 I = ISTART,IEND                                            DSPLIN
      J = J + 2                                                         DSPLIN
      J1 = J + 1                                                        DSPLIN
      L = L + 1                                                         DSPLIN
      DT = TX(I,1) - TBG                                                DSPLIN
      A(L,1) = DT                                                       DSPLIN
      A(L,2) = 0.0D0                                                    DSPLIN
      A(L,3) = 1.0D0                                                    DSPLIN
      IL = (I-1)*2 + 3                                                  DSPLIN
      IL2 = (I-1)*2 + 4                                                 DSPLIN
      W(J,1) = -LOBS(IL,1)                                              DSPLIN
      W(J1,1) = -LOBS(IL2,1)                                            DSPLIN
C* P MATRIX LOCATE                                                      DSPLIN
      DO 110 K = 1,3                                                    DSPLIN
```

```
      PF(J,K) = P(IL,K)                                                 DSPLIN
  110 PF(J1,K) = P(IL2,K)                                               DSPLIN
  100 CONTINUE                                                          DSPLIN
C*                                                                      DSPLIN
C* SET PROBLEM SIZE                                                     DSPLIN
C*                                                                      DSPLIN
      N = J1                                                            DSPLIN
      M = 4                                                             DSPLIN
      PCODE = 0                                                         DSPLIN
      CALL SOLV(A,PF,W,X,VX,V,N,M,IA,3,IPF,3,IW,IX,IVX,IV,ICOD,PCODE,   DSPLIN
     a         WK)                                                      DSPLIN
      IF(ICOD.EQ.0)GO TO 200                                            DSPLIN
      WRITE(6,740)ISTART,IEND,ICOD                                      DSPLIN
      WRITE(6,9000)                                                     DSPLIN
 9000 FORMAT(/,3X,'INSUFFICIENT POINTS; INCREASING INTERVAL',/)         DSPLIN
      TED = TED + DINC                                                  DSPLIN
      IF(ISTOP.EQ.1)WRITE(6,9010)                                       DSPLIN
 9010 FORMAT(/,3X,'NO POINTS LEFT - TERMINATE HERE',/)                  DSPLIN
      IF(ISTOP.EQ.1)GO TO 2000                                          DSPLIN
      GO TO 1000                                                        DSPLIN
C*                                                                      DSPLIN
C* TESTING THE RESIDUALS IF SOLV PASSES                                 DSPLIN
C*                                                                      DSPLIN
 200  ITOTAL = 0                                                        DSPLIN
C* CORRECT OUTPUT POINTERS                                              DSPLIN
      J = ISTART -1                                                     DSPLIN
      J1 = IEND - 1                                                     DSPLIN
      WRITE(6,710)J,J1                                                  DSPLIN
      DO 210 I = ISTART,IEND                                            DSPLIN
C* CORRECT OUTPUT POINTERS AND MODIFY TIME OUTPUT                        DSPLIN
         II = I -ISTART + 1                                             DSPLIN
         ITRUE = I-1                                                    DSPLIN
         TIME = TX(I,1)/SCTIME                                          DSPLIN
         XLEVEL = XSC*DSQRT(VAR(I,1,1))                                 DSPLIN
         IF(DABS(V(II,1)).LT.XLEVEL)GO TO 220                           DSPLIN
         L = DABS(V(II,1))/XLEVEL                                       DSPLIN
         WRITE(6,720)ITRUE,TIME,L                                       DSPLIN
         ITOTAL = ITOTAL + 1                                            DSPLIN
  220    XLEVEL = XSC*DSQRT(VAR(I,2,2))                                 DSPLIN
         II = II + 1                                                    DSPLIN
         IF(DABS(V(II,1)).LT.XLEVEL)GO TO 210                           DSPLIN
         L = DABS(V(II,1))/XLEVEL                                       DSPLIN
         WRITE(6,730)ITRUE,TIME,L                                       DSPLIN
  210  CONTINUE                                                         DSPLIN
 3000 IF(ISTOP.EQ.1)GO TO 2000                                          DSPLIN
      ISOLD = ISTART                                                    DSPLIN
      IEOLD = IEND                                                      DSPLIN
      TBG = TBG + DINC                                                  DSPLIN
```

```
      TED = TBG + DINT                                                  DSPLIN
      GO TO 1000                                                        DSPLIN
 2000 WRITE(6,9100)                                                     DSPLIN
 9100 FORMAT(3X,'FILTER TESTS COMPLETED',/)                             DSPLIN
 700  FORMAT(/,/,3X,'SUB FILTER DETECT THESE POINTS AS OUTLIERS',       DSPLIN
     @ /,3X,'TIME SPAN ',1PD12.5,' INCREMENT WIDTH ',1PD12.5,/,/)       DSPLIN
 710  FORMAT(/,3X,'POINT SPAN  < ',I3,',',I3,'>',/)                     DSPLIN
 720  FORMAT(3X,'NO',I4,' TIME ',1PD12.5,' E RESIDUAL FAILED',          DSPLIN
     @        ' (SEVERITY : ',I2,' )')                                  DSPLIN
 730  FORMAT(3X,'NO',I4,' TIME ',1PD12.5,' N RESIDUAL FAILED',          DSPLIN
     @        ' (SEVERITY : ',I2,' )')                                  DSPLIN
 740  FORMAT(/,3X,'POINT SPAN  < ',I3,',',I3,'> ERROR CODE :',I2,/)     DSPLIN
 750  FORMAT(/,3X,'SCALE FACTOR FOR ',F6.1,'  PERCENT CONFIDENCE',      DSPLIN
     @ ' ELLIPSES = ',F10.4,/)                                          DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE FIT(TIME,XX,YY,VXY,IOPT,ICODE,IRJET,ALPKNT,            DSPLIN
     @               A,P,VAR,VX,VXOLD,LOBS,WKS)                         DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                          DSPLIN
C ********************************************************************* DSPLIN
C *  DESCRIPTION OF ROUTINE                                           * DSPLIN
C *  VERSION     : 12 JULY  1982                                      * DSPLIN
C *  DEPLOYMENT : CURVE FITTING                                       * DSPLIN
C *  QUALITIES   : HAS THE POSITION AND 1ST DERIVATIVE OF THE         * DSPLIN
C *                FORMER KNOT WEIGHTED : VARIABLE WEIGHTING ON SLOPE * DSPLIN
C *                                       AND POSITION OF KNOT IS USED * DSPLIN
C *  METHOD      : SEE THE OPTION SET FOR ANALYSIS OF PROGRAM         * DSPLIN
C *                CAPABILITIES                                       * DSPLIN
C ********************************************************************* DSPLIN
C  INPUT PARAMETERS                                                     DSPLIN
C   TIME    - TIME OF DATA POINT                                        DSPLIN
C   XX,YY   - COORDINATES OF DATA POINT                                 DSPLIN
C   VXY     - COVARIANCE MATRIX OF DATA POINTS                          DSPLIN
C                                                                       DSPLIN
C   IOPT    - VECTOR OF OPTIONS GOVERNING THE CURVE FITTING             DSPLIN
C*   A) IOPT(1)  : MIN NUMBER OF DATA POINTS BEFORE DOING THE FIRST     DSPLIN
C                  CUBIC POLYNOMINAL FIT                                DSPLIN
C*   B) IOPT(2)  : LOCATION OF KNOT  = 1 MIDWAY BET TWO DATA POINTS     DSPLIN
C                                    = 0 CONCIDENCE WITH THE DATA POINT DSPLIN
C*   C) IOPT(3)  : % OF INTERVAL OVERHANGING .IE. % OF PTS              DSPLIN
C                  OUTSIDE THE INTERVAL WITH THE INTERVAL AS 100 %      DSPLIN
C*   D) IOPT(4)  : MAGNIFICATION FACTOR ON THE SLOPE VARIANCE           DSPLIN
C                  IF <1 THEN NO WEIGHT  ; 100 ACTUAL COMPUTED VARIANCE DSPLIN
C                  >100 INCREASE IN WEIGHT ; 1<A) 4)<100 % WT.          DSPLIN
C*   E) IOPT(5)  : MIN  NUMBER OF DATA POINTS PER INTERVAL PERMITTED    DSPLIN
C                                                                       DSPLIN
C*   F) IOPT(6)  : INCLUSION OF FORWARD OUTER POINTS IN REJECTION TESTS DSPLIN
C                  IF  1 - YES : 0 - NO                                 DSPLIN
```

```
C*    G) IOPT(7)   : MINIMUM LAG OF ONE POINT TO BE ENFORCED             DSPLIN
C                       1 - YES    OR   0 - NC                           DSPLIN
C*    H) IOPT(8)   : ALL DIAGONSTIC PRINTOUT : 1 - YES : 0 - NO          DSPLIN
C                                                                        DSPLIN
C*    I) IOPT(9)   : PRINT COMPUTED KNOTS : 1 - YES ; 0 - NO             DSPLIN
C                                                                        DSPLIN
C*    J) IOPT(10) : PRINT COMPUTED KNCTS COVAR MATRIX : 1 - YES ; 0 - NO DSPLIN
C                                                                        DSPLIN
C*    K) IRJET   - NO OF FAILURES ALLOWED                                DSPLIN
C*    L) ALPKNT  - X LEVEL OF SIGNIFICANCE TESTS                         DSPLIN
C                                                                        DSPLIN
C OUTPUT PARAMETERS                                                      DSPLIN
C    ICODE   = 0 POINT READ AND PROCESSED BUT KNOT NOT RECOMMENDED       DSPLIN
C            = 1 FATAL ERROR DETECTED IN THE SUBROUTINE                  DSPLIN
C            = 2 KNOT ESTABLISHED.                                       DSPLIN
C  IF AND ONLY IF ICODE = 2                                              DSPLIN
C    TIME    = NEXT KNOT TIME                                            DSPLIN
C                                                                        DSPLIN
C WORK ARRAYS - A,P,VAR,VX,VXOLD,LOBS ( AS DIM BELOW )                   DSPLIN
C             - WKS (WORK ARRAY AS REQUESTED BY SOLV ROUTINE )           DSPLIN
C                                                                        DSPLIN
C CALLS - OBS,CHOLD,REJECT                                               DSPLIN
C ADDITIONAL NOTES                                                       DSPLIN
C  IOPT(1) MUST BE AT LEAST (100 + IOPT(3)) PERCENT LARGER               DSPLIN
C     THAN IOPT(5).                                                      DSPLIN
C  ALL POINTERS IN COMMON BELOW HAVE TO BE RESET BEFORE                  DSPLIN
C     BEGINNING TO LOCATE OPTIMUM KNCTS.                                 DSPLIN
C  7 JULY 1982 - UPGRADE TO 100 PTS PER INTERVAL                         DSPLIN
      COMMON /INIT/TK,ISWT2,ISWT,INO,PTX,PLOBS,PVAR                      DSPLIN
C                                                                        DSPLIN
C* ARRAYS FEED INTO THE PROGRAM                                          DSPLIN
      REAL*8 VXY(2,2)                                                    DSPLIN
C* ARRAYS USED BY THE PROGRAM                                            DSPLIN
C* STORAGE OF ONLY INPUT DATA POINTS                                     DSPLIN
      REAL*8 TX(100,1),LOBS(200,1),VAR(100,2,2)                          DSPLIN
C* TEMPORARY                                                             DSPLIN
      REAL*8 TK(2,1),AOBS(7,1),X(8,1),VX(8,8),XOLD(8,1),VXOLD(8,8)       DSPLIN
C* ACCOUNTS FOR THE KNCT INFORMATION                                     DSPLIN
      REAL*8 A(102,7),P(204,3),W(204,1),V(204,1)                         DSPLIN
C* WORK ARRAY USED BY THE SOLV ROUTINE SEE SOLV FOR DIMENSIONS           DSPLIN
      REAL*8 WKS(8,204)                                                  DSPLIN
      INTEGER IOPT(10),PTX,PLOBS,PVAR                                    DSPLIN
C* ARRAY SIZES                                                           DSPLIN
C*  VARIABLE INOBS - MAXIMUM NUMBER OF OBSERVATIONS(PER INTERVAL)        DSPLIN
      INOBS  = 100                                                       DSPLIN
      IDTX   = INCBS                                                     DSPLIN
      IDLOBS = INOBS*2                                                   DSPLIN
      IDVAR  = INOBS                                                     DSPLIN
```

```
      IDA     =  INOBS + 2                                                DSPLIN
      IDP     =  INOBS*2 + 4                                              DSPLIN
      IDW     =  INOBS*2 + 4                                              DSPLIN
      IDV     =  INOBS*2 + 4                                              DSPLIN
      IDX     =  8                                                        DSPLIN
      IDVX    =  8                                                        DSPLIN
C* PROGRAM BEGINS BY CHECKING ON OPTIONS                                  DSPLIN
C* COMPUTE THE LAG IN THE INTERVAL FIT                                    DSPLIN
C* TRUNCATED TO THE INTEGER VALUE : MINUMUM VALUE DETERMINED BY THE       DSPLIN
C*  OPTION IOPT(7).                                                       DSPLIN
      LAG = (INO*IOPT(3))/(100 + IOPT(3))                                 DSPLIN
      IF(LAG.LT.1.AND.IOPT(7).EQ.1)LAG = 1                                DSPLIN
      IL = IOPT(1) - LAG                                                  DSPLIN
      IF(ISWT2.EQ.0.AND.ISWT.EQ.0.AND.IL.LT.4)GO TO 9000                  DSPLIN
      IF(IOPT(4).EQ.0.AND.IOPT(5).LT.4)GO TO 9000                         DSPLIN
      IF(IOPT(5).LT.2)GO TO 9000                                          DSPLIN
C* CLEAR P MATRIX                                                         DSPLIN
       IF(ISWT2.EQ.0.AND.ISWT.EQ.0)CALL DCLEAR(P,IDP,3)                   DSPLIN
C* SET ERROR CODE                                                         DSPLIN
      ICODE = 0                                                           DSPLIN
C* INCREASE COUNTERS FOR NEXT POINT                                       DSPLIN
      IF(IOPT(8).EQ.1)WRITE(6,7005)                                       DSPLIN
      INO = INO + 1                                                       DSPLIN
      PTX = PTX + 1                                                       DSPLIN
      PLOBS = PLOBS + 1                                                   DSPLIN
      PVAR = PVAR + 1                                                     DSPLIN
C* STORAGE OF THE INPUT DATA                                              DSPLIN
      TX(PTX,1) = TIME                                                    DSPLIN
      LOBS(PLOBS,1) = XX                                                  DSPLIN
      PLOBS = PLOBS + 1                                                   DSPLIN
      LOBS(PLOBS,1) = YY                                                  DSPLIN
      DO 10 I = 1,2                                                       DSPLIN
         DO 10 J = 1,2                                                    DSPLIN
 10   VAR(PVAR,I,J) = VXY(I,J)                                            DSPLIN
C* POINT TESTS                                                            DSPLIN
      IF(IOPT(8).EQ.1)WRITE(6,7060)LAG                                    DSPLIN
      IF(LAG.LT.1)GO TO 30                                                DSPLIN
      IL = INO - LAG                                                      DSPLIN
      IF(ISWT.EQ.0.AND.INO.GE.IOPT(1))GO TO 100                           DSPLIN
      IF(ISWT.EQ.1.AND.IL.GE.IOPT(5))GO TO 100                            DSPLIN
  30   IF(IOPT(8).EQ.1)WRITE(6,7000)INO                                   DSPLIN
       RETURN                                                             DSPLIN
C* ASSIGN KNOT TIME                                                       DSPLIN
 100  IL2 = IL + 1                                                        DSPLIN
      TKOLD = TK(2,1)                                                     DSPLIN
      IF(IOPT(2).EQ.0)TK(2,1) = TX(IL,1)                                  DSPLIN
      IF(IOPT(2).EQ.1)TK(2,1) = (TX(IL2,1) + TX(IL,1))/2.0D0              DSPLIN
      IF(ISWT2.EQ.0)TKOLD = TK(2,1)                                       DSPLIN
```

```
      IF(ISWT2.EQ.0)GO TO 105                                           DSPLIN
      DO 107 I = 1,8                                                    DSPLIN
         XOLD(I,1) = X(I,1)                                             DSPLIN
         DO 107 J = 1,8                                                 DSPLIN
  107 VXOLD(I,J) = VX(I,J)                                              DSPLIN
  105 IF(ISWT.EQ.1)GO TO 200                                            DSPLIN
C* FIRST POLYNOMINAL FIT                                                DSPLIN
      TK(1,1) = TX(1,1)                                                 DSPLIN
      DTK = TK(2,1) - TK(1,1)                                           DSPLIN
      DO 110 I = 1,INO                                                  DSPLIN
         CALL OBS(TX(I,1),TK(1,1),DTK,AOBS)                             DSPLIN
         DO 110 J = 1,7                                                 DSPLIN
  110 A(I,J) = AOBS(J,1)                                                DSPLIN
      DO 120 I = 1,PLOBS                                                DSPLIN
 120     W(I,1) = -LOBS(I,1)                                            DSPLIN
      IP = 0                                                            DSPLIN
      DO 130 I = 1,PVAR                                                 DSPLIN
      IP = IP + 1                                                       DSPLIN
         DO 140 J = 1,2                                                 DSPLIN
            DO 140 K = 1,2                                              DSPLIN
  140       VXY(J,K) = VAR(I,J,K)                                       DSPLIN
      CALL CHOLD(VXY,2,2,2,DET,IER)                                     DSPLIN
      IF(IER.NE.0)GO TO 9100                                            DSPLIN
      P(IP,2) = VXY(1,1)                                                DSPLIN
      P(IP,3) = VXY(1,2)                                                DSPLIN
      IP = IP + 1                                                       DSPLIN
      P(IP,1) = VXY(2,1)                                                DSPLIN
      P(IP,2) = VXY(2,2)                                                DSPLIN
 130  CONTINUE                                                          DSPLIN
C* CALL SOLV ROUTINE                                                    DSPLIN
      IF(IOPT(8).EQ.1)WRITE(6,7010)INO                                  DSPLIN
      IN = INO*2                                                        DSPLIN
      CALL SOLV(A,P,W,X,VX,V,IN,8,IDEG,SO,                              DSPLIN
     @          IDA,7,IDP,3,IDW,IDX,IDVX,IDV,ICO,IOPT(8),WKS)           DSPLIN
      IF(ICO.NE.0)GO TO 9200                                            DSPLIN
C* OUTPUT OF KNOT INFORMATION                                           DSPLIN
      IF(IOPT(9).EQ.1.OR.IOPT(10).EQ.1)WRITE(6,7010)INO                 DSPLIN
      IF(IOPT(9).EQ.1)CALL DPRINT(X,IDX,1,IDX,1,'KNOTS ',3)             DSPLIN
      IF(IOPT(10).EQ.1)CALL DPRINT(VX,                                  DSPLIN
     @                    IDVX,IDVX,IDVX,IDVX,'COVARIANCE',5)           DSPLIN
C* N.B OUTSIDE POINTS NOT INCLUDED IN INTERVAL TEST                     DSPLIN
      IF(IOPT(6).EQ.0)IN = IN - LAG*2                                   DSPLIN
C* ALTERNATIVE REJECTION ROUTINE PLACED HERE                            DSPLIN
CS    CALL REJECT(TX,IN,V,VAR,ICR,IRJET,ALPKNT,IDTX,IDV,IDVAR,IOPT(8))  DSPLIN
      CALL REJECT(IDEG,SO,ALPKNT,ICR)                                   DSPLIN
C* REJECT IF MAXIMUM NUMBER OF POINTS PER INTERVAL IS REACHED           DSPLIN
      IF(INO.EQ.INOBS)ICR = 22                                          DSPLIN
      IF(INO.EQ.INOBS)WRITE(6,7070)                                     DSPLIN
```

```
      IF(IOPT(6).EQ.0)IN = IN - LAG*2                                   DSPLIN
C ALTERNATIVE REJECTION ROUTINES TO BE PLACED HERE                       DSPLIN
CS    CALL REJECT(TX,IN,V,VAR,ICR,IRJET,ALPKNT,IDTX,IDV,IDVAR,IOPT(8))  DSPLIN
      CALL REJECT(IDEG,SO,ALPKNT,ICR)                                   DSPLIN
C* REJECT IF MAXIMUM NUMBER OF POINTS PER INTERVAL IS REACHED            DSPLIN
      IF(INO.EQ.INOBS)ICR = 22                                          DSPLIN
      IF(INO.EQ.INOBS)WRITE(6,7070)                                     DSPLIN
C* REJECTION                                                             DSPLIN
      IF(ICR.EQ.22)GO TO 300                                            DSPLIN
C* NC REJECTION                                                          DSPLIN
      ISWT2 = 1                                                         DSPLIN
      IF(IOPT(9).EQ.1)WRITE(6,7020)TK                                   DSPLIN
      RETURN                                                            DSPLIN
C*  RESET MATRICES AND POINTERS                                          DSPLIN
  300 TK(1,1) = TKOLD                                                   DSPLIN
      IF(ISWT2.EQ.0)WRITE(6,7050)                                       DSPLIN
      IF(ISWT2.EQ.1)GO TO 305                                           DSPLIN
      DO 307 I = 1,8                                                    DSPLIN
          XOLD(I,1) = X(I,1)                                            DSPLIN
         DO 307 J = 1,8                                                 DSPLIN
  307 VXOLD(I,J) = VX(I,J)                                              DSPLIN
  305 IEXTRA = ISWT2                                                    DSPLIN
      ISWT2 = 0                                                         DSPLIN
C* SHIFT TX,LOBS AND VAR MATRICES                                        DSPLIN
      IL2 = LAG + IEXTRA                                                DSPLIN
      IL = PTX - LAG - IEXTRA                                           DSPLIN
      DO 310 I = 1,IL2                                                  DSPLIN
      IL = IL + 1                                                       DSPLIN
  310 TX(I,1) = TX(IL,1)                                                DSPLIN
      PTX = LAG  + IEXTRA                                               DSPLIN
      IL = PLOBS - LAG*2   - IEXTRA*2                                   DSPLIN
      IL2 = LAG*2  +  IEXTRA*2                                          DSPLIN
      DO 320 I = 1,IL2                                                  DSPLIN
          IL = IL + 1                                                   DSPLIN
  320    LOBS(I,1) = LOBS(IL,1)                                         DSPLIN
      PLOBS = LAG*2  + IEXTRA*2                                         DSPLIN
      IL = PVAR - LAG - IEXTRA                                          DSPLIN
      IL2 = LAG   + IEXTRA                                              DSPLIN
      DO 330 I = 1,IL2                                                  DSPLIN
          IL = IL + 1                                                   DSPLIN
         DO 330 J = 1,2                                                 DSPLIN
            DO 330 K = 1,2                                              DSPLIN
  330   VAR(I,J,K) = VAR(IL,J,K)                                        DSPLIN
      PVAR = LAG   + IEXTRA                                             DSPLIN
      INO = LAG + IEXTRA                                                DSPLIN
C* SHIFT COMPLETED                                                       DSPLIN
C* GENERATE POSITION AND SLOPE INFORMATION                               DSPLIN
      DO 335 I = 1,2                                                    DSPLIN
```

```
           DO 335 J = 1,7                                                DSPLIN
     335 A(I,J) = 0.0D0                                                  DSPLIN
         A(1,1) = 1.0D0                                                  DSPLIN
         A(2,3) = 1.0D0                                                  DSPLIN
         DO 340 I = 1,4                                                  DSPLIN
         IL = I + 4                                                      DSPLIN
    340  W(I,1) = -XOLD(IL,1)                                            DSPLIN
C* NO CORRELATION BETWEEN POSITION AND SLOPE ASSUMED                      DSPLIN
         CALL CHOLD(VXOLD,IDVX,IDVX,IDVX,DET,IER)                        DSPLIN
         IF(IER.NE.0)GO TO 9300                                          DSPLIN
         P(1,2) = (IOPT(4)/100.0D0)*VXOLD(5,5)                           DSPLIN
         P(1,3) = (IOPT(4)/100.0D0)*VXOLD(5,6)                           DSPLIN
         P(2,1) = (IOPT(4)/100.0D0)*VXOLD(6,5)                           DSPLIN
         P(2,2) = (IOPT(4)/100.0D0)*VXOLD(6,6)                           DSPLIN
C* WEIGHTED SLOPE COVARIANCE                                              DSPLIN
         P(3,2) = (IOPT(4)/100.0D0)*VXOLD(7,7)                           DSPLIN
         P(3,3) = (IOPT(4)/100.0D0)*VXOLD(7,8)                           DSPLIN
         P(4,1) = (IOPT(4)/100.0D0)*VXOLD(8,7)                           DSPLIN
         P(4,2) = (IOPT(4)/100.0D0)*VXOLD(8,8)                           DSPLIN
         IF(IOPT(9).EQ.1)WRITE(6,7040)TK(1,1)                            DSPLIN
         IF(IOPT(8).EQ.1)CALL DPRINT(XOLD,8,1,8,1,                       DSPLIN
        @                              'OLD KNOT VALUES.KEEP',10)        DSPLIN
         TIME = TK(1,1)                                                  DSPLIN
         ICODE = 2                                                       DSPLIN
         RETURN                                                          DSPLIN
C* ERRORS                                                                 DSPLIN
   9000  WRITE(6,9010)IOPT                                               DSPLIN
         STOP                                                            DSPLIN
   9100  WRITE(6,9110)IER                                                DSPLIN
         ICODE = 1                                                       DSPLIN
         RETURN                                                          DSPLIN
   9200  WRITE(6,9210)                                                   DSPLIN
         RETURN                                                          DSPLIN
   9300  WRITE(6,9310)                                                   DSPLIN
         ICODE = 1                                                       DSPLIN
         RETURN                                                          DSPLIN
   7005  FORMAT(//,3X,'INPUT NEW DATA POINT',/)                          DSPLIN
   7000  FORMAT(/,3X,'INSUFFICIENT POINTS ',I4)                          DSPLIN
   7010  FORMAT(/,3X,'NUMBER OF POINTS',I4)                              DSPLIN
   7020  FORMAT(/,3X,'PASSED : KNOT TIMES : ',1PD12.5,3X,1PD12.5)        DSPLIN
   7030  FORMAT(///,3X,'SOLUTION OF CURRENT POINT')                      DSPLIN
   7040  FORMAT(/,3X,'NEXT KNOT TIME ',1PD15.8)                          DSPLIN
   7050  FORMAT(//,3X,'**WARNING** NEW KNOT TIME ACCEPTED ALTHOUGH',     DSPLIN
        @        ' REJECTED BY TESTS',//)                                DSPLIN
   7060  FORMAT(/,3X,'COMPUTED LAG ',I4,' POINTS')                       DSPLIN
   7070  FORMAT(/,3X,'**WARNING ** MAXIMUM PTS PER INTERVAL REACHED.',   DSPLIN
        @        ' NEW KNOT TIME ACCEPTED',//)                           DSPLIN
   9010  FORMAT(//,3X,'FIT : OPTION ERROR : INPUT ',10I4,//)             DSPLIN
```

```
      9110 FORMAT(/,3X,'**FATAL** CHOLD : ',                                  DSPLIN
          @              'INPUT COVARIANCE ERROR; CODE = ',I4)                  DSPLIN
      9210 FORMAT(/,3X,'**NON FATAL** SOLV FAILURE. MOVE TO NEXT POINT')        DSPLIN
      9310 FORMAT(/,3X,'**FATAL** CHOLD : INVERSE OF COMPUTED COVARIANCE',      DSPLIN
          @              ' FAILED',//)                                          DSPLIN
           END                                                                  DSPLIN
           SUBROUTINE FORM(T,B)                                                 DSPLIN
C *************************************************************               DSPLIN
C * VERSION      : JANUARY 1982                                     *         DSPLIN
C * DESCRIPTION  : FORMULATES THE MATRIX TO RECOVER THE SMOOTHED    *         DSPLIN
C *                POSITION OF A POINT AT TIME T FROM THE           *         DSPLIN
C *                COEFFICIENTS.                                    *         DSPLIN
C *************************************************************               DSPLIN
C INPUT PARAMETER                                                              DSPLIN
C        T  - TIME DIFF BETWEEN THE DATA POINT AND THE PRECEEDING KNOT        DSPLIN
C OUTPUT PARAMETER                                                             DSPLIN
C        B  - MATRIX IN WHICH THE COEFFICIENTS ARE STORED                     DSPLIN
C                                                                              DSPLIN
           IMPLICIT REAL*8 (A-H,O-Z)                                            DSPLIN
           REAL*8 B(2,8)                                                        DSPLIN
           B(1,1) = 1.0D00                                                      DSPLIN
           B(1,3) = T                                                           DSPLIN
           B(1,5) = T*T                                                         DSPLIN
           B(1,7) = B(1,5)*T                                                    DSPLIN
           DO 10 I = 1,7,2                                                      DSPLIN
           I2 = I + 1                                                           DSPLIN
   10      B(2,I2) = B(1,I)                                                     DSPLIN
           RETURN                                                               DSPLIN
           END                                                                  DSPLIN
           SUBROUTINE INTERV(T,TKNOT,N,INT,K,ICODE)                             DSPLIN
           IMPLICIT REAL*8(A-H,O-Z)                                             DSPLIN
C   *************************************************************             DSPLIN
C   * VERSION      : JANUARY 1982                                     *       DSPLIN
C   * DESCRIPTION  : TO LOCATE THE INTERVAL IN WHICH THE TIME 'T'     *       DSPLIN
C   *                LIES IN AN ARRAY SEQUENCE OF KNOT TIMES          *       DSPLIN
C   *************************************************************             DSPLIN
C   INPUT PARAMETERS                                                           DSPLIN
C        T     - TIME IN WHICH THE INTERVAL NUMBER IS REQUIRED                 DSPLIN
C        TKNOT - ARRAY HOLDING THE KNOT TIME SEQUENCE                          DSPLIN
C                DIMENSIONED TKNOT(N,1)                                        DSPLIN
C        N     - NUMBER OF KNOT'S TIMES CONTAINED IN THE ARRAY KNOT            DSPLIN
C        K     - DECLARED SIZE OF ARRAY TKNOT                                  DSPLIN
C                                                                              DSPLIN
C   OUTPUT PARAMETERS                                                          DSPLIN
C        INT   - INTERVAL IN WHICH T LIES                                      DSPLIN
C        ICODE -  1  SUCCESSFUL EXECUTION                                      DSPLIN
C                 0  INPUT TIME T OUT OF DEFINED INTERVAL                      DSPLIN
C                                                                              DSPLIN
```

```
C  CALLS - DABS
       REAL*8 TKNOT(K,1)
       DATA XLIMIT/1.0D-7/
C*
       ICODE = 0
C* THE NEXT THREE LINES IS USED TO SPEED UP SEARCH TIME
       MIDDLE = N/2
       IF(T.GT.TKNOT(MIDDLE,1))II = MIDDLE
       IF(T.LE.TKNOT(MIDDLE,1))II = 1
C*
       DO 10 I = II,N
       IF(T.LT.XLIMIT.AND.TKNOT(I,1).LT.XLIMIT)DT = 0.0D00
       IF(T.GT.XLIMIT.OR.TKNOT(I,1).GT.XLIMIT)DT = T - TKNOT(I,1)
       IF(I.NE.1.AND.I.NE.N)GO TO 20
       IF(I.EQ.1.AND.DABS(DT).LT.XLIMIT)GO TO 30
       IF(I.EQ.N.AND.DABS(DT).LT.XLIMIT)GO TO 40
   20  IF(DT.LT.0.0D00)GO TO 50
   10  CONTINUE
       RETURN
   50  IF(I.GT.1)GO TO 40
       RETURN
   40  INT = I - 1
       GO TO 100
   30  INT = 1
  100  ICODE = 1
       RETURN
       END
       SUBROUTINE LS(A1,ADDA1,A2,ADDA2,P,W1,W2,X,VX,R,N,M,IA1,
      &              IA2,IP,IW1,IW2,IX,IER,NAM,VM)
       IMPLICIT REAL*8(A-H,C-Z)
C ****************************************************************
C * VERSION      : 28TH    APRIL 1982                            *
C * DESCRIPTION : FULLY OPTIMISED FUNCTIONALLY                    *
C *               CONSTRAINED LS MODEL ESTIMATION  USING          *
C *               THE SPIN ROUTINE.                               *
C ****************************************************************
C INPUT PARAMETERS
C     A1,A2    - 1ST AND 2ND DESIGN MATRICES (VARIABLE PROFILE BANDED)
C ADDA1,ADDA2- ADDRESS SEQUENCES FOR A1 AND A2 MATRICES
C     W1,W2    - MISCLOSURE VECTORS
C     P        - WEIGHT MATRICES
C     R,N,M    - DIMENSION OF PROBLEM; R - NUMBER OF CONSTRAINT EQNS
C                                      N - NUMBER OF OBSERVATION EQNS
C                                      M - NUMBER OF UNKNOWNS
C     IA1,IA2,IP,IW1,IW1,IX
C              - DECLARED DIM OF A1,A2,P,W1,W2 X,VX MATRICES
C
C OUTPUT PARAMETERS
```

```
C     X,VX     - SOLUTION VECTOR AND VARIANCES                                  DSPLIN
C     IX       - DECLARED DIM OF X AND VX MATRICES                              DSPLIN
C     IER      - ERROR CODES - 0 : SUCCESSFUL COMPLETION OF ROUTINE             DSPLIN
C                              1 : ERROR DETECTED                               DSPLIN
C                                                                               DSPLIN
C WORK ARRAYS - NAM,VM   ( AS DIM BELOW )                                       DSPLIN
C                                                                               DSPLIN
C ADDITIONAL NOTES                                                              DSPLIN
C     MATRICES USED BY THE SUBROUTINE MUST BE MODIFIED AS FOLLOWS               DSPLIN
C     IF A CHANGE IN THE DECLARED DIMENSIONS OF MATRICES IS IN EFFECT.          DSPLIN
C     THE FOLLOWING IS THE MODIFICATION PROCEDURE.                              DSPLIN
C                                                                               DSPLIN
C*        ATPW(IX,1)        A2N(IA2*2,IX)                                       DSPLIN
C*        VM  (IA2*2,IA2*2)     Q  (IA2*2,1)     NAM(IX,IA2*2)                  DSPLIN
C*                                                                              DSPLIN
C*                               *********                                      DSPLIN
      REAL*8 A1(IA1,7),A2(IA2,11),P(IP,3),W1(IW1,1),W2(IW2,1),                  DSPLIN
     @ X(IX,1),VX(IX,IX)                                                        DSPLIN
C* USED BY PROGRAM  **  WORK ARRAYS - RELIES ON OTHER MATRIX SIZES              DSPLIN
      REAL*8 NAM(IX,IW2),VM(IW2,IW2)                                            DSPLIN
C* LOCAL ARRAYS  * CURRENT SET FOR IOBSNO = 500 : IKNOT = 32                    DSPLIN
C*    CURRENT SETTINGS                                                          DSPLIN
      REAL*8  ATPW(128,1),A2N(60,128),                                          DSPLIN
     @        Q(60,1)                                                           DSPLIN
      REAL*8 QSUM,ONE                                                           DSPLIN
      INTEGER*4 R,M,N,ADDA1(IA1),ADDA2(IA2)                                     DSPLIN
      EQUIVALENCE (Q(1),ATPW(1),A2N(1))                                         DSPLIN
C* FUNCTIONS                                                                    DSPLIN
      IROW(I) = (I+1)/2                                                         DSPLIN
      ICOL(I) = 2 - ((I+1)/2 - (I/2))                                           DSPLIN
C* DIMENSION SETTING VARIABLE                                                   DSPLIN
      IDVM = IA2*2                                                              DSPLIN
      IDATPW = IX                                                               DSPLIN
C* WRITE UNIT                                                                   DSPLIN
      IWR = 6                                                                   DSPLIN
      ONE = 1.0D0                                                               DSPLIN
C* DEFAULT ERROR SET                                                            DSPLIN
      IER = 0                                                                   DSPLIN
C* TEST ON CONDITION NUMBER                                                     DSPLIN
      XLEVEL = 1.0D10                                                           DSPLIN
C*                                                                              DSPLIN
C*  SINCE THE NORMAL MATRIX IS SYMMETRICAL, THE ROUTINE ONLY                    DSPLIN
C*  OPERATES ON THE LOWER HALF OF THE MATRIX AND EQUATES ITS COUNTERPARTDSPLIN
C*                                                                              DSPLIN
      DO 10 I = 1,M                                                             DSPLIN
          IEND = I - 10                                                         DSPLIN
         DO 10 L = 1,M                                                          DSPLIN
             QSUM = 0.0D00                                                      DSPLIN
```

```
                IF(L.LT.IEND.OR.L.GT.I)GO TO 50                      DSPLIN
               DO 20 J = 1,N                                          DSPLIN
                  JJ = IROW(J)                                        DSPLIN
                  LL = L - (ADDA1(JJ) + ICOL(J) - 1)  + 1             DSPLIN
                  IF(LL.LT.1.OR.LL.GT.7)GO TO 20                      DSPLIN
                  IB = J - 1                                          DSPLIN
                  IE = J + 1                                          DSPLIN
                  IF(IB.LT.1)IB = 1                                   DSPLIN
                  IF(IE.GT.N)IE = N                                   DSPLIN
                 DO 30 K = IB,IE                                      DSPLIN
                    KK = IROW(K)                                      DSPLIN
                    JAC = J - K + 2                                   DSPLIN
                    II = I - (ADDA1(KK) + ICOL(K) - 1)  + 1           DSPLIN
                    IF(II.LT.1.OR.II.GT.7)GO TO 30                    DSPLIN
                 QSUM = QSUM + ONE*A1(KK,II)*P(K,JAC)*A1(JJ,LL)       DSPLIN
   30         CONTINUE                                                DSPLIN
   20       CONTINUE                                                  DSPLIN
  50      VX(I,L) = QSUM                                              DSPLIN
          IF(L.NE.I)VX(L,I) = VX(I,L)                                 DSPLIN
 10       CONTINUE                                                    DSPLIN
C* CONSTRUCT THE ATPW MATRIX                                           DSPLIN
          DO 60 I = 1,M                                               DSPLIN
              QSUM = 0.D00                                            DSPLIN
             DO 65 J = 1,N                                            DSPLIN
                 IB = J - 1                                           DSPLIN
                 IE = J + 1                                           DSPLIN
                 IF(IB.LT.1)IB = 1                                    DSPLIN
                 IF(IE.GT.N)IE = N                                    DSPLIN
                DO 65 K = IB,IE                                       DSPLIN
                    KK = IROW(K)                                      DSPLIN
                    II = I - (ADDA1(KK)+ICOL(K) - 1) + 1              DSPLIN
                    IF(II.LT.1.OR.II.GT.7)GO TO 65                    DSPLIN
                    JP = J - K + 2                                    DSPLIN
              QSUM = QSUM + ONE*A1(KK,II)*P(K,JP)*W1(J,1)             DSPLIN
  65      CONTINUE                                                    DSPLIN
           ATPW(I,1) = QSUM                                           DSPLIN
  60      CONTINUE                                                    DSPLIN
C* CHOICE OF INVERSE ROUTINES                                          DSPLIN
          CALL COND(VX,M,IX,XCOND,1)                                  DSPLIN
          CALL SPIN(VX,M,IX,DET,IDEXP)                                DSPLIN
CS        CALL CHOLD(VX,M,IX,7,DET,IER)                               DSPLIN
          CALL CCND(VX,M,IX,XCOND,2)                                  DSPLIN
          IF(IER.EQ.0.AND.XCOND.GT.XLEVEL)IER = 1                     DSPLIN
          WRITE(IWR,88)IER                                            DSPLIN
          IF(IER.NE.0)GO TO 999                                       DSPLIN
          CALL DMULT(VX,M,M,ATPW,M,1,X,IX,IX,IDATPW,1,IX,1)           DSPLIN
          CALL SCMULT(-1.0D00,X,X,M,1,IX,1)                           DSPLIN
          DO 70 I = 1,R                                               DSPLIN
```

```
         DO 70 J = 1,M                                          DSPLIN
             QSUM = 0.0D00                                      DSPLIN
            DO 75 K = 1,M                                       DSPLIN
                II = IROW(I)                                    DSPLIN
                KK = K - (ADDA2(II) + ICOL(I) - 1)  + 1         DSPLIN
                IF(KK.LT.1.OR.KK.GT.11)GO TO 75                 DSPLIN
         QSUM = QSUM + A2(II,KK)*VX(K,J)                        DSPLIN
   75   CONTINUE                                                DSPLIN
        A2N(I,J) = QSUM                                         DSPLIN
   70   CONTINUE                                                DSPLIN
        DO 100 I = 1,R                                          DSPLIN
           DO 100 J = 1,R                                       DSPLIN
               QSUM = 0.0D00                                    DSPLIN
            DO 105 K = 1,M                                      DSPLIN
                JJ = IROW(J)                                    DSPLIN
                KK = K - (ADDA2(JJ) + ICOL(J) - 1)  + 1         DSPLIN
                IF(KK.LT.1.OR.KK.GT.11)GO TO 105                DSPLIN
        QSUM = QSUM + A2N(I,K)*A2(JJ,KK)                        DSPLIN
    105 CONTINUE                                                DSPLIN
        VM(I,J) = QSUM                                          DSPLIN
    100 CONTINUE                                                DSPLIN
C* CHOICE OF INVERSE ROUTINES                                    DSPLIN
        CALL COND(VM,R,IDVM,XCOND,1)                            DSPLIN
        CALL SPIN(VM,R,IDVM,DET,IDEXP)                          DSPLIN
CS      CALL CHOLD(VM,R,IDVM,R,DET,IER)                         DSPLIN
        CALL COND(VM,R,IDVM,XCOND,2)                            DSPLIN
        IF(IER.EQ.0.AND.XCOND.GT.XLEVEL)IER = 1                 DSPLIN
        WRITE(IWR,88)IER                                        DSPLIN
C* CONSTRUCT NAM MATRIX                                         DSPLIN
        DO 250 I = 1,M                                          DSPLIN
           DO 250 L = 1,R                                       DSPLIN
               QSUM = 0.0D0                                     DSPLIN
               NAM(I,L) = 0.D00                                 DSPLIN
            DO 260 J = 1,R                                      DSPLIN
               DO 260 K = 1,M                                   DSPLIN
                   JJ = IROW(J)                                 DSPLIN
                   KK = K - (ADDA2(JJ) + ICOL(J) - 1)  + 1      DSPLIN
                   IF(KK.LT.1.OR.KK.GT.11)GO TO 260             DSPLIN
        QSUM = QSUM + ONE*VX(I,K)*A2(JJ,KK)*VM(J,L)             DSPLIN
   260        CONTINUE                                          DSPLIN
   250  NAM(I,L) = QSUM                                         DSPLIN
C* CONSTRUCT THE COVARIANCE                                     DSPLIN
        DO 500 I = 1,M                                          DSPLIN
           DO 500 J = 1,M                                       DSPLIN
               QSUM = 0.0D0                                     DSPLIN
            DO 510 K = 1,R                                      DSPLIN
    510         QSUM = QSUM + NAM(I,K)*A2N(K,J)                 DSPLIN
    500 VX(I,J) = VX(I,J) - QSUM                                DSPLIN
```

```
C* CONSTRUCT THE Q VECTOR                                               DSPLIN
      DO 300 I = 1,R                                                    DSPLIN
         QSUM = 0.0D0                                                   DSPLIN
         DO 310 J = 1,M                                                 DSPLIN
            II = IROW(I)                                                DSPLIN
            JI = J - (ADDA2(II) + ICOL(I) - 1)  + 1                     DSPLIN
            IF(JI.LT.1.OR.JI.GT.11)GO TO 310                            DSPLIN
            QSUM = QSUM + A2(II,JI)*X(J,1)                              DSPLIN
  310       CONTINUE                                                    DSPLIN
  300 Q(I,1) = W2(I,1) + QSUM                                           DSPLIN
C* CORRECT THE X VECTOR                                                 DSPLIN
      DO 400 I = 1,M                                                    DSPLIN
         QSUM = 0.0D0                                                   DSPLIN
         DO 410 J = 1,R                                                 DSPLIN
  410       QSUM = QSUM + NAM(I,J)*Q(J,1)                               DSPLIN
  400    X(I,1) = X(I,1) - QSUM                                         DSPLIN
C* COMPUTATIONAL CHECK ON ACCURACY                                      DSPLIN
      XLR = 0.0D00                                                      DSPLIN
      DO 600 I = 1,M                                                    DSPLIN
         DO 600 J = 1,R                                                 DSPLIN
            QSUM = 0.0D0                                                DSPLIN
            DO 610 K = 1,M                                              DSPLIN
               JJ = IROW(J)                                             DSPLIN
               KK = K - (ADDA2(JJ) + ICOL(J) - 1)  + 1                  DSPLIN
               IF(KK.LT.1.OR.KK.GT.11)GO TO 610                         DSPLIN
               QSUM = QSUM + VX(I,K)*A2(JJ,KK)                          DSPLIN
  610          CONTINUE                                                 DSPLIN
         XNUM = QSUM                                                    DSPLIN
         IF(DABS(XNUM).GT.DABS(XLR))XLR = XNUM                          DSPLIN
  600 CONTINUE                                                          DSPLIN
C* WRITE INTERNAL COMPUTATIONAL ACCURACY                                DSPLIN
      WRITE(IWR,99)XLR                                                  DSPLIN
   99 FORMAT(3X,'INT COMP ACC ; ',1PD12.5)                              DSPLIN
   88 FORMAT(3X,'ERROR CODE    ; ',I4)                                  DSPLIN
      RETURN                                                            DSPLIN
  599 WRITE(IWR,900)                                                    DSPLIN
  900 FORMAT(//,3X,'DATA POINT SEQUENCE VS KNOT SEQUENCE NOT COMPATIBLE'DSPLIN
     @ /,3X,'AS IT LEADS TO A SINGULAR INVERSE  MATRIX IN THE PROGRAM.',DSPLIN
     @ /,3X,'PROCESSING FOR THE CURRENT SET IS HALTED AND SHIFTED TO',  DSPLIN
     @ /,3X,'THE NEXT SET',///)                                         DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
```

```
      SUBROUTINE OBS(TXPT,TKNOT,DINTK,A)                                DSPLIN
      IMPLICIT REAL*8(A-H,M-Z)                                          DSPLIN
C ******************************************************************** DSPLIN
C * VERSION      : JANUARY 1982                                       * DSPLIN
C * DESCRIPTION : TO GENERATE THE COEFFICIENTS OF AN OBSERVATION     * DSPLIN
C *               EQUATION AND SAVE IT IN 'A' MATRIX.                * DSPLIN
C ******************************************************************** DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C     TXPT  - TIME OF DATA POINT                                        DSPLIN
C     TKNOT - TIME OF BEFORE KNOT OF THE INTERVAL IN WHICH THE TIME     DSPLIN
C             DATA POINT FALLS                                          DSPLIN
C     DINTK - TIME INTERVAL BETWEEN THE BEFORE KNOT AND AFTER KNOT      DSPLIN
C             OF THE INTERVAL                                           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C     A     - THE MATRIX IN WHICH THE COEFFICIENTS ARE STORED           DSPLIN
C                                                                       DSPLIN
C CALLS - DCLEAR                                                        DSPLIN
C                         **********                                    DSPLIN
      REAL*8 A(7,1),X,Y                                                 DSPLIN
      CALL DCLEAR(A,7,1)                                                DSPLIN
      DR = TXPT-TKNOT                                                   DSPLIN
      DTR = DR/DINTK                                                    DSPLIN
      X = (2.0D00*DTR - 3.0D00)*DTR*DTR                                 DSPLIN
      Y = 1.0D0 - DTR                                                   DSPLIN
```

```
      A(1,1) = (1.0D00 + X)                                             DSPLIN
      A(3,1) = DR*Y*Y                                                   DSPLIN
      A(5,1) = -X                                                       DSPLIN
      A(7,1) = -DR*DTR*Y                                                DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE OPTION(CHANGE,IREAD,IFLT,IAPRI,ISCT,IELIPS,ECHO,       DSPLIN
     @          PRCOV,PRPMAT,PRTX,PRTK,PRDTK,PRLOBS,ICHECK,             DSPLIN
     @          PRAMAT,PRXMAT,PRVARX,PRKNT,PRRESD,PRCOFF,PRCOFV,        DSPLIN
     @          ITYPE,SPLIN,PRPTS,PRKNOT,DRAW,PRGEN,IANLYS,IPLOT,       DSPLIN
     @          ICANYS,SCTIME,DINT,DINC,ALPH,ALPHA,CALPHA,              DSPLIN
     @          FESC,ESC,XSTART,XINT,GSTART,GINT)                       DSPLIN
C ***********************************************************************DSPLIN
C * VERSION     : 22 SEPTEMBER 1982                                    *DSPLIN
C * DESCRIPTION : HOLDS ALL OPTION DEFAULTS FOR THE MAIN PROGRAM AND   *DSPLIN
C * ALLOWS CHANGES TO THESE OPTIONS. ALL PARAMETERS DESIGNATED AS      *DSPLIN
C * ARGUMENTS ARE DESCRIBED IN THE MAIN. THE FOLLOWING ARE, INSTEAD,   *DSPLIN
C * THE INPUT SEQUENCE OF CODES, THEIR DESCRIPTION AND POSSIBLE        *DSPLIN
C * ALTERNATIVES.                                                      *DSPLIN
C ***********************************************************************DSPLIN
C PRELUDE                                                               DSPLIN
C  IN GENERAL,'NO' IS DENOTED BY '0' AND 'Y' IS DENOTE BY '1'           DSPLIN
C  OPTIONAL INPUT ARE ADDITIONAL INPUT REQUIRED WHEN CERTAIN ROUTINES   DSPLIN
C  ARE INVOKED. THE FOLLOWING NOTATION IS USED.                         DSPLIN
C     A B C      - DENOTES COMPULSORY REQUESTED INPUTS                  DSPLIN
C     I II III   - DENOTES RANGE OF POSSIBLE INPUTS CODES FOR EACH OPTION DSPLIN
C     (1,2,,10) - ARE THE INPUT CODES FOR EACH OF THE CNTRL VARIABLES   DSPLIN
C                                                                       DSPLIN
C          ******** INPUT SEQUENCE IS AS FOLLOWS *********               DSPLIN
C VARIABLE : DESCRIPTION AND INPUT CODES REQUESTED                      DSPLIN
C                                                                       DSPLIN
C OPTON :   CHANGE ALL OPTIONS ? (1,0)                                  DSPLIN
C          IF NO SKIP THE REST OF THE INPUTS                            DSPLIN
C                                                                       DSPLIN
C*READ IN 26 MAIN  OPTIONS FOLLOWED BY SELECTED OPTIONS                 DSPLIN
C IREAD :  SELECTION OF INPUT DATA FORMAT(1,2,3,5,7,9)                  DSPLIN
C IFLT  :  CNTRLS THE IMPLEMENTATION OF LINEAR FILTER ROUTINE(1,0)      DSPLIN
C IAPRI :  CHOICE OF APRIORI SOLUTION VECTORS(0,1,2)                    DSPLIN
C ISCT  :  CNTRLS THE IMPLEMENTATION OF THE SUBR TIMSC(1,0)             DSPLIN
C ICHI  :  INPUT NEW % CONFIDENCE INTERVAL ON CHI-SQ TEST(1,0)          DSPLIN
C IELIPS:  INPUT NEW % CONFIDENCE ELLIPSE ON COMPUTED ELLIPSES(1,0)     DSPLIN
C ECHO  :  INPUT VERIFICATION OF DATA INPUT INTO PROGRAM(1,0)           DSPLIN
C                                                                       DSPLIN
C PRCOV :  PRINT INPUT COVARIANCE MATRICES OF DATA POINTS(1,0)          DSPLIN
C PRPMAT:  PRINT BANDED WEIGHT MATRIX(1,0)                              DSPLIN
C PRTX  :  PRINT DATA POINT TIMES - SCALED(1,0)                         DSPLIN
C PRTK  :  PRINT KNOT TIMES- SCALED(1,0)                                DSPLIN
C PRDTK :  PRINT KNOT INTERVAL TIMES - SCALED(1,0)                      DSPLIN
```

```
C PRLOBS: PRINT FORMULATED OBSERVATION VECTOR(1,0)                      DSPLIN
C                                                                       DSPLIN
C ICHECK: IMPLEMENTATION OF SUBR CHECK(1,0)                             DSPLIN
C                                                                       DSPLIN
C PRAMAT: PRINT DESIGN MATRICES (1,0) - SEE MAIN                        DSPLIN
C PRXMAT: PRINT LS ESTIMATED SOLUTION VECTOR(1,0)                       DSPLIN
C PRVARX: PRINT COVARIANCE MATRIX OF LS ESTIMATES(1,0)                  DSPLIN
C PRKNT : PRINT 'PRKNT' KNOT VALUES AND VARIANCE(1,,KNO)- SEE MAIN      DSPLIN
C PRRESD: PRINT RESIDUAL VECTOR(1,0)                                    DSPLIN
C PRCOFF: PRINT FORMULATED COEFFICIENTS(1,0)                            DSPLIN
C PRCOFV: PRINT COAVRIANCE MATRIX OF COEFFICIENTS(1,0)                  DSPLIN
C                                                                       DSPLIN
C ITYPE : PLOTTER  I - (1)  LINE PRINTER CHOSEN                         DSPLIN
C                 II- (0)  ZETA/GOULD 5000 PLOTTER                      DSPLIN
C SPLIN : NO OF TIME SUBR DESIGN IS TO BE CALLED(1,,99) - SEE MAIN      DSPLIN
C IANLYS: NO OF TIMES SUBR ANALYS IS TO BE CALLED(1,,99)- SEE MAIN      DSPLIN
C IPLOT : PLOTTING REQUIREMENTS OF SUBR ANALYS(1,0) - SEE SUBR ANALYS   DSPLIN
C                                                                       DSPLIN
C*OPTIONAL INPUT(ONLY WHEN INVOKED FROM THE ABOVE)                       DSPLIN
C  IF SPLIN  OPTION >= 1 :INPUT PRPTS,PRKNOT,DRAW AND PRGEN             DSPLIN
C                         SEE SUBR DESIGN FOR DETAILS (A-D)             DSPLIN
C      IANLYS OPTION >= 1 :INPUT ITS 10 OPTIONS                         DSPLIN
C                         SEE SUBR ANALYS FOR DETAILS (A-J)             DSPLIN
C                         SUBOPTION : IF ICANYS(8) = 1  INVOKED         DSPLIN
C                            GSTART - TIME TO BEGIN PRINT OF VALUES     DSPLIN
C                            GINT   - TIME BETWEEN COMPUTATIONS         DSPLIN
C      ISCT   OPTION = 1 :INPUT THE TIME SCALE FACTOR                   DSPLIN
C      ICHI   OPTION = 1 :INPUT NEW PERCENTAGE CONFIDENCE INTERVAL      DSPLIN
C      IFLT   OPTION = 1 :INPUT INTERVAL WIDTH,INCREMENT AND TEST       DSPLIN
C                        X LEVEL TO BE APPLIED - SEE FILTER (A-C)       DSPLIN
C      IELIPS OPTION = 1 :INPUT NEW PERCENTAGE CONFIDENCE ELLIPSES      DSPLIN
C                         REQUIRED.                                     DSPLIN
C      DRAW OR       = 1 OR 2 :INPUT TIME OF BEGINNING OF CURVE AND POINTDSPLIN
C      ICANYS(9)                  XSTART - TIME OF BEGINNING OF CURVE   DSPLIN
C      OPTION                     XINT   - TIME BETWEEN COMPUTED POSITIONS DSPLIN
C*END INPUT SEQUENCE                                                    DSPLIN
C                                                                       DSPLIN
C ADDITIONAL INFORMATION                                                DSPLIN
C        SPLIN  - PLOTTING OF SPLINE  **  TOTAL CONTROL                 DSPLIN
C        DRAW   - DRAW THE SPLINE                                       DSPLIN
C        PRGEN  - PRINT GENERATED DISCRETE POSITIONS OF DRAWN SPLINE    DSPLIN
C        IPLOT  - FOR PLOTTING IN THE ANALYS SUBROUTINE                 DSPLIN
C        ISCT   - TIME SCALING FACTOR  ON/OFF SWITCH                    DSPLIN
C                                                                       DSPLIN
C CALLS         - ELSFAC                                                DSPLIN
C                                                                       DSPLIN
C*PROGRAM BEGINS                                                        DSPLIN
      REAL*8 ELSFAC,CALPHA,GSTART,GINT                                  DSPLIN
```

```
      REAL*8 SCTIME,DINT,DINC,XSTART,XINT,ALPH,ALPHA,FESC,ESC           DSPLIN
      INTEGER  CHANGE,IREAD,IFLT,IAPRI,ISCT,ICHI,IELIPS,ECHO,           DSPLIN
     @          PRCOV,PRPMAT,PRTX,PRTK,PRDTK,PRLOBS,ICHECK,             DSPLIN
     @          PRAMAT,PRXMAT,PRVARX,PRKNT,PRRESD,PRCOFF,PRCOFV,        DSPLIN
     @          ITYPE,SPLIN,PRPTS,PRKNOT,DRAW,PRGEN,IANLYS,IPLOT,       DSPLIN
     @          ICANYS(10)                                              DSPLIN
C* READ UNIT                                                            DSPLIN
      IRD = 5                                                           DSPLIN
C* WRITE UNIT                                                           DSPLIN
      IWR = 6                                                           DSPLIN
C*                                                                      DSPLIN
C  OPTIONS SELECTED  (DEFAULT)                                          DSPLIN
C*                                                                      DSPLIN
C* DATA SET CHANGE CONTROL VARIABLE                                     DSPLIN
      CHANGE = 0                                                        DSPLIN
C* READ SPECIFICATION  ?                                                DSPLIN
      IREAD  = 1                                                        DSPLIN
C* TO CALL FILTER ROUTINE CPTION                                        DSPLIN
      IFLT   = 0                                                        DSPLIN
C* CHOICE OF APRIORI VALUES                                             DSPLIN
      IAPRI  = 0                                                        DSPLIN
C* TIME SCALING  OPTION                                                 DSPLIN
      ISCT   = 0                                                        DSPLIN
      SCTIME = 1.0D0                                                    DSPLIN
C* CHI-SQUARE TEST ON VARIANCE FACTOR CONDIENCE INTERVAL                DSPLIN
      ICHI   = 0                                                        DSPLIN
      CALPHA = 99.0                                                     DSPLIN
C* CONFIDENCE ELLIPSE SCALE FACTOR OPTION: GOVERNS ALL ELLIPSES         DSPLIN
      IELIPS = 0                                                        DSPLIN
      ESC    = 1                                                        DSPLIN
C* INPUT DATA PRINT AND DATA PRIOR TO COMPUTATIONS                      DSPLIN
      ECHO   = 1                                                        DSPLIN
      PRCOV  = 0                                                        DSPLIN
      PRPMAT = 0                                                        DSPLIN
      PRTX   = 0                                                        DSPLIN
      PRTK   = 1                                                        DSPLIN
      PRDTK  = 0                                                        DSPLIN
      PRLOBS = 0                                                        DSPLIN
C* OPTION ON THE CHECK ROUTINE                                          DSPLIN
      ICHECK = 0                                                        DSPLIN
C* COMPUTED DATA PRINT                                                  DSPLIN
      PRAMAT = 0                                                        DSPLIN
      PRXMAT = 1                                                        DSPLIN
      PRVARX = 0                                                        DSPLIN
      PRKNT = 0                                                         DSPLIN
      PRRESD = 1                                                        DSPLIN
      PRCOFF = 1                                                        DSPLIN
      PRCOFV = 0                                                        DSPLIN
```

```
C* PLOTTING CONTROL
      ITYPE  = 0                                                        DSPLIN
      SPLIN  = 1                                                        DSPLIN
      PRPTS  = 1                                                        DSPLIN
      PRKNOT = 1                                                        DSPLIN
      DRAW   = 0                                                        DSPLIN
      PRGEN  = 0                                                        DSPLIN
C* PLOT OTHER THAN SPLINE DRAWING                                       DSPLIN
      IANLYS = 0                                                        DSPLIN
      IPLOT  = 0                                                        DSPLIN
C* IS THE DEFAULT SET OPTIONS TO BE CHANGED ?                           DSPLIN
      READ(IRD,*)IOPTON                                                 DSPLIN
      IF(IOPTON.EQ.0)RETURN                                             DSPLIN
C* READ IN THE NEW OPTIONS                                              DSPLIN
       READ(IRD,*)CHANGE,IREAD,IFLT,IAPRI,ISCT,ICHI,IELIPS,ECHO,        DSPLIN
     @          PRCOV,PRPMAT,PRTX,PRTK,PRDTK,PRLOBS,ICHECK,             DSPLIN
     @          PRAMAT,PRXMAT,PRVARX,PRKNT,PRRESD,PRCOFF,PRCOFV,        DSPLIN
     @          ITYPE,SPLIN,IANLYS,IPLOT                                DSPLIN
       IF(ECHO.EQ.1)WRITE(IWR,7560)CHANGE,IREAD,IFLT,IAPRI,ISCT,ICHI,   DSPLIN
     @          IELIPS,ECHO,PRCOV,PRPMAT,PRTX,PRTK,PRDTK,PRLOBS,ICHECK, DSPLIN
     @          PRAMAT,PRXMAT,PRVARX,PRKNT,PRRESD,PRCOFF,PRCOFV,        DSPLIN
     @          ITYPE,SPLIN,IANLYS,IPLOT                                DSPLIN
C* OPTIONS RELATED TO SPLINE PLOT                                       DSPLIN
      IF(SPLIN.NE.0)READ(IRD,*)PRPTS,PRKNOT,DRAW,PRGEN                  DSPLIN
      IF(ECHO.EQ.1.AND.SPLIN.NE.0)WRITE(IWR,7560)PRPTS,PRKNOT,DRAW,PRGENDSPLIN
C* OPTIONS RELATED TO ANALYS SUBROUTINE                                 DSPLIN
      IF(IANLYS.NE.0)READ(IRD,*)(ICANYS(I),I = 1,10)                    DSPLIN
      IF(ECHO.EQ.1.AND.IANLYS.NE.0)WRITE(IWR,7560)(ICANYS(I),I=1,10)    DSPLIN
      IF(IANLYS.NE.0.AND.ICANYS(8).EQ.1)READ(IRD,*)GSTART,GINT          DSPLIN
      IF(ECHO.NE.0.AND.ICANYS(8).EQ.1)WRITE(IWR,7530)GSTART,GINT        DSPLIN
C* OPTION RELATED TO THE TIMSC ROUTINE                                  DSPLIN
      IF(ISCT.EQ.1)READ(IRD,*)SCTIME                                    DSPLIN
      IF(ISCT.EQ.1.AND.ECHO.EQ.1)WRITE(IWR,7520)SCTIME                  DSPLIN
C* OPTION RELATED TO THE CHI SQUARED TEST ON VARIANCE FACTOR            DSPLIN
      IF(ICHI.EQ.1)READ(IRD,*)CALPHA                                    DSPLIN
      IF(ICHI.EQ.1.AND.ECHO.EQ.1)WRITE(IWR,7520)CALPHA                  DSPLIN
C* OPTIONS RELATED TO THE FILTER ROUTINE                                DSPLIN
      IF(IFLT.EQ.0)GO TO 10                                             DSPLIN
      READ(IRD,*)DINT,DINC,ALPH                                         DSPLIN
      IF(ECHO.EQ.1)WRITE(IWR,7520)DINT,DINC,ALPH                        DSPLIN
      FESC = ELSFAC(ALPH)                                               DSPLIN
C* CONFIDENCE ELLIPSE SET AT THE ALPHA PERCENTAGE LEVEL                 DSPLIN
 10   IF(IELIPS.NE.1)GO TO 20                                           DSPLIN
      READ(IRD,*)ALPHA                                                  DSPLIN
      WRITE(IWR,7520)ALPHA                                              DSPLIN
      ESC = ELSFAC(ALPHA)                                               DSPLIN
C* READ POINT OF START OF CURVE AND INTERVAL- FOR DESIGN AND ANALYS     DSPLIN
 20   IF(DRAW.EQ.0.AND.IANLYS.EQ.0)GO TO 999                            DSPLIN
```

```
      IF(DRAW.EQ.1)GO TO 35                                             DSPLIN
      IF(ICANYS(9).NE.1.AND.ICANYS(9).NE.2)GO TO 999                    DSPLIN
 35   READ(IRD,*)XSTART,XINT                                            DSPLIN
      IF(ECHO.EQ.1)WRITE(IWR,7530)XSTART,XINT                           DSPLIN
 999  IF(IREAD.GE.5)WRITE(6,7540)                                       DSPLIN
 7520 FORMAT(3X,'ECHO : ',6(1PD12.5,3X))                                DSPLIN
 7530 FORMAT(3X,'ECHO : ',6(1PD12.5,3X))                                DSPLIN
 7540 FORMAT(/,3X,'OPTION : INPUT VARIANCES WILL BE MODIFIED')          DSPLIN
 7560 FORMAT(/,3X,'OPTION LIST ',100(35I3,/,15X))                       DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE OUT(KNO,NO,R,M,N,                                      DSPLIN
     @               PRTX,PRTK,PRLOBS,PRDTK,PRCOV,PRPMAT,PRAMAT,        DSPLIN
     @               TX,TK,LOBS,DTK,VAR,P,A1,A2,ADDA1,ADDA2)            DSPLIN
C *****************************************************************     DSPLIN
C * VERSION          : 9 JUNE 1982                                *     DSPLIN
C * DESCRIPTION      : PRINTS THE MATRICES PRIOR TO LS COMPUTATIONS *   DSPLIN
C *****************************************************************     DSPLIN
C  INPUT PARAMETERS                                                     DSPLIN
C   KNO,NO,R,M,N - NO OF KNOTS,POINTS AND DIM OF LS PROBLEM             DSPLIN
C   PRTX,PRTK,PRLOBS,PRDTK,PRCOV,PRPMAT,PRAMAT - PRINT CODES            DSPLIN
C   TX,TK,LOBS,VAR,P,A1,A2,ADDA1,ADDA2 - ARRAYS AS IN MAIN PROGRAM      DSPLIN
C       ( VARIABLE NAMES ARE AS IN THE MAIN PROGRAM )                   DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C   NONE                                                                DSPLIN
C                                                                       DSPLIN
C CALLS - DPRINT                                                        DSPLIN
C                        ************                                   DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV      DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDX0,      DSPLIN
     @        IDCOF,IDVCOF                                              DSPLIN
      REAL*8 TX(IDTX,1),TK(IDTK,1),LOBS(IDLOBS,1),VAR(IDVAR,2,2),       DSPLIN
     @       DTK(IDDTK,1),P(IDP,3),A1(IDA1,7),A2(IDA2,1)                DSPLIN
      INTEGER KNO,NO,R,M,N,ADDA1(IDA1),ADDA2(IDA2),                     DSPLIN
     @       PRTX,PRTK,PRLOBS,PRDTK,PRCOV,PRPMAT,PRAMAT                 DSPLIN
C* WRITE UNIT                                                           DSPLIN
      IWR = 6                                                           DSPLIN
C* BEGIN                                                                DSPLIN
      L = NO + 1                                                        DSPLIN
      IL = NO + 2                                                       DSPLIN
      IF(PRTX.EQ.0)GO TO 10                                             DSPLIN
      WRITE(IWR,7000)(TX(I,1),I = 2,L)                                  DSPLIN
 10   IF(PRLOBS.EQ.1)CALL DPRINT(LOBS,N,1,IDLOBS,1,'OBS-VECTOR',5)      DSPLIN
      IF(PRCOV.EQ.0)GO TO 20                                            DSPLIN
      WRITE(IWR,7020)(((VAR(I,J,K),K = 1,2),J = 1,2),I = 1,IL)          DSPLIN
 20   IF(PRTK.EQ.1)CALL DPRINT(TK,KNO,1,IDTK,1,'TIME OF KNOTS.',7)      DSPLIN
      L = KNO - 1                                                       DSPLIN
```

```
      IF(PRDTK.EQ.1)CALL DPRINT(DTK,KNO,1,IDDTK,'INTERVAL BET KNOTS',9) DSPLIN
      IF(PRPMAT.EQ.1)CALL DPRINT(P,N,3,IDP,3,'BANDED  P-MATRIX',8)      DSPLIN
      IF(PRAMAT.NE.1)RETURN                                             DSPLIN
      L = N/2                                                           DSPLIN
      WRITE(IWR,7030)(ADDA1(I),I=1,L)                                   DSPLIN
      CALL DPRINT(A1,L,7,IDA1,7,'CONDENSED  A1-MATRIX',10)              DSPLIN
      L = R/2                                                           DSPLIN
      WRITE(IWR,7040)(ADDA2(I),I=1,L)                                   DSPLIN
      CALL DPRINT(A2,L,11,IDA2,11,'CONDENSED  A2-MATRIX',10)            DSPLIN
C*                                                                      DSPLIN
 7000 FORMAT(//,3X,'INPUT TIME OF DATA POINTS',//,                      DSPLIN
     @100(2X,8(1PD12.5,1X),/),//)                                       DSPLIN
 7020 FORMAT(//,3X,'INPUT COVARIANCE',//,                               DSPLIN
     @ 100(5X,1PD12.5,3X,1PD12.5,/,5X,1PD12.5,3X,1PD12.5,//),//)        DSPLIN
 7030 FORMAT(/,3X,'ADDRESS A1',/,3X,10(10I4,/,3X))                      DSPLIN
 7040 FORMAT(/,3X,'ADDRESS A2',/,3X,10(10I4,/,3X))                      DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE PLOT(ZIHF,ZIHC,ZIDPX,ZIDPY,ZLX,ZLY,ZUX,ZUY)            DSPLIN
C   ******************************************************************DSPLIN
C   * VERSION      : 24TH MARCH 1982                                   *DSPLIN
C   * DESCRIPTION : LAYS OUT THE PLOT REQUIREMENTS: PLOT SIZE, GRID ETC *DSPLIN
C   ******************************************************************DSPLIN
C   PARAMETERS REQUESTED BY SUBROUTINE - READ DIRECTLY FROM UNIT IRD    DSPLIN
C                                                                       DSPLIN
C     A)  READ IN THE TITLE TO BE PRINTED ON PLOT                       DSPLIN
C     B)  READ IN THE TITLE WITH THE X AXIS- EASTINGS                   DSPLIN
C     C)  READ IN THE TITLE WITH THE Y AXIS - NORTHINGS                 DSPLIN
C     D)  READ IN SIZE (INCH) AND CORNER COORD OF THE PLOT              DSPLIN
C     E)  READ THE ORIGIN(AXES) AND THE INTERVAL BET TICK MARKS         DSPLIN
C         FIRST ALONG X THEN ALONG Y                                    DSPLIN
C     F)  READ PRINT AXES OFFSET(OFFX,OFFY)                             DSPLIN
C             AND PRINT SCALE FACTORS(SCX,SCY)                          DSPLIN
C                                                                       DSPLIN
C CALLS - AREA,RECT,CHRPRT,AXS (FORTPLOT LIBRARY ROUTINES )             DSPLIN
C                                                                       DSPLIN
C ADDITIONAL INFORMATION                                                DSPLIN
C  COMPUTING THE LENGTH OF THE AXES                                     DSPLIN
C  WORKINGS - THE SECTION FIRST COMPUTES THE LENGTH OF THE AXES BY      DSPLIN
C             DEDUCTING 2* INTERVAL FORM THE PLOT SIZE. THIS IS THEN    DSPLIN
C             DIVIDED BY THE INTERVAL AND THE NEAREST INTEGER ACCEPTED  DSPLIN
C             AS THE MAX OF OF TICKS IN THE LINE. FROM THIS , THE       DSPLIN
C             LENGTH OF THE AXES ARE COMPUTED - IN INCHES.              DSPLIN
C  VARIABLES                                                            DSPLIN
C   ZINC - INTERVAL BETWEEN TICK MARKS ON THE X - AXIS                  DSPLIN
C   ZINT - INTERVAL BETWEEN TICK MARKS ON THE Y - AXIS                  DSPLIN
C   ZHT  - HEIGHT OF THE TITLE CHARACTERS IN Y USER UNITS               DSPLIN
C   ZIDP - THE VARIABLE TO BE SET IN USER UNITS FOR THE DISPLACEMENTS   DSPLIN
```

```
C           OF THE NUMBERS OF THE KNOTS AND NOT KNOTS FROM THE POINT.       DSPLIN
C    ZIHF - VARIABLE USED TO DEFINE THE HEIGHT OF THE SYMBOLS              DSPLIN
C           IN Y USER UNITS                                                 DSPLIN
C    ZCHT - VARIABLE USED TO DEFINE THE HT OF THE CHARACTERS ALONG THE     DSPLIN
C           AXES IN Y USER UNITS                                            DSPLIN
C    ZIHC - VARIABLE DEFINING THE RADIUS OF THE CIRCLE IN X USER UNITS     DSPLIN
C                                                                           DSPLIN
C    NOTE                                                                   DSPLIN
C         AREA WITHIN AXES IS LOWER LEFT (ZLX,ZLY)                          DSPLIN
C                                 UPPER RIGHT (ZUX,ZUY)                     DSPLIN
C         ZIHF AND ZIHC ARE THE NEGATIVE OF THEIR COORD VALUES             DSPLIN
C                                 ##################                        DSPLIN
C* SUBROUTINE BEGINS                                                        DSPLIN
      LOGICAL*1 XAXIS(50),YAXIS(45),TITLE(41)                               DSPLIN
      DATA XAXIS/49*' ',Z01/,YAXIS/44*' ',Z01/                              DSPLIN
      DATA TITLE/40*' ',Z01/                                                DSPLIN
C* READ FROM UNIT                                                           DSPLIN
      IRD = 5                                                               DSPLIN
C*                                                                          DSPLIN
      READ(IRD,9040)(TITLE(I),I = 1,40)                                     DSPLIN
      READ(IRD,9040)(XAXIS(I),I = 1,49)                                     DSPLIN
      READ(IRD,9040)(YAXIS(I),I = 1,44)                                     DSPLIN
      READ(IRD,*)ZXSIZE,ZYSIZE,ZXMIN,ZYMIN,ZXMAX,ZYMAX                      DSPLIN
      READ(IRD,*)ZAX,ZAY,ZINC,ZINT                                          DSPLIN
      READ(IRD,*)OFFX,OFFY,SCX,SCY                                          DSPLIN
C*                                                                          DSPLIN
      IF(SCX.EQ.0)SCX = 1.0                                                 DSPLIN
      IF(SCY.EQ.0)SCY = 1.0                                                 DSPLIN
      IF(ZINC.EQ.0)ZINC = 1.0                                               DSPLIN
      IF(ZINT.EQ.0)ZINT = 1.0                                               DSPLIN
      CALL AREA(ZXSIZE,ZYSIZE)                                              DSPLIN
      CALL SETPLT(ZXMIN,ZYMIN,ZXMAX,ZYMAX)                                  DSPLIN
C* CONSTRUCT RECTANGLE AROUND PHYSICAL PLOT SIZE                            DSPLIN
      ZX = ZXMAX - ZXMIN                                                    DSPLIN
      ZY = ZYMAX - ZYMIN                                                    DSPLIN
      ZXMIN = ZXMIN - ZX/1000                                               DSPLIN
      ZYMIN = ZYMIN - ZY/1000                                               DSPLIN
C* SCALE                                                                    DSPLIN
      XSC = ZX/ZXSIZE                                                       DSPLIN
      YSC = ZY/ZYSIZE                                                       DSPLIN
      CALL RECT(ZXMIN,ZYMIN,-ZX,-ZY)                                        DSPLIN
C* PRINTING OF TITLE ON PLOT                                                DSPLIN
      ZHT = ZX/44                                                           DSPLIN
      ZHTY = ZHT*(YSC/XSC)                                                  DSPLIN
      ZXP = ZXMIN + 2.0*ZHT                                                 DSPLIN
      ZYP = ZYMAX - 2.0*ZHTY                                                DSPLIN
      CALL CHRPRT(ZXP,ZYP,TITLE,0.0,-ZHTY)                                  DSPLIN
C* DRAWING THE AXES                                                         DSPLIN
```

```
      ZX = ZX - 2.0*(ZAX-ZXMIN)                                         DSPLIN
      ZY = ZY - 2.0*(ZAY-ZYMIN)                                         DSPLIN
      ILX = (ZX/ZINC) + 0.5                                             DSPLIN
      ILY = (ZY/ZINT) + 0.5                                             DSPLIN
      ZLX = (ILX*ZINC)/XSC                                              DSPLIN
      ZLY = (ILY*ZINT)/YSC                                              DSPLIN
      ZIDPX = ZHT                                                       DSPLIN
      ZIDPY = ZHTY                                                      DSPLIN
      ZIHF = -ZHTY/2.0                                                  DSPLIN
      ZCHT = -ZHTY*(2.0/3.0)                                            DSPLIN
      ZIHC = -ZHT/4.0                                                   DSPLIN
C* THIS IS TO MODIFY THE NUMBERS PLOTTED WITH THE AXES SO THAT THEY FIT DSPLIN
C* THE FORMAT DEFINED BY THE AXS ROUTINE.                               DSPLIN
C* FOR THE X AXIS                                                       DSPLIN
      ZAT = (ZAX - OFFX)/SCX                                            DSPLIN
      ZINC = ZINC/SCX                                                   DSPLIN
      CALL AXS(ZAX,ZAY,XAXIS,ZLX,0.0,ZAT,ZINC,-ILX,ZCHT)                DSPLIN
C* FOR THE Y AXIS                                                       DSPLIN
      ZAT = (ZAY - OFFY)/SCY                                            DSPLIN
      ZINX = ZINT/SCY                                                   DSPLIN
      CALL AXS(ZAX,ZAY,YAXIS,ZLY,1.57,ZAT,ZINX,ILY,ZCHT)                DSPLIN
C* SETTING THE LIMITS OF THE AREA DEFINED BY THE AXES                   DSPLIN
      ZUX = ZAX + ZLX*XSC                                               DSPLIN
      ZUY = ZAY + ZLY*YSC                                               DSPLIN
      ZLX = ZAX                                                         DSPLIN
      ZLY = ZAY                                                         DSPLIN
 9040 FORMAT(100A1)                                                     DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE POINT(T,X1,X2,VX1,VX2,VX12,ICODE,KNO,TK,COF,VCOF)      DSPLIN
      IMPLICIT REAL*8(A-H,N-Z)                                          DSPLIN
C ***************************************************************       DSPLIN
C * VERSION       : 21ST MARCH 1982                             *       DSPLIN
C * DESCRIPTION   : COMPUTES THE POSTION AND COVARIANCE OF A    *       DSPLIN
C *                 POINT AT TIME T .                           *       DSPLIN
C ***************************************************************       DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  T        - TIME AT WHICH THE POSITION AND COVARIANCE IS REQUIRED     DSPLIN
C  KNO      - NO OF KNOTS                                               DSPLIN
C  TK       - VECTOR OF KNOT TIMES                                      DSPLIN
C  COF,VCOF- COEFFICIENTS AND COVARIANCE MATRICES                       DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  X1,X2    - THE POSITION OF THE POINT                                 DSPLIN
C  VX1,VX2 - VARIANCES IN X1 AND X2 RESPECTIVELY                        DSPLIN
C  VX12     - COVARIANCE BETWEEN X1 AND X2 COORDINATES                  DSPLIN
C  ICODE    - 1  SUCCESSFUL LOCATION OF INTERVAL                        DSPLIN
C             0  CANNOT LOCATE TIME WITHIN DEFINED INTERVALS            DSPLIN
```

```
      C                                                                 DSPLIN
      C CALLS - INTERV,FORM,DMULT                                       DSPLIN
      C       - TRAPS                                                   DSPLIN
      C              ####################                               DSPLIN
      C*   PROGRAM BEGINS                                               DSPLIN
            COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV DSPLIN
            COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO, DSPLIN
           @         IDCOF,IDVCOF                                       DSPLIN
            REAL*8 X(2,1),VX(2,2),B(2,8),TK(IDTK,1),                    DSPLIN
           @         COF(IDCOF,8),VCOF(IDVCOF,8,8),CO(8,1),VCO(8,8)     DSPLIN
            INTEGER*4 KNO                                               DSPLIN
            DATA ISWT/0/,B/16*0.0D0/                                    DSPLIN
            ICODE = 1                                                   DSPLIN
            CALL INTERV(T,TK,KNO,J,IDTK,IC)                             DSPLIN
            IF(IC.EQ.0)GO TO 10                                         DSPLIN
            DR = T - TK(J,1)                                            DSPLIN
            CALL FORM(DR,B)                                             DSPLIN
      C** EXTRACTING THE COEFFICIENTS AND COVARIANCES                   DSPLIN
            IF(ISWT.EQ.J)GO TO 30                                       DSPLIN
            DO 20 IJ = 1,8                                              DSPLIN
            CO(IJ,1) = COF(J,IJ)                                        DSPLIN
            DO 20 IK = 1,8                                              DSPLIN
         20 VCO(IJ,IK) = VCOF(J,IJ,IK)                                  DSPLIN
            ISWT = J                                                    DSPLIN
            CALL TRAPS(0,0,10000,0,0)                                   DSPLIN
         30 CALL DMULT(B,2,8,CO,8,1,X,2,8,8,1,2,1)                      DSPLIN
            DO 40 I = 1,2                                               DSPLIN
             DO 40 L = 1,2                                              DSPLIN
             VX(I,L) = 0.0D0                                            DSPLIN
              DO 40 J = 1,8                                             DSPLIN
               DO 40 K = 1,8                                            DSPLIN
         40 VX(I,L) = VX(I,L) + B(I,K)*VCO(K,J)*B(L,J)                  DSPLIN
            X1 = X(1,1)                                                 DSPLIN
            X2 = X(2,1)                                                 DSPLIN
            VX1 = VX(1,1)                                               DSPLIN
            VX2 = VX(2,2)                                               DSPLIN
            VX12 = VX(1,2)                                              DSPLIN
            RETURN                                                      DSPLIN
         10 WRITE(6,900)T                                               DSPLIN
            ICODE = 0                                                   DSPLIN
            RETURN                                                      DSPLIN
        900 FORMAT(//,2X,'TIME ',D12.5,' AT WHICH POSITICN IS REQUESTEC ', DSPLIN
           @           /,'LIES OUTSIDE THE TIME DEFINED BY THE OUTER',  DSPLIN
           @           ' KNOTS',//)                                     DSPLIN
            END                                                         DSPLIN
            SUBROUTINE PREPAR(KNO,NO,R,M,N, TX,TK,DTK,AOBS,ACON,        DSPLIN
           @                  A1,ADDA1,A2,ADDA2,                        DSPLIN
           @                  XO,LCBS,W1,W2,IAPRI)                      DSPLIN
```

```
      IMPLICIT REAL*8(A-H,O-Y),REAL*4(Z)                                DSPLIN
C ********************************************************************DSPLIN
C *   VERSION       :  7 JUNE 1982                                      DSPLIN
C *   DESCRIPTION   :  PREPARES MATRICES FOR INPUT INTO THE LS ROUTINE  DSPLIN
C ********************************************************************DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  KNO,NO,R,M,N - AS IN MAIN                                            DSPLIN
C  TX,TK,DTK,XO,LOBS                                                    DSPLIN
C                 - AS IN MAIN                                          DSPLIN
C  IAPRI          - INPUT CODE OF APRIORI SOLN VECTOR                   DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  A1,A2          - VARIABLE PROFILE BANDED DESIGN MATRICES( DIM BELOW) DSPLIN
C  ADDA1,ADDA2    - ADDRESS SEQUENCE FOR A1 AND A2 MATRICES RESP.       DSPLIN
C  W1,W2          - MISCLOSURE VECTORS                                  DSPLIN
C                                                                       DSPLIN
C WORK ARRAYS                                                           DSPLIN
C AOBS,ACON - AS DIMENSIONED IN MAIN                                    DSPLIN
C                                                                       DSPLIN
C CALLS - DCLEAR,INTERV,CONSTR,APRORI,OBS                               DSPLIN
C                             *************                             DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,ICVAR,ICV      DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,      DSPLIN
     @        IDCOF,IDVCOF                                              DSPLIN
C* ARRAYS                                                               DSPLIN
      REAL*8 A1(IDA1,7),A2(IDA2,11),TX(IDTX,1),TK(IDTK,1),DTK(IDDTK,1), DSPLIN
     @       Z0,AOBS(7,1),ACON(11,1),                                   DSPLIN
     @       XO(IDXO,1),LOBS(IDLOBS,1),W1(IDW1,1),W2(IDW2,1)            DSPLIN
C* OTHERS                                                               DSPLIN
      INTEGER ADDA1(IDA1),ADDA2(IDA2),KNO,NO,R,M,N,IAPRI                DSPLIN
C* FUNCTIONS                                                            DSPLIN
      IROW(I) = (I+1)/2                                                 DSPLIN
      ICOL(I) = (2-((I+1)/2 - (I/2)))                                   DSPLIN
C* PSEUDO MATRIX SIZES  A1(N,M) , A2(R,M) , P(N,N) , VAR(N-R,N)         DSPLIN
      R = 2*KNO - 4                                                     DSPLIN
      M = 4*KNO                                                         DSPLIN
      N = 2*NO+ 8                                                       DSPLIN
C* CLEARING THE A MATRICES                                              DSPLIN
      CALL DCLEAR(A1,IDA1,7)                                            DSPLIN
      CALL DCLEAR(A2,IDA2,11)                                           DSPLIN
C* FORMULATION OF THE A1 MATRIX                                         DSPLIN
      J = 1                                                             DSPLIN
      L = NO + 2                                                        DSPLIN
      DO 31 II = 3,L                                                    DSPLIN
      J = J + 1                                                         DSPLIN
      CALL INTERV(TX(J,1),TK,KNO,INT,IDTK,ICODE)                        DSPLIN
      IF(ICODE.EQ.0)WRITE(6,9000)J                                      DSPLIN
      IF(ICODE.EQ.0)STOP                                                DSPLIN
```

```
      IR = (INT-1)*4 + 1                                                DSPLIN
      CALL OBS(TX(J,1),TK(INT,1),DTK(INT,1),AOBS)                       DSPLIN
      ADDA1(II) = IR                                                    DSPLIN
      DO 41 I = 1,7                                                     DSPLIN
   41 A1(II,I) = AOBS(I,1)                                              DSPLIN
   31 CONTINUE                                                          DSPLIN
C* FORMULATION OF THE REMAINDER OF THE A1 MATRIX - OUTER KNOTS          DSPLIN
      J = -1                                                            DSPLIN
      DO 101 I = 1,2                                                    DSPLIN
      J = J + 2                                                         DSPLIN
      ADDA1(I) = J                                                      DSPLIN
  101 A1(I,1) = 1.0D0                                                   DSPLIN
      J = -1                                                            DSPLIN
      LK = NO + 3                                                       DSPLIN
      L  = NO + 4                                                       DSPLIN
C* SLIGHT MANIPULATION NEEDED TO MAKE THE LAST 4 X 4 BLOCK DIAGONAL     DSPLIN
      DO 111 I = LK,L                                                   DSPLIN
      J = J + 2                                                         DSPLIN
      II = M + ( J - 4)                                                 DSPLIN
      ADDA1(I) = II                                                     DSPLIN
  111 A1(I,1) = 1.0D00                                                  DSPLIN
C* FORMULATICN OF A2 MATRIX - SINGULAR                                  DSPLIN
      J = 0                                                             DSPLIN
      K = -3                                                            DSPLIN
      LK = R/2                                                          DSPLIN
      DO 121 I = 1,LK                                                   DSPLIN
      J = J + 1                                                         DSPLIN
      J1 = J + 1                                                        DSPLIN
      K = K + 4                                                         DSPLIN
      CALL CONSTR(DTK(J,1),DTK(J1,1),ACON)                              DSPLIN
      ADDA2(I) = K                                                      DSPLIN
      DO 51 II = 1,11                                                   DSPLIN
   51 A2(I,II) = ACON(II,1)                                             DSPLIN
  121 CONTINUE                                                          DSPLIN
C* INCOPORATION OF THE APRIORI VALUES IF ANY                            DSPLIN
      CALL APRORI(XO,M,IDXO,IAPRI)                                      DSPLIN
      IF(IAPRI.GT.0)CALL DPRINT(XO,M,1,IDXO,'APRIORI VALUES',7)         DSPLIN
C* COMPUTE MISCLOSE VECTORS                                             DSPLIN
      DO 131 I = 1,N                                                    DSPLIN
      SUM = 0.0D0                                                       DSPLIN
      DO 132 J = 1,M                                                    DSPLIN
      II = IROW(I)                                                      DSPLIN
      JJ = J - (ADDA1(II) + ICOL(I) -1 ) + 1                            DSPLIN
      IF(JJ.LT.1.OR.JJ.GT.7)GO TO 132                                   DSPLIN
      SUM = SUM + A1(II,JJ)*XO(J,1)                                     DSPLIN
  132 CONTINUE                                                          DSPLIN
  131 W1(I,1) = SUM - LOBS(I,1)                                         DSPLIN
      DO 133 I = 1,R                                                    DSPLIN
```

```fortran
      W2(I,1) = 0.0D0
      DO 133 J = 1,M
      II = IROW(I)
      JJ = J - (ADDA2(II) + ICOL(I) -1 ) + 1
      IF(JJ.LT.1.OR.JJ.GT.11)GO TO 133
      W2(I,1) = W2(I,1) + A2(II,JJ)*XO(J,1)
  133 CONTINUE
 9000 FORMAT(///,'  DATA POINT ',I4,' NOT WITHIN INTERVAL SPECIFIED',
     @' BY END KNOTS',//)
      RETURN
      END
      SUBROUTINE  PRNKNT(X,M,IDX,VX,IDVX,PRKNT)
      IMPLICIT REAL*8 (A-H,P-Z)
C ****************************************************************
C * VERSION       : 1 AUGUST 1982                                *
C * DESCRIPTION : TO PRINT THE POSITION AND SLOPE                *
C *               INFORMATION OF THE PRKNT-TH  KNOT.             *
C ****************************************************************
C INPUT PARAMETERS
C  X,VX    - L.S ESTIMATES FROM THE LS SUBROUTINE
C  M       - ACTUAL DIMENSION OF X AND ROW DIMENSION OF VX
C  IDX     - DECLARED DIMENSION OF X VECTOR
C  IDVX    - DECLARED DIMENSION OF VX MATRIX
C  PRKNT   - THE NUMBER OF THE KNOT THAT INFORMATION IS REQUIRED
C
C OUTPUT PARAMETERS
C  NONE
C
C CALLS    - DSQRT
C
C ** BEGINS
      REAL*8 X(IDX,1),VX(IDVX,IDVX),STD(4,1),CORR1,CORR2
      INTEGER PRKNT
C* ASSIGN WRITE UNIT
      IWR = 6
C* LOOP POINTERS ARE BEING DETERMINED
      IB  = PRKNT*4 - 3
      IE  = PRKNT*4
C* CHECK FOR ERRONEOUS KNOT REQUEST INPUT
      IF(IE.GT.M)GO TO 999
C* EXTRACT STD DEVIATIONS AND CORRELATIONS
      DO 100 I = IB,IE
          II = I - IB + 1
  100     STD(II,1) = DSQRT(VX(I,I))
      II = IB + 1
      CORR1 = (VX(IB,II)/(STD(1,1)*STD(2,1)))*100
      II = IE-1
      CORR2 = (VX(IE,II)/(STD(3,1)*STD(4,1)))*100
```

```
      WRITE(IWR,10)PRKNT                                                DSPLIN
      WRITE(IWR,20)(X(I,1),I=IB,IE)                                     DSPLIN
      WRITE(IWR,30)                                                     DSPLIN
      WRITE(IWR,20)((VX(I,J),J=IB,IE),I=IB,IE)                          DSPLIN
      WRITE(IWR,50)(STD(I,1),I=1,4),CORR1,CORR2                         DSPLIN
      RETURN                                                            DSPLIN
  999 WRITE(IWR,40)                                                     DSPLIN
      RETURN                                                            DSPLIN
   10 FORMAT(//,3X,'ELEMENTS OF THE ',I2,'TH KNOT VECTOR',              DSPLIN
     @       /,3X,'======================================',/)           DSPLIN
   20 FORMAT(/,100(/,3X,4(1PD14.7,1X)))                                 DSPLIN
   30 FORMAT(//,3X,'COVARIANCE OF THE KNOT PARAMETERS',                 DSPLIN
     @       /,3X,'=================================',/)                DSPLIN
   40 FORMAT(//,3X,'SUBR PRNKNT - KNOT INFORMATION REQUESTED ',         DSPLIN
     @              'IS GREATER THAN THE NO. OF KNOTS',//)              DSPLIN
   50 FORMAT(/,3X,'SQUARE ROOTED DIAGONAL ELEMENTS OF ABOVE',           DSPLIN
     @       /,3X,4(1PD14.7,1X),                                        DSPLIN
     @      //,3X,'CORRELATION COEFFICIENTS (*100)',                    DSPLIN
     @       /,3X,2(1PD14.7,1X))                                        DSPLIN
      END                                                               DSPLIN
      SUBROUTINE RDATA(NO,TX,VAR,LOBS,P,MAT,                            DSPLIN
     @                   IREAD,ECHO,XX,YY,CXY,TUME)                     DSPLIN
C ******************************************************************** DSPLIN
C * VERSION     : 7 JUNE 1982                                        * DSPLIN
C * DESCRIPTION : READING OF DATA POINTS AND END KNOT VECTORS.THIS   * DSPLIN
C *               SUBROUTINE IS THE DRIVER FOR THE READ ROUTINE      * DSPLIN
C ******************************************************************** DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C    NO    - NO. OF DATA POINTS                                         DSPLIN
C    IREAD - OPTION CODE FOR READ FORMAT(SEE READ ROUTINE FOR CODES)    DSPLIN
C    ECHO  - OPTION CODE FOR ECHOING ALL INPUT TO PROGRAM               DSPLIN
C            1 - YES , 0 - NO                                           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C    TX    - VECTOR OF KNOT TIMES ( DIM IN MAIN )                       DSPLIN
C    VAR   - ARRAY OF INPUT COVARIANCES OF DATA POINTS ( DIM IN MAIN)   DSPLIN
C    LOBS  - VECTOR OF OBSERVABLES ( DIM IN MAIN )                      DSPLIN
C    P     - FIXED BANDED WEIGHT MATRIX FOR OBSERVABLES ( DIM IN MAIN)  DSPLIN
C                                                                       DSPLIN
C WORK VARIABLES/ARRAYS                                                 DSPLIN
C    MAT,XX,YY,CXY,TUME                                                 DSPLIN
C                                                                       DSPLIN
C CALLS - READ, CHOLD OR DINV                                           DSPLIN
C ADDITIONAL INFORMATION                                                DSPLIN
C    READING IN ,REARRANGING AND STORAGE OF INPUT DATA POINTS           DSPLIN
C    NOTE THE ISWT SWITCH IS USED TO DETECT THE 1ST AND LAST POINT      DSPLIN
C    AND TO PREVENT FAILURE IN THE DO LOOP                              DSPLIN
C                                                                       DSPLIN
```

```
      COMMON /DIM1/ICBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV       DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,       DSPLIN
     @          IDCOF,IDVCOF                                             DSPLIN
      REAL*8 TX(IDTX,1),VAR(IDVAR,2,2),MAT(2,2),LOBS(IDLOBS,1),          DSPLIN
     @          P(IDP,3),TUME,XX,YY,CXY,DET                              DSPLIN
      INTEGER ECHO,IREAD                                                 DSPLIN
      J = -1                                                             DSPLIN
      L = NO + 2                                                         DSPLIN
      ISWT = 1                                                           DSPLIN
      DO 11 I = 1,L                                                      DSPLIN
      J = J + 2                                                          DSPLIN
      J1 = J + 1                                                         DSPLIN
      CALL READ(L,I,TX(I,1),XX,YY,MAT(1,1),MAT(2,2),CXY,IREAD,ECHO)      DSPLIN
C* REASSIGNMENT                                                          DSPLIN
      VAR(I,1,1) = MAT(1,1)*MAT(1,1)                                     DSPLIN
      VAR(I,2,2) = MAT(2,2)*MAT(2,2)                                     DSPLIN
      VAR(I,1,2) = CXY                                                   DSPLIN
      VAR(I,2,1) = VAR(I,1,2)                                            DSPLIN
   12 LOBS(J,1) = XX                                                     DSPLIN
      LOBS(J1,1) = YY                                                    DSPLIN
C* STORE IN MAT                                                          DSPLIN
      MAT(1,1) = MAT(1,1)*MAT(1,1)                                       DSPLIN
      MAT(2,2) = MAT(2,2)*MAT(2,2)                                       DSPLIN
      MAT(1,2) = CXY                                                     DSPLIN
      MAT(2,1) = MAT(1,2)                                                DSPLIN
C* INVERSE                                                               DSPLIN
      CALL CHOLD(MAT,2,2,2,DET,IER)                                      DSPLIN
      IF(IER.NE.0)WRITE(6,9000)                                          DSPLIN
      IF(IER.NE.0)STOP                                                   DSPLIN
C* REASSIGNMENT                                                          DSPLIN
      P(J,2) = MAT(1,1)                                                  DSPLIN
      P(J,3) = MAT(1,2)                                                  DSPLIN
      P(J1,1) = MAT(2,1)                                                 DSPLIN
      P(J1,2) = MAT(2,2)                                                 DSPLIN
C* CLEAR                                                                 DSPLIN
      P(J,1) = 0                                                         DSPLIN
      P(J1,3) = 0                                                        DSPLIN
      IF(I.NE.1.AND.I.NE.L)GO TO 11                                      DSPLIN
      IF(I.EQ.1.AND.ISWT.EQ.2)GO TO 11                                   DSPLIN
      IF(I.EQ.L.AND.ISWT.EQ.3)GO TO 999                                  DSPLIN
      J = J + 2                                                          DSPLIN
      J1 = J + 1                                                         DSPLIN
C* TUME - DUMMY                                                          DSPLIN
      CALL READ(L,I,TUME,XX,YY,MAT(1,1),MAT(2,2),CXY,4,ECHO)             DSPLIN
      ISWT = 1 + ISWT                                                    DSPLIN
      GO TO 12                                                           DSPLIN
   11 CONTINUE                                                           DSPLIN
  999 RETURN                                                             DSPLIN
```

```
 9000 FORMAT(/,'  ERROR : INPUT COVARIANCE SINGULAR - PROG STOPPED ',//)DSPLIN
      END                                                               DSPLIN
      SUBROUTINE READ(IEND,LINE,T,X,Y,SX,SY,SXY,ICODE1,ICODE2)          DSPLIN
      IMPLICIT REAL*8(A-H,C-Z)                                          DSPLIN
      REAL*8 T,X,Y,SX,SY,SXY                                            DSPLIN
C *********************************************************           DSPLIN
C *   VERSION      : 12TH SEPT  1982                         *           DSPLIN
C *   DESCRIPTION : READ IN DATA ACCORDING TO OPTION CODE     *           DSPLIN
C *********************************************************           DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  ICODE1  - CHOICE OF THE DIFFERENT READS                              DSPLIN
C             1 - NORMAL INPUT                                          DSPLIN
C             2 - NORMAL INPUT BUT WITH CORRELATION COEFFICIENT         DSPLIN
C             3 - LOREX DATA FORMAT WITH NO MODIFICATION TO VARIANCES   DSPLIN
C             4 - BEGININING AND END KNOT SLOPE AND VARIANCE INTAKE     DSPLIN
C             5 - LOREX DATA FORMAT WITH MODIFICATION TO VARIANCES      DSPLIN
C             7 - LOREX DATA FORMAT WITH VELO AND DIR (CORRS APPLIED)   DSPLIN
C             9 - LOREX DATA FORMAT WITH VELO AND DIR (CORRS NOT APPLIED)DSPLIN
C  ICODE2  - INPUT ECHO REQUEST ;  1 - YES MODIF TC VAR PRINTED.        DSPLIN
C                                  0 - NO                               DSPLIN
C  IEND    - TOTAL NUMBER OF LINES TO BE READ(END KNOT INPUT DETECTN)   DSPLIN
C  LINE    - CURRENT LINE OF INPUT                                      DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  T,X,Y   - DATA POINT DEFINED BY COORDS X,Y, AT TIME T                DSPLIN
C  SX,SY   - STANDARD DEVIATIONS OF X,Y                                 DSPLIN
C  SXY     - COVARIANCE                                                 DSPLIN
C                                                                       DSPLIN
C CALLS    - UPSTD, ERRORS                                              DSPLIN
C          - DABS                                                       DSPLIN
C                                                                       DSPLIN
C PROGRAM BEGINS                                                        DSPLIN
C* READ INPUT UNIT NUMBER - IRD                                         DSPLIN
      IRD = 5                                                           DSPLIN
C* WRITE INPUT UNIT NUMBER - IWR                                        DSPLIN
      IWR = 6                                                           DSPLIN
C*                                                                      DSPLIN
      IF(ICODE1.NE.1)GO TO 10                                           DSPLIN
C* NORMAL INPUT ; COVARIANCE READ                                       DSPLIN
      READ(IRD,*)T,X,Y,SX,SY,SXY                                        DSPLIN
      IF(ICODE2.EQ.1)WRITE(IWR,110)T,X,Y,SX,SY,SXY                      DSPLIN
      RETURN                                                            DSPLIN
 10   IF(ICODE1.NE.2)GO TO 20                                           DSPLIN
C* CORRELATION COEFFICIENT READ                                         DSPLIN
      READ(IRD,*)T,X,Y,SX,SY,SXY                                        DSPLIN
      IF(ICODE2.EQ.1)WRITE(IWR,110)T,X,Y,SX,SY,SXY                      DSPLIN
      SXY = (SXY/100.0D0)*SX*SY                                         DSPLIN
      RETURN                                                            DSPLIN
```

```
 20    IF(ICODE1.NE.3.AND.ICODE1.NE.5                                   DSPLIN
      @ .AND.ICODE1.NE.7.AND.ICODE1.NE.9)GO TO 30                       DSPLIN
C ----------------------------------------------------------------------DSPLIN
C     INPUT VARIATION WITH A CORRELATION COEFFICIENT                    DSPLIN
C     NOTE THAT THE X,Y (EAST,NORTH) READ SEQUENCE HAS BEEN CHANGED     DSPLIN
C        TO MATCH LOREX INPUT DATA FORMAT AND THAT THE VELOCITY AND THE DSPLIN
C        DIRECTION OF MOTION IS READ IN                                 DSPLIN
C        VELO AND DIREC ARE REQUESTED ONLY FOR THE FIXES AND NOT END    DSPLIN
C        KNOT INFORMATION.                                              DSPLIN
C ----------------------------------------------------------------------DSPLIN
C* END KNOT INFORMATION INTAKE                                          DSPLIN
       IF(LINE.NE.1.AND.LINE.NE.IEND)GO TO 23                           DSPLIN
       READ(IRD,*)IDAY,IHRMIN,Y,X,SY,SX,SXY                             DSPLIN
       IF(ICODE2.EQ.1)WRITE(IWR,120)IDAY,IHRMIN,Y,X,SY,SX,SXY           DSPLIN
       GO TO 24                                                         DSPLIN
C* DATA POINT ,VELO AND DIRECTION INTAKE                                DSPLIN
 23    IF(ICODE1.EQ.3.OR.ICODE1.EQ.5)READ(IRD,*)IDAY,IHRMIN,Y,X,SY,SX,SXYDSPLIN
       IF(ICODE1.EQ.7)READ(IRD,*)IDAY,IHRMIN,Y,X,SY,SX,SXY,VELO,DR      DSPLIN
       IF(ICODE1.EQ.9)READ(IRD,*)IDAY,IHRMIN,Y,X,SY,SX,SXY,VELO,DR      DSPLIN
C* ECHO OUTPUT                                                          DSPLIN
       IF(ICODE2.EQ.1.AND.ICODE1.EQ.7)                                  DSPLIN
      @              WRITE(IWR,120)IDAY,IHRMIN,Y,X,SY,SX,SXY,VELO,DR    DSPLIN
       IF(ICODE2.EQ.1.AND.(ICODE1.EQ.3.OR.ICODE1.EQ.5.OR.ICODE1.EQ.9))  DSPLIN
      @              WRITE(IWR,120)IDAY,IHRMIN,Y,X,SY,SX,SXY            DSPLIN
C* CONVERSION OF CORRELATION TO COVARIANCE                              DSPLIN
 24    IF(DABS(SXY).EQ.100.0)SXY = (SXY/DABS(SXY))*99.999D0             DSPLIN
       SXY = (SXY/100.0D0)*SX*SY                                        DSPLIN
       IHR = IHRMIN/100                                                 DSPLIN
       XMIN = (IHRMIN - (IHRMIN/100)*100)                               DSPLIN
       T = IDAY + (IHR/24.0D0) + (XMIN/1440.0D0)                        DSPLIN
       IF(ICODE1.NE.5.AND.ICODE1.NE.7)RETURN                            DSPLIN
C* AVOID MODIFICATIONS ON THE FIRST AND LAST LINES                      DSPLIN
         IF(LINE.EQ.1.OR.LINE.EQ.IEND)RETURN                            DSPLIN
C* STANDARD DEVIATION CORRECTION                                        DSPLIN
         CALL UPSTD(SX,SY,SXY,ICODE2)                                   DSPLIN
C* CROSS TRACK CORRECTION                                               DSPLIN
         IF(ICODE1.NE.7)RETURN                                          DSPLIN
         DIR = DR*(3.14159265D0/180.0D0)                                DSPLIN
         CALL ERRORS(X,Y,SX,SY,SXY,VELO,DIR,ICODE2)                     DSPLIN
       RETURN                                                           DSPLIN
 30    IF(ICODE1.NE.4)GO TO 500                                         DSPLIN
C* SLOPE AND ITS CORRELATION INPUT                                      DSPLIN
       READ(IRD,*)Y,X,SY,SX,SXY                                         DSPLIN
       IF(ICODE2.EQ.1)WRITE(IWR,110)Y,X,SY,SX,SXY                       DSPLIN
       SXY = (SXY/100.0D0)*SX*SY                                        DSPLIN
       RETURN                                                           DSPLIN
C*                                                                      DSPLIN
C* FURTHER INPUT FORMATS TO BE HERE                                     DSPLIN
```

```
C*
 500    WRITE(IWR,130)ICODE1                                            DSPLIN
        RETURN                                                          DSPLIN
 110    FORMAT(3X,'ECHO : ',6(F12.1,3X))                                DSPLIN
 120    FORMAT(3X,'ECHO : ',I3,I6,3X,7(F12.1,3X))                       DSPLIN
 130    FORMAT(3X,'CODING ERROR ** SUBROUTINE READ   ; ',I4,//)         DSPLIN
        END                                                             DSPLIN
        SUBROUTINE REJECT(IDEGOF,S0,ALPHA,ICODE)                        DSPLIN
C   ***********************************************************         DSPLIN
C   * VERSION        : 21 JUNE 1982                            *        DSPLIN
C   * DESCRIPTION    : REJECTION CRITERA BASED ON THE CHI-SQUARED *     DSPLIN
C   *                  TEST ON THE VARIANCE FACTOR. ALPHA IS THE  *     DSPLIN
C   *                  CONFIDENCE INTERVAL USED IN THE TEST.    *       DSPLIN
C   ***********************************************************         DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C    IDEGOF   - DEGREES OF FREEDOM                                      DSPLIN
C    S0       - VARIANCE FACTOR                                         DSPLIN
C    ALPHA    - PERCENTAGE CONFIDENCE INTERVAL                          DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C    ICODE  - 11  PASSED THE TEST                                       DSPLIN
C    ICODE  - 22  FAILED THE TEST                                       DSPLIN
C                                                                       DSPLIN
C CALLS  - CHITES                                                       DSPLIN
C PROGRAM BEGINS                                                        DSPLIN
        REAL*8 S0,ALPHA                                                 DSPLIN
        INTEGER  IDEGOF,ICODE,ICODE2                                    DSPLIN
C CALL THE CHI-SQUARE TEST ROUTINE CHITES                               DSPLIN
        CALL CHITES(S0,IDEGOF,ALPHA,ICODE2)                             DSPLIN
C CODE ASSIGNMENT                                                       DSPLIN
        ICODE = 22                                                      DSPLIN
        IF(ICODE2.EQ.0)ICODE = 11                                       DSPLIN
        IF(ICODE2.EQ.1)ICODE = 11                                       DSPLIN
        RETURN                                                          DSPLIN
        END                                                             DSPLIN
        SUBROUTINE RESID(A,N,M,DX,W,V,IA,JA,ID,IW,IV)                   DSPLIN
C ******************************************************                DSPLIN
C * VERSION      : JAN 1982                            *                DSPLIN
C * DESCRIPTION : COMPUTES RESIDUAL VECTOR FOR SOLV ROUTINE *           DSPLIN
C ******************************************************                DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  A (N,M) - DESIGN MATRIX (DECLARED DIM A<IA,IJ>)                      DSPLIN
C  DX (M)  - LS ESTIMATES  (DECLARED DIM DX<ID>)                        DSPLIN
C  W   (N) - MISCLOSURE VECTOR(DEC DIM W<IW>)                           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMTERS                                                      DSPLIN
C  V (N)    - RESIDUAL VECTOR (DEC DIM V<IV>)                           DSPLIN
C                                                                       DSPLIN
```

```
C* ROUTINE BEGINS                                                            DSPLIN
         REAL*8 A(IA,JA),DX(ID,1),W(IW,1),V(IV,1)                            DSPLIN
C* FUNCTIONS                                                                 DSPLIN
        IROW(I) = (I+1)/2                                                    DSPLIN
        ICOL(I) = 2 - ((I+1)/2 - (I/2))                                      DSPLIN
C*                                                                           DSPLIN
        DO 10 I = 1,N                                                        DSPLIN
            V(I,1) = 0.0D0                                                   DSPLIN
            II = IROW(I)                                                     DSPLIN
           DO 20 J = 1,M                                                     DSPLIN
            JJ = J - ICOL(I) + 1                                             DSPLIN
            IF(JJ.LT.1.OR.JJ.GT.JA)GO TO 20                                  DSPLIN
            V(I,1) = V(I,1) + A(II,JJ)*DX(J,1)                               DSPLIN
   20      CONTINUE                                                          DSPLIN
  10    V(I,1) = -(V(I,1) + W(I,1))                                          DSPLIN
        RETURN                                                               DSPLIN
        END                                                                  DSPLIN
        SUBROUTINE REMAIN(R,M,N,A1,A2,ADDA1,ADDA2,W1,W2,X,V,V2)              DSPLIN
C ********************************************************************      DSPLIN
C * VERSION      : 7 JUNE 1982                                       *      DSPLIN
C * DESCRIPTION : COMPUTE THE RESIDUALS OF OBSERVATIONS AND          *      DSPLIN
C *               CONSTRAINTS. THE LATTER IS USED AS A CHECK ON THE  *      DSPLIN
C *               PRECISION OF THE COMPUTATIONS.                     *      DSPLIN
C ********************************************************************      DSPLIN
C INPUT PARAMETERS                                                           DSPLIN
C  R,M,N         - DIMENSION OF LS PROBLEM (SEE MAIN OR LS ROUTINE)          DSPLIN
C  A1,A2         - VARIABLE BANDED DESIGN MATRICES                           DSPLIN
C  ADDA1,ADDA2   - ADDRESS SEQUENCE OF DESIGN MATRICES                       DSPLIN
C  W1,W2         - MISCLOSURE VECTORS                                        DSPLIN
C  X             - LEAST SQUARE ESTIMATES                                    DSPLIN
C                                                                            DSPLIN
C OUTPUT PARAMTERS                                                           DSPLIN
C  V,V2     - RESIDUAL VECTORS                                               DSPLIN
C                                                                            DSPLIN
C CALLS - DMAG                                                               DSPLIN
C                         ***********                                        DSPLIN
        COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,IDV          DSPLIN
        COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,         DSPLIN
     @           IDCOF,IDVCOF                                                DSPLIN
C* ARRAYS                                                                    DSPLIN
        REAL*8 A1(IDA1,7),A2(IDA2,1),W1(IDW1,1),W2(IDW2,1),                  DSPLIN
     @           V(IDV,1),V2(IDV2,1),X(IDX,1),SUM,XLR                        DSPLIN
        INTEGER R,M,N,ADDA1(IDA1),ADDA2(IDA2)                                DSPLIN
C* FUNCTIONS                                                                 DSPLIN
        IROW(I) = (I+1)/2                                                    DSPLIN
        ICOL(I) = (2-((I+1)/2 - (I/2)))                                      DSPLIN
C* OBSERVATION RESIDUALS                                                     DSPLIN
        DO 134 I = 1,N                                                       DSPLIN
```

```
      SUM = 0.0D0                                                       DSPLIN
      DO 135 J = 1,M                                                    DSPLIN
      II = IROW(I)                                                      DSPLIN
      JJ = J - (ADDA1(II) + ICOL(I) -1 ) + 1                            DSPLIN
      IF(JJ.LT.1.OR.JJ.GT.7)GO TO 135                                   DSPLIN
      SUM = SUM + A1(II,JJ)*X(J,1)                                      DSPLIN
  135 CONTINUE                                                          DSPLIN
  134 V(I,1) = -(SUM + W1(I,1))                                         DSPLIN
C* CONSTRAINT RESIDUALS                                                 DSPLIN
      DO 136 I = 1,R                                                    DSPLIN
      SUM = 0.0D0                                                       DSPLIN
      DO 138 J = 1,M                                                    DSPLIN
      II = IROW(I)                                                      DSPLIN
      JJ = J - (ADDA2(II) + ICOL(I) -1 ) + 1                            DSPLIN
      IF(JJ.LT.1.OR.JJ.GT.11)GO TO 138                                  DSPLIN
      SUM = SUM + A2(II,JJ)*X(J,1)                                      DSPLIN
  138 CONTINUE                                                          DSPLIN
  136 V2(I,1) = -(SUM + W2(I,1))                                        DSPLIN
C* PRINTS THE LARGEST CONSTRAINT RESIDUAL                               DSPLIN
      CALL DMAG(V2,R,1,IDV2,1,XLR,2)                                    DSPLIN
      WRITE(6,9090)XLR                                                  DSPLIN
 9090 FORMAT('   LR CONST MIS : ',1PD12.5)                              DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE RTOP(X0,Y0,XT,YT,DIST,DIR)                             DSPLIN
      REAL*8 X0,Y0,XT,YT,DIR,DIST,PHI                                   DSPLIN
C ***************************************************************      DSPLIN
C * VERSION      : 26 JUNE 1982                                  *      DSPLIN
C * DESCRIPTION : TO BE USED ONLY WITH THE LOCAL LEFT            *      DSPLIN
C *               HANDED TWO DIMENSIONAL COORDINATE SYSTEM.*            DSPLIN
C *               COMPUTES AZIMUTH AND DISTANCE BETWEEN          *      DSPLIN
C *               TWO POINTS.                                    *      DSPLIN
C ***************************************************************      DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  X0,Y0  - EASTING AND NORTHING OF FROM STATION                        DSPLIN
C  XT,YT  - EASTING AND NORTHING OF TO STATION                          DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  DIR    - AZIMUTH IN RADIANS(RANGE - 0 TO 2*PHI)                      DSPLIN
C  DIST   - DISTANCE IN UNITS OF COORDINATES                            DSPLIN
C                                                                       DSPLIN
C CALLS   -DATAN2,DSQRT                                                 DSPLIN
C                                                                       DSPLIN
C EXAMPLE   (X0,Y0)-----DIST,DIRECTION------>>>>>(XT,YT)                DSPLIN
C                                                                       DSPLIN
      DATA PHI/3.141592654D0/                                           DSPLIN
      DIR = DATAN2(XT-X0,YT-Y0)                                         DSPLIN
      IF(DIR.LT.0.0D0)DIR = DIR + PHI*2.0D0                             DSPLIN
```

```
      DIST = DSQRT((X0-XT)*(X0-XT) + (Y0-YT)*(Y0-YT))                   DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SATELV(X0,Y0,XP,YP,DRS,ELEV)                           DSPLIN
C *************************************************************         DSPLIN
C * VERSION      : 27 JUNE 1982                                *        DSPLIN
C * DESCRIPTION : COMPUTES THE ELEVATION OF THE TRANSIT        *        DSPLIN
C *               SATELLITE AT POINT OF CLOSEST APPROACH.       *        DSPLIN
C *************************************************************         DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  X0,Y0   - POSITION OF OBSERVING STATION                              DSPLIN
C  XP,YP   - POSITION OF POLE IN LOCAL COORDINATE SYSTEM                DSPLIN
C  DRS     - DIRECTION OF VELOCITY VECTOR AT CLOSEST APPROACH           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  ELEV    - ELEVATION OF SATELLITE AT POINT OF CLOSEST APPROACH        DSPLIN
C                                                                       DSPLIN
C CALLS    - RTOP                                                       DSPLIN
C          - DATAN2,DCOS,DSIN                                           DSPLIN
C                                                                       DSPLIN
C REFERENCE : THESIS - S.H. QUEK , U.N.B                                DSPLIN
C                                                                       DSPLIN
C PROGRAM BEGINS                                                        DSPLIN
      REAL*8 X0,Y0,XP,YP,DRS,DRO,BETA,ALPHA,D,Z,X,Y,ELEV,NINTY,PHI,     DSPLIN
     @       RE,C                                                       DSPLIN
C  RADIUS OF THE EARTH AT THE VICINITY OF THE OBSERVER(POLE-WGS 72 ELL) DSPLIN
      DATA RE/6356750.5200/                                             DSPLIN
C  RATIO : RADIUS OF THE EARTH / RADIUS OF SATELLITE ORBIT              DSPLIN
      DATA C/0.87456000/                                                DSPLIN
C OTHER                                                                 DSPLIN
      DATA PHI/3.14159265400/,NINTY/1.57079632700/                      DSPLIN
C                                                                       DSPLIN
      CALL RTOP(XP,YP,X0,Y0,D,DRO)                                      DSPLIN
      DRSS = DRS                                                        DSPLIN
      IF(DRS.LT.DRO)DRSS = DRS + PHI*2.000                              DSPLIN
      BETA = DRSS - DRO                                                 DSPLIN
      IF(BETA.GE.PHI)BETA   = BETA - PHI                                DSPLIN
      IF(BETA.GT.NINTY)BETA = PHI - BETA                                DSPLIN
      Z = D*DSIN(BETA)                                                  DSPLIN
      ALPHA = DATAN2(Z,RE)                                              DSPLIN
      X = DSIN(ALPHA)                                                   DSPLIN
      Y = DCOS(ALPHA) - C                                               DSPLIN
      ELEV = DATAN2(Y,X)                                                DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SELKNT(TKRD,KNO,NO,ICHECK,TX,TK,LOBS,VAR,ICKNOT,       DSPLIN
     @              IRJET,ALPKNT,DTK,DD,                                DSPLIN
     @              FWK1,FWK2,FWK3,FWK4,FWK5,FWK6,FWK7)                 DSPLIN
```

```
      IMPLICIT REAL*8(A-H,O-Y)                                          DSPLIN
C ******************************************************************   DSPLIN
C * VERSION          : 17 JUNE 1982                                  *   DSPLIN
C * DESCRIPTION      : ENFORCES THE CHOSEN KNOT SELECTION SCHEME.    *   DSPLIN
C *                    CHECKS AND PRINTS THE DATA VS KNOT PLACEMENTS, *  DSPLIN
C *                    AND GENERATION OF KNOT INTERVAL               *   DSPLIN
C ******************************************************************   DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  TKRD    - TYPE OF KNOT SELECTION                                     DSPLIN
C             0  EQUAL INTERVAL IN TIME BETWEEN KNOTS OF DATA SPAN      DSPLIN
C             1  KNOTS TIMES ARE READ IN EARLIER                        DSPLIN
C             2  CALLING THE TREND EVALUATION KNOT FITTING ALTGM        DSPLIN
C             3  KNOT TIMES WITH EQUAL DATA PTS (KNC VARIABLE)          DSPLIN
C  KNC,NO - NO OF KNOTS AND DATA POINTS                                 DSPLIN
C  ICHECK - OPTION CODE ON CHECK ROUTINE                                DSPLIN
C  TX,TK,LOBS,VAR                                                       DSPLIN
C         - SEE MAIN PROGRAM                                            DSPLIN
C  ICKNOT,IRJET,ALPKNT                                                  DSPLIN
C         - SEE SKNOT ROUTINE                                           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMTERS                                                      DSPLIN
C  KNC,TK - NO. OF KNOTS AND KNOT TIMES(REVISED OR GENERATED)           DSPLIN
C  DTK    - VECTOR OF KNOT INTERVALS                                    DSPLIN
C                                                                       DSPLIN
C WORK ARRAYS/VARIABLES                                                 DSPLIN
C  FWK1,FWK2,FWK3,FWK4,FWK5,FWK6,FWK7  (AS REQUESTED BY SKNOT ROUTINE)  DSPLIN
C                                                                       DSPLIN
C CALLS    - SKNOT, CHECK                                               DSPLIN
C                          *******                                      DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV      DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,      DSPLIN
     @       IDCOF,IDVCOF                                               DSPLIN
      REAL*8 TX(IDTX,1), LOBS(IDLOBS,1),VAR(IDVAR,2,2),                 DSPLIN
     @       TK(IDTK,1),DTK(IDDTK,1),DD                                 DSPLIN
C* WORK ARRAYS DIMENSIONED AS IN FIT ROUTINE :CALLED IN SKNOT ROUTINE    DSPLIN
      REAL*8 FWK1(102,7),FWK2(204,3),FWK3(100,2,2),FWK4(8,8),FWK5(8,8), DSPLIN
     @       FWK6(200,1),FWK7(8,204)                                    DSPLIN
      INTEGER*4 TKRD,NO,KNC,ICHECK,ICKNOT(10)                           DSPLIN
C*                                                                      DSPLIN
C* SELECTING THE VARIOUS KNOT TIME SCHEMES                              DSPLIN
      IF(TKRD.EQ.1)GO TO 20                                             DSPLIN
      IF(TKRD.EQ.2)GO TO 10                                             DSPLIN
      IF(TKRD.EQ.3)GO TO 50                                             DSPLIN
C* GENERATE KNOT TIMES WITH EQUAL TIME INTERVALS                        DSPLIN
C* DESCRIPTION : TAKING THE FIRST AND LAST KNOT TIME AND DIVIDING IT    DSPLIN
C*               BY THE NO OF KNOTS MINUS ONE                           DSPLIN
      L = NO + 2                                                        DSPLIN
      DD = (TX(L,1)-TX(1,1))/(KNO-1)                                    DSPLIN
```

```
      DO 30 I = 1,KNO                                                   DSPLIN
   30 TK(I,1) = TX(1,1) + DD*(I-1)                                      DSPLIN
      GO TO 20                                                          DSPLIN
C* GENERATE KNOT TIMES WITH EQUAL NUMBER OF POINTS PER INTERVAL         DSPLIN
C* KNOTS TIME ARE INBETWEEN DATA TIMES.                                 DSPLIN
   50 INOPT = KNO                                                       DSPLIN
      L = 2                                                             DSPLIN
      I1 = NO + 1                                                       DSPLIN
      KNO = NO + 2                                                      DSPLIN
      I = INOPT + 1                                                     DSPLIN
      II = I + 1                                                        DSPLIN
      TK(1,1) = TX(1,1)                                                 DSPLIN
   60 TK(L,1) = (TX(I,1) + TX(II,1))/2.0D0                              DSPLIN
      I = I + INOPT                                                     DSPLIN
      II = I + 1                                                        DSPLIN
      L = L + 1                                                         DSPLIN
      IF(I.LT.I1)GO TO 60                                               DSPLIN
      TK(L,1) = TX(KNO,1)                                               DSPLIN
      KNO = L                                                           DSPLIN
      GO TO 20                                                          DSPLIN
C* USE THE GENERATING KNOT ROUTINE INSTEAD                              DSPLIN
   10 CALL SKNOT(KNO,NO,TX,TK,LOBS,VAR,ICKNOT,IRJET,ALPKNT,             DSPLIN
     @           FWK1,FWK2,FWK3,FWK4,FWK5,FWK6,FWK7)                    DSPLIN
C* PRINTS AND CHECKS THE KNOT VS DATA POINT SEQUENCE                    DSPLIN
   20 CALL CHECK(KNO,NO,TX,TK,ICHECK)                                   DSPLIN
C* DESCRIPTION : WHAT THE ROUTINE CHECK DOES IS DESCRIBED IN THE        DSPLIN
C*              ROUTINE PROGRAM LISTING.                                DSPLIN
C* GENERATION OF KNOT INTERVALS                                         DSPLIN
      DO 40 I = 2,KNO                                                   DSPLIN
      I1 = I - 1                                                        DSPLIN
   40 DTK(I1,1) = TK(I,1) - TK(I1,1)                                    DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SIGDIS(KNO,NO,TIM,SDIST,TK,COF,VCOF,LOBS,IDIM)         DSPLIN
      IMPLICIT REAL*8(A-H,O-Y)                                          DSPLIN
C ****************************************************************     DSPLIN
C * VERSION       : 11 JUNE 1982                                   *    DSPLIN
C * DESCRIPTION   : EVALUATES SIGNED DISTANCES BETWEEN SPLINED     *    DSPLIN
C *                 POINT AND RAW POINT                            *    DSPLIN
C ****************************************************************     DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  KNO,NO - NO OF KNOT AND DATA POINTS                                  DSPLIN
C  TK     - VECTOR OF KNOT TIMES                                        DSPLIN
C  COF    - COEFFICIENT MATRIX OF SET OF KNOTS                          DSPLIN
C  VCOF   - COVARIANCE MATRICES OF COEFFICIENTS                         DSPLIN
C  LOBS   - VECTOR OF OBSERVABLES                                       DSPLIN
C  IDIM   - DIMENSION OF ARRAYS TIM AND SDIST                           DSPLIN
C                                                                       DSPLIN
```

```
C OUTPUT PARAMETERS                                                          DSPLIN
C  SDIST  - SIGNED DISTANCES BETWEEN RAW AND SMOOTHED DATA POINTS            DSPLIN
C                                                                            DSPLIN
C CALLS    - POINT                                                           DSPLIN
C          - DSQRT,DATAN2                                                    DSPLIN
C                     ***********************                                DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV           DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,           DSPLIN
     @       IDCOF,IDVCOF                                                    DSPLIN
      REAL*8 TIME,TIM(IDIM,1),SDIST(IDIM,1),TK(IDTK,1),                      DSPLIN
     @       COF(IDCOF,8),VCOF(IDVCOF,8,8),LOBS(IDLOBS,1)                    DSPLIN
      INTEGER KNO,NO,IX                                                      DSPLIN
      DATA PI/3.14159265358979300/                                           DSPLIN
C*                                                                           DSPLIN
      PIX2  = PI*2.0D0                                                       DSPLIN
      DO 10 I = 1,NO                                                         DSPLIN
      IX = (I+2)*2 - 1                                                       DSPLIN
      OX = LOBS(IX,1)                                                        DSPLIN
      IX = (I+2)*2                                                           DSPLIN
      OY = LOBS(IX,1)                                                        DSPLIN
      CALL POINT(TIM(I,1),X,Y,VX,VY,VXY,ICO,KNO,TK,COF,VCOF)                 DSPLIN
      SDIST(I,1) = DSQRT((OX-X)*(OX-X) + (OY-Y)*(OY-Y))                      DSPLIN
      I2 = I + 1                                                             DSPLIN
      IF(I2.GT.NO)I2 = I - 1                                                 DSPLIN
      TIME = TIM(I,1) + (TIM(I2,1) - TIM(I,1))/2.0D0                         DSPLIN
      CALL POINT(TIME,X2,Y2,VX,VY,VXY,ICO,KNC,TK,COF,VCOF)                   DSPLIN
      DIR1 = DATAN2(X2-X,Y2-Y)                                               DSPLIN
      IF(DIR1.LT.0.0D0)DIR1 = DIR1 + PIX2                                    DSPLIN
      IF(I2.LT.I)DIR1 = DIR1 + PI                                            DSPLIN
      IF(DIR1.GT.PIX2)DIR1 = DIR1 - PIX2                                     DSPLIN
      DIR2 = DATAN2(OX-X,OY-Y)                                               DSPLIN
      IF(DIR2.LT.0.0D0)DIR2 = DIR2 + PIX2                                    DSPLIN
      CORR = PIX2 - DIR1                                                     DSPLIN
      ANGLE = DIR2 + CORR                                                    DSPLIN
      IF(ANGLE.GT.PIX2)ANGLE = ANGLE - PIX2                                  DSPLIN
      IF(ANGLE.GT.PI)SDIST(I,1)= -SDIST(I,1)                                 DSPLIN
   10 CONTINUE                                                               DSPLIN
      RETURN                                                                 DSPLIN
      END                                                                    DSPLIN
      SUBROUTINE SIMULA(TIME,GX,GY)                                          DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                               DSPLIN
C ****************************************************************          DSPLIN
C * VERSION      : 24TH MARCH 1982                                 *         DSPLIN
C * DESCRIPTION : GENERATOR FOR THE SIMULATED DATA SERIES          *         DSPLIN
C ****************************************************************          DSPLIN
C  INPUT PARAMETERS                                                          DSPLIN
C       TIME - TIME AT WHICH SIMULATED POSITION IS REQUIRED                  DSPLIN
C                                                                            DSPLIN
```

```
C  OUTPUT PARAMETERS                                                    DSPLIN
C        X     - POSITION IN THE X DIRECTION                            DSPLIN
C        Y     - POSITION IN THE Y DIRECTION                            DSPLIN
C                                                                       DSPLIN
C CALLS        - DSIN                                                   DSPLIN
C                                                                       DSPLIN
C* ROUTINE BEGINS                                                       DSPLIN
C* OMEGA OF X AND Y OF GENERATOR                                        DSPLIN
      DATA PIE/3.14159265358979300/                                     DSPLIN
      WX = 2.0D00*PIE/440.0D00                                          DSPLIN
      WY = 2.0D00*PIE/70.0D00                                           DSPLIN
C* COMPUTING SIMULATED VALUES OF  X AND Y AT TIME TFIX                  DSPLIN
      GX = 1400.0D00*DSIN(WX*(TIME)+ PIE/90.0D00)                       DSPLIN
      GY = 250.0D00*DSIN(WY*(TIME)+4.0*PIE/18.0D00)+3.0*TIME            DSPLIN
     @ + 250                                                            DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SKNOT(KNO,NO,TX,TK,LOBS,VAR,IOPTON,IRJET,ALPKNT,       DSPLIN
     &                WK1,WK2,WK3,WK4,WK5,WK6,WKS)                      DSPLIN
      IMPLICIT REAL*8 (A-H,O-Z)                                         DSPLIN
C *************************************************************         DSPLIN
C * VERSION      : 17 JUNE 1982                                *        DSPLIN
C * DESCRIPTION : DOES THE KNOT SELECTION RUN USING INFORMATION *       DSPLIN
C *               STORED IN THE ARRAYS.                         *       DSPLIN
C *************************************************************         DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  NO,TX,LOBS,VAR                                                       DSPLIN
C          - SEE MAIN PROGRAM FOR DESCRIPTION                           DSPLIN
C  IOPTON  - OPTION SET FOR FIT RCUTINE (SEE FIT ROUTINE)               DSPLIN
C  IRJET,ALPKNT                                                         DSPLIN
C          - REJECT LEVELS FCR REJECT ROUTINE (SEE REJECT ROUTINE)      DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  KNO,TK  - NO OF GENERATED KNOTS AND KNOT TIMES(DIM IN COMMON)        DSPLIN
C                                                                       DSPLIN
C WORK ARRAYS                                                           DSPLIN
C  WK1,WK2,WK3,WK4,WK5,WK6,WKS                                          DSPLIN
C          - AS REQUESTED BY FIT ROUTINE                                DSPLIN
C                                                                       DSPLIN
C CALLS    - FIT                                                        DSPLIN
C                               *************                           DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV      DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,      DSPLIN
     @        IDCOF,IDVCOF                                              DSPLIN
      COMMON /INIT/TKFIT,ISWT2,ISWT,INO,PTX,PLOBS,PVAR                  DSPLIN
      REAL*8 TX(IDTX,1),TK(IDTK,1),VAR(IDVAR,2,2),LOBS(IDLOBS,1),       DSPLIN
     @        A(2,2)                                                    DSPLIN
C* WORK ARRAYS DIMENSIONED AS IN FIT ROUTINE                            DSPLIN
```

```
      REAL*8 WK1(102,7),WK2(204,3),WK3(100,2,2),WK4(8,8),WK5(8,8),          DSPLIN
     @       WK6(200,1),WKS(8,204),TKFIT(2,1)                               DSPLIN
      INTEGER IOPTON(10),KNO,NO,PTX,PLOBS,PVAR                              DSPLIN
C* SET THE LEVEL OF REJECTION IN THE LAST TEST BELOW                        DSPLIN
       DATA ISET/4/                                                          DSPLIN
C* BEGIN                                                                     DSPLIN
C* WRITE UNITS                                                               DSPLIN
      IWR = 6                                                                DSPLIN
C* WRITE THE OPTIONS OF THE REJECTION                                        DSPLIN
      WRITE(6,920)IRJET,ALPKNT                                               DSPLIN
      WRITE(6,930)(IOPTON(I),I=1,10)                                         DSPLIN
C  RESET SWITCHES IN THE FIT ROUTINE                                         DSPLIN
      ISWT2 = 0                                                              DSPLIN
      ISWT  = 0                                                              DSPLIN
      INO   = 0                                                              DSPLIN
      PTX   = 0                                                              DSPLIN
      PLOBS = 0                                                              DSPLIN
      PVAR  = 0                                                              DSPLIN
      TKFIT(1,1) =0.0D0                                                      DSPLIN
      TKFIT(2,1) = 0.0D0                                                     DSPLIN
C* FEED IN THE FIRST KNOT                                                    DSPLIN
      TK(1,1) = TX(1,1)                                                      DSPLIN
      DO 10 I = 1,2                                                          DSPLIN
          DO 10 J = 1,2                                                      DSPLIN
   10 A(I,J) = VAR(1,I,J)                                                    DSPLIN
      CALL FIT(TK(1,1),LOBS(1,1),LOBS(2,1),A,IOPTON,ICODE,                   DSPLIN
     @         IRJET,ALPKNT,WK1,WK2,WK3,WK4,WK5,WK6,WKS)                     DSPLIN
C* SET POINTERS                                                              DSPLIN
      LJ = 3                                                                 DSPLIN
      LK = 2                                                                 DSPLIN
      IL = NO + 1                                                            DSPLIN
      DO 20 I = 2,IL                                                         DSPLIN
         LJ = LJ + 2                                                         DSPLIN
         LJ2 = LJ + 1                                                        DSPLIN
         TIME = TX(I,1)                                                      DSPLIN
         DO 30 J = 1,2                                                       DSPLIN
             DO 30 K = 1,2                                                   DSPLIN
   30    A(J,K) = VAR(I,J,K)                                                 DSPLIN
        CALL FIT(TIME,LOBS(LJ,1),LOBS(LJ2,1),A,IOPTON,ICODE,                 DSPLIN
     @         IRJET,ALPKNT,WK1,WK2,WK3,WK4,WK5,WK6,WKS)                     DSPLIN
        IF(ICODE.EQ.1)STOP                                                   DSPLIN
        IF(ICODE.NE.2)GO TO 20                                               DSPLIN
        TK(LK,1) = TIME                                                      DSPLIN
        LK = LK + 1                                                          DSPLIN
        IF(LK.GT.IDTK)GO TO 900                                              DSPLIN
   20   CONTINUE                                                             DSPLIN
      IL = IL + 1                                                            DSPLIN
C* LK HAS BEEN INCREMENTED ABOVE                                             DSPLIN
```

```
      TK(LK,1) = TX(IL,1)                                               DSPLIN
      KNO = LK                                                          DSPLIN
C* CHECK THE NUMBER OF POINTS IN THE LAST SECTION                       DSPLIN
      LJ = LK-1                                                         DSPLIN
      IL = NO + 1                                                       DSPLIN
      LK = 0                                                            DSPLIN
      DO 40 I = 2,IL                                                    DSPLIN
      IF(TX(I,1).GT.TK(LJ,1))LK = LK + 1                                DSPLIN
   40 CONTINUE                                                          DSPLIN
      IF(LK.GT.ISET)RETURN                                              DSPLIN
C* MODIFY THE 2ND LAST KNOT TO BE OF EQUAL INTERVAL ON BOTH SIDES       DSPLIN
      IL = KNO - 2                                                      DSPLIN
      TIME = TK(KNO,1)- TK(IL,1)                                        DSPLIN
      LJ = KNO - 1                                                      DSPLIN
      TK(LJ,1) = TK(IL,1) +  TIME/2.0D0                                 DSPLIN
      RETURN                                                            DSPLIN
  900 WRITE(6,910)IDTK                                                  DSPLIN
  910 FORMAT(/,3X,'NO. OF GENERATED KNOTS EXCEED ',I4,                  DSPLIN
     @       /,3X,'(MAXIMUM SET BY ARRAY DECLARATION)',/,/)             DSPLIN
  920 FORMAT(//,3X,'KNOT SELECTION ALGORITHM ',/,                       DSPLIN
     @       /,3X,'MINIMUM POINTS BEFORE REJECTING FIT : ',I4,          DSPLIN
     @       /,3X,'CONFIDENCE LEVEL CHOSEN             : ',F6.1,' %')   DSPLIN
  930 FORMAT(//,3X,'OPTION SET : ',I3,I3,I4,10I3)                       DSPLIN
      STOP                                                              DSPLIN
C* FUTURE MODIFICATIONS TO BE PLACED HERE                               DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SMOOTH(NO,KNO,SC,ESC,TX,TK,COF,VCOF)                   DSPLIN
      IMPLICIT REAL*8(A-H,O-Z)                                          DSPLIN
C   *************************************************************       DSPLIN
C   * VERSION : 18 APRIL 1982                                    *       DSPLIN
C   * DESCRIPTION : COMPUTE SMOOTH INPUT DATA POINTS             *       DSPLIN
C   *************************************************************       DSPLIN
C  INPUT PARAMETERS                                                     DSPLIN
C   KNO,NO   - NO OF KNOT AND DATA POINTS                               DSPLIN
C   SC       - SCALE FACTOR ON TIME DOMAIN                              DSPLIN
C   ESC      - SCALE FACTOR FOR ELLIPSE COMPUTATIONS                    DSPLIN
C   TX,TK    - VECTOR OF DATA POINT TIMES AND KNOT TIMES                DSPLIN
C   COF,VCOF- COEFFICIENTS AND COVARIANCES OF COEFFICIENTS              DSPLIN
C                                                                       DSPLIN
C  OUTPUT PARAMETERS                                                    DSPLIN
C   NONE     ( ALL OUTPUT PRINTED TO UNIT IWR -ASSIGNED BELOW)          DSPLIN
C                                                                       DSPLIN
C  CALLS     - POINT,ELIPSE,RADD                                        DSPLIN
C            - DSQRT                                                    DSPLIN
C                       ***************                                 DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,ICVAR,ICV      DSPLIN
      COMMON /DIM2/ICKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,      DSPLIN
     @       IDCOF,IDVCOF                                               DSPLIN
```

```
      REAL*8 TX(IDTX,1),TK(IDTK,1),COF(IDCOF,8),VCOF(IDVCOF,8,8)        DSPLIN
      INTEGER NO,KNO                                                    DSPLIN
C* WRITE UNIT                                                           DSPLIN
      IWR = 6                                                           DSPLIN
C* TITLE                                                                DSPLIN
      WRITE(IWR,9010)                                                   DSPLIN
      WRITE(IWR,9020)                                                   DSPLIN
C* COMPUTE THE SMOOTHED VALUES OF THE DATA POINTS                        DSPLIN
      IL = NO + 2                                                       DSPLIN
C* REMOVE MODIFICATION LATER(L = NO+1 AND BELOW IS 2,IL)                 DSPLIN
      DO 10 I = 1,IL                                                    DSPLIN
      CALL POINT(TX(I,1),X,Y,VX,VY,VXY,ICO,KNO,TK,COF,VCOF)             DSPLIN
      CALL ELIPSE(VX,VY,VXY,A,B,ESC,DIR)                                DSPLIN
      SX = DSQRT(VX)                                                    DSPLIN
      SY = DSQRT(VY)                                                    DSPLIN
      DR = RADD(DIR)                                                    DSPLIN
      II = I - 1                                                        DSPLIN
      TRU = TX(I,1)*SC                                                  DSPLIN
      WRITE(IWR,9030)II,TRU,X,Y,SX,SY,VXY,A,B,DR                        DSPLIN
   10 CONTINUE                                                          DSPLIN
 9010 FORMAT(/,3X,'SMOOTHED DATA POINTS',/)                             DSPLIN
 9020 FORMAT(//,10X,'PREDICTED FIXES AT SPECIFIED TIMES',/10X,          DSPLIN
     .17('=='),//,2X,'NO',6X,'TIME',12X,                                DSPLIN
     .'POSITION(E,N)            SIGMA POSN(E,N)',13X,' COVARIANCE',4X,  DSPLIN
     .'SEMI-MAJOR  SEMI-MINOR   ORIENT(DEG) ',//)                       DSPLIN
 9030 FORMAT(2X,I4,2X,F10.4,10(F12.3,2X))                               DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SPIN(Q,N,MM,DET,IDEXP)                                 DSPLIN
C                                                                       DSPLIN
C     SUBROUTINE SPIN IS A MATRIX INVERSION ROUTINE FOR SYMMETRIC       DSPLIN
C     POSITIVE-DEFINITE MATRICES.  THE MATRIX INVERTED IS THE UPPER     DSPLIN
C     N BY N PORTION OF THE MATRIX Q WHICH IS DIMENSIONED MM BY MM      DSPLIN
C     IN THE CALLING ROUTINE.                                           DSPLIN
C                                                                       DSPLIN
C  INPUT:                                                               DSPLIN
C        Q - THE MATRIX DIMENSIONED MM BY MM WHICH CONTAINS THE         DSPLIN
C            MATRIX TO BE INVERTED.                                     DSPLIN
C                                                                       DSPLIN
C        N - THE DIMENSION OF THE ACTUAL PART( UPPER LEFT CORNER)       DSPLIN
C            OF Q WHICH IS TO BE INVERTED. ( N MAY BE EQUAL BUT MUST    DSPLIN
C            NOT BE LARGER THAN MM) .                                   DSPLIN
C        MM- DIMENSIONED SIZE OF Q IN THE CALLING ROUTINE.              DSPLIN
C                                                                       DSPLIN
C                                                                       DSPLIN
C  OUTPUT:                                                              DSPLIN
C                                                                       DSPLIN
C        Q - THE UPPER LEFT N BY N PORTION CONTAINS THE INVERSE OF      DSPLIN
```

```
C          THE INPUT UPPER LEFT N BY N PORTION.                            DSPLIN
C                                                                          DSPLIN
C      DET - THE NON-EXPONENT PORTION OF THE DETERMINANT OF THE            DSPLIN
C            INPUT N BY N (UPPER LEFT PORTION OF Q) MATRIX. SEE            DSPLIN
C            IDEXP  BELOW.                                                 DSPLIN
C                                                                          DSPLIN
C      IDEXP - THE EXPONENT (OF 10) PART OF THE DETERMINANT DESCRIBED      DSPLIN
C              UNDER DET ABOVE. THUS THE DETERMINANT IS RETURNED IN        DSPLIN
C              TWO PARTS CORRESPONDING TO                                  DSPLIN
C                    DETERMINANT = DET * 10 ** IDEXP .                     DSPLIN
C              THIS IS DONE TO AVOID UNDER OR OVERFLOW IN THE              DSPLIN
C              COMPUTATION OF THE DETERMINANT.  TO PRINT THE DETERM-       DSPLIN
C              INANT THE USER SHOULD PRINT BOTH NUMBERS AS FOLLOWS;        DSPLIN
C              (FOR EXAMPLE)                                               DSPLIN
C                 PRINT 10,DET,IDEXP                                       DSPLIN
C              10 FORMAT(' ','DETERMINANT= ',F7.4,'D',I4)                  DSPLIN
C                                                                          DSPLIN
C                                                R.R. STEEVES              DSPLIN
C                                                SEPT., 1979               DSPLIN
C                                                                          DSPLIN
C                                                                          DSPLIN
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*4 (I-N)                           DSPLIN
      DIMENSION Q(MM,MM)                                                   DSPLIN
      DET=0.D0                                                             DSPLIN
      DO 4 J=1,N                                                           DSPLIN
      DO 4 I=1,J                                                           DSPLIN
      IF(I.EQ.1) GO TO 2                                                   DSPLIN
      M=I-1                                                                DSPLIN
      SUM=0.0D0                                                            DSPLIN
      DO 1 K=1,M                                                           DSPLIN
    1 SUM=SUM+Q(K,I)*Q(K,J)                                                DSPLIN
      Q(I,J)=Q(I,J)-SUM                                                    DSPLIN
    2 IF(I.EQ.J) GO TO 3                                                   DSPLIN
      Q(I,J)=Q(I,J)/Q(I,I)                                                 DSPLIN
      GO TO 4                                                              DSPLIN
    3 CONTINUE                                                             DSPLIN
      DET=DET+DLOG10(Q(I,I))                                               DSPLIN
      Q(I,I)=DSQRT(Q(I,I))                                                 DSPLIN
    4 CONTINUE                                                             DSPLIN
      IDEXP=DET                                                            DSPLIN
      RPART=DET-IDEXP                                                      DSPLIN
      APART=DABS(RPART)                                                    DSPLIN
      IF(APART.LT.1.D-20) DET=1.D0                                         DSPLIN
      IF(APART.LT.1.D-20) GO TO 10                                         DSPLIN
      DET=10**RPART                                                        DSPLIN
   10 CONTINUE                                                             DSPLIN
      DO 7 J=1,N                                                           DSPLIN
      DO 7 I=1,J                                                           DSPLIN
```

```
      IF(I.LT.J) GO TO 5                                                DSPLIN
      Q(J,J)=1.0D0/Q(J,J)                                               DSPLIN
      GO TO 7                                                           DSPLIN
    5 SUM=0.0D0                                                         DSPLIN
      M=J-1                                                             DSPLIN
      DO 6 K=I,M                                                        DSPLIN
    6 SUM=SUM-Q(I,K)*Q(K,J)                                             DSPLIN
      Q(I,J)=SUM/Q(J,J)                                                 DSPLIN
    7 CONTINUE                                                          DSPLIN
      DO 9 J=1,N                                                        DSPLIN
      DO 9 I=1,J                                                        DSPLIN
      SUM=0.0D0                                                         DSPLIN
      DO 8 K=J,N                                                        DSPLIN
    8 SUM=SUM+Q(I,K)*Q(J,K)                                             DSPLIN
      Q(I,J)=SUM                                                        DSPLIN
      IF(I.EQ.J) GO TO 9                                                DSPLIN
      Q(J,I)=SUM                                                        DSPLIN
    9 CONTINUE                                                          DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SOLV(A,P,W,X,VX,V,N,M,IDEG,SO,                         DSPLIN
     @               IA,JA,IP,JP,IW,IX,IVX,IV,ICODE,PCODE,ATP)          DSPLIN
      IMPLICIT REAL*8(A-H,P-Z)                                          DSPLIN
C ***************************************************************       DSPLIN
C * VERSION     : 12TH APRIL 1982                                 *     DSPLIN
C * DESCRIPTION : SOLUTION X = (ATWA)-1 ATWB - WITH SPECIAL       *     DSPLIN
C *               BANDED DESIGN(A) MATRIX                         *     DSPLIN
C ***************************************************************       DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  A      - DESIGN MATRIX                                               DSPLIN
C  P      - FIXED BANDWIDTH WEIGHT MATRIX FOR OBSERVABLES               DSPLIN
C  N,M    - NO OF OBS EQN AND UNKNOWNS                                  DSPLIN
C  IA,JA,IP,JP,IW,IX,IVX,IV                                             DSPLIN
C         - DECLARED DIMENSIONS OF RESP MATIRCES (SEE BELOW)            DSPLIN
C  PCODE - PRINTING CODES                                               DSPLIN
C                0 - NO DIAGNOSTIC EXCEPT FOR FATAL ERRORS              DSPLIN
C                1 - ALL PERTINENT INFORMATION PER PROCESSING           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  X,VX     - LS ESTIMATES AND COVARIANCES                              DSPLIN
C  IDEG,SO- DEG OF FREEDOM AND VARIANCE FACTOR                          DSPLIN
C  ICODE   - 0   SOLUTION PASSED ALL TESTS                              DSPLIN
C          - 1   INVERSE FAILURE OF DESIGN MATRIX                       DSPLIN
C          - 2   SEVERE LOST OF COMPUTING PRECISION                     DSPLIN
C          - 3   CONDITION NUMBER TEST FAILED                           DSPLIN
C                                                                       DSPLIN
C WORK ARRAY                                                            DSPLIN
C   ATP                                                                 DSPLIN
```

```
      C                                                                 DSPLIN
      C ADDITIONAL INFORMATION                                          DSPLIN
      C  XLEVEL - LARGEST VALUE OF INTERNAL COMPUTATIONAL ACCURACY ACCEPTABLE DSPLIN
      C           BEFORE REJECTION OF COMPUTATIONS WITH ICODE = 2       DSPLIN
      C  XLCON  - LARGEST VALUE OF CONDITION NUMBER ACCEPTED            DSPLIN
      C           IF EXCEED, ICODE = 3                                  DSPLIN
      C 7 JULY  - UPGRADE TO TAKE 204 SIZE MATRICES                     DSPLIN
      C                                                                 DSPLIN
      C CALLS   - COND,CHOLD,RESID,DMAG,BMULT1,SCMULT                   DSPLIN
      C                     ####################                        DSPLIN
      C                                                                 DSPLIN
            REAL*8 A(IA,JA),P(IP,JP),W(IW,1),X(IX,1),VX(IVX,IVX),       DSPLIN
           @   ATP(8,204),ATPW(8,1),DET,VTV,S0,V(IV,1),PV(204,1),ATPV(8,1) DSPLIN
            INTEGER PCODE                                               DSPLIN
            EQUIVALENCE (ATPV(1),ATPW(1),PV(1))                         DSPLIN
            DATA XLEVEL/1.0D-5/,XLCON/1.0D+8/                           DSPLIN
      C* FUNCTION                                                       DSPLIN
            IROW(I) = (I+1)/2                                           DSPLIN
            ICOL(I) = (2 - (((I+1)/2) - (I/2)))                         DSPLIN
      C* NUMBER OF CODIAGONALS                                          DSPLIN
            ICO     = (JP-1)/2                                          DSPLIN
      C* ERROR CODE SET                                                 DSPLIN
            ICODE  = 0                                                  DSPLIN
      C* DIMENSION ALLOCATION                                           DSPLIN
            IDATP  = 8                                                  DSPLIN
            JDATP  = 56                                                 DSPLIN
            IDATPV = 8                                                  DSPLIN
            IDPV   = 56                                                 DSPLIN
            IDATPW = 8                                                  DSPLIN
      C* FORM NORMALS                                                   DSPLIN
            DO 10 I = 1,M                                               DSPLIN
               DO 10 J = 1,N                                            DSPLIN
                   ATP(I,J) = 0.0D0                                     DSPLIN
                   IB = J - ICO                                         DSPLIN
                   IE = J + ICO                                         DSPLIN
                   IF(IB.LT.1)IB = 1                                    DSPLIN
                   IF(IE.GT.N)IE = N                                    DSPLIN
                 DO 10 K = IB,IE                                        DSPLIN
                       JAC = J - K + ICO + 1                            DSPLIN
                       IJ = I - ICOL(K) + 1                             DSPLIN
                       IF(IJ.LT.1.OR.IJ.GT.JA)GO TO 10                  DSPLIN
                       IK = IROW(K)                                     DSPLIN
            ATP(I,J) = ATP(I,J) + A(IK,IJ)*P(K,JAC)                     DSPLIN
         10 CONTINUE                                                    DSPLIN
            DO 20 I = 1,M                                               DSPLIN
               DO 30 J = 1,M                                            DSPLIN
                   VX(I,J) = 0.0D0                                      DSPLIN
                   IF(J.GT.I)GO TO 20                                   DSPLIN
```

```
                DO 40 K = 1,N                                          DSPLIN
                     JI = J - ICOL(K) + 1                              DSPLIN
                     IF(JI.LT.1.OR.JI.GT.JA)GO TO 40                   DSPLIN
                     IK = IROW(K)                                      DSPLIN
                     VX(I,J) = VX(I,J) + ATP(I,K)*A(IK,JI)             DSPLIN
   40     CONTINUE                                                     DSPLIN
          IF(I.NE.J)VX(J,I) = VX(I,J)                                  DSPLIN
   30     CONTINUE                                                     DSPLIN
   20     CONTINUE                                                     DSPLIN
          DO 50 I = 1,M                                                DSPLIN
             ATPW(I,1) = 0.0D0                                         DSPLIN
            DO 50 J = 1,N                                              DSPLIN
   50     ATPW(I,1) = ATPW(I,1) + ATP(I,J)*W(J,1)                      DSPLIN
          IPRINT = 4                                                   DSPLIN
          CALL COND(VX,M,IVX,XCOND,1)                                  DSPLIN
          CALL CHOLD(VX,M,IVX,M,DET,IER)                               DSPLIN
          IF(PCODE.EQ.1)IPRINT = 2                                     DSPLIN
          CALL COND(VX,M,IVX,XCOND,IPRINT)                             DSPLIN
          IF(XCOND.GT.XLCON)GO TO 800                                  DSPLIN
          IF(IER.EQ.0)GO TO 70                                         DSPLIN
          WRITE(6,999)IER                                              DSPLIN
          ICODE = 1                                                    DSPLIN
          RETURN                                                       DSPLIN
   70     CALL DMULT(VX,M,M,ATPW,M,1,X,IVX,IVX,IDATPW,1,IX,1)          DSPLIN
          DO 60 I = 1,M                                                DSPLIN
   60     X(I,1) = -X(I,1)                                             DSPLIN
          CALL RESID(A,N,M,X,W,V,IA,JA,IX,IW,IV)                       DSPLIN
C* COMPUTATION CHECK                                                   DSPLIN
          CALL DMULT(ATP,M,N,V,N,1,ATPV,IDATP,JDATP,IV,1,IDATPV,1)     DSPLIN
          CALL DMAG(ATPV,M,1,IDATPV,1,XLR,2)                           DSPLIN
          IF(PCODE.EQ.1)WRITE(6,980)XLR                                DSPLIN
          IF(XLR.GT.XLEVEL)GO TO 700                                   DSPLIN
          IF(N.EQ.M)RETURN                                             DSPLIN
          IDEG = N-M                                                   DSPLIN
C* COMPUTE QUADRATIC NORM AND CORRECT COVARIANCE BY VARIANCE FACTOR    DSPLIN
          CALL BMULT1(P,N,N,JP,V,N,1,PV,IP,IV,1,IDPV,1)                DSPLIN
          VTV = 0.0D0                                                  DSPLIN
          DO 80 I = 1,N                                                DSPLIN
   80     VTV = VTV + V(I,1)*PV(I,1)                                   DSPLIN
C* VARIANCE FACTOR                                                     DSPLIN
          S0 = VTV/IDEG                                                DSPLIN
          IF(PCODE.EQ.1)WRITE(6,990)IDEG,VTV,S0                        DSPLIN
          CALL SCMULT(S0,VX,VX,M,M,IVX,IVX)                            DSPLIN
          RETURN                                                       DSPLIN
  700     WRITE(6,710)                                                 DSPLIN
  710     FORMAT(/,3X,'**FATAL** INT COMP TEST : FAILED',/)            DSPLIN
          ICODE = 2                                                    DSPLIN
          RETURN                                                       DSPLIN
```

```
  800 WRITE(6,820)                                                      DSPLIN
  820 FORMAT(/,3X,'**FATAL** CONDITION NUMBER TOO LARGE ',/)            DSPLIN
      ICODE = 3                                                         DSPLIN
      RETURN                                                            DSPLIN
 999  FORMAT(3X,'CHOLD FAILURE ',I4,' CALL FROM SOLV')                  DSPLIN
 990  FORMAT(3X,'DEGREES OF FREEDOM          ',I4,                      DSPLIN
     @ /,     3X,'EUCLIDEAN NORM OF RESD    ',1PD12.5,                  DSPLIN
     @ /,     3X,'VARIANCE FACTOR           ',1PD12.5)                  DSPLIN
 980  FORMAT(3X,'INT COMP CHECK ; SOLV      ',1PD12.5)                  DSPLIN
      END                                                               DSPLIN
      SUBROUTINE TIMSC(KNO,NO,ISCT,SCTIME,TX,TK,IDTX,IDTK,TKRD)         DSPLIN
C ******************************************************************** DSPLIN
C * VERSION     : 9 JUNE 1981                                         * DSPLIN
C * DESCRIPTION : DOES THE SCALING OF THE DATA AND KNOT TIMES         * DSPLIN
C ******************************************************************** DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  KNO,NO      - NO OF KNOTS AND DATA POINTS                            DSPLIN
C  ISCT        - CODE FOR USING A SCALE FACTOR ON THE TIME DOMAIN       DSPLIN
C  SCTIME      - SCALE FACTOR ON THE TIME DOMAIN                        DSPLIN
C  TX(IDTX,1)- VECTOR OF DATA TIMES                                     DSPLIN
C  TK(IDTK,1)- VECTOR OF KNOT TIMES                                     DSPLIN
C  TKRD        - CODE ON SELKNT ROUTINE  (SEE SELKNT ROUTINE)           DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  TK(IDTK,1)- SCALED VECTOR OF KNOT TIMES                              DSPLIN
C  TX(IDTX,1)- SCALED VECTOR OF DATA POINT TIMES                        DSPLIN
C                                                                       DSPLIN
C ADDITIONAL INFORMATION                                                DSPLIN
C  SCTIME IS THE SCALING FACTOR FOR THE TIME DOMAIN.                    DSPLIN
C  THE SCALING IS DONE REPEATEDLY AS THE NUMBER OF KNOTS                DSPLIN
C  IS CHANGED UNDER EACH DIFFERENT CASE. THE SCALE FACTOR               DSPLIN
C  IS REMOVED AUTOMATICALLY AFTER EACH RUN.                             DSPLIN
C                                                                       DSPLIN
C CALLS        - SCMULT                                                 DSPLIN
C                      ************                                     DSPLIN
C ARRAYS                                                                DSPLIN
      REAL*8 TX(IDTX,1),TK(IDTK,1),SCTIME                               DSPLIN
      INTEGER  KNO,NO,TKRD,ISCT                                         DSPLIN
C* WRITE UNIT                                                           DSPLIN
      IWR = 6                                                           DSPLIN
      L = NO + 2                                                        DSPLIN
C* SKIP IF SCTIME NOT READ                                              DSPLIN
      IF(ISCT.EQ.0)RETURN                                               DSPLIN
      CALL SCMULT(SCTIME,TX,TX,L,1,IDTX,1)                              DSPLIN
      IF(TKRD.EQ.1)CALL SCMULT(SCTIME,TK,TK,KNO,1,IDTK,1)               DSPLIN
      WRITE(IWR,9000)SCTIME                                             DSPLIN
 9000 FORMAT(//,3X,'NOTE : A SCALE FACTOR OF ',1PD12.5,                 DSPLIN
     @            ' ON THE TIME DOMAIN IS IN EFFECT',/)                 DSPLIN
```

```
      RETURN
      END                                                               DSPLIN
      SUBROUTINE UPSTD(SX,SY,SXY,ICODE)                                 DSPLIN
C **********************************************************             DSPLIN
C * VERSION      : SEPT 1982                                     *      DSPLIN
C * DESCRIPTION : MINIMUM ALLOWABLE MINOR AXIS TO BE             *      DSPLIN
C *                ENFORCED.                                      *      DSPLIN
C **********************************************************             DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  SX,SY,SXY   - STD DEVIATIONS AND COVARIANCE                          DSPLIN
C  ICODE       - PRINT CODES                                            DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  SX,SY,SXY   - SCALED STD DEVIATIONS AND COVARIANCE                   DSPLIN
C                                                                       DSPLIN
C FURTHER INFORMATION                                                   DSPLIN
C  MINB        - SET MINIMUM ALLOWABLE SEMI-MINOR AXIS  OF THE ERROR    DSPLIN
C                ELLIPSES AND IF IT FALLS BELOW THIS                    DSPLIN
C                SET FIGURE. THE SEMI-MINOR AXIS IS SCALED TO           DSPLIN
C                THIS VALUE WITH THE MAJOR AXIS BEING SCALED ACCORDINGLY.DSPLIN
C  SETMIN      - MINIMUM VALUE ALLOWED OF THE SEMI MINOR AXIS. IF IT    DSPLIN
C                FALLS BELOW THIS NUMBER, THE AXIS IS REPLACED BY SETMIN.DSPLIN
C                                                                       DSPLIN
C CALLS        - ELIPSE,ELLSIG                                          DSPLIN
C                      *****************                                DSPLIN
      REAL*8 SX,SY,SXY,MINB,SETMIN,SCB,A,B,DR,VX,VY                     DSPLIN
      DATA  MINB/40.0D0/,SETMIN/1.0D0/                                  DSPLIN
        VX = SX*SX                                                      DSPLIN
        VY = SY*SY                                                      DSPLIN
        CALL ELIPSE(VX,VY,SXY,A,B,1.0D0,DR)                             DSPLIN
        IF(ICODE.EQ.1)WRITE(6,9000)SY,SX,SXY,A,B                        DSPLIN
        IF(B.LT.SETMIN)B = SETMIN                                       DSPLIN
        IF(B.EQ.SETMIN)CALL ELLSIG(A,B,DR,SX,SY,SXY,1)                  DSPLIN
        IF(B.GT.MINB)RETURN                                             DSPLIN
        SCB = MINB/B                                                    DSPLIN
        B = MINB                                                        DSPLIN
        A = SCB*A                                                       DSPLIN
        SX = SCB*SX                                                     DSPLIN
        SY = SCB*SY                                                     DSPLIN
        SXY = SCB*SCB*SXY                                               DSPLIN
        IF(ICODE.EQ.1)WRITE(6,9010)MINB,SY,SX,SXY,A,B,SCB               DSPLIN
 9000   FORMAT(/,3X,'INPUT ERROR ELLIPSE (SN,SE,SEN,A AND B) : ',       DSPLIN
     @             6(F9.1,2X),' M')                                     DSPLIN
 9010   FORMAT(  3X,'ACCEPTED (MIN: ',F5.1,')- MOD BY UPSTD    : ',     DSPLIN
     @             5(F9.1,2X),' M     ( X ',F6.1,' )')                  DSPLIN
        RETURN                                                          DSPLIN
        END                                                             DSPLIN
      SUBROUTINE VARFAC(S0,DEGOF,NO,R,M,N,ALPHA,TX,P,V,PV,VTWV,DX1,DX2) DSPLIN
```

```
C ***************************************************************************       DSPLIN
C * VERSION        : 21 JUNE 1982                                           *       DSPLIN
C * DESCRIPTION    : COMPUTES THE DEGREES OF FREEDOM AND THE                *       DSPLIN
C *                  APSOTERIORI VARIANCE FACTOR.                           *       DSPLIN
C *                  PERFORMS THE CHI SQUARED STATISTICAL TEST ON           *       DSPLIN
C *                  THE VARIANCE FACTOR AT THE ALPHA PERCENT               *       DSPLIN
C *                  CONFIDENCE INTERVAL.                                   *       DSPLIN
C ***************************************************************************       DSPLIN
C INFUT PARAMETERS                                                                  DSPLIN
C  NO,R,M,N    - AS DEFINED IN THE MAIN                                             DSPLIN
C  ALPHA       - PERCENTAGE LEVEL OF SIGNIFICANCE TEST TO BE USED                   DSPLIN
C  TX,P,V      - AS DEFINED IN THE MAIN                                             DSPLIN
C                                                                                   DSPLIN
C OUTPUT PARAMETERS                                                                 DSPLIN
C  SO          - VARIANCE FACTOR                                                    DSPLIN
C  DEGOF       - DEGREES OF FREEDOM                                                 DSPLIN
C  VTWV        - QUADRATIC NORM OF RESIDUALS                                        DSPLIN
C                                                                                   DSPLIN
C WORK ARRAYS                                                                       DSPLIN
C              - DX1,DX2                                                            DSPLIN
C ADDITIONAL INFORMATICN                                                            DSPLIN
C   NEED TO CORRECT THE DEGREES OF FREEDOM IF THE OUTER KNOTS                       DSPLIN
C   ARE COINCIDENT WITH THE OUTER DATA POINTS. THE ROUTINE SCANS THRU               DSPLIN
C   THE WHOLE SET FOR THE COINCIDENCE OUTER DATA POINTS.                            DSPLIN
C                                                                                   DSPLIN
C CALLS        - BMULT1,CHITES                                                      DSPLIN
C              - CABS                                                               DSPLIN
C                              ****************                                     DSPLIN
C TRANSFER OF MATRIX DIMENSIONS VIA COMMON BLOCK                                    DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IOW1,IDTX,IDP,IOPV,IDLOBS,IDVAR,ICV                  DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,                  DSPLIN
     @       IDCOF,IDVCOF                                                           DSPLIN
C* ARRAYS                                                                           DSPLIN
      REAL*8 TX(IDTX,1),P(IDP,3),PV(IDPV,1),V(IDV,1)                                DSPLIN
      REAL*8 VTWV,DX1,DX2,SO,ALPHA                                                  DSPLIN
C* OTHERS                                                                           DSPLIN
      INTEGER R,M,N,NO,DEGCF                                                        DSPLIN
C* WRITE UNIT                                                                       DSPLIN
      IWR = 6                                                                       DSPLIN
C* POSTERIORI VARIANCE FACTOR                                                       DSPLIN
      CALL BMULT1(P,N,N,3,V,N,1,PV,IDV,IDP,1,IDPV,1)                                DSPLIN
      VTWV = 0.0D00                                                                 DSPLIN
      DO 10 I = 1,N                                                                 DSPLIN
 10       VTWV = VTWV + V(I,1)*PV(I,1)                                              DSPLIN
C* DEGREES OF FREEDOM FOR THE SYSTEM IS                                             DSPLIN
      DEGOF = N + R - M                                                             DSPLIN
C* CORRECTION TO DEGREES OF FREEDOM                                                 DSPLIN
      L = NO + 1                                                                    DSPLIN
```

```
      K = NO + 2
      DO 20 I = 2,L                                                     DSPLIN
      DX1 = TX(1,1)-TX(I,1)                                             DSPLIN
      DX2 = TX(K,1) - TX(I,1)                                           DSPLIN
      IF(DABS(DX1).LT.1.0D-10)DEGOF = DEGOF - 2                         DSPLIN
      IF(DABS(DX2).LT.1.0D-10)DEGOF = DEGOF - 2                         DSPLIN
   20 CONTINUE                                                          DSPLIN
C* PRINTS                                                               DSPLIN
      IF(DEGOF.EQ.0)S0=1                                                DSPLIN
      IF(DEGOF.NE.0)S0 = VTWV/DEGOF                                     DSPLIN
      WRITE(IWR,9000)DEGOF,S0                                           DSPLIN
C* CHI-SQUARE TEST                                                      DSPLIN
      IF(DEGOF.NE.0)CALL CHITES(S0,DEGOF,ALPHA,ICODE)                   DSPLIN
      IF(ICODE.EQ.0)WRITE(6,9010)ALPHA                                  DSPLIN
      IF(ICODE.NE.0)WRITE(6,9020)ALPHA                                  DSPLIN
 9000 FORMAT(///,3X,'DEG   OF  FDM ',I4,/,                              DSPLIN
     @           3X,'UNIT VARIANCE ',1PD12.5)                           DSPLIN
 9010 FORMAT(/,3X,'CHI-SQUARE TEST ON VARIANCE FACTOR AT ',F6.2,        DSPLIN
     @           ' CONFIDENCE INTERVAL - PASSED',/)                     DSPLIN
 9020 FORMAT(/,3X,'CHI-SQUARE TEST ON VARIANCE FACTOR AT ',F6.2,        DSPLIN
     @           ' CONFIDENCE INTERVAL - FAILED',/)                     DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE VELO(T,SPEED,DIR,VE,VN,SVE,SVN,CVEN,ICODE,             DSPLIN
     &                KNO,TK,COF,VCOF)                                  DSPLIN
      IMPLICIT REAL*8(A-H,M-Z)                                          DSPLIN
C *******************************************************************   DSPLIN
C * VERSION     : 3RD APRIL  1982                                   *   DSPLIN
C * DESCRIPTION : COMPUTE SMOOTHED VELOCITY AT TIME T               *   DSPLIN
C *******************************************************************   DSPLIN
C  INPUT  PARAMETERS                                                    DSPLIN
C    T      - TIME VELOCITY IS REQUIRED                                 DSPLIN
C    KNO    - NO OF KNOTS                                               DSPLIN
C    TK,COF,VCOF                                                        DSPLIN
C           - SEE MAIN PROGRAM                                          DSPLIN
C                                                                       DSPLIN
C  OUTPUT PARAMETERS                                                    DSPLIN
C    SPEED - COMPOSITE VELOCITY                                         DSPLIN
C    DIR   - DIRECTION OF COMPOSITE VELOCITY (RADIANS)                  DSPLIN
C    VE    - VELOCITY EASTWARDS                                         DSPLIN
C    VN    - VELOCITY NORTHWARDS                                        DSPLIN
C    SVE   - STANDARD DEVIATION OF VELOCITY EASTWARDS                   DSPLIN
C    SVN   - STANDARD DEVIATION OF VELOCITY NORTHWARDS                  DSPLIN
C    CVEN  - COVARIANCE OF THE TWO VELOCITY COMPONENTS                  DSPLIN
C    ICODE - 1 SUCCESSFUL CALL                                          DSPLIN
C          - 0 ERROR IN INPUT TIME                                      DSPLIN
C                                                                       DSPLIN
C CALLS    - INTERV,DMULT                                               DSPLIN
```

```
C          - DSQRT,DATAN2,DABS                                          DSPLIN
C                              ****************                          DSPLIN
      COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,ICLOBS,IDVAR,ICV      DSPLIN
      COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXC,      DSPLIN
     @      IDCOF,IDVCOF                                                DSPLIN
      REAL*8 X(2,1),VX(2,2),B(2,8),TK(IDTK,1),                          DSPLIN
     @           COF(IDCOF,8),VCOF(IDVCOF,8,8),CO(8,1),VCO(8,8)         DSPLIN
      INTEGER*4 KNO                                                     DSPLIN
      DATA ISWT/0/,B/4*0.0D0,1.0D0,2*0.0D0,1.0D0,8*0.0D0/               DSPLIN
      DATA PI/3.14159265358979D0/                                       DSPLIN
C* BEGINS                                                               DSPLIN
      ICODE = 1                                                         DSPLIN
      CALL INTERV(T,TK,KNO,J,IDTK,IC)                                   DSPLIN
      IF(IC.EQ.0)GO TO 10                                               DSPLIN
      DR = T - TK(J,1)                                                  DSPLIN
C* FEED IN THE DESIGN MATRIX B                                          DSPLIN
      B(1,5) = 2.0D0*DR                                                 DSPLIN
      B(1,7) = 3.0D0*DR*DR                                              DSPLIN
      B(2,6) = B(1,5)                                                   DSPLIN
      B(2,8) = B(1,7)                                                   DSPLIN
C* EXTRACTING THE COEFFICIENTS AND COVARIANCES                          DSPLIN
C*  IF THE INTERVAL IS THE SAME AS THE LAST CALL, THE TRANSFER OF       DSPLIN
C*  COEFF AND COVARIANCE IS OMITTED                                     DSPLIN
      IF(ISWT.EQ.J)GO TO 30                                             DSPLIN
      DO 20 IJ = 1,8                                                    DSPLIN
      CO(IJ,1) = COF(J,IJ)                                              DSPLIN
      DO 20 IK = 1,8                                                    DSPLIN
   20 VCO(IJ,IK) = VCOF(J,IJ,IK)                                        DSPLIN
      ISWT = J                                                          DSPLIN
   30 CALL DMULT(B,2,8,CO,8,1,X,2,8,8,1,2,1)                            DSPLIN
C* COMPUTE COVARIANCE                                                   DSPLIN
      DO 40 I = 1,2                                                     DSPLIN
           DO 40 L = 1,2                                                DSPLIN
                VX(I,L) = 0.0D0                                         DSPLIN
                  DO 40 J = 1,8                                         DSPLIN
                       DO 40 K = 1,8                                    DSPLIN
   40 VX(I,L) = VX(I,L) + B(I,K)*VCO(K,J)*B(L,J)                        DSPLIN
      SPEED = DSQRT(X(1,1)*X(1,1) + X(2,1)*X(2,1))                      DSPLIN
      IF(DABS(X(1,1)).LT.1.0D-20.AND.DABS(X(2,1)).LT.1.0D-20)DIR=0.0D0  DSPLIN
      IF(DABS(X(1,1)).LT.1.0D-20.AND.DABS(X(2,1)).LT.1.0D-20)GO TO 50   DSPLIN
      DIR = DATAN2(X(1,1),X(2,1))                                       DSPLIN
      IF(DIR.LT.0)DIR = DIR + 2.0D0*PI                                  DSPLIN
   50 VE = X(1,1)                                                       DSPLIN
      VN = X(2,1)                                                       DSPLIN
      SVE = DSQRT(VX(1,1))                                              DSPLIN
      SVN = DSQRT(VX(2,2))                                              DSPLIN
      CVEN = VX(1,2)                                                    DSPLIN
      RETURN                                                            DSPLIN
```

```
   10  WRITE(6,900)T
       ICODE = 0
       RETURN
  900  FORMAT(//,2X,'TIME ',D12.5,' AT WHICH POSITION IS REQUIRED IS ',
      @'OUTSIDE TIME INTERVAL OF OUTER KNOTS(VELO)',//)
       END
       SUBROUTINE VSMOTH(SCTIME,NO,KNO,TX,TK,COF,VCOF)
       IMPLICIT REAL*8(A-H,O-Z)
C  ***************************************************************
C  * VERSION     :  24  APRIL 1982                                *
C  * DESCRIPTION : COMPUTE SMOOTHED  DATA POINTS                  *
C  ***************************************************************
C INFUT PARAMETERS
C  NC,KNO   - NO OF POINTS AND KNOTS
C  SCTIME   - SCALE FACTOR ON TIME DOMAIN
C  TX,TK,COF,VCOF
C           - SEE MAIN FOR DESCRIPTION
C
C OUTPUT PARAMETERS
C  NCNE
C
C CALLS     - VELO,RADD
C                                       ***************
       COMMON /DIM1/IOBSNO,IDA1,IDW1,IDTX,IDP,IDPV,IDLOBS,IDVAR,ICV
       COMMON /DIM2/IDKNOT,IDA2,IDTK,IDDTK,IDX,IDVX,IDW2,IDV2,IDXO,
      @         IDCOF,IDVCOF
       REAL*8 TX(IDTX,1),TK(IDTK,1),COF(IDCOF,8),VCOF(IDVCOF,8,8)
       INTEGER NC,KNO
C* WRITE UNIT
       IWR = 6
C* CCNVERSION SCALE FACTOR TC CORRECT UNITS
       DCONV = 24.0D0
C* TITLE
       WRITE(IWR,9010)
       WRITE(IWR,9020)
C* COMPUTE THE SMCOTHED VALUES OF THE VELOCITIES
       IL = NO + 2
       DO 10 I = 1,IL
       CALL VELO(TX(I,1),SPEED,DIR,VE,VN,SVE,SVN,CVEN,ICO,
      &          KNO,TK,CCF,VCOF)
       DR = RADD(DIR)
       II = I - 1
       TRU = TX(I,1)/SCTIME
C* CONVERSION FROM M PER DAY TO M PER HOUR - DCONV = 24.0
       SPEED = SCTIME*SPEED/DCONV
       VE = SCTIME*VE/DCONV
       VN = SCTIME*VN/DCONV
       SVE = SCTIME*SVE/DCONV
```

```
      SVN = SCTIME*SVN/DCONV                                            DSPLIN
      CVEN = SCTIME*SCTIME*CVEN/(DCONV*DCONV)                           DSPLIN
      WRITE(IWR,9030)II,TRU,SPEED,DR,VE,VN,SVE,SVN,CVEN                 DSPLIN
   10 CONTINUE                                                          DSPLIN
 9010 FORMAT(///,3X,'SMOOTHED VELOCITIES AT INPUT DATA TIMES',/)        DSPLIN
 9020 FORMAT(//,10X,'PREDICTED VELOCITIES AT SPECIFIED TIMES',/10X,     DSPLIN
     .19('=='),//,2X,'NO',6X,'TIME',9X,'SPEED(M/HR)',3X,'DIR(DEG)',7X,  DSPLIN
     .'VELOCITIES(E,N):M/HR        SIGMA VELO(E,N)',9X,' COVARIANCE',   DSPLIN
     .//)                                                               DSPLIN
 9030 FORMAT(2X,I4,2X,F10.4,10(F12.3,2X))                               DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE LS2(C,D,G,H,P,W1,W2,X,VX,V,R,M,N,IC,JC,ID,JD,          DSPLIN
     @                 IG,JG,IH,JH,IP,JP,IW1,IW2,IX,IV,IER)             DSPLIN
       IMPLICIT REAL*8(A-H,P-Z)                                         DSPLIN
C *****************************************************************    DSPLIN
C * VERSION     : JANUARY 1982                                     *    DSPLIN
C * DESCRIPTION : PERFORMS THE METHOD OF LEAST SUARES USING THE    *    DSPLIN
C *               ELIMINATION OF CONSTRAINT EQUATIONS APPROACH .   *    DSPLIN
C *****************************************************************    DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C    R,M,N  - LS PROBLEM SPECS ; R - NO OF CONSTRAINT EQNS              DSPLIN
C                                M - NO OF UNKNOWNS                     DSPLIN
C                                N - NO OF OBSERVATIONS                 DSPLIN
C    C, D   - PARTITIONED MATRICES OF A1 DESIGN MATRIX - PRIMARY MODEL  DSPLIN
C    G, H   - PARTITIONED MATRICES OF A2 DESIGN MATRIX - SECONDARY MODELDSPLIN
C              DIMENSIONS     H - R X R                                 DSPLIN
C                             G - R X M                                 DSPLIN
C                             C - N X M                                 DSPLIN
C                             D - N X R                                 DSPLIN
C              DECLARED DIMENSIONS IS AS STATED BELOW                   DSPLIN
C    W1,W2  - MISCLOSURE VECTORS                                        DSPLIN
C    IC,JC,ID,JD,IG,JG,IH,JH,IP,JP,IW1,IW2                              DSPLIN
C           - DECLARED DIMENSIONS OF MATRICES( SEE BELOW)               DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C    X,VX   - LS ESTIMATES AND COVARIANCE                               DSPLIN
C    V      - RESIDUALS                                                 DSPLIN
C    IX,IV  - DECLARED DIMENSIONS OF MATRICES X,VX AND V(SEE BELOW)     DSPLIN
C    IER    - ERROR CODE                                                DSPLIN
C                                                                       DSPLIN
C ADDITIONAL NOTES                                                      DSPLIN
C  CURRENT ARRAY SETTINGS LIMIT THE SOLUTION TO ONLY 100 ESTIMATES      DSPLIN
C                                                                       DSPLIN
C CALLS     - COND,DINV,CHOLD,DMULT,DPRINT,SCMULT,DMAG,PCOPY            DSPLIN
C                               ****************                        DSPLIN
      REAL*8 C(IC,JC),D(ID,JD),G(IG,JG),H(IH,JH),                       DSPLIN
     @          P(IP,JP),W1(IW1,1),W2(IW2,1),                           DSPLIN
```

```
     @          X(IX,1),VX(IX,IX),V(IV,1)                                DSPLIN
C* ARRAYS USED BY THE PROGRAM                                            DSPLIN
        REAL*8 A(58,50),W(58,1),ATPW(50,1),ATP(50,58),HG(50,50),         DSPLIN
     @          HW(50,1),XT(50,1),NHGT(50,50),VM(50,50),ATPV(50,1)       DSPLIN
       INTEGER*4 R,M,N                                                   DSPLIN
C*                                                                       DSPLIN
       EQUIVALENCE (ATP(1),NHGT(1)),                                     DSPLIN
     @             (A(1),VM(1)),                                         DSPLIN
     @             (W(1),XT(1),ATPV(1))                                  DSPLIN
C*                                                                       DSPLIN
C* FUNCTION - COMPUTES THE NO OF CODIAGONALS OF THE P MATRIX             DSPLIN
       ICO = (JP-1)/2                                                    DSPLIN
C*                                                                       DSPLIN
C* PROGRAM BEGINS BY INVERTING THE H MATRIX                              DSPLIN
C* POSSIBILITY OF INCOPORATING A TEST FOR THE SINGULARITY OF THE         DSPLIN
C* H MATRIX HERE .CURRENTLY IT IS NOT NEEDED.                            DSPLIN
       CALL COND(H,R,IH,1)                                               DSPLIN
       CALL DINV(R,H,IH)                                                 DSPLIN
       CALL COND(H,R,IH,2)                                               DSPLIN
C* NOTE THAT THE ABOVE INVERSE IS USED AS THE H MATRIX IS NON-SYSMETRICADSPLIN
       CALL DMULT(H,R,R,G,R,M,HG,IH,JH,IG,JG,50,50)                      DSPLIN
       CALL DMULT(H,R,R,W2,R,1,HW,IH,JH,IW2,1,50,1)                      DSPLIN
C* COMPUTING THE REVISED A AND W MATRICES                                DSPLIN
       DO 10 I = 1,N                                                     DSPLIN
           DO 10 J =1 ,M                                                 DSPLIN
               SUM = 0.0D0                                               DSPLIN
               DO 20 K = 1,R                                             DSPLIN
   20  SUM = SUM + D(I,K)*HG(K,J)                                        DSPLIN
   10  A(I,J) = C(I,J) - SUM                                             DSPLIN
       DO 50 I = 1,N                                                     DSPLIN
           SUM = 0.0D0                                                   DSPLIN
           DO 52 J = 1,R                                                 DSPLIN
   52          SUM = SUM   + D(I,J)*HW(J,1)                              DSPLIN
   50  W(I,1) = W1(I,1) - SUM                                            DSPLIN
C* NORMAL LEAST SQUARES WITH REVISED MATRICES                            DSPLIN
       DO 30 I = 1,M                                                     DSPLIN
          DO 30 J = 1,N                                                  DSPLIN
              ATP(I,J) = 0.0D0                                           DSPLIN
              IB = J -ICO                                                DSPLIN
              IE = J + ICO                                               DSPLIN
              IF(IB.LT.1)IB = 1                                          DSPLIN
              IF(IE.GT.N)IE = N                                          DSPLIN
             DO 30 K = IB,IE                                             DSPLIN
                JAC = J - K + ICO + 1                                    DSPLIN
   30  ATP(I,J) = ATP(I,J) + A(K,I)*P(K,JAC)                             DSPLIN
       DO 40 I = 1,M                                                     DSPLIN
          DO 40 J = 1,M                                                  DSPLIN
              VX(I,J) = 0.0D0                                            DSPLIN
```

```
                DO 40 K = 1,N
   40     VX(I,J) = VX(I,J) + ATP(I,K)*A(K,J)                             DSPLIN
          CALL CHOLD(VX,M,IX,M,DET,IER)                                   DSPLIN
          IF(IER.NE.0)STOP                                                DSPLIN
          CALL DMULT(ATP,M,N,W,N,1,ATPW,50,58,58,1,50,1)                  DSPLIN
C* FORMULATION OF THE X VECTOR                                            DSPLIN
          CALL DMULT(VX,M,M,ATPW,M,1,X,IX,IX,50,1,IX,1)                   DSPLIN
          CALL SCMULT(-1.0D0,X,X,M,1,IX,1)                                DSPLIN
          CALL DPRINT(X,M,1,IX,1,'X1-VECTOR.',5)                          DSPLIN
C*                                                                        DSPLIN
C* COMPUTATION OF RESIDUAL AND INTERNAL COMPUTATIONAL ACCURACY            DSPLIN
C* WITH THE FUTURE POSSIBILITY OF HALTING PROCESS AND IS PUT HERE         DSPLIN
C* INSTEAD OF AT THE CUSTOMERY END IS DUE TO THE EXISTANCE OF THE         DSPLIN
C* EQUIVALENCE STATEMENT ON CERTAIN ARRAYS                                DSPLIN
C*                                                                        DSPLIN
C* RESIDUAL COMPUTATION                                                   DSPLIN
          DO 110 I = 1,N                                                  DSPLIN
              SUM = 0.0D0                                                 DSPLIN
             DO 120 J = 1,M                                               DSPLIN
  120           SUM = SUM + A(I,J)*X(J,1)                                 DSPLIN
  110      V(I,1) = -(W(I,1) + SUM)                                       DSPLIN
C* ATPV MATRIX COMPUTATION                                                DSPLIN
          CALL DMULT(ATP,M,N,V,N,1,ATPV,50,58,IV,1,50,1)                  DSPLIN
          CALL DMAG(ATPV,M,1,50,1,XLR,2)                                  DSPLIN
          IF(XLR.GT.1.0D-10)WRITE(6,9010)                                 DSPLIN
C* RECOVERY OF THE LOWER HALF OF THE X VECTOR ; STORED IN XT VECTOR       DSPLIN
          DO 70 I = 1,R                                                   DSPLIN
              SUM = 0.0D0                                                 DSPLIN
             DO 72 K = 1,M                                                DSPLIN
   72           SUM = SUM + HG(I,K)*X(K,1)                                DSPLIN
   70     XT(I,1) = -(HW(I,1) + SUM)                                      DSPLIN
C* TRANSFER XT VECTOR TO X VECTOR                                         DSPLIN
          L = M + 1                                                       DSPLIN
          CALL PCOPY(XT,1,1,R,1,X,L,1,50,1,IX,1)                          DSPLIN
          CALL DPRINT(XT,R,1,50,1,'X2-VECTOR.',5)                         DSPLIN
C* COMPUTE THE COVARIANCE MATRIX OF X1X2                                  DSPLIN
          DO 80 I = 1,M                                                   DSPLIN
              DO 80 J = 1,R                                               DSPLIN
                  NHGT(I,J) = 0.0D0                                       DSPLIN
                 DO 90 K = 1,M                                            DSPLIN
   90                NHGT(I,J) = NHGT(I,J) + VX(I,K)*HG(J,K)              DSPLIN
   80     NHGT(I,J) = -NHGT(I,J)                                          DSPLIN
C* COMPUTE THE COVARIANCE MATRIX OF X2                                    DSPLIN
          CALL DMULT(HG,R,M,NHGT,M,R,VM,50,50,50,50,50,50)                DSPLIN
          CALL SCMULT(-1.0D0,VM,VM,R,R,50,50)                             DSPLIN
C* TRANSFER OF THE COVARIANCE MATRIX OF X2 IN VX                          DSPLIN
          L = M + 1                                                       DSPLIN
          CALL PCOPY(VM,1,1,R,R,VX,L,L,50,50,IX,IX)                       DSPLIN
```

```
C* TRANSFER OF THE COVARIANCE MATRIX X1X2 AND GENERATE THE COVARIANCE   DSPLIN
C* X2X1 IN THE VX MATRIX.                                               DSPLIN
      DO 100 I = 1,M                                                    DSPLIN
           DO 100 J = 1,R                                               DSPLIN
               K = J + M                                                DSPLIN
               VX(I,K) = NHGT(I,J)                                      DSPLIN
  100 VX(K,I) = VX(I,K)                                                 DSPLIN
      IER = 0                                                           DSPLIN
      RETURN                                                            DSPLIN
 9010 FORMAT(3X,'TERMINAL ERROR  FAILURE OF SOLV2 ',//)                 DSPLIN
      END                                                               DSPLIN
      SUBROUTINE TRNSFM(ADDR,A,N,M,C1,C2,IC,IA,JA,IC1,JC1,IC2,JC2)      DSPLIN
      REAL*8 A(IA,JA),C1(IC1,JC1),C2(IC2,JC2)                           DSPLIN
      INTEGER*4 ADDR(IA)                                                DSPLIN
C ****************************************************************      DSPLIN
C * VERSION      : JANUARY 1982                                   *     DSPLIN
C * DESCRIPTION : REARRANGEMENT OF DESIGN MATRICES A1 OR A2       *     DSPLIN
C *               USED IN THE FUNCTIONAL CONSTRAINT  MODEL TO     *     DSPLIN
C *               THAT USED IN THE ELIMINATION OF CONSTRAINTS     *     DSPLIN
C *               MODEL.                                          *     DSPLIN
C ****************************************************************      DSPLIN
C INPUT PARAMETERS                                                      DSPLIN
C  A        - DESIGN MATRIX EITHER A1 OR A2 - (CALLED 'A' HENCEFORTH)   DSPLIN
C  N,M      - UPPER LEFT HAND PORTION OF THE MATRIX A USED              DSPLIN
C  IA,JA    - DECLARED SIZE OF MATRIX A                                 DSPLIN
C  ADDR     - ADDRESS SEQUENCE(DECLARED SIZE=IA) OF BANDED MATRIX A     DSPLIN
C  IC       - CODING ON THE TYPE OF INPUT BANDED MATRIX A               DSPLIN
C              IC = 1 THEN THE ASSUMED BAND WIDTH IS 7  : I.E  FOR A1   DSPLIN
C                 = 2 THEN THE ASSUMED BAND WIDTH IS 11 : I.E  FOR A2   DSPLIN
C                                                                       DSPLIN
C OUTPUT PARAMETERS                                                     DSPLIN
C  C1,C2    - REARRANGED AND PARTITIONED MATRICES OF MATRIX A           DSPLIN
C  IC1,JC1- DECLARED SIZE OF C1 MATRIX                                  DSPLIN
C  IC2,JC2- DECLARED SIZE OF C2 MATRIX                                  DSPLIN
C                                                                       DSPLIN
C ADDITIONAL NOTES                                                      DSPLIN
C  THE ROUTINE CAN TAKE EITHER A BANDED MATRIX OF 7 OR 11 OR ANY OTHER  DSPLIN
C     DESIRED(I.E. BY ASSIGNING THE IBAND VARIABLE TO THE DESIRED       DSPLIN
C     BANDWIDTH OF THE INPUT MATRIX A.                                  DSPLIN
C  THERE  EXIST THE POSSIBILITY OF BANDING C1 AND C2 MATRICES.          DSPLIN
C     THE ALGORITHM CURRENTLY GENERATES FULL MATRICES.                  DSPLIN
C                                                                       DSPLIN
C CALLS     - DCLEAR                                                    DSPLIN
C* FUNCTION DEFINATION                                                  DSPLIN
      ICOL(I) = (2 - ((I+1)/2 - (I/2)))                                 DSPLIN
      IROW(I) = (I+1)/2                                                 DSPLIN
C* CLEARING ALL MATRICES                                                DSPLIN
      CALL DCLEAR(C1,IC1,JC1)                                           DSPLIN
```

```
      CALL DCLEAR(C2,IC2,JC2)                                           DSPLIN
C* BANDWIDTH SPECIFICATIONS                                             DSPLIN
      IF(IC.EQ.1)IBAND = 7                                              DSPLIN
      IF(IC.EQ.2)IBAND = 11                                             DSPLIN
C* COUNTERS                                                             DSPLIN
      II = -3                                                           DSPLIN
      IE = M-2                                                          DSPLIN
      K = 0                                                             DSPLIN
C* FORM C1                                                              DSPLIN
  100 II = II + 4                                                       DSPLIN
      I2 = II + 1                                                       DSPLIN
      DO 20 J = II,I2                                                   DSPLIN
         K = K + 1                                                      DSPLIN
         DO 20 I = 1,N                                                  DSPLIN
            IR = IROW(I)                                                DSPLIN
            IPOS = ADDR(IR) + ICOL(I) -1                                DSPLIN
            JJ = J - IPOS + 1                                           DSPLIN
            IF(JJ.LT.1.OR.JJ.GT.IBAND)GO TO 20                          DSPLIN
            C1(I,K) = A(IR,JJ)                                          DSPLIN
   20 CONTINUE                                                          DSPLIN
C* END COUNTER                                                          DSPLIN
      IF(I2.NE.IE)GO TO 100                                             DSPLIN
C*                                                                      DSPLIN
C   C1 COMPLETED                                                        DSPLIN
C*                                                                      DSPLIN
C* RESET COUNTERS                                                       DSPLIN
      II = -1                                                           DSPLIN
      K = 0                                                             DSPLIN
      IE = M                                                            DSPLIN
C* FORMS C2                                                             DSPLIN
  200 II = II + 4                                                       DSPLIN
      I2 = II + 1                                                       DSPLIN
      DO 30 J = II,I2                                                   DSPLIN
         K = K + 1                                                      DSPLIN
         DO 30 I = 1,N                                                  DSPLIN
           IR = IROW(I)                                                 DSPLIN
           IPOS = ADDR(IR) + ICOL(I) - 1                                DSPLIN
           JJ = J - IPOS + 1                                            DSPLIN
           IF(JJ.LT.1.OR.JJ.GT.IBAND)GO TO 30                           DSPLIN
           C2(I,K) = A(IR,JJ)                                           DSPLIN
   30 CONTINUE                                                          DSPLIN
C*  END DETECTOR                                                        DSPLIN
      IF(I2.NE.IE)GO TO 200                                             DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE PERM(X,VX,M,II)                                        DSPLIN
      REAL*8 X(II,1),VX(II,II),T                                        DSPLIN
      INTEGER*4 P(100,1),TRACK(100)                                     DSPLIN
```

```
C ******************************************************************** OSPLIN
C * VERSION       : JANUARY 1982                                     * OSPLIN
C * DESCRIPTION : PERFORMS THE PERMUTATION OF MATRICES X AND VX      * OSPLIN
C *               (SOLUTION VECTOR AND ITS COVARIANCE) COMPUTED      * OSPLIN
C *               VIA THE ELIMINATION OF CONSTRAINTS MODEL TO THAT   * OSPLIN
C *               AS IF WERE COMPUTED VIA THE FUNCTIONAL CONSTRAINT    OSPLIN
C *               MODEL.                                             * OSPLIN
C ******************************************************************** OSPLIN
C INPUT PARAMETERS                                                     OSPLIN
C  X,VX - SOLUTION VECTOR AND ITS COVARIANCE                           OSPLIN
C  II   - DELCLARED DIMENSIONS OF V AND VX                             OSPLIN
C  M    - UPPER LEFT HAND CORNER CONTAINING THE MATRIX ELEMENTS        OSPLIN
C                                                                      OSPLIN
C OUTPUT PARAMETERS                                                    OSPLIN
C  X,VX  - PERMUTATED MATRICES X AND VX.                               OSPLIN
C                                                                      OSPLIN
C ADDITIONAL NOTES                                                     OSPLIN
C  THE PERMUTATION SEQUENCE IS FIXED BY THE PROGRAM.                   OSPLIN
C  THE ALOGORITHM  WILL OPERATE ONLY ON AN EVEN ORDER SQUARE MATRIX.   OSPLIN
C  THIS IS NOT CHECKED ANYWHERE.                                       OSPLIN
C                                                                      OSPLIN
C*PROGRAM BEGINS                                                       OSPLIN
C START POINTERS                                                       OSPLIN
      J = M/2-1                                                        OSPLIN
      I = -1                                                           OSPLIN
      IK = M - 3                                                       OSPLIN
      DO 10 K = 1,IK,4                                                 OSPLIN
C* INCREMENT                                                           OSPLIN
         I = I + 2                                                     OSPLIN
         I2 = I + 1                                                    OSPLIN
         J = J + 2                                                     OSPLIN
         J2 = J + 1                                                    OSPLIN
         K2 = K + 1                                                    OSPLIN
         K3 = K + 2                                                    OSPLIN
         K4 = K + 3                                                    OSPLIN
C* FEED                                                                OSPLIN
         P(K,1) = I                                                    OSPLIN
         P(K2,1) = I2                                                  OSPLIN
         P(K3,1) = J                                                   OSPLIN
 10   P(K4,1) = J2                                                     OSPLIN
C* SETS TRACK SEQUENCE ; MONITORS THE ACTUAL LOCATION OF THE ROW/COLUMN OSPLIN
      DO 20 I = 1,M                                                    OSPLIN
 20   TRACK(I) = I                                                     OSPLIN
C*                                                                     OSPLIN
C* ROW EXCHANGES OF THE MATRIX                                         OSPLIN
C*                                                                     OSPLIN
      DO 30 I = 1,M                                                    OSPLIN
         IS = I                                                        OSPLIN
```

```
            IR = P(I,1)                                                 DSPLIN
            ITR = TRACK(IR)                                             DSPLIN
            DO 40 J = 1,M                                               DSPLIN
               T = VX(IS,J)                                             DSPLIN
               VX(IS,J) = VX(ITR,J)                                     DSPLIN
               VX(ITR,J) = T                                            DSPLIN
   40    CONTINUE                                                       DSPLIN
         DO 42 K = 1,M                                                  DSPLIN
            IF(IS.EQ.TRACK(K))GO TO 44                                  DSPLIN
   42    CONTINUE                                                       DSPLIN
   44    TRACK(K) = ITR                                                 DSPLIN
         TRACK(IR) = IS                                                 DSPLIN
   30    CONTINUE                                                       DSPLIN
         DO 45 I = 1,M                                                  DSPLIN
   45    TRACK(I) = I                                                   DSPLIN
C*                                                                      DSPLIN
C* DOING THE COLUMN EXCHANGES                                           DSPLIN
C*                                                                      DSPLIN
         DO 50 J = 1,M                                                  DSPLIN
            IS = J                                                      DSPLIN
            IR = P(J,1)                                                 DSPLIN
            ITR = TRACK(IR)                                             DSPLIN
            DO 60 I = 1,M                                               DSPLIN
               T = VX(I,IS)                                             DSPLIN
               VX(I,IS) = VX(I,ITR)                                     DSPLIN
               VX(I,ITR) = T                                            DSPLIN
   60    CONTINUE                                                       DSPLIN
         T = X(IS,1)                                                    DSPLIN
         X(IS,1) = X(ITR,1)                                             DSPLIN
         X(ITR,1) = T                                                   DSPLIN
         DO 55 K = 1,M                                                  DSPLIN
            IF(IS.EQ.TRACK(K))GO TO 57                                  DSPLIN
   55    CONTINUE                                                       DSPLIN
C*                                                                      DSPLIN
  57     TRACK(K) = ITR                                                 DSPLIN
         TRACK(IR) = IS                                                 DSPLIN
   50    CONTINUE                                                       DSPLIN
C*                                                                      DSPLIN
C* TASK COMPLETED                                                       DSPLIN
         RETURN                                                         DSPLIN
         END                                                            DSPLIN
C ********************************************************************* DSPLIN
C * VERSION      : 1980                                               * DSPLIN
C * DESCRIPTION  : BLOCK OF DESIGNATED LEVEL THREE ROUTINES           * DSPLIN
C * CONTENTS     : CTIME,DADD,DCLEAR,DCOPY,CINV,DMAG,DMULT,           * DSPLIN
C *                DPRINT,DSUBT,DTRAN,PCCPY,RADD,SCMULT               * DSPLIN
C * EXTERNAL     : DABS,DATAN                                         * DSPLIN
C ********************************************************************* DSPLIN
```

```
      SUBROUTINE CTIME(TIME,IDAY,IHR,IMIN)                               DSPLIN
C*---- CONVERTS TIME IN DAYS TO DAY HOUR AND MINS                        DSPLIN
C*                                                                       DSPLIN
      REAL*8 TIME,XHR                                                    DSPLIN
      IDAY = TIME                                                        DSPLIN
      XHR = (TIME-IDAY)*24.0D0                                           DSPLIN
      IHR = XHR                                                          DSPLIN
      IMIN = (XHR - IHR)*60.0D0                                          DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE DADD(F,S,K,L,R,N,M)                                     DSPLIN
C*-----TO ADD MATRICES F + S = R                                         DSPLIN
C*                                                                       DSPLIN
      DOUBLE PRECISION F(N,M),S(N,M),R(N,M)                              DSPLIN
      DO 10 I = 1,K                                                      DSPLIN
      DO 10 J = 1,L                                                      DSPLIN
   10 R(I,J) = F(I,J) + S(I,J)                                           DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE DCLEAR(A,N,M)                                           DSPLIN
C*---- SUBROUTINE CLEARS CONTENTS OF A MATRIX                            DSPLIN
C*                                                                       DSPLIN
      DOUBLE PRECISION A(N,M)                                            DSPLIN
      DO 10 I = 1,N                                                      DSPLIN
      DO 10 J = 1,M                                                      DSPLIN
  10  A(I,J) = 0.0                                                       DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE DCOPY(FR,TO,N,M,K,L,II,IJ)                              DSPLIN
C*------TO COPY MATRICES                                                 DSPLIN
C*      N,M IS THE PART TO BE COPIED                                     DSPLIN
C*      K,L IS THE DECLARED SIZE OF FR MATRICES                          DSPLIN
C*       II,IJ IS THE DECLARED SIZE OF TO MATRIX                         DSPLIN
C*                                                                       DSPLIN
      DOUBLE PRECISION FR(K,L) , TO(II,IJ)                               DSPLIN
      DO 10 I = 1,N                                                      DSPLIN
      DO 10 J = 1,M                                                      DSPLIN
  10  TO(I,J) = FR(I,J)                                                  DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE DINV(N,A,L)                                             DSPLIN
C*----- SUBROUTINE INVERTS THE GIVEN MATRIX                              DSPLIN
C*     ORIGINAL MATRIX IS DESTROYED                                      DSPLIN
C*       L  IS THE SIZE OF THE DECLARED MATRIX A                         DSPLIN
C*       N  IS THE PART OF THE MATRIX A TO BE INVERTED                   DSPLIN
C*                                                                       DSPLIN
      INTEGER*4 L,N                                                      DSPLIN
      DOUBLE PRECISION A(L,L),X                                          DSPLIN
```

```
      DO 10 I = 1,N                                                     DSPLIN
      X = A(I,I)                                                        DSPLIN
      IF(DABS(X).LT.1.0D-10)GO TO 999                                   DSPLIN
      A(I,I) = DBLE(1.0)                                                DSPLIN
      DO 20 J = 1,N                                                     DSPLIN
   20 A(I,J) = A(I,J)/X                                                 DSPLIN
      DO 10 K = 1,N                                                     DSPLIN
      IF(K-I)30,10,30                                                   DSPLIN
   30 X = A(K,I)                                                        DSPLIN
      A(K,I) = DBLE(0.0)                                                DSPLIN
      DO 40 J = 1,N                                                     DSPLIN
   40 A(K,J) = A(K,J) - X*A(I,J)                                        DSPLIN
   10 CONTINUE                                                          DSPLIN
      RETURN                                                            DSPLIN
  999 WRITE(6,100)I                                                     DSPLIN
  100 FORMAT(/,3X,'DIAGONAL ELEMENT',I4,' IS LT 1.0D-10 IN DINV ')      DSPLIN
      CALL DPRINT(A,N,N,L,L,'PARTIALLY  INVERTED MATRIX',13)            DSPLIN
      STOP                                                              DSPLIN
      END                                                               DSPLIN
      SUBROUTINE DMAG(A,N,M,K,L,XNUMB,IC)                               DSPLIN
C-----FINDS THE LARGEST OR SMALLEST VALUE IN A MATRIX                   DSPLIN
C*    DECLARED DIMENSIONS OF MATRIX A IS K,L                            DSPLIN
C*                                                                      DSPLIN
C*    IF CODE IC = 0  THEN THE LARGEST VALUE IS SEEKED                  DSPLIN
C*               = 1  THEN THE SMALLEST VALUE IS SEEKED                 DSPLIN
C*               = 2  THEN THE LARGEST ABSOLUTE VALUE IS SEEKED         DSPLIN
C*                                                                      DSPLIN
      REAL*8  A(K,L),XNUMB,XX                                           DSPLIN
      XNUMB=A(1,1)                                                      DSPLIN
      IF(IC.EQ.2)XNUMB = DABS(A(1,1))                                   DSPLIN
      IF(IC.EQ.0.OR.IC.EQ.1.OR.IC.EQ.2)GO TO 20                         DSPLIN
      WRITE(6,30)IC                                                     DSPLIN
   30 FORMAT(//,3X,' CODING ERROR ON SUBROUTINE DMAG ',//)              DSPLIN
      STOP                                                              DSPLIN
   20 DO 10 I = 1,N                                                     DSPLIN
      DO 10 J = 1,M                                                     DSPLIN
      XX = A(I,J)                                                       DSPLIN
      IF(IC.EQ.2)XX = DABS(XX)                                          DSPLIN
      IF(IC.EQ.0.AND.XX.GT.XNUMB)XNUMB = XX                             DSPLIN
      IF(IC.EQ.1.AND.XX.LT.XNUMB)XNUMB= XX                              DSPLIN
      IF(IC.EQ.2.AND.XX.GT.XNUMB)XNUMB = XX                             DSPLIN
   10 CONTINUE                                                          DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE DMULT(F,I,J,S,K,L,R,NF,MF,NS,MS,NR,MR)                 DSPLIN
C*-----TO MULTIPLY MATRICES F X S = R                                   DSPLIN
C*                                                                      DSPLIN
      DOUBLE PRECISION F(NF,MF),S(NS,MS),R(NR,MR)                       DSPLIN
```

```
      IF(J.NE.K)WRITE(6,10)                                                   DSPLIN
C **** SET THE OUTPUT WARNING DEVICE                                           DSPLIN
   10 FORMAT(1X,'FATAL ** MATRICES NOT CONFORMAL IN DMULT SUBT')              DSPLIN
      IF(J.NE.K) CALL EXIT                                                    DSPLIN
      DO 20 II= 1,I                                                           DSPLIN
      DO 20 IJ = 1,L                                                          DSPLIN
      R(II,IJ)  = 0.0                                                         DSPLIN
      DO 20 IK = 1,J                                                          DSPLIN
   20 R(II,IJ) = R(II,IJ) + F(II,IK)*S(IK,IJ)                                 DSPLIN
      RETURN                                                                  DSPLIN
      END                                                                     DSPLIN
      SUBROUTINE DPRINT(A,N,M,K,L,ITITLE,IC)                                  DSPLIN
C*-----TO PRINT OUT MATRICES OF DIMENSION N,M                                  DSPLIN
C*      ITITLE OF THE NAME ASSIGNED TO THE MATRIX. EG 'ITITLE'                DSPLIN
C*      IC IS THE NUMBER , MULTIPLES OF 2, OF THE CHARACTERS                  DSPLIN
C*      IN THE TITLE OF THE MATRIX.                                           DSPLIN
C*                                                                            DSPLIN
C*      YOU ARE ADVISED TO HAVE TITLES WITH EVEN NUMBER OF                    DSPLIN
C*      CHARACTERS IN THEM. FOR ODD NUMBERED CHARACTERS, USE BLANKS           DSPLIN
C*      TO MAKE IT EVEN. SUBROUTINE WILL FAIL IF IC*2 > NO.                   DSPLIN
C*      OF CHARACTERS IN ITITLE                                               DSPLIN
C*                                                                            DSPLIN
C*      EXAMPLE OF USE - CALL DPRINT(MATRIX,3,3,5,5,'ATA ',4)                 DSPLIN
C*                                                                            DSPLIN
      DOUBLE PRECISION A(K,L)                                                 DSPLIN
      INTEGER*2 ITITLE(IC),ILIN(30)                                           DSPLIN
      DATA ILIN(1)/'=='/                                                      DSPLIN
      DO 40 II = 2,30                                                         DSPLIN
   40 ILIN(II) = ILIN(1)                                                      DSPLIN
      IF(IC.EQ.0)IC = 1                                                       DSPLIN
      WRITE(6,30)(ITITLE(I),I = 1,IC)                                         DSPLIN
      WRITE(6,50)(ILIN(I),I = 1,IC)                                           DSPLIN
      IF(M.EQ.1)WRITE(6,20)(A(I,1),I = 1,N)                                   DSPLIN
      IF(M.EQ.1)GO TO 60                                                      DSPLIN
      DO 10 I = 1,N                                                           DSPLIN
   10 WRITE(6,20)(A(I,J),J= 1,M)                                              DSPLIN
   60 CALL SPACE(3)                                                           DSPLIN
   20 FORMAT(30(/,3X,8(1PD12.5,1X)))                                          DSPLIN
   30 FORMAT(///,3X,30A2)                                                     DSPLIN
   50 FORMAT(3X,30A2)                                                         DSPLIN
      RETURN                                                                  DSPLIN
      END                                                                     DSPLIN
      SUBROUTINE DSUBT(FR,TO,K,L,R,N,M)                                       DSPLIN
C*-----TO SUBTRACT MATRICE FR - TO = R                                         DSPLIN
C*                                                                            DSPLIN
      DOUBLE PRECISION FR(N,M),TO(N,M),R(N,M)                                 DSPLIN
      DO 10 I = 1,K                                                           DSPLIN
      DO 10 J = 1,L                                                           DSPLIN
```

```
   10 R(I,J) = FR(I,J) - TC(I,J)                                         DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE DTRAN(FR,TO,K,L,N,M)                                    DSPLIN
C*----- TO TRANSPOSE MATRICES                                            DSPLIN
C*       K,L IS THE PART TO BE TRANSPOSED                                DSPLIN
C*       N,M IS THE SIZE OF THE DECLARED MATRIX                          DSPLIN
C*                                                                       DSPLIN
      DOUBLE PRECISION FR(N,M),TO(M,N)                                   DSPLIN
      DO 10 I = 1,K                                                      DSPLIN
      DO 10 J = 1,L                                                      DSPLIN
   10 TO(J,I) = FR(I,J)                                                  DSPLIN
      RETURN                                                             DSPLIN
      END                                                                DSPLIN
      SUBROUTINE PCOPY(FR,T1,T2,B1,B2,TO,TS1,TS2,II,K,M,N)               DSPLIN
C-----PART COPYING OF A MATRIX                                           DSPLIN
C*                                                                       DSPLIN
      IMPLICIT INTEGER*4(I)                                              DSPLIN
      INTEGER*4 II,K,M,N,T1,T2,B2,B1,TS1,TS2,J                           DSPLIN
      REAL*8 FR(II,K), TO(M,N)                                           DSPLIN
C*                                                                       DSPLIN
C*   TRANSFER OF PORTIONS OF A MATRIX TO ANOTHER MATRIX                  DSPLIN
C*                                                                       DSPLIN
C*      FR  | T1,T2      |          TO  |TS1,TS2        |                DSPLIN
C*    MATRIX|            |   =    MATRIX|               |                DSPLIN
C*          |       B1,B2|              |            #,#|                DSPLIN
C*                                                                       DSPLIN
C* BEGIN- #,# CALCULATED BY SUBROUTINE                                   DSPLIN
      IXD = B1-T1+ 1                                                     DSPLIN
      IYD = B2-T2+ 1                                                     DSPLIN
      IF(IXD.EQ.1)GO TO 10                                               DSPLIN
      IF(IYD.EQ.1)GO TO 20                                               DSPLIN
      DO 30 I = 1,IXD                                                    DSPLIN
      IXN = TS1+I-1                                                      DSPLIN
      IXO = T1+I-1                                                       DSPLIN
      DO 30 J = 1,IYD                                                    DSPLIN
      IYN = TS2 + J-1                                                    DSPLIN
      IYO = T2 + J-1                                                     DSPLIN
      IF(IXN.GT.M.OR.IYN.GT.N)GO TO 60                                   DSPLIN
   30 TO(IXN,IYN) = FR(IXO,IYO)                                          DSPLIN
      RETURN                                                             DSPLIN
   60 WRITE(6,1111)IXN,IYN                                               DSPLIN
 1111 FORMAT(///,' DIMENSIONS EXCEEDED IN PCOPY',I5,I10,//)              DSPLIN
      STOP                                                               DSPLIN
C*                                                                       DSPLIN
   10 DO 40 I = 1,IYD                                                    DSPLIN
      IYN = TS2 + I-1                                                    DSPLIN
      IYO = T2 + I-1                                                     DSPLIN
```

```
   40 TO(TS1,IYN) = FR(T1,IYO)                                          DSPLIN
      RETURN                                                            DSPLIN
C*                                                                      DSPLIN
   20 DO 50 I = 1,IXO                                                   DSPLIN
      IXN = TS1 + I-1                                                   DSPLIN
      IXO = T1 + I-1                                                    DSPLIN
   50 TO(IXN,TS2) = FR(IXO,T2)                                          DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      REAL FUNCTION RADD*8(DEG)                                         DSPLIN
      REAL*8 DEG                                                        DSPLIN
C-----TO CONVERT RADIANS TO DEGREES AND DECIMALS OF A DEGREE            DSPLIN
C*                                                                      DSPLIN
      RADD = DEG*(45.0D00/DATAN(1.0D00))                                DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SCMULT(G,A,B,I,J,K,L)                                  DSPLIN
      DOUBLE PRECISION A(K,L),B(K,L),G                                  DSPLIN
C-----SCALAR MUTIPLICATION OF A MATRIX                                  DSPLIN
C*     THE CONSTANT TERM G ON ENTRY MUST BE IN DB                       DSPLIN
C*     THE MATRIX TO BE MULTIPLIED BY G IS A                            DSPLIN
C*     THE RESULTANT MATRIX IS B                                        DSPLIN
C*     NOTE THAT B CAN BE THE SAME MATRIX AS A                          DSPLIN
C*                                                                      DSPLIN
      DO 10 II = 1,I                                                    DSPLIN
      DO 10 IJ = 1,J                                                    DSPLIN
   10 B(II,IJ) = G*A(II,IJ)                                             DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
      SUBROUTINE SPACE(I)                                               DSPLIN
C*----TO SKIP LINES                                                     DSPLIN
C*      I  LINES TO BE SKIPPED                                          DSPLIN
C*                                                                      DSPLIN
      DO 10 II = 1,I                                                    DSPLIN
      WRITE(6,*)                                                        DSPLIN
   10 CONTINUE                                                          DSPLIN
      RETURN                                                            DSPLIN
      END                                                               DSPLIN
```

VITA

Candidate's full name:	SEE HEAN QUEK
Place and date of birth:	Melaka, Malaysia 17th August 1957
Present Address:	179-K Jalan Panjang Melaka, Malaysia.
Schools attended:	Bandar Hilir English School, Melaka, Malaysia. 1964 - 1970
	Malacca High School Melaka, Malaysia. 1970 - 1977
Post-secondary education (dates and degrees obtained):	North East London Polytechnic, London, United Kingdom. 1977 - 1980 BSc. (Hons) in Land Surveying.