ALERT PROGRAM FOR NAVSTAR GLOBAL POSITIONING SYSTEM, TRANSIT, LAGEOS, AND STARLETTE SATELLITES

S. P. MERTIKAS
D. DELIKARAOGLOU
R. SANTERRE

June 1986

TECHNICAL REPORT
NO. 85

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.

ALERT PROGRAM FOR NAVSTAR GLOBAL POSITIONING SYSTEM, TRANSIT, LAGEOS, AND STARLETTE SATELLITES

S.P. Mertikas
D. Delikaraoglou
R. Santerre

Department of Surveying Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

June 1986
Latest Reprinting February 1991

PREFACE

A satellite alert program is one which, given appropriate input descriptions of satellite ephemerides, the user's position, and the planned observing time span, can produce as output the identification, azimuth, and elevation of all satellites which are visible at each of a sequence of epochs during the observing time span.

Over the past few years, a series of alert programs has evolved in the Department of Surveying Engineering at the University of New Brunswick. The first of these was designed to predict the visibility of Global Positioning System (GPS) satellites. Later versions were extended to be able to produce alerts for Transit, Lageos, and Starlette satellites (or, for that matter, for practically any satellite for which ephemeris information is available).

At the present time, this family of alert programs consists of five members:

* ALERT which runs on the IBM 3081 mainframe computer at UNB
* GEPSAL which runs on the HP 1000 computer in the Department
* MacGEPSAL which runs on the Apple Macintosh microcomputer
* MacSAT which runs on the Apple Macintosh microcomputer
* SATRACK which runs on the Apple II microcomputer.

In general, these programs will produce alerts for any satellite for which ephemerides are provided. However, in the case of a constellation of GPS satellites, measures of the strength of the satellite constellation geometry (called "dilution of precision" or DOP factors) are also provided by the first three of the above programs. The last four of these programs have interactive input and graphic output capabilities as well.

This report describes the GPS aspects of the alert problem, gives the basic algorithms underlying the entire family of alert programs, and documents the IBM version. Chapters 1 through 3 give an overview of the GPS system, review the GPS user navigation solution, and describe the DOP factors used as accuracy measures for GPS. Chapter 4 deals with the satellite alert algorithm itself. Appendix A explains the notation used in this report. Appendix B explains the input and output formatting and parameters for the IBM version of the alert software. Appendix C contains a listing and description of the subroutines used by the IBM version of the alert software. Appendix D provides program listings and sample input and output file listings for the IBM version of the alert software.

The HP 1000 version of the alert software (GEPSAL) is documented in Technical Memorandum No. 9 of the Department of Surveying Engineering [Doucet, 1986]. The Macintosh and Apple II alert software (MacGEPSAL, MacSAT, SATRACK) are designed to be self-documenting.

This work was supported by a Natural Sciences and Engineering Research Council of Canada Strategic Grant entitled "Marine Geodesy Applications" held by Dr. David Wells. The ideas and software documented here are the result of efforts by many others in addition to the authors, among them Ken Doucet, Jan Garmulewicz, Hal Janes, Alfred Kleusberg, Richard B. Langley, See Hean Quek, and David Wells. Their collaboration and help are greatly appreciated and acknowledged.

All five versions of the software are available from:

Geodetic Research Laboratory
Department of Surveying Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, New Brunswick
Canada
E3B 5A3

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GPS OVERVIEW

The Navigation Satellite Timing and Ranging (NAVSTAR) system, also known as the Global Positioning System (GPS), is a satellite-based radionavigation system being developed by the United States Department of Defense. The GPS, which relies on range and range-rate measurements, will provide highly accurate three-dimensional position, velocity, and timing information. This new radionavigation system has arisen out of the requirements for continuous, all weather, worldwide, extremely accurate, common grid, three-dimensional navigation.

The GPS is presently scheduled to be fully operational by the late 1980s. In the meantime, six prototype satellites have been placed in orbit, with four additional prototype satellite launches planned until 1985. These satellites serve as a means of validating and establishing the operability of the system. The final operational satellite constellation, as presently planned, will consist of 18 satellites in six orbital planes, with three satellites equally spaced in each plane [Payne, 1982]. A detailed account of the system and its principles of operation can be found in Wells et al. [1982], and Payne [1982].

To use the GPS, a particular set of GPS satellites must be selected for navigation based on some criterion. This selection is aided by a computer programme computing GPS satellite "alerts". "Satellite alert" is a term which refers to the determination of the number, identity, and relevant characteristics of the satellites which are in view for an observer's site at a particular time. Visibility is an important aspect. Because of the very high frequency (L_1 = 1575 MHz; L_2 = 1227 MHz) of the GPS signals, visibility is limited to the line of sight. Therefore, proper scheduling of satellite observations requires that information be available on how many satellites are visible, for how long, and in what configuration. A satellite alert programme provides this information.

So far we have established one of the goals of a satellite alert programme: To determine those satellites which are visible, or will be visible, for an observer. When the GPS is fully operational, 4 to 7 satellites will always be visible. This sets up another goal for the programme: To search through all combinations of visible satellites to find which set of four satellites yields the best navigation performance. In the case where differential GPS techniques (two or more receivers simultaneously tracking the same satellites) are being used, a satellite alert programme plays an even more important role: To identify which satellites will be visible to all receivers, and to control proper scheduling of observations to the "best" sequence of satellites. The criterion of a "best navigation performance" is based on the geometry of the satellites and the user's location, and is represented by the so-called Geometric Dilution of Precision (GDOP) parameters.

The reason for using four satellites is that a conventional GPS user requires range measurements from four satellites. He must solve for the three positional coordinates, such as (X_j, Y_j, Z_j) or (ϕ_j, λ_j, h_j), and for the user clock bias Δt_{u_j}. In other words, for three-dimensional navigation at least four satellites are needed.

2. USER NAVIGATION SOLUTION

As mentioned previously, the GPS is a radionavigation system. Signals are emitted from the satellites and are received by a user. To obtain a position fix (X, Y, Z, Δt_u), the user must be equipped with a receiver capable of tracking at least four satellite signals simultaneously (multichannel receiver) or sequentially (single channel receiver). In the receiver, the received satellite signal is matched against an exact replica of the emitted signal in an attempt to determine the time-of-arrival, i.e., the time spent by the signal to cover the distance from the satellite to the user.

By multiplying this time by the speed of light, the user can determine his distance or range ($\tilde{\rho}_{ij}$ (t_k)) from the satellite. This distance, which is in terms of raw receiver measurements and includes atmospheric delays and user clock bias, is called a <u>pseudorange</u>. Pseudoranges will be designated by the small Greek letter $\tilde{\rho}_{ij}$. The subscript i denotes the satellites and j the receiver.

In our computations, we shall consider an earth-fixed, earth-centred reference system, as shown in Figure 2.1, characterized by the following properties:

1. The x-axis passes through the intersection of the equator and the Greenwich meridian.
2. The z-axis passes through the North Pole.
3. The y-axis completes the right-handed coordinate system.

The basic GPS measurement is pseudoranges ($\tilde{\rho}_{ij}$) but, because of the satellite motion, the system allows us to measure another quantity; integrated Doppler range differences or delta range observations. In this

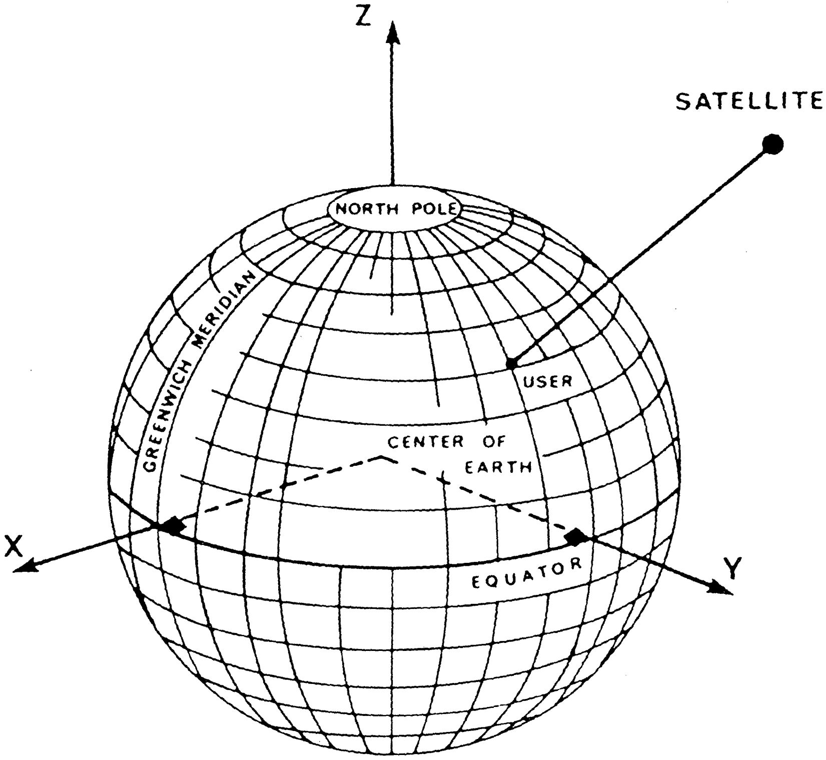


FIGURE 2.1

WGS-72 Reference System.

report, we will be dealing only with the first basic measurement of the GPS, that is, pseudoranges ($\tilde{\rho}_{ij}$). Observation equations will be developed for the determination of a position fix based on these measurements.

At the present time, the operational satellite constellation is planned to consist of 18 satellites in six orbital planes, with three satellites equally spaced in each plane, the planes to have an inclination of 55^o, and the nominal period of a satellite revolution to be 11 hr, 57 min, 57.26 sec [Payne, 1982]. This 18-satellite constellation may sometimes experience occasional outages. An outage is when a user can only obtain data from less than four satellites, and a complete three-dimensional navigation solution (X, Y, Z, Δt_u) is not possible.

For a complete three-dimensional position fix, at least four range measurements are needed. Three measurements are required for the position determination (X_j, Y_j, Z_j) and one for the user clock bias (Δt_{u_j}). The user is usually equipped with a fairly inexpensive crystal clock, which means that another unknown (Δt_{u_j}) is added to the computations.

To compute a position from satellite range data measurements, the following information is required for each measurement:

1. Position of the tracked satellites and time of signal transmission [$x_i(t_k)$, $y_i(t_k)$, $z_i(t_k)$].
2. Time of transmission of the received signal [t_{s_i}].
3. Estimates of the deterministic time delays.

The position of each tracked satellite with respect to our reference system (WGS-72) can be computed as a function of time from the six orbital elements. The most current information (taken from van Dierendonck et al. [1978]) is given as:

$\mu = 3.986\ 008 \cdot 10^{14}\ m^3/sec^2$	Universal Gravitational Constant(WGS-72)
$\omega_e = 7.292\ 115\ 147 \cdot 10^{-5}$ rad/sec	Earth's rotation rate (WGS-72)
$a = (\sqrt{a})^2$	Semi-major axis
$\eta_o = \sqrt{\mu/a^3}$	Computed mean motion
$t_k = t - t_{oe}$	Time from reference epoch
$\eta = \eta_o + \Delta n$	Corrected mean motion
$M_k = M_o + \eta t_k$	Mean anomaly
$M_k = E_k - e \sin E_k$	Kepler's equation for eccentric anomaly
$\cos V_k = (\cos E_k - e)/(1 - e \cos E_k)$ $\sin V_k = \sqrt{1-e^2} \sin E_k/(1 - e \cos E_k)$	True anomaly
$\phi_k = V_k + \omega$	Argument of latitude
$\delta u_k = C_{uc} \cos 2\phi_k + C_{us} \sin 2\phi_k$	Correction to argument of latitude (2nd harmonic perturbations)
$\delta r_k = C_{rc} \cos 2\phi_k + C_{rs} \sin 2\phi_k$	Correction to orbit radius (2nd harmonic perturbations)
$\delta i_k = C_{ic} \cos 2\phi_k + C_{is} \sin 2\phi_k$	Correction to inclination angle (2nd harmonic perturbations)
$u_k = \phi_k + \delta u_k$	Corrected argument of latitude
$r_k = a(1 - e \cos E_k) + \delta r_k$	Corrected orbit radius
$i_k = i_o + \delta i_k$	Corrected inclination
$x_k = r_k \cos u_k$ $y_k = r_k \sin u_k$	Position in orbital plane
$\Omega_k = \Omega_o + (\dot{\Omega} - \omega_e)t_k - \omega_e t_{oe}$	Corrected longitude of ascending node
$X_k = x_k \cos\Omega_k - y_k \cos i_k \sin\Omega_k$ $Y_k = x_k \sin\Omega_k + y_k \cos i_k \cos\Omega_k$ $Z_k = y_k \sin i_k$	Earth fixed coordinates

The above satellite ephemeris, along with system time, satellite clock behaviour data, and transmitter status information, is supplied by means of the GPS navigation message [van Dierendonck, 1978].

Let us consider the jth ground station and the ith satellite. The position vector of the ground station is

$$\underline{R}_J = [X_J,\ Y_J,\ Z_J]^T \quad .$$

The position vector and Cartesian coordinates of the ith satellite, at some epoch $t_k(\tau)$ (a function of the conventional GPS time) are

$$\underline{r}_i(t_k(\tau)) = [x_i(t_k),\ y_i(t_k),\ z_i(t_k)]^T \quad .$$

The Cartesian coordinates of the geometric range vector between the ith satellite and the jth ground station are

$$\underline{\rho}_{ij}(t_k) = [\xi_{ij}(t_k),\ \eta_{ij}(t_k),\ \zeta_{ij}(t_k)]^T \quad .$$

The length of $\underline{\rho}_{ij}$ is denoted by ρ_{ij}. From Figure 2.2, the geometric range vector is

$$\underline{\rho}_{ij}(t_k) = \underline{r}_i(t_k) - \underline{R}_j \quad . \qquad (2.1)$$

The pseudorange $\tilde{\rho}_{ij}$, which a receiver can measure, is defined as

$$\tilde{\rho}_{ij}(t_k) = \rho_{ij}(t_k) + c(\Delta t_{u_j} - \Delta t_{s_i}) + c\cdot\Delta t_{A_i} = \rho_{ij}(t_k) + \Delta\rho_{RCV} - \Delta\rho_{SAT} + \Delta\rho_{ATM} \qquad (2.2)$$

where

$\tilde{\rho}_{ij}$ = pseudorange (metres)

ρ_{ij} = geometric (true) range (metres)

c = speed of light (metres/second)

Δt_{u_j} = user clock time bias (seconds)

Δt_{s_i} = satellite i clock time bias (seconds)

$c\Delta t_{A_i}$ = atmospheric delays (ionospheric, tropospheric)(metres).

The atmospheric delays $c\Delta t_{A_i}$ are introduced by propagation error due to the atmosphere, specifically the ionospheric and tropospheric delay.

Ionospheric delays are estimated by the user (j) by measuring pseudoranges $\tilde{\rho}_{ij}(t_k)$ at two different frequencies (L_1=1575 MHz;L_2=1227 MHz).

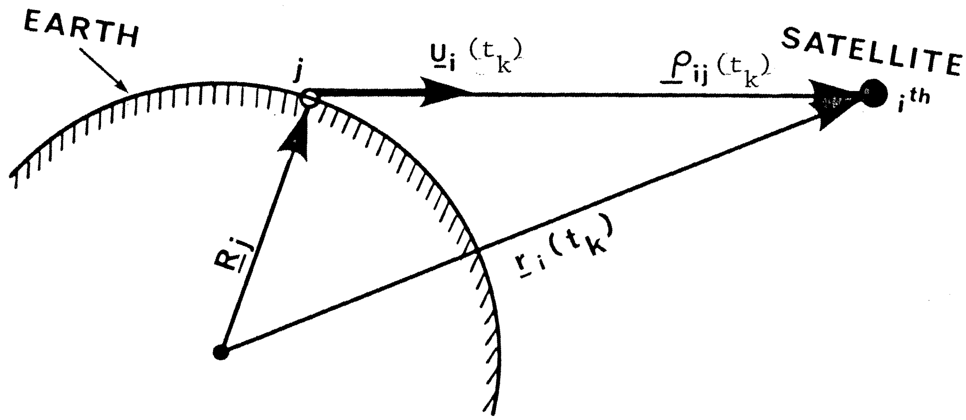
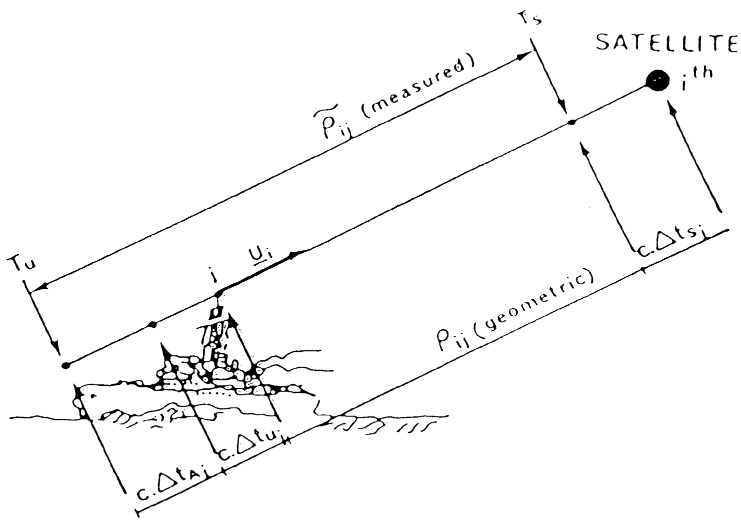


FIGURE 2.2
Navigation Solution Geometry



$c \cdot \Delta t_{A_j}$ = atmospheric delays (metres)

$c \cdot \Delta t_{u_j}$ = user clock time bias (metres)

$c \cdot \Delta t_{S_i}$ = satellite i clock time bias (metres)

ρ_{ij} = geometric range (metres)

$c(T_u - T_s) = \tilde{\rho}_{ij}$ = pseudorange (metres)

FIGURE 2.3

Pseudorange Observation.

This is done because the ionosphere has a delay effect which is approximately inversely proportional to the square of the frequency ($\sim 1/f^2$) [van Dierendonck et al. 1978]. For single channel receivers, ionospheric delays must be estimated from a mathematical model. Coefficients of a polynomial model are provided by means of the navigation satellite message.

Tropospheric delays are frequency independent. Estimation models for the troposphere are based on geometry and altitude. Approximation models for the estimation of ionospheric and tropospheric delays are given in Ward [1981].

The satellite clock time bias Δt_{s_i} is again provided by the satellite message, whereas the user clock time bias Δt_{u_j} is considered unknown and is solved for through the navigation solution.

The mathematical model $F(\underline{X}, \underline{L}) = 0$ for an observation of pseudorange is in the form

$$F(\underline{X}, \underline{L}) = \rho_{ij} + c(\Delta t_{u_j} - \Delta t_{s_i}) + c\Delta t_{A_i} - \tilde{\rho}_{ij} = 0 \quad , \tag{2.3}$$

where i designates the satellite and j the ground station.

The geometric (true) range at some epoch t_k is

$$\rho_{ij}(t_k) = \sqrt{\{X_j - x_i(t_k)\}^2 + \{Y_j - y_i(t_k)\}^2 + \{Z_j - z_i(t_k)\}^2} \quad . \tag{2.4}$$

Substituting the above in the general mathematical model of the pseudorange:

$$F(\underline{X},\underline{L}) = \sqrt{\{X_j - x_i(t_k)\}^2 + \{Y_j - y_i(t_k)\}^2 + \{Z_j - z_i(t_k)\}^2}$$
$$+ c(\Delta t_{u_j} - \Delta t_{s_i}) + c\Delta t_{A_i} - \tilde{\rho}_{ij} = 0 \quad . \tag{2.5}$$

Expanding the above equation of pseudorange observation in a Taylor series about an initial approximate user's position and clock bias

$$\underline{X}_j^{(o)} = [X_j^{(o)} \; Y_j^{(o)} \; Z_j^{(o)} \; \Delta t_{u_j}^{(o)}]^T$$

and using the measured values of the observation vector

$$\underline{L}^{(o)} = [\tilde{\rho}_{1j}^{(o)} \; \tilde{\rho}_{2j}^{(o)} \; \tilde{\rho}_{3j}^{(o)} \; \tilde{\rho}_{4j}^{(o)} \; \ldots]^T \quad ,$$

we get

$$\{\frac{\partial F}{\partial \underline{\chi}_j}\Big|_{\underline{\chi}_j^{(o)}}\}(\underline{\chi}_j - \underline{\chi}_j^{(o)}) + (-I)(\tilde{\underline{\rho}}_{ij} - \tilde{\underline{\rho}}_{ij}^{(o)}) + F(\underline{\chi}_j^{(o)}, \tilde{\underline{\rho}}_{ij}^{(o)}) = 0 \qquad (2.6)$$

In our familiar notation of surveying, the above can be written as

$$A \cdot \underline{\delta\chi} + B \cdot \underline{V} + \underline{W} = \underline{0} \quad , \qquad (2.7)$$

where

$$A = \{\frac{\partial F}{\partial \underline{\chi}_j}\Big|_{\underline{\chi}_j^{(o)}}\} = \text{design matrix}$$

$$\underline{\delta\chi} = \underline{\chi}_j - \underline{\chi}_j^{(o)} = \text{correction vector}$$

$$B = -I = \{\frac{\partial F}{\partial \underline{\rho}_{ij}}\Big|_{\underline{\rho}_{ij}^{(o)}}\} = \{\frac{\partial F}{\partial \underline{L}}\Big|_{\underline{L}^{(o)}}\} = \text{design matrix}$$

$$\underline{V} = \underline{L} - \underline{L}^{(o)} = \text{residual vector;} \quad \tilde{\rho}_{ij} = \tilde{\rho}_{ij}^{(o)} + v_i$$

$$\underline{W} = F(\underline{\chi}_j^{(o)}, \tilde{\underline{\rho}}_{ij}^{(o)}) = \text{misclosure vector.}$$

The above equation (2.7) is the linearized equation which relates pseudorange measurements to the desired user navigation information, either $[X_j, Y_j, Z_j]$ or $[\phi_j, \lambda_j, h_j]$, as well as the user clock bias Δt_{u_j}.

When four satellites are available (i = 1, 2, 3, 4), the linearized equations can be written as

$$A \cdot \underline{\delta\chi} + B \circ \underline{V} + \underline{W} = \underline{0} \quad , \qquad (2.8)$$

where

$$A = \text{design matrix} = \left\{ \frac{\partial F}{\partial \underline{X}_j} \bigg|_{\underline{X}_j^{(o)}} \right\} = \begin{vmatrix} \frac{X_j^{(o)} - x_1(t_k)}{\rho_{1j}^{(o)}} & \frac{Y_j^{(o)} - y_1(t_k)}{\rho_{1j}^{(o)}} & \frac{Z_j^{(o)} - z_1(t_k)}{\rho_{1j}^{(o)}} & c \\ \frac{X_j^{(o)} - x_2(t_k)}{\rho_{2j}^{(o)}} & \frac{Y_j^{(o)} - y_2(t_k)}{\rho_{2j}^{(o)}} & \frac{Z_j^{(o)} - z_2(t_k)}{\rho_{2j}^{(o)}} & c \\ \frac{X_j^{(o)} - x_3(t_k)}{\rho_{3j}^{(o)}} & \frac{Y_j^{(o)} - y_3(t_k)}{\rho_{3j}^{(o)}} & \frac{Z_j^{(o)} - z_3(t_k)}{\rho_{3j}^{(o)}} & c \\ \frac{X_j^{(o)} - x_4(t_k)}{\rho_{4j}^{(o)}} & \frac{Y_j^{(o)} - y_4(t_k)}{\rho_{4j}^{(o)}} & \frac{Z_j^{(o)} - z_4(t_k)}{\rho_{4j}^{(o)}} & c \end{vmatrix} \quad (2.9)$$

$$\underline{\delta X} = \text{correction vector} = \underline{X}_j - \underline{X}_j^{(o)} = \begin{vmatrix} X_j - X_j^{(o)} \\ Y_j - Y_j^{(o)} \\ Z_j - Z_j^{(o)} \\ \Delta t_{u_j} - \Delta t_{u_j}^{(o)} \end{vmatrix} \quad (2.10)$$

$$\underline{V} = \text{residual vector} = \underline{L} - \underline{L}^{(o)} = \begin{vmatrix} \tilde{\rho}_{1j} - \tilde{\rho}_{1j}^{(o)} \\ \tilde{\rho}_{2j} - \tilde{\rho}_{2j}^{(o)} \\ \tilde{\rho}_{3j} - \tilde{\rho}_{3j}^{(o)} \\ \tilde{\rho}_{4j} - \tilde{\rho}_{4j}^{(o)} \end{vmatrix} \quad (2.11)$$

$$B = \text{design matrix} = \{\frac{\partial F}{\partial \underline{L}}\Big|_{\underline{L}}(o)\} = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix} = -I \qquad (2.12)$$

$$\underline{W} = \text{misclosure vector} = F(\underline{X}^{(o)}, \underline{L}^{(o)}) = \begin{vmatrix} \rho_{1j}^{(o)} + c\cdot(\Delta t_{u_j}^{(o)} - \Delta t_{s_1}) + c\cdot\Delta t_{A_1} - \tilde{\rho}_{1j}^{(o)} \\ \rho_{2j}^{(o)} + c\cdot(\Delta t_{u_j}^{(o)} - \Delta t_{s_2}) + c\cdot\Delta t_{A_2} - \tilde{\rho}_{2j}^{(o)} \\ \rho_{3j}^{(o)} + c\cdot(\Delta t_{u_j}^{(o)} - \Delta t_{s_3}) + c\cdot\Delta t_{A_3} - \tilde{\rho}_{3j}^{(o)} \\ \rho_{4j}^{(o)} + c\cdot(\Delta t_{u_j}^{(o)} - \Delta t_{s_4}) + c\cdot\Delta t_{A_4} - \tilde{\rho}_{4j}^{(o)} \end{vmatrix} \qquad (2.13)$$

where $\rho_{ij}^{(o)}(t_k) = \sqrt{(X_j - x_i(t_k^2)) + (Y_j - y_i(t_k^2)) + (Z_j - z_i(t_k^2))}$. (2.14)

The quantities to be computed (δX_j, δY_j, δZ_j, $\delta \Delta t_{u_j}$) are corrections that the user will make to his current estimate of position ($X_j^{(o)}$, $Y_j^{(o)}$, $Z_j^{(o)}$) and his clock bias $\Delta t_{u_j}^{(o)}$.

It should be noted that the coefficients in the first three columns of the design matrix A are the negative direction cosines of the line of sight from the user (j) to the satellite (i). For all four equations, the coefficient of Δt_{u_j} is the speed of light c.

Let $\underline{U}_i(t_k) = (u_i(t_k), v_i(t_k), w_i(t_k))^T$ be the unit vector from the user position (j) to the ith satellite. u_i, v_i, and w_i are the x, y, and z components of this unit vector $\underline{U}_i$, as shown in Figures 2.4 and 2.5. It is known from analytical geometry that the components of the unit vector $\underline{U}_i$

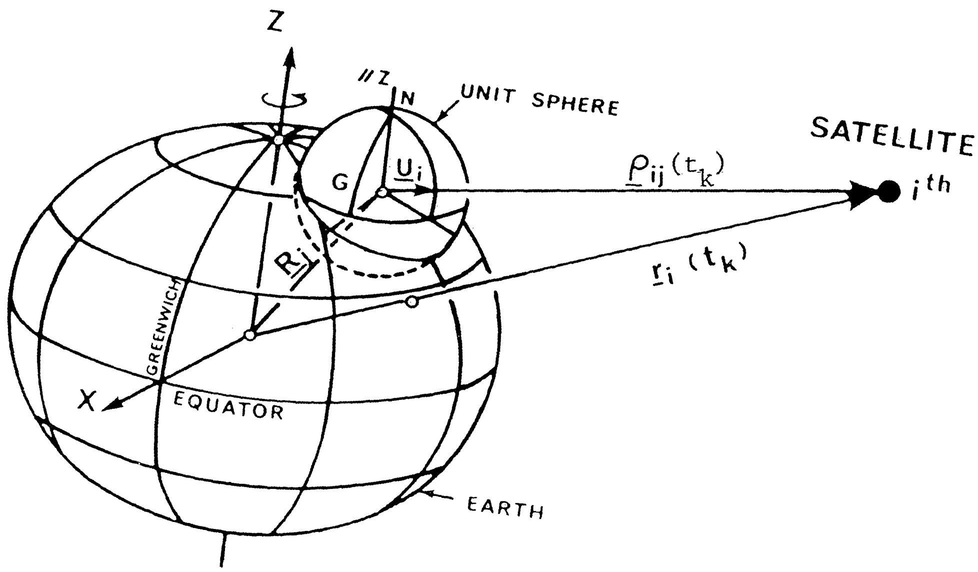


FIGURE 2, 4
Satellite Geometry

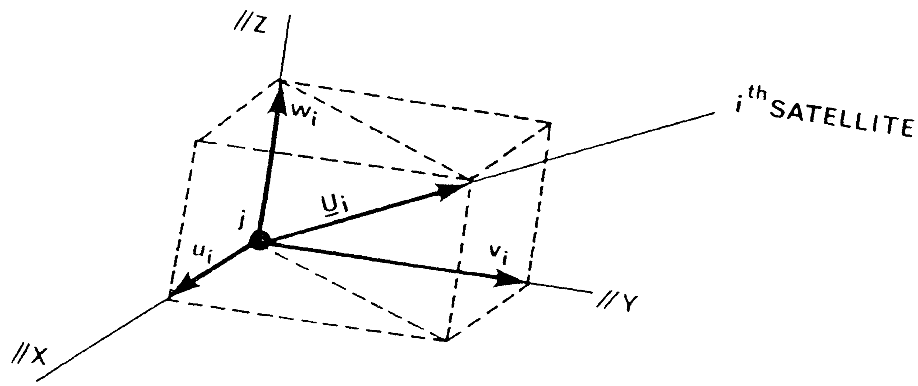


FIGURE 2.5
Unit Vectors $\underline{U}_i$.

are:

$$\left(\frac{x_i(t_k) - X_j}{\rho_{ij}}, \frac{y_i(t_k) - Y_j}{\rho_{ij}}, \frac{z_i(t_k) - Z_j}{\rho_{ij}}\right)^T = (u_i(t_k), v_i(t_k), w_i(t_k))^T .$$

Therefore the design matrix

$$A = \left\{\frac{\partial F}{\partial \underline{\chi}_j}\Big|_{\underline{\chi}_j}(o)\right\}$$

can be expressed in an equivalent form with direction cosines as

$$\begin{vmatrix} -u_1 & -v_1 & -w_1 & c \\ -u_2 & -v_2 & -w_2 & c \\ -u_3 & -v_3 & -w_3 & c \\ -u_4 & -v_4 & -w_4 & c \end{vmatrix} = A = \left\{\frac{\partial F}{\partial \underline{\chi}_j}\Big|_{\underline{\chi}_j^{(o)}}\right\} . \qquad (2.15)$$

Assuming that the weight matrix of the observations is known, an estimate of the correction vector $\underline{\delta\chi} = \underline{\chi}_j - \underline{\chi}_j^{(o)}$, based on the least squares principle, is given by:

$$\underline{\delta\chi} = \underline{\chi}_j - \underline{\chi}_j^{(o)} = (A^T PA)^{-1} A^T P\underline{W} . \qquad (2.16)$$

The final solution vector is

$$\underline{\chi}_j = \underline{\chi}_j^{(o)} + \underline{\delta\chi} = \underline{\chi}_j^{(o)} + (A^T PA)^{-1} A^T P\underline{W} . \qquad (2.17)$$

It is obvious that the above process is iterative and this final vector $\underline{\chi}_j$ can be used as a new approximation for another iteration. The number of iterations depends on an error criterion. Usually three iterations are adequate.

When a solution in latitude (ϕ), longitude (λ), and height (h) of the user position is required, either a simple conversion of the (X_j, Y_j, Z_j, Δt_{u_j}) is applied into a (ϕ_j, λ_j, h_j, Δt_{u_j}) solution after the above procedure is performed, or the design matrix should have rows of four elements such that

$$A_{\ell} = [\frac{\partial F}{\partial \phi} \;\; \frac{\partial F}{\partial \lambda} \;\; \frac{\partial F}{\partial h} \;\; \frac{\partial F}{\partial \Delta t_u}] \qquad (\ell = 1,2,3,\ldots) \quad . \tag{2.18}$$

For marine navigation, we can consider our height as known (usually it is taken as equal to 10 metres), and determine only two coordinates of position and the user clock bias. In such a case, the sought receiver solution would be

$$\underline{X}_j = \begin{vmatrix} \phi_j \\ \lambda_j \\ \Delta t_{u_j} \end{vmatrix} \quad , \tag{2.19}$$

and the design matrix $A = \{\frac{\partial F}{\partial \underline{X}_j}\Big|_{\underline{X}_j^{(o)}}\}$

$$A = \begin{vmatrix} \frac{\partial F_1}{\partial \phi} & \frac{\partial F_1}{\partial \lambda} & \frac{\partial F_1}{\partial \Delta t_u} \\ \frac{\partial F_2}{\partial \phi} & \frac{\partial F_2}{\partial \lambda} & \frac{\partial F_2}{\partial \Delta t_u} \\ \frac{\partial F_3}{\partial \phi} & \frac{\partial F_3}{\partial \lambda} & \frac{\partial F_3}{\partial \Delta t_u} \\ \frac{\partial F_4}{\partial \phi} & \frac{\partial F_4}{\partial \lambda} & \frac{\partial F_4}{\partial \Delta t_u} \end{vmatrix} \quad . \tag{2.20}$$

The partial derivatives of the general mathematical model F with respect to ϕ, λ, and Δt_u are

$$\frac{\partial F}{\partial \phi} = \frac{\partial F}{\partial X} \cdot \frac{\partial X}{\partial \phi} + \frac{\partial F}{\partial Y} \cdot \frac{\partial Y}{\partial \Phi} + \frac{\partial F}{\partial Z} \cdot \frac{\partial Z}{\partial \phi}$$

$$\frac{\partial F}{\partial \lambda} = \frac{\partial F}{\partial X} \cdot \frac{\partial X}{\partial \phi} + \frac{\partial F}{\partial Y} \cdot \frac{\partial Y}{\partial \lambda} + \frac{\partial F}{\partial Z} \cdot \frac{\partial Z}{\partial \lambda}$$

$$\frac{\partial F}{\partial \Delta t_u} = c$$

or in a matrix notation

$$\begin{vmatrix} \frac{\partial F}{\partial \phi} \\ \frac{\partial F}{\partial \lambda} \\ \frac{\partial F}{\partial \Delta t_u} \end{vmatrix} = \begin{vmatrix} \frac{\partial X}{\partial \phi} & \frac{\partial Y}{\partial \phi} & \frac{\partial Z}{\partial \phi} & 0 \\ \frac{\partial X}{\partial \lambda} & \frac{\partial Y}{\partial \lambda} & \frac{\partial Z}{\partial \lambda} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \frac{\partial F}{\partial X} \\ \frac{\partial F}{\partial Y} \\ \frac{\partial F}{\partial Z} \\ c \end{vmatrix} \quad . \tag{2.21}$$

The partial derivatives involved are given in McCaskill et al. [1976] as

$$\frac{\partial X}{\partial \phi} = [\frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} + h] \sin\phi \cdot \cos\lambda + \frac{a \cdot e^2 \cdot \sin\phi \cdot \cos^2\phi \cos\lambda}{(1-e^2 \sin^2 \phi)^{3/2}} \tag{2.22}$$

$$\frac{\partial Y}{\partial \phi} = -[\frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} + h] \sin\phi \cdot \sin\lambda + \frac{a \cdot e^2 \cdot \sin\phi \cdot \cos^2\phi \cdot \sin\lambda}{(1-e^2 \sin^2 \phi)^{3/2}} \tag{2.23}$$

$$\frac{\partial Z}{\partial \phi} = [\frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} + h - \frac{a \cdot e^2}{(1-e^2 \sin^2 \phi)^{1/2}}] \cos\phi + \frac{a \cdot e^2 (1-e^2) \sin^2\phi \cdot \cos\phi}{(1-e^2 \sin^2 \phi)^{3/2}} \tag{2.24}$$

$$\frac{\partial X}{\partial \lambda} = [\frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} + h] \cos\phi \cdot \sin\lambda \tag{2.25}$$

$$\frac{\partial Y}{\partial \lambda} = [\frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} + h] \cos\phi \cdot \cos\lambda \tag{2.26}$$

$$\frac{\partial Z}{\partial \lambda} = 0 \quad . \tag{2.27}$$

3. ACCURACY MEASURES FOR GPS PERFORMANCE

The accuracy measures for the GPS performance are much simpler than the conventional ones associated with the error ellipse and the ellipsoid of constant probability. The use of those conventional measures is complicated by the orientation of the axes and the propagation of the elliptical errors. Instead, a circular form is employed which is easier to use and understand.

This section is devoted to the establishment of a meaningful accuracy statement for the GPS performance for a uniform interpretation. User accuracy is dependent upon various factors; however there are two primary ones: The range error, and geometry. The former is expressed by the User Equivalent Range Error (UERE), and the latter by the Geometric Dilution of Precision (GDOP).

UERE is based on the assumption that there is no correlation between satellite measurements. It represents the combined accuracy parameter of satellite measurements and reflects the total error contribution of the GPS system. UERE involves "system" errors, such as uncertainties of the ephemeris data, propagation errors, clock errors, etc. In other words, each pseudorange observation toward a specific satellite is associated with an observed range error, known as UERE.

The use of the GDOP value was originally developed in LORAN navigation systems [Swanson, 1978]. The GDOP value is a measure of the satellite geometry. It is a quantity which is used extensively in determining the information content due to satellite geometry and results in a measure of the overall geometrical strength to the solution. It provides a method of quantitatively determining whether a particular satellite geometry is good or bad.

It was found before that an estimate of the correction vector is given by equation (2.16) as

$$\underline{\delta\chi} = \underline{\chi}_j - \underline{\chi}_j^{(o)} = (A^T PA)^{-1} A^T P\underline{W} \quad .$$

The cofactor matrix [Vanicek and Krakiwsky, 1982] of the estimate $\underline{\delta\chi}$ ($\delta\phi$, $\delta\lambda$, δh, $\delta\Delta t_u$) is given by

$$Q_{\underline{\delta\chi}} = (A^T PA)^{-1} = \begin{vmatrix} q^2_\phi & q_{\phi\lambda} & q_{\phi h} & q_{\phi\Delta t_u} \\ q_{\phi\lambda} & q^2_\lambda & q_{\lambda h} & q_{\lambda\Delta t_u} \\ q_{h\phi} & q_{h\lambda} & q^2_h & q_{h\Delta t_u} \\ q_{\Delta t_u\phi} & q_{\Delta t_u\lambda} & q_{\Delta t_u h} & q^2_{\Delta t_u} \end{vmatrix} \quad . \qquad (3.1)$$

Geometric Dilution of Precision is defined as the square root of the trace of the above cofactor matrix after setting the weight matrix P equal to the identity matrix, that is

$$GDOP = \sqrt{\text{trace } (A^T A)^{-1}} = \sqrt{q^2_\phi + q^2_\lambda + q^2_h + q^2_{\Delta t_u}} \quad . \qquad (3.2)$$

Other quantities of interest, along with the Geometric Dilution of Precision, are the horizontal, the vertical, the positional, the time and the horizontal time dilution of precision defined as follows:

$$\begin{aligned} HDOP &= \sqrt{q^2_\phi + q^2_\lambda} \\ VDOP &= \sqrt{q^2_h} = q_h \\ PDOP &= \sqrt{q^2_\phi + q^2_\lambda + q^2_h} \\ TDOP &= q_{\Delta t_u} \quad . \\ HTDOP &= \sqrt{q^2_\phi + q^2_\lambda + q^2_{\Delta t_u}} \end{aligned} \qquad (3.3)$$

For a complete three-dimensional position fix (ϕ, λ, h, Δt_u) the covariance matrix of the estimate $\underline{\delta\chi}$ is

$$C_{\underline{\delta\chi}} = \sigma_o^2 Q_{\underline{\delta\chi}} = \begin{vmatrix} \sigma_\phi^2 & \sigma_{\phi\lambda} & \sigma_{\phi h} & \sigma_{\phi\Delta t_u} \\ \sigma_{\lambda\phi} & \sigma_\lambda^2 & \sigma_{\lambda h} & \sigma_{\lambda\Delta t_u} \\ \sigma_{h\phi} & \sigma_{h\lambda} & \sigma_h^2 & \sigma_{h\Delta t_u} \\ \sigma_{\Delta t_u\phi} & \sigma_{\Delta t_u\lambda} & \sigma_{\Delta t_u h} & \sigma_{\Delta t_u}^2 \end{vmatrix} , \qquad (3.4)$$

where σ_o^2 is the variance factor [Vanicek and Krakiwsky, 1982].

By setting the weight matrix P equal to the identity matrix, equation (3.4) gives

$$C_{\underline{\delta\chi}} = \sigma_{range}^2 \, Q_{\underline{\delta\chi}} = (\text{UERE}) \, Q_{\underline{\delta\chi}} \quad . \qquad (3.5)$$

From the above relationship an approximate measure in the total user error would be:

$$\text{user error} = \sqrt{\sigma_\phi^2 + \sigma_\lambda^2 + \sigma_h^2 + \sigma_{\Delta t_u}^2} = (\text{UERE})\sqrt{q_\phi^2 + q_\lambda^2 + q_h^2 + q_{\Delta t_u}^2} \; . \qquad (3.6)$$

The product of the DOP factors by an estimate in the range measurements (σ_{range} = UERE) results in a user error such that

$$\text{user error} = (\text{UERE}) \, (\text{GDOP}) \quad . \qquad (3.7)$$

The same is true for the other DOP factors. For example, a PDOP value of 2.5 and a UERE of $\pm$ 4 m (1σ) would result in a user position error (1σ) of

(PDOP) $\times$ (UERE) = 2.5($\pm$ 4 m) = $\pm$ 10 metres.

It is mentioned in Ward [1981] that a PDOP value of 3 or less is expected in the full 18-satellite constellation.

Geometric dilution of precision values can be described as a measure of the navigator's position uncertainty per unit of measurement noise. It has been conceded that GDOP values are statistically distributed in a non-Gaussian fashion [Jorgensen, 1980].

So far, GDOP has been derived in an analytical way. Another way for the determination of GDOP is based on the computation of the volume of a special tetrahedron formed by the satellites and the user's location.

Let $\underline{\rho}_{ij}((t_k)$ be the geometric range vector, $\underline{R}_j$ the position vector of the jth user, and $\underline{r}_i(t_k)$ the position vector of the ith satellite, as shown in Figure 3.1.

The magnitude of the cross-product of the pair $\underline{R}_j$ and $\underline{\rho}_{ij}$ is defined as follows:

$$|\underline{R}_j \times \rho_i| = |\underline{R}_j||\underline{\rho}_{ij}| \sin(90^\circ + E) \quad , \tag{3.8}$$

whereas the dot product for the same vectors is

$$\underline{R}_j \cdot \underline{\rho}_{ij} = |\underline{R}_j||\rho_{ij}| \cos(90^\circ + E) \quad . \tag{3.9}$$

Dividing (3.9) by (3.8), we obtain:

$$\frac{\cos(90^\circ + E)}{\sin(90^\circ + E)} = \frac{-\sin E}{\cos E} = \frac{-\dfrac{\underline{R}_j \cdot \underline{\rho}_{ij}}{|\underline{R}_j|\;|\underline{\rho}_{ij}|}}{\dfrac{|\underline{R}_j \times \underline{\rho}_{ij}|}{|\underline{R}_j|\;|\underline{\rho}_{ij}|}} \quad . \tag{3.10}$$

Therefore the elevation angle E of the ith satellite can be obtained by

$$\tan E = \frac{-\dfrac{\underline{R}_j \cdot \underline{\rho}_{ij}}{|\underline{R}_j|\;|\underline{\rho}_{ij}|}}{\dfrac{|\underline{R}_j \times \underline{\rho}_{ij}|}{|\underline{R}_j|\;|\underline{\rho}_{ij}|}} = -\frac{\underline{R}_j \cdot \underline{\rho}_{ij}}{|\underline{R}_j \times \underline{\rho}_{ij}|} \tag{3.11}$$

An allowable elevation angle for the determination of whether a satellite is considered visible is

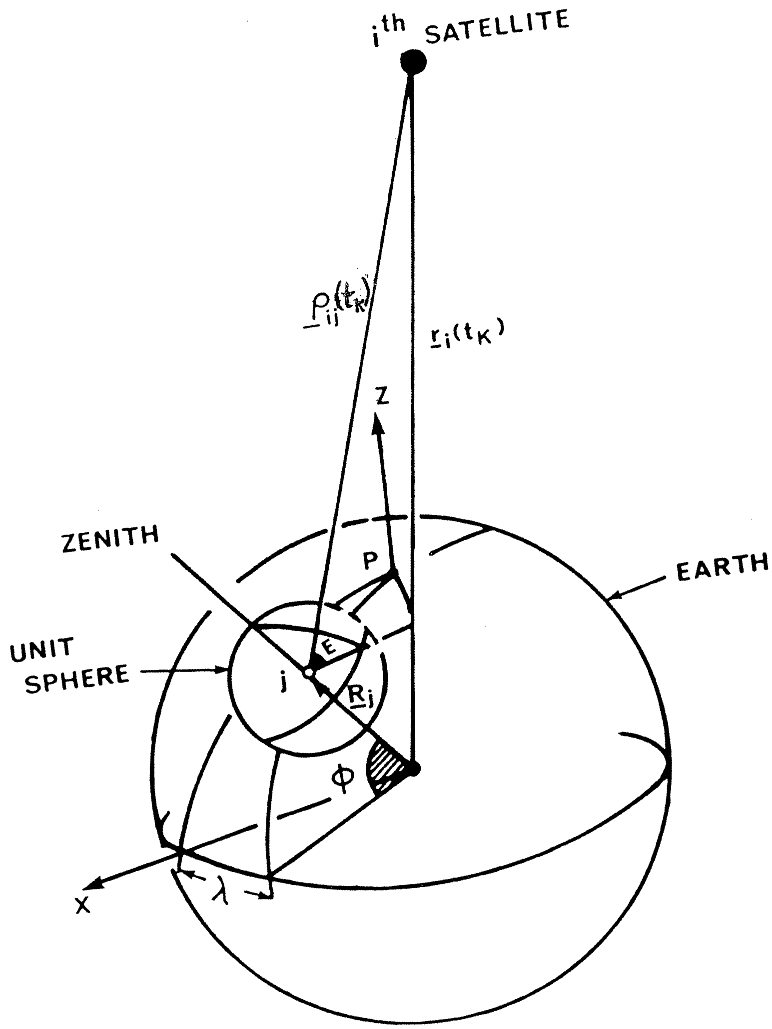


FIGURE 3.1
Elevation Angle E of the ith Satellite.

$$E \geq 5^\circ \quad . \tag{3.12}$$

Therefore, candidate satellites to be considered for visibility are those whose elevation angle E is greater or equal to 5°. Any satellite with an elevation angle of less than 5° is masked out by terrain, antenna limitations, foliage, obstructions, etc. Based on the criterion of (3.12), one can determine the number of visible satellites for a particular user (j) and time (t_k).

Let $\underline{U}_i$ be the unit vector from the user (j) to the ith visible satellite, as shown in Figure 3.2. When the full 18 satellites are in operation, four to seven satellites will be visible, on a continuous basis, at any site on the globe [USDOD, 1982]. All unit vectors $\underline{U}_i$ are centred at the user's location (j) and enclosed within a unit sphere. If we calculate all the combinations of unit vectors $\underline{U}_i$ of four satellites, we end up with a set of four unit vectors each time. It can be seen from Figure (3.3) that a special tetrahedron (e.g., 1-2-3-4) is formed by those four unit vectors.

Variability of satellite geometry depends on the orientation of the four satellite positions available. This is, in turn, a function of the user's location (j) and time (t_k) because of satellite motion and earth rotation. It has been shown that the GDOP value is inversely proportional to the volume of this special tetrahedron (1-2-3-4) [Bogen, 1974]. Hence, the largest volume yields the smallest value of GDOP and vise versa.

Determination of the maximum volume of a tetrahedron among all other volumes formed by all the other combinations of four satellites also implies the determination of those satellites with the best navigation performance. The best navigation performance relies on the geometry of the four satellites and the smallest value of GDOP.

The volume (V) of the tetrahedron (1-2-3-4) can be computed using the scalar triple product

$$V = 1/6 \; \underline{C}(\underline{A} \times \underline{B}) \quad . \tag{3.13}$$

The previous account takes into consideration geometrical aspects related to satellite geometry and the user's location, which has as a final goal the selection of satellites with the best navigation performance. It can seen that the geometrical interpretation is easier to understand and visualize.

The minimum number of observations constitutes the necessary and sufficient elements for a unique set of estimates for the solution. Any additional observations, which are said to be redundant with respect to the

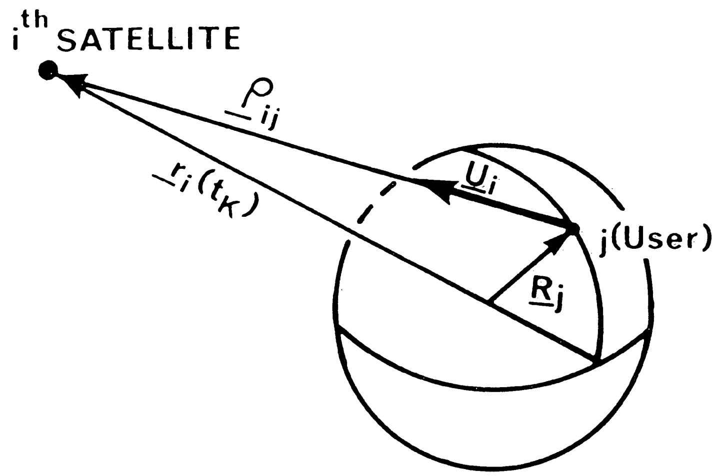


FIGURE 3.2

Unit Vector $\underline{U}_i$ to ith Satellite.

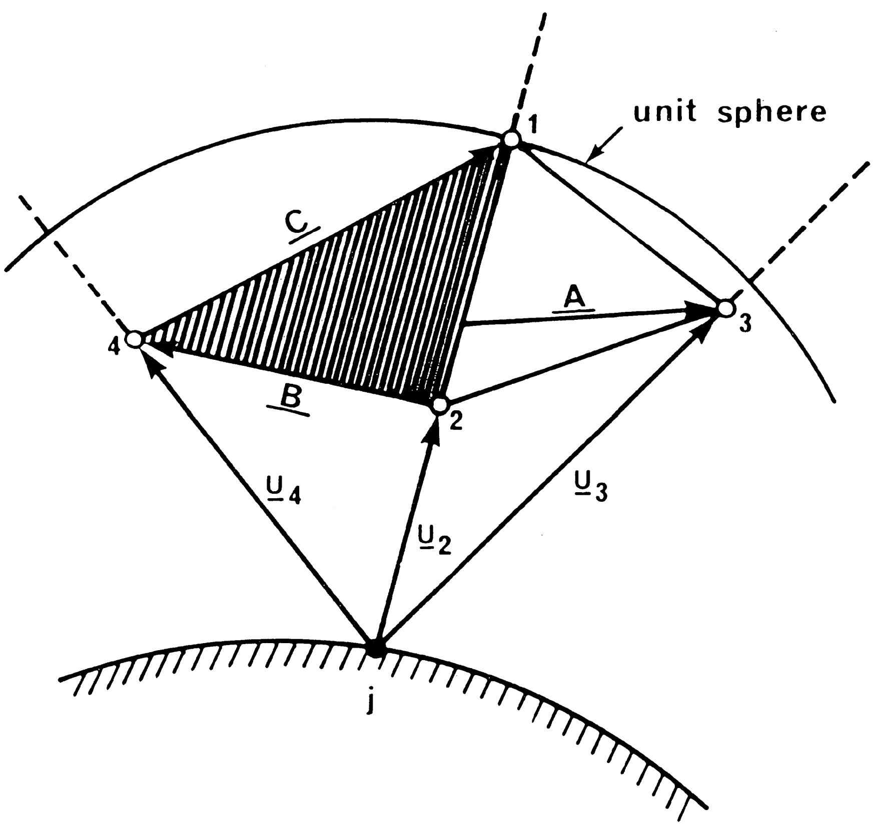


FIGURE 3.3

Tetrahedron Formed by Four Unit Vectors $\underline{U}_i$.

model (four-parameter, three-parameter, two-parameter solution), should always be taken into consideration for a more precise and reliable solution [Mikhail, 1976; Vanicek and Krakiwsky, 1982]. In this case, the corresponding GDOP value will not only incorporate four satellites but all those used for the solution (multi-dimensional GDOP).

4. ALERT ALGORITHM

ALERT primarily determines the number of visible satellites from which a program selection is made which yields the "best" navigation performance according to a set geometric criteria. For GPS, the full 18-satellite constellation is used. However, as previously mentioned, the capabilities of the program have been expanded beyond that of computing just GPS alerts. A variety of satellites can now be tracked and their alerts computed given the appropriate ephemeral data. The required alert output information is user specified, selected from a variety of three optional data types.

The basic input to the program is as follows:

1. Type of satellite one wishes to track (e.g., GPS, Transit, LAGEOS, STARLETTE).
2. Type of ephemeral data to be used:
   - (a) NNSS nodal crossing data
   - (b) Keplerian parameters
   - (c) GPS theoretical or tracked ephemeris
   - (d) NASA prediction bulletin (GAST used)
   - (e) NASA prediction bulletin (GAST not used)
3. Required alert output data:
   - (a) simple approach providing elevation and azimuth of satellites
   - (b) elevation, azimuth, and dilution of precision for the "best" four satellites
   - (c) elevation, azimuth, and dilution of precision for all satellites tracked.
4. The position of the user and the time interval desired for alert computation.

5. The time increment used for output (in minutes).

The output of the program basically consists of the following: user's location; minimal satellite elevation (e.g., 5 degrees); beginning and end date of the alerts; hour and minute of the alerts; Z-count (for GPS satellites only); identity, azimuth, and elevation of all visible satellites; and the dilution of precision. A visibility summary is also produced but only for GPS satellites.

For a more detailed summary and explanation of the user options, input formatting, card sequence, and the available type of output data, consult Appendix B of this report.

The algorithms used by the program to calculate the satellite coordinates in the earth-fixed system (see Chapter 2) are obviously dependent on the type of ephemeral information used. Should the ephemeral information supplied be of the theoretical or tracked GPS ephemeris type, Chapter 2 of this report can be consulted for a discussion on the computational algorithm used to produce the satellite coordinates. However, should other user ephemeral options be used, the following algorithms would be subsequently utilized by the program.

ALGORITHM TO CALCULATE THE SATELLITE COORDINATES IN THE EARTH FIXED SYSTEM

(A) NNSS nodal crossing data (IPARA2=1)

$t_k = t - t_o$

t : alert time

t_o: time of ascending node

$\lambda\Omega_k = \lambda\Omega_o - (\omega e - \dot{\Omega})t_k$

$\omega e - \dot{\Omega} = \frac{\text{westward motion}}{\text{nodal period}}$

$\lambda\Omega$: longitude (east) of ascending node

$\phi_k = V_k + \omega = (n + \dot{\omega})t_k$

$n + \dot{\omega} = \frac{2\pi}{\text{nodal period}}$

$i = 90^o$; $e = 0$; $r = 7456$ km

$x_k = r \cos \phi_k$

$y_k = r \sin \phi_k$

$X_k = x_k \cos \lambda\Omega_k$

$Y_k = x_k \sin \lambda\Omega_k$

$Z_k = y_k$

(B) NNSS Keplerian Parameters or mean orbit by tracking (IPARA2=2)

$t_k = t - t_o$

t : alert time

t_o: time of perigee

$\lambda\Omega_o = \alpha\Omega_o - GAST_o$

$\lambda\Omega_k = \lambda\Omega_o - (\omega e - \dot{\Omega})\ t_k$

$\lambda\Omega$: longitude (east) of ascending node

$\alpha\Omega_o$: right ascension of ascending node

$M_o = E_o = V_o = 0$

$M_k = n \cdot t_k$

$M_k = E_k - e \sin E_k$

$\cos V_k = (\cos E_k - e)/(1 - e \cos E_k)$

$\sin V_k = (1 - e^2)^{1/2} \sin E_k/(1 - e \cos E_k)$

$\omega_k = \omega_o - |\dot{\omega}|\ t_k$

$\phi_k = V_k + \omega_k$

$r_k = a(1 - e \cos E_k)$

$x_k = r_k \cos \phi_k$

$y_k = r_k \sin \phi_k$

$X_k = x_k \cos \lambda\Omega_k - y_k \cos i \sin \lambda\Omega_k$

$Y_k = x_k \sin \lambda\Omega_k + y_k \cos i \cos \lambda\Omega_k$

$Z_k = y_k \sin i$

(C) Theoretical or Tracked GPS Ephemeris (IPARA2=3)

See Chapter 2 of this report.

(D) NASA Prediction Bulletin (GAST used) (IPARA2=4)

$t_k = t - t_o$

t : alert time

t_o: time which refers the right ascension of ascending node (Part I NPB)

$\lambda\Omega_o = \alpha\Omega_o - GAST_o$

$\alpha\Omega_o$: right ascension of ascending node

$GAST_o$: Greenwich sideral time

$\lambda\Omega_o$: longitude (east) of ascending node

$$\lambda\Omega_k = \lambda\Omega_o - (\omega e - \dot{\Omega})\ t_k$$

$$M_k = M_o + nt_k$$

$$M_k = E_k - e \sin E_k$$

$$\cos V_k = (\cos E_k - e)/(1 - e \cos E_k)$$

$$\sin V_k = (1 - e^2)^{1/2} \sin E_k/(1 - e \cos E_k)$$

$$\omega_k = \omega_o + \dot{\omega}\ t_k$$

$$\phi_k = V_k + \omega_k$$

$$r_k = a(1 - e \cos E_k)$$

$$x_k = r_k \cos \phi_k$$

$$y_k = r_k \sin \phi_k$$

$$X_k = x_k \cos \lambda\Omega_k - y_k \cos i \sin \lambda\Omega_k$$

$$Y_k = x_k \sin \lambda\Omega_k + y_k \cos i \cos \lambda\Omega_k$$

$$Z_k = y_k \sin i$$

$\dot{\Omega}$, $\dot{\omega}$ are calculated from

$$\dot{\Omega} = \omega e - \frac{\text{westward motion}}{\text{nodal period}}$$

$$\dot{\omega} = \frac{2\pi}{\text{nodal period}} - n$$

Westward motion and nodal period are obtained from data of the Part II NPB.

(E) NASA Prediction Bulletin (GAST unused) (IPARA2=5)

t : alert time

t_o: time which refers the longitude (west) of ascending node (Part II NPB)

t_I: time which refers the mean anomaly (Part I NPB)

$$M_o = M_I - n(t_I - t_o)$$

$$t_k = t - t_o$$

$$\lambda\Omega_k = (360 - \lambda\Omega_o) - (\omega e - \dot{\Omega})\ t_k$$

$$M_k = M_o + n\ t_k$$

$$M_k = E_k - e \sin E_k$$

$$\cos V_k = (\cos E_k - e)/(1 - e \cos E_k)$$

$$\sin V_k = (1 - e^2)^{1/2} \sin E_k/(1 - e \cos E_k)$$

$$\omega_k = \omega_o + \dot{\omega}(t - t_I)$$

$$\phi_k = V_k + \omega_k$$

$$r_k = a(1 - e \cos E_k)$$

$$x_k = r_k \cos \phi_k$$

$$y_k = r_k \sin \phi_k$$

$$X_k = x_k \cos \lambda\Omega_k - y_k \cos i \sin \lambda\Omega_k$$

$$Y_k = x_k \sin \lambda\Omega_k + y_k \cos i \cos \lambda\Omega_k$$

$$Z_k = y_k \sin i$$

To better understand how the program ALERT works, the following block diagram, Figure 4.1, depicts the operative sequence of events which are encountered to produce the output. A complete list of the subroutines and functions which ALERT uses can be found in Appendix C of this report. In addition, a listing of the IBM program can be found in Appendix D.

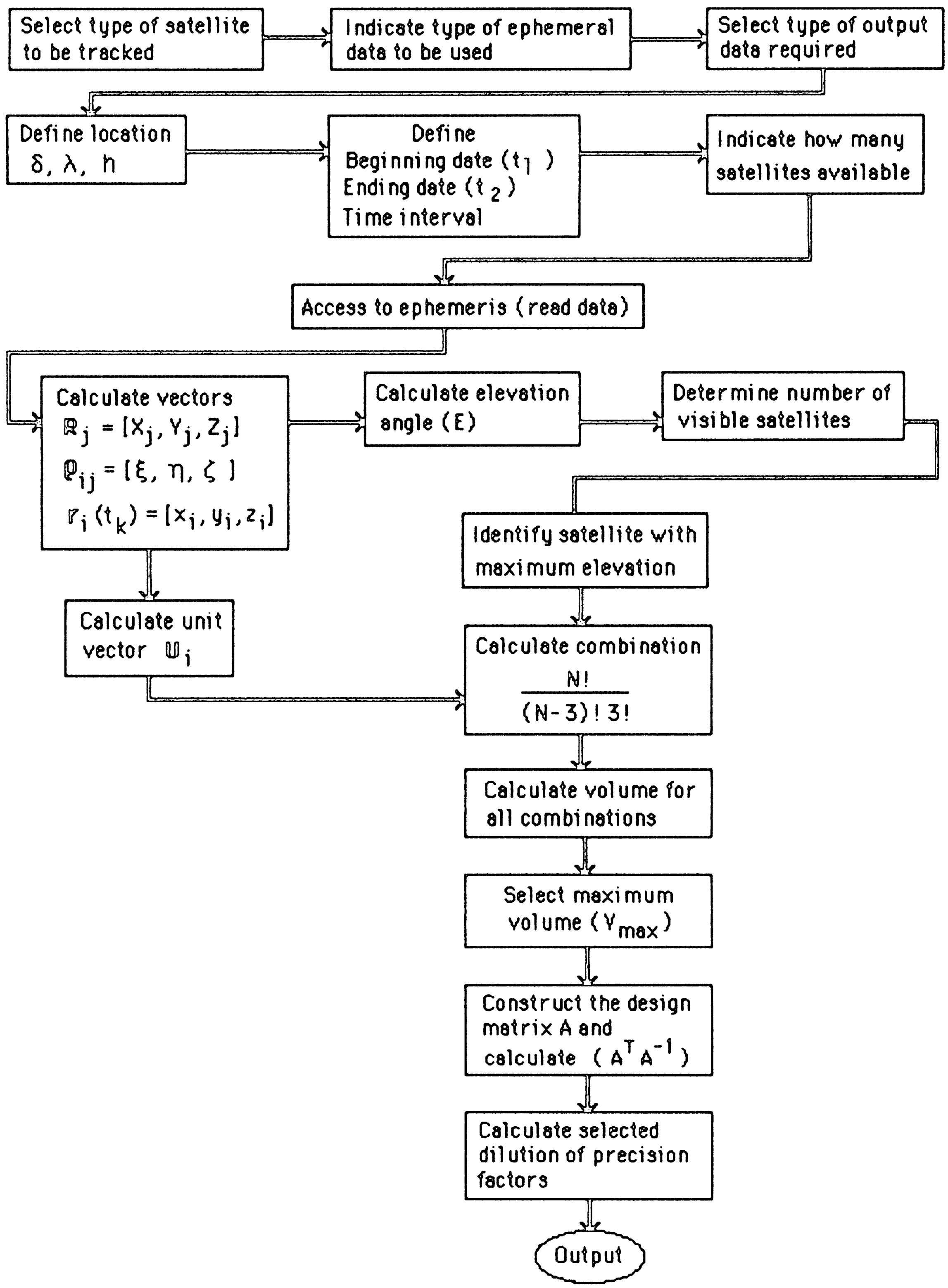


FIGURE 4.1
Block Diagram of the Alert program.

REFERENCES

Bogen, A.H. (1974). Geometric performance of the Global Positioning System. Aerospace Corporation, NTIS# AD-783210.

Brogan, L.W. (1980). Improvements and extensions of the geometric dilution of precision concept for selecting navigation measurements. IEEE, PLANS 80, Atlantic City, N.J., December 8-11, pp. 27-32.

Chapman, C.J. Jr. (1976). Comparison of low-cost computer algorithms for Global Positioning System users. Air Force Institute of Technology (AU), Ohio, NTIS# AD-A022 989.

Delikaraoglou, D. (1981). Application of NAVSTAR/GPS to precise positioning. Presented at annual meeting of the Canadian Geophysical Union, Calgary, Alberta, May.

Delikaraoglou, D. (1982). The continuing evolution of NAVSTAR/GPS survey navigation capabilities. Recent test results. EOS, Transactions American Geophysical Union, Vol. 63, No. 18, p. 300 (abstract only).

Jorgensen, P.S. (1980). NAVSTAR/Global Positioning System--18 satellite constellation. Navigation, Journal of the Institute of Navigation, Vol. 27, No. 2, pp. 89-100.

Krakiwsky, E.J. and D.E. Wells (1971). Mathematical models for positioning by satellite. Department of Surveying Engineering Lecture Note 17, University of New Brunswick, Fredericton, N.B.

McCaskil, T., J. Buisson, and A. Buonaguro (1976). A sequential range navigation algorithm for a medium altitude navigation satellite. Navigation, Journal of the Institute of Navigation, Vol. 23, No. 2, pp. 164-177.

Mikhail, E.M. (1976). Observations and Least Squares. Harper and Row, New York.

Milliken, R.J. and C.J. Zoller (1978). Principle of operation of NAVSTAR and system characteristics. Navigation, Journal of the Institute of Navigation, Vol. 25, No. 2, pp. 95-106.

Payne, C.R. Jr. (1982). NAVSTAR Global Positioning System: 1982. Proceedings of the Third International Geodetic Symposium on Satellite Doppler Positioning, Las Cruces, NM, February.

Swanson, E.R. (1978). Geometric dilution of precision. Navigation, Journal of the Institute of Navigation, Vol. 25, No. 4, pp. 425-429.

U.S. Department of Defense (1982). Federal radionavigation plan. Vols I, II, III, IV, March.

van Dierendonck, A.J., S.S. Russell, E.R. Kopitzke, and M. Birnbaum (1978). The GPS navigation message. Navigation, Journal of the Institute of Navigation, Vol. 25, No. 2, pp. 147-165.

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

Ward, P. (1981). An inside view of pseudorange and delta pseudorange measurements in a digital NAVSTAR/GPS receiver. Presented at the ITC/USA/'81 International Telemetering Conference, GPS--military and civil applications. San Diego, CA, October.

Wells, D.E. and D. Delikaraoglou (1980). NAVSTAR performance analysis. Department of Surveying Engineering Technical Report 75, University of New Brunswick, Fredericton, N.B.

Wells, D.E., and D. Delikaraoglou (1982). NAVSTAR data analysis. Department of Surveying Engineering Report, University of New Brunswick, Fredericton, N.B.

Wells, D.E. and G. Lachapelle (1981). Impact of NAVSTAR/GPS on land and offshore positioning in '80's. Proceedings of the Canadian Petroleum Association Colloquium III, on Petroleum Mapping and Surveys in the 80's, Banff, Alberta, October, pp. 305-322.

Wells, D.E., D. Delikaraoglou and P. Vanicek (1981). Navigating with the Global Positioning System. Presented at 74th annual meeting of CIS, St. John's, Newfoundland, May 22.

Wells, D.E., P. Vanicek and D. Delikaraoglou (1981). The application of NAVSTAR/GPS to geodesy in Canada. Pilot Study. Department of Surveying Engineering Technical Report 76, University of New Brunswick, Fredericton, N.B.

Doucet, K. (1986). Programs GEPSAL, GEPOS, GEPTH, POLAR, and TRACK--Description and user's guide. Department of Surveying Engineering Technical Memorandum 9, University of New Brunswick, Fredericton, N.B.

APPENDIX A

NOTATION USED

NOTATION USED

1. Position vector and Cartesian coordinates of the jth ground station (receiver):

$$\underline{R}_J = [X_J, Y_J, Z_J]^T \quad .$$

2. Position vector and Cartesian coordinates of the ith satellite at some epoch $t_k(\tau)$:

$$\underline{r}_i[t_k(\tau)] = [x_i(t_k(\tau)), y_i(t_k(\tau)), z_i(t_k(\tau))]^T \quad .$$

3. Geometric (true) range vector from jth receiver to ith satellite:

$$\underline{\rho}_{ij} = [\xi_{ij}, \eta_{ij}, \zeta_{ij}]^T \quad .$$

4. Unit vector from the jth receiver to the ith satellite:

$$U_i(t_k) = (u(t_k), v(t_k), w(t_k))^T \quad .$$

5. Magnitude of geometric (true) range:

$$\rho_{ij}(t_k) = \sqrt{\{X_j - x_i(t_k)\}^2 + \{Y_j - y_i(t_k)\}^2 + \{Z_j - z_i(t_k)\}^2} \quad .$$

6. Observed pseudorange vector from jth receiver to ith satellite:

$$\tilde{\underline{\rho}}_{ij}(t_k) = \rho_{ij}(t_k) + \Delta\rho_{RCV}(t_k) - \Delta\rho_{SAT}(t_k) + \Delta\rho_{ATM}(t_k) \quad .$$

7. Superscript zero in parenthesis indicates preliminary values:

$$\rho_{ij}^{(o)} \quad .$$

8. Observation vector of pseudoranges from jth receiver to four satellites:

$$\underline{L} = [\tilde{\rho}_{1j}, \tilde{\rho}_{2j}, \tilde{\rho}_{3j}, \tilde{\rho}_{4j}]^T \quad .$$

9. Receiver's solution:

$$\underline{X}_j = [X_j, Y_j, Z_j, \Delta t_{u_j}]^T \quad .$$

10. Vector of residuals:

$$\underline{V} = \underline{L} - \underline{L}^{(o)} \quad .$$

11. Misclosure vector:

$$\underline{W} = F(\underline{X}^{(o)}, \underline{L}^{(o)}) \quad .$$

12. Design matrices:

$$A = \{\frac{\partial F}{\partial \underline{X}_j}\bigg|_{\underline{X}_J}(o)\} \quad ; \quad B = \{\frac{\partial F}{\partial \underline{L}}\bigg|_{\underline{L}}(o)\}$$

13. Correction vector:

$$\underline{\delta X} = \underline{X}_j - \underline{X}_j^{(o)} \quad .$$

APPENDIX B

INPUT AND OUTPUT OF THE ALERT PROGRAM

INPUT AND OUTPUT OF THE ALERT PROGRAM

Card No. 1

Kind of satellite (IPARA1)

Kind of input data (IPARA2)

Kind of output data (IPARA3)

Where:

IPARA1=1 GPS satellites

IPARA1=2 TRANSIT satellites

IPARA1=3 Lageos satellite(s)

IPARA1=4 Starlette satellite(s)

IPARA2=1 NNSS nodal crossing data

IPARA2=2 NNSS Keplerian parameters or mean orbit by tracking

IPARA2=3 Theoretical or tracked GPS ephemeris

IPARA2=4 NASA Prediction Bulletin (GAST used)

IPARA2=5 NASA Prediction Bulletin (GAST unused)

If IPARA3=1 (simple approach), the output data will have this form:

a) Coordinates of the user's location, minimal satellite elevation.
b) Beginning and ending date for the alert.
c) Hour, minute (z-count, only for GPS).
d) Identity, elevation(s) and azimuth(s) of all visible satellites for the user's location and time.
e) Visibility summary (GPS only).

If IPARA3=2 (multiple approach), the output data will have this form (just for GPS):

a) Coordinates of the user's location, minimal satellite elevation.
b) Beginning and ending date for the alert.
c) Hour, minute and z-count (z-count for GPS only).
d) Identity, elevations and azimuths of all visible satellites for the user's location and time (only if the number of visible satellites is greater than 1).

If number of visible satellites equals two

e) The horizontal dilution of precision (HDOP).

If number of visible satellites equals three

e) The positional and the horizontal time dilution of precision (PDOP), (HTDOP).

If number of visible satellites is greater than three

e) The four satellites which yield the best navigation performance, and the geometric dilution of precision (GDOP).

f) Visibility summary (GPS only).

If IPARA3=3, the output data will have this form (for GPS only):

a) Coordinates of the user's location, minimal satellite elevation.

b) Beginning and ending date for the alert.

c) Hour, minute, and z-count (z-count for GPS only).

d) Identity, elevations, and azimuths of all the visible satellites for the user's location and interval time.

e) GDOP, PDOP, HTDOP, HDOP, and VDOP for all satellites.

f) Visibility summary (GPS only).

NOTE: Restrictions for the GPS visibility summary:

a) The start time must be an integer hour (i.e., $11^h\ 00^m\ 00^s$).

b) The increment time must equal to (1 or 3 or 5 or 15 or 30 or 60) minutes.

For example, if we want to compute the alert program for GPS satellites from NASA Prediction Bulletin (GAST used) in the simple approach, the first card would be

<u>Card No. 1</u>

1st card	0	1	0	4	0	1
	1	2	3	4	5	6

Table B.1
Summary of First Six Cards

Name and format of variable

Card No. 1

IPARA1,IPARA2,IPARA3	3I2	kind of satellite, kind of input, kind of output

Card No. 2

LATD,LATM,LATS,LOND,LONM,LONS, RHT, ELEVMN	6I4 F6.2 F3.0	lat(ϕ), lon(λ), h, minimal elevation

Card No. 3

KYEAR,KDAY,KHOUR,KMIN,KSEC	5I4	beginning date

Card No. 4

IYEAR,IDAY,IHOUR,IMIN,ISEC	5I4	ending date

Card No. 5

IDT	I2	time increment (minutes)

Card No. 6

NSAT,ID(I)	19I2	satellite number and ID

For NNSS nodal crossing data (IPARA2=1) the card No. 7 will have this format:

Name and format of variable

Card No. 7

IY	I5	input year
MDAY	I5	input day
JSAT	I5	satellite identification
PERIOD	F10.3	nodal period (minutes)
WMOTN	F10.3	westward motion (degrees)
TNODE	F10.3	time of ascending node (minutes)
XLNODE	F10.3	longitude (east) of ascending node (degrees)

and so on for each satellite.

For NNSS Keplerian parameters or mean orbit by tracking (IPARA2=2), cards No. 7 and 8 will have this format:

Name and format of variable

Card No. 7

IY	I5	input year
MDAY	I5	input day
JSAT	I5	satellite identification
TP	F10.4	time of perigee (minutes)
XN	F10.7	mean motion (degrees/minute)
W	F10.4	argument of perigee (degrees)
WDOT	F10.7	argument of perigee dot (degrees/minute)
EC	F10.6	eccentricity

Card No. 8

AO	F10.2	semi-major axis (metres)
OM	F10.4	right ascension of ascending node (degrees)
OMDOT	F10.7	right ascension dot (degrees/minute)
COSI	F10.6	cosine of inclination
GAST	F10.4	right ascension of Greenwich (degrees)
SINI	F10.6	sine of inclination

and so on for each satellite.

For NASA Prediction Bulletin (GAST used) (IPARA2=4) cards No. 7 and 8 will have this format:

Name and format of variable

Card No. 7

IDSAT	I5	satellite identification
IY	I5	input year
DAY	F13.8	input day (day and fraction of day)
XINC	F10.4	inclination (degrees)
RA	F10.4	right ascension of ascending node (degrees)
EC	F10.7	eccentricity
AP	F10.4	argument of perigee (degrees)

Card No. 8

XMA	F10.4	mean anomaly (degrees)
XN	F12.8	mean motion (revolutions/day)
PN	F9.3	nodal period (minutes)
WMTN	F9.3	westward motion (degrees)

and so on for each satellite.

For NASA Prediction Bulletin (GAST unused) (IPARA2=5) cards No. 7 and 8 will have this format:

Name and format of variable

Card No. 7

IDSAT	I5	satellite identification
IY	I5	input year
DAY	F13.8	input day (day and fraction of day)
XINC	F10.4	inclination (degrees)
EC	F10.7	eccentricity
AP	F10.4	argument of perigee (degrees)
XMA	F10.4	mean anomaly (degrees)

Card No. 8

XN	F12.8	mean motion (revolutions/day)
PN	F9.3	nodal period (minutes)
WMTN	F9.3	westward motion (degrees)
XLANW	F8.2	longitude (west) of ascending node (degrees)
KYAN	I5	year of ascending node
KDAYAN	I5	day of ascending node
KHAN	I3	hour of ascending node
XMAN	F6.2	time of ascending node (minutes and fraction of minute)

and so on for each satellite.

Theoretical or tracked GPS ephemeris (IPARA2=3)

The orbital element of the satellites are stored in such a way that they accommodate to the IBM format (see Wells and Delikaraoglou [1982]). Table B.2 shows the orbital parameters used, and Tables B.3(a) and B.3(b) show the IBM format in which the data should be stored.

UNIVERSITY OF NEW BRUNSWICK
COMPUTER CODING FORM

```
NNSS NODAL CROSSING DATA (IPARA2=1)
 1983   60   13    06.926    26.807     7.84     53.98

NNSS KEPLERIAN PARAMETERS (IPARA2=2)
 1982  350   13   56.1484 3.3688047   62.4118  .0019975   .0021
   7456.59  218.8682 -.0000233   .005838   98.3882   .999983

NASA PREDICTION BULLETIN (GAST USED) (IPARA2=4)
    3 1983  96.11950818   63.6893   48.8951  .0020243  120.0282
  240.1574  2.00562924  717.980  180.002

NASA PREDICTION BULLETIN (GAST UNUSED) (IPARA2=5)
    3 1983  96.11900818   63.6893  .0020243  120.0282  240.1574
  2.00562924  717.980  180.002  187.80 1983  100  2 35.20
```

TABLE B.2

Orbital Parameters for the 18-Satellite Constellation.

SAT	M_o	e	a	t_{oe}	Ω_o	i_o	$\dot{\Omega}_o$
1	0.0	3.0D-3	5153.6783	585728	0.0	.9599D0	.6262D-8
2	2.094	3.0D-3	5153.6783	585728	0.0	.9599D0	.6262D-8
3	4.189	3.0D-3	5153.6783	585728	0.0	.9599D0	.6262D-8
4	0.698	3.0D-3	5153.6783	585728	1.047	.9599D0	.6262D-8
5	2.793	3.0D-3	5153.6783	585728	1.047	.9599D0	.6262D-8
6	4.887	3.0D-3	5153.6783	585728	1.047	.9599D0	.6262D-8
7	1.396	3.0D-3	5153.6783	585728	2.094	.9599D0	.6262D-8
8	3.491	3.0D-3	5153.6783	585728	2.094	.9599D0	.6262D-8
9	5.585	3.0D-3	5153.6783	585728	2.094	.9599D0	.6262D-8
10	2.094	3.0D-3	5153.6783	585728	3.142	.9599D0	.6262D-8
11	4.189	3.0D-3	5153.6783	585728	3.142	.9599D0	.6262D-8
12	0.00	3.0D-3	5153.6783	585728	3.142	.9599D0	.6262D-8
13	2.793	3.0D-3	5153.6783	585728	4.189	.9599D0	.6260D-8
14	4.887	3.0D-3	5153.6783	585728	4.189	.9599D0	.6262D-8
15	0.698	3.0D-3	5153.6783	585728	4.189	.9599D0	.6262D-8
16	3.491	3.0D-3	5153.6783	585728	5.236	.9599D0	.6262D-8
17	5.585	3.0D-3	5153.6783	585728	5.236	.9599D0	.6262D-8
18	1.396	3.0D-3	5153.6783	585728	5.236	.9599D0	.6262D-8

M_o = mean anomaly at reference time (radians).

e = orbital eccentricity.

a = semi major axis of orbital ellipse (metres).

t_{oe} = ephemeris reference time (seconds).

Ω_o = right ascension at reference time (radians).

i_o = inclination angle at reference time (radians).

$\dot{\Omega}_o$ = rate of right ascension (rad/sec).

w = argument of the perigee = 0.

SUB#3 / IDSAT / IDRECEIVER	HOW (68157)	α_0	α_1	Clock
α_2	α_3	β_0	β_1	Clock
β_2	β_3	T_{GD}	AODC	Clock
t_{oc}	a_2	a_1	a_0	Clock
SUB#4 / IDSAT / IDRECEIVER	HOW(68157)	EPH(1)=AODE	EPH(2)=C_{rs}(=0)	Satellite Ephemeris
EPH(3)= Δn	EPH(4)= M_0(semicircle)	EPH(5)=C_{uc}(=0)	EPH(6)= e	Satellite Ephemeris
EPH(7)= C_{us}(=0)	EPH(8)=$\sqrt{a}$ ($\sqrt{\text{metre}}$)	EPH(9)= t_{oe} (sec)	0	Satellite Ephemeris
0	0	0	0	Satellite Ephemeris
SUB#5 / IDSAT / IDRECEIVER	HOW (=68157)	EPH(10)=C_{ic}(=0)	EPH(11)=Ω_0(semicircle)	Satellite Ephemeris
EPH(12)=C_{is}(=0)	EPH(13)=i_0(semicircle)	EPH(14)=C_{rc}(=0)	EPH(15)= ω(semicircle)	Satellite Ephemeris
EPH(16)= $\dot{\Omega}$(sem/sec)	EPH(17)= IDSAT	0	0	Satellite Ephemeris
0	0	0	0	Satellite Ephemeris

Table B.3(a)

Format of Orbital Elements.

TABLE B.3(b)

Explanation of Orbital Parameters.

(For more information see van Dierendonck et al. [1978].)

α_o	ionospheric correction parameter (s)
α_1	ionospheric correction parameter (s)
α_2	ionospheric correction parameter (s)
α_3	ionospheric correction parameter (s)
β_o	ionospheric correction parameter (s)
β_1	ionospheric correction parameter (s)
β_2	ionospheric correction parameter (s)
β_3	ionospheric correction parameter (s)
T_{GD}	group delay differential between L1 and L2 (s)
AODC	age of data (t_{oc} - last upload) (s)
t_{oc}	clock correction parameter reference time (s)
a_2	clock correction parameter (s^{-1})
a_1	clock correction parameter (s^o)
a_o	clock correction parameter (s)
e	eccentricity
t_{oa}	almanac reference time (s)
δ_i	inclination difference from 60^o
$\dot{\Omega}$	rate of right ascension (s^{-1})
$\sqrt{a}$	square root of semi major axis ($\sqrt{M}$)
Ω_o	right ascension at t_{oa}
ω	argument of perigee
M_o	mean anomaly at t_{oa}
AODE	age of data (ephemeris)(t_{oe} - t_L (last upload))(s)

APPENDIX C

LIST OF SUBROUTINES AND FUNCTIONS

Program No.	Program Name IBM/HP 1000
1	DOPS: computes dilution of precision factors.
2	DOP4S: computes dilution of precision factors from 4 satellites (4 "best" satellites).
3	DOP3S: computes dilution of precision factors from 3 satellites.
4	DOP2S: computes dilution of precision factors from 2 satellites.
5	VOLUME: to select the set of 4 satellites which provide the "best" navigation geometry.
6	TIME: computes the number of seconds between two given epochs.
7	DAYFRA: transforms day and fraction of day into day, hour, minute, and second.
8	MINFRA: transforms minute and fraction of minute into hour, minute, and second.
9	NNSSNC: computes the satellite coordinates in the earth-fixed system from NNSS nodal crossing data.
10	NNSSKE: computes the satellite coordinates in the earth-fixed system from NNSS Keplerian parameters.
11	NPBST: computes the satellite coordinates in the earth-fixed system from NASA prediction bulletin (GAST used).
12	NPB: computes satellite coordinates in the earth-fixed system from NASA prediction bulletin (GAST unused).
13	EPNCD: reads input for NNSS nodal crossing data.
14	EPKE: reads input for NNSS Keplerian parameters.
15	EPNPBG: reads input for NASA prediction bulletin (GAST used).
16	EPNPB: reads input for NASA prediction bulletin (GAST unused).
17	DATUM: initializes datum parameters. Default (WGS-72).
18	ZCOUNT: given a time, compute the GPS satellite z-count.
19	READEF: reads ephemeris disc file (GPS satellites only).
20	DASET: copy a vector.
21	PLHXYZ: compute Cartesian coordinates x, y, z given ellipsoidal coordinates ϕ, λ, h.
22	STXYZ: computes satellite coordinates.
23	DERIV: computes slant ranges to the satellites and derivatives with respect to ϕ, λ, and elevation angle.
24	MATMPY: computes the product of two matrices.
25	ANMLY: convert eccentricity and mean anomaly to eccentric and time anomaly.
26	ROTREF: computes product matrix of sequence of rotations and reflections.

Program No.	Program Name IBM/HP 1000
27	NCLOK: extracts clock values from input records.
28	NEPHM: extracts ephemeris values from input record.
29	RANGE: computes station-to-station range given station and satellite Cartesian coordinates.
30	ZFRDAY: converts z-count into fraction of a day.
31	SPIN: matrix inversion program for positive semi-definite matrices.
32	DJUL: computes modified Julian date from year, month, and day.
33	THETA: computes Greenwich mean siderial time given modified Julian date.

APPENDIX D

- PROGRAM LISTING
- SAMPLE DATA (INPUT)
- SAMPLE OUTPUT

PROGRAM LISTING

```
//VISIBLE4 JOB NOTIFY=1212003
/*JOBPARM S=59,L=99,R=5012
/*SERVICE -4
// EXEC FORTVCLG,RC=5012K,RG=1216K
//FORT.SYSIN DD *
C -----------------------------------------------------------------------00002
C                                                                         00001
C                                                                         00003
C                                                                         00005
C  NAME       :   ALERT                                                   00004
C  TYPE       :   MAIN PROGRAMME                                          00006
C                                                                         00007
C  PURPOSE    :   DETERMINES THE # AND ID OF ALL VISIBLE SATELLITES       00008
C                 ALONG WITH THE FOUR SATELLITES WHICH YIELD THE          00009
C                 THE BEST NAV PERFORMANCE AND GDOP VALUES.               00010
C
C                 COMPUTES ALERT FOR DIFFERENT SATELLITES AND
C                 DIFFERENT INPUTS
C                                                                         00011
C  AUTHOR     :   JUNE 1981 S. MERTIKAS                                   00012
C                                                                         00013
C  MODIFIED   :   FEBR 1983 D. DELIKARAOGLOU                              00014
C                                                                         00015
C  MODIFIED   :   JUNE 1983 R. SANTERRE
C
      IMPLICIT REAL*8(A-H,O-Z)                                            00016
      LOGICAL*1 ISTAR/'*'/ , IBLANK/'-'/
      LOGICAL*1 ICRL(100)
C                                                                         00017
C   THE FOLLOWING ARRAYS HAVE ROW DIMENSION 'MXVS' EQUAL                  00018
C   TO THE MAXIMUM NUMBER OF ANTICIPATED VISIBLE SATELLITES               00019
C                                                                         00020
      INTEGER*4 AZBUF(9),ELEBUF(9) ,IVIS(9),IHRVIS(18,100),IHH(25)        00021**2
      REAL*4 ELV,AZIM
      DIMENSION VISAT(9,3) , ELV(9) , AZIM(9)                             00022**2
C                                                                         00023
C                                                                         00024
      DIMENSION DTM(7)       , REC(16)      , EPH(18)    , CLK(6) ,       00025
     #          EPHBUF(324)  , CLKBUF(108)  , IDSAT(18)  , Y(5)   ,       00026
     #          XSVBUF(18,3) , XRCV(3)      , XSV(3)                      00027
      DIMENSION IBEST(4)     , KK(4)        , A(9,9)     ,                00028
     #          B(9,9)       , IAZ(4)       , IELEV(4)                    00029
      DIMENSION IY(18)     , MDAY(18) ,  JSAT(18) ,  PERIOD(18),
     #          WMOTN(18)  , TNODE(18),  XLNODE(18)
      DIMENSION TP(18)     , XN(18)   ,  W(18)    ,  WDOT(18)  ,
     #          EC(18)     , AO(18)   ,  OM(18)   ,  OMDOT(18) ,
     #          COSI(18)   , GAST(18) ,  SINI(18)
      DIMENSION XINC(18)   , RA(18)   ,  AP(18)   ,  XMA(18)   ,
     #          PN(18)     , WMTN(18) ,  DAY(18)
      DIMENSION XLANW(18)  , KYAN(18) ,  KDAYAN(18), KHAN(18)  ,
     #          XMAN(18)
C                                                                         00030
C ----------------------------------------------------------------        00031
C     DEFINE I/O LOGICAL UNITS                                            00032
C                                                                         00033
C                 ICR    -  CARD READER                                   00034**3
C                 IPR    -  LINE PRINTER DETAILED OUTPUT                  00035**3
C                 IPR1   -  LINE PRINTER SUMMARY OUTPUT                   00036**3
C           IPR2  IPR3   -  OUTPUT FILE UNIT FOR PLOTTING                 00037**3
C                 LEPH   -  INPUT FILE UNIT FOR SATELLITE EPHEMERIS       00038**3
C                 LU     -  OUTPUT FILE UNIT FOR OPTIMUM TRACKING         00039**3
```

```
C                         SCHEDULE IF REQUIRED                                00040**3
C                                                                             00041**3
C      IF ANY OF THE I/O UNITS IS NOT USED SPECIFY IN YOUR JCL CARDS          00042**3
C                 //GO.FT##F001 DD  DUMMY         FOR THIS UNIT               00043**3
C                                                                             00044**3
C                                                                             00045
C ------------------------------------------------------------------          00046
C                                                                             00047
      DATA ICR / 5/     ,   IPR / 6/    ,    IPR1/16/   ,                     00048**3
     #     IPR2/17/     ,   LEPH/18/    ,    LU  /25/  ,IPR3 /19/             00049**3
C ----------------------------------------------------------------------
C     KIND OF INPUT AND KIND OF OUTPUT
C
C         IPARA1=1  :  GPS SATELLITES
C         IPARA1=2  :  TRANSIT SATELLITES
C         IPARA1=3  :  LAGEOS SATELLITE(S)
C         IPARA1=4  :  STARLETTE SATELLITE(S)
C
C         IPARA2=1  :  NNSS NODAL CROSSING DATA
C         IPARA2=2  :  KEPLERIAN PARAMETERS
C         IPARA2=3  :  GPS THEORETICAL OR TRACKED EPHEMERIS
C         IPARA2=4  :  NASA PREDICTION BULLETIN (GAST USED)
C         IPARA2=5  :  NASA PREDICTION BULLETIN (GAST UNUSED)
C
C         IPARA3=1  :  SIMPLE APPROACH (ELEV,AZ)
C         IPARA3=2  :  ELEV,AZ AND DIL. OF PRECISION FOR THE BEST 4 SAT
C         IPARA3=3  :  ELEV,AZ AND DIL. OF PRECISION FOR ALL SATELLITES
C ----------------------------------------------------------------------
      READ(ICR,134) IPARA1,IPARA2,IPARA3
C                                                                             00050
C ------------------------------------------------------------------          00051
C     DEFINITION  OF SOME  BASIC  CONSTANTS                                   00052
C ------------------------------------------------------------------          00053
C                                                                             00054
      IGAP=0
      PI=DARCOS(-1.D0)                                                        00055
      DTR=PI/180.D0                                                           00056
      C=299792458.D0                                                          00057
      RMEAN=6371.D3                                                           00058
      F1=1575.D6                                                              00059
      F2=1227.D6                                                              00060
C                                                                             00062
      ICD=5                                                                   00063
      SOBS= 4.D0                                                              00064
C                                                                             00065
C        MXVS - MAXIMUM NUMBER OF VISIBLE SATELLITES                          00066
C               (SEE NOTE IN DIMENSION STATEMENT ABOVE)                       00067
C                                                                             00068
      MAXSAT = 18                                                             00069**2
      MXVS = 9                                                                00070**2
      IDA = 9                                                                 00071**2
      MREC = 0                                                                00072**4
C                                                                             00073
C--------------------------------------------------------------------         00074
C                      DEFINE REFERENCE  ELLIPSOID                            00075
C--------------------------------------------------------------------         00076
C                                                                             00077
      CALL  DATUM(DTM,0,ICR)                                                  00078
      BE=DSQRT((1.D0-DTM(2)**2)*DTM(1)**2)                                    00079
C                                                                             00080
C---------------------------------------------------------------------        00081
```

```
C        DEFINE LOCATION AND TIME INTERVAL FOR ALERT                      00082
C                                                                         00083
C               LATITUDE  IN DEG , MIN , SEC                              00084
C               LONGITUDE IN DEG , MIN , SEC (+VE EAST)                   00085
C               HEIGHT    IN METRES                                       00086
C               BEGINNING TIME - YR, DAY, HR, MIN, SEC                    00087
C               END TIME       - YR, DAY, HR, MIN, SEC                    00088
C               TIME INCREMENT (FOR OUTPUT) IN MINUTES                    00089
C                                                                         00090
C-----------------------------------------------------------------------  00091
C                                                                         00092
      READ(ICR,1) LATD,LATM,LATS,LOND,LONM,LONS,RHT,ELEVMN                00093
      KRHT=IDINT(RHT)
      KELEV=IDINT(ELEVMN)
      WRITE(IPR,2)                                                        00094
      GO TO (310,320,330,340,350) , IPARA2
  310 WRITE(IPR,311)
      GO TO 360
  320 WRITE(IPR,321)
      GO TO 360
  330 WRITE(IPR,331)
      GO TO 360
  340 WRITE(IPR,341)
      GO TO 360
  350 WRITE(IPR,351)
  360 CONTINUE
      GO TO(155,160,165,170) , IPARA1
  155 WRITE(IPR,156)
      GO TO 200
  160 WRITE(IPR,161)
      GO TO 200
  165 WRITE(IPR,166)
      GO TO 200
  170 WRITE(IPR,171)
  200 CONTINUE
      WRITE(IPR,8000) KELEV
      WRITE(IPR,3) LATD,LATM,LATS,LOND,LONM,LONS                          00095
      WRITE(IPR,7000) KRHT
C                                                                         00096
C-----------------------------------------------------------------------  00097
C         COMPUTE THE Z-COUNT FOR THE :                                   00098
C               BEGINNING  DATE (ZCNT1)                                   00099
C                ENDING    DATE (ZCNT2)                                   00100
C-----------------------------------------------------------------------  00101
C                                                                         00102
      READ(ICR,4) KYEAR,KDAY,KHR,KMIN,KSEC                                00103
      KKHR=KHR
      KKYEAR=KYEAR
      KKDAY=KDAY
      KKMIN=KMIN
      KKSEC=KSEC
      READ(ICR,4) IYEAR,IDAY,IHR,IMIN,ISEC                                00104
      READ(ICR,7) IDT                                                     00105
      IF(IPARA3 .EQ. 1 .AND. IPARA1 .NE.1) GO TO 135
      ZCNT1 = ZCOUNT(KYEAR,KDAY,KHR,KMIN,KSEC)                            00106
      ZZZ=ZCNT1
      ZCNT2 = ZCOUNT(IYEAR,IDAY,IHR,IMIN,ISEC)                            00107
      WRITE(IPR,5) KDAY,KYEAR,KHR,KMIN,KSEC,ZCNT1                         00108**2
      WRITE(IPR,6) IDAY,IYEAR,IHR,IMIN,ISEC,ZCNT2                         00109**2
      GO TO 136
  135 WRITE(IPR,35) KDAY,KYEAR,KHR,KMIN,KSEC
```

```
      WRITE(IPR,36) IDAY,IYEAR,IHR,IMIN,ISEC
  136 CONTINUE
C
C_____________________________________________________________________
C
C      HOW  MANY  SATELLITES  YOU   HAVE  AVAILABLE?
C
C          IF SOME ARE AVAILABLE:
C                                A) DEFINE  NSAT=?
C                                B) DEFINE  IDSAT=?
C          IF ALL ARE  AVAILABLE:
C                                A) NSAT=18
C                                B) IDSAT(I)=I  ,I=1,18
C                                C) EPHBUF SHOULD BE FULL
C---------------------------------------------------------------------
C
      READ(ICR,8) NSAT  ,(IDSAT(I),I=1,NSAT)
      IF (NSAT .NE. MAXSAT) GO TO 9
      DO 10 I=1,MAXSAT
      IDSAT(I)=I
   10 CONTINUE
    9 MAXSAT=NSAT
C
C------------------------------------------------------------------
C       READ EPHEMERIS PARAMETERS FOR ALL AVAILABLE SATELLITES
C--------------------------------------------------------------------
C
      GO TO(43,44,210,131,132) , IPARA2
C
  210 CALL READEF(REC,DTM,EPH,EPHBUF,CLK,CLKBUF,MAXSAT,ICR,IPR)
C
C
      DO 11  I=1,MAXSAT
            ISAT=IDSAT(I)
            IPSN=IDSAT(I)*6-5
            CALL DASET(CLKBUF(IPSN),CLK(1),6)
            IPSN=IDSAT(I)*18-17
            CALL DASET(EPHBUF(IPSN),EPH(1),18)
   11 CONTINUE
      GO TO 133
C
   43 CALL EPNCD(NSAT,IY,MDAY,JSAT,PERIOD,WMOTN,TNODE,XLNODE,ICR)
      GO TO 133
C
   44 CALL EPKE(NSAT,IY,MDAY,JSAT,TP,XN,W,WDOT,EC,AO,OM,OMDOT,
     #     COSI,GAST,SINI,ICR)
      GO TO 133
C
  131 CALL EPNPBG(NSAT,IDSAT,IY,DAY,XINC,RA,EC,AP,XMA,XN,PN,WMTN,ICR)
      GO TO 133
C
  132 CALL EPNPB(NSAT,IDSAT,IY,DAY,XINC,EC,AP,XMA,XN,PN,WMTN,XLANW,
     #            KYAN,KDAYAN,KHAN,XMAN,ICR)
C
  133 CONTINUE
C
      IF(IPARA3 .EQ. 1 .AND. IPARA1 .EQ. 1) GO TO 600
      IF(IPARA3 .EQ. 1) GO TO 137
      IF(IPARA3 .EQ. 2) WRITE(IPR,12)
      IF(IPARA3 .EQ. 3) WRITE(IPR,820)
      WRITE(IPR1,94)
```

```
      GO TO 138
  600 WRITE(IPR,601)
      GO TO 138
  137 WRITE(IPR,37)
  138 CONTINUE
C
C----------------------------------------------------------------------
C        CONVERSION  OF  RECEIVER  LONGITUDE  AND  LATITUDE
C         EXPRESSED  IN  (DEG, MIN, SEC) INTO  RADIANS
C----------------------------------------------------------------------
C
      XLAT= DTR*  (LATD+(DFLOAT(LATM)+(DFLOAT(LATS)/60.))/60.)
      XLON= DTR*  (LOND+(DFLOAT(LONM)+(DFLOAT(LONS)/60.))/60.)
C
C----------------------------------------------------------------------
C       CONVERT RECEIVER GEODETIC  CORDINATES (RPHI,RLAT,RHT)
C                         INTO CARTESIAN  ONES (XR ,YR  ,ZR )
C----------------------------------------------------------------------
C
      CALL PLHXYZ(XLAT,XLON,RHT,DTM(3),DTM(4),DTM(5),DTM(1),
     #            BE,XR,YR,ZR)
      XRCV(1)=XR
      XRCV(2)=YR
      XRCV(3)=ZR
C
C----------------------------------------------------------------------
C        COMPUTE  SATELLITE COORDINATES
C----------------------------------------------------------------------
C
       IPP=0
   14 J = 0
      IP=MOD(KMIN,15)
      IF(IP .EQ. 0) IPP=IPP+1
      IF(IPARA1 .NE. 1) GO TO 139
      ZCNT1 = ZCOUNT(KYEAR,KDAY,KHR,KMIN,KSEC)
  139 CONTINUE
      DO 15 K=1,10
          IVIS(K)    = 0
          ELEBUF(K)  = 0
          AZBUF(K)   = 0
          ELV(K)=0.
          AZIM(K)=0.
      IF(IPARA3 .EQ. 1) GO TO 15
          VISAT(K,1)= 0
          VISAT(K,2)= 0
          VISAT(K,3)= 0
   15 CONTINUE
      IF(IPARA3 .EQ. 1) GO TO 140
      ELVMAX=0.DO
  140 CONTINUE
      DO 17 I=1,NSAT
      GO TO(141,142,220,143,144) , IPARA2
  220     IPSN=IDSAT(I)*18-17
          CALL DASET(EPHBUF(IPSN),EPH(1),18)
          Y(1)=EPH(8)
          Y(2)=EPH(6)
          Y(3)=EPH(11)
          Y(4)=EPH(13)
          Y(5)=EPH(15)
          TIME=ZCNT1*6.-EPH(9)
          CALL STXYZ(AMYE,C,DTM,EPH,ICD,IFLAG4,IPR,PHIK,PI,RING,
```

```
     #               RLNG,TIME,XR,YR,ZR,XS,YS,ZS,Y,IPC,IPR)
C
      GO TO 145
C
  141 CALL NNSSNC(IY(I),MDAY(I),TNODE(I),XLNODE(I),PERIOD(I),
     #             WMOTN(I),KYEAR,KDAY,KHR,KMIN,KSEC,PI,XS,YS,ZS)
      GO TO 145
C
  142 CALL NNSSKE(IY(I),MDAY(I),TP(I),XN(I),W(I),WDOT(I),EC(I),AO(I),
     #             OM(I),OMDOT(I),COSI(I),GAST(I),SINI(I),KYEAR,KDAY,
     #             KHR,KMIN,KSEC,DTM(7),PI,XS,YS,ZS,IPR)
      GO TO 145
C
  143 CALL NPBST(IY(I),DAY(I),XINC(I),RA(I),EC(I),AP(I),XMA(I),XN(I),
     #           PN(I),WMTN(I),KYEAR,KDAY,KHR,
     #           KMIN,KSEC,PI,DTM(7),DTM(6),XS,YS,ZS,IPR)
      GO TO 145
C
  144 CALL NPB(IY(I),DAY(I),XINC(I),EC(I),AP(I),XMA(I),XN(I),PN(I),
     #         WMTN(I),XLANW(I),KYAN(I),KDAYAN(I),KHAN(I),XMAN(I),KYEAR,
     #         KDAY,KHR,KMIN,KSEC,PI,DTM(7),DTM(6),XS,YS,ZS,IPR)
C
  145 CONTINUE
C
          XSV(1)=XS
          XSV(2)=YS
          XSV(3)=ZS
C
      IF(IPARA3 .EQ. 1) GO TO 146
          XSVBUF(I,1) = XSV(1)
          XSVBUF(I,2) = XSV(2)
          XSVBUF(I,3) = XSV(3)
  146 CONTINUE
C
C-----------------------------------------------------------------------
C       CALCULATE  SATELLITE   ELEVATION  ANGLE(ELEV) AND
C                              AZIMUTH(AZ)   IN  DEGREES
C----------------------------------------------------------------------
C
        CALL DERIV(IPR,DTM(1),BE,XLAT/DTR,XLON/DTR,RHT,XSV,RNG,DSDP,
     #             DSDL,ELEV,AZ,IER)
C
C-----------------------------------------------------------------------
C        DETERMINE  NUMBER  OF  VISIBLE  SATELLITES
C-----------------------------------------------------------------------
C
          IF(ELEV.GT.ELEVMN)  GO TO 20
              GO  TO  17
C
C-----------------------------------------------------------------------
C        IDENTIFY  SATELLITE  WITH  MAXIMUM  ELEVATION
C        COMPUTE  DIRECTION  COSINES
C        OR COMPILE SATELLITES , ELEVATIONS AND AZIMUTHS ONLY
C-----------------------------------------------------------------------
C
   20     J = J + 1
      IF(IPARA3 .NE. 1) GO TO 148
      IF (IPARA3 .EQ. 1 .AND. IPARA1 .EQ. 1) GO TO 300
      IF(J .GT. 3) WRITE(IPR,150)
      ELEBUF(J)=ELEV
      AZBUF(J)=AZ
```

```
      IVIS(J)=IDSAT(I)
      GO TO 17
  300 IF(J .GT. MXVS) J=J-1
      IF(J .GT. MXVS) GO TO 17
      ELEBUF(J)=ELEV
      AZBUF(J)=AZ
      IVIS(J)=IDSAT(I)
      ELV(J)=ELEV
      AZIM(J)=AZ
      IF(IP .EQ. 0) IHRVIS(I,IPP)=1
      GO TO 17
  148 CONTINUE
      IF(J .GT. MXVS) WRITE(IPR,1040) J,MXVS,J
          IF(J .GT. MXVS) J = J - 1
          IF(J .GT. MXVS) GO TO 17
          ELEBUF(J) = ELEV
          AZBUF(J)  = AZ
          IVIS(J)=IDSAT(I)
          IF(IP .EQ. 0) IHRVIS(I,IPP)=1
          IF(ELEV.GT.ELVMAX) GO TO 30
          GO TO 32
   30     ELVMAX=ELEV
          KSAT=J
C  32     VISAT(J,1)= ( XSV(1)-XRCV(1) )/RNG
C         VISAT(J,2)= ( XSV(2)-XRCV(2) )/RNG
C         VISAT(J,3)= ( XSV(3)-XRCV(3) )/RNG
   32     CONTINUE
C
C         DIRECTION COSINES IN THE TANGENTIAL SYSTEM (NORD,EST,HEIGHT)
C         MODIFICATION 23 MAI 1985 R.SANTERRE.
C
          VISAT(J,1) = DCOS(AZ*DTR) * DCOS(ELEV*DTR)
          VISAT(J,2) = DSIN(AZ*DTR) * DCOS(ELEV*DTR)
          VISAT(J,3) = DSIN(ELEV*DTR)
C
   17     CONTINUE
      IF(IPARA3 .NE. 1) GO TO 152
      IF(J .EQ. 0) IGAP=IGAP+1
      IF(J .NE. 0) IGAP=0
      IF(J .EQ. 0 .AND. IGAP .EQ. 1 .AND. IPARA1 .EQ. 1)WRITE(IPR,901)
      IF(J .EQ. 0 .AND. IGAP .EQ. 1 .AND. IPARA1 .NE. 1)WRITE(IPR,902)
      IF (J .EQ. 0) GO TO 153
      IF(IPARA3 .EQ. 1 .AND. IPARA1 .EQ. 1) GO TO 400
      IF(J .GT. 3) GO TO 153
      GO TO (471,472,473) ,J
  471 WRITE(IPR,481) KHR,KMIN,(IVIS(N),N=1,J),(ELEBUF(N),N=1,J),
     #               (AZBUF(N),N=1,J)
      GO TO 153
  472 WRITE(IPR,482) KHR,KMIN,(IVIS(N),N=1,J),(ELEBUF(N),N=1,J),
     #               (AZBUF(N),N=1,J)
      GO TO 153
  473 WRITE(IPR,483) KHR,KMIN,(IVIS(N),N=1,J),(ELEBUF(N),N=1,J),
     #               (AZBUF(N),N=1,J)
      GO TO 153
  400 IZCNT=IDINT(ZCNT1)
      WRITE(IPR3,1117) KHR,KMIN,(IVIS(I),I=1,9),(ELV(I),I=1,9),
     #                  (AZIM(I),I=1,9)
      GO TO(451,452,453,454,455,456,457,458,459) ,J
  451 WRITE(IPR,461) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #               (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
      GO TO 153
```

```
 452 WRITE(IPR,462) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO153
 453 WRITE(IPR,463) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO153
 454 WRITE(IPR,464) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO153
 455 WRITE(IPR,465) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO153
 456 WRITE(IPR,466) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO 153
 457 WRITE(IPR,467) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO153
 458 WRITE(IPR,468) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO 153
 459 WRITE(IPR,469) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
    #              (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J)
     GO TO 153
 152 CONTINUE
     IF(J .EQ. 0) IGAP=IGAP+1
     IF(J .NE. 0) IGAP=0
     IF(J .EQ. 0 .AND. IGAP .EQ.1) WRITE(IPR,903)
     IF(J .EQ. 0)  GO TO 25
     WRITE(14,1) J
     DO 13 L = 1,NSAT
     IS = IDSAT(L)
     DO 18 K = 1,J
     IF(IS .EQ. IVIS(K)) I1 = L
     IF(IS .EQ. IVIS(K)) GO TO 16
  18 CONTINUE
     GO TO 13
  16 WRITE(14,1110) IVIS(K),(XSVBUF(I1,M),M=1,3)
1110 FORMAT(I4,3F14.4)
  13 CONTINUE
C
C
     IF(IPARA3 .EQ. 3) GO TO 24
C-----------------------------------------------------------------------
C      IF IPARA3 EQUAL 2
C
C  1.  CALCULATION OF THE VOLUME OF THE TETRAHEDRON FORMED BY THE
C      FOUR UNIT VECTORS TO EACH SATELLITE FOR ALL COMBINATIONS
C  2.  SELECTION OF THE MAXIMUM VOLUME(VOL)
C  3.  SELECTION OF THOSE VISIBLE SATELLITES WITH BEST NAVIGATION
C      PERFORMANCE (MAX VOLUME)
C-----------------------------------------------------------------------
C
     CALL VOLUME(KSAT,VISAT,MXVS,J,KK,VOL)
     DO 45 L=1 ,4
        IBEST(L) = IVIS( KK(L) )
        IELEV(L) = ELEBUF( KK(L) )
        IAZ(L)   = AZBUF(  KK(L) )
  45 CONTINUE
     IF(J .GE. 4) NVIS = 4
     IF(J .LT. 4) NVIS = J
```

```
      GO TO (25,21,22,23) , NVIS
      GO TO 25
C
C ------------------------------------------------------------------------
C         FOR 3 VISIBLE SATELLITES COMPUTE
C
C             PDOP  =  SQRT( TRACE COV(X , Y , Z))
C             HTDOP =  SQRT( TRACE COV(X , Y , CLOCK) )
C             HDOP  =  SQRT( TRACE COV(X , Y))
C
C ------------------------------------------------------------------------
C
   21 CALL DOP2S(A,B,IDA,VISAT,MXVS,C,HDOP)
      IZCNT = IDINT(ZCNT1)
      TOFDAY = ZFRDAY( ZCNT1 ) * 24.DO
      MREC = MREC + 1
      WRITE(IPR1,92) MREC,KHR,KMIN,IZCNT,(IVIS(N),N=1,NVIS),
     #               (ELEBUF(N),N=1,NVIS),(AZBUF(N),N=1,NVIS),
     #               HDOP
      WRITE(IPR ,93) KHR,KMIN,IZCNT,(IVIS(N),N=1,NVIS),
     #               (ELEBUF(N),N=1,NVIS),(AZBUF(N),N=1,NVIS),
     #               HDOP
      WRITE(IPR2,97)  TOFDAY,HDOP,PDOP,HTDOP,GDOP
      GO TO 25
C
   22 CALL DOP3S(A,B,IDA,VISAT,MXVS,C,HTDOP,PDOP,HDOP)
      IZCNT = IDINT(ZCNT1)
      MREC = MREC + 1
      WRITE(IPR,90) KHR,KMIN,IZCNT,(IVIS(N),N=1,NVIS),
     #              (ELEBUF(N),N=1,NVIS),(AZBUF(N),N=1,NVIS),PDOP,HTDOP
      TOFDAY = ZFRDAY( ZCNT1 ) * 24.DO
      WRITE(IPR1,95) MREC,KHR,KMIN,IZCNT,(IVIS(N),N=1,NVIS),
     #               (ELEBUF(N),N=1,NVIS),(AZBUF(N),N=1,NVIS),
     #               HDOP,PDOP,HTDOP
      WRITE(IPR2,97)  TOFDAY,HDOP,PDOP,HTDOP,GDOP
      GO TO 25
C
C ------------------------------------------------------------------------
C
C        FOR THE FOUR SATELLITES WHICH GIVE THE MAXIMUM
C        VOLUME OF THE THE FUNDAMENTAL TETRAHEDRON COMPUTE
C
C             GDOP  =  SQRT( TRACE COV(X , Y , Z , CLOCK) )
C             PDOP  =  SQRT( TRACE COV(X , Y , Z))
C             HTDOP =  SQRT( TRACE COV(X , Y , CLOCK) )
C             HDOP  =  SQRT( TRACE COV(X , Y))
C
C ------------------------------------------------------------------------
C
   23 CALL DOP4S(A,B,IDA,VISAT,MXVS,C,KK,GDOP,HTDOP,PDOP,HDOP)
      MREC = MREC + 1
      IZCNT = IDINT(ZCNT1)
      M=J-3
      GO TO(514,515,516,517,518,519) ,M
  514 WRITE(IPR,524) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #               (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #               (IBEST(N),N=1,4),GDOP
      GO TO 26
  515 WRITE(IPR,525) KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #               (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #               (IBEST(N),N=1,4),GDOP
```

```
      GO TO 26
  516 WRITE(IPR,526)  KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                (IBEST(N),N=1,4),GDOP
      GO TO 26
  517 WRITE(IPR,527)  KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                (IBEST(N),N=1,4),GDOP
      GO TO 26
  518 WRITE(IPR,528)  KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                (IBEST(N),N=1,4),GDOP
      GO TO 26
  519 WRITE(IPR,529)  KHR,KMIN,IZCNT,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                (IBEST(N),N=1,4),GDOP
   26 CONTINUE
      TOFDAY = ZFRDAY(ZCNT1) * 24.D0                                    00310**3
      WRITE(IPR1,96) MREC,KHR,KMIN,IZCNT,(IVIS(N),N=1,NVIS),            00311**4
     #                (ELEBUF(N),N=1,NVIS),(AZBUF(N),N=1,NVIS),         00312**3
     #                HDOP,PDOP,HTDOP,GDOP                              00313**3
      WRITE(IPR2,97)   TOFDAY,HDOP,PDOP,HTDOP,GDOP                      00314**6
      WRITE(LU,88) ZCNT1,(IVIS(I),I=1,MXVS),(IBEST(J),J=1,4)                 **6
      GO TO 25
C-----------------------------------------------------------------------
C
C     DILUTION OF PRECISION FOR ALL SATELLITE
C
C-----------------------------------------------------------------------
   24 CALL DOPS(A,B,IDA,VISAT,MXVS,J,C,GDOP,PDOP,HTDOP,HDOP,VDOP)
      GO TO(801,802,803,804,805,806,807,808,809) ,J
  801 WRITE(IPR,811)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                VDOP
      GO TO 25
  802 WRITE(IPR,812)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                HDOP,VDOP
      GO TO 25
  803 WRITE(IPR,813)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  804 WRITE(IPR,814)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                 GDOP,PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  805 WRITE(IPR,815)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                 GDOP,PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  806 WRITE(IPR,816)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                 GDOP,PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  807 WRITE(IPR,817)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                 GDOP,PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  808 WRITE(IPR,818)  KHR,KMIN,(IVIS(N),N=1,J),
     #                (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
```

```
     #                GDOP,PDOP,HTDOP,HDOP,VDOP
      GO TO 25
  809 WRITE(IPR,819) KHR,KMIN,(IVIS(N),N=1,J),
     #               (ELEBUF(N),N=1,J),(AZBUF(N),N=1,J),
     #                GDOP,PDOP,HTDOP,HDOP,VDOP
   25 CONTINUE
      GDOP = 0.
      PDOP = 0.
      HTDOP = 0.
      HDOP = 0.
      VDOP = 0.
  153 CONTINUE
C
C
C-------------------------------------------------------------------
C       ITERATION FOR THE SPECIFIED TIME INCREMENT OF (IDT) MINUTES
C-------------------------------------------------------------------
C
      IF(KYEAR.GE.IYEAR.AND.KDAY.GE.IDAY.AND.KHR.GE.IHR
     #  .AND.KMIN.GE.IMIN)  GO TO 2221
         KMIN = KMIN  + IDT
         IF(KMIN.LT.60) GO TO 14
             KMIN = KMIN  - 60
             KHR  = KHR   + 1
             IF(KHR.LT.24) GO TO 14
                  KDAY  = KDAY  + 1
                  KHR   = KHR   - 24
             GO TO 14
C----------------------------------------------------------------------
C     VISIBILITY SUMMARY FOR GPS SATELLITES
C
C LIMITATIONS: START TIME MUST BEGIN AT A INTEGER HOUR EX: 12 00 00
C              INCREMENT TIME MUST BE (1 OR 3 OR 5 OR 15 OR 30 OR 60)MIN
C----------------------------------------------------------------------
 2221 IF(IPARA1 .NE. 1) GO TO 2222
      WRITE(IPR,5100) KELEV
      WRITE(IPR,5110)LATD,LATM,LATS,LOND,LONM,LONS
      WRITE(IPR,5111) KRHT
      WRITE(IPR,5120)KKDAY,KKYEAR,KKHR,KKMIN,KKSEC,ZZZ
      WRITE(IPR,5130)IDAY,IYEAR,IHR,IMIN,ISEC,ZCNT2
      DO 5000 J=1,NSAT
      WRITE(IPR,5002) IDSAT(J)
      DO 5001 K=1,IPP
      ICRL(K)=IBLANK
      IF(IHRVIS(J,K) .EQ. 1) ICRL(K)=ISTAR
 5001 CONTINUE
      WRITE(IPR,5601) (ICRL(K),K=1,IPP)
 5601 FORMAT('+',12X,100A1)
 5000 CONTINUE
      NFH=INT(IPP/4)+1
      DO 5900 L=1,NFH
      KKKR=KKHR+L-1
      IF(IDT .EQ. 30) KKKR=KKKR*2-KKHR
      IF(IDT .EQ. 60) KKKR=KKKR*4-KKHR*3
      IHH(L)=KKKR
      IF(IHH(L) .GE. 24) IHH(L)=IHH(L)-24
 5900 CONTINUE
      WRITE(IPR,5600) (IHH(II),II=1,NFH)
 5100 FORMAT('1',7X,'VISIBILITY (ELEVATION > ',I2,' DEGREES) OF GPS',
     #       ' SATELLITES',///)
 5002 FORMAT('0',5X,'GPS',I3)
```

```
 5600 FORMAT(10X,25I4)
 5110 FORMAT(9X,'LAT=',I3,'DEG',2X,I3,'MIN',2X,I3,'SEC',                      0340
     #  /,(7X,' LONG=',I3,'DEG',2X,I3,'MIN',2X,I3,'SEC'))                      0341
 5111 FORMAT(11X,'H=',I3,' METRES',/)
 5120 FORMAT(8X,'BEGIN ALERT:  DAY ',I4,' , ',I5,4X,                           0343**2
     #           2(I2,':'),I2,'   ZCNT1 = ',F7.0,/)                            00344**2
 5130 FORMAT(8X,'  END ALERT:  DAY ',I4,' , ',I5,4X,                           0345**2
     #           2(I2,':'),I2,'   ZCNT2 = ',F7.0,/////)                        00346**2
C
      GO TO 2222
C
    1 FORMAT(6I4,F6.2,F3.0)                                                    00337
    2 FORMAT('1',44X,'YOUR LOCATION AND DESIRED TIME INTERVAL FOR ALERT 00338
     # ',/)                                                                    00339
    3 FORMAT(25X,'LAT=',I3,'DEG',2X,I3,'MIN',2X,I3,'SEC',                      00340
     #  /,(23X,' LONG=',I3,'DEG',2X,I3,'MIN',2X,I3,'SEC'))                     00341
 7000 FORMAT(27X,'H=',I3,' METRES',/)
    4 FORMAT(5I4)                                                              00342
    5 FORMAT(14X,'BEGIN ALERT:  DAY ',I4,' , ',I5,4X,                          00343**2
     #           2(I2,':'),I2,'   ZCNT1 = ',F7.0,/)                            00344**2
    6 FORMAT(14X,'  END ALERT:  DAY ',I4,' , ',I5,4X,                          00345**2
     #           2(I2,':'),I2,'   ZCNT2 = ',F7.0,///)                          00346**2
    7 FORMAT(I2)                                                               00347
    8 FORMAT(19I2)                                                             00348
   12 FORMAT(1X,'HR MIN ZCNT',7X,' VISIBLE SATELLITES',                        00349**2
     #      12X,'ELEVATIONS',22X,'AZIMUTHS',27X,'SELECT',4X,'GDOP',            00350**2
     @        /,117X,'(PDOP)',2X,'(HTDOP)',/,117X,'(HDOP)',//)                 00351**2
   88 FORMAT(F8.0,13I3)                                                             **6
   90 FORMAT(2I3,I6,2(1X,3I3,18X),3I5,34X,F5.1,3X,F5.1/)                       00354**2
   92 FORMAT(10X,I4,2I3,I6,2(1X,2I3,8X),2I5,13X,F6.1,18X,/)                    00355**5
   93 FORMAT(2I3,I6,2(1X,2I3,21X),2I5,39X,F5.1,/)                              00356**3
   94 FORMAT(15X,'HR MIN ZCNT',3X,'SATELLITES',5X,                             00357**5
     #    'ELEVATIONS',6X,'AZIMUTHS',11X,'  HDOP  PDOP  HTDOP GDOP',//)        00358**5
   95 FORMAT(10X,I4,2I3,I6,2(1X,3I3,5X),3I5,8X,3F6.1,6X,/)                     00359**5
   96 FORMAT(10X,I4,2I3,I6,2(1X,4I3,2X),4I5,3X,4F6.1,/)                        00360**5
   97 FORMAT(F10.5,4F7.1)                                                           **6
  130      FORMAT(5X,G18.9,G18.9)                                              00362
 1000      FORMAT(5X,9G14.8)                                                   00363
 1030 FORMAT(//,5X,'SAT #',I3,6(/,5X,'CLK(',I1,')=',E20.10),                   00364
     #        /,18(/,5X,'EPH(',I2,')=',E20.10))                                00365
 1040 FORMAT(5X,'***  WARNING :  # VISIBLE SATELLITES ',I4,                    00366
     #         '  >  # OF MAXIMUM VISIBLE SATELLITES EXPECTED ',I4,            00367
     #         /,10X,'INCREASE  MXVS  AND THE ROW DIMENSIONS OF ',             00368**4
     #        'THE APPROPRIATE ARRAYS TO AT LEAST',I4,'   ***',/)              00369**4
   35 FORMAT(14X,'BEGIN ALERT:  DAY ',I4,' , ',I5,4X,2(I2,':'),I2,/)
   36 FORMAT(14X,'  END ALERT:  DAY ',I4,' , ',I5,4X,2(I2,':'),I2,///)
   37 FORMAT(15X,'HR MIN',6X,'SATELLITE(S)',4X,'ELEVATION(S)',
     #       6X,'AZIMUTH(S)',//)
  134 FORMAT(3I2)
  150 FORMAT(5X,'WARNING:  VISIBLE SATELLITES GREATER THAN 3')
  156 FORMAT(63X,'GPS SATELLITES',/)
  161 FORMAT(61X,'TRANSIT SATELLITES',/)
  166 FORMAT(61X,'LAGEOS SATELLITE(S)',/)
  171 FORMAT(60X,'STARLETTE SATELLITE(S)',/)
 8000 FORMAT(53X,'SATELLITE ELEVATION > ',I2,' DEGREES',//)
  311 FORMAT(56X,'NNSS NODAL CROSSING DATA',/)
  321 FORMAT(45X,'NNSS KEPLERIAN PARAMETERS OR MEAN ORBIT TRACKED',/)
  331 FORMAT(62X,'GPS EPHEMERIS',/)
  341 FORMAT(50X,'NASA PREDICTION BULLETIN  (GAST USED)',/)
  351 FORMAT(49X,'NASA PREDICTION BULLETIN  (GAST UNUSED)',/)
  461 FORMAT(2I3,I6,1X,I3,30X,I3,29X,I5,/)
```

```
 462 FORMAT(2I3,I6,1X,2I3,27X,2I3,26X,2I5,/)
 463 FORMAT(2I3,I6,1X,3I3,24X,3I3,23X,3I5,/)
 464 FORMAT(2I3,I6,1X,4I3,21X,4I3,20X,4I5,/)
 465 FORMAT(2I3,I6,1X,5I3,18X,5I3,17X,5I5,/)
 466 FORMAT(2I3,I6,1X,6I3,15X,6I3,14X,6I5,/)
 467 FORMAT(2I3,I6,1X,7I3,12X,7I3,11X,7I5,/)
 468 FORMAT(2I3,I6,1X,8I3,9X,8I3,8X,8I5,/)
 469 FORMAT(2I3,I6,1X,9I3,6X,9I3,5X,9I5,/)
 481 FORMAT(14X,2I3,7X,I3,13X,I3,14X,I5)
 482 FORMAT(14X,2I3,7X,2I3,10X,2I3,11X,2I5)
 483 FORMAT(14X,2I3,2(7X,3I3),8X,3I5)
 524 FORMAT(2I3,I6,1X,4I3,16X,4I3,15X,4I5,25X,4I3,F5.1,/)
 525 FORMAT(2I3,I6,1X,5I3,13X,5I3,12X,5I5,20X,4I3,F5.1,/)
 526 FORMAT(2I3,I6,1X,6I3,10X,6I3,9X,6I5,15X,4I3,F5.1,/)
 527 FORMAT(2I3,I6,1X,7I3,7X,7I3,6X,7I5,10X,4I3,F5.1,/)
 528 FORMAT(2I3,I6,1X,8I3,4X,8I3,3X,8I5,5X,4I3,F5.1,/)
 529 FORMAT(2I3,I6,2(1X,9I3),9I5,4I3,F5.1,/)
 601 FORMAT(1X,'HR MIN ZCNT',6X,' VISIBLE SATELLITES',18X,
    #        'ELEVATIONS',32X,'AZIMUTHS',//)
 811 FORMAT(2I3,1X,I3,25X,I3,24X,I5,60X,F5.1,/)
 812 FORMAT(2I3,1X,2I3,22X,2I3,21X,2I5,50X,2F5.1,/)
 813 FORMAT(2I3,1X,3I3,19X,3I3,18X,3I5,35X,4F5.1,/)
 814 FORMAT(2I3,1X,4I3,16X,4I3,15X,4I5,25X,5F5.1,/)
 815 FORMAT(2I3,1X,5I3,13X,5I3,12X,5I5,20X,5F5.1,/)
 816 FORMAT(2I3,1X,6I3,10X,6I3,9X,6I5,15X,5F5.1,/)
 817 FORMAT(2I3,1X,7I3,7X,7I3,6X,7I5,10X,5F5.1,/)
 818 FORMAT(2I3,1X,8I3,4X,8I3,3X,8I5,5X,5F5.1,/)
 819 FORMAT(2I3,2(1X,9I3),9I5,5F5.1,/)
 820 FORMAT(1X,'HR MIN',6X,' VISIBLE SATELLITES',                  00349**2
    #      12X,'ELEVATIONS',22X,'AZIMUTHS',24X,'GDOP PDOP HTDP HDOP'  00350**2
    @      ,' VDOP')                                                  00351**2
 901 FORMAT(1X,122('-'),/)
 902 FORMAT(14X,61('-'))
 903 FORMAT(1X,129('-'),/)
1117 FORMAT(2I3,9I2,9F6.2,9F8.2)
2222 STOP                                                             00370
     END                                                              00371
```

```
      SUBROUTINE DOPS(A,B,IDA,VISAT,MXVS,J,C,GDOP,PDOP,HTDOP,HDOP,VDOP)
C
C   NAME           DOPS
C
C   PURPOSE        COMPUTE DILUTION OF PRECISION FACTORS
C
C   VERSION        MARCH    1984  -  R. SANTERRE
C
C
C   PARAMETERS     I   A , B   -  WORKING ARRAYS OF DECLARED ROW
C                                 IDA .GE. 4
C                  I   VISAT   -  ARRAY OF DIRECTION COSINES
C                                 TO THE VISIBLE SATELLITES
C                                 IN THE TANGENTIAL SYSTEM (NORTH,EAST,HGT)
C                  I   MXVS    -  DECLARED ROW DIMENSION OF 'VISAT'
C                  I   J       -  NUMBER OF VISIBLE SATELLITES
C                  I   C       -  SPEED OF LIGHT
C                  I   KK      -  VECTOR OF POSITION INDICATORS WITHIN
C                                 'VISAT' FOR THE FOUR SELECTED SATELLITES
C
C                  O   GDOP    -  SQRT( TRACE COV(N,E,H,CLOCK))
C                  O   PDOP    -  SQRT( TRACE COV(N,E,H))
C                  O   HTDOP   -  SQRT( TRACE COV(N,E,CLOCK))
C                  O   HDOP    -  SQRT( TRACE COV(N,E))
C                  O   VDOP    -  SQRT( TRACE COV(H))
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(IDA,1) , B(IDA,1) , VISAT(MXVS,1)
C
C ----------------------------------------------------------------------
C          COMPUTE 4-D DILUTION OF PRECISION FACTOR GDOP
C ----------------------------------------------------------------------
C
      IF(J .LT. 4) GO TO 100
      MA = J
      DO 10 I = 1,MA
      A(I,1) = - VISAT( I , 1 )
      A(I,2) = - VISAT( I , 2 )
      A(I,3) = - VISAT( I , 3 )
      A(I,4) =   C
   10 CONTINUE
C
      CALL MATMPY(A,A,B,4,MA,4,IDA,IDA,IDA,2)
      CALL SPIN(B,4,IDA,DET,IEXP)
C
      GDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) + B(4,4) * C*C )
  100 CONTINUE
C
C ----------------------------------------------------------------------
C          COMPUTE 3-D DILUTION OF PRECISION FACTORS
C ----------------------------------------------------------------------
C
C
C                      COMPUTE PDOP FACTOR FIRST
C
      IF(J .LT. 3) GO TO 200
      MA=J
      DO 15 I = 1,MA
      A(I,1) = - VISAT( I , 1 )
      A(I,2) = - VISAT( I , 2 )
```

```
      A(I,3) = - VISAT( I , 3 )
   15 CONTINUE
C
      CALL MATMPY(A,A,B, 3,MA, 3,IDA,IDA,IDA,2)
      CALL SPIN(B, 3,IDA,DET,IEXP)
C
      PDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) )
C
C                      THEN HTDOP FACTOR
C
      DO 20 I = 1,MA
      A(I,1) = - VISAT( I , 1 )
      A(I,2) = - VISAT( I , 2 )
      A(I,3) =    C
   20 CONTINUE
C
      CALL MATMPY(A,A,B, 3,MA, 3,IDA,IDA,IDA,2)
      CALL SPIN(B, 3,IDA,DET,IEXP)
C
      HTDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) * C*C )
  200 CONTINUE
C
C
C ---------------------------------------------------------------------
C        COMPUTE 2-D DILUTION OF PRECISION FACTOR HDOP
C ---------------------------------------------------------------------
C
      IF(J .LT. 2) GO TO 300
      MA=J
      DO 25 I = 1,MA
      A(I,1) = - VISAT( I , 1 )
      A(I,2) = - VISAT( I , 2 )
   25 CONTINUE
C
      CALL MATMPY(A,A,B, 2,MA, 2,IDA,IDA,IDA,2)
      CALL SPIN(B, 2,IDA,DET,IEXP)
C
      HDOP = DSQRT( B(1,1) + B(2,2) )
  300 CONTINUE
C
C----------------------------------------------------------------------
C        COMPUTE 1-D DILUTION OF PRECISION FACTOR VDOP
C----------------------------------------------------------------------
      MA=J
      DO 30 I=1,MA
      A(I,1) = - VISAT(I,3)
   30 CONTINUE
      CALL MATMPY(A,A,B,1,MA,1,IDA,IDA,IDA,2)
      VDOP=DSQRT(1.DO/B(1,1))
      RETURN
      END
```

```
      SUBROUTINE DOP4S(A,B,IDA,VISAT,MXVS,C,KK,GDOP,HTDOP,PDOP,HDOP)
C
C    NAME           DOP4S
C
C    PURPOSE        COMPUTE DILUTION OF PRECISION FACTORS
C                   FROM FOUR SATELLITES
C
C    VERSION        FEBRUARY 1983  -  D. DELIKARAOGLOU
C
C                   THIS VERSION USES THE FOUR SATELLITES WHICH
C                   GIVE THE MAXIMUM VOLUME OF THE FUNDAMENTAL
C                   TETRAHEDRON AND COMPUTES GDOP , HTDOP ,
C                   PDOP AND HDOP
C
C    PARAMETERS     I   A , B   -  WORKING ARRAYS OF DECLARED ROW
C                                  IDA .GE. 4
C                   I   VISAT   -  ARRAY OF DIRECTION COSINES
C                                  TO THE VISIBLE SATELLITES
C                                  IN THE TANGENTIAL SYSTEM (NORTH,EAST,HGT)
C                   I   MXVS    -  DECLARED ROW DIMENSION OF 'VISAT'
C                   I   C       -  SPEED OF LIGHT
C                   I   KK      -  VECTOR OF POSITION INDICATORS WITHIN
C                                  'VISAT' FOR THE FOUR SELECTED SATELLITES
C
C                   O   GDOP    -  SQRT( TRACE COV(N,E,H,CLOCK))
C                   O   HTDOP   -  SQRT( TRACE COV(N,E,CLOCK))
C                   O   PDOP    -  SQRT( TRACE COV(N,E,H))
C                   O   HDOP    -  SQRT( TRACE COV(N,E))
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(IDA,1) , B(IDA,1) , VISAT(MXVS,1) , KK(1)
C
C ---------------------------------------------------------------------
C        COMPUTE 4-D DILUTION OF PRECISION FACTOR GDOP
C ---------------------------------------------------------------------
C
      MA = 4
      DO 10 I = 1,MA
      A(I,1) = - VISAT( KK(I),1 )
      A(I,2) = - VISAT( KK(I),2 )
      A(I,3) = - VISAT( KK(I),3 )
      A(I,4) =    C
   10 CONTINUE
C
      CALL MATMPY(A,A,B,MA,MA,MA,IDA,IDA,IDA,2)
      CALL SPIN(B,MA,IDA,DET,IEXP)
C
      GDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) + B(4,4) * C*C )
C
C ---------------------------------------------------------------------
C        COMPUTE 3-D DILUTION OF PRECISION FACTORS
C ---------------------------------------------------------------------
C
C
C                    COMPUTE PDOP FACTOR FIRST
C
      DO 15 I = 1,MA
      A(I,1) = - VISAT( KK(I),1 )
      A(I,2) = - VISAT( KK(I),2 )
      A(I,3) = - VISAT( KK(I),3 )
```

```
   15 CONTINUE
C
      CALL MATMPY(A,A,B, 3,MA, 3,IDA,IDA,IDA,2)
      CALL SPIN(B, 3,IDA,DET,IEXP)
C
      PDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) )
C
C                     THEN HTDOP FACTOR
C
      DO 20 I = 1,MA
      A(I,1) = - VISAT( KK(I),1 )
      A(I,2) = - VISAT( KK(I),2 )
      A(I,3) =    C
   20 CONTINUE
C
      CALL MATMPY(A,A,B, 3,MA, 3,IDA,IDA,IDA,2)
      CALL SPIN(B, 3,IDA,DET,IEXP)
C
      HTDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) * C*C )
C
C
C ---------------------------------------------------------------------
C        COMPUTE 2-D DILUTION OF PRECISION FACTOR HDOP
C ---------------------------------------------------------------------
C
      DO 25 I = 1,MA
      A(I,1) = - VISAT( KK(I),1 )
      A(I,2) = - VISAT( KK(I),2 )
   25 CONTINUE
C
      CALL MATMPY(A,A,B, 2,MA, 2,IDA,IDA,IDA,2)
      CALL SPIN(B, 2,IDA,DET,IEXP)
C
      HDOP = DSQRT( B(1,1) + B(2,2) )
      RETURN
      END
```

```
      SUBROUTINE DOP3S(A,B,IDA,VISAT,MXVS,C,HTDOP,PDOP,HDOP)
C
C    NAME            DOP3S
C
C    PURPOSE         COMPUTE DILUTION OF PRECISION FACTORS
C                    FROM THREE SATELLITES
C
C    VERSION         FEBRUARY 1983  -  D. DELIKARAOGLOU
C
C    PARAMETERS      I    A , B   -  WORKING ARRAYS OF DECLARED ROW
C                                    IDA .GE. 3
C                    I    VISAT   -  ARRAY OF DIRECTION COSINES
C                                    TO THE VISIBLE SATELLITES
C                                    IN THE TANGENTIAL SYSTEM(NORTH,EAST,HGT)
C                    I    MXVS    -  DECLARED ROW DIMENSION OF 'VISAT'
C                    I    C       -  SPEED OF LIGHT
C
C                    O    HTDOP   -  SQRT( TRACE COV(N,E,CLOCK))
C                    O    PDOP    -  SQRT( TRACE COV(N,E,H))
C                    O    HDOP    -  SQRT( TRACE COV(N,E))
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(IDA,1) , B(IDA,1) , VISAT(MXVS,1)
C
      MA = 3
      DO 10 I = 1,MA
      A(I,1) = - VISAT( I,1 )
      A(I,2) = - VISAT( I,2 )
      A(I,3) =   C
   10 CONTINUE
C
      CALL MATMPY(A,A,B,MA,MA,MA,IDA,IDA,IDA,2)
      CALL SPIN(B,MA,IDA,DET,IEXP)
      HTDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) * C*C )
C
      DO 15 I = 1,MA
      A(I,1) = - VISAT( I,1 )
      A(I,2) = - VISAT( I,2 )
      A(I,3) = - VISAT( I,3 )
   15 CONTINUE
C
      CALL MATMPY(A,A,B,MA,MA,MA,IDA,IDA,IDA,2)
      CALL SPIN(B,MA,IDA,DET,IEXP)
      PDOP = DSQRT( B(1,1) + B(2,2) + B(3,3) )
C
C ---------------------------------------------------------------------
C                   COMPUTE 2-D DILUTION OF PRECISION FACTOR HDOP
C ---------------------------------------------------------------------
C
      DO 20 I = 1,MA
      A(I,1) = - VISAT( I,1 )
      A(I,2) = - VISAT( I,2 )
   20 CONTINUE
C
      CALL MATMPY(A,A,B, 2,MA, 2,IDA,IDA,IDA,2)
      CALL SPIN(B, 2,IDA,DET,IEXP)
C
      HDOP = DSQRT( B(1,1) + B(2,2) )
      RETURN
      END
```

```
      SUBROUTINE DOP2S(A,B,IDA,VISAT,MXVS,C,HDOP)                       00528**3
C                                                                       00529**3
C    NAME          DOP2S                                                00530**3
C                                                                       00531**3
C    PURPOSE       COMPUTE DILUTION OF PRECISION FACTOR                 00532**3
C                  FROM TWO SATELLITES                                  00533**3
C                                                                       00534**3
C    VERSION       FEBRUARY 1983  -  D. DELIKARAOGLOU                   00535**3
C                                                                       00536**3
C    PARAMETERS    I   A , B   -  WORKING ARRAYS OF DECLARED ROW        00537**3
C                                 IDA .GE. 3                            00538**3
C                  I   VISAT   -  ARRAY OF DIRECTION COSINES            00539**3
C                                 TO THE VISIBLE SATELLITES             00540**3
C                                 IN THE TANGENTIAL SYSTEM (NORTH,EST,HGT)
C                  I   MXVS    -  DECLARED ROW DIMENSION OF 'VISAT'     00541**3
C                  I   C       -  SPEED OF LIGHT                        00542**3
C                                                                       00543**3
C                  O   HDOP    -  SQRT( TRACE COV(N,E))                 00544**3
C                                                                       00545**3
C                                                                       00546**3
      IMPLICIT REAL*8(A-H,O-Z)                                          00547**3
      DIMENSION A(IDA,1) , B(IDA,1) , VISAT(MXVS,1)                     00548**3
C                                                                       00549**3
      MA = 2                                                            00550**3
      DO 10 I = 1,MA                                                    00551**3
      A(I,1) = - VISAT( I,1 )                                           00552**3
      A(I,2) = - VISAT( I,2 )                                           00553**3
   10 CONTINUE                                                          00554**3
C                                                                       00555**3
      CALL MATMPY(A,A,B,MA,MA,MA,IDA,IDA,IDA,2)                         00556**3
      CALL SPIN(B,MA,IDA,DET,IEXP)                                      00557**3
      HDOP = DSQRT( B(1,1) + B(2,2) )                                   00558**3
C                                                                       00559**3
      RETURN                                                            00560**3
      END                                                               00561**3
```

```
      SUBROUTINE VOLUME(KSAT,VISAT,MXVS,NN,IDSAT,VOL)
C
C    NAME          VOLUME
C
C    PURPOSE       TO SELECT THE SET OF FOUR GPS SATELLITES
C                  WHICH PROVIDE BEST NAVIGATION GEOMETRY
C                  USING THE MAXIMUM TETRAHEDRON CRITERION
C
C    PARAMETERS
C
C                 KSAT      I     POSITION OF THE SATELLITE WITH
C                                 MAXIMUM ELEVATION.
C                 VISAT     I     MATRIX CONTAINING DIRECTION
C                                 COSINES OF ALL VISIBLE SAT.
C                 MXVS      I     MAXIMUM POSSIBLE NUMBER OF VISIBLE
C                                 SATELLITES (EQUAL TO THE DECLARED ROW
C                                 DIMENSION OF MATRIX 'VISAT' IN CALLING
C                                 PROGRAM)
C                 NN        I     NUMBER OF VISIBLE SATELLITES
C                 IDSAT     O     ID OF VISIBLE SATELLITES WHICH
C                                 PROVIDE BEST NAVIGATION PERFORMANCE.
C                 VOL       O     VOLUME OF THE TETRAHEDRON FORMED BY
C                                 UNIT VECTORS TO EACH SATELLITE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION VISAT(MXVS,3) , A(3), B(3) ,  C(3) , IDSAT(4)
      N1= NN - 1
      N2= NN - 2
      VOLMX=0.D0
C
      DO 1 I=1,N2
      IF(I.EQ.KSAT) GO TO 1
      K= I + 1
          DO 2 J= K ,N1
              IF(J.EQ.KSAT) GO TO 2
              L= J + 1
                DO 3 M= L, NN
                IF(M.EQ.KSAT) GO TO 3
C
      DO 4 II= 1 , 3
      A(II)= VISAT(I,II) - VISAT(KSAT,II)
      B(II)= VISAT(J,II) - VISAT(KSAT,II)
      C(II)= VISAT(M,II) - VISAT(KSAT,II)
4     CONTINUE
C
      VOL= (A(1)*(B(2)*C(3) - B(3)*C(2))
     #   -  A(2)*(B(1)*C(3) - B(3)*C(1))
     #   +  A(3)*(B(1)*C(2) - B(2)*C(1))) / 6.
C
      IF(DABS(VOL).GT.VOLMX) GO TO 10
      GO TO 11
10    VOLMX= DABS(VOL)
      IDSAT(1)= KSAT
      IDSAT(2)= J
      IDSAT(3)= M
      IDSAT(4)= I
11    CONTINUE
3     CONTINUE
2     CONTINUE
1     CONTINUE
      VOL= VOLMX
```

```
RETURN
END
```

```
      SUBROUTINE TIME(IYEARO,IDAYO,IHO,IMO,ISO,IYEART,IDAYT,
     #                IHT,IMT,IST,ITIME)
C----------------------------------------------------------------------
C     NAME:     TIME
C     PURPOSE:  TO COMPUTE THE NUMBER OF SECONDS BETWEEN TWO
C               GIVEN INSTANTS
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  IYEARO:      YEAR OF THE INITIAL INSTANT
C        I  IDAYO:       DAY OF THE INITIAL INSTANT
C        I  IHO:         HOUR OF THE INITIAL INSTANT
C        I  IMO:         MINUTE OF THE INITIAL INSTANT
C        I  ISO:         SECOND OF THE INITIAL INSTANT
C        I  IYEART:      YEAR OF THE FINAL INSTANT
C        I  IDAYT:       DAY OF THE FINAL INSTANT
C        I  IHT:         HOUR OF THE FINAL INSTANT
C        I  IMT:         MINUTE OF THE FINAL INSTANT
C        I  IST:         SECOND OF THE FINAL INSTANT
C
C        O  ITIME:       NUMBER OF SECONDS
C
C     COMMENT: IYEART-IYEARO  EQUAL  -1, 0, OR 1
C----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     -----------------------------------------------------------
C     TEST IF THE FINAL TIME PRECEDES OR FOLLOWS THE INITIAL TIME
C     -----------------------------------------------------------
      IF(IYEART-IYEARO) 10,20,30
C
   10 IYEAR=IYEART
C
C     ----------------------------------------------------
C     TEST IF IYEART IS LEAP AND COMPUTE THE NUMBER OF DAYS
C     ----------------------------------------------------
      IF(IYEAR/4 .NE. IYEAR/4.) GO TO 100
      IDD=-366+IDAYT-IDAYO
      GO TO 40
  100 IDD=-365+IDAYT-IDAYO
      GO TO 40
C
C     -------------------------
C     COMPUTE THE NUMBER OF DAYS
C     -------------------------
   20 IDD=IDAYT-IDAYO
      GO TO 40
C
   30 IYEAR=IYEARO
C
C     ----------------------------------------------------
C     TEST IF IYEARO IS LEAP AND COMPUTE THE NUMBER OF DAYS
C     ----------------------------------------------------
      IF(IYEAR/4 .NE. IYEAR/4.) GO TO 200
      IDD=366+IDAYT-IDAYO
      GO TO 40
  200 IDD=365+IDAYT-IDAYO
C
C     -------------------------------------------------------
```

```
C     COMPUTE THE NUMBER OF SECONDES BETWEEN THE TWO INSTANTS
C     -------------------------------------------------------
   40 IHH=IDD*24+IHT-IHO
      IMM=IHH*60+IMT-IMO
      ITIME=IMM*60+IST-ISO
C
      RETURN
      END
```

```
      SUBROUTINE DAYFRA(DAYF,IDAY,IH,IM,IS)
C-----------------------------------------------------------------------
C     NAME:     DAYFRA
C     PURPOSE: TO TRANSFORM DAY AND FRACTION OF DAY
C              IN DAY,HOUR,MINUTE AND SECOND
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  DAYF:        DAY AND FRACTION OF DAY
C
C        O  IDAY:        DAY
C        O  IH:          HOUR
C        O  IM:          MINUTE
C        O  IS:          SECONDE
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      IDAY=IDINT(DAYF)
      R=(DAYF-DFLOAT(IDAY))*24.D0
      IH=IDINT(R)
      U=(R-DFLOAT(IH))*60.D0
      IM= IDINT(U)
      T=(U-DFLOAT(IM))*60.D0
      IS=IDINT(T)
      IF((T-IS) .GE. .5) IS=IS+1
C
      RETURN
      END
```

```
      SUBROUTINE MINFRA(XMINF,IH,IM,IS)
C-----------------------------------------------------------------------
C     NAME:     MINFRA
C     PURPOSE:  TO TRANSFORM MINUTE AND FRACTION OF MINUTE
C               IN HOUR,MINUTE AND SECOND
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  XMINF:       MINUTE AND FRACTION OF MINUTE
C
C        O  IH:          HOUR
C        O  IM:          MINUTE
C        O  IS:          SECOND
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      R=XMINF/60.DO
      IH=IDINT(R)
C
      U=(R-DFLOAT(IH))*60.DO
      IM=IDINT(U)
C
      T=(U-DFLOAT(IM))*60.DO
      IS=IDINT(T)
      IF((T-IS) .GE. .5) IS=IS+1
C
      RETURN
      END
```

```
      SUBROUTINE NNSSNC(IY,MDAY,TNODE,XLNODE,PERIOD,WMOTN,KYEAR,
     #                  KDAY,KHR,KMIN,KSEC,PI,XS,YS,ZS)
C----------------------------------------------------------------------
C     NAME:     NNSSNC
C     PURPOSE:  TO COMPUTE THE SATELLITE COORDINATES IN THE EARTH FIXED
C                SYSTEM FROM NNSS NODAL CROSSING DATA
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  IY:          INPUT YEAR
C        I  MDAY:        INPUT DAY
C        I  TNODE:       TIME OF ASCENDING NODE (MINUTES)
C        I  XLNODE:      LONGITUDE (EAST) OF ASCENDING NODE (DEGREES)
C        I  PERIOD:      NODAL PERIOD (MINUTES)
C        I  WMOTN:       WESTWARD MOTION (DEGREES)
C        I  KYEAR:       YEAR OF THE ALERT
C        I  KDAY:        DAY OF THE ALERT
C        I  KHR:         HOUR OF THE ALERT
C        I  KMIN:        MINUTE OF THE ALERT
C        I  KSEC:        SECOND OF THE ALERT
C        I  PI:          VALUE OF PI
C
C        O  XS:          SATELLITE X COORDINATE (METRES)
C        O  YS:          SATELLITE Y CORRDINATE (METRES)
C        O  ZS:          SATELLITE Z COORDINATE (METRES)
C----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---------------------------------------------
C     TRANSFORM TNODE IN HOURS,MINUTES AND SECONDS
C     ---------------------------------------------
      CALL MINFRA(TNODE,IH,IM,IS)
C
C     ----------------------------------------------------------------
C     COMPUTE THE NUMBER OF SECONDES BETWEEN THE TIME OF ASCENDING NODE
C     AND THE TIME OF THE ALERT
C     ----------------------------------------------------------------
      CALL TIME(IY,MDAY,IH,IM,IS,KYEAR,KDAY,KHR,KMIN,KSEC,ITIME)
C
C     ---------------------------------------------------------
C     COMPUTE LONGITUDE OF ASCENDING NODE (RAD) AT THE ALERT TIME
C     ---------------------------------------------------------
      FF=WMOTN/PERIOD/60.D0*DFLOAT(ITIME)
      RLANT=(XLNODE-FF)*PI/180.D0
C
C     ---------------------------------------------------
C     COMPUTE ARGUMENT OF PERIGEE (RAD) AT THE ALERT TIME
C     ---------------------------------------------------
      RPHIT=2.D0*PI/(PERIOD*60.D0)*DFLOAT(ITIME)
C
C     ---------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE ORBITAL PLAN (METRES)
C     ---------------------------------------------------------
      R=7456.D3
      XPT=R*DCOS(RPHIT)
      YPT=R*DSIN(RPHIT)
C
C     ----------------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE EARTH FIXED SYSTEM (METRES)
```

```
C     ----------------------------------------------------------------
      XS=XPT*DCOS(RLANT)
      YS=XPT*DSIN(RLANT)
      ZS=YPT
C
      RETURN
      END
```

```
      SUBROUTINE NNSSKE(IY,MDAY,TP,XN,W,WDOT,EC,AO,OM,OMDOT,COSI,GAST,
     #                  SINI,KYEAR,KDAY,KHR,KMIN,KSEC,WE,PI,XS,YS,ZS,
     #                  IPR)
C-----------------------------------------------------------------------
C     NAME:    NNSSKE
C     PURPOSE: TO COMPUTE THE SATELLITE COORDINATES IN THE EARTH FIXED
C              SYSTEM (WGS-72) FROM NNSS KEPLERIAN PARAMETERS
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  IY:          INPUT YEAR
C        I  MDAY:        INPUT DAY
C        I  TP:          TIME OF PERIGEE (MINUTES)
C        I  XN:          MEAN MOTION (DEGREES/MINUTE)
C        I  W:           ARGUMENT OF PERIGEE (DEGREES)
C        I  WDOT:        ARGUMENT OF PERIGEE DOT (DEGREE/MINUTE)
C        I  EC:          ECCENTRICITY
C        I  AO:          SEMI-MAJOR AXIS (METRES)
C        I  OM:          RIGHT ASCENSION OF ASCENDING NODE (DEGREES)
C        I  OMDOT:       RIGHT ASCENSION DOT (DEGREE/MINUTE)
C        I  COSI:        COSINE OF INCLINAISON
C        I  GAST:        RIGHT ASCENSION OF GREENWICH (DEGREES)
C        I  SINI:        SINE OF INCLINAISON
C        I  KYEAR:       YEAR OF THE ALERT
C        I  KDAY:        DAY OF THE ALERT
C        I  KHR:         HOUR OF THE ALERT
C        I  KMIN:        MINUTE OF THE ALERT
C        I  KSEC:        SECOND OF THE ALERT
C        I  WE:          EARTH ROTATION RATE (RAD/SEC)
C        I  PI:          VALUE OF PI
C        I  IPR:         PRINT CODE FOR LINE CODE
C
C        O  XS:          SATELLITE X COORDINATE (METRES)
C        O  YS:          SATELLITE Y COORDINATE (METRES)
C        O  ZS:          SATELLITE Z COORDINATE (METRES)
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ------------------------------------------------------------
C     TRANSFORM THE TIME OF PERIGEE IN HOURS , MINUTES AND SECONDES
C     ------------------------------------------------------------
      CALL MINFRA(TP,IH,IM,IS)
C
C     ---------------------------------------------------------
C     COMPUTE THE NUMBER OF SECONDES BETWEEN THE TIME OF PERIGEE
C     AND THE TIME OF THE ALERT
C     ---------------------------------------------------------
      CALL TIME(IY,MDAY,IH,IM,IS,KYEAR,KDAY,KHR,KMIN,KSEC,ITIME)
C
C     -------------------------------------------------------------
C     COMPUTE LONGITUDE OF ASCENDING NODE (RAD) AT THE REFERENCE TIME
C     -------------------------------------------------------------
      RLANO=(OM-GAST)*PI/180.DO
C
C     ---------------------------------------------------------
C     COMPUTE LONGITUDE OF ASCENDING NODE (RAD) AT THE ALERT TIME
C     ---------------------------------------------------------
      RLANT=RLANO-(WE-(OMDOT*PI/10800.DO))*DFLOAT(ITIME)
C
```

```
C     ------------------------------------------
C     COMPUTE MEAN ANOMALY (RAD) AT THE ALERT TIME
C     ------------------------------------------
      RXMAT=XN*PI/10800.D0*DFLOAT(ITIME)
C
C     --------------------------------------------------
C     COMPUTE ECCENTRIC AND TRUE ANOMALY FROM MEAN ANOMALY
C     --------------------------------------------------
      CALL ANMLY(RXMAT,EC,REAT,RTAT,IFLAG,PI,IPR)
C
C     --------------------------------------------------
C     COMPUTE ARGUMENT OF LATITUDE (RAD) AT THE ALERT TIME
C     --------------------------------------------------
      RPHIT=RTAT+W*PI/180.D0-DABS(WDOT)*PI/10800.D0*DFLOAT(ITIME)
C
C     ----------------------------
C     COMPUTE ORBIT RADIUS (METRES)
C     ----------------------------
      RT=A0*(1.D0-EC*DCOS(REAT))*1000.D0
C
C     ----------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE ORBITAL PLAN (METRES)
C     ----------------------------------------------------------
      XPT=RT*DCOS(RPHIT)
      YPT=RT*DSIN(RPHIT)
C
C     -------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE EARTH FIXED SYSTEM
C     -------------------------------------------------------
      XS=XPT*DCOS(RLANT)-YPT*COSI*DSIN(RLANT)
      YS=XPT*DSIN(RLANT)+YPT*COSI*DCOS(RLANT)
      ZS=YPT*SINI
C
      RETURN
      END
```

```
      SUBROUTINE NPBST(IY,DAY,XINC,RA,EC,AP,XMA,XN,PN,WMTN,
     #                 KYEAR,KDAY,KHR,KMIN,KSEC,PI,WE,
     #                 GM,XS,YS,ZS,IPR)
C-----------------------------------------------------------------------
C     NAME:    NPBST
C     PURPOSE: TO COMPUTE THE SATELLITE COORDINATES IN THE EARTH FIXED
C              SYSTEM (WGS-72) FROM NASA PREDICTION BULLETIN (GAST USED)
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  IY:         INPUT YEAR
C        I  DAY:        DAY AND FRACTION OF DAY OF THE INPUT
C        I  XINC:       INCLINAISON (DEGREES)
C        I  RA:         RIGHT ASCENSION OF THE ASCENDING NODE (DEGREES)
C        I  EC:         ECCENTRICITY
C        I  AP:         ARGUMENT OF PERIGEE (DEGREES)
C        I  XMA:        MEAN ANOMALY (DEGREES)
C        I  XN:         MEAN MOTION (REVOLUTIONS/DAY)
C        I  PN:         NODAL PERIOD (MINUTES)
C        I  WMTN:       WESTWARD MOTION (DEGREES)
C        I  KYEAR:      YEAR OF THE ALERT
C        I  KDAY:       DAY OF THE ALERT
C        I  KHR:        HOUR OF THE ALERT
C        I  KMIN:       MINUTE OF THE ALERT
C        I  KSEC:       SECOND OF THE ALERT
C        I  PI:         VALUE OF PI
C        I  WE:         EARTH ROTATION RATE (RAD/SEC)
C        I  GM:         UNIVERSAL GRAVITATIONAL CONSTANT (M*3/SEC*2)
C        I  IPR:        PRINT CODE FOR LINE CODE
C
C        O  XS:         SATELLITE X COORDINATE (METRES)
C        O  YS:         SATELLITE Y COORDINATE (METRES)
C        O  ZS:         SATELLITE Z COORDINATE (METRES)
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---------------------------------------------------------------
C     TRANSFORM DAY AND FRACTIONAL PORTION OF DAY IN DAYS , HOURS,
C     MINUTES AND SECONDS
C     ---------------------------------------------------------------
      CALL DAYFRA(DAY,IDAY,IH,IM,IS)
C
C     -----------------------------------------------------------------
C     COMPUTE THE NUMBER OF SECONDES BETWEEN THE TIME OF REFERENCE AND
C     THE TIME OF THE ALERT
C     -----------------------------------------------------------------
      CALL TIME(IY,IDAY,IH,IM,IS,KYEAR,KDAY,KHR,KMIN,KSEC,ITIME)
C
C     -----------------------------------------
C     COMPUTE GAST (RAD) AT THE REFERENCE TIME
C     -----------------------------------------
      XMJDO=DJUL(IY-1,12,31.DO)
      XMJD=XMJDO+DAY
      RGAST=THETA(XMJD)
C
C
C     --------------------------------------
C     COMPUTE RIGHT ASCENSION DOT (RAD/SEC)
C     --------------------------------------
```

```
      AA=WMTN/PN*PI/10800.D0
      RSRAD=WE-AA
C
C     -----------------------------
C     COMPUTE MEAN MOTION (RAD/SEC)
C     -----------------------------
      RSXN=XN*2.D0*PI/86400.D0
C
C     -----------------------------------------
C     COMPUTE ARGUMENT OF PERIGEE DOT (RAD/SEC)
C     -----------------------------------------
      BB=2.D0*PI/PN/60.D0
      RSAPD=BB-RSXN
C
C     --------------------------------------
C     COMPUTE ORBIT SEMI-MAJOR AXIS (METRES)
C     --------------------------------------
      A0=(GM/RSXN**2.D0)**(1.D0/3.D0)
C
C     --------------------------------------------
C     COMPUTE MEAN ANOMALY (RAD) AT THE ALERT TIME
C     --------------------------------------------
      RXMAT=XMA*PI/180.D0+RSXN*DFLOAT(ITIME)
C
C     ----------------------------------------------------
C     COMPUTE ECCENTRIC AND TRUE ANOMALY FROM MEAN ANOMALY
C     ----------------------------------------------------
      CALL ANMLY(RXMAT,EC,REAT,RTAT,IFLAG,PI,IPR)
C
C     ---------------------------------------------------
C     COMPUTE ARGUMENT OF PERIGEE (RAD) AT THE ALERT TIME
C     ---------------------------------------------------
      RAPT=AP*PI/180.D0+RSAPD*DFLOAT(ITIME)
C
C     ----------------------------------------------------
C     COMPUTE ARGUMENT OF LATITUDE (RAD) AT THE ALERT TIME
C     ----------------------------------------------------
      RPHIT=RTAT+RAPT
C
C     -----------------------------
C     COMPUTE ORBIT RADIUS (METRES)
C     -----------------------------
      RT=A0*(1.D0-EC*DCOS(REAT))
C
C     ----------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE ORBITAL PLAN (METRES)
C     ----------------------------------------------------------
      XPT=RT*DCOS(RPHIT)
      YPT=RT*DSIN(RPHIT)
C
C     ----------------------------------------------------------
C     COMPUTE LONGITUDE OF ASCENDING NODE (RAD) AT THE ALERT TIME
C     ----------------------------------------------------------
      CC=(RSRAD-WE)*DFLOAT(ITIME)-RGAST
      RLANT=RA*PI/180.D0+CC
C
C     ---------------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE EARTH FIXED SYSTEM (METRES)
C     ---------------------------------------------------------------
      XS=XPT*DCOS(RLANT)-YPT*DCOS(XINC*PI/180.D0)*DSIN(RLANT)
      YS=XPT*DSIN(RLANT)+YPT*DCOS(XINC*PI/180.D0)*DCOS(RLANT)
```

```
      ZS=YPT*DSIN(XINC*PI/180.D0)
C
      RETURN
      END
```

```
      SUBROUTINE NPB(IY,DAY,XINC,EC,AP,XMA,XN,PN,WMTN,XLANW,
     #               KYAN,KDAYAN,KHAN,XMAN,KYEAR,KDAY,KHR,KMIN,
     #               KSEC,PI,WE,GM,XS,YS,ZS,IPR)
C-----------------------------------------------------------------------
C     NAME:     NPB
C     PURPOSE:  TO COMPUTE THE SATELLITE COORDINATES IN THE EARTH FIXED
C               SYSTEM (WGS-72) FROM NASA PREDICTION BULLETIN
C
C     VERSION : JUNE 1983 , R. SANTERRE
C
C     PARAMETERS:
C        I  IY:          INPUT YEAR
C        I  DAY:         INPUT DAY (DAY AND FRACTION OF DAY)
C        I  XINC:        INCLINAISON (DEGREES)
C        I  EC:          ECCENTRICITY
C        I  AP:          ARGUMENT OF PERIGEE (DEGREES)
C        I  XMA:         MEAN ANOMALY (DEGREES)
C        I  XN:          MEAN MOTION (REVOLUTIONS/DAY)
C        I  PN:          NODAL PERIOD (MINUTES)
C        I  WMTN:        WESTWARD MOTION (DEGREES)
C        I  XLANW:       LONGITUDE OF ASCENDING NODE (DEGREES WEST)
C        I  KYAN:        YEAR OF ASCENDING NODE
C        I  KDAYAN:      DAY OF ASCENDING NODE
C        I  KHAN:        HOUR OF ASCENDING NODE
C        I  XMAN:        MINUTES AND FRACTION OF MINUTE OF ASCENDING NODE
C        I  KYEAR:       YEAR OF THE ALERT
C        I  KDAY:        DAY OF THE ALERT
C        I  KHR:         HOUR OF THE ALERT
C        I  KMIN:        MINUTE OF THE ALERT
C        I  KSEC:        SECOND OF THE ALERT
C        I  PI:          VALUE OF PI
C        I  WE:          EARTH ROTATION RATE (RAD/SEC)
C        I  GM:          UNIVERSAL GRAVITATIONAL CONSTANT (M*3/SEC*2)
C        I  IPR:         PRINT CODE FOR LINE CODE
C
C        O  XS:          SATELLITE X COORDINATE (METRES)
C        O  YS:          SATELLITE Y COORDINATE (METRES)
C        O  ZS:          SATELLITE Z COORDINATE (METRES)
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     ---------------------------------------------------------
C     TRANSFORM DAY AND FRACTIONAL PORTION OF DAY (TIME INPUTS)
C     IN HOURS , MINUTES AND SECONDES
C     ---------------------------------------------------------
      CALL DAYFRA(DAY,IDAY,IH,IM,IS)
C
C     ---------------------------------------------------------
C     TRANSFORM MINUTES AND FRACTIONAL PORTION OF MINUTE
C     (TIME OF ASCENDIND NODE) IN HOURS , MINUTES AND SECONDES
C     ---------------------------------------------------------
      CALL MINFRA(XMAN,IHAN,IMAN,ISAN)
C
C     ------------------------------------------------------------
C     COMPUTE NUMBER OF SECONDES BETWEEN THE TIME OF ASCENDING NODE
C     AND THE INPUTS TIME
C     ------------------------------------------------------------
      CALL TIME(KYAN,KDAYAN,KHAN,IMAN,ISAN,IY,IDAY,IH,IM,IS,ITIME1)
C
C     ----------------------------------------------------
```

```
C      COMPUTE MEAN ANOMALY (DEGREES) AT THE REFERENCE TIME
C      (TIME OF ASCENDING NODE)
C      ----------------------------------------------------
       XMA=XMA-XN*360.D0/86400.D0*DFLOAT(ITIME1)
C
C      ---------------------------------------------------------
C      COMPUTE THE NUMBER OF SECONDES BETWEEN THE REFERENCE TIME
C      (TIME OF ASCENDING NODE) AND THE TIME OF THE ALERT
C      ---------------------------------------------------------
       CALL TIME(KYAN,KDAYAN,KHAN,IMAN,ISAN,KYEAR,KDAY,KHR,KMIN,KSEC,
      #          ITIME)
C
C      -------------------------------------
C      COMPUTE RIGHT ASCENSION DOT (RAD/SEC)
C      -------------------------------------
       AA=WMTN/PN*PI/10800.D0
       RSRAD=WE-AA
C
C      -----------------------------
C      COMPUTE MEAN MOTION (RAD/SEC)
C      -----------------------------
       RSXN=XN*2.D0*PI/86400.D0
C
C      -----------------------------------------
C      COMPUTE ARGUMENT OF PERIGEE DOT (RAD/SEC)
C      -----------------------------------------
       BB=2.D0*PI/PN/60.D0
       RSAPD=BB-RSXN
C
C      --------------------------------------
C      COMPUTE ORBIT SEMI-MAJOR AXIS (METRES)
C      --------------------------------------
       A0=(GM/RSXN**2.D0)**(1.D0/3.D0)
C
C      ------------------------------------------
C      COMPUTE MEAN ANOMALY (RAD) AT THE ALERT TIME
C      ------------------------------------------
       RXMAT=XMA*PI/180.D0+RSXN*DFLOAT(ITIME)
C
C      ---------------------------------------------------
C      COMPUTE ECCENTRIC AND TRUE ANOMALY FROM MEAN ANOMALY
C      ---------------------------------------------------
       CALL ANMLY(RXMAT,EC,REAT,RTAT,IFLAG,PI,IPR)
C
C      ---------------------------------------------------
C      COMPUTE ARGUMENT OF PERIGEE (RAD) AT THE ALERT TIME
C      ---------------------------------------------------
       RAPT=AP*PI/180.D0+RSAPD*DFLOAT(ITIME-ITIME1)
C
C      ---------------------------------------------------
C      COMPUTE ARGUMENT OF LATITUDE (RAD) AT THE ALERT TIME
C      ---------------------------------------------------
       RPHIT=RTAT+RAPT
C
C      ----------------------------
C      COMPUTE ORBIT RADIUS (METRES)
C      ----------------------------
       RT=A0*(1.D0-EC*DCOS(REAT))
C
C      -------------------------------------------------------------
C      COMPUTE SATELLITE COORDINATES IN THE ORBITAL PLAN (METRES)
```

```
C     ----------------------------------------------------------
      XPT=RT*DCOS(RPHIT)
      YPT=RT*DSIN(RPHIT)
C
C     -----------------------------------------------------------
C     COMPUTE LONGITUDE OF ASCENDING NODE (RAD) AT THE ALERT TIME
C     -----------------------------------------------------------
      CC=(RSRAD-WE)*DFLOAT(ITIME)
      RLANT=(360.D0-XLANW)*PI/180.D0+CC
C
C     ----------------------------------------------------------------
C     COMPUTE SATELLITE COORDINATES IN THE EARTH FIXED SYSTEM (METRES)
C     ----------------------------------------------------------------
      XS=XPT*DCOS(RLANT)-YPT*DCOS(XINC*PI/180.D0)*DSIN(RLANT)
      YS=XPT*DSIN(RLANT)+YPT*DCOS(XINC*PI/180.D0)*DCOS(RLANT)
      ZS=YPT*DSIN(XINC*PI/180.D0)
C
      RETURN
      END
```

```
      SUBROUTINE EPNCD(NSAT,IY,MDAY,ISAT,PERIOD,WMOTN,TNODE,
     #                 XLNODE,ICR)
C
C     TO READ INPUTS NNSS NODAL CROSSING DATA
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION IY(NSAT)      ,   MDAY(NSAT)     ,    ISAT(NSAT)    ,
     #          PERIOD(NSAT)  ,   WMOTN(NSAT)    ,    TNODE(NSAT)   ,
     #          XLNODE(NSAT)
C
      DO 10 I=1,NSAT
      READ(ICR,20) IY(I)      ,  MDAY(I)    ,  ISAT(I) ,
     #             PERIOD(I),    WMOTN(I)   ,  TNODE(I),
     #             XLNODE(I)
C
   20 FORMAT(3I5,4F10.3)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EPKE(NSAT,IY,MDAY,ISAT,TP,XN,W,WDOT,EC,AO,OM,OMDOT,
     #                COSI,GAST,SINI,ICR)
C
C     TO READ INPUTS NNSS KEPLERIAN PARAMETERS
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION IY(NSAT)    ,     MDAY(NSAT)    ,    ISAT(NSAT)    ,
     #          TP(NSAT)    ,     XN(NSAT)      ,    W(NSAT)       ,
     #          WDOT(NSAT)  ,     EC(NSAT)      ,    AO(NSAT)      ,
     #          OM(NSAT)    ,     OMDOT(NSAT)   ,    COSI(NSAT)    ,
     #          GAST(NSAT)  ,     SINI(NSAT)
C
      DO 10 I=1,NSAT
C
      READ(ICR,20) IY(I)    ,     MDAY(I)       ,    ISAT(I)       ,
     #             TP(I)    ,     XN(I)         ,    W(I)          ,
     #             WDOT(I)  ,     EC(I)         ,    AO(I)         ,
     #             OM(I)    ,     OMDOT(I)      ,    COSI(I)       ,
     #             GAST(I)  ,     SINI(I)
C
   20 FORMAT(3I5,2(F10.4,F10.7),F10.6,/,F10.2,F10.4,F10.7,F10.6,
     #       F10.4,F10.6)
C
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EPNPBG(NSAT,IDSAT,IY,DAY,XINC,RA,EC,AP,XMA,XN,PN,
     #                  WMTN,ICR)
C
C     TO READ INPUTS NASA PREDICTION BULLETIN (GAST USED)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION IDSAT(NSAT)    ,    IY(NSAT)     ,  DAY(NSAT)    ,
     #          XINC(NSAT)     ,    RA(NSAT)     ,  EC(NSAT)     ,
     #          AP(NSAT)       ,    XMA(NSAT)    ,  XN(NSAT)     ,
     #          PN(NSAT)       ,    WMTN(NSAT)
C
      DO 10 I=1,NSAT
C
      READ(ICR,20) IDSAT(I)    ,    IY(I)          ,  DAY(I)     ,
     #             XINC(I)     ,    RA(I)          ,  EC(I)      ,
     #             AP(I)       ,    XMA(I)         ,  XN(I)      ,
     #             PN(I)       ,    WMTN(I)
C
   20 FORMAT(2I5,F13.8,2F10.4,F10.7,F10.4,/,F10.4,F12.8,2F9.3)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EPNPB(NSAT,IDSAT,IY,DAY,XINC,EC,AP,XMA,XN,PN,WMTN,
     #                 XLANW,KYAN,KDAYAN,KHAN,XMAN,ICR)
C
C     TO READ INPUTS NASA PREDICTION BULLETIN
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION IDSAT(NSAT)    ,   IY(NSAT)      ,   DAY(NSAT)    ,
     #          XINC(NSAT)     ,   EC(NSAT)      ,   AP(NSAT)     ,
     #          XMA(NSAT)      ,   XN(NSAT)      ,   PN(NSAT)     ,
     #          WMTN(NSAT)     ,   XLANW(NSAT)   ,   KYAN(NSAT)   ,
     #          KDAYAN(NSAT)   ,   KHAN(NSAT)    ,   XMAN(NSAT)
C
      DO 10 I=1,NSAT
C
      READ(ICR,20) IDSAT(I)    ,   IY(I)    ,   DAY(I)     ,
     #             XINC(I)     ,   EC(I)    ,   AP(I)      ,
     #             XMA(I)      ,   XN(I)    ,   PN(I)      ,
     #             WMTN(I)     ,   XLANW(I) ,   KYAN(I)    ,
     #             KDAYAN(I)   ,   KHAN(I)  ,   XMAN(I)
C
   20 FORMAT(2I5,F13.8,F10.4,F10.7,2F10.4,/,F12.8,2F9.3,F8.2,2I5,I3,
     #       F6.2)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DATUM ( DTM , IDTM , LUI )
      IMPLICIT   REAL * 8 ( A-H , O-Z )
      DIMENSION  DTM(7)
C$
C NAME        DATUM
C
C PURPOSE     INITIALIZE DATUM PARAMETERS
C
C CALLING     CALL DATUM ( DTM , IDTM , LUI )
C
C PARAMETERS  O   DTM   DATUM PARAMETERS
C                         DTM(1) ELLIPSOID SEMIMAJOR AXIS (M)
C                         DTM(2) ELLIPSOID ECCENTRICITY
C                         DTM(3) X TRANSLATION COMPONENT (M)
C                         DTM(4) Y TRANSLATION COMPONENT (M)
C                         DTM(5) Z TRANSLATION COMPONENT (M)
C                         DTM(6) GRAVITATIONAL CONSTANT (M**3/SEC**2)
C                         DTM(7) EARTH ROTATION RATE (RAD/SEC)
C             I   IDTM  DEFAULT SWITCH
C                         0 = TAKE DEFAULT VALUES
C                         NONZERO = READ USER SPECIFIED VALUES FROM LUI
C             I   LUI   INPUT LOGICAL UNIT NUMBER
C$
      IF(IDTM.NE.0) GO TO 3
C---------------------------------------------------------------
C     WGS-72 DATUM PARAMETERS
C---------------------------------------------------------------
      DTM(1) = 6378135.0D0
      DTM(2) = 8.181881123D-2
      DTM(3) = 0.D0
      DTM(4) = 0.D0
      DTM(5) = 0.D0
      GO TO 4
C--------------------------------------------------------------
C     USER DEFINED DATUM
C--------------------------------------------------------------
    3 READ(LUI,100) (DTM(I),I=1,7)
  100 FORMAT(7G12.4)
      BDVDA = DTM(2) / DTM(1)
      DTM(2) = DSQRT ( (1.D0 + BDVDA) * (1.D0 - BDVDA) )
C
    4 DTM(6) = 398600.8D9
      DTM(7) = 7.292115147D-5
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION ZCOUNT(IYR,IDAY,IHR,MIN,ISEC)
      IMPLICIT REAL * 8 ( A - H , O - Z )
C$
C NAME             ZCOUNT
C
C PURPOSE          GIVEN A TIME EPOCH, COMPUTE THE Z-COUNT
C
C CALLING          X = ZCOUNT(IYR,IDAY,IHR,MIN,ISEC)
C
C PARAMETERS       I   IYR        YEAR
C                  I   IDAY       DAY IN YEAR
C                  I   IHR        HOUR
C                  I   MIN        MINUTE
C                  I   ISEC       SECOND
C                  O   ZCOUNT      NUMBER OF 6-SECOND INTERVALS
C                                 SINCE SATURDAY/SUNDAY MIDNIGHT
C$
C REFERENCE YEAR  JYEAR=1982
C DAY=0 FOR 1982 IS THURSDAY=5 (I.E. DAY 5 IN THE WEEK)
      JYEAR = 1982
      JDAYO = 5
C THE TERM (IYR-JYEAR+2)/4 IS TO ACCOUNT FOR LEAP YEARS
      IWEEK = JDAYO + (IYR - JYEAR) + (IYR - JYEAR + 2) / 4
      IWEEK = MOD( (IWEEK+IDAY) , 7)
      IF(IWEEK .LE. O) IWEEK = IWEEK + 7
      ZCOUNT = (DBLE(IHR*3600 + MIN*60 + ISEC)
     #        + (IWEEK-1)*86400.DO ) / 6.DO
      RETURN
      END
```

```
      SUBROUTINE READEF(REC,DTM,EPH,EPHBUF,CLK,CLKBUF,NSAT,LU,IPR)
      IMPLICIT   REAL * 8 ( A-H , O-Z )
      DIMENSION  REC(1) , DTM(1) , EPH(1) , EPHBUF(1) , CLK(1) ,
     $           CLKBUF(1)
      DATA       PI        / 3.141592653589793D0 /
C$
C NAME          READEF
C
C PURPOSE       READE IS TO READ GPS FILES IN THE EXCHANGE SHELTECH/IBM
C               FORMAT
C
C CALLING       CALL READEF(REC,DTM,EPH,EPHBUF,CLK,CLKBUF,NSAT,LU,IPR)
C
C PARAMETERS    I   REC    = RECORD DATA BUFFER
C               I   DTM    = DATUM PARAMETERS
C                            DTM(1) = ELLIPSOID SEMIMAJOR AXIS (M)
C                            DTM(2) = ELLIPSOID ECCENTRICITY
C                            DTM(3) = X TRANSLATION COMPONENT (M)
C                            DTM(4) = Y TRANSLATION COMPONENT (M)
C                            DTM(5) = Z TRANSLATION COMPONENT (M)
C                            DTM(6) = GRAVITATIONAL CONSTANT (M**3/SEC**2)
C                            DTM(7) = EARTH ROTATION RATE (RAD/SEC)
C               I/O CLK    = SATELLITE CLOCK PARAMETERS
C               I   LU     = INPUT DATA LOGICAL UNIT
C               I   IPR    = PRINT CODE FOR LINE PRINTER
C               O   EPH    = SATELLITE EPHEMERIS PARAMETERS
C                            EPH(1) = AGE OF EPHEMERIS (SEC)
C                            EPH(2) = RADIUS SIN HARMONIC AMPL (M)
C                            EPH(3) = CORRECT COMPUTED MEAN MOTION (RAD/SEC)
C                            EPH(4) = MEAN ANOMALY AT REF TIME (RAD)
C                            EPH(5) = LATITUDE COS HARMONIC AMPL (RAD)
C                            EPH(6) = ECCENTRICITY
C                            EPH(7) = LATITUDE SIN HARMONIC AMPL (RAD)
C                            EPH(8) = ORBIT SEMIMAJOR AXIS (M)
C                            EPH(9) = EPHEMERIS REF TIME (SEC)
C                            EPH(10)= INCLINATION COS HARMONIC AMPL (RAD)
C                            EPH(11)= RIGHT ASCENSION AT REF TIME (RAD)
C                            EPH(12)= INCLINATION SIN HARMONIC AMPL (RAD)
C                            EPH(13)= INCLINATION AT REF TIME (RAD)
C                            EPH(14)= RADIUS COS HARMONIC AMPL (M)
C                            EPH(15)= ARGUMENT OF PERIGEE
C                            EPH(16)= RATE OF RIGHT ASCENSION (RAD/SEC)
C                            EPH(17)= SATELLITE I.D.
C                            EPH(18)= CORRECTED MEAN MOTION (RAD/SEC)
C               O   REC    = RECORD DATA BUFFER
C               O   CLKBUF = BUFFER OF CLOCK PARAMS OF ALL AVAIL SATS
C               O   EPHBUF = BUFFER OF EPHEMERIS PARAMS OF ALL AVAIL SATS
C               I   NSAT   = NUMBER OF SATELLITES
C               I   LU     = INPUT DATA LOGICAL UNIT
C               I   IPR    = PRINT CODE FOR LINE PRINTER
C
C EXTERNALS     NCLOK , DASET , NEPHM
C
C COMMENTS      THE ROUTINE CURRENTLY READS SATELLITE EPHEMERIS DATA
C               ONLY
C$
      IPC = 0
      N = 3 * NSAT
C
      DO 900 I = 1,N
        READ ( LU , 1000 , END = 901 ) ( REC(K) , K = 1 , 16 )
```

```
 1000   FORMAT ( 4D20.12 )
C-------------------------------------------------------------------
C                           IDENTIFY SATELLITE
C-------------------------------------------------------------------
C
        TEMP  = REC(1) / 1D4
        TEMP1 = DMOD ( TEMP , 1D2 )
        ISV = IDINT( TEMP1 )
C-------------------------------------------------------------------
C                           IDENTIFY RECORD TYPE
C                                0 - MEASUREMENT RECORD
C                                1 - HEADER INFORMATION
C                                2 - WEATHER DATA
C                                3 - EPHEMERIS DATA SUBFRAME 1
C                                4 - EPHEMERIS DATA SUBFRAME 2
C                                5 - EPHEMERIS DATA SUBFRAME 3
C                                6 - MESSAGE SUBFRAME 4
C                                7 - ALMANAC SUBFRAME 5
C-------------------------------------------------------------------
        TEMP = REC(1) / 1D6
        ITYPE = IDINT ( TEMP )
        IF ( ITYPE .EQ. 0 ) GO TO 10
        GO TO  (1,2,3,4,5,6,7) , ITYPE
    1   CONTINUE
    2   CONTINUE
        GO TO 11
    3   CALL NCLOK(IPR,IPC,ISV,REC,CLK)
        IPSN = ISV*6 - 5
        IT = 6
        CALL DASET(CLK(1),CLKBUF(IPSN),IT)
        GO TO 11
C-------------------------------------------------------------------
C                              FOR THIS TEST  READ EPHEMERIS ONLY
C-------------------------------------------------------------------
    4   CONTINUE
    5   CONTINUE
        ISUBFR = ITYPE - 2
        CALL NEPHM(ISV,REC,ISUBFR,DTM,PI,EPH)
        IPSN = ISV*18 - 17
        IT = 18
        IF ( ITYPE .EQ. 5 ) CALL DASET ( EPH(1) , EPHBUF(IPSN) , IT )
        GO TO 11
    6   CONTINUE
    7   CONTINUE
   10   CONTINUE
   11   CONTINUE
  900   CONTINUE
  901 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DASET(DIN,DOUT,LEN)
      IMPLICIT    REAL * 8 ( A-H , O-Z )
      INTEGER * 2 ITEMP
      DIMENSION   DIN(1) , DOUT(1)
C$
C NAME          DASET
C
C PURPOSE       COPY A VECTOR
C
C CALLING       CALL DASET(DIN,DOUT,LEN)
C
C PARAMETERS    I   DIN(LEN)
C               O   DOUT(LEN)
C               I   LEN
C
C$
      ITEMP = LEN
      DO 1 I = 1,ITEMP
    1  DOUT(I) = DIN(I)
      RETURN
      END
```

```
      SUBROUTINE PLHXYZ(PHI,RLAM,H,XO,YO,ZO,A,B,X,Y,Z)
      IMPLICIT REAL * 8 ( A-Z )
C$
C NAME          PLHXYZ
C
C PURPOSE       COMPUTE CARTESIAN COORDINATES X , Y , Z
C               GIVEN ELLIPSOIDAL COORDINATES PHI , RLAM , H.
C
C CALLING       CALL PLHXYZ(PHI,RLAM,H,XO,YO,ZO,A,B,X,Y,Z)
C
C PARAMETERS    I   PHI       LATITUDE (RAD)
C               I   RLAM      EAST LONGITUDE (RAD)
C               I   H         HEIGHT (M)
C               I   XO,YO,ZO  DATUM TRANSLATION COMPONENTS (M)
C               I   A,B       ELLIPSOID SEMIMAJOR AND SEMIMINOR AXES (M)
C               O   X,Y,Z     CARTESIAN COORDINATES (M)
C$
      E2 = (A*A - B*B) / (A*A)
      SP = DSIN(PHI)
      CP = DCOS(PHI)
      N  = A / DSQRT(1DO - E2 * SP**2)
      X  = XO + ( N + H ) * CP * DCOS(RLAM)
      Y  = YO + ( N + H ) * CP * DSIN(RLAM)
      Z  = ZO + ( N * (1DO - E2) + H ) * SP
      RETURN
      END
```

```
      SUBROUTINE STXYZ ( AMYE , C , DTM , EPH , ICD , IFLG4 , LUI ,
     $                   PHIK , PI , RINC , RLNG , TIME , XR , YR ,
     $                   ZR, XS, YS, ZS, Y, IPC, LUO )
      IMPLICIT REAL * 8 ( A-H , O-Z )
      DIMENSION   EPH(18) , DTM(7) , Y(ICD) , TS(3) , TR(3)
C$
C NAME          STXYZ
C
C PURPOSE       COMPUTE SATELLITE COORDINATES
C
C CALLING       CALL STXYZ ( AMYE , C , DTM , EPH , ICD ,
C                            IFLG4 , LUI , PHIK , PI , RINC ,
C                            RLNG , TIME , XR , YR , ZR , XS ,
C                            YS, ZS, Y, IPC, LUO )
C
C PARAMETERS    O  AMYE    ECCENTRIC ANOMALY (RAD)
C               I  C       SPEED OF LIGHT (M/SEC)
C               I  DTM     DATUM PARAMETERS
C                          DTM(1) ELLIPSOID SEMIMAJOR AXIS (M)
C                          DTM(2) ELLIPSOID ECCENTRICITY
C                          DTM(3) X TRANSLATION COMPONENT (M)
C                          DTM(4) Y TRANSLATION COMPONENT (M)
C                          DTM(5) Z TRANSLATION COMPONENT (M)
C                          DTM(6) GRAVITATIONAL CONSTANT (M**3/SEC**2)
C                          DTM(7) EARTH ROTATION RATE (RAD/SEC)
C               I  EPH     SATELLITE EPHEMERIS PARAMETERS
C                          EPH(1) AGE OF EPHEMERIS (SEC)
C                          EPH(2) RADIUS SIN HARMONIC AMPL (M)
C                          EPH(3) CORRECT COMPUTED MEAN MOTION (RAD/SEC)
C                          EPH(4) MEAN ANOMALY AT REF TIME (RAD)
C                          EPH(5) LATITUDE COS HARMONIC AMPL (RAD)
C                          EPH(6) ORBITAL ECCENTRICITY
C                          EPH(7) LATITUDE SIN HARMONIC AMPL (RAD)
C                          EPH(8) ELLIPSE SEMIMAJOR AXIS (M)
C                          EPH(9) EPHEMERIS REF TIME (SEC)
C                          EPH(10)INCLINATION COS HARMONIC AMPL (RAD)
C                          EPH(11) RIGHT ASCENSION AT REF  TIME (RAD)
C                          EPH(12) INCLINATION SIN HARMONIC AMPL (RAD)
C                          EPH(13) INCLINATION AT REF TIME (RAD)
C                          EPH(14) RADIUS COS HARMONIC AMPL (M)
C                          EPH(15) ARGUMENT OF PERIGEE (RAD)
C                          EPH(16) RATE OF RIGHT ASCENSION  (RAD/SEC)
C                          EPH(17) SATELLITE I.D.
C                          EPH(18) CORRECTED MEAN MOTION (RAD/SEC)
C                I    ICD  COLUMN DIMENSION OF D MATRIX
C                O    IFLG4 FLAG INDICATES PROCESSING
C                                CONVERGENCE PROBLEMS
C                                 0 ==> CONTINUE PROCESSING
C                                 1 ==> ABORT PROCESSING
C                I    LUI   INPUT  LOGICAL UNIT
C                O    PHIK  ARGUMENT OF LATITUDE (RAD)
C                I    PI    OBVIOUS
C                O    RINC  INCLINATION (RADIANS)
C                O    RLNG  LONGITUDE OF ASCENDING NODE (RAD)
C                I    TIME  GPS TIME MEASURED FROM REF EPOCH (SEC)
C                I    XR    RECEIVER WGS-72 X COORDINATE (M)
C                I    YR    RECEIVER WGS-72 Y COORDINATE (M)
C                I    ZR    RECEIVER WGS-72 Z COORDINATE (M)
C                O    XS    SATELLITE WGS-72 X COORDINATE (M)
C                O    YS    SATELLITE WGS-72 Y COORDINATE (M)
C                O    ZS    SATELLITE WGS-72 Z COORDINATE (M)
```

```
C                I    Y      VECTOR OF BIAS ESTIMATES
C                            Y(2)   ORBIT ELLIPSE SEMIMAJOR AXIS (M)OF
C                            Y(3)   ORBIT ECCENTRICITY
C                            Y(4)   ASC NODE RIGHT ASC AT REF TIME (RAD)
C                            Y(5)   ORBIT INCLINATION AT REF TIME (RAD)
C                            Y(6)   ARGUEMENT OF PERIGEE OF ELLIPSE (RAD)
C                            Y(1)   FOR DOPPLER = FREQUENCY OFFSET (HZ)
C                                   FOR RANGE   = TIME OFFSET UNKNOWN (SEC)
C                I    IPC    PRINT CODE:
C                                       1 = NORMAL
C                                       2 = EXTENDED
C                                       3 = SUPER EXTENDED
C                                       4 = EVEN LONGER!!
C                I    LUO    LISTING LU
C
C EXTERNALS      ANMLY , RANGE
C$
      IFLG4 = 0
C-----------------------------------------------------------------------
C     MEAN ANOMALY
C-----------------------------------------------------------------------
      ANYM = EPH(4) + EPH(18) * TIME
      IF ( IPC .LT. 5 ) GO TO 100
      WRITE(LUO,1000)
      WRITE(LUO,1001) ( I, EPH(I), I=1,18 )
      WRITE(LUO,1002) ( I, Y(I), I=1,7 )
      WRITE(LUO,1102)(I,DTM(I), I = 1,7)
      WRITE(LUO,1003)  TIME
C     WRITE(LUO,1004)
C     WRITE(LUO,1005) ANYM, Y(1), AMYE, AMYT
  100 CONTINUE
C-----------------------------------------------------------------------
C     COMPUTE ECCENTRIC AND TRUE ANOMALIES
C-----------------------------------------------------------------------
      CALL ANMLY ( ANYM , Y(2) , AMYE , AMYT , IFLG2 , PI , LUI )
C-----------------------------------------------------------------------
C     COMPUTE ARGUMENT OF LATITUDE
C-----------------------------------------------------------------------
      PHIK = AMYT + Y(5)
      IF ( IPC .LT. 5 ) GO TO 200
      WRITE(LUO,1010)
      WRITE(LUO,1005) ANYM, Y(2), AMYE, AMYT
      WRITE(LUO,1011) PHIK
  200 CONTINUE
C-----------------------------------------------------------------------
C     COMPUTE 2ND HARMONIC PERTURBATION CORRECTION
C-----------------------------------------------------------------------
      CPK = DCOS ( 2.DO * PHIK )
      SPK = DSIN ( 2.DO * PHIK )
      DUK = EPH(7) * SPK + EPH(5) * CPK
      DRK = EPH(14) * CPK + EPH(2) * SPK
      DIK = EPH(10) * CPK + EPH(12) * SPK
C-----------------------------------------------------------------------
C     CORRECT ARGUMENT OF LATITUDE
C-----------------------------------------------------------------------
      PHIK = PHIK + DUK
C-----------------------------------------------------------------------
C     CORRECT RADIUS
C-----------------------------------------------------------------------
      RK = Y(1) * ( 1.DO - Y(2) * DCOS(AMYE) ) + DRK
C-----------------------------------------------------------------------
```

```
C     CORRECT INCLINATION
C-----------------------------------------------------------------------
      RINC = Y(4) + DIK
C-----------------------------------------------------------------------
C     POSITIONS IN ORBITAL PLANE
C-----------------------------------------------------------------------
      XK = RK * DCOS(PHIK)
      YK = RK * DSIN(PHIK)
C-----------------------------------------------------------------------
C     BEGIN ITERATIONS TO PROPERLY CORRECT COORDS FOR PROPAGATION TIME
C-----------------------------------------------------------------------
      CINC = DCOS(RINC)
      SINC = DSIN(RINC)
      PROP= 0.D0
      IF ( IPC .LT. 5 ) GO TO 300
      WRITE(LUO,1020) CPK, SPK, DUK, DRK, DIK
      WRITE(LUO,1011) PHIK
      WRITE(LUO,1021) RK
      WRITE(LUO,1022) RINC, XK, YK
      WRITE(LUO,1023) CINC, SINC
  300 CONTINUE
      DO 1 I=1,10
C-----------------------------------------------------------------------
C     SOLVE FOR CORRECTED LONGITUDE
C-----------------------------------------------------------------------
      RLNG = Y(3) + ( EPH(16) - DTM(7) ) * TIME + DTM(7) * ((-1.D0) *
     & PROP - EPH(9) )
C-----------------------------------------------------------------------
C     COMPUTE EARTH FRAME COORDINATES OF SATELLITE
C-----------------------------------------------------------------------
      CLNG = DCOS(RLNG)
      SLNG = DSIN(RLNG)
      XS = XK * CLNG - YK * CINC * SLNG
      YS = XK * SLNG + YK * CINC * CLNG
      ZS = YK * SINC
C-----------------------------------------------------------------------
C     COMPUTE RANGE TO SATELLITE IN SECONDS
C-----------------------------------------------------------------------
      TS(1) = XS
      TS(2) = YS
      TS(3) = ZS
      TR(1) = XR
      TR(2) = YR
      TR(3) = ZR
          C = 299792458.D0
      RNG = RANGE ( TS , TR ) / C
      IF ( IPC .LT. 5 ) GO TO 400
      WRITE(LUO,1030) RLNG, CLNG, SLNG
      WRITE(LUO,1031) XS, YS, ZS
      WRITE(LUO,1032) XR, YR, ZR
      WRITE(LUO,1033) RNG , C
  400 CONTINUE
      IF ( DABS( PROP - RNG ) .LT. 1.D-12) GO TO 2
      PROP = RNG
    1 CONTINUE
      IF ( I .LT. 10 ) GO TO 2
      IFLG4 = 1
    2 CONTINUE
      IF ( IPC .EQ. 5 ) WRITE(LUO,1040)
C-----------------------------------------------------------------------
C                           FORMAT STATEMENTS
```

```
C-----------------------------------------------------------------------
 1000 FORMAT( '0', 'ENTERING  STCRD  (FORM)' )
 1001 FORMAT( ' EPH', I3, ' = ', F30.15 )
 1102 FORMAT( ' DTM', I3, ' = ', F30.15 )
 1002 FORMAT( ' Y  ', I3, ' = ', F30.15 )
 1003 FORMAT( ' TIME    = ', F30.15 )
 1004 FORMAT( ' BEFORE  ANMLY  ' )
 1005 FORMAT( ' ANYM, Y(1), AMYE, AMYT ', / 2F30.15 / 2F30.15 )
 1010 FORMAT( ' AFTER  ANMLY  ' )
 1011 FORMAT( ' PHIK    = ', F30.15 )
 1020 FORMAT( ' CPK, SPK, DUK, DRK, DIK ' / 5D15.11 )
 1021 FORMAT( ' RK      = ', D20.10 )
 1022 FORMAT( ' RINC, XK, YK ', 3D20.10 )
 1023 FORMAT( ' CINC, SINC ', 2F30.15 )
 1030 FORMAT( ' RLNG, CLNG, SLNG ', 3D20.10 )
 1031 FORMAT( ' XS, YS, ZS ', 3F20.5 )
 1032 FORMAT( ' XR, YR, ZR ', 3F20.3 )
 1033 FORMAT( ' RNG     = ', F20.15 ,  ' C = ' , F20.5 )
 1040 FORMAT( ' LEAVING  STCRD  (FORM) ', // )
      RETURN
      END
```

```
      SUBROUTINE DERIV(LU,AE,BE,XLAT,XLON,HGT,XS,
     $                S,DSDP,DSDL,ELEV,AZ,IER)
      IMPLICIT      REAL * 8 ( A-H , O-Z )
      DIMENSION     ANG(3) , DRDL(3) , DRDP(3) , DX(3) , NAXIS(3) ,
     $              ROT(3,3) , XR(3) , XS(1) , XT(3)
      DATA          NAXIS    /3,2,-2/,
     $              PI       /3.14159265358979D0/
C$
C NAME         DERIV
C
C PURPOSE      COMPUTE SLANT RANGE(S) TO SATELLITE, ITS
C              DERIVATIVES WITH RESPECT TO LAT (DSDP), LONG (DSDL),
C              AND ELEVATION ANGLE (ELEV)
C
C CALLING      CALL DERIV (LU,AE,BE,XLAT,XLON,HGT,XS,
C                          S,DSDP,DSDL,ELEV,AZ,IER)
C
C PARAMETERS   I   LU       LISTING LU
C              I   AE       EARTH SEMIMAJOR AXIS (M)
C              I   BE       EARTH SEMIMINOR AXIS (M) OR RECIP FLATNG
C              I   XLAT     RECEIVER LATITUDE (DEG)
C              I   XLON     RECEIVER LONGITUDE (DEG)
C              I   HGT      RECEIVER ELLIPSOID HEIGHT (M)
C              I   XS       SATELLITE LOCAL TERRESTRIAL COORDS (M)
C              O   S        RECEIVER TO SATELLITE SLANT RANGE (M)
C              O   DSDP     DERIV OF S WITH RESPECT TO LAT  (M/DEG)
C              O   DSDL     DERIV OF S WITH RESPECT TO LONG (M/DEG)
C              O   ELEV     SATELLITE ELEVATION ABOVE HORIZON (DEG)
C              O   AZ       SATELLITE AZIMUTH FROM NORTH (DEG)
C              O   IER      0 SUCCESSFUL RETURN
C
C EXTERNALS    ROTREF
C$
      IER = 0
C-----------------------------------------------------------------
C                        COMPUTE RECEIVER LOCAL TERRESTRIAL COORDS XR
C-----------------------------------------------------------------
      CP = DCOS(XLAT*PI/180.D0)
      SP = DSIN(XLAT*PI/180.D0)
      CL = DCOS(XLON*PI/180.D0)
      SL = DSIN(XLON*PI/180.D0)
      BOA = (BE / AE)**2
      IF(BE .LT. 6.D3) BOA = (1. - 1./BE)**2
      RN = AE / DSQRT(CP**2 + BOA * SP**2)
      RM = RN * BOA * (RN/AE)**2
      XR(1) = (RN+HGT) * CP * CL
      XR(2) = (RN+HGT) * CP * SL
      XR(3) = (RN*BOA + HGT) * SP
C-----------------------------------------------------------------
C                        COMPUTE DERIVS OF XR WRT TO LAT AND LONG
C-----------------------------------------------------------------
      DRDP(1) = -(RM+HGT) * SP * CL * PI / 180.D0
      DRDP(2) = -(RM+HGT) * SP * SL * PI / 180.D0
      DRDP(3) =  (RM+HGT) * CP      * PI / 180.D0
      DRDL(1) = -XR(2) * PI / 180.D0
      DRDL(2) =  XR(1) * PI / 180.D0
      DRDL(3) = 0.D0
C-----------------------------------------------------------------
C                           COMPUTE SLANT RANGE AND ITS DERIVATIVES
C-----------------------------------------------------------------
      DO 10 I = 1,3
```

```
 10   DX(I) = XS(I) - XR(I)
      S = DSQRT(DX(1)**2 + DX(2)**2 + DX(3)**2)
      DSDP = -(DX(1)*DRDP(1) + DX(2)*DRDP(2) + DX(3)*DRDP(3)) / S
      DSDL = -(DX(1)*DRDL(1) + DX(2)*DRDL(2) + DX(3)*DRDL(3)) / S
C-----------------------------------------------------------------
C                         COMPUTE ELEVATION AND AZIMUTH ANGLE
C-----------------------------------------------------------------
      ANG(1) =(XLON - 180.)*PI/180.DO
      ANG(2) =(XLAT - 90.)*PI/180.DO
      ANG(3) = O.DO
      IT = 3
      CALL ROTREF(IT,NAXIS,ANG,ROT)
      DO 20 I = 1,3
        XT(I) = O.DO
        DO 20 J = 1,3
 20       XT(I) = XT(I) + ROT(I,J) * DX(J)
      ELEV = DATAN2(XT(3),DSQRT(XT(1)**2 + XT(2)**2)) * 180.DO/PI
      AZ = DATAN2(XT(2),XT(1)) * 180.DO/PI
      RETURN
      END
```

```
      SUBROUTINE MATMPY(M1,M2,M3,L,M,N,JL,JM,JN,ICODE)
      REAL*8    M1(JL,1) , M2(JM,1) , M3(JN,1)
C$
C NAME              MATMPY
C
C PURPOSE           TO COMPUTE THE PRODUCT OF TWO MATRICES IN ANY
C                   ALLOWABLE TRANSPOSE COMBINATION AS FOLLOWS:
C
C CALLING           CALL MATMPY(M1,M2,M3,L,M,N,JL,JM,JN,ICODE)
C
C                         OPTION ICODE               PRODUCT M3
C
C                               1                      M1*M2
C                               2                    (M1)T*M2
C                               3                    M1*(M2)T
C                               4                    (M1)T*(M2)T
C
C
C                     (L,M),(M,N),(L,N) ARE THE DIMENSIONS OF THE
C                                       PRE- , POST- AND PRODUCT-MATRICES
C                                       RESPECTIVELY
C                              JL,JM,JN ARE CORRESPONDING DECLARED ROW
C                                       DIMENSIONS AT THE CALLINE PROGRAM
C$
      DO 11 I = 1,L
        DO 11 J = 1,N
          M3(I,J) = 0.DO
          DO 11 K = 1,M
            GO TO(1,2,3,4),ICODE
    1       CONTINUE
C    M3 = M1 * M2
            M3(I,J) = M3(I,J) + M1(I,K)*M2(K,J)
            GO TO 11
    2       CONTINUE
C    M3 = M1 TRANSPOSE * M2
            M3(I,J) = M3(I,J) + M1(K,I)*M2(K,J)
            GO TO 11
    3       CONTINUE
C    M3 = M1 * M2 TRANSPOSE
            M3(I,J) = M3(I,J) + M1(I,K)*M2(J,K)
            GO TO 11
    4       CONTINUE
C    M3 = M1 TRANSPOSE * M2 TRANSPOSE
            M3(I,J) = M3(I,J) + M1(K,I)*M2(J,K)
   11 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ANMLY(AMYM,ORBEC,AMYE,AMYT,IFLAG,PI,LU)
      IMPLICIT REAL*8(A-H,O-Z)
C$
C NAME          ANMLY
C
C PURPOSE       CONVERT ECCENTRICITY AND MEAN ANOMALY TO ECCENTRIC AND
C               TRUE ANOMALY
C
C CALLING       CALL ANMLY(AMYM,ORBEC,AMYE,AMYT,IFLAG,PI,LU)
C
C PARAMETERS    I AMYM = MEAN ANOMALY (RAD)
C               I ORBEC= ORBITAL ECCENTRICITY
C               O AMYE = ECCENTRIC ANOMALY (RAD)
C               O AMYT = TRUE ANOMALY (RAD)
C               O IFLAG= 0 OK
C                        1 NONCONVERGENCE ON KEPLERS EQUATION FOR ECC ANOM
C               I PI
C               I LU   = DIAGNOSTIC PRINTOUT LU
C$
C
      IFLAG = 0
      PI2 = PI * 2.0D0
      ORBECS = ORBEC * ORBEC
      AMYM = DMOD( AMYM,PI2 )
C----------------------------------------------------------------
C               COMPUTE FIRST APPROXIMATION TO ECCENTRIC ANOMALY
C----------------------------------------------------------------
      S = DSIN(AMYM)
      A = ORBEC * S
      B = A * ORBEC * DCOS(AMYM)
      C = ORBECS * A * (1.0D0 - 1.5D0 * S * S)
      AMYE = AMYM + A + B + C
C ---------------------------------------------------------------------
C               ITERATE ON ECCENTRIC ANOMALY
C ---------------------------------------------------------------------
      I = 0
   10 I = I + 1
      E1 = AMYE
      AMYE = E1+(AMYM-E1+ORBEC * DSIN(E1))/(1.0D0 - ORBEC * DCOS(E1))
      IF ( I .LE. 8 ) GO TO 30
      IFLAG = 1
      GO TO 40
   30 IF ( DABS(AMYE-E1) .GT. 1.0D-10 ) GO TO 10
C-------------------------------------------------------------------
C                            ITERATIONS COMPLETE
C-------------------------------------------------------------------
   40 CONTINUE
      SE = DSIN(AMYE)
      DIFF = DABS(AMYM-(AMYE-ORBEC*SE))
      IF ( DIFF .LT. 1.0D-10 ) GO TO 60
      IFLAG = 1
C     WRITE(LU,50) DIFF, I, AMYM, AMYE
C  50 FORMAT(" /ANML2: ECCENTRICITY MISCLOSURE="D20.10/
C    # "          ITERATIONS="I4/"          AMYM="D20.10" AMYE="D20.10)
   60 CONTINUE
C------------------------------------------------------------------
C                              COMPUTE TRUE ANOMALY
C------------------------------------------------------------------
      SF = DSQRT ( 1.0D0 - ORBECS ) * SE
      CF = DCOS ( AMYE ) - ORBEC
      AMYT = DATAN2 ( SF , CF )
```

```
RETURN
END
```

```
      SUBROUTINE ROTREF(NUM,NAXIS,ANGLE,ROT)
      IMPLICIT REAL * 8 ( A-H , O-Z )
      DIMENSION  ROT(3,3) , R1(3,3) , R2(3) ,
     $           ANGLE(NUM) , NAXIS(NUM)
C$
C NAME          ROTREF
C
C PURPOSE       COMPUTE PRODUCT MATRIX OF SEQUENCE OF
C               ROTATIONS AND REFLECTIONS
C
C CALLING       CALL ROTREF(NUM,NAXIS,ANGLE,ROT)
C
C PARAMETERS    I   NUM   = NUMBER OF ROTATIONS AND REFLECTIONS IN SEQUENCE
C               I   NAXIS = SEQUENCE OF ROTATION AND REFLECTION AXES
C                           FOR ROTATIONS USE    1, 2, OR 3
C                           FOR REFLECTIONS USE  -1, -2, OR -3
C               I   ANGLE = SEQUENCE OF ROTATION ANGLES IN RADIANS
C                           IGNORE (ASSUME ZERO) FOR REFLECTIONS
C               O   ROT   = 3X3 PRODUCT MATRIX
C$
      EPS = 1D-15
C-----------------------------------------------------------------------
C                   INITIALIZE (SET 'ROT' = IDENTITY MATRIX)
C-----------------------------------------------------------------------
      DO 10 I=1,3
        DO 10 J=1,3
          ROT(I,J)=0.DO
          IF(I.EQ.J)  ROT(I,J)=1.DO
 10       CONTINUE
C-----------------------------------------------------------------------
C                   PROCESS SEQUENCE OF ROTATIONS AND REFLECTIONS
C-----------------------------------------------------------------------
      DO 100 N=1,NUM
C-----------------------------------------------------------------------
C                   DEFINE 3 AXES FOR CURRENT ROTATION OR REFLECTION
C-----------------------------------------------------------------------
      N1=IABS(NAXIS(N))
      N2=N1+1
      IF(N2.GT.3)  N2=N2-3
      N3=N2+1
      IF(N3.GT.3)  N3=N3-3
C-----------------------------------------------------------------------
C                        DEFINE DIAGONAL ELEMENTS
C-----------------------------------------------------------------------
      R1(N1,N1) = 1.DO
      IF(NAXIS(N).LT.O.DO)  R1(N1,N1) = -1.DO
      R1(N2,N2) = DCOS(ANGLE(N))
      IF(NAXIS(N).LT.O.DO)  R1(N2,N2) = 1.DO
      R1(N3,N3) = R1(N2,N2)
C-----------------------------------------------------------------------
C                        DEFINE NONZERO OFF-DIAGONAL ELEMENTS
C-----------------------------------------------------------------------
      R1(N2,N3) = DSIN(ANGLE(N))
      IF(NAXIS(N).LT.O.DO)  R1(N2,N3) = O.DO
      R1(N3,N2) = -R1(N2,N3)
C-----------------------------------------------------------------------
C                        DEFINE ZERO OFF-DIAGONAL ELEMENTS
C-----------------------------------------------------------------------
      R1(N1,N2) = O.DO
      R1(N1,N3) = O.DO
      R1(N2,N1) = O.DO
```

```
      R1(N3,N1) = 0.D0
C-----------------------------------------------------------------------
C                           FORM PRODUCT (SET 'ROT' = 'R1' * 'ROT')
C-----------------------------------------------------------------------
      DO 100 J=1,3
      DO 30 I=1,3
      R2(I) = 0.D0
      DO 30 K=1,3
   30 R2(I) = R2(I) + R1(I,K)*ROT(K,J)
C
      DO 100 I=1,3
      ROT(I,J) = R2(I)
      IF(DABS(ROT(I,J)).LT.EPS)  ROT(I,J) = 0.D0
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NCLOK ( LUO, IPC, ISAT, REC, CLK )
      IMPLICIT REAL * 8 ( A-H , O-Z )
      DIMENSION   CLK(*) , REC(*)
C$
C NAME          NCLOK
C
C PURPOSE       EXTRACT CLOCK VALUES FROM INPUT RECORD
C
C CALLING       CALL NCLOK (LUO,IPC,ISAT,REC,CLK)
C
C PARAMETERS    I   LUO   LISTING LU
C               I   IPC   PRINT CONTROL
C               I   ISAT  SATELLITE ID
C               I   REC   INPUT RECORD
C               O   CLK   OUTPUT ARRAY
C
C EXTERNAL      DASET
C$
C-----------------------------------------------------------------------
C                           EXTRACT THE CLOCK VALUES
C-----------------------------------------------------------------------
      CALL DASET ( REC(12), CLK, 5 )
      CLK(6) = FLOAT(ISAT)
C-----------------------------------------------------------------------
C                           PRINT CLOCK VALUES IF REQUIRED
C-----------------------------------------------------------------------
      IF ( IPC .GE. 3 ) WRITE(LUO,10100) (REC(J),J=3,11)
      RETURN
C-----------------------------------------------------------------------
C                           FORMAT STATEMENTS
C-----------------------------------------------------------------------
10100 FORMAT( / ' CLOCK VALUES: ALPHA 0-3, BETA 0-3, TGD' /
     1        2(4D20.10 /) , D20.10 )
      END
```

```
      SUBROUTINE NEPHM ( ISAT, REC, J, DTM, PI, EPH )
      IMPLICIT  REAL * 8 ( A-H , O-Z )
      DIMENSION EPH(*) , DTM(*) , REC(*)
C$
C NAME          NEPHM
C
C PURPOSE       EXTRACT EPHEMERIS VALUES FROM INPUT RECORD
C
C CALLING       CALL NEPHM(ISAT,REC,J,DTM,PI,EPH)
C
C PARAMETERS    I    ISAT   SATELLITE ID.
C               I    REC    INPUT RECORD
C               I    J      SUBFRAME ID.
C               I    DTM    DATUM PARAMETERS
C               I    PI     OBVIOUS
C               O    EPH    OUTPUT ARRAY
C
C EXTERNALS     DASET, DFLOT
C$
C-----------------------------------------------------------------------
C                         SEE IF FIRST OR SECOND HALF
C-----------------------------------------------------------------------
      IF ( J .EQ. 3 ) GO TO 200
C-----------------------------------------------------------------------
C                         FIRST HALF - GET VALUES
C-----------------------------------------------------------------------
      IT = 9
      CALL DASET ( REC(3), EPH   , IT)
C-----------------------------------------------------------------------
C                         CONVERT TO BASE UNITS
C-----------------------------------------------------------------------
      EPH(3) = EPH(3) * PI
      EPH(4) = EPH(4) * PI
      EPH(8) = EPH(8) * EPH(8)
C-----------------------------------------------------------------------
C                         COMPUTE CORRECTED MEAN POSITION
C-----------------------------------------------------------------------
      EPH(18) = DSQRT(DTM(6)) / EPH(8)**1.5D0 + EPH(3)
      GO TO 300
C-----------------------------------------------------------------------
C                         SECOND HALF - GET DATA AND
C                         CONVERT TO BASE UNITS
C-----------------------------------------------------------------------
  200 CONTINUE
      IT = 7
      CALL DASET ( REC(3), EPH(10), IT)
      EPH(11) = EPH(11) * PI
      EPH(13) = EPH(13) * PI
      EPH(15) = EPH(15) * PI
      EPH(16) = EPH(16) * PI
      EPH(17) = FLOAT(ISAT)
  300 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION RANGE(XSV,XRV)
      IMPLICIT REAL * 8 ( A-H , O-Z )
      DIMENSION XRV(3) , XSV(3)
C$
C NAME          RANGE
C
C PURPOSE       COMPUTE STATION-TO-SATELLITE RANGE, GIVEN
C               STATION AND SATELLITE CARTESIAN COORDINATES
C
C CALLING       X = RANGE(XSV,XRV)
C
C PARAMETERS    I    XRV(3)   STATION CARTESIAN COORDINATES
C               I    XSV(3)   SATELLITE CARTESIAN COORDINATES
C               O    RANGE    STATION-TO-SATELLITE RANGE
C$
      R = 0D0
      DO 10 I = 1 , 3
   10   R = ( XSV(I) - XRV(I) ) ** 2 + R
      RANGE = DSQRT(R)
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION ZFRDAY(ZCNT)
C
C     PURPOSE            CONVERT 'ZCNT' TIME INTO FRACTION OF DAY
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      TSEC = ZCNT *  6.
      TIDD = TSEC / 86400.D0
      ITIDD = IDINT(TIDD)
      ZFRDAY = TIDD - DBLE(ITIDD)
      RETURN
      END
```

```
      SUBROUTINE SPIN(Q,N,MM,DET,IDEXP)
      IMPLICIT REAL * 8 ( A-H , O-Z )
      REAL*8        Q(MM,MM)
C$
C NAME          SPIN
C
C PURPOSE       MATRIX INVERSION ROUTINE FOR SYMMETRIC POSITIVE-DEFINITE
C               MATRICES.  THE MATRIX INVERTED IS THE UPPER N BY N
C               PORTION OF THE MATRIX Q (DIMENSIONED MM BY MM
C               IN CALLING ROUTINE).
C
C CALLING       CALL SPIN (Q,N,MM,DET,IDEXP)
C
C PARAMETERS    I/O  Q      = MATRIX TO BE INVERTED (DIMENSIONED MM BY MM)
C                             ON OUTPUT, UPPER LEFT N BY N SUBMATRIX CONTAINS
C                             INVERSE OF INPUT UPPER LEFT N BY N SUBMATRIX.
C               I    N      = DIMENSION OF THE ACTUAL PART (UPPER LEFT CORNER)
C                             OF Q TO BE INVERTED (N .LE. MM)
C               I    MM     = DIMENSIONED SIZE OF Q IN THE CALLING ROUTINE.
C               O    DET    = NON-EXPONENT PORTION OF THE DETERMINANT OF THE
C                             INPUT N BY N (UPPER LEFT CORNER) MATRIX.
C               O    IDEXP  = EXPONENT (OF 10) PART OF THE DETERMINANT. THUS
C                             DETERMINANT = DET * 10 ** IDEXP     .
C
C$
      DET = 0.0D0
      DO  4 J = 1,N
        DO  4 I = 1,J
          IF (I.EQ.1) GO TO 2
            M = I - 1
            SUM = 0.0D0
            DO 1  K = 1,M
    1         SUM = SUM + Q(K,I) * Q(K,J)
            Q(I,J) = Q(I,J) - SUM
    2     IF (I.EQ.J) GO TO 3
            Q(I,J) = Q(I,J) / Q(I,I)
       GO TO 4
    3  CONTINUE
            DET = DET + DLOG10(Q(I,I))
            Q(I,I) = DSQRT(Q(I,I))
    4 CONTINUE
      IDEXP = DET
      RPART = DET - IDEXP
      APART = DABS(RPART)
      IF (APART.LT.1.0D-20) DET = 1.D0
      IF (APART.LT.1.0D-20) GO TO 10
        DET = 10.0D0 ** RPART
   10 CONTINUE
      DO  7 J = 1,N
        DO  7 I = 1,J
          IF (I.LT.J) GO TO 5
            Q(J,J) = 1.0D0 / Q(J,J)
          GO TO 7
    5       SUM = 0.0D0
           M = J-1
            DO  6 K = I,M
    6         SUM = SUM - Q(I,K) * Q(K,J)
            Q(I,J) = SUM / Q(J,J)
    7 CONTINUE
      DO  9 J = 1,N
        DO  9 I = 1,J
```

```
      SUM = 0.0D0
      DO  8 K = J,N
8       SUM = SUM + Q(I,K) * Q(J,K)
      Q(I,J) = SUM
      IF (I.EQ.J) GO TO 9
      Q(J,I) = SUM
9 CONTINUE
  RETURN
  END
```

```
      FUNCTION DJUL(J1,M1,T)
C*
CC NAME          : DJUL  (FUNCTION)
CC
CC      XMJD= DJUL(J1,M1,T)
CC
CC PURPOSE       : COMPUTES THE MODIFIED JULIAN DATE (MJD)
CC                 FROM YEAR,MONTH AND DAY
CC                 MJD = JULIAN DATE - 2400000.5
CC
CC PARAMETERS    : J1 : YEAR (E.G. 1984)                I*4
CC                 M1 : MONTH(E.G. 2 FOR FEBRUARY)      I*4
CC                 T  : DAY OF MONTH                    R*8
CC
CC AUTHOR        : DJUL IS A MEMBER OF ASTLIB
CC                 ASTRONOMICAL INSTITUTE , UNIVERSITY OF BERNE
CC                 SWITZERLAND
CC
      REAL*8 DJUL,T
      J=J1
      M=M1
      IF(M.GT.2)GO TO 1
      J=J-1
      M=M+12
1     I=J/100
      K=2-I+I/4
      DJUL=(365.25D0*J-DMOD(365.25D0*J,1.D0))-679006.D0
      DJUL=DJUL+AINT(30.6001*(M+1))+T+K
      RETURN
      END
```

```
      FUNCTION THETA(T)
C*
CC NAME         :  THETA (FUNCTION)
CC
CC    TH=THETA(T)
CC
CC PURPOSE      :  COMPUTATION OF MEAN SIDERAL TIME GREENWICH
CC                 AT EPOCH T (MODIFIED JULIAN DATE)
CC                 SEE "EXPLANATORY SUPPLEMENT TO THE EPHEMERIS"
CC                 EDITION 1961
CC
CC PARAMETER    :  T : EPOCH IN MODIFIED JULIAN DATE
CC                 T AND THETA: REAL*8
CC
CC AUTHOR       :  THETA IS A MEMBER OF ASTLIB
CC                 ASTRONOMICAL INSTITUTE, UNIVERSITY OF BERNE
CC                 SWITZERLAND
CC
      REAL*8 T,TH,THETA,PI,ZPI
      PI=3.141592653589793D0
      ZPI=2*PI
      TH=T-15019.5D0
      THETA=279.6909833D0+360.9856473353D0*TH+.2902D-12*TH**2
      THETA=DMOD(THETA/180*PI,ZPI)
      RETURN
      END
```

IBM 3032 SAMPLE DATA (INPUT)

```
//LKED.SYSLIB  DD  DSN=UNB1.FORTLIB,DISP=SHR
//          DD
//          DD
//          DD
//          DD  DSN=A.M12129.SELIBOBJ,DISP=SHR
//          DD  DSN=A.M1212B.GPSLIB,DISP=SHR
//          DD  DSN=A.M1212B.ASTLIB,DISP=SHR
//GO.FT14F001 DD DUMMY
//GO.FT16F001 DD DUMMY
//GO.FT17F001 DD DUMMY
//GO.FT19F001 DD DUMMY
//*GO.FT19F001 DD DSN=A.M1212B.QUE.ALERT,UNIT=SYSDA,DISP=(NEW,CATLG),
//* SPACE=(TRK,(3,2),RLSE),DCB=(RECFM=FB,LRECL=150,BLKSIZE=15000)
//GO.FT25F001 DD DUMMY
//GO.SYSIN DD *
010401
  45  26  34 283  44  42 50.0015.
1983 213  21  10  00
1983 214   2  10  00
05
 5
   4 1983 192.49547046   63.1997  167.5589  .0039040  214.2885
  145.4408  2.00575775  717.980  180.002
   6 1983 193.34519674   63.7477   45.8370  .0024403  119.4508
  240.8805  2.00564376  717.958  179.997
   8 1983 192.41335369   63.2219  167.5884  .0031790  335.1053
   24.6996  2.00563908  717.943  179.993
   5 1983 186.05356987   63.7280  168.3037  .0040514  227.8061
  131.8517  2.00561920  717.967  179.998
   9 1983 191.40597304   63.5613   46.0623  .0092091   73.5887
  287.5003  2.00570750  717.977  180.001
//
```

IBM 3032 SAMPLE OUTPUT

YOUR LOCATION AND DESIRED TIME INTERVAL FOR ALERT

NASA PREDICTION BULLETIN (GAST USED)

GPS SATELLITES

SATELLITE ELEVATION > 15 DEGREES

LAT= 45DEG 26MIN 34SEC
LONG=283DEG 44MIN 42SEC
H= 50 METRES

BEGIN ALERT: DAY 213 , 1983 21:10: 0 ZCNT1 = 27100.

END ALERT: DAY 214 , 1983 2:10: 0 ZCNT2 = 30100.

HR	MIN	ZCNT	VISIBLE SATELLITES	ELEVATIONS	AZIMUTHS
21	10	27100	6 8	19 53	-40 -151
21	15	27150	6 8	22 55	-39 -149
21	20	27200	6 8	24 58	-39 -147
21	25	27250	6 8	26 60	-39 -145
21	30	27300	6 8	28 63	-38 -143
21	35	27350	6 8	30 65	-38 -140
21	40	27400	6 8	33 68	-38 -137
21	45	27450	6 8	35 70	-38 -133
21	50	27500	6 8	37 73	-38 -129
21	55	27550	6 8	40 75	-38 -123
22	0	27600	6 8	42 77	-39 -116
22	5	27650	6 8	44 79	-40 -106
22	10	27700	6 8	46 81	-41 -93
22	15	27750	6 8	49 82	-42 -75
22	20	27800	6 8	51 82	-43 -55
22	25	27850	4 6 8	16 53 81	-127 -45 -36
22	30	27900	4 6 8	18 55 80	-125 -47 -20
22	35	27950	4 6 8	20 57 78	-124 -50 -9
22	40	28000	4 6 8 9	22 59 77 16	-122 -53 0 -32
22	45	28050	4 6 8 9	24 61 75 18	-120 -57 6 -33

22	50	28100	4	6	8	9		26	63	72	20		-118	-61	11	-33		
22	55	28150	4	6	8	9		28	64	70	22		-116	-66	15	-33		
23	0	28200	4	6	8	9		29	65	68	24		-115	-72	19	-34		
23	5	28250	4	6	8	9		31	66	66	26		-112	-78	22	-35		
23	10	28300	4	6	8	9		33	67	64	28		-110	-85	25	-36		
23	15	28350	4	6	8	9		35	67	62	31		-108	-92	28	-37		
23	20	28400	4	6	8	9		37	66	59	33		-106	-99	30	-38		
23	25	28450	4	6	8	9		39	66	57	35		-104	-105	33	-39		
23	30	28500	4	6	8	9		40	65	55	37		-101	-111	35	-41		
23	35	28550	4	6	8	9		42	63	53	39		-99	-117	37	-43		
23	40	28600	4	6	8	9		44	62	51	40		-96	-122	39	-45		
23	45	28650	4	6	8	5	9	45	60	49	16	42	-93	-126	41	-107	-47	
23	50	28700	4	6	8	5	9	47	58	47	17	44	-90	-130	43	-105	-50	
23	55	28750	4	6	8	5	9	49	56	45	19	45	-87	-133	45	-103	-53	
0	0	28800	4	6	8	5	9	50	53	43	21	46	-84	-136	47	-101	-56	
0	5	28850	4	6	8	5	9	52	51	41	22	47	-81	-138	49	-99	-59	
0	10	28900	4	6	8	5	9	53	49	39	24	48	-77	-140	51	-97	-63	
0	15	28950	4	6	8	5	9	55	46	38	25	49	-74	-142	53	-95	-66	
0	20	29000	4	6	8	5	9	56	44	36	27	50	-70	-144	55	-93	-70	
0	25	29050	4	6	8	5	9	58	41	34	29	50	-66	-145	56	-91	-74	
0	30	29100	4	6	8	5	9	59	39	32	30	50	-62	-146	58	-88	-78	
0	35	29150	4	6	8	5	9	60	36	30	32	50	-57	-148	60	-86	-82	
0	40	29200	4	6	8	5	9	61	34	29	33	49	-53	-149	62	-84	-87	
0	45	29250	4	6	8	5	9	62	31	27	35	48	-48	-150	64	-82	-90	
0	50	29300	4	6	8	5	9	63	29	25	37	47	-43	-150	66	-79	-94	
0	55	29350	4	6	8	5	9	64	26	24	38	46	-37	-151	67	-77	-98	
1	0	29400	4	6	8	5	9	65	24	22	40	45	-31	-152	69	-74	-101	
1	5	29450	4	6	8	5	9	65	21	20	41	44	-26	-153	71	-72	-104	
1	10	29500	4	6	8	5	9	66	19	19	43	42	-20	-153	73	-69	-107	
1	15	29550	4	6	8	5	9	66	17	17	45	40	-13	-154	74	-66	-110	

1	20	29600	4	8	5	9	66	15	46	39	-7	76	-64	-112
1	25	29650	4	5	9		66	48	37		-1	-61	-115	
1	30	29700	4	5	9		66	49	35		5	-58	-117	
1	35	29750	4	5	9		66	51	33		11	-55	-119	
1	40	29800	4	5	9		66	52	31		17	-52	-121	
1	45	29850	4	5	9		65	54	29		23	-48	-122	
1	50	29900	4	5	9		64	55	26		29	-45	-124	
1	55	29950	4	5	9		63	56	24		34	-41	-126	
2	0	30000	4	5	9		63	58	22		39	-37	-127	
2	5	30050	4	5	9		62	59	20		44	-34	-128	
2	10	30100	4	5	9		61	61	18		49	-29	-130	

```
VISIBILITY (ELEVATION > 15 DEGREES) OF GPS SATELLITES


 LAT= 45DEG   26MIN   34SEC
LONG=283DEG   44MIN   42SEC
   H= 50 METRES

BEGIN ALERT:  DAY  213 ,  1983    21:10: 0   ZCNT1 =  27100.

  END ALERT:  DAY  214 ,  1983     2:10: 0   ZCNT2 =  30100.



GPS  4 -----***************

GPS  6 *****************---

GPS  8 *****************---

GPS  5 ----------**********

GPS  9 ------**************
      21  22  23   0   1   2
```

UNIVERSITY OF NEW BRUNSWICK
COMPUTER CODING FORM

```
NNSS NODAL CROSSING DATA (IPARA2=1)

 1983      60    13      06.926      26.807        7.84      53.98


NNSS KEPLERIAN PARAMETERS (IPARA2=2)

 1982     350    13      56.1484  3.3688047      62.4118  .0019975      .0021
     7456.59    218.8682  -.0000233     .005838     98.3882    .999983


NASA PREDICTION BULLETIN (GAST USED) (IPARA2=4)

    3 1983    96.11950818    63.6893    48.8951   .0020243   120.0282
  240.1574    2.00562924   717.980  180.002


NASA PREDICTION BULLETIN (GAST UNUSED) (IPARA2=5)

    3 1983    96.11900818    63.6893   .0020243  120.0282   240.1574
  2.00562924    717.980   180.002   187.80 1983   100   2 35.20
```

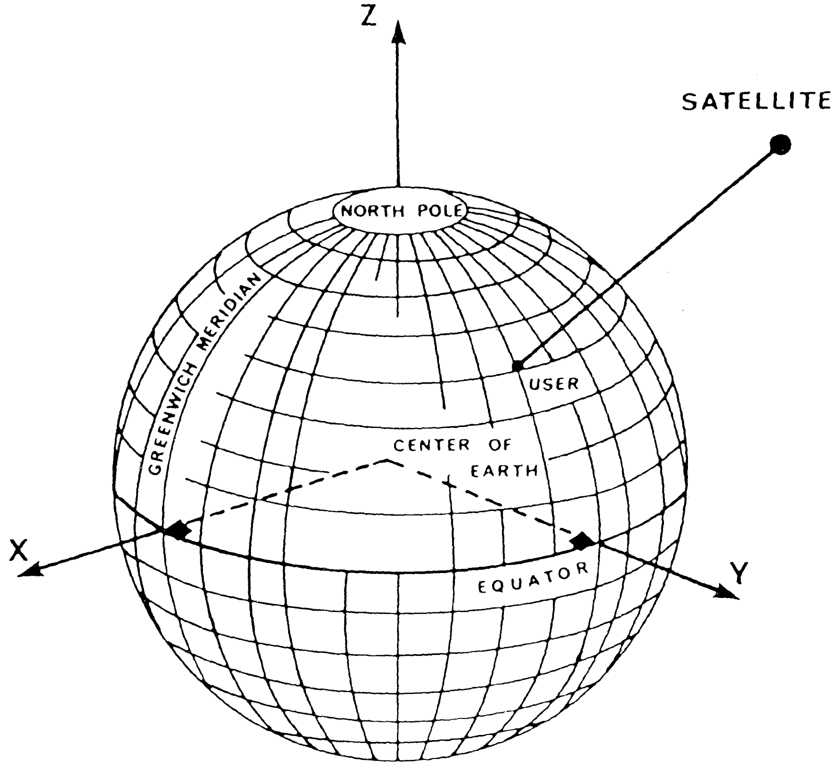


FIGURE 2.1

WGS-72 Reference System.