

ESD RECORD COPY

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

ESD ACCESSION LIST

ESTI Call No. AL 61614
Copy No. of CYS.

Technical Note**1968-10**

M. C. Crocker
E. H. Swenson

**Synchronous Satellite
Stationkeeping Simulation**

14 May 1968

Prepared under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lexington, Massachusetts



A00672960

The work reported in this document was performed at Lincoln Laboratory,
a center for research operated by Massachusetts Institute of Technology,
with the support of the U.S. Air Force under Contract AF 19(628)-5167.

This report may be reproduced to satisfy needs of U.S. Government agencies.

This document has been approved for public release and sale;
its distribution is unlimited.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

SYNCHRONOUS SATELLITE STATIONKEEPING SIMULATION

M. C. CROCKER
E. H. SWENSON

Group 63

TECHNICAL NOTE 1968-10

14 MAY 1968

LEXINGTON

MASSACHUSETTS

ABSTRACT

A program has been written to simulate the east-west stationkeeping of a synchronous satellite. Different ways of implementing the thrust sequence of rocket motors and solar sails are discussed.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	UNDERLYING THEORY TO THE COMPUTER SIMULATION	1
	A. Forces Acting On the Satellite	1
	B. Orbital Mechanics	1
	C. Longitudinal Perturbation Force	5
III.	STATIONKEEPING BY SOLAR SAILING	5
	A. Sail Type I	7
	B. Sail Type II	9
	C. Sail Type III	9
	D. Results of Computer Simulations	11
IV.	AUTOMATIC STATIONKEEPING	
	A. Stationkeeping by Constant Force	11
	B. Automatic Stationkeeping - Unidirectional Firing	14
	C. Effects of Sensor and Clock Errors	17
	D. Automatic Stationkeeping Unidirectional Firing With Force Cut-Down	19
	E. Automatic Stationkeeping Bidirectional Thrusting	23
Appendix A	Main Program of (SSS)	26
Appendix B		29
Appendix C	Integration Function and Subroutine	35
Appendix D	Double Precision Function FLAM (XLAM)	39
Appendix E	Subroutine NEWFOR	43
Appendix F		50
Appendix G		55
Appendix H	Subroutine MINMAX (X, N, XMIN, XMAX)	60
Appendix I	Subroutines ALVISE and CURT	62
Appendix J	Listing of the Program	65
References		108
Acknowledgement		109

NOMENCLATURE

a	=	area of the sail
C	=	velocity of light
\vec{F}	=	component of solar force perpendicular to the sail vector
$F_{a\lambda}$	=	average annual force on the satellite by radiation pressure in the longitudinal direction
F_r	=	force on the satellite in the radial direction
F_θ	=	force on the satellite in the colatitude direction
F_λ	=	force on the satellite in the longitudinal direction
$J_{n,m}$	=	coefficients of the tesseral harmonics
$J_{n,o}$	=	coefficients of the zonal harmonics
m	=	mass of the satellite
P_m	=	Legendre polynomials
r	=	distance from the center of the earth to the satellite
R	=	radius of the earth
\vec{R}	=	radius vector
S	=	solar constant = 1.39×10^3 watts/m ²
\vec{S}	=	sail vector
\vec{S}_N	=	normal to the sail
\vec{T}	=	orbital tangent vector
\vec{U}	=	sun vector
V	=	earth's potential
α	=	solar absorptivity
θ	=	colatitude of the satellite
λ	=	longitude of the satellite
λ_{nm}	=	longitude of the n and mth harmonic
μ	=	mass of the earth x gravitational constant
ω	=	earth's rotational rate
I_{sp}	=	specific impulse

SYNCHRONOUS SATELLITE STATIONKEEPING SIMULATION

I. INTRODUCTION

The motion of a satellite with a period of one sidereal day is perturbed by the gravitational attraction of the sun and moon, radiation pressure and the tangential component of the earth's gravitational geopotential. A Synchronous Satellite Simulation program (SSS) was written to study these effects and to determine the feasibility of maintaining a satellite near a specified synchronous position by automatically firing rocket motors and by solar sailing. The different parts of the SSS program are described in the Appendices.

II. UNDERLYING THEORY TO THE COMPUTER SIMULATION

A. Forces Acting On the Satellite

The influence of the gravitational attraction of the sun and moon contribute to a small diurnal oscillation in the longitude of the satellite of the order of 0.5 degrees¹ at the most. A typical daily longitude variation has been calculated by Molitor and Kaplan² and is shown in Fig. 1. However, the sun and moon's gravitational attraction does cause the orbit normal to precess³ so that the inclination changes by about 0.85 degrees per year.

Radiation pressure will change the eccentricity of the orbit a small amount for satellites with ordinary area to mass ratios that are symmetric about a plane that contains the center of mass of the satellite and is perpendicular to the equatorial plane. By making the satellite not symmetric, it is possible to East-West stationkeep a synchronous satellite by radiation pressure forces. Methods for doing this are presented in a paper by Black et al⁴.

B. Orbital Mechanics

The motion of a satellite is determined primarily by the earth's gravitational potential function

$$V = -\frac{\mu m}{r} \left[1 - \sum_{n=2}^{\infty} J_{n,0} \left(\frac{R}{r} \right)^2 P_n^0 (\cos \theta) + \sum_{n=2}^{\infty} \sum_{m=1}^{\infty} J_{n,m} \left(\frac{R}{r} \right)^n P_n^m (\cos \theta) \cos (\lambda - \lambda_{n,m}) \right] \quad (1)$$

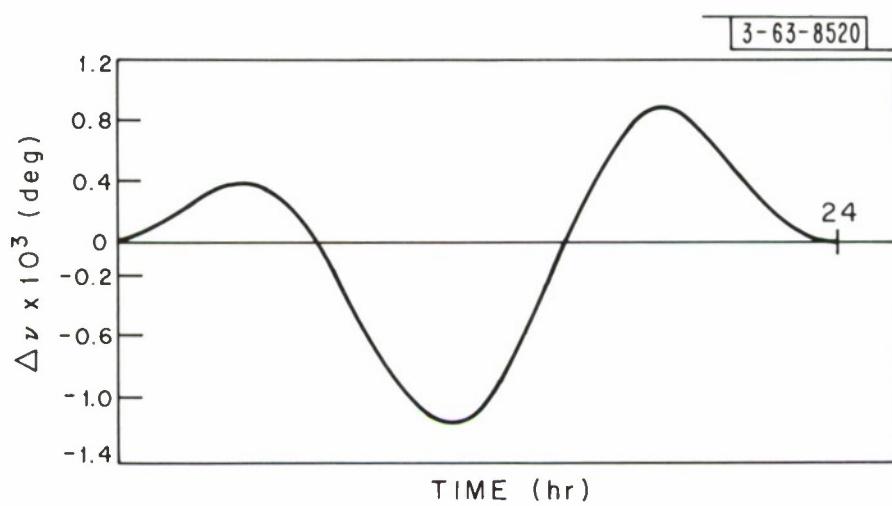


Fig. 1. Typical daily longitude variation of a 24-hour satellite due to solar and lunar perturbations.

For the purposes of studying east-west stationkeeping the effect of the gravitational attraction of the moon, sun and planets was neglected. The SSS program has the option of neglecting terms in the potential function greater than order two or three. The inaccuracies in measuring and producing surfaces with specified reflective properties produce uncertainties in the forces acting on the satellite which mask the above effects.

The motion of a 24-hour satellite is best studied by using an earth fixed system of coordinates rotating at the earth's angular rate ω .^{5,7} The kinetic energy of the satellite expressed in this rotating system is

$$T = \frac{1}{2} m [\dot{r}^2 + r^2 \dot{\theta}^2 + r^2 \sin^2 \theta (\dot{\lambda} + \omega)^2] \quad (2)$$

Let us denote the forces on the satellite produced by rocket thrusters or radiation pressure devices as F_r , F_θ , F_λ . Now by using Lagrange's equations and retaining terms to the second order, the following equations of motion result.

$$\ddot{r} - r \dot{\theta}^2 - r \sin^2 \theta (\dot{\lambda} + \omega)^2 = -\mu/r^2 + 3\mu J_{20} R^2 (3 \cos^2 \theta - 1)/2r^4$$

$$+ 9\mu J_{22} R^2 \sin^2 \theta \cos 2(\lambda - \lambda_{22})/r^4 + F_r/m \quad (3)$$

$$\begin{aligned} \ddot{\theta} &= \frac{3\mu J_{20} R^2}{2r^5} \sin 2\theta - \frac{3\mu J_{22} R^2}{r^5} \sin 2\theta \cos 2(\lambda - \lambda_{22}) \\ &+ \frac{(\dot{\lambda} + \omega)^2}{2} \sin 2\theta - 2\dot{r}\dot{\theta}/r + F_\theta/mr \end{aligned} \quad (4)$$

$$\begin{aligned} \ddot{\lambda} &= \frac{6\mu J_{22} R^2}{r^5} \sin 2(\lambda - \lambda_{22}) - 2 \operatorname{ctn} \theta \dot{\theta}(\dot{\lambda} + \omega) + F_\lambda/mr \sin \theta \\ &- \frac{2\dot{r}(\dot{\lambda} + \omega)}{r} \end{aligned} \quad (5)$$

When the force terms are set to zero F_θ , F_r , and F_λ the program has been found to simulate the motion of Syncom II to a high degree of accuracy as shown in Fig. 2.⁶ The discontinuity in the curves is due to a rocket motor firing.

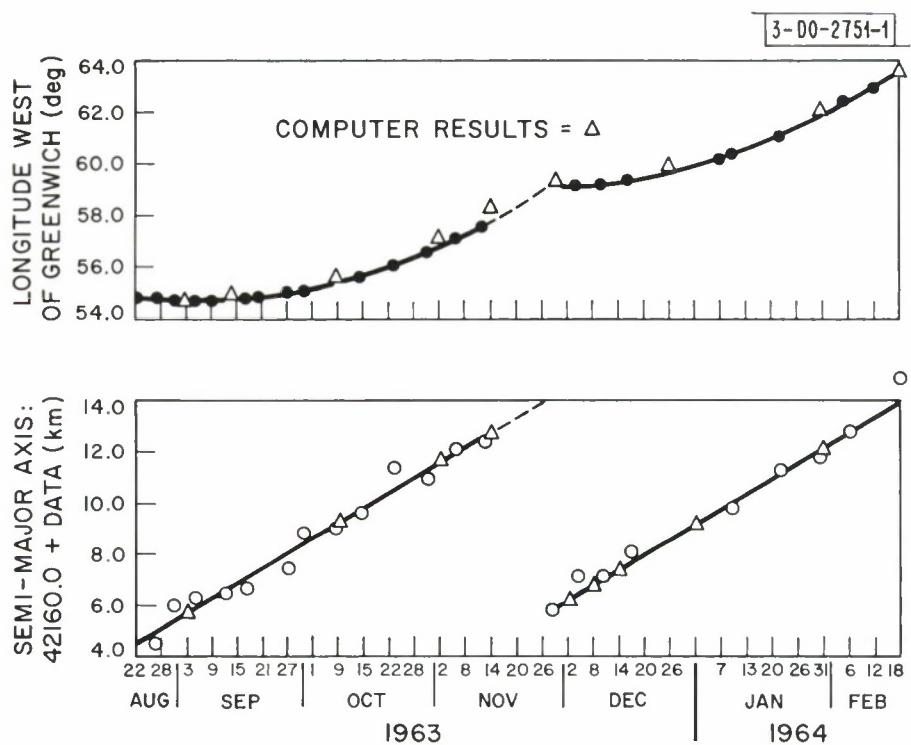


Fig. 2. SYNCOM II drift.

C. Longitudinal Perturbation Force

The longitudinal force per unit mass on a satellite due to the earth's geo-potential is given by taking the gradient of the potential

$$f_\lambda/m = -\nabla_\lambda V/m = \frac{1}{mr \sin \theta} \frac{\partial V}{\partial \lambda} = \frac{6J_{22}R^2\mu}{r^4} \sin \theta \sin 2(\lambda - \lambda_{22}) \quad (6)$$

For an equatorial orbit $\theta = 90^\circ$ and

$$f_\lambda/m = -5.568 \times 10^{-8} \sin 2(\lambda - \lambda_{22}) \text{ newtons/kilogram} \quad (7)$$

The stable equilibrium points are located at the positions $(\lambda - \lambda_{22}) = 90^\circ$ and 270° .

If a satellite were placed at these points, the satellite would remain there or oscillate about them with a small amplitude. No stationkeeping apparatus would be required if the satellite were placed at these points and the portion of the earth "visible" to the satellite was satisfactory. The stable equilibrium points are located off the western coast of South America and over the Indian Ocean. However, if the satellite is used for communications between the United States and Europe or the United States and the Far East the satellite will require stationkeeping apparatus. The visibilities of satellites placed for communication between the United States and Europe and between the United States and the Far East are shown in Fig. 3.

III. STATIONKEEPING BY SOLAR SAILING

The instantaneous radiation pressure forces, on the sail of a satellite, which has its plane in the north-south direction are given by the following vector equations, assuming there is no angular dependence of the solar absorptivity. These equations hold for any orbit.

$$\begin{aligned} F_\lambda &= \frac{Sa}{c} (h(2-\alpha) (\vec{S}_N \cdot \vec{U})^2 | -\vec{S}_N \cdot \vec{T} | \\ &\quad - s\alpha |\vec{S}_N \cdot \vec{U}| (\vec{U} \cdot \vec{S}) \sin [\cos^{-1}(-\vec{S}_N \cdot \vec{T})]) \end{aligned} \quad (8)$$

3-00-2736

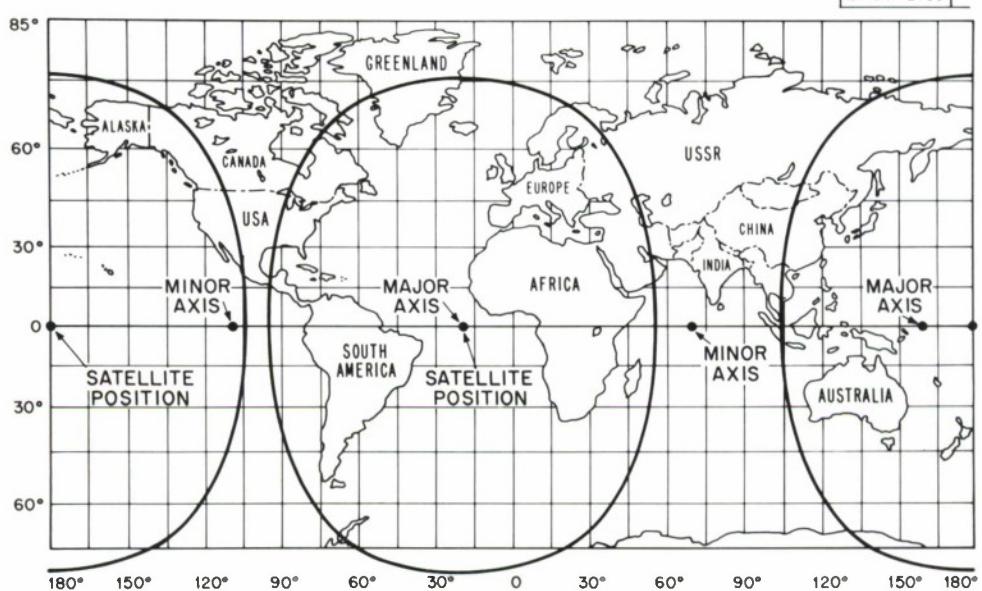


Fig. 3. Visibilities of satellites placed at stable equilibrium points.

$$F_r = \frac{Sa}{c} [g(2 - \alpha) (\vec{S}_N \cdot \vec{U})^2 \sin [\cos^{-1}(-\vec{S}_N \cdot \vec{T})] - f\alpha |\vec{S}_N - \vec{U}| |\vec{S}_N - \vec{T}| (\vec{U} - \vec{S})] \quad (9)$$

$$F_\theta = -\frac{Sa}{c} \alpha \cdot |\vec{S}_N \cdot \vec{U}| U_z \quad (10)$$

where h , s , g , f are +1 with the following exceptions:

$$\begin{aligned} h &= -1 \text{ when } (\vec{F} \cdot \vec{T}) \leq 0 \\ s &= -1 \text{ when } (\vec{S} \cdot \vec{T}) \leq 0 \\ g &= -1 \text{ when } (\vec{F} \cdot \vec{R}) \leq 0 \\ f &= -1 \text{ when } (\vec{S} \cdot \vec{R}) \leq 0 \end{aligned}$$

A. Sail Type I

The simplest type of sail for a satellite that has one side oriented to the earth and one side oriented toward the north direction, is a sail whose plane is coincident with the radius to the satellite from the center of the earth and the north-south direction. (See Fig. 4) This sail has a low solar absorptivity on one side and a high solar absorptivity on the other. The thermal emissivity of both sides is approximately the same. Under these conditions the force resultant from thermal radiation from both sides of the sail cancels for a thin sail. The net force along the orbit is then due only to solar radiation pressure. The average annual force in the longitudinal direction is given by

$$F_{a\lambda} = \frac{\alpha_2 - \alpha_1}{4} \frac{Sa}{c} \langle \cos i \rangle \quad (11)$$

where $\langle \cos i \rangle$ is average annual effect of the sun's movement above and below the orbital plane of the satellite. If one assumes that the earth's orbit about the sun is circular and an equatorial orbit for the satellite $\langle \cos i \rangle \approx 0.96$ and α_2 is the solar absorptivity of the sail facing in the direction of orbital

3-63-7841

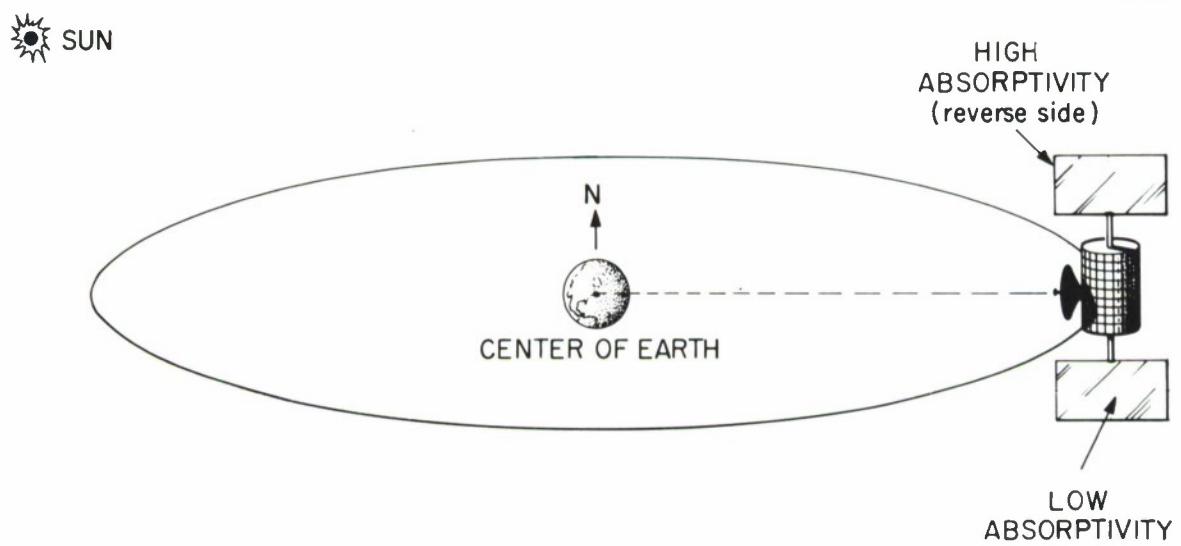


Fig. 4. Solar sail Type I.

motion and α_1 is solar absorptivity of the opposite face. This type of sail does not lend itself to being adapted to the stationkeeping methods to be discussed in Section IV.

B. Sail Type II

The second type of sail is similar to the first except both sides of the sail have the same highly reflective surface and can be rotated along the earth satellite line so that the plane of the sail goes from the north-south to east-west direction. The sail is left in the north-south portion for half the orbit starting (or ending) with the sun-earth line and the east-west position for the other half of the orbit. (See Fig. 5) The average annual force in the longitude direction is

$$F_{a\lambda} = \left[\pm \left(\frac{2 - \alpha}{4} \frac{Sa}{c} \right) \langle \cos i \rangle - \alpha \frac{Sa}{2\pi c} \langle \sin 2i \rangle \right] \quad (12)$$

The positive sign is used when the north-south position is taken for half the orbit starting with the earth sun line and the negative sign is used when the north-south position ends with the earth sun line. This type of sail is more complicated than the first type because it requires a rotating mechanism and sensor logic to determine the location of the earth sun line. It does require less sail area than the first type. It is also adaptable to the previously mentioned methods of thrusting by rotating the sail into or out of position with respect to the earth-sun line.

C. Sail Type III

The third type of sail has both sides highly reflective. It rotates about the north-south direction continuously with a period of 12 hours. (See Fig. 6) The average annual force in longitudinal direction is

$$F_{a\lambda} = \frac{Sa}{\pi c} \frac{2}{3} + 2 (1 - \alpha) \sin 2\varphi \langle \cos i \rangle \quad (13)$$

where φ is the angle between the plane of the sail and the projection of the earth sun line on the equatorial plane when the satellite is crossing the projection of the earth sun line on the equatorial plane. This type of sail requires the least



3-63-7843

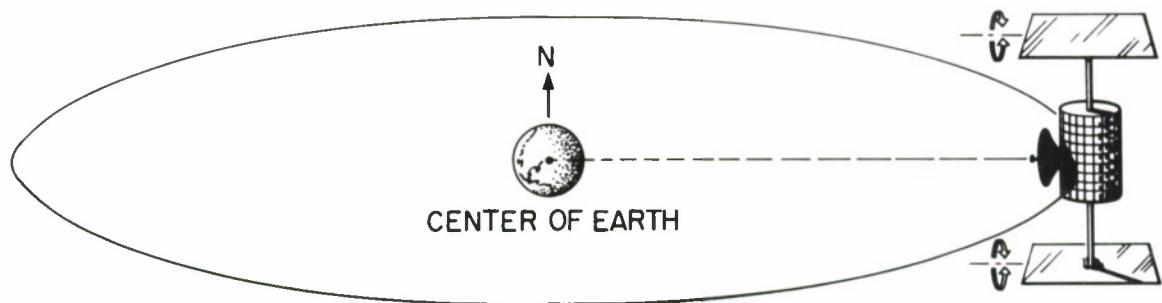


Fig. 5. Solar sail Type II.



3-63-7842

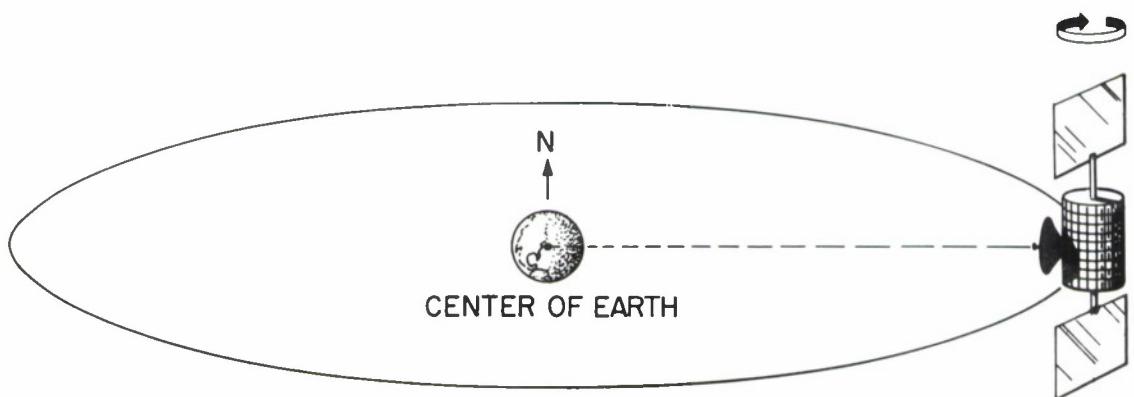


Fig. 6. Solar sail Type III.

amount of sail area for stationkeeping. If one sets $\varphi = 45^\circ$ and assumes a solar absorptivity of 0.9 for one side of the Type I sail and 0.05 for the other types of sails, the relative area needed to produce the same average annual force in the longitudinal direction for Type I, Type II and Type III sails is in the ratio of 3.84 : 1.70 : 1 respectively. A typical satellite with a mass of 100 Kg would require a Type III sail with an area of 1.47 square meters to stationkeep at the position of highest longitudinal force due to the earth's geopotential. By varying the value of φ it is possible to vary the longitudinal force along or against the orbital motion. Therefore, this type of sail is adaptable to any method of automatic stationkeeping.

D. Results of Computer Simulations

The computer simulation of stationkeeping a synchronous satellite by solar sailing revealed little difference between the three types of sail considered for stationkeeping. A typical stationkeeping simulation is shown in Fig. 7. In this simulation sail Type III was used with $\varphi = -45^\circ$ and $\alpha = .1$. The area of the sail was 1.35 square meters for a 100 Kg satellite. The satellite was stationkept near a position at which a maximum longitudinal force occurs. (296° E longitude)

IV. AUTOMATIC STATIONKEEPING

A. Stationkeeping by Constant Force

Different methods of applying a tangential force to counteract the longitudinal perturbational force were simulated by the program. The simplest of these is to fire a rocket motor with the same value of longitudinal force that is acting upon the satellite. However, in practice the force cannot be exactly matched for each longitudinal position of the satellite. The method is limited if there is no control over the force level, once the satellite is placed in a longitudinal position within the longitude regions $45^\circ < \lambda - \lambda_{22} < 135^\circ$ and $225^\circ < \lambda - \lambda_{22} < 315^\circ$. If the force level is not exactly matched for a particular longitudinal position the satellite will undergo an oscillation about the longitude for which it is exactly matched. A simulation of an unmatched force level was done by the SSS program and is shown in Fig. 8. The satellite was initially at 291° longitude and the force level was equal to the longitudinal force at 283.5° longitude.

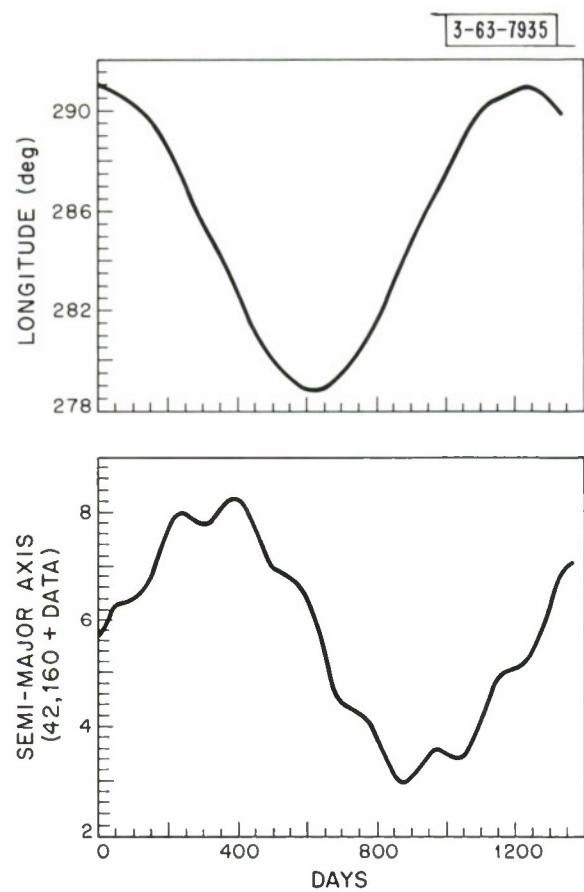


Fig. 7. Variations in semi-major axis and longitude versus time.

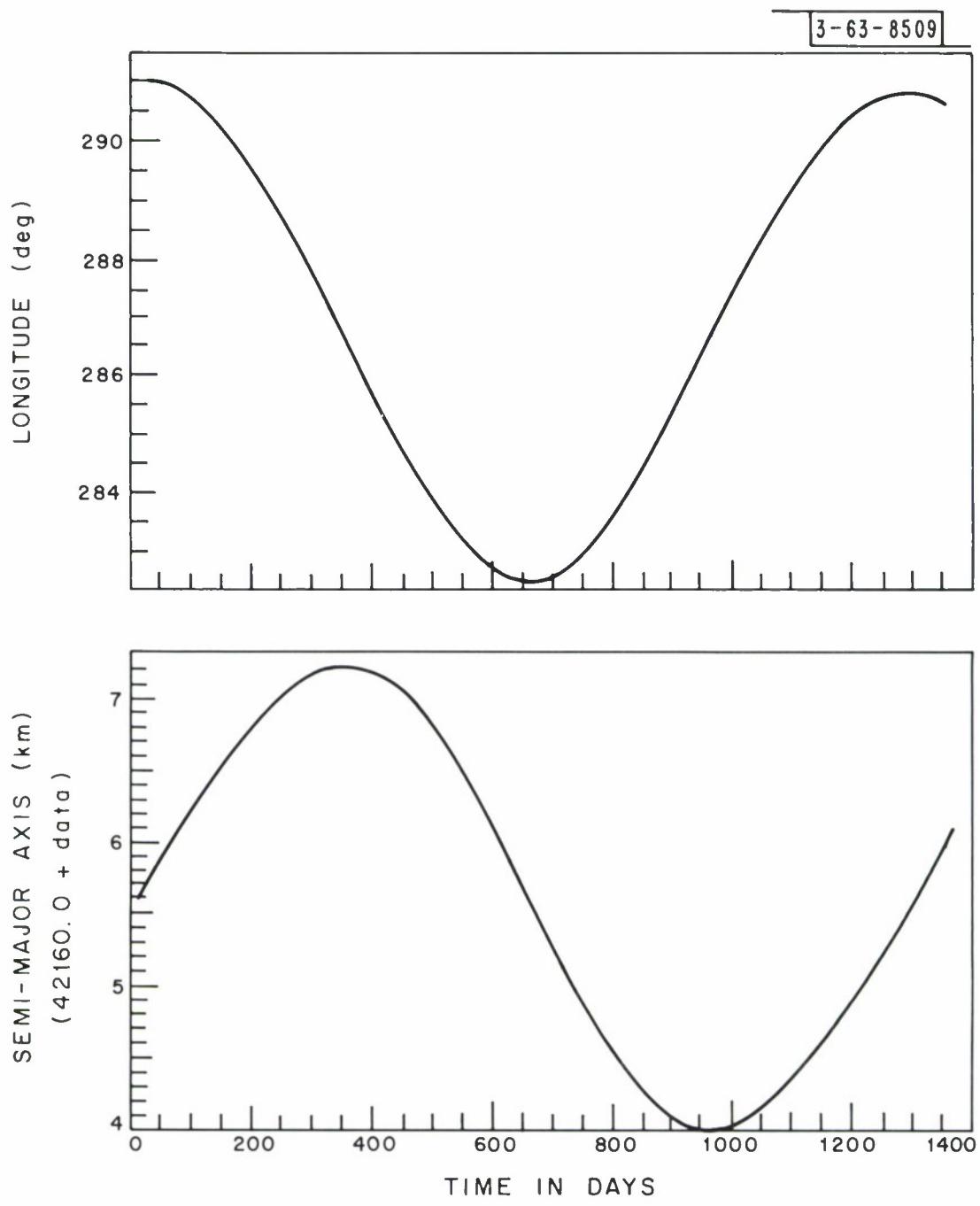


Fig. 8. Simulation of an unmatched force level.

B. Automatic Stationkeeping - Unidirectional Firing

Automatic stationkeeping was proposed by W. E. Morrow⁸ in a memo dated 1 July 1965. A block diagram of the logic required by the satellite for internally computing its position and rocket firing times is shown in Fig. 9. This description does not describe the LES-6 stationkeeping logic.

It is the purpose of the coincidence logic to provide a pulse derived from fan beam sensors at a predetermined satellite position with respect to the earth and sun. The satellite may be assumed spinning so that the output from the sensors are in the form of pulses or for non-spinning satellite the output from the sensors may be chopped to produce the pulses. This coincidence pulse permits comparison between a running clock and a fixed clock in the decision logic. Almost any position in the orbit can be used for establishing the satellite's position where the optical sensors can view an illuminated portion of the earth. For definiteness let us pick the 6 a.m. position shown in Fig. 10.

The decision logic determines whether the satellite is early or late in arriving at the "6 a.m. position" each day. The device compares the coincidence signal with the output signal from the clock logic.

If the signal from the coincidence logic occurs in position A of Fig. 11, the satellite is early for the three days shown and no signal is sent to the thrust logic. If a signal from the coincidence logic falls in position B then a signal is sent to the thrust logic.

The thrust logic's purpose is to fire the rocket motor in the direction of the orbit tangent for a specified length of time near the 6 a.m. position. This is repeated daily until the satellite is "on station". The thrusting is then stopped until another signal is received.

The clock must have an accuracy of 5×10^{-10} /day for 5 years and put out a signal (alarm) at 10 hr 12 min GMT each day. This signal is sent to the decision logic for further processing.

The purpose of the clock logic is to apply a daily correction to the clock to allow for the fact that the sun is ahead of or behind the clock according to

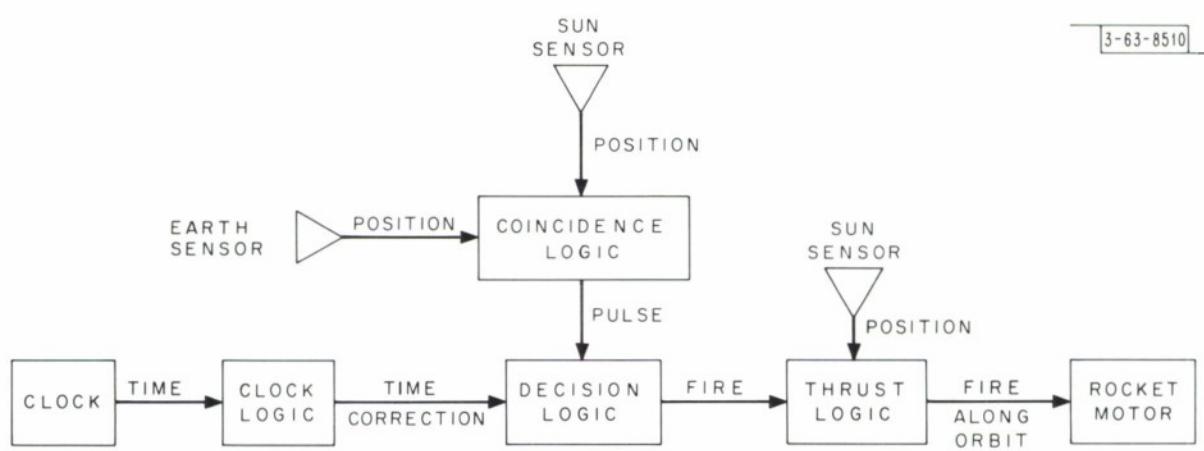


Fig. 9. Satellite logic for computing position and rocket firing times.

3-63-8512

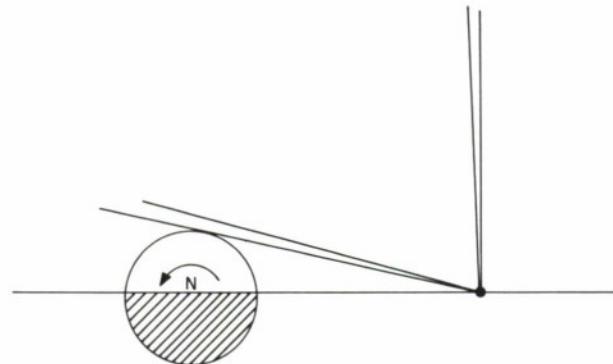


Fig. 10. 6 a.m. position of satellite in orbit.

3-63-8513

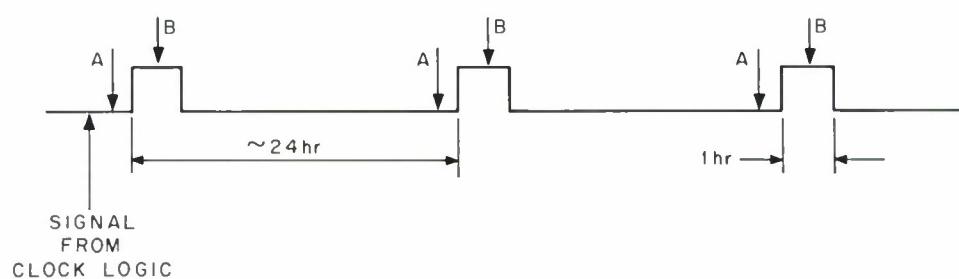


Fig. 11. Signal from clock logic.

the time of year. The correction that must be added to the time is shown in Fig. 12. The curve shown in Fig. 12 has to be approximated by a function that is easy to program into the satellite logic. The errors associated with the degree of exactness to which the clock logic will correct for the difference between apparent solar and solar mean time were accounted for from tables furnished by A. Braga-Illa for LES-6.

As an example of how the correction is applied, let us consider the correction at 15 January 1967. To do an exact calculation of the time to be added we would have to go through an iterative process. However, for our purposes the correction may be taken as the correction that occurs on January 15.0 which is 9.0 min. Therefore, the clock logic is to put out a signal to the decision logic which starts at 10 hr 21 min GMT and lasts for one hour. When the sun is ahead of the clock the correction has to be subtracted from 24 hours and the result added to the time of the previous days output.

The method is a little cumbersome in logic design. A better way to do it would be to shift 0 time base line so that all corrections become additions.

It is possible to do without the clock logic by using an optical sensor that senses the declination of the sun and relative position of the earth satellite line at the same time to correct for the actual longitudinal motion of the sun relative to the "mean" sun in the celestial sphere. The sensor can be visualized as having a cluster of pencil beams arranged in the familiar analemma found on some globes.⁹ A sensor of this type is being constructed by C. Burrowes for use in LES-6.

C. Effects of Sensor and Clock Errors

There are errors associated with the measurement of longitude due to the beamwidth of the sensors and the degree of exactness to which the clock logic will correct for the difference between apparent solar and solar mean time. There are also errors induced because the LES-6 satellite spin axis is not aligned perfectly in the north-south direction. The source of error has been investigated by B. J. Moriarty.¹⁰ The shape of the probability density function of the sum of the errors is not known. Therefore, a uniform probability

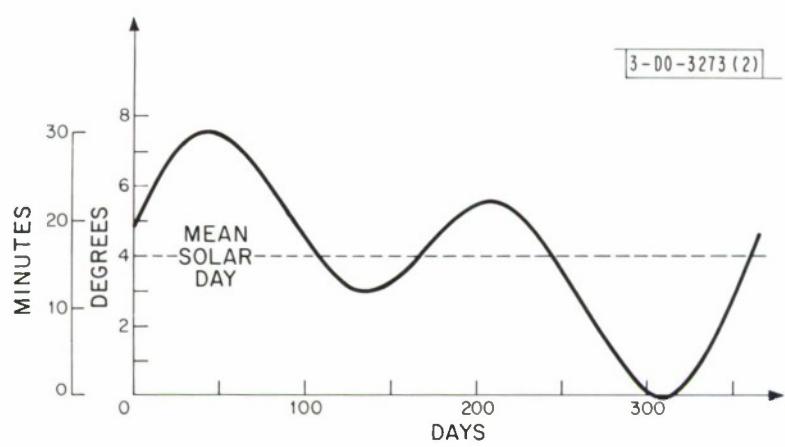


Fig. 12. Ephemeris correction vs. time of the year.

function with a width of one degree of longitude was assumed for use in the SSS program. The errors associated with the analemma sensor for LES-6 were separately accounted for from tables furnished by C. Burrowes.

The influence that the errors have on the motion of the satellite is shown in Figs. 13 and 14. Fig. 13 is the result of an SSS computer simulation of an automatic stationkeeping system using unidirectional firing with no errors present in determining longitude. Fig. 14 is the result of a simulation under the same conditions as the simulation of Fig. 13 except the clock logic errors and sensor errors are introduced. The errors do not significantly affect the range of longitudes over which the satellite may be stationkept. The errors in longitude sensing were less than 0.7 degrees due to clock errors and less than 0.5 degrees due to random sensor errors.

D. Automatic Stationkeeping Unidirectional Firing With Force Cut-Down

The best method of automatic stationkeeping so far proposed is that proposed by Roger Brockett.¹¹ This method uses the least amount of fuel and stationkeeps within the smallest range of longitude. As an example of how this method operates, let us assume the satellite is initially located at a longitude such that the earth's gravitational component along the orbit will cause the longitude of the satellite to decrease in time. The longitude of the satellite is allowed to decrease until it goes past a selected longitude, λ_o . At this point the rocket motors are fired for a specified length of time such that the impulse given per day is approximately five times the "impulse" given to the satellite by the earth's geopotential. This is the same as the method previously described. However, the longitude is sampled once per day and its value is stored in a memory bank in the satellite. If the longitude measured on a given day is greater than that measured the day before, the rocket impulse given per day is reduced until the satellite again passes the selected longitude, λ_o , at which time the thrusting is stopped. The amount that the impulse may be reduced is a variable that may be put in the input to the SSS program. A typical example of this stationkeeping method is shown in Fig. 15.

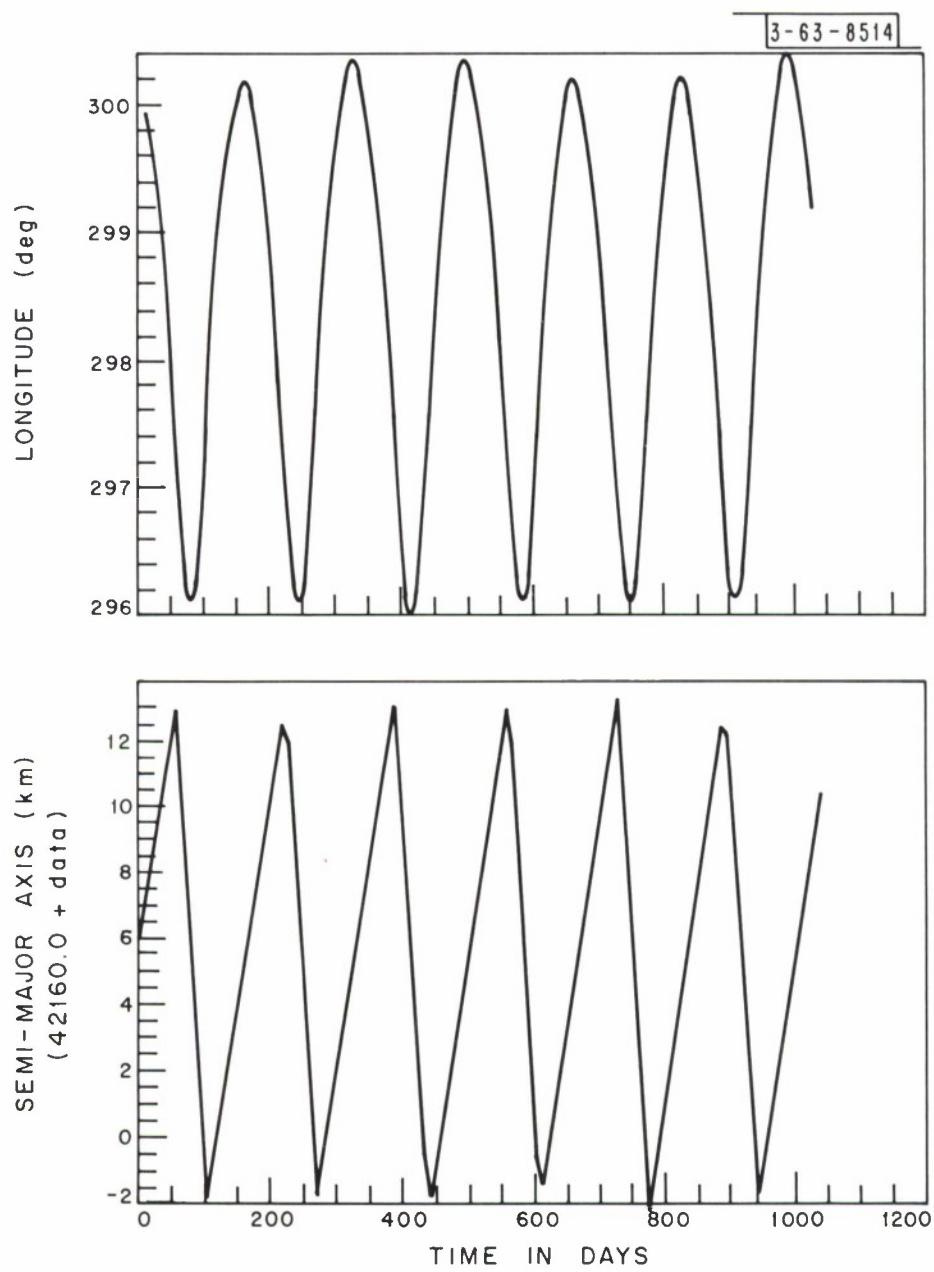


Fig. 13. Results of automatic stationkeeping with unidirectional firing with no errors in determining longitude.

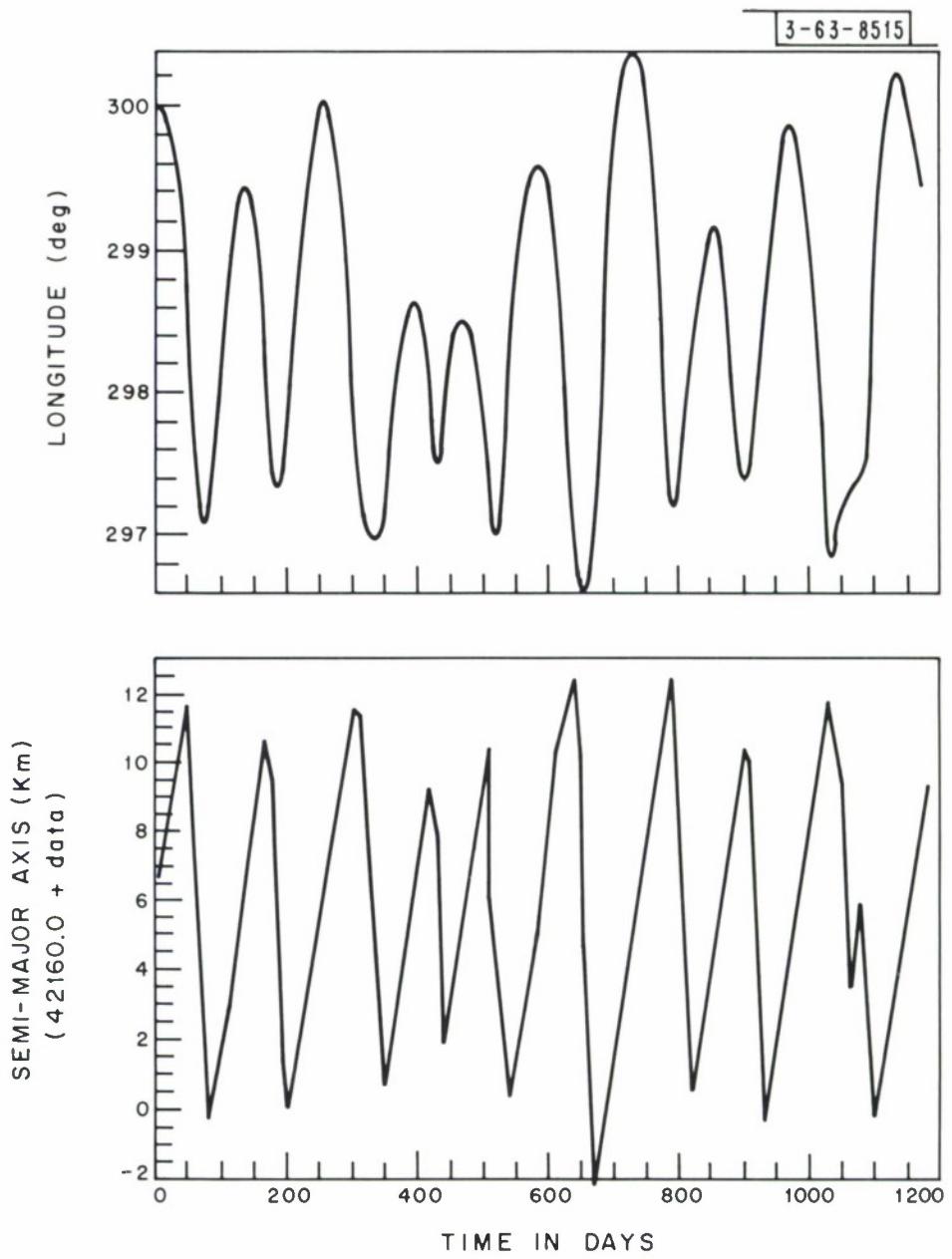


Fig. 14. Results of automatic stationkeeping with unidirectional firing with clock logic error and sensor errors introduced.

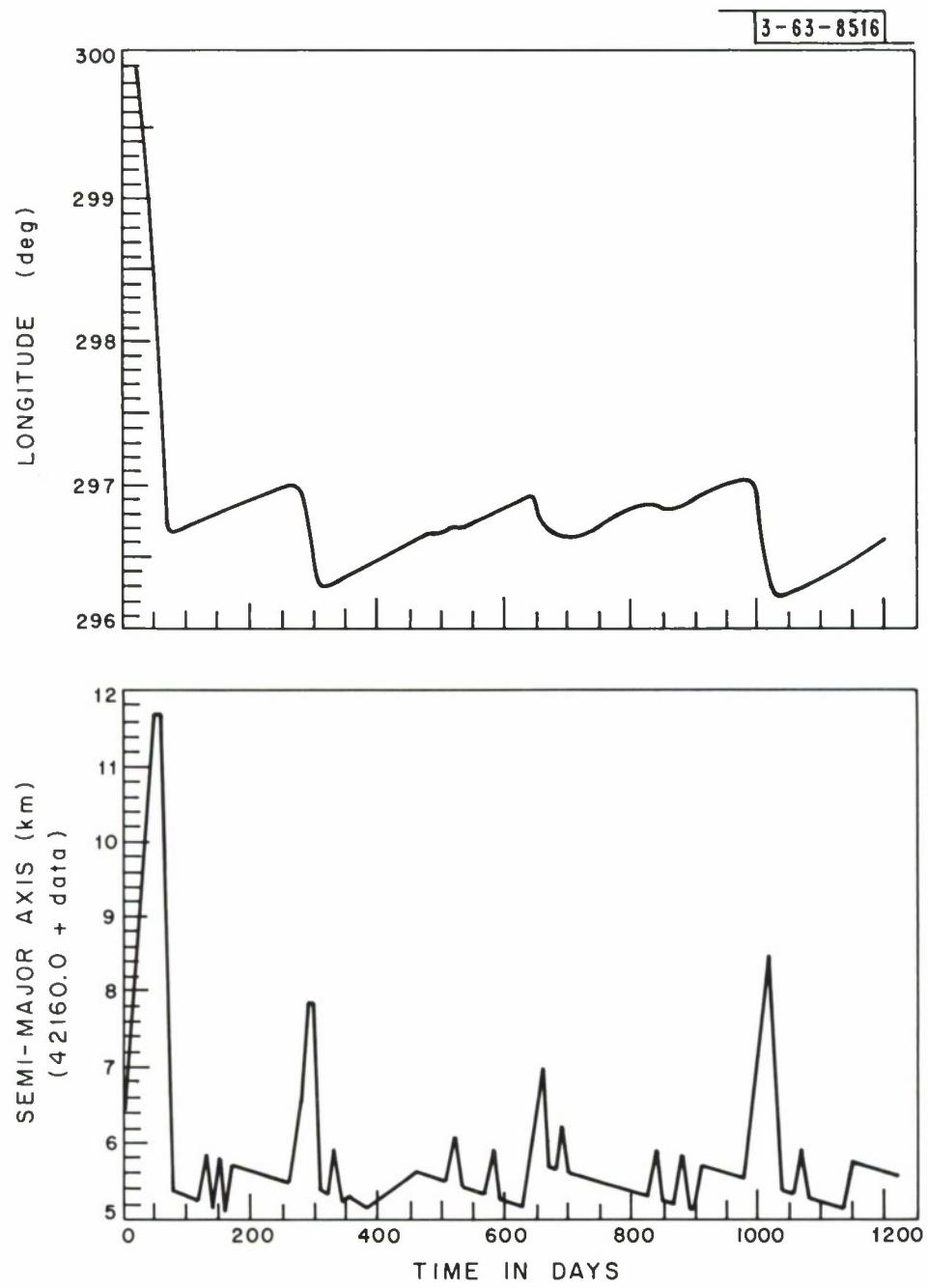


Fig. 15. Results of automatic stationkeeping with unidirectional firing with force cut-down.

E. Automatic Stationkeeping Bidirectional Thrusting

It is possible to stationkeep with a system which thrusts in one direction when the satellite exceeds a given longitude and in the other direction when the satellite is less than another given longitude. In other words, whenever the satellite has drifted out of a given band of longitude, thrusting is done to return the satellite to the band. This method of stationkeeping is not recommended for use with thrust systems that consume fuel at a low specific impulse because a lot more fuel is needed than with the methods previously mentioned. However, it is adaptable to solar sailing techniques where no fuel is used. A simulation of the bidirectional technique is shown in Fig. 16. This simulation was run under the same conditions as the unidirectional firing simulation of Fig. 13. The fuel consumption for stationkeeping a 132 Kg satellite for 1000 days was 3.85 Kg for a rocket system which has an $I_{sp} = 70$ for bidirectional thrusting. Under the same conditions only 0.86 Kg are needed for unidirectional thrusting.

Another method of bidirectional thrusting has been proposed by A. Braga-Illa. The logic of this method is explained in Appendix E describing subroutine Newfor of the SSS program. It is well suited for solar sailing techniques. A simulation of this method is shown in Fig. 17.

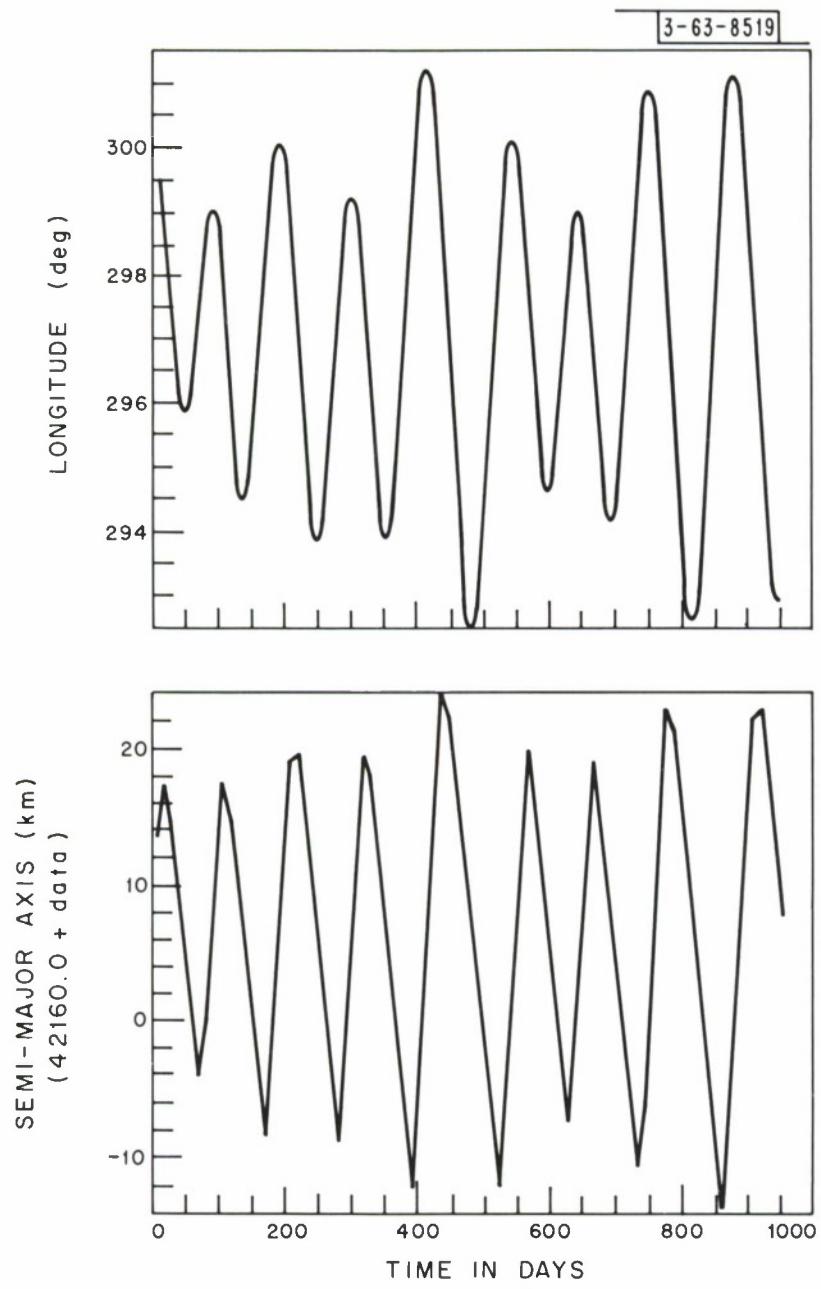


Fig. 16. Results of automatic stationkeeping using bidirectional firing.

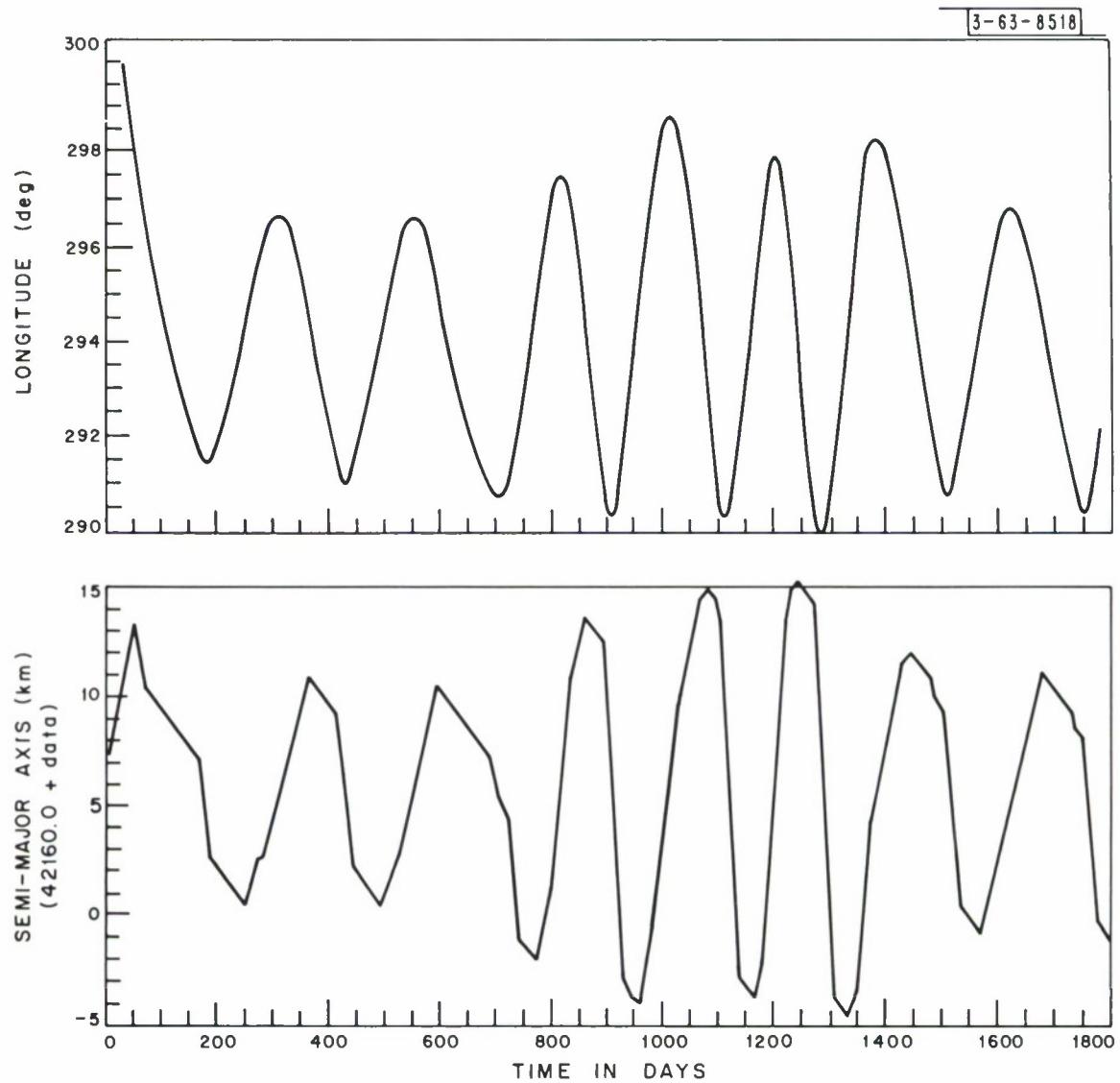


Fig. 17. Results of automatic stationkeeping using A. Braga-Illa's force scheme.

APPENDIX A

MAIN PROGRAM OF (SSS)

The main routine of the SSS program issues calls to the subroutines, solves the equations of motion, and produces the output of the program (Fig. A-1 and Appendix J). The initial section of comment cards gives a brief description of the program, its control variables and how to run it. The first executable statements are definitions of constants. The Main program calls GETINP to obtain the input data and control variables. After a sequence of housekeeping details -- moving variables to and from commons, changing variable types, and changing variables in degrees to radians -- the values necessary for the first entry into NEWFOR, FLAM, and CIND are initialized. These variables are initialized after the call to GETINP so that they are reinitialized at the beginning of subsequent sets of input data. The control variable I1 determines if the branch using second harmonics in the equations of motion or the branch using third harmonics is taken. In either case calls are issued to SSL and FLAM. SSL initializes CIND for solution of differential equations; FLAM returns the force in the λ direction. Certain logical paths determined by the input data return other variables to Main through COMMON (see Appendix D). After the call to FLAM, D2R, D2XLAM and D2THET are calculated. Both branches meet at statement 23, where three checks are made. First, is this time through the end of a day? Second, is this time through the end of ten days? Third, is this time through the end of amount of time asked for? If answer to all three questions are no, then it is end of one of the 86,400 times per day that the variables D2R, D2XLAM, and D2THET are calculated. At statement 6 the calls to FINDV, and DPNV provide T, R, THETA, XLAM, DR, DTHETA, and DXLAM for the following loop. Again a branch is taken based on I1. If it is the end of a day, the following are printed out.

TDAYS	days since beginning of this set of input data
R	radius of orbit
DR	change in radius in last loop of the day
TDEG	THETA in degrees

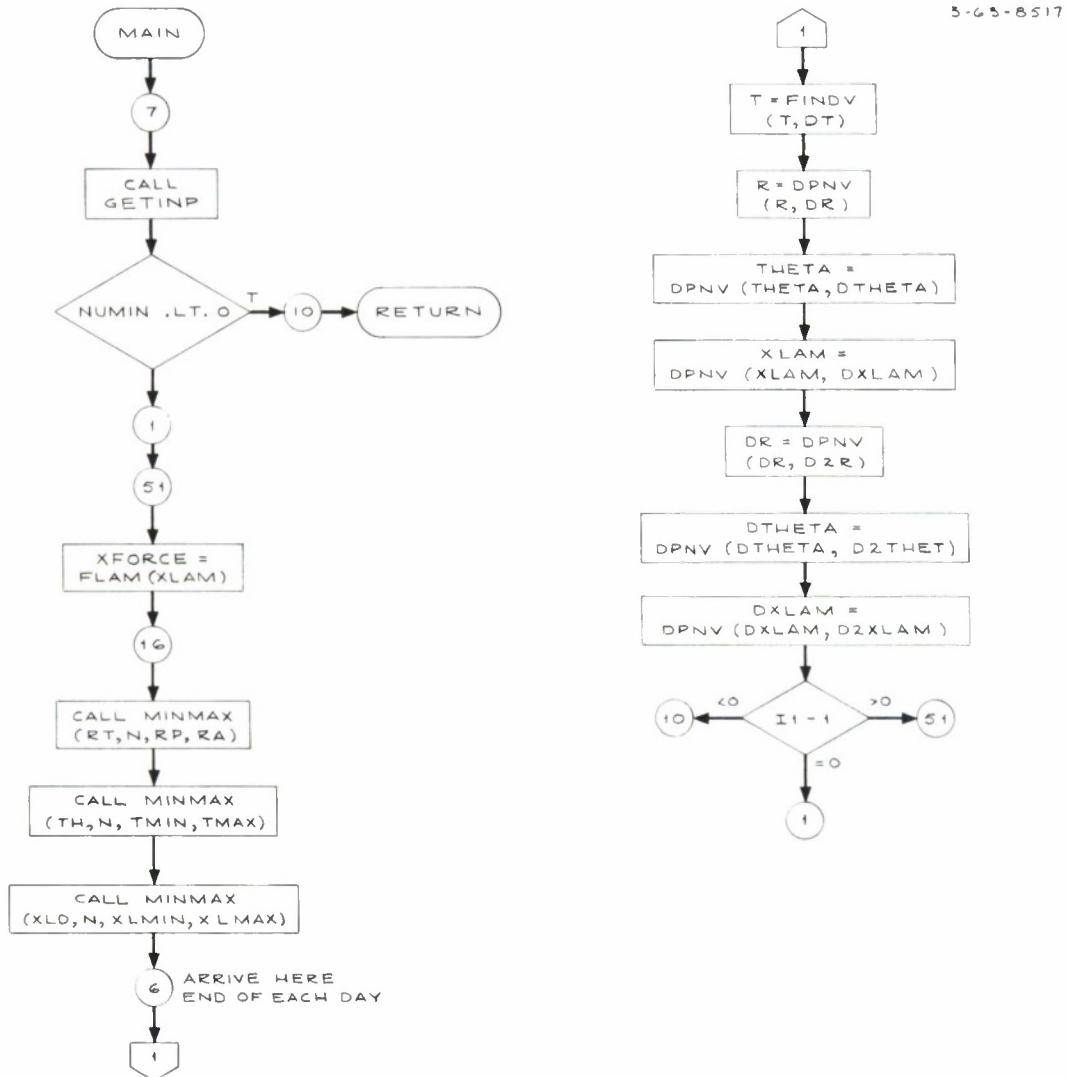
DTDEG	change in THETA in last loop of the day
XLDEG	XLAM in degrees
DXLDEG	change in XLAM in last loop of the day
XELONG	east longitude in degrees

After this printout a branch is made to question three. If the answer to question two is yes also, there precedes this printout a line of print containing the following averaged over previous 10 days.

A	radius
E	eccentricity of the orbit
XI	inclination
XIM1D	minimum inclination
XLMAD	maximum inclination
DIFFER	XLMAD - XLMID

Every ten days there is also punched a card containing TDAY, A, and XLAM, to be used by a plotting program. After this there is a branch to question 3. If the answer to question 3 is yes, and the amount of output for one set of data is finished, the program branches to 7, CALL GETINP. The last set of data contains NUM = -1, which causes program to terminate normally. Each set of data, except the last, takes 1 1/2 - 2 1/2 hours of computer time.

MAIN PROGRAM FLOW CHART



APPENDIX B

The subroutine GETINP establishes the logical variables, all the input, and the modes of output. There are eight classes of integers that are used in conjunction with the logical variables: I1, I2, I3, I4, I5, I6, I7, I8. All the parameters are initialized in GETINP and then those that are to be changed are read in through a namelist setups. Using the logical parameters and the numerical parameters, the subroutine SETUP and its entry point RITOUT prepare the variable format for PRINT and/or PUNCH so that at the beginning of each output one may quickly review the particular set of parameters used. Control is then returned to the main program which then prints the titles for the output, prints and/or punches the orbital information every ten days.

I1 determines whether the calculations in the main program use second order harmonics or third order harmonics. I1 is set to 1 which selects the second order harmonics. If third is true, I2 then equals 2 and the third order harmonics are used. I2 controls the punched output. I2 is initially 1 which produces punched output. When no punched output is desired, I2 should read in as 2. The punched output is used by a separate program Z-PLOT which produces one frame of alphanumeric data and a second frame with 2 graphs longitude vs. days and semi-major axis vs. days. I3 controls the solar time correction and primitive uses of a solar sail. I3 is initialized as 1, no correction. If I3 equals 2 or linear is true, there is a linear solar time correction. If I3 equals 3 or LINRAN is true, there is a linear solar time correction with random error. If I3 equals 4 or perfect is true, the solar time is corrected perfectly. If I3 equals 5 or STEP is true, the solar time is corrected by a step-function. When I3 equals 6 or STPRAN is true, the step function correction is modified by calling the random error function. If I3 equals 7 or EPH is true, the solar time correction is done by an ephemeris sun sensor. When I3 equals 8 or EPHRAN is true, the ephemeris sun sensor correction is adjusted using the random error function. The remaining I3 values pertain to

the solar sail. When I3 equals 9 or SAIL1 is true, a fixed sail is used with a constant force along lambda. If I3 equals 10 or SAIL3 is true, a fixed sail scheme 3 is used. I4 determines which XJ22 and XLAM22 are to be used. The program is set up to use I4 equal to 2 which sets XJ22 at -1.700E-6 and XLAM at -19 degrees. There are no logical variables that go with I4. I5 establishes the force cutdown. Initially I5 equals 1, no force cutdown. When I5 equals 2, there is a force cutdown equal to the input number CUT. To obtain I5 equals 2, it can be read in directly or CUTDOWN can be read in as true. I6 and I8 are the more sophisticated uses of the solar sail. I6 equal to 1 or SAIL2 equal to true fixes the sail and allows one to also make a sun correction. I6 is initialized as 0. I6 equal to 2 or TACSL is true the double precision function FLAM calls TACK and the tacked sail scheme is used. I8 flips the sail according to three schemes: I8 equal 1 or FLIPW equal to true fixes the sail through half the orbit and flips it off with the orbit; I8 equals 2 or FLIPAG true fixes the sail through half the orbit and flips if off against the orbit; I8 equals 3 or SAIL4 true flips the sail due to logic within the satellite. I7 determines if the orbit elements are averaged at the end of every ten days before they are punched. Initially I7 equals 2 and AVERAG is false, which loops the main program around the averaging sequence. When I7 equals 1 or AVERAG is true, the main program averages every ten days and punches the averaged figures.

The basic units for the parameters are kilometers, seconds, degrees. They are initialized as follows:

A	=	42165.5D0	radius of earth in kilometers
A1	=	0.9D0	solar absorptivity for solar sail
A2	=	0.1D0	solar absorptivity for solar sail
ALONGR	=	0.0	force along the radius to the satellite
CONST	=	-4.45D-8	kilonewtons/kilogram = 1.6 * (factor = 5.6)/365.25/(86.4 * 6) sign is negative in second and fourth coordinates

CUT	=	0.0	
DATE	=	0.0	
E	=	0.0	
FORCCT	=	0.0	force along theta
FORLAM	=	1.0	force along lambda
IRUN	=	0	a counter so one can keep track of the various computer runs
ITHDAY	=	10	frequency in days of print and/or punch execution
NUM	=	6	number of impulses given (used in FLAM)
OFFSET	=	-90.0D0	angle by which sail can be offset in the TACK scheme
OL	=	0.0D0	argument of perigee for first orbit, in degrees
TEND	=	1800.0D0	number of solar days program is to run for
XDISP	=	70.0D0	specific impulse of thruster in seconds
XI	=	0.0D0	inclination of the orbit in degrees
XLAMD	=	300.0D0	degrees of longitude measure east of Greenwich
XLIMD	=	297.0D0	minimum angle for thrusting, simple scheme
XLIMXD	=	350.0D0	maximum angle for thrusting, simple scheme
XMASS	=	132.0D0	mass of satellite in kilograms

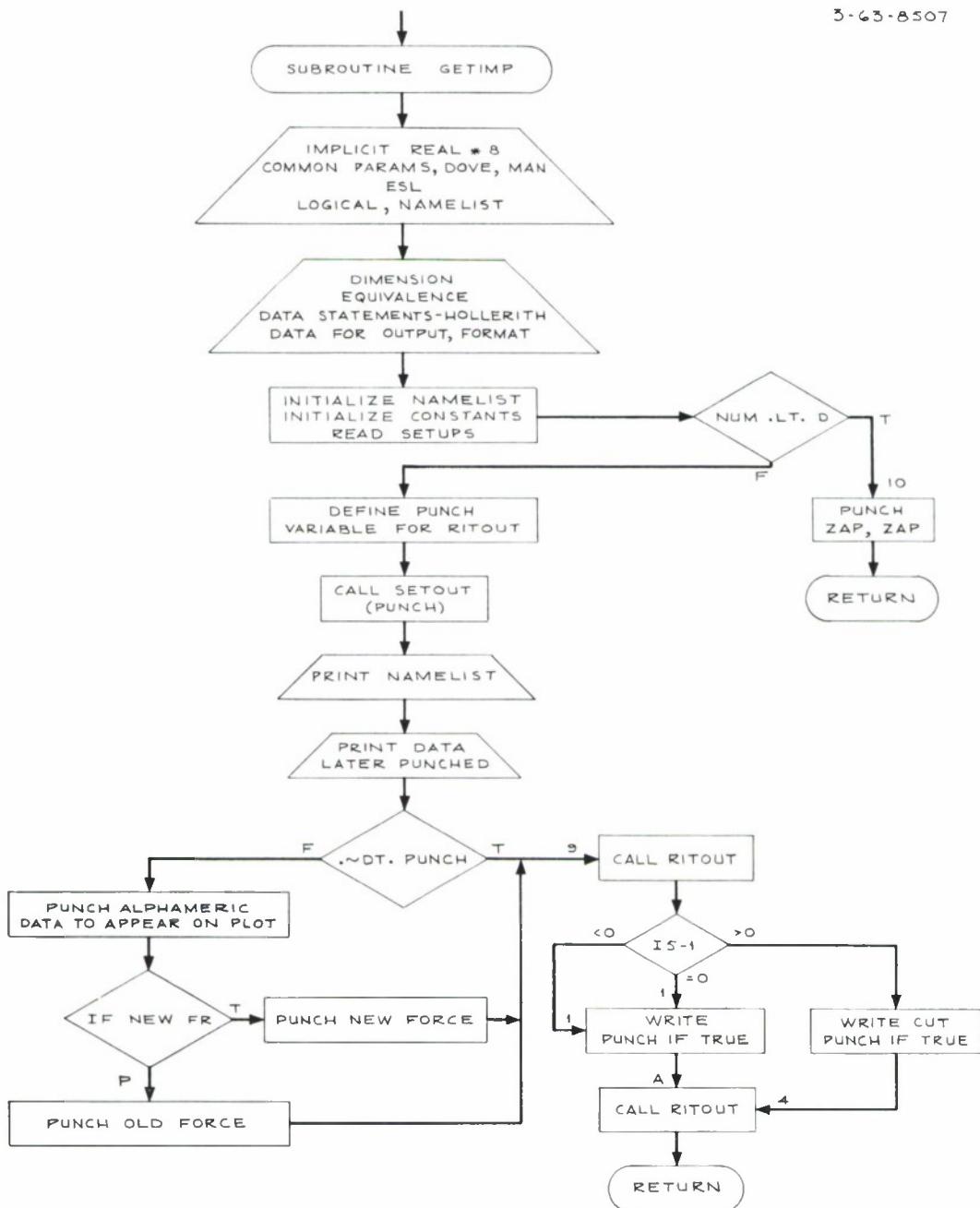
In addition there is a logical parameter, NEWFR, that has no numerical equivalent. NEWFR is initially FALSE. When NEWFR is true, the SUBROUTINE NEWFOR is called by FLAM and the magnitude of the impulse if modified accordingly.

To change any of the parameters or logical variables the NAMELIST/SETUPS/ is used & SETUPS is on the first card beginning of column 2. The variables desired changes are then punched separated by commas. To set a logical variable to TRUE just use T for example: NEWFR = T would be

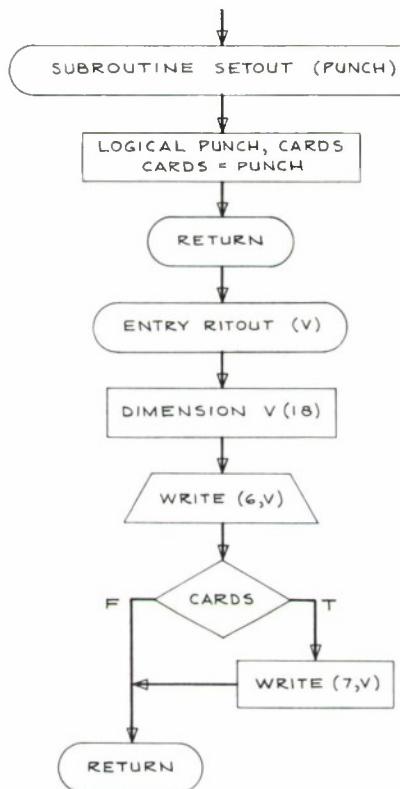
sufficient to set NEWFR true. Do not punch in column 73 or beyond. When using a second card begin in column 3. Do not follow the last variable with a comma; instead leave one or more spaces and punch &END. The set of cards with the SETUPS changes is then placed behind the //FT05F001 control card.

The parameters are printed and/or punched according to a variable FORMAT routine, RITOUT. RITOUT is an ENTRY point in SETOUT. The first call to SETOUT establishes whether or not to punch when RITOUT is called. RITOUT is called for each I variable. To add an I variable to the program, IY, where Y is an integer greater than 8, eight additions to GETINP are necessary. First a comment card to describe the IY variable and the X uses to which IY can be put. Secondly, it must be initialized. The remaining additions are to setup the call to RITOUT. There are X + 1 dimension statements as follows: Dimension VFIY(16, 6) and X statement like dimension XVFIYX(16). For each X there is also an EQUIVALENCE statement such as: EQUIVALENCE (XVFIXY(1), VFIX(1, Y). For each value IX can assume there must be a data statement in Hollorith form containing the alphanumeric data you desired printed and/or punched to describe the alternative used. You need to add IX to the 101 format statement, and one call to RITOUT, for example: Call RITOUT (VFIY(1, X)).

GETINP thus establishes the particular parameters and logical choices for one computer run, which takes approximately two hours on the IBM/360 with TEND = 1800.



3-63-8508



APPENDIX C

INTEGRATION FUNCTION AND SUBROUTINE

The double precision function CIND(K, A, B) and its related functions FINDV, DPNV, and SUBROUTINE SSL find the first and second derivatives and calculate new values for R, THETA, XLAM, DR, DTHETA, DXLAM, and T for each cycle through the main program. The call to SSL sets up the constants for SSL. FIND calls CIND with K = 0 which routes the control to the proper place to find a new T. For the other variables DPNV is used. In the computer GO TO statements only the first two solutions are used by this program. The sequence of four subprograms was written in another language outside Lincoln, so there are more choices than this program uses.

The integration formula used is the Adams four-point formula:

$$y_{n+1} = y_n + \frac{h}{24} (55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}) \quad (C-1)$$

where $h = x_{n+1} - x_n$ is the constant incremental change of the independent variable x , y being a function of x .

Starting formulas are needed to obtain the first three points. The following set of formulas is used to obtain y_1 , y_2 , and y_3 :

$$y_1^{(1)} = y_0 + hy_0' \quad (C-2)$$

$$y_1^{(2)} = y_0 + \frac{h}{2} (y_1^{(1)'} + y_0') \quad (C-3)$$

$$y_1^{(3)} = y_0 + \frac{h}{6} (2y_1^{(2)'} + y_1^{(1)'} + 3y_0') \quad (C-4)$$

$$y_2^{(1)} = y_1^{(3)} + \frac{h}{2} (3y_1^{(3)'} - y_0') \quad (C-5)$$

$$y_2^{(2)} = y_1^{(3)} + \frac{h}{12} (5y_2^{(1)'} + 8y_1^{(3)'} - y_0^{'}) \quad (C-6)$$

$$y_3^{(1)} = y_2^{(2)} + \frac{h}{12} (23y_2^{(2)'} - 16y_1^{(3)'} + 5y_0^{'}) \quad (C-7)$$

$$y_3^{(2)} = y_2^{(2)} + \frac{h}{24} (9y_3^{(1)'} + 19y_2^{(2)'} - 5y_1^{(3)'} + y_0^{'}) \quad (C-8)$$

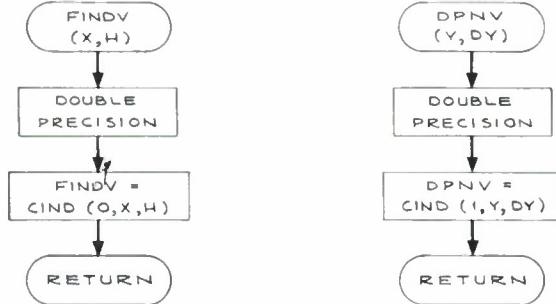
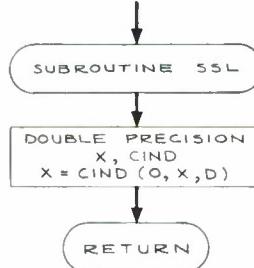
where the superscripts indicate the order of the approximation. The past values of the derivatives are stored in reserved blocks. For a more complete discussion of the starting equations, the reader is invited to contact the author. In particular, a substitution of equations (1) and (2) and similar expressions for their derivatives into equation (3) will show that equation (3) reduces to the first four terms of the Taylor series.

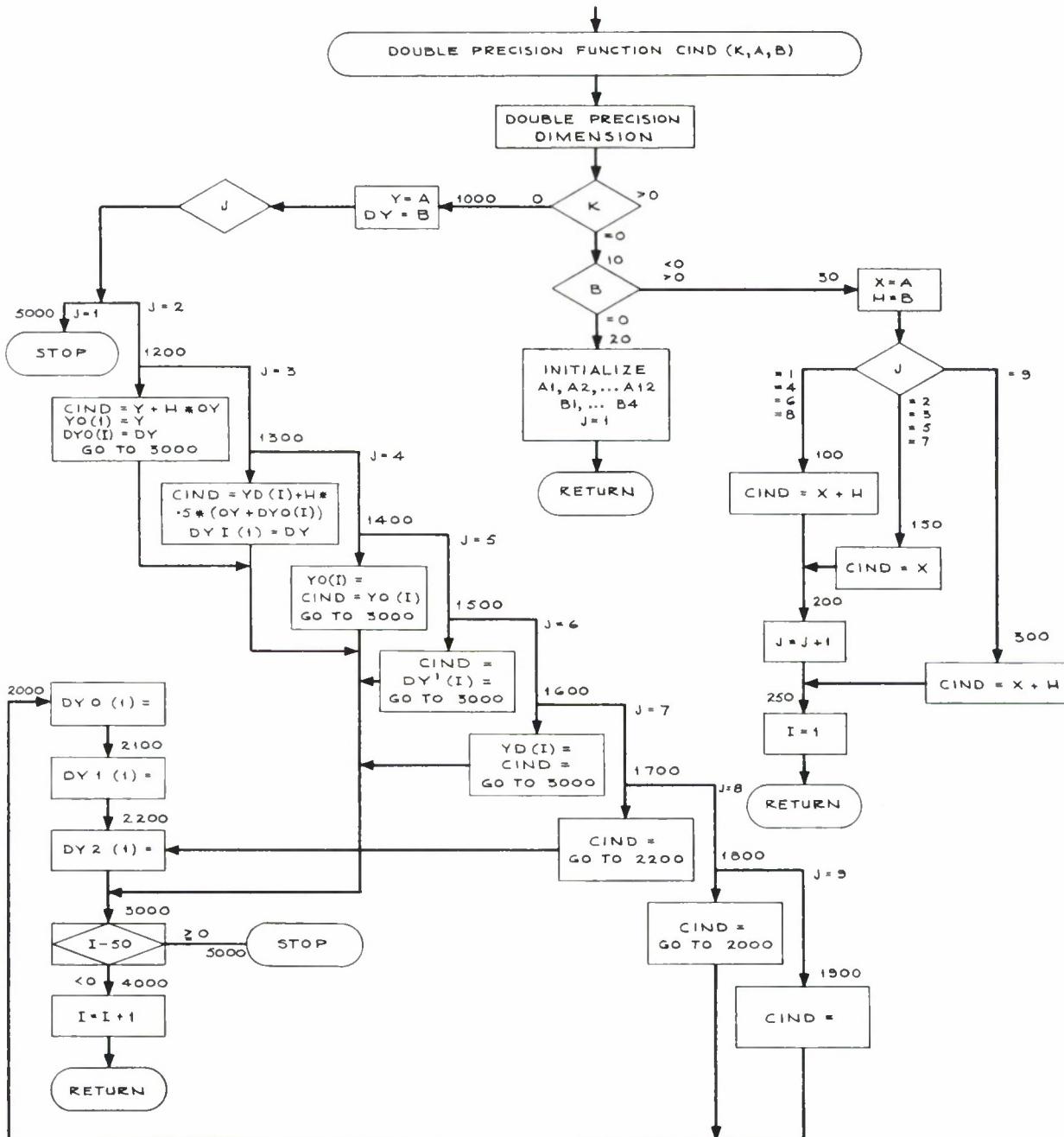
The truncation error of formula I is $\frac{251}{720} h^5 y^V(s)$, where $x_n < s < x_{n+1}$.

The accumulation of the increments $(\Delta y)_n = \frac{h}{24} (55y_n^{' } - 59y_{n-1}^{' } + 37y_{n-2}^{' } - 9y_{n-3}^{' })$ is performed in double precision to substantially reduce the round-off error.

Complete underflow-overflow testing is made. An underflow is corrected and a zero stored.

3-63-8505





APPENDIX D

DOUBLE PRECISION FUNCTION FLAM (XLAM)

The double precision function FLAM(XLAM) computes a force, FLAM, simulating the force of the thruster or solar sail depending on the logic with which the program is run. FLAM is returned to the main program. The force is applied NUM times an orbit, controlled by the gate K. When K is positive the gate is closed returning both force and FLAM as 0. When K is negative control passes to label 8. The various routes for the logic depend on the force scheme used for the particular run in question. The variable 15 controls the cutdown; when 15 is 1 there is no cutdown. When 15 is 2, the force is cut by the factor CUT, as long as LLL is also positive. LLL is the control variable whose sign is determined by the satellites position within a specified band, delineated by XLIM and XLIMX. LLL is negative when XLAM is inside the band and positive when XLAM is outside the band. 13, for values 1-8, controls the type of clock error returned by CORR(TEL). From this an XTEMP is calculated. XLAM is the actual position of the satellite; XTEMP is where the satellite's logic thinks it is. The old force calculations use LLL, whereas the new force calculations use the subroutine NEWFOR. As long as the counter M is less than NUM, control returns to the main program with FLAM = FORCE. Once an orbit the fuel consumption, the total number of impulses, JFK, CORR the corrected longitude in degrees are printed out. The fuel consumption is kept as a running sum. After the fuel report is printed, on the last loop during which there is thrusting, K is set positive and the other counters are initialized for the next orbit.

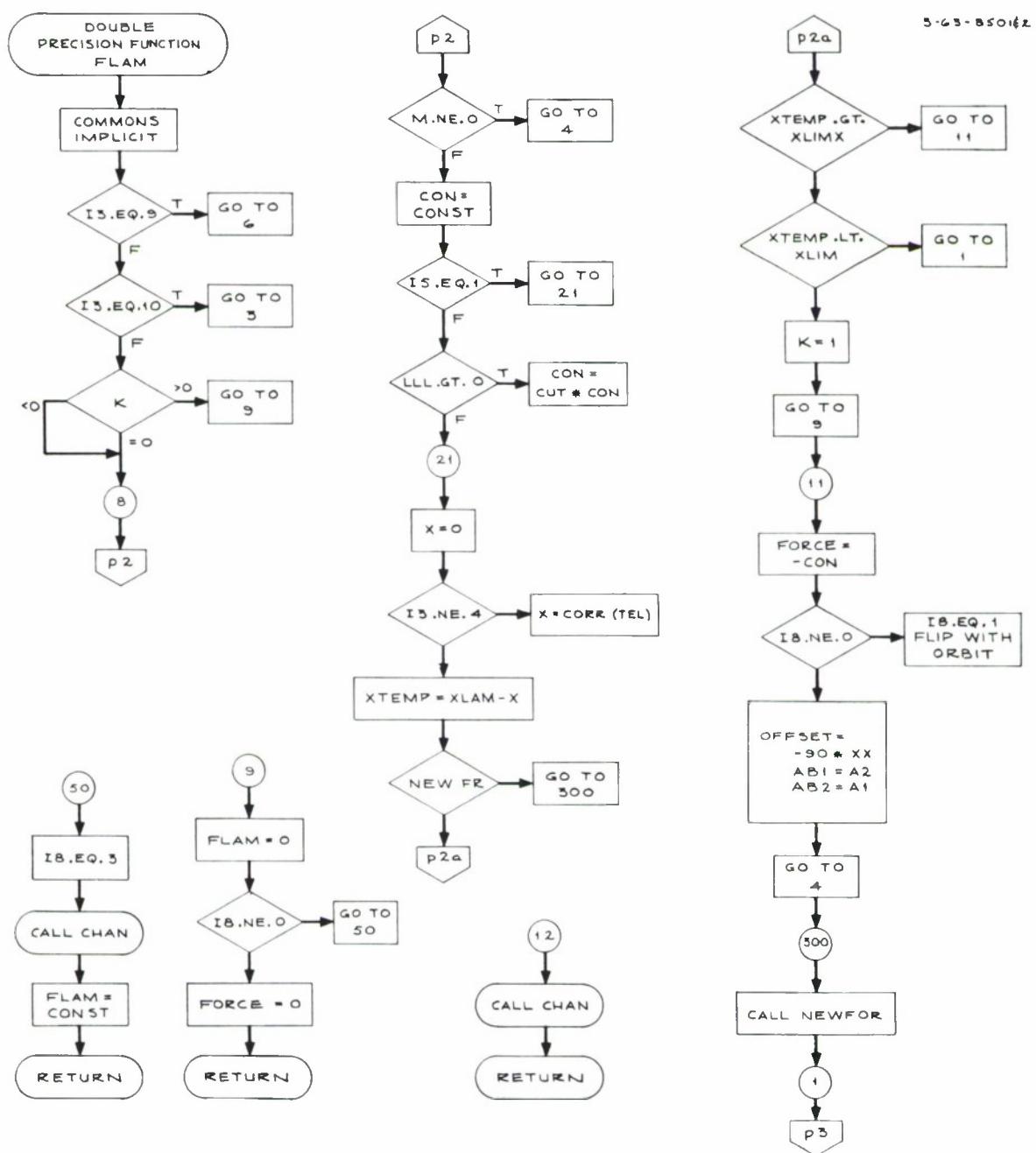
For the various schemes of solar sailing, there are deviations from the above possibilities. When K is positive and the sail is flipped off, FORCE = CONST and CHAN is called. When a simple sail is used FLAM = CONST and return. For this sail there is a constant force along λ . When a fixed sail is used, 13 = 10, FLAM = CONST and TACK or CHAN is called depending on 16.

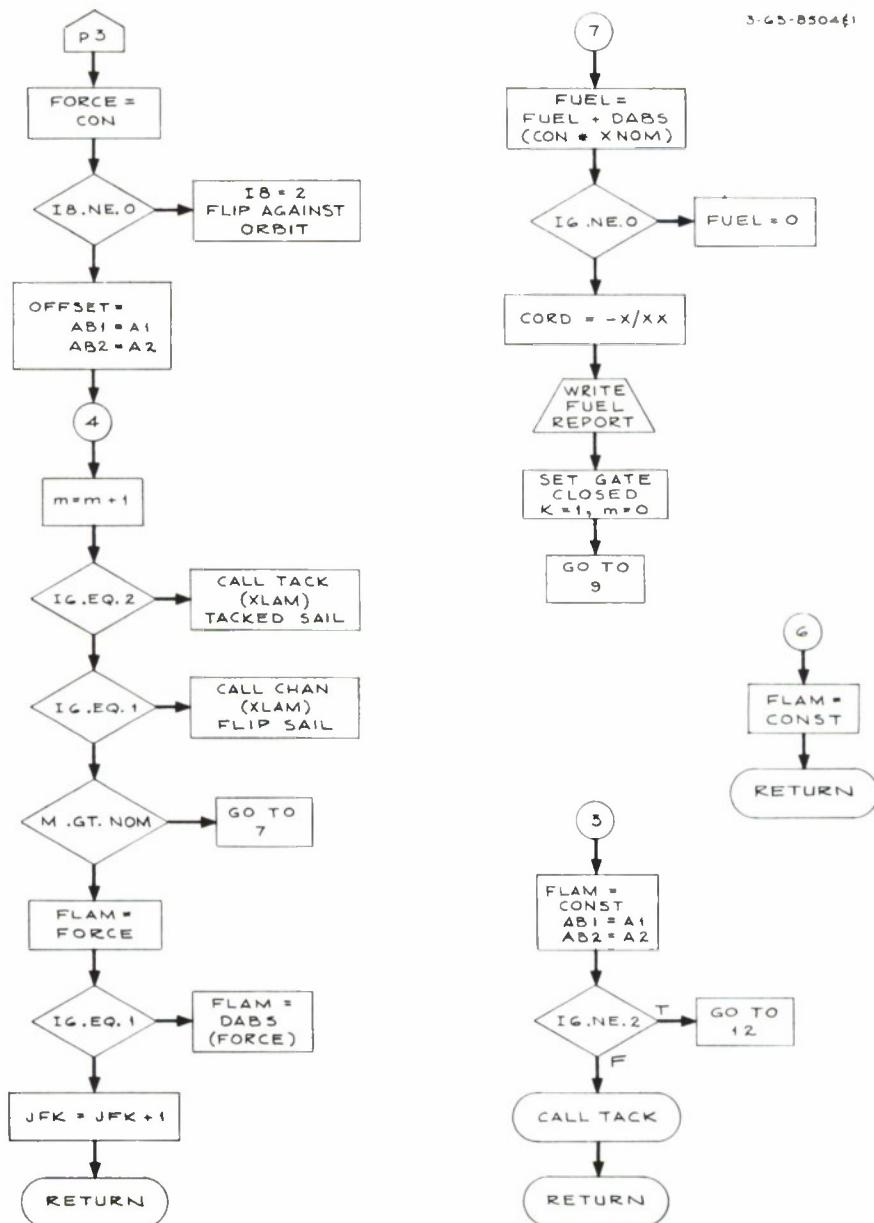
Tack calculates the forces in other directions as in addition to the force along λ the FLAM returns. Control returns to the Main program.

When K is negative (or zero) the force FLAM is calculated including the CUTDOWN or not. The first time through CON = CONST, XLAM is correlated by CORR(TEL). It is then determined if the modified XLAM lies within the band, if so FLAM = 0 and control is returned to the main program. If the true longitude is outside the desired stationkeeping range control passes to statement 1 or 11 within FLAM. If the satellite is too far east, FORCE = - CON, the dial can be flipped with the orbit if that scheme is used, or the sail can be offset -90° if that scheme is used. If the satellite is too far west, FORCE = CON, the sail can be flipped against the orbit, OFFSET can be 0° . In either case control passes to the same point 4, where K will return control for the rest of the orbit as long as the impulses are being given.

At statement 4 M dutifully counts the impulses. CHAN or TACK or neither is called according to 16. If M exceeds NUM, the fuel consumption report is issued. On all the sail schemes the fuel is set to zero. And the counters are reset for the next orbit as in the non-sailing schemes.

The double precision function FLAM(XLAM) returns to the main program NUM times an orbit the force along λ necessary to stationkeep within a specified range according to various force schemes using thrusters and solar sails.





APPENDIX E
SUBROUTINE NEWFOR

Subroutine NEWFOR is an alternative method of calculating the force necessary to keep a satellite in a synchronous orbit using the bilateral thrusting method proposed by A. Braga-Illa.

A. Theory

Figure 18 shows a sample of longitude vs. time, and the regions $\pm A$, $\pm B$, $\pm C$. The value of the force, con, at any point will be determined by which region we are in at that point and where we were before.

- I In region C and $-C$, $con = C_3$ and $-C_3$ resp.
- In region B and $-B$, $con = C_2$ and $-C_2$ resp.

For regions A and $-A$ there is a complicated scheme for determining the value:

- II If we get into region $+A$ via $+B$ or $-A$ via $-B$, then $con = 0$.
- If $+A$ via $-A$, $con = C_1$ or $-C_1$ resp.

III As soon as we get the sequence $(A-A+A)$ or $(-A+A-A)$ we start damping the force basically as follows. Assume we get the sequence $+A-A+A$ and then the curve never leaves the $+A$, $-A$ region.

Region:	A	B	$+A$	$-A$	$+\bar{A}$	$-A$	$+A$	$-A$	$+A$	$-A$	$+A$	$-A$
Force:	C_1	C_2	0	$-C_1$	$\frac{C_1}{2}$	$\frac{-C_1}{2}$	$\frac{-C_1}{4}$	$\frac{-C_1}{4}$	$\frac{C_1}{8}$	$\frac{-C_1}{8}$	$\frac{C_1}{16}$	$\frac{-C_1}{16}$
Region:	$+A$	$-A$	$+A$	$-A$	$+A$	$-A$						
Force:	$\frac{C_1}{32}$	$\frac{-C_1}{32}$	0	$\frac{-C_1}{32}$	0	$\frac{-C_1}{32}$						

where \bar{A} is the region where we start damping. After (*) we do not damp but keep $\pm C_1/32, 0$ as long as we stay in $+A-A$ and do not stray into B.

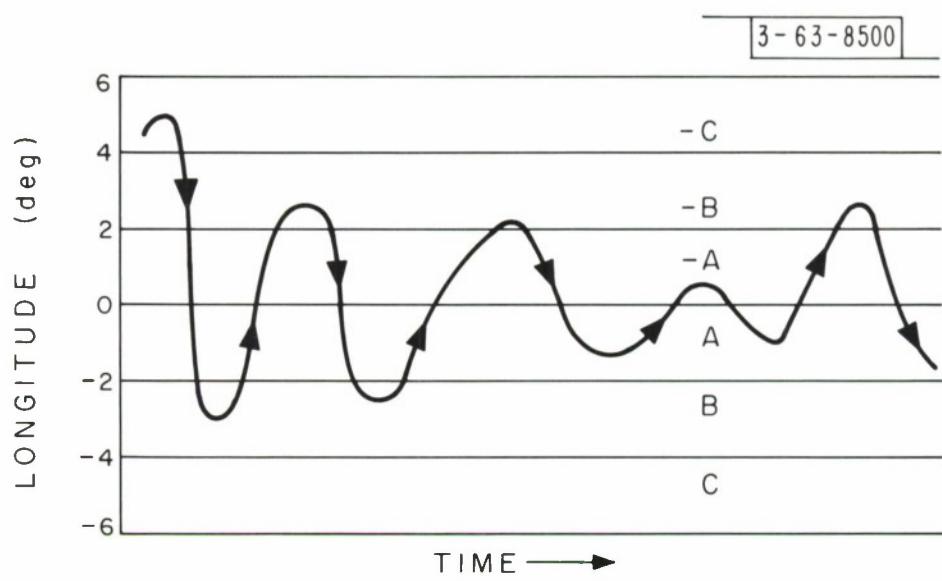


Fig. 18. Longitude vs time.

The most practical way to show the deviations from the basic scheme is via figure 19. Note the following:

- (1) When we get into region C (see arrow 1) we start over in the sense that no damping occurs until we get the $\pm A \mp A \pm A$ sequence again (see arrow 2).
- (2) When damping has started and we go into region $\pm B$, and then back into $\pm A$ we keep the damped value that we had before going into region $\pm B$, with the appropriate sign (see arrows 3, 4).
- (3) When we reach the maximum damping stage ($\pm C_1/32, 0$) and then go into region $\pm B$, upon re-entry into $\pm A$ the force = 0 by part II and if we then go into $\mp A$ from there, force = $C_1/32$ with the appropriate sign (see arrows 5, 6).

B. Data Cards

To specify that subroutine NEWFOR is to be used to calculate the force we must specify four quantities on the data cards:

NEWFR = T, $C_1 =$, $C_2 =$, $C_3 =$

where C_2 , C_3 are the values assigned to the force in regions B and C respectively and C_1 is the non-zero undamped value of the force in region A.

3-63-8499

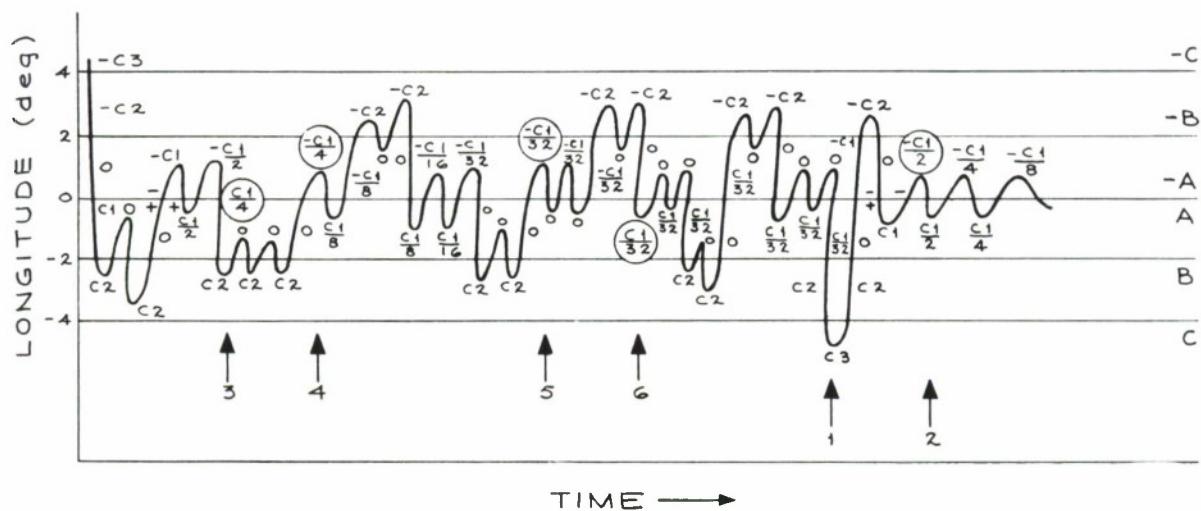
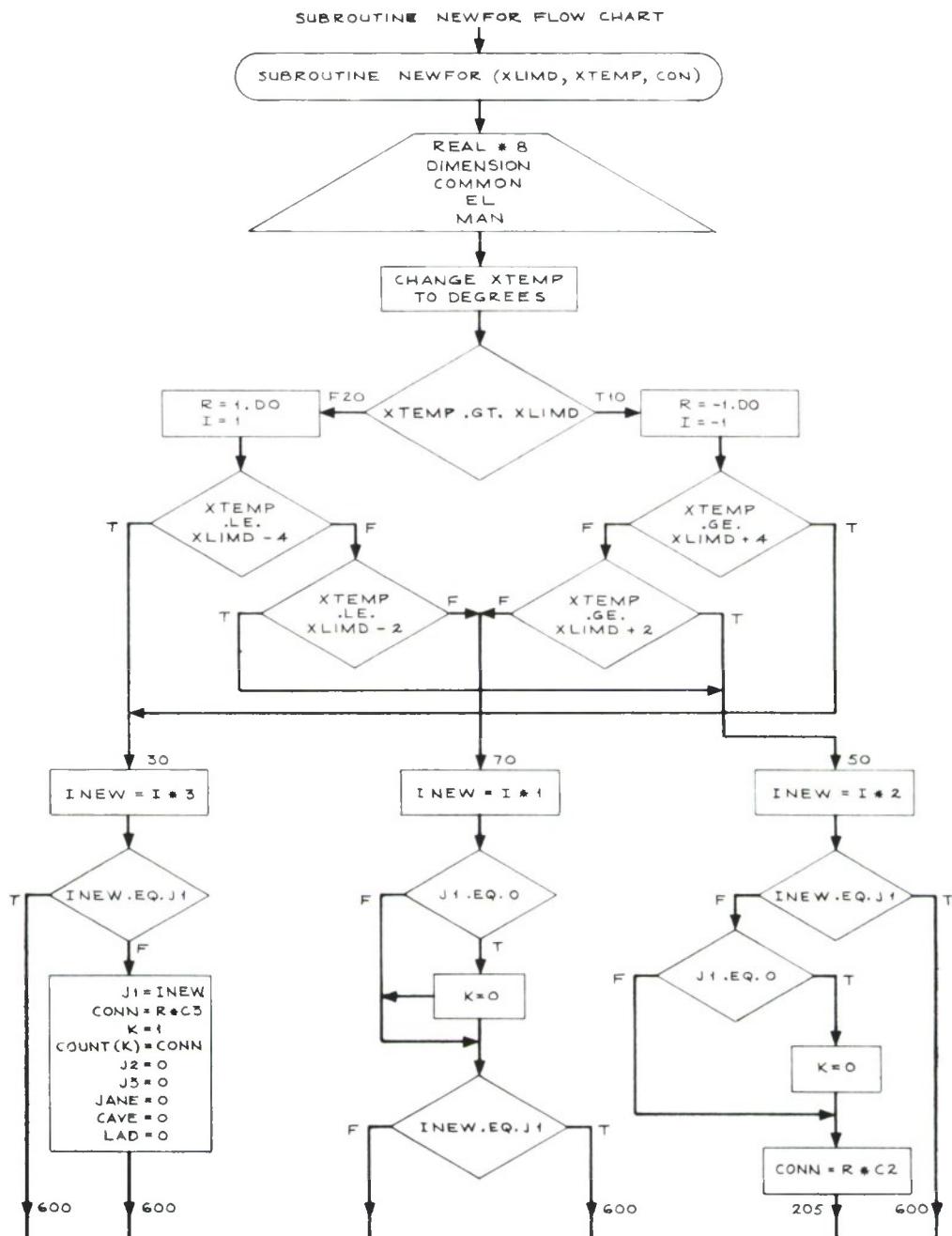
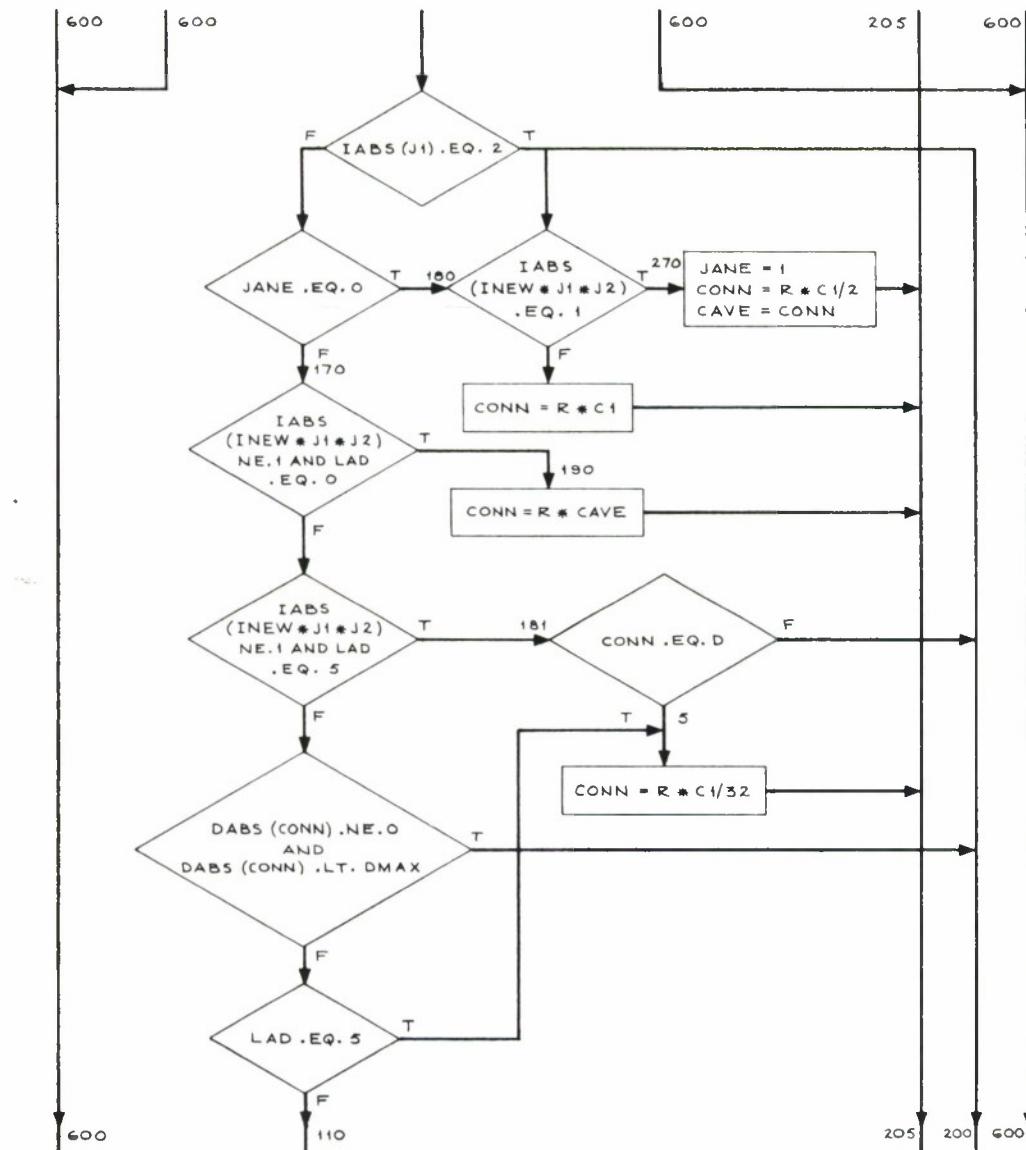
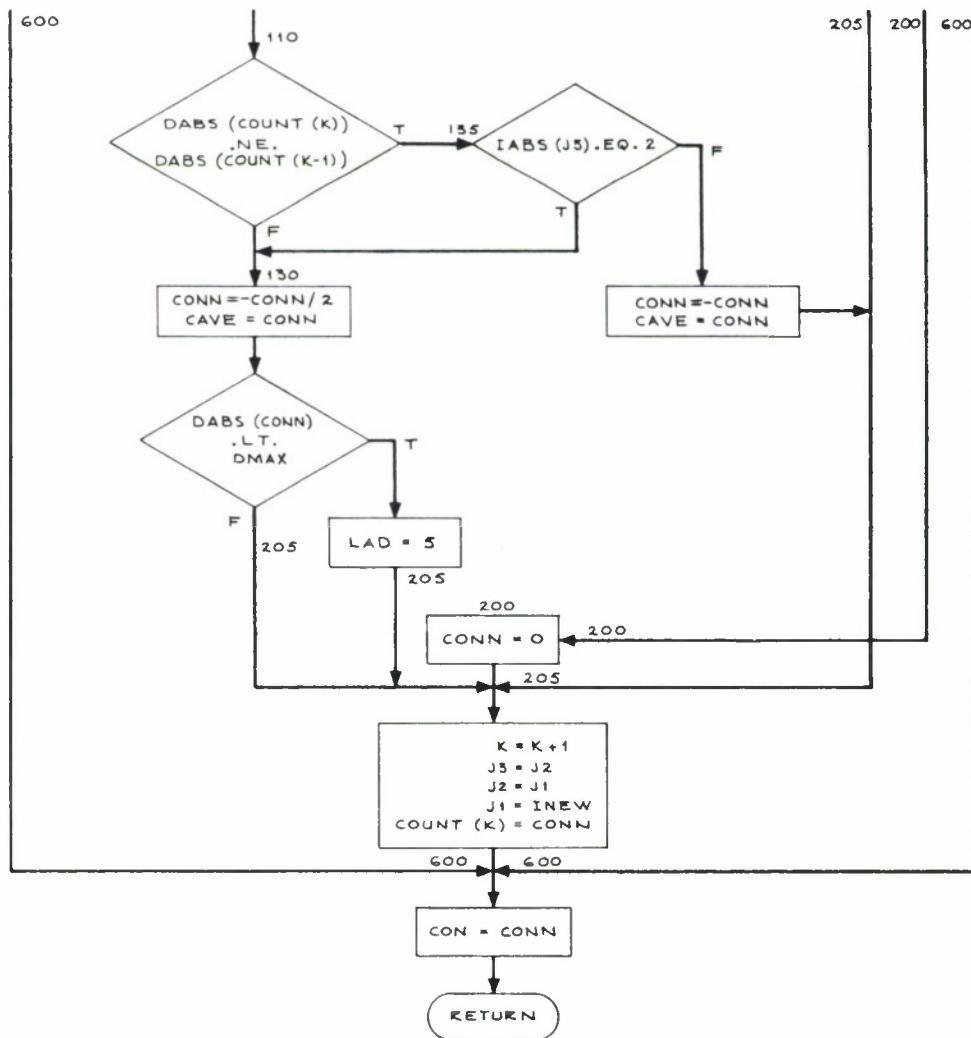


Fig. 19. Example of possible use of force scheme of A. Braga-Illa.







APPENDIX F

DOUBLE PRECISION FUNCTION CORR (TEL) returns to FLAM an angle by which the longitude XLAM is thereby modified. In FLAM: X = CORR(TEL), XTEMP = XLAM - X. The program calculates the apparent right ascension of the sun and the time of ephemeris transit, good for the year 1967, and accurate to about four seconds of arc of the right ascension and about 1.5 seconds on the ephemeris transit. There are eight options in this program determined by the I3 variable. Two of the options call the SUBROUTINE CURT, and two of the options call the SUBROUTINE ALVISE. When I3 = 4, the logic in FLAM bypasses the call to CORR.

If I3 is 7 or 8, immediately after converting TEL from seconds of days, the program branches to 110. Day is modulated by 365, and CURT is called to determine VALUE. If I3 = 8, VALUE is changed to radians and a random error is added on. If I3 = 7, CORR is simply VALUE in radians and control is returned to FLAM. CURT picks up an ephemeris sun sensor value from the step function tables stored in the BLOCK DATA/SPETS/. These tables simulate the logic in an electronic sun sensor within the satellite.

For the other options, the program calculates the mean anomaly, sun's coordinates, referenced to a geocentric coordinate system, right ascension of the sun, right ascension of the mean sun, and DIFF the difference between the right ascension of the sun and the right ascension of the mean sun added to 6.28 when $|DIFF|$ is less than 3. Now the program branches according to I3 to calculate VALUE. In each case after VALUE is calculated, control goes to statement 31 where CORR = DIFF - VALUE.

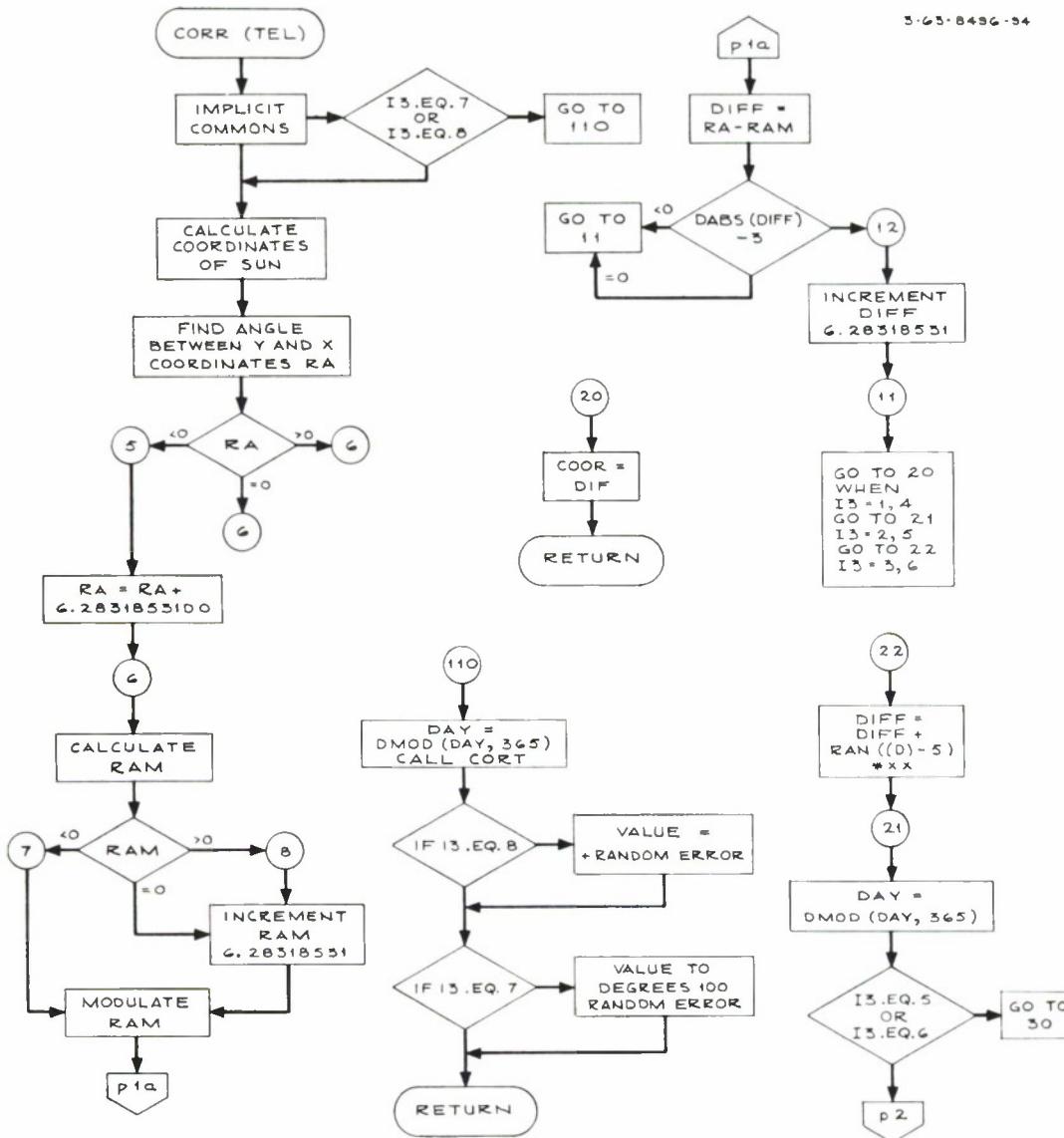
If I3 is 5 or 6, VALUE is returned from a call to SUBROUTINE ALVISE. When I3 = 6 DIFF has a random error added in. ALVISE as a subroutine operates identical to CURT, looking up a value in a BLOCK DATA. When control is returned to CORR, CORR branches to 31.

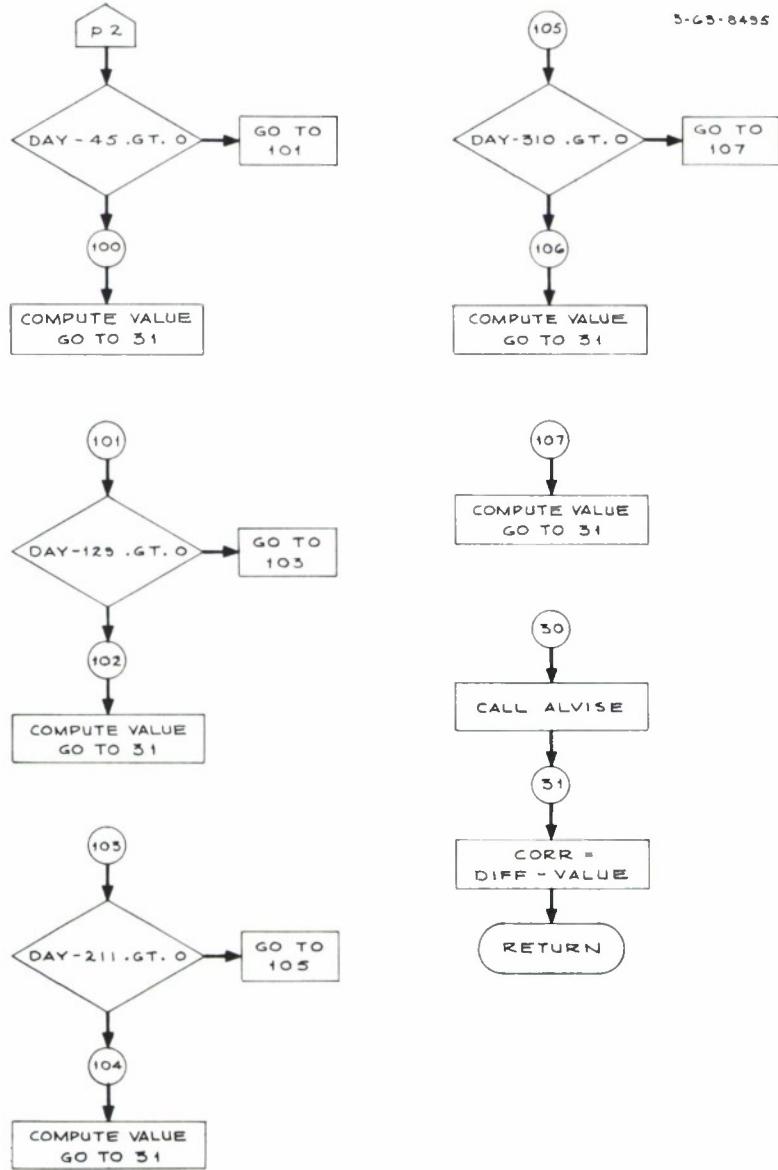
If I3 = 1, CORR = DIFF and control is returned to FLAM. When I3 = 2, DAY is modulated by 365. The formula for VALUE is determined by which part of the year DAY falls.

<u>Day</u>	<u>Value</u>
0 - 45	100 VALUE = .08D0 * (DAY + 2.25D0) * XX
46 - 129	102 VALUE = -.0584D0 * (DAY - 111. D0) * XX
130 - 211	104 VALUE = .033D0 * (DAY - 162. D0) * XX
212 - 310	106 VALUE = -0.0584D0 * (DAY - 240. D0) * XX
310 - 360	107 VALUE = .08D0 * (DAY - 363. D0) * XX

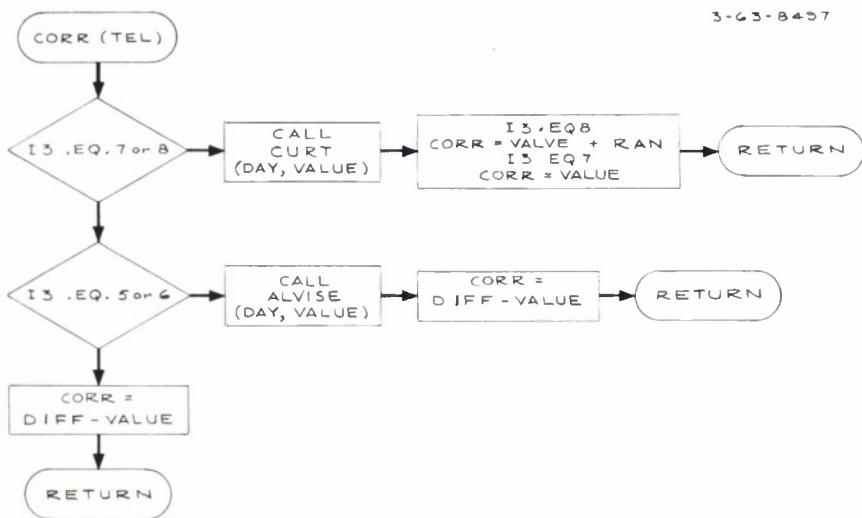
After each calculation of VALUE control branches to 31. When I3 is 3, DIFF is modified using the random error function, then logic follows the same for I3 = 2. There is a loop for I3 = 4, although CORR is not called when I3 = 4, to avoid an error message from the compiler.

FUNCTION CORR FLOW CHART





3-63-8457



APPENDIX G

The SUBROUTINE TACK (XLAM) and its ENTRY CHAN (XLAM) return to MAIN via FLAM three forces, FORCCT, FORLAM, and ALONGR, which are used to calculate D2THET, D2XLAM, and D2R respectively. The ENTRY CHAN(XLAM) fixes the sail along the radius to the satellite from the earth. According to the I8 variables, the sail remains upright on the radius or is flipped throughout part of the orbit. FORCCT is the solar force exerted by the sail in the direction of theta. FORLAM is the solar force exerted by the sail in the λ direction. ALONGR is the solar radiation force exerted along the radius. L. Black has pointed out that the most efficient sail would be one of high thermal emissivity on both sides with a high solar absorptivity on one side and a high solar reflectivity on the other. A sail made out of aluminized microsheet stuck on an aluminum sheet painted black on the back side would be a good approximation to this. The average force along the orbit for a sail whose plane is coincident with the orbit radius vector and the north-south direction is given approximately by

$$F = \frac{AS}{2C}$$

where

A = is the area of the sail

S = solar constant

C = velocity of light

The sail placed on angle bisector of angle formed by sun and orbital tangent in direction of the motion is the most efficient sail; the program allows also an offset from this position, OFFSET.

When the solar rays hit the sail they generate two forces which must be resolved into the three components needed. The two forces are one a force perpendicular to the sail itself and opposite to the side of the sail hit by the solar radiation; the second is a force along the sail due to its various

absorptivity properties. Both sides of the sail are reflective when the sail is "tacked" and when it is flipped. When ENTRY CHAN(XLAM) is used, the sail is fixed on the radius from the earth to the satellite, and one side is highly reflective while the other side is highly absorptive. The three components of the generated forces are the force along the orbit, FORLAM; the force along the radius vector of the satellite, ALONGR, and the force along λ , FORLAM.

In this subroutine all the vectors are unit vectors and lie in the orbital plane of the satellite. The coordinate system is geocentric, with the X-axis lying along Aries. XNORMA, YNORMA are the coordinates of the normal to the sail itself. The coordinates of the sun are calculated as in Moulton pp 171, and then the vector is projected into the orbital plane with coordinates XPROJ, YPROJ. XFORCE, YFORCE are the coordinates of the orbital tangent. XSAIL, YSAIL are the coordinates of the normal to the sail. XRESAL, YRESAL are the coordinates of the sail itself. The angle between the sail and the radius is ANCHOR. DOTPRO is the cosine of the angle between the normal to the sail and the projection of the sun. The absolute value of the cosines is used to simplify the logic in determining quadrants. ROCK is the absolute value of the cosine of the angle between the sail and the radius. TRAP is the absolute value of the cosine of the angle between the sun and the normal to the sail. When DOTPRO is less than zero the sail is receiving the solar force on its absorptive side. POT is the force exerted perpendicular to the sail due to the solar radiation pressure. $POT = (2 - A) * DOTPRO ** 2$. ALOSAL is the force along the sail due to a non-perfect sail.

$$ALOSAL = -A * TRAP * (XSUN * SRESAL + YSUN * YRESAL)$$

There is a logical sequence to test whether the sail is "ON" with the flipped sail logic. If it is on one continues, if not the program branches to 40, where a sequence of equations set up the small forces when the sail is turned "OFF". Four variables are used to set up the signs of the forces according to the quadrant the sail, sun, and orbital tangent are in. They are COD, FISH, HERRING, and SALT. They are initially +1 and are made -1 when necessary. The solar

force is exerted opposite to the normal to the sail when the angle between the sun's projection and the sail is between zero and 180; the solar force is exerted along the normal to the sail when the angle between the sun's projection and the sail is between 180° and 360° . The coordinates of the solar force are XSORAF, YSORAF. HERRING is negative when the angle between the solar force and the orbital tangent is greater than 180. SALT is negative when the angle between the sail and the orbital tangent is greater than 180. COD is negative when the angle between the solar force and the radius is greater than 180. FISH is negative when the angle between the radius and the sail is greater than 180. The three components of the solar radiation force acting on the satellite are:

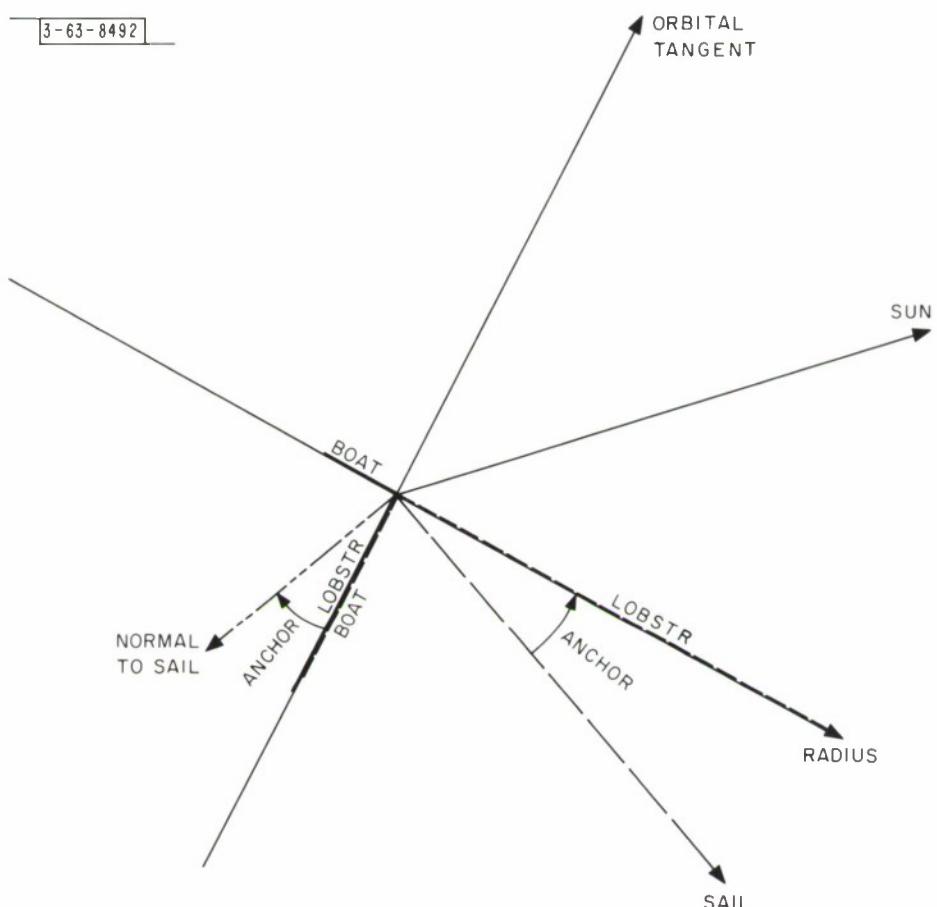
$$\left. \begin{array}{l} \text{FORLAM} = \text{HERRING} * \text{POT} * \text{ROCK} + \text{SALT} * \text{ALOSAL} * \text{BOAT} \\ \text{ALONGR} = \text{COD} * \text{POT} * \text{BOAT} + \text{FISH} * \text{ALOSAL} * \text{ROCK} \\ \text{FORCCT} = -\text{AB1} * \text{TRAP} * \text{ZSUN} \end{array} \right\} \begin{array}{l} \text{See} \\ \text{Fig-} \\ \text{ure} \\ 20 \end{array}$$

When the sail is flipped to its "OFF" position there are slight forces acting upon the satellite. DOTPR1 is the cosine of the angle between the sun (not its projection) and the sail. DOTPR2 is the absolute value of ZSUN. FORCCT is negative when ZSUN is positive. Otherwise the three forces are:

$$\begin{array}{l} \text{FORLAM} = \text{AB1} * \text{DOTPR2} * \text{DOTPRO} \\ \text{ALONGR} = -\text{AB1} * \text{DOTPR2} * \text{DOTPRI} \\ \text{FORCCT} = (2 - \text{AB1}) * \text{DOTPR2} \end{array}$$

AB1 is the absorptivity received by the program as A1 and A2 and set equal to the proper one depending on the quadrant one is in. Following is a diagram of the forces for an offset sail in one quadrant.

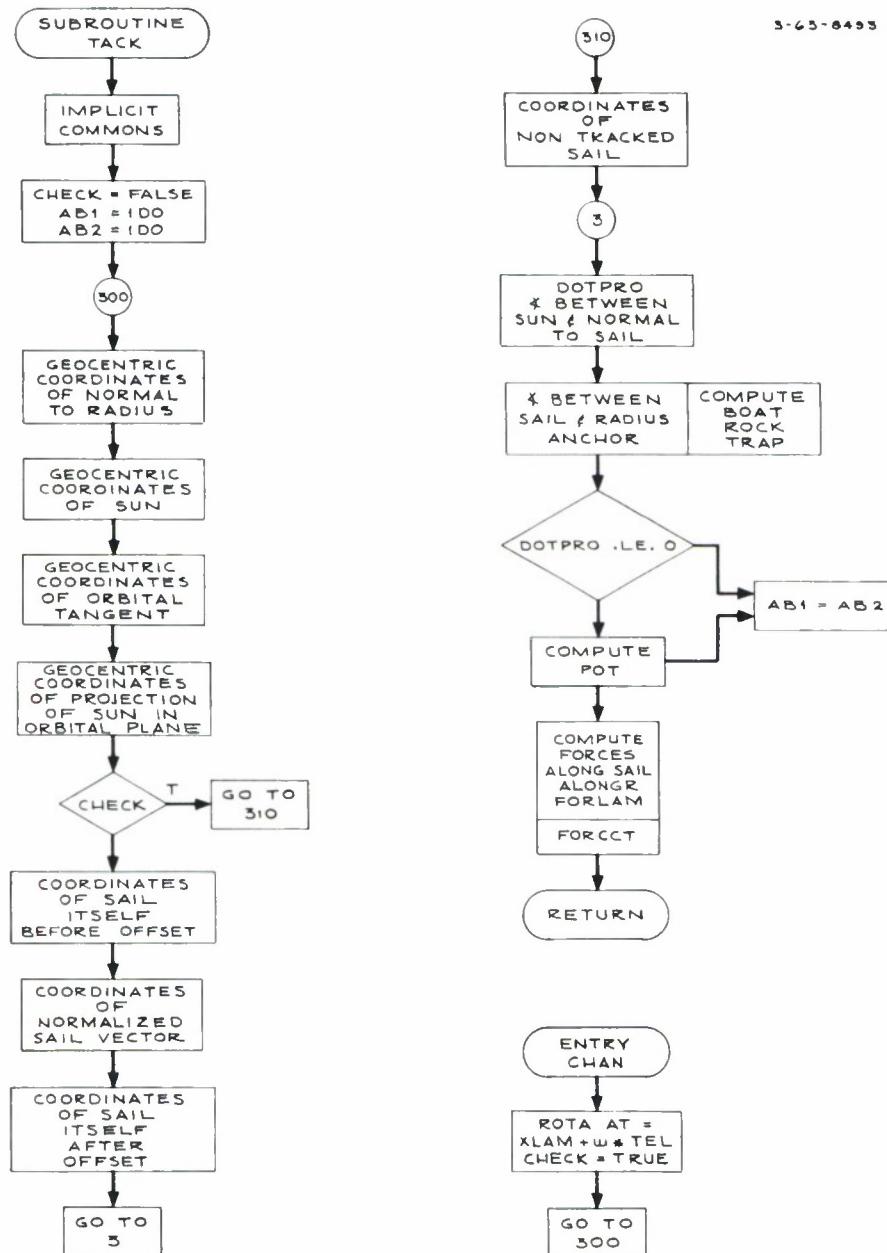
3-63-8492



FORCE EXERTED BY SUN ALONG NORMAL TO SAIL

Fig. 20. VECTOR DIAGRAM OF FORCES acting on solar sail.

SUBROUTINE TACK FLOW CHART

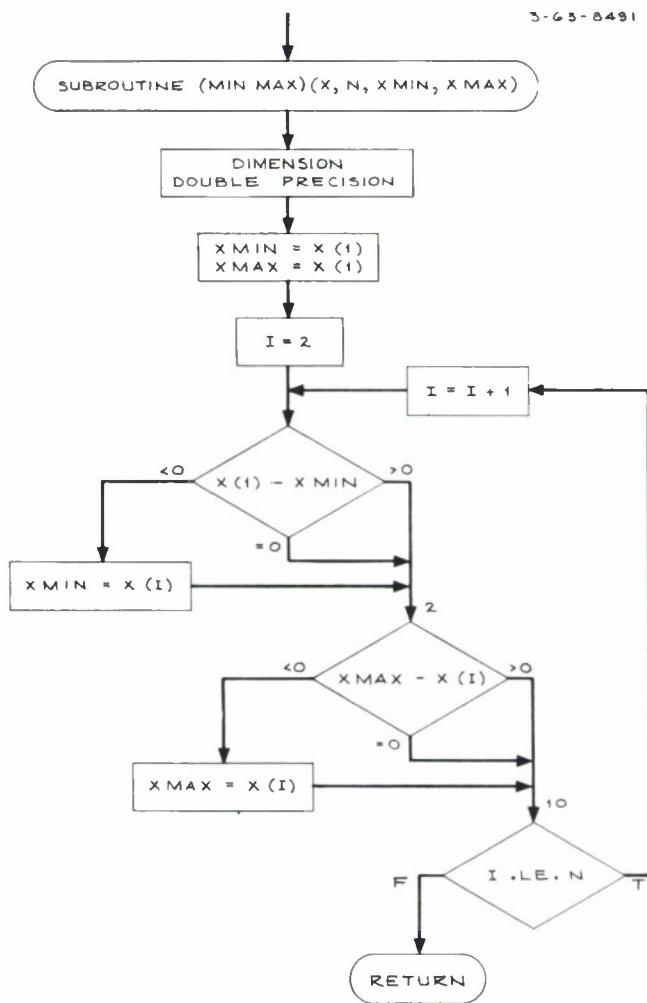


APPENDIX H

SUBROUTINE MINMAX (X, N, XMIN, XMAX)

Subroutine minmax is designed to find the maximum and minimum value of a set of N numbers.

3-63-8491



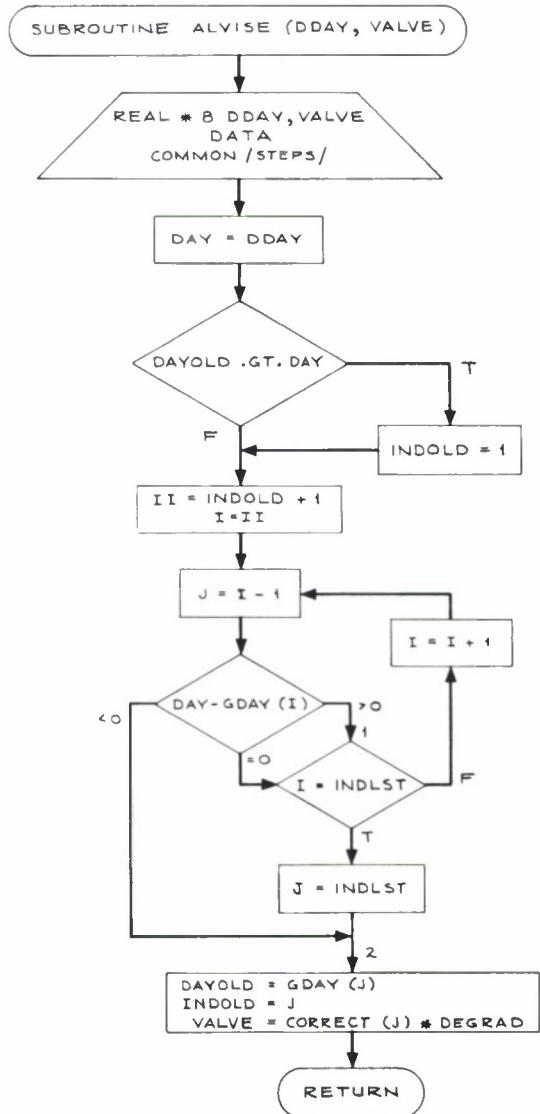
APPENDIX I

SUBROUTINES ALVISE AND CURT

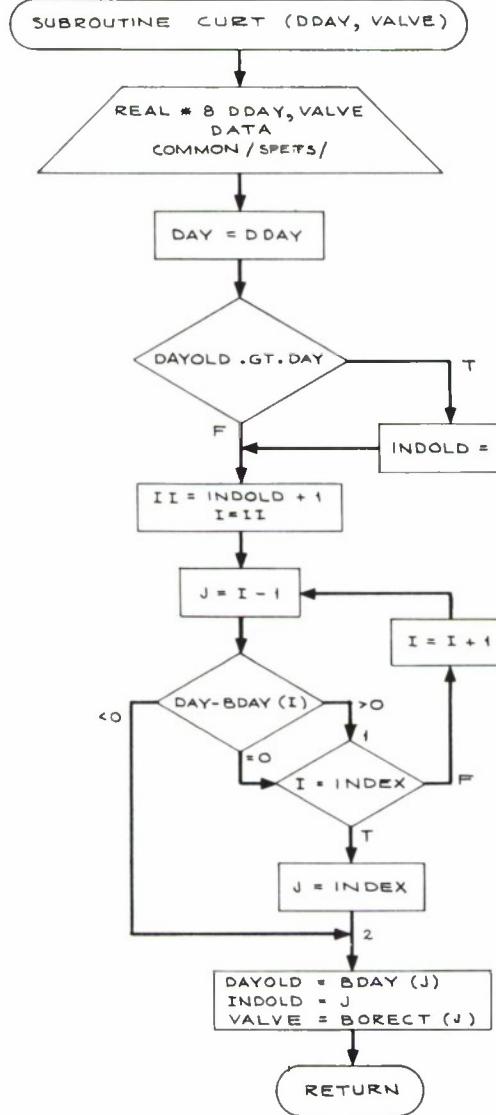
SUBROUTINE ALVISE contains the information for the step function approximation to the equation of time used for the LES-6 satellite.

SUBROUTINE CURT contains the information for the step function approximation to the equation of time used for the LES-6 satellite analemma sensor.

5-63-8489



3-43-8490



APPENDIX J

LISTING OF THE PROGRAM

LEVEL 2 FEB 67

OS/360 FORTRAN H DATE 68.067/17.15.43

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NJLIST,VJOECK,LDA),MAP,NOEDIT,1D

C THIS VERSION OF Z800 HAS NO REAL 16 OPTION IN IT.

C XLM measured east of Greenwich

C IF I1 = 1, 2ND ORDER HARMONICS USED
C IF I1 = 2, 3RD ORDER HARMONICS USED
C IF I2 = 1, PUNCH OUTPUT
C IF I2 = 2, NO PUNCH OUTPUT
C I3 CONTROLS SUN CORRECTION I3 = 1 NO CORRECTION
C I3 = 2 CORRECTED LINEARLY
C I3 = 3 CORRECTED LINEARLY W/ RANDO
C I3 = 4 CORRECTED PERFECTLY
C I3 = 5 CORRECT BY STEP FUNCTION
C I3 = 6 CORRECTED BY S-F W/ RANDOM
C I3 = 7 CORRECT BY EPHemeris SJN SENS
C I3 = 8 CORRECT BY EPHemeris SS W/R
C I3 = 9 1ST SUN SAIL CORRECTION
C I3 = 10 CONSTANT FORCE ALONG LAMBDA
C I3 = 11 SOLAR SAIL3 FIXED SAIL

C IF I4 = 1 XJ22=-1.816E-6 AND XLAM22 = -15.0*.01745
C IF I4 = 2 XJ22=-1.700E-6 AND XLAM22 = -19.0*.01745
C IF I5 = 1, NO FORCE CUTDOWN
C IF I5 = 2, FORCE CUT BY INPUT NO. CUT

C I6 = 1 SOLAR SAIL2

C IF I7 = 1, PLOTTING AVERAGE OF XLM FOR 10TH DAY
C IF I7 = 2, PLOTTING XLM AT END OF EVERY 10TH DAY
C IF I8 = 3, SAIL IS FLIPPED BY LOGIC IN SATELLITE

C IF NEWFR=TRUE, USE NEW FORCE CALCULATIONS
C IF NEWFR=FALSE, USE OLD FORCE CALCULATIONS

C CONST IS CALCULATED IN THE FOLLOWING MANNER--CONST=F/W EXPRESSED IN
C KILONEUTONS PER KILOGRAM. FOR OUR CASE
C = $1.6 \times (\text{FACTOR}=5.6) / 365.25 / (86.4 * 6) = 4.45F-8$

C THE SIGN SHOULD BE NEGATIVE IN THE SFCDNO AND FOURTH COORDINATES

C Z800 IS A COMBINATION OF VARIOUS ZEPOT 6 AND ZEPOT PROGRAMS
C IT USES EITHER THE Z6 OR Z7 COMPUTATION CONTROLLED BY I1
C TIME, A, + XL PUNCHED OUT EVERY 10TH DAY UNDER CONTROL JF I2

C D A N G F ? D A N G E R

C NEVER PUT COMMON /MAN/ IN MAIN PROGRAM

C WRITTEN BY CHAN CROCKER MIT LINCOLN LABORATORY
C ROOM C-266 Lexington Mass. TEL. 617-962-5500 Ext. 5307

C COMMENT CARD FPSIL = -6.* J22

C


```

ISN 0014          ESUN = 0.01672300      0103
ISN 0015          WS = 282.3728700 * XX   0104
ISN 0016          SIN1= 05IN (23.445800 * XX) 0105
ISN 0017          COS1 =0CDS(23.4432800 * XX) 0106
ISN 0018          XDT = 0T                0107
ISN 0019          NDUT = 6.283185300 / (W * DT) 0108
ISN 0020          7 CALL GETINP          0109
ISN 0021          IF (NUMIN .LT. 0) GD TO 1D 0110
ISN 0023          C
ISN 0024          J1=1
ISN 0025          J2=-1
ISN 0026          J3=1
ISN 0027          KAY = 0
ISN 0028          JANE=1
ISN 0029          LAD=5
ISN 0030          CAVE=CONST/32          0119
ISN 0031          C MOVE THE ELEMENTS FROM THE NAMEO COMMON TO BLANK COMMON
ISN 0032          NUM = NUMIN
ISN 0033          CDNST = CONST1
ISN 0034          I3 = 13T
ISN 0035          I5 = 15T
ISN 0036          CUT = CUTBY
ISN 0037          TWTYDY = FITHDY*DPRINT          0120
ISN 0038          C
ISN 0039          WRITE (6, 9)
ISN 0040          9 FORMAT(126H DT )          0121
ISN 0041          2          LAM
ISN 0042          R
ISN 0043          DLAM
ISN 0044          DR
ISN 0045          E*
ISN 0046          T
ISN 0047          LONG
ISN 0048          C
ISN 0049          C
ISN 0050          C
ISN 0051          C
ISN 0052          C
ISN 0053          C
ISN 0054          C
ISN 0055          C
ISN 0056          C
ISN 0057          C
ISN 0058          C
ISN 0059          C
ISN 0060          C
ISN 0061          C
ISN 0062          C
ISN 0063          C
ISN 0064          C
ISN 0065          C
ISN 0066          C
ISN 0067          C
ISN 0068          C
ISN 0069          C
ISN 0070          C
ISN 0071          C
ISN 0072          C
ISN 0073          C
ISN 0074          C
ISN 0075          C
ISN 0076          C
ISN 0077          C
ISN 0078          C
ISN 0079          C
ISN 0080          C
ISN 0081          C
ISN 0082          C
ISN 0083          C
ISN 0084          C
ISN 0085          C
ISN 0086          C
ISN 0087          C
ISN 0088          C
ISN 0089          C
ISN 0090          C
ISN 0091          C
ISN 0092          C
ISN 0093          C
ISN 0094          C
ISN 0095          C
ISN 0096          C
ISN 0097          C
ISN 0098          C
ISN 0099          C
ISN 0100          C
ISN 0101          C
ISN 0102          C
ISN 0103          C
ISN 0104          C
ISN 0105          C
ISN 0106          C
ISN 0107          C
ISN 0108          C
ISN 0109          C
ISN 0110          C
ISN 0111          C
ISN 0112          C
ISN 0113          C
ISN 0114          C
ISN 0115          C

```

```

0153
ISN 0053 C3 = GM * EPSIL * RE * RE
0154 C1 = (C3/EPSIL) * XJ2 * 3.00/2.00
ISN 0054 C2 = C3 * 3.00/2.00
0155 C4 = C3 / 2.00
ISN 0056 C M=0
ISN 0057 K=-1
ISN 0058 JFK=0
ISN 0059 N = 0
ISN 0060 XNUM1 = NUM
ISN 0061 XNUM = XNUM * 1000.00 * XMASS * XOT / (XISP * G)
ISN 0062 XNUM = XNUM * 1000.00 * XMASS * XOT / (XISP * G)
ISN 0063 FUEL = 0.00
ISN 0064 XLAM22 = -15.400 *XX
ISN 0065 XLIM = XLIM0*XX
ISN 0066 XLIMX = XLIMX0*XX
ISN 0067 T = DATE * 86400.00
ISN 0068 TPRINT= T + 86400.00
ISN 0069 TOUT = TWTOY + T
ISN 0070 V = (90.00-OL)*XX
ISN 0071 P = A * (1.00 - E*E)
ISN 0072 R = P / (1.00 + E * OCOS(V))
ISN 0073 H = NSQRT(GM*p)
ISN 0074 OR = H * E * OSIN(V)
ISN 0075 THETA = 90.00 - XI
ISN 0076 THETA = THETA*XX
ISN 0077 TENO = TENO * 86400.00
ISN 0078 XLOEY = XLAMO * XX
ISN 0079 XLAM = XLAMO * XX
ISN 0080 OFFSET = OFFSET * XX
ISN 0081 OTHETA= 0.00
C
C CALCULATE OXLAM
ISN 0082 O2R = H*H*(1.00-R/P)/R**3
C BETTER CALCULATION OF OXLAM D2R IS NOT 0
ISN 0083 GO TO (2, 52),11
ISN 0084 2 SQ = GM/(R**3 * (OSIN(THETA))**2)
1 -3.00*GM*XJ2*RF*RF*
2 (3.00*(OCOS(THETA))**2-1.00)/(2.00*R**5*(DSIN(THETA))**2)
3 +3.00 * C3
4 *DCOS(2.00)*(XLAM-XLAM22) / (2.00*R**5)
C OXLAM = NSQRT(SQ1-W
ISN 0085 C
ISN 0086 C CALL SSU
ISN 0087 1 SITH = OSIN(THETA)
ISN 0088 SITHSQ= SITH * SITH

```

```

ISN 0089          COTH = OCOS(THETA)          0203
ISN 0090          COTHSQ= COTH * COTH        0204
ISN 0091          SIXLT2= COS(2.00 * (XLAM-XLAM22)) 0205
ISN 0092          COXLT2= COS(2.00 * (XLAM-XLAM22)) 0206
ISN 0093          SITHT2 = OSIN (2.00 * THETA)    0207
ISN 0094          DXLW = XLAM + W           0208
ISN 0095          DXLWSQ= DXLW*DXLW         0209
ISN 0096          R4 = R**4                 0210
ISN 0097          R5 = R4*R                 0211
ISN 0098          TEL = T                  0212
ISN 0099          XFORCE = FLAM (XLAM)       0213
CC               02R = R * DTHTA**2 + R * SITHSQ * DXLWSQ   0214
ISN 0100          1 - GM/R**2 + C1 * (3.00 * COTHSQ -1.00) / R4 0215
ISN 0101          D2XLAM=-C3 * SIXLT2 / R5          0216
ISN 0102          1 - DXLW * COTH * 2.00 * DTHTA / SITH 0217
ISN 0103          2 - 2.00 * DR * DXLW / R           0218
ISN 0104          3 + (XFORCE / (SITH * R)) * FDLAM 0219
CC               02THET=-2.00 * DR * DTHTA / R          0220
ISN 0105          1 + DXLWSQ * SITHT2 / 2.00      0221
ISN 0106          2 + C1 * SITHT2 / R5           0222
ISN 0107          3 + C4 * SITHT2 * COXLT2 / R5 + XFDRC * FRECT / R 0223
ISN 0108          GO TO 23                      0224
ISN 0109          C
ISN 0110          52 SITH = DSIN(THETA)          0225
ISN 0105          SITHSQ= SITH * SITH          0226
ISN 0106          COTH = OCOS(THETA)          0227
ISN 0107          COTHSQ= COTH* COTH        0228
ISN 0108          REJR = RE/R                0229
ISN 0109          REDRSQ= REDRSQ          0230
ISN 0110          FR = GM/(R**2) * (-1.00 + REDRSQ * 0231
ISN 0105          1 (1.5 * XJ20 * (3.00 * COTHSQ - 1.00) 0232
ISN 0106          2 + 9.00 * XJ22 * SITHSQ * DCOS (2.00*(XLAM-XLAM22)) 0233
ISN 0107          3 + 2.00 * REDR * XJ30 * (5.00 * COTHSQ -3.00) * COTH 0234
ISN 0108          4 + 6.00 * REDR * XJ31 * (5.00 * COTHSQ -1.00) * SITH * 0235
ISN 0109          5 (DCOS(XLAM-XLAM31))          0236
ISN 0110          6 + 60.00 * REDR * XJ32 * SITHSQ*COTH*DCOS (2.00*(XLAM-XLAM32)) 0237
ISN 0111          7 + 60.00 * REDR * XJ33 * SITHSQ* DCOS (3.00* 0238
ISN 0112          8 (XLAM-XLAM33))          0239
ISN 0111          BETTER CALCULATION OF DXLAM D2R IS NOT 0 0240
ISN 0112          SQ = (D2R -FR) / (R * SITHSQ) 0241
ISN 0112          DXLAM = DSQR1(SQ) - W 0242
ISN 0111          C
ISN 0112          C

```

```

ISN 0113      CALL SSS          0253
C      51 SITH = OSIN (THETA)    0254
ISN 0114      SITHSO= SITH * SITH 0255
ISN 0115      SITHCU= SITHSQ * SITH 0256
ISN 0116      SITHT2= OSIN (2.00 * THETA) 0257
ISN 0117      COTH = OCOS (THETA) 0258
ISN 0118      COTHSQ= COTH * COTH 0259
ISN 0119      COTHS5= 5.00 * COTHSQ 0260
ISN 0120      SICOTH= SITH*COTH 0261
ISN 0121      R2 = P* R 0262
ISN 0122      R4 = R**4 0263
ISN 0123      RS = R4*R 0264
ISN 0124      REDR = RF/R 0265
ISN 0125      REOR= REOR*REOR 0266
ISN 0126      REORSQ= REOR*REOR 0267
ISN 0127      REXJ30= REDP * XJ30 0268
ISN 0128      REXJ31= REDP * XJ31 0269
ISN 0129      RXCS51= REXJ31 * (COTHSS - 1.00) 0270
ISN 0130      REXJ32= REOR * XJ32 0271
ISN 0131      REXJ33= REDP * XJ33 0272
ISN 0132      RXSISQ= REXJ33 * SITHSQ 0273
ISN 0133      GMOPSQ= G4/R2 0274
ISN 0134      ORT2DR= 2.00 * OR/R 0275
ISN 0135      OXLW = OXLAM + W 0276
ISN 0136      DXLWSO= OXLW*DXLW 0277
ISN 0137      XLAMT?= 2.00 * XLAM 0278
ISN 0138      SIXLT?= DSIN(XLAMT?) 0279
ISN 0139      COXLIT2= DCOS(XLAMT2) 0280
ISN 0140      XL22T2= 2.00 * (XLAM-XLAM22) 0281
ISN 0141      CX22T2= DCOS (XL22T2) 0282
ISN 0142      CXJ22T= XJ22 * CX22T2 0283
ISN 0143      XL31 = XLAM-XLAM31 0284
ISN 0144      COXL31= OCOS (XL31) 0285
ISN 0145      XL32T2= 2.00 * (XLAM-XLAM32) 0286
ISN 0146      CX32T2= DCOS (XL32T2) 0287
ISN 0147      XL33T3= 3.00 * (XLAM-XLAM33) 0288
ISN 0148      TEL = T 0289
ISN 0149      XFIRCF = FLAM (XLAM) 0290
C      FK = GMOPSQ * {1.00 0291
ISN 0150      + * REORSQ * {1.500 * XJ20 * {3.00 * COTHSO - 1.00 } 0292
C      1   + 9.00 * SITHSO * CJJ2T 0293
C      2   + 2.00 * RFXJ30 * {COTHSS - 3.00 } * COTH 0294
C      3   + 6.00 * RXCS51 * SITH * CXPL31 0295
C      4   + 6.00 * RXJ32 * SITHSO * COTH * CX32T2 0296
C      5   + 6.00 * RXJ33 * SITHCU * NCOS (XL33T3) 0297
C      6   + 6.00 * ALONGR 0298
C      7   + XFORCE * ALONGR 0299
C      0300
ISN 0151      FXLAM = GMOPSQ * REORSQ * {5.00 * XJ22 * SITH * SIXT? 0301
ISN 0151      + 1.500 * RXCS51 * SITHCU * NSIN(XL31) 0302

```

```

2      +30.00 * REXJ32          * SICOTH * DSINXL32T2)
3      +45.00 * RXSISQ          * DSINTXL33T3) )
4      + XFORCE * FORLAM

ISN 0152      C   FTHETA= -GMORSQ * RERSQ * (-3.00 * XJ20 * SICOTH
1           +6.00 * XJ22 * SICOTH * CX22T2
2           -1.500 * REXJ30 * (COTH55-1.00) * SITH
3           +1.500 * REXJ31 * (15.00 * COTH5Q - 11.00) * CJTH * COXL31
4           +15.00 * REXJ32 * (3.00 * COTH5Q - 1.00) * SITH * CX32T2
5           +45.00 * RXSISQ * COTH
6           + XFORCE * FORCCT

ISN 0153      C   D2R = R * OTHETA**2 + R * SITHSQ * OXLWSQ + FQ
ISN 0154      C   O2XLAM= -2.00 * OR * OXLWR
1           -2.00 * COTH * OTHETA * OXLW/SITH + FXLAM/(R*SITH)
C
C   D2THET= -2.00 * DR * OTHETA/R + FTHETA/R + DXLWSQ * SITHT2/2.07
C
ISN 0155      C   CHANGE THE FOLLOWING CAROS IF A DIFFERENT OUTPUT MODE IS DESIRED
156      23 IF (T - TOUT) 21,14, 14
157      14 IF ( N - NOUT) 15, 15, 16
158      15 N = N+ 1
159      RT(N) = R
160      TH(N) = THETA
161      XLD(N) = XLAM
162      GO TO 21
163      CALL MINMAX (RT,N,RP,RA)
164      A = (RA + RP) /2.00
165      F = RA/A - 1.00
166      CALL MINMAX (TH, N, TMIN, TMAX)
167      XI = S0*DO - TMIN/XX
168      CALL MINMAX (XL0,N,XLMIN,XLMAX)
169      XLMID = XLMIN/XX
170      XLWD = XLMAX/XX
171      DIFFER = XLMAD - XLMID
172      WRITE (6,*,25A,E,XI,XLMIN,XLMAD,DIFFER
173      25 FORMAT(4H A = ,1PE15.6,4H E = ,1PE15.6,5H XI = ,1PF15.6,5H LMI =
11PE15.6,5H LMA= ,1PE15.6, 5H DIF= , 1PE15.6)
C
ISN 0174      C   GU TO (12,13),12
ISN 0175      12   GU TO (137,138),17
ISN 0176      137  Y2 = (XLMID + XLMAD) / 2.00
ISN 0177      GU TO 139
ISN 0178      138  Y2 = XLAM/XX
ISN 0179      139  COUNTUE
ISN 0180      WRITE (7,114) DAYS, A, Y2
ISN 0181      114  FORMAT (3F20.7)
C

```

```

0353 13 TOUT = TOUT + TWTDY
0354 N = 0
0355 GO TO 23
0356 21 IF(T - TPRINT) 4,3,3
0357 C
0358 3 TOEG = THETAXX
0359 XLOEG=XLAM/XX
0360 TDAYS = 1/86400.00
0361 DXLAM= 86400.00 * DXLAM/XX
0362 OTDEG = 86400.00 * OTTHETA/XX
0363 XLDFT = XLOEG
0364 IF (XLOEG-XLOEY) 34, 34, 35
0365 LLL = -1
0366 GO TO 36
0367 35 LLL = 1
0368 36 XLOEY = XLDEG
0369 XELONG= 360.00 - XLOEG
0370 XELONG = 360. - XLOEG
0371 C
0372 WRITE(6,15)TDAYS,R,DR,TDEG,DDEG,XLDEG,DXLDEG,XELONG
0373 5 FORMAT(1X,1P15.6)
0374 C
0375 K=-1
0376 TPRINT = TPRINT + OPRIN1
0377 4 IF (T.LT.TEND) GO TO 6
0378 IF (1.2*FOU) WRITE(7,88) ZAP
0379 88 FORMAT (F20.5)
0380 GO TO 7
0381 C
0382 6 T=FINDV(T,OT)
0383 R = DPNV(R,DR)
0384 THETA = DPNV(THETA,OTTHETA)
0385 XLAM = DPNV(XLAM,DXLAM)
0386 OR = DPNV(DR,DR)
0387 OTHETA = DPNV(OTHETA,O2THET)
0388 DXLAM = DPNV(DXLAM,O2XLAM)
0389 C
0390 C
0391 C
0392 C
0393 C
0394 C
0395 C
0396 C
0397 C
0398 C
0399 C
0400 C
0401 C
0402 C
0403 C
0404 C
0405 C
0406 C
0407 C
0408 C
0409 C
0410 C
0411 C
0412 C
0413 C
0414 C
0415 C
0416 C
0417 C
0418 C
0419 C
0420 C
0421 C
0422 C
0423 C
0424 C
0425 C
0426 C
0427 C
0428 C
0429 C
0430 C
0431 C
0432 C
0433 C
0434 C
0435 C
0436 C
0437 C
0438 C
0439 C
0440 C
0441 C
0442 C
0443 C
0444 C
0445 C
0446 C
0447 C
0448 C
0449 C
0450 C
0451 C
0452 C
0453 C
0454 C
0455 C
0456 C
0457 C
0458 C
0459 C
0460 C
0461 C
0462 C
0463 C
0464 C
0465 C
0466 C
0467 C
0468 C
0469 C
0470 C
0471 C
0472 C
0473 C
0474 C
0475 C
0476 C
0477 C
0478 C
0479 C
0480 C
0481 C
0482 C
0483 C
0484 C
0485 C
0486 C
0487 C
0488 C
0489 C
0490 C
0491 C
0492 C
0493 C
0494 C
0495 C
0496 C
0497 C
0498 C
0499 C
0500 C
0501 C
0502 C
0503 C
0504 C
0505 C
0506 C
0507 C
0508 C
0509 C
0510 C
0511 C
0512 C
0513 C
0514 C
0515 C
0516 C
0517 C
0518 C
0519 C
0520 C
0521 C
0522 C
0523 C
0524 C
0525 C
0526 C
0527 C
0528 C
0529 C
0530 C
0531 C
0532 C
0533 C
0534 C
0535 C
0536 C
0537 C
0538 C
0539 C
0540 C
0541 C
0542 C
0543 C
0544 C
0545 C
0546 C
0547 C
0548 C
0549 C
0550 C
0551 C
0552 C
0553 C
0554 C
0555 C
0556 C
0557 C
0558 C
0559 C
0560 C
0561 C
0562 C
0563 C
0564 C
0565 C
0566 C
0567 C
0568 C
0569 C
0570 C
0571 C
0572 C
0573 C
0574 C
0575 C
0576 C
0577 C
0578 C
0579 C
0580 C
0581 C
0582 C
0583 C
0584 C
0585 C
0586 C
0587 C
0588 C
0589 C
0590 C
0591 C
0592 C
0593 C
0594 C
0595 C
0596 C
0597 C
0598 C
0599 C
0600 C
0601 C
0602 C
0603 C
0604 C
0605 C
0606 C
0607 C
0608 C
0609 C
0610 C
0611 C
0612 C
0613 C
0614 C
0615 C
0616 C
0617 C
0618 C
0619 C
0620 C
0621 C
0622 C
0623 C
0624 C
0625 C
0626 C
0627 C
0628 C
0629 C
0630 C
0631 C
0632 C
0633 C
0634 C
0635 C
0636 C
0637 C
0638 C
0639 C
0640 C
0641 C
0642 C
0643 C
0644 C
0645 C
0646 C
0647 C
0648 C
0649 C
0650 C
0651 C
0652 C
0653 C
0654 C
0655 C
0656 C
0657 C
0658 C
0659 C
0660 C
0661 C
0662 C
0663 C
0664 C
0665 C
0666 C
0667 C
0668 C
0669 C
0670 C
0671 C
0672 C
0673 C
0674 C
0675 C
0676 C
0677 C
0678 C
0679 C
0680 C
0681 C
0682 C
0683 C
0684 C
0685 C
0686 C
0687 C
0688 C
0689 C
0690 C
0691 C
0692 C
0693 C
0694 C
0695 C
0696 C
0697 C
0698 C
0699 C
0700 C
0701 C
0702 C
0703 C
0704 C
0705 C
0706 C
0707 C
0708 C
0709 C
0710 C
0711 C
0712 C
0713 C
0714 C
0715 C
0716 C
0717 C
0718 C
0719 C
0720 C
0721 C
0722 C
0723 C
0724 C
0725 C
0726 C
0727 C
0728 C
0729 C
0730 C
0731 C
0732 C
0733 C
0734 C
0735 C
0736 C
0737 C
0738 C
0739 C
0740 C
0741 C
0742 C
0743 C
0744 C
0745 C
0746 C
0747 C
0748 C
0749 C
0750 C
0751 C
0752 C
0753 C
0754 C
0755 C
0756 C
0757 C
0758 C
0759 C
0760 C
0761 C
0762 C
0763 C
0764 C
0765 C
0766 C
0767 C
0768 C
0769 C
0770 C
0771 C
0772 C
0773 C
0774 C
0775 C
0776 C
0777 C
0778 C
0779 C
0780 C
0781 C
0782 C
0783 C
0784 C
0785 C
0786 C
0787 C
0788 C
0789 C
0790 C
0791 C
0792 C
0793 C
0794 C
0795 C
0796 C
0797 C
0798 C
0799 C
0800 C
0801 C
0802 C
0803 C
0804 C
0805 C
0806 C
0807 C
0808 C
0809 C
0810 C
0811 C
0812 C
0813 C
0814 C
0815 C
0816 C
0817 C
0818 C
0819 C
0820 C
0821 C
0822 C
0823 C
0824 C
0825 C
0826 C
0827 C
0828 C
0829 C
0830 C
0831 C
0832 C
0833 C
0834 C
0835 C
0836 C
0837 C
0838 C
0839 C
0840 C
0841 C
0842 C
0843 C
0844 C
0845 C
0846 C
0847 C
0848 C
0849 C
0850 C
0851 C
0852 C
0853 C
0854 C
0855 C
0856 C
0857 C
0858 C
0859 C
0860 C
0861 C
0862 C
0863 C
0864 C
0865 C
0866 C
0867 C
0868 C
0869 C
0870 C
0871 C
0872 C
0873 C
0874 C
0875 C
0876 C
0877 C
0878 C
0879 C
0880 C
0881 C
0882 C
0883 C
0884 C
0885 C
0886 C
0887 C
0888 C
0889 C
0890 C
0891 C
0892 C
0893 C
0894 C
0895 C
0896 C
0897 C
0898 C
0899 C
0900 C
0901 C
0902 C
0903 C
0904 C
0905 C
0906 C
0907 C
0908 C
0909 C
0910 C
0911 C
0912 C
0913 C
0914 C
0915 C
0916 C
0917 C
0918 C
0919 C
0920 C
0921 C
0922 C
0923 C
0924 C
0925 C
0926 C
0927 C
0928 C
0929 C
0930 C
0931 C
0932 C
0933 C
0934 C
0935 C
0936 C
0937 C
0938 C
0939 C
0940 C
0941 C
0942 C
0943 C
0944 C
0945 C
0946 C
0947 C
0948 C
0949 C
0950 C
0951 C
0952 C
0953 C
0954 C
0955 C
0956 C
0957 C
0958 C
0959 C
0960 C
0961 C
0962 C
0963 C
0964 C
0965 C
0966 C
0967 C
0968 C
0969 C
0970 C
0971 C
0972 C
0973 C
0974 C
0975 C
0976 C
0977 C
0978 C
0979 C
0980 C
0981 C
0982 C
0983 C
0984 C
0985 C
0986 C
0987 C
0988 C
0989 C
0990 C
0991 C
0992 C
0993 C
0994 C
0995 C
0996 C
0997 C
0998 C
0999 C
0999 C

```

LEVEL 2 FER 67

OS/360 FORTRAN H

DATE 68.067/17.15.57

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINFCNT=50,SOURCE,ERC0IC,NULIST,NDFCK,LOAD,MAP,NOINIT,IN

```
ISN 0002      SUBROUTINE ALVISE(DDAY,VALUE)
C          SUBROUTINE TO PICK UP A NEW EPHEMERIS CORRECTION VALUE
C          THE ROUTINE ASSUMES TABLES DEFINING A STEP FUNCTION HAVE BEEN CO
C          MPILED AND STORED IN /STEPS/ COMMON
C
C          THE ROUTINE SCANS THE GDAY ARRAY LOOKING FOR THE FIRST DAY LESS
C          THAN THE INPUT DAY. IT THEN USES THE INDEX FOUND BY THE SCAN TO
C          REFERENCE THE CORRECT TABLE
C
C          REAL*8 DDAY,VALUE
C          DATA DAYOLD,INDOLD,DEGRAD/366.,81.,174532925F-1/
C
C          IN AN ATTEMPT TO SPEED UP THE TABLE LOOK UP WE ARE KEEPING THE
C          OF THE OLD VALUE ON THE LAST CYCLE AND STARTING THE IJ FROM THE
C          PREVIOUS DAY VALUE
C          COMMON /STEPS/ GDAY(200),CORRECT(200),INDLST
C          DAY = DDAY
C          IF(DAYOLD .GT. DDAY)  INDOLD = 1
C          I    =  INDOLD + 1
C          DO I=II,INDLST
C              J = I - 1
C              IF(DAY - (GDAY(I)) .GT. 0.)  I = I + 1
C          CONTINUE
C          J =  INDLST
C
C          UNABLE TO FIND A CORRECT VALUE
C          IMPOSSIBLE -----
C
C          ?  COUNTNUF
C          DAYOLD = GDAY(J)
C          INDOLD = J
C          VALUE = CORRECT(J)* DEGRAD
C          RETURN
C
ISN 0003
ISN 0004
ISN 0005
ISN 0006
ISN 0007
ISN 0009
ISN 0010
ISN 0011
ISN 0012
ISN 0013
ISN 0014
ISN 0015
ISN 0016
ISN 0017
ISN 0018
ISN 0019
ISN 0020
ISN 0021
ISN 0022
ISN 0023
ISN 0024
ISN 0025
ISN 0026
ISN 0027
ISN 0028
ISN 0029
ISN 0030
```

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 68-067/17:16:01

```
MAIN:DPRI=00;INFCNI=50;SOURCE•EBCDIC;NOISI;NODCK;NODAO;MAP:NODELL;IS
```

ISN 0035	DATA GOAY(32),CORRECT(32) /125.,-0.50	/ /	0480
ISN 0036	DATA GOAY(33),CORRECT(33) /130.,-0.75	/ /	0481
ISN 0037	DATA GOAY(34),CORRECT(34) /135.,-1.00	/ /	0482
C	C *	* *	0483
C	C *	* *	0484
C	C *	* *	0485
C	C *	* *	0486
C	C *	* *	0487
C	C *	* *	0488
C	STATE B		
ISN 0038	DATA GOAY(35),CORRECT(35) /142.,-0.75	/ /	0489
ISN 0039	DATA GOAY(36),CORRECT(36) /149.,-0.50	/ /	0490
ISN 0040	DATA GOAY(37),CORRECT(37) /156.,-0.25	/ /	0491
ISN 0041	DATA GOAY(38),CORRECT(38) /163.,0.00	/ /	0492
ISN 0042	DATA GOAY(39),CORRECT(39) /170.,0.25	/ /	0493
ISN 0043	DATA GOAY(40),CORRECT(40) /177.,0.50	/ /	0494
ISN 0044	DATA GOAY(41),CORRECT(41) /184.,0.75	/ /	0495
ISN 0045	DATA GOAY(42),CORRECT(42) /191.,1.00	/ /	0496
ISN 0046	DATA GOAY(43),CORRECT(43) /198.,1.25	/ /	0497
ISN 0047	DATA GOAY(44),CORRECT(44) /205.,1.50	/ /	0498
ISN 0048	DATA GOAY(45),CORRECT(45) /212.,1.75	/ /	0499
C	C *	* *	0500
C	C *	* *	0501
C	C *	* *	0502
C	C *	* *	0503
C	C *	* *	0504
C	C *	* *	0505
C	STATE C		
ISN 0049	DATA GOAY(46),CORRECT(46) /216.,1.50	/ /	0506
ISN 0050	DATA GOAY(47),CORRECT(47) /220.,1.25	/ /	0507
ISN 0051	DATA GOAY(48),CORRECT(48) /224.,1.00	/ /	0508
ISN 0052	DATA GOAY(49),CORRECT(49) /228.,0.75	/ /	0509
ISN 0053	DATA GOAY(50),CORRECT(50) /232.,0.50	/ /	0510
ISN 0054	DATA GOAY(51),CORRECT(51) /236.,0.25	/ /	0511
ISN 0055	DATA GOAY(52),CORRECT(52) /240.,0.00	/ /	0512
ISN 0056	DATA GOAY(53),CORRECT(53) /244.,-0.25	/ /	0513
ISN 0057	DATA GOAY(54),CORRECT(54) /248.,-0.50	/ /	0514
ISN 0058	DATA GOAY(55),CORRECT(55) /252.,-0.75	/ /	0515
ISN 0059	DATA GOAY(56),CORRECT(56) /256.,-1.00	/ /	0516
ISN 0060	DATA GOAY(57),CORRECT(57) /260.,-1.25	/ /	0517
ISN 0061	DATA GOAY(58),CORRECT(58) /264.,-1.50	/ /	0518
ISN 0062	DATA GOAY(59),CORRECT(59) /264.,-1.75	/ /	0519
ISN 0063	DATA GOAY(60),CORRECT(60) /272.,-2.00	/ /	0520
ISN 0064	DATA GOAY(61),CORRECT(61) /276.,-2.25	/ /	0521
ISN 0065	DATA GOAY(62),CORRECT(62) /291.,-2.50	/ /	0522
ISN 0066	DATA GOAY(63),CORRECT(63) /284.,-2.75	/ /	0523
ISN 0067	DATA GOAY(64),CORRECT(64) /238.,-3.00	/ /	0524
ISN 0068	DATA GOAY(65),CORRECT(65) /292.,-3.25	/ /	0525
ISN 0069	DATA GOAY(66),CORRECT(66) /296.,-3.50	/ /	0526
ISN 0070	DATA GOAY(67),CORRECT(67) /300.,-3.75	/ /	0527

```

ISN 0071           DATA GDAY(68),CORRECT(68)/ 3D4.,-4.00/
ISN 0072           DATA GDAY(69),CORRECT(69)/ 308.,-4.25/
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C           STATE D
C
ISN 0073           DATA GDAY(70),CORRECT(70)/ 311.,-4.00/
ISN 0074           DATA GDAY(71),CORRECT(71)/ 314.,-3.75/
ISN 0075           DATA GDAY(72),CORRECT(72)/ 317.,-3.50/
ISN 0076           DATA GDAY(73),CORRECT(73)/ 320.,-3.25/
ISN 0077           DATA GDAY(74),CORRECT(74)/ 323.,-3.00/
ISN 0078           DATA GDAY(75),CORRECT(75)/ 326.,-2.75/
ISN 0079           DATA GDAY(76),CORRECT(76)/ 329.,-2.50/
ISN 0080           DATA GDAY(77),CORRECT(77)/ 332.,-2.25/
ISN 0081           DATA GDAY(78),CORRECT(78)/ 335.,-2.00/
ISN 0082           DATA GDAY(79),CORRECT(79)/ 338.,-1.75/
ISN 0083           DATA GDAY(80),CORRECT(80)/ 341.,-1.50/
ISN 0084           DATA GDAY(81),CORRECT(81)/ 344.,-1.25/
ISN 0085           DATA GDAY(82),CORRECT(82)/ 347.,-1.00/
ISN 0086           DATA GDAY(83),CORRECT(83)/ 350.,-0.75/
ISN 0087           DATA GDAY(84),CORRECT(84)/ 353.,-0.50/
ISN 0088           DATA GDAY(85),CORRECT(85)/ 356.,-0.25/
ISN 0089           DATA GDAY(86),CORRECT(86)/ 359., 0.00/
ISN 0090           DATA GDAY(87),CORRECT(87)/ 362., 0.25/
ISN 0091           DATA GDAY(88),CORRECT(88)/ 365., 0.50/
C
C           CONTROL VALUE TO DEFINE THE LAST ELEMENT OF THE TABLES
C
ISN 0092           DATA INDST /99/
ISN 0093           ENO

```

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 6R.067/17.16.05

COMPILER OPTIONS - NAME= MAIN,DPT=00,LINFCNT=50,SOURCE,FRCNIC,NOLIST,NOECK,LDA0,MAP,NOFDT,1D

```
ISN 0002      SUBROUTINE CURT (DDAY, VALUE)
C          C SUBROUTINE TO PICK UP AN EPHEMERIS SUN SENSOR VALUE
C          C THE ROUTINE ASSUMES TABLES DEFINING A STEP FUNCTION HAVE BEEN
C          C COMPILED AND STORED IN /SPETS/ COMMON
C          C THE ROUTINE SCANS THE BODY ARRAY LOOKING FOR THE FIRST DAY LESS
C          C THAN THE INPUT DAY. IT THEN USES THE INDEX FOUND BY THE SCAN TO
C          C REFERENCE THE BORECT TABLE
C          REAL * 8 DDAY, VALUE
C          DATA DAYOLD, INDOOL /366.*24./
C          C AN ATTEMPT TO SPEED UP THE LOOK UP WE ARE KEEPING TRACK OF THE
C          C OLD VALUE ON THE LAST CYCLE AND STARTING THE TLY FROM THE
C          C PREVIOUS DAY VALUE.
C          COMMON /SPETS/ BOAY(100), BORECT(100), INDEX
C          C
C          DAY = DDAY
C          IF(DAYOLD .GT. DDAY) INDOOL = 1
C          II = INDOOL + 1
C          DO 1 I=II,INDEX,1
C          J = I-1
C          IF(DAY-BDAY(I)) 2,1,1
C          1 CONTINUE
C          J = INDEX
C          2 CONTINUE
C          DAYOLD = BDAY(J)
C          INDOOL = J
C          VALUE = BORECT(J)
C          RETURN
C          ENO
ISN 0003
ISN 0004
ISN 0005
ISN 0006
ISN 0007
ISN 0008
ISN 0009
ISN 0010
ISN 0011
ISN 0012
ISN 0013
ISN 0014
ISN 0015
ISN 0016
ISN 0017
ISN 0018
ISN 0019
ISN 0020
```

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 68.067/17.16.08

COMPILER OPTIONS - NAME= MAIN,OPT=0,LINECNT=50,SDURCC,FBCOIC,NOLIST,NODECK,LDA,MAP,NOEDIT,IO

```

ISN 0002      C      BLOCK DATA
ISN          C      DATA SUBROUTINE THE STEP FUNCTIONS FOR AN EPHEMERIS SUN SENSOR
ISN          VALUE
ISN 0003      CDMMDN /SPETS/ BDAY(100),BIRECT(100),INDEX
ISN 0004      DATA BOAY(1),BIRECT(1)/0.,-2.0/
ISN 0005      DATA BDAY(2),BIRECT(2)/10.0,0.9/
ISN 0006      DATA BDAY(3),BIRECT(3)/20.,-1.1/
ISN 0007      DATA BDAY(4),BIRECT(4)/30.,-1.0/
ISN 0008      DATA BOAY(5),BIRECT(5)/40.,-0.8/
ISN 0009      DATA BDAY(6),BIRECT(6)/50.,-0.5/
ISN 0010      DATA 3DAY(7),BIRECT(7)/60.,-0.3/
ISN 0011      DATA BUAY(8),BIRECT(8)/61.,0.0/
ISN 0012      DATA 3DAY(9),BIRECT(9)/98.,0.0/
ISN 0013      DATA 3DAY(10),BIRECT(10)/99.,0.3/
ISN 0014      DATA BDAY(11),BIRECT(11)/109.,0.4/
ISN 0015      DATA BOAY(12),BIRECT(12)/119.,0.6/
ISN 0016      DATA BOAY(13),BIRECT(13)/129.,1.0/
ISN 0017      DATA BOAY(14),BIRECT(14)/139.,0.8/
ISN 0018      DATA BDAY(15),BIRECT(15)/149.,-0.8/
ISN 0019      DATA BDAY(16),BIRECT(16)/159.,0.8/
ISN 0020      DATA BOAY(17),BIRECT(17)/169.,-1.0/
ISN 0021      DATA BOAY(18),BIRECT(18)/179.,-0.8/
ISN 0022      DATA BOAY(19),BIRECT(19)/189.,-0.6/
ISN 0023      DATA BOAY(20),BIRECT(20)/199.,-0.4/
ISN 0024      DATA BOAY(21),BIRECT(21)/200.,0.C/
ISN 0025      DATA BOAY(22),BIRECT(22)/314.,0.0/
ISN 0026      DATA 3DAY(23),BIRECT(23)/315.,0.5/
ISN 0027      DATA BOAY(24),BIRECT(24)/325.,0.7/
ISN 0028      DATA BDAY(25),BIRECT(25)/335.,1.0/
ISN 0029      DATA BDAY(26),BIRECT(26)/345.,1.1/
ISN 0030      DATA BOAY(27),BIRECT(27)/355.,1.6/
ISN 0031      DATA BDAY(28),BIRECT(28)/365.,-2.0/
ISN 0032      DATA INDEX /28/
ISN 0033

```

LEVEL 2 FEB 67

OS/360 FORTRAN H DATE 68-067/17-16-11

```

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,FBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ND
ISN 0002      C   SUBROUTINE GETINP          0624
ISN 0003      C   IMPLICIT REAL * 8 IA-H,O-Z  0625
C
C BEWARE      SOME ENTRIES IN COMMON/PARAMS/ ARE DIFFERENT FROM THEIR 0626
C             COUNTERPARTS IN COMMON/PARAMS/ IN MAIN PROGRAM 0627
C BEWARE      COMMON/PARAMS/FITHDY,CONST * XLIIMXD,XDISP, XMASS * CUT 0628
C             E,XI,OL * XLAND * WFND * ALNGR, DATE, 0629
C             FORCCT,FORLAM, A1, A2, W, T1, T2, 0630
C             13, I4, 15, 16, 17, NUM 0631
C
C             COMMON/DOVE/XLIMO 0632
C             COMMON/MAN/C1,C2,C3,DMAX,CONN,NAV,N 0633
C             COMMON/ESL/NEWFR 0634
C             COMMON/SUPSTF/ AB2, OFFSET, 18 0635
C
C             LOGICAL PUNCH,CUTDWN,AVERAGE,THIRD,LINEAR,LINRAN,PERFCT,STEP, 0643
C             1 STRAN,PRINT,NEWFR,EPH,EPRAN,SAIL1,SAIL2,SAIL3 0644
C             2, FLIPW,FLIPAG,TACSL,SAIL4 0645
C
C             NAMELIST /SETUPS/ NUM * CONST * XLIIMD,XDISP,XMASS,CUT 0646
C             1A,E,XI,OL,XLAND,TEND,ALONGR,DATE,I2,PUNCH,I5,CUTOWN,I7,AVERAG,I1, 0647
C             2THIRD,I3,LINEAR,LINRAN,PERFCT,STEP,STRAN,PRINT,IHDAY,NEWFR, 0648
C             3 CI,C2,C3,EPH,EPRAN,SAIL1,SAIL2,A1,A2,SAIL3 0649
C             4,IB,IRUN,FLIPW,FLIPAG,TACSL,SAIL4,OFFSET,NAV 0650
CC
C             DIMENSION VF11(9,2),VF12(6,2),VF13(16,10),VF14(14,2),VF17(11,2) 0651
C             1, VF16(16,6) 0652
C             2, VF17(16,6) 0653
C             DIMENSION XVFI11(8),XVF12(16),XVF13(16),XVF136(16),XVF137(16), 0654
C             1 XVFI13(16),XVF135(16),XVF136(16),XVF137(16),XVF138(16), 0655
C             2 ,XVF139(16),XVF141(14),XVF142(14),XVF171(11),XVF172(11),XVF1216 0656
C             3 ,XVF161(16),XVF160(8),XVF130(16) 0657
C             4 ,XVF181(16),XVF182(16),XVF180(16) 0658
C             5 ,XVF183(16) 0659
C
C             EQUIVVALFCF (XVF11(1),VF11(1,1),(XVF112(1),VF11(1,2)), 0660
C             1 (XVF121(1),VF121(1)),(XVF122(1),VF12(1,2)), 0661
C             2 (XVF131(1),VF131(1)),(XVF132(1),VF13(1,2)), 0662
C             3 (XVF133(1),VF133(1,3)),(XVF134(1),VF13(1,4)), 0663
C             4 (XVF135(1),VF135(1,5)),(XVF136(1),VF13(1,6)), 0664
C             5 (XVF171(1),VF171(1)),(XVF172(1),VF17(1,2)), 0665
C             6 (XVF141(1),VF141(1)),(XVF142(1),VF14(1,2)), 0666

```



```

0722 DATA XVF160 /25H(20H
0723 DATA XVF142 /56H117H XJ22 = -1.700E-6 10X, 24H XLAM22 = -19.0 *
0724 1 .01745 1 /
0725 DATA XVF141 /56H117H XJ22 = -1.816E-6 10X, 24H XLAM22 = -15.0 *
0726 1 .01745 1 /
0727 DATA XVF1310 /30H(25H FIXED SAIL SOLAR SAIL1 CONSTANT FDRCF ALDNG LAMDA)/
0728 DATA XVF139 /44H(39H SOLAR SAIL1 CONSTANT FDRCF ALDNG LAMDA)/
0729 DATA XVF138 /58H(53H SUN SENSOR CORRECTION, RANDOM ERROR UF +/- .5
0730 1DEGREES)/
0731 DATA XVF137 /27H(22H SUN SENSOR CORRECTION)/
0732 DATA XVF136 /61H(56H STEP FUNCTION CORRECTION, RANDOM ERROR OF
0733 1+/- 5 DEGREES)/
0734 DATA XVF135 /30H(25H STEP FUNCTION CORRECTION)/
0735 DATA XVF134 /25H(20H CORRECTED PERFECTLY)/
0736 DATA XVF133 /59H(53H LINEAR SUN CORRECTION, RANDOM ERROR OF +/--
0737 1.5 DEGREES)/
0738 DATA XVF132 /27H(22H LINEAR SUN CORRECTION)/
0739 DATA XVF131 /23H(18H NO SUN CORRECTION)/
0740 DATA XVF122 /21H(16H NO PUNCH OUTPUT)/
0741 DATA XVF121 /18H(13H PUNCH OUTPUT)/
0742 DATA XVF112 /30H(25H 3RD ORDER HARMONICS USEFDY)/
0743 DATA XVF111 /30H(25H 2ND ORDER HARMONICS USEFDY)/
0744 C
0745 C
0746 17 FORMAT(8H XDISP =,F10.2,4X,3H XMASS =,F10.2,3X,9H ALNGL =,F10.2,
0747 1 ,OFFSET =, F10.2)
0748 20 FORMAT( 4H A = ,1PE15.6,4H E = ,1PE15.5, 4H I = ,1PE15.6,
0749 1 4H O = ,1PF15.6, 7H XLAM = , 1PE15.5, 7H TEND = ,1PE15.6)
0750 31 FORMAT(6H1NUM= ,15.9H CONST = ,1PE15.6, 9H XLIMD = ,1PE15.6,
0751 110H XLIMDXD = ,1PE15.6)
0752 39 FORMAT(11H NO FORCE CUT DOWN )
0753 40 FORMAT(11H FORCE CUT BY ,F10.4 )
0754 42 FORMAT(6H NUM =,13.5X,8H CONST =,E10.2,5X,8H XLIMD =,E10.2)
0755 43 FORMAT( 9H XLIMDXD =,E10.4)
0756 44 FORMAT( 4H A = ,E10.2,4X,4H E = ,E10.2,4X,5H XI = ,F10.2,4X,5H JL = ,
0757 1 E10.2)
0758 45 FORMAT( 7H XLAM =,F10.2,4X,7H TEND =,E10.2)
0759 50 FORMAT( ,RUN NUMBER= , 14)
0760 51 FORMAT( ,1 )
0761 52 FORMAT( *OYR=,*I5.5X,*DAY=*,I5)
0762 101 FORMAT( 17H DATE =,F10.2,5X,3H11=,13,3X,3H12=,13,3X,3H13=,13,3X,
0763 1 3H14=,13,3X,3H15=,I3,
0764 2 FORMAT( ,22X)
0765 889 FORMAT( NEW FORCE ROUTINE USED) )
0766 999 FORMAT( ORIGINAL FDRCF ROUTINE USED) )
0767 C
0768 C
0769 THESE CONSTANTS ARE NOT INITIALIZED BY DATA STATEMENTS BECAUSE
0770 WE COULD LOOP BACK TO THIS POINT IN THE PROGRAM AND WE WA
0771 WANT THEM RESET AT THAT POINT.

```

AND MOST OF THEM APPEAR IN A LABELED COMMON.....

```

C   C   C   A   = 42165.500
C   C   C   A1  = .900
C   C   C   A2  = .100
C   C   C   ALONGR= .0.D0
C   C   C   CDNST =--45D-8
C   C   C   CUT   = .00
C   C   C   DATE  = .0.D0
C   C   C   E    = .0.D0
C   C   C   FDRCCF = 0.000
C   C   C   FORLAM = 1.000
C   C   C   IRUN   = 0
C   C   C   ITHDAY = 10
C   C   C   NAV=4
C   C   C   NUM   = 6
C   C   C   DFFSET= -90.000
C   C   C   OI   = 0.00
C   C   C   TEND  = 1800.D0
C   C   C   XDISP = 70.00
C   C   C   XI   = 0.D0
C   C   C   XLAMO = 300.00
C   C   C   XLIMD = 297.D0
C   C   C   XLMXN= 350.00
C   C   C   XMASS = 132.D0
C
C   C   C   INITIAL CONDITIONS
C   C   C   PRINT = .TRUF.
C   C   C   I1   = I
C   C   C   I2   = 1
C   C   C   I3   = 1
C   C   C   I4   = 2
C   C   C   I5   = 1
C   C   C   I6   = 3
C   C   C   I7   = 2
C   C   C   I8   = 4
C
C   C   C   CONTROL NAMELIST PRINT
C   C   C   SECOND ORDER HARMONICS
C   C   C   PUNCH = .TRUE.
C   C   C   NO CORRECTION
C   C   C   NO CUT DOWN
C
C   C   C   NOT AVERAGE.D0
C   C   C   SAIL IS NOT FLIPPED
C
C   C   C   I1   = 1
C   C   C   I2   = 1
C   C   C   I3   = 1
C   C   C   I4   =2
C   C   C   I5   = 1
C   C   C   I6   = 3
C   C   C   I7   = 2
C   C   C   I8   = 4
C
C   C   C   0811
C   C   C   0812
C   C   C   0813
C   C   C   0814
C   C   C   0815
C   C   C   0816
C   C   C   0817
C
C   C   C   0819
C   C   C   0820
C   C   C   0821
C   C   C   0822
C   C   C   0823
C   C   C   0824
C   C   C   0825
C   C   C   0826
C   C   C   0827
C
C   C   C   0828
C   C   C   0829
C   C   C   0830
C   C   C   0831
C   C   C   0832
C   C   C   0833
C   C   C   0834
C   C   C   0835

```

```

C          AVERAG = •TRUE•
ISN 0086    CUTDN = •FALSE•
ISN 0087    EPH = •FALSE•
ISN 0088    EPHRAN = •FALSE•
ISN 0089    FLIPAG; = •FALSE•
ISN 0090    FLIPW = •FALSE•
ISN 0091    LINEAR = •FALSE•
ISN 0092    LINRAN = •FALSE•
ISN 0093    NEWFR=•FALSE•
ISN 0094    PERFECT = •FALSE•
ISN 0095    PRINT = •TRUE•
ISN 0096    PUNCH = •TRUE•
ISN 0097    SAIL1 = •FALSE•
ISN 0098    SAIL2 = •FALSE•
ISN 0099    SAIL3 = •FALSE•
ISN 0100    SAIL4 = •FALSE•
ISN 0101    STEP = •FALSE•
ISN 0102    STPRAN = •FALSE•
ISN 0103    TACSL = •FALSE•
ISN 0104    THIRD = •FALSE•
ISN 0105    XMAX=DAS(S1)/25.00
ISN 0106    XLM = XLM0

C          READ THF NEW CONTROL VALUES
C
C          RFAD (5,SETUPS)
C          C1 = CONST / 4.00
C          C2 = CONST
C          C3 = CONST
C          CONST=CONST/32
C          DMAX=DAS(S1)/25.00
C          XLM = XLM0

C          CHECK NUM FOR THE END OF DATA BEFORE WE GO THROUGH THE JUTPUT
C          PHASE OF THIS SECTION
C          IF (NUM .LT. 0) GO TO 10
C
C          MODIFY THE CONTROL VARIABLES AS A FUNCTION OF THF INPUT
C
C          IF (THIRD) II=2
C
C          IF (THIRD) II=2
C
C          DEFINE THE PUNCH VARIABLE FOR THE JUTPUT SUBROUTINE
C          •FITJUT•
C          IF (PUNCH) I? = I
C          IF (I? .EQ. 1) PUNCH = •TRUE•
C          CALL SETPUT (PUNCH)
C
C          IF (AVERAGE) I7=1

```

```

ISN 0125          IF(CUTDWN) I5=2          0869
ISN 0127          IF(LEPH) I3=7          0870
ISN 0129          IF(EPHRAN) I3=R          0871
ISN 0131          IF(FLIPAG) I8 = 2          0872
ISN 0133          IF(FLIPW) I8 = 1          0873
ISN 0135          IF(LLINEAR) I3=2          0874
ISN 0137          IF(LINRAN) I3=3          0875
ISN 0139          IF(PERFECT) I3=4          0876
ISN 0141          IF(SAIL1) I3 = 9          0877
ISN 0143          IF(SAIL2) I6 = 1          0878
ISN 0145          IF(SAIL3) I3 = 10         0879
ISN 0147          IF(SAIL4) I8 = 3          0880
ISN 0149          IF(SAIL4) I6 = 1          0881
ISN 0151          IF(STEP) I3=5          0882
ISN 0153          IF(STPRAN) I3=6          0883
ISN 0155          IF(TACSL) I6 = 2          0884
C               PRINT OUT THE NAMELIST AS A FUNCTION OF THE PARAMETER PRINT
C               WRITE (6,51)
C               IF(PRINT) WRITE(6,SETUPS)          0885
ISN 0157          IF(PRINT) WRITE(6,SETUPS)          0886
ISN 0158          CALL TIMOAT (IYR,IOAY)          0887
ISN 0160          WRITE (6,52) IYR,IOAY          0888
ISN 0161          FITHDY = ITHDAY          0889
ISN 0162          C
C               WRITE (6, 17, '31NUM,CONST,XLIMD,XLIMD
ISN 0163          WRITE (6, 17) XDISP, XMASS, ALONGR,OFFSET          0890
ISN 0164          WRITE (6,20A*,XI,OL,XLAM,END)          0891
ISN 0165          WRITE (6,101) DATE, 11, 12, 13, 14, 15, 16, 17,IR          0892
ISN 0166          C
C               IF(.NOT. PUNCH) GO TO 9          0893
ISN 0167          C
C               18 WRITE (7, 42) NUM, CONST, XLIMD          0894
ISN 0169          WRITE (7, 43) XLIMD          0895
ISN 0170          WRITE (7, 17) X0ISP, XMASS, ALONGR , OFFSET          0896
ISN 0171          WRITE (7, 44) A, E, XI, OL          0897
ISN 0172          WRITE (7, 45) XLAM, TEND          0898
ISN 0173          WRITE (7,101) DATE, 11, 12, 13, 14, 15, 16, 17,IR          0899
ISN 0174          C
ISN 0175          IF(NEWFR) GO TO 12          0900
ISN 0177          C
C               WRITE(7,999)          0901
ISN 0178          GO TO 9          0911
ISN 0179          12 WRITE(7,888)          0912
ISN 0180          CONTINUE          0913
ISN 0181          CALL RITOUT(VFI1(1,1))          0914
ISN 0182          CALL RITOUT(VFI2(1,12))          0915
ISN 0183          CALL RITOUT(VF13(1,13))          0916
ISN 0184          CALL RITOUT(VF4(1,14))          0917
                                         0918

```

```

ISN 0185      IF (SAIL4) I6 = 0
ISN 0187      CALL RITOUT (VF16 (1,16))
C
ISN 0188      IF (15-1)I,1,3
C
ISN 0189      1   WRITE(6,39)
ISN 0190      IF (PUNCH) WRITE(7,39)
ISN 0191      GO TO 4
C
ISN 0192      3   WRITE(6,40)CUT
ISN 0193      IF (PUNCH) WRITE(7,40)CUT
ISN 0194      4   CONTINUE
ISN 0195      CALL RITOUT (VF17(1,17))
ISN 0196      CALL RITCUT (VF18(1,18))
ISN 0197      CALL RITCUT (VF19(1,19))
ISN 0198      WRITE (6,5) 1RUN
ISN 0199      IF (PUNCH) WRITE(7,50) 1RUN
ISN 0200      IF (SAIL4) I6 = 1
ISN 0201      IF (16*EQ.3) I6=0
ISN 0202      IF (18*EQ.4) I3=0
ISN 0203      RETURN
ISN 0204
ISN 0205
ISN 0206
ISN 0207
ISN 0208      C
ISN 0209      10  IF (PUNCH) WRITE(7,11) ZAP,ZAP
ISN 0210      11  FORMATT(2F2.0,7)
C
ISN 0212      RETURN
ISN 0213      END
C
ISN 0214      0919
ISN 0215      0920
ISN 0216      0921
ISN 0217      0922
ISN 0218      0923
ISN 0219      0924
ISN 0220      0925
ISN 0221      0926
ISN 0222      0927
ISN 0223      0928
ISN 0224      0929
ISN 0225      0930
ISN 0226      0931
ISN 0227      0932
ISN 0228      0933
ISN 0229      0934
ISN 0230      0935
ISN 0231      0936
ISN 0232      0937
ISN 0233      0938
ISN 0234      0939
ISN 0235      0940
ISN 0236      0941
ISN 0237      0942

```

LEVEL 2 FEB 67

05/360 FORTRAN H

DATE 58.067/17.16.73

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,ERCDIC,NOLIST,NOECK,LNAM,MAP,NNFNT,IN

```
ISN 0002      SUBROUTINE MINMAX (X, N, XMIN, XMAX )          n943
ISN 0003          DIMENSION X(1500)                         n944
ISN 0004          DOUBLE PRECISION X, XMIN, XMAX           n945
ISN 0005          XMIN = X(1)                           n946
ISN C006          XMAX = X(1)                           n947
ISN 0007          DO 10 I=2,N                         n948
ISN 0008          IF (X(I)-XMIN) 1,2,2                 n949
ISN 0009          1 XMIN = X(I)                         n950
ISN 0010          2 IF (XMAX-X(I)) 3,10,10            n951
ISN 0011          3 XMAX = X(I)                         n952
ISN 0012          10 CONTINUEF                      n953
ISN 0013          RETURN                                n954
ISN 0014          END                                    n955
```

LEVEL 2 FEB 67

OS/360 FORTRAN H DATE 68.067/17.16.26

```

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,FACDTC,NOLIST,NOECK,LOAD,MAP,NOFDIT,IN

ISN 0002      SUBROUTINE NEWFOR(XLIMO,XTEMP,CON)          0956
C IN THIS PROGRAM WE WILL CALCULATE A FORCE,CON. THIS FORCE WILL DEPEND
C ON WHICH REGION THE VARIABLE XTEMP IS IN. WHEN WE GET THE SEQUENCE
C OF REGIONS +A-A+A OR -A+A-A WE WILL CALCULATE THE FORCE DIFFERENTLY.
C

ISN 0003      IMPLICIT REAL*8 (A-H,O-Z)                  0957
ISN 0004      DIMENSION CCOUNT(1800), XT(3)                0958
ISN 0005      COMMON/MAN/C1,C2,C3,OMAX,CONN,NAV,N          0959
ISN 0006      COMMON/EL/CAVE,J1,J2,J3,JANE,K,LAO        0960
ISN 0007      DATA NI /0/                                0960

C THIS SUBROUTINE WORKS WITH QUANTITIES IN OEGRFEST(UNITS)          0961
ISN 0008      XTEMP = XTEMP / .1745329251994329D-01    0962
ISN 0009      NI=NI+1                                    0963
ISN 0010      XT(NI)=XTEMP                            0964
ISN 0011      IF(NI.EQ.1) GO TO 8
ISN 0012      CUN=CONN
ISN 0013      IF(NI.LT.NAV)RETURN
ISN 0014      SUM=0.0
ISN 0015      DO 3 J=1,NAV
ISN 0016      SUM=XT(J) + SUM
ISN 0017      3
ISN 0018      XTEMP=SUM/NAV
ISN 0019      NI=0
ISN 0020      8
ISN 0021      N=2
ISN 0022      C THIS SECTION WILL DETERMINE WHERE XTEMP LIES          0965
ISN 0023      IF(XTEMP .GE.XLIMO) GO TO 10
ISN 0024      GO TO 20
C AT THIS POINT XTEMP IS GREATER THAN OR EQUAL TO XLIM          0966
ISN 0025      10 R=-1.0D0
ISN 0026      I=-1
ISN 0027      IF(XTEMP.GE.(XLIMO+4.00))GO TO 30
ISN 0028      IF(XTEMP.GE.(XLIMO+2.00))GO TO 50
ISN 0029      GO TO 70
ISN 0030      C AT THIS POINT XTEMP IS LESS THAN XLIMO          0967
ISN 0031      20 R=1.0D0
ISN 0032      I=1
ISN 0033      IF(XTEMP.LE.(XLIMO-4.00))GO TO 30
ISN 0034      IF(XTEMP.LE.(XLIMO-2.00))GO TO 50
ISN 0035      GO TO 70
ISN 0036      C XTEMP IS IN REGION C. SET JL=3 OR -3, FORCE=C3, OR -C3 AND J2,J3=0. 0968
ISN 0037      C TEST TO SEE IF WE HAVE BEEN IN C AWHILE          0969
ISN 0038      30 INEW = I*3
ISN 0039      IF(INEW .EQ. JI)GO TO 600
ISN 0040      JL = INEW
ISN 0041      CONN = R*C3
ISN 0042      CONN = R*C3
ISN 0043      K = I
ISN 0044

```

```

ISN 0045          0993
ISN 0046          0994
ISN 0047          0995
ISN 0048          0996
ISN 0049          0997
ISN 0050          0998
ISN 0051          0999
ISN 0052          1000
ISN 0053          1001
ISN 0054          1002
ISN 0055          1003
ISN 0056          1004
ISN 0057          1005
ISN 0058          1006
ISN 0059          1007
ISN 0060          1008
ISN 0062          1009
ISN 0064          1010
ISN 0066          1011
ISN 0068          1012
ISN 0070          1013
ISN 0072          1014
ISN 0074          1015
ISN 0076          1016
ISN 0078          1017
ISN 0079          1018
ISN 0080          1019
ISN 0082          1020
ISN 0083          1021
ISN 0085          1022
ISN 0086          1023
ISN 0087          1024
ISN 0088          1025
ISN 0089          1026
ISN 0090          1027
ISN 0091          1028
ISN 0092          1029
ISN 0093          1030
ISN 0094          1031
ISN 0095          1032
ISN 0096          1033
ISN 0097          1034
ISN 0098          1035
ISN 0099          1036
ISN 0100          1037
ISN 0101          1038
ISN 0102          1039
ISN 0103          1040
ISN 0104          1041
ISN 0105          1042

C   COUNT(K) = CONN
C   J2 = 0
C   J3 = 0
C   JANE = 0
C   CAVE = 0.D0
C   LAO = 0
C   GO TO 600
C
C   WE ARE IN REGION B. HAVE WE CHANGED REGIONS--IF SO, UPDATE AND SET
C   FORCE = C2 OR -C2.
50    INEW = 1*2
      IF(INEW .EQ. J1) GO TO 600
      IF(J1 .EQ.0) K=0
      CONN = R*C2
      GO TO 205
C
C   WE ARE IN REGION A, AND MUST DECIDE WHAT VALUE TO GIVE THE FORCE.
C   CHECK TO SEE IF WE ARE IN THE SAME REGION. HAVE WE CHANGED REGIONS?
70    INEW = I*1
      IF(J1 .EQ. 0)K=0
      IF(INEW .EQ. J1) GO TO 600
      C   DO WE JUST GO FROM B TO A.
      IF(IABS(J1) .EQ. 2)GO TO 200
      C   TEST FOR INITIAL SEQUENCE
      IF(JANE .EQ. 0) GO TO 180
C
C   HAVE PASSED THE INITIAL SEQUENCE AND CHANGED REGIONS. TEST FOR MAX DA
C   AND ZERO BUMP SECTION.
170   IF(IABS(INEW*J1*J2) .NE. 1 .AND. LAO .EQ. 0) GO TO 190
      IF(IABS(INEW*J1*J2) .NE. 1 .AND. LAO .EQ. 5) GO TO 181
      IF(OABS(CONN) .NE. 0.00 .AND. OABS(CONN) .LT. OMAX) GO TO 200
      IF(LAD .EQ. 5) GO TO 5
C
C   IN A AND SEQUENCING. JUST CHANGED REGIONS, MAX DAMP AND ZERO BUMP TAK
C   TAKEN CARE OF. ALSO COMING FROM B. (INEW*J1*J2=1)
110   IF(OABS(COUNT(K)) .NE. OABS(COUNT(K-1))) GO TO 135
C
C   CIF WE GET HERE IT MEANS WE ARE READY TO DIVIDE BY 2
130   CONN = -CONN/2.D0
      CAVE = CONN
      IF(OABS(CONN) .LT. OMAX)LAO=5
      GO TO 205
C
C   ABSOLUTE VALUE REMAINS THE SAME IF J3 .NE. 2.
135   IF(IABS(J3) .EQ. 2) GO TO 130
      CONN = -CONN
      CAVE = CONN
      GO TO 205
C
C   TEST HERE FOR INITIAL SEQUENCE.

```

```

ISN 0088      130 IF (IAS((INEW*J1*J2) *EQ. 1 ) GO TO 297
C   WE ARE IN A BUT HAVE NOT REACHED INITIAL SEQUENCE.
ISN 0090      CONN = R*C1
ISN 0091      GO TO 205
C
C   IN MAX DAMP ZERO BUMP SECTION AND INEW*J1*J2 NE 1
ISN 0092      181 IF (INN *EQ. 0.00) GO TO 5
ISN 0093      GO TO 200
C
C   INITIAL SEQUENCE JUST REACHED. MEANS CHANGE OF REGION
ISN 0095      207 JANE = 1
ISN 0096      CONN = R*C1/2.00
ISN 0097      CAVE = CONN
ISN 0098      GO TO 205
C
C   IN A AFTER THE INITIAL SEQUENCE. INEW*J1*J2 = NE. 1 .
ISN 0099      190 CONNE=CAVE
ISN 0100      GO TO 205
C
C   IN ZERO BUMP AND MAXIMUM DAMP SECTION
ISN 0101      5 CONN = R*C1/32.00
ISN 0102      GO TO 205
C
C   FROM B TO A
ISN 0103      200 CONN = 0.000
ISN 0104      C REGIONS HAVE JUST CHANGED. UPDATE SEQUENCE
ISN 0105      205 K = K+1
ISN 0106      J3 = J2
ISN 0107      J2 = J1
ISN 0108      J1 = INEW
ISN 0109      COUNT(K) = CONN
ISN 0110      600 CONN = CONN
ISN 0111      RETURN
END

```

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 68-067/17-16-32

LEVEL 2 FEB 67

DS/360 FORTRAN H

DATE 58.367/17.16.35

- COMPILE OPTIONS - NAME= MAIN,DPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NNFDIT,IN

```
1098
      SUBROUTINE SSL
      DOUBLE PRECISION X
      DOUBLE PRECISION CIND
      X=CIND(0,X,0.000)
      RETURN
      END
      ISN 0002
      ISN 0003
      ISN 0004
      ISN 0005
      ISN 0006
      ISN 0007
```

LEVEL 2 FEB 67

OS/360 FORTRAN H DATE 68.067/17.16.38

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50, SOURCE,EBCDIC,NOLIST,NOECK,LOAD,MAP,NOEDIT, ID

ISN 0002 C SUBROUTINE TACK(XLAM) 1104
C C
ISN 0003 C WRITTEN BY ELLEN SWENSON X5309 1105
ISN 0004 C IMPLICIT REAL * 8 (A-H,O-Z) 1105
ISN 0005 C DOUBLE PRECISION LORSTR 1107
ISN 0006 C LOGICAL CHECK 1108
C CHECK = .FALSE.
ISN 0007 C COMMON/Z8/XLIM , CONST , XLIMX , CUT , XNUM , FUEL , 1109
1 WS , COSI , FSUN , SINI , TEL , JFK ,
2 LLL , M , K , NUM , I3 , I5 , 1110
COMMON/PARAMS/FITHDAY,CONSTI,XLIMXD,XDISP, XMASS , UTBY , A ,
1 E , XI , OL , XLAMD , TEND , ALONGR , DATE , 1114
2 FORCT,FORLAM, A1 , A2 , W , I1 , I2 ,
3 13T , 14 , 15T , 16 , 17 , NUMIN , 1115
ISN 0009 C COMMON/SUPSTF / ABS , AR2 , OFFSET , I8 1116
C OBJECT 1117
C TO OBTAIN MAXIMUM FORCE ALONG ORBITAL TANGENT 1118
C 1120
C UNDERLYING THEORY 1121
C LARRY BLACK'S IDEA-- OPTIMUM POSITION FOR SAIL IS ALONG 1122
C BISECTOR OF THE ANGLE BETWEEN THE PROJECTION OF THE SUN IN THE 1123
C ORBITAL PLANE AND THE ORBITAL TANGENT 1124
C THERE IS PROVISION FOR CHANGING THE POSITION OF SAIL 1125
C ROTATE SAIL *OFFSET* DEGREES FROM TACKED POSITION 1126
C THERE IS AN ENTRY CHAN AND I8 VARIABLES 1127
C CHAN PLACES SAIL ON THE RADIUS 1128
C I8 FLIPS THE SAIL THREE WAYS 1129
C FLIPS SAIL WITH ORBIT 1130
C FLIPS SAIL AGAINST ORBIT 1131
C FLIPS SAIL EVERYWHERE 1132
C BOTH SIDES OF THE SAIL ARE REFLECTIVE 1133
ISN 0010 ABS = .100 1134
ISN 0011 AR2 = .100 1135
C 1136
C ALONGR IS FORCE ALONG R 1137
C FORCT IS FORCE ALONG CTHETA 1138
C FORLAM IS FORCE ALONG LAMBDA 1139
C 1140
C OARCUS(DOTPROJ) IS ANGLE BETWEEN NORMAL TO THE SAIL AND THE SUN 1141
C CTHETA IS ANGLE BETWEEN ARCS AND RADIUS 1142
C 1143
ISN 0012 300 CONTINF 1144
C GEOCENTRIC COORDINATES OF NORMAL TO RADIUS 1145
1 CTHETA = XLAM + W * TEL 1146
XNORMA = DSIN(CTHETA) 1147
1148
ISN 0013 1149
ISN 0014 1150
1151
1152

```

ISN 0015      YNORMA = -OCOS(CTHETA)
ISN 0016      ZNORMA = 0.000
C
C GEOCENTRIC COORDINATES OF SUN
C V = TRUE ANOMALY (MOULYON P171)   XM = MEAN ANOMALY
ISN 0017      DAY = TEL/86400.00
ISN 0018      XM = (1.98560000 * DAY -3.392900) * XX
ISN 0019      V = XM + 2.00 * ESUN * OSIN(XM) + 5.00 * FSUN * DSIN(2.00
                  * XM) / 4.00
C
ISN 0020      C
ISN 0021      U = V + WS
                  SINU= DSIN(U)
C
ISN 0022      XSUN= OCOS(U)
ISN 0023      YSUN= SINU * COSI
ISN 0024      ZSUN= SINU * SINI
C
C GEOCENTRIC COORDINATES OF THE ORBITAL TANGENT
ISN 0025      XFORCE = -XNORMA
ISN 0026      YFORCE = -YNORMA
ISN 0027      ZFORCE = ZNORMA
C
C GEOCENTRIC COORDINATES OF PROJECTION OF THE SUN IN THE ORBITAL PLANE
ISN 0028      XMAINE = OSQRT1(XSUN**2 + YSUN**2)
ISN 0029      XPROJ = XSUN/XMAINE
ISN 0030      YPROJ = YSUN/XMAINE
C
ISN 0031      IF (CHECK) GO TO 310
C
C COORDINATES OF SAIL ITSELF
ISN 0033      XTEMP = XFORCE + XPROJ
ISN 0034      YTEMP = YFORCE + YPROJ
C
C COORDINATES OF NORMALIZED SAIL VECTOR
ISN 0035      YMAINE= DSQRT1(XTFMP**2 + YTTEMP**2)
ISN 0036      XTEMP = XTEMP / YMAINE
ISN 0037      YTTEMP = YTTEMP / YMAINE
C
C COORDINATES OF SAIL ITSELF
ISN 0038      SINOFF= OSIN(OFFSET)
ISN 0039      COSOFF= OCOS(OFFSET)
ISN 0040      XRESAL= XTEMP* COSOFF + YTTEMP* SINOFF
ISN 0041      YRESAL= YTTEMP * COSOFF - XTEMP * SINOFF
ISN 0042      XSAIL = YRESAL
ISN 0043      YSAIL = -XRESAL
ISN 0044      GO TO 3
C
ISN 0045      C 310  CONTINUE
ISN 0046      XSAIL = YNORMA
ISN 0047      YSAIL = YNORMA

```

```

ISN 0048
ISN 0049      XRESAL = -YSAIL
ISN 0050      YRFSAL = XSAIL
C      3  OUTPRO = XSAIL *XSUN + YSAIL *YSUN
C      C ANGLE BETWEEN SAIL AND RADIUS = ANCHOR
C      LOSTR= XFORC*(-YSAIL) + YFORCE*(-YSAIL)
C      ANCHOR= DARCOS(LOSTR)
C      B1AT = DSIN(ANCHOR)

ISN 0051
ISN 0052
ISN 0053      C   ROCK = OABS(LLOSTR)
ISN 0054      TRAP = DARS(DOTPRO)
ISN 0055      IF(OUTPRO .LE. 0.00) AB1 = AB2
ISN 0056      DOT = (2.000 - AB1) * OUTPRO**2
ISN 0058      C   ALUSAL IS FORCE ALONG SAIL DUE TO NON-PERFECT SAIL
ISN 0059      ALOSAL= -AB1 * TRAP *(XSUN*XRESAL + YSUN*YRFSAL)
C      C TO DETERMIN DIRECTION OF FORCE COMPONENTS OF ALUSAL AND FOR LAW
ISN 0060      IF (IB .EQ. 1 .AND. DOTPRO .LE. 0.00) GO TO 40
ISN 0062      IF (IB .EQ. 2 .AND. DOTPRO .GE. 0.00) GO TO 40
ISN 0064      IF (IB .EQ. 3) GO TO 40
C      COD FISH HERRING SALT
C      ARE INITIALIZED TO +1 ABD MADE NEGATIVE WHEN NECESSARY
C      C00 IS NEGATIVE WHEN
C      C DIRECTION OF SOLAR FORCE IS MORE THAN 90 DEG FROM RADIUS
C      C FISH IS NEGATIVE WHEN
C      C SAIL IS MORE THAN 90 DEGREES FROM RADIUS
C      C HERING IS NEGATIVE WHEN
C      C DIRECTION OF SOLAR FORCE IS MORE THAN 90DEG FROM JBITAL TAN
C      C SALT IS NEGATIVE WHEN
C      C SAIL IS MORE THAN 90 DEGREES FROM ORBITAL TANGENT
C      C
ISN 0066      COD = 1.000
ISN 0067      FISH = 1.000
ISN 0068      HERING= 1.000
ISN 0069      SALT = 1.000
C      C COORDINATES OF RADIUS
C      C X COORDINATE -YNORMA
C      C Y COORDINATE XNORMA
C      C COORDINATES OF SOLAR RADIATION FORCE
C      C TESTING ON Z-COMPONENT OF CROSS PRODUCT OF SAIL CROSS SUN
C      C

```

```

ISN 0070 IF((YRESAL * YPROJ - XPROJ * YRESAL) * LE.0.D0) GO TO 7
ISN C072 YSORAF= XSAIL
ISN 0C73 YSORAF= YSAIL
ISN 0074 GO TO 6

ISN 0075 C 7 YSORAF= -XSAIL
ISN 0076 C YSORAF= -YSAIL

ISN 0077 C 6 CONTINUE
ISN 0078 IF((XSORAF*XFORCE + YSORAF*YFORCE).LE.0.000) HERING=-1.0D0
ISN 0080 IF((XPESAL * XFORCE + YRESAL * YFORCE) * LE.0.D0) SALT=-1.0D0
ISN 0082 IF((XSORAF * (-YNORMA) + YSORAF * XNORMA).LE.0.000) COO=-1.000
ISN 0084 IF((-YNORMA)*XRESAL + XNORMA * YRESAL * LE. 0.D0) FISH=-1.D0

ISN 0086 C FORLAM=
ISN 0087 ALONGR= HERING*POT*POCK + SALT*ALDSAL*BOAT
ISN 0088 FORCCT= COD*POT*BOAT + FISH*ALDSAL*ROCK
ISN 0089 C RETURN

ISN 0090 C 40 CONTINUE
ISN 0091 DOTPR2= DABSI(ZSUN)
ISN 0092 DOTPR1= XSUN * (-YSAIL) + YSUN * XSAIL
ISN 0093 FORLAM= AB1 * DOTPR2 * DOTPRO
ISN 0094 ALONGR= -AB1 * DOTPR2 * DOTPR1
ISN 0095 C TO OBTAIN FORCCT TEST ON SIGN OF ZSUN
ISN 0096 IF(ZSUN .GE.0.D0) FORCCT = -(2.00 - AB1)*DOTPR2**2
ISN 0097 IF(ZSUN .LE.0.D0) FORCCT = (2.00 - AB1)*DOTPR2**2
ISN CC99 RETURN

ISN 0100 C ENTRY CHAN(XLAM)
ISN 0101 ROTAAT= XLAM + W*TEL
ISN 0102 CHECK = .TRUE.
ISN 0103 GO TO 300
ISN 0104 FND

```

LEVEL 2 FEB 67

DS/360 FORTRAN H

DATE 68.067/17.16.46

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,ERCDIC,NOLIST,NODECK,LOAD,MAP,INDEXIT,10

```

ISN 0002      DOUBLE PRECISION FUNCTION CINOK,A,RI          1290
ISN 0003      DOUBLE PRECISION Y0150I,DY0150I,DY1150I,A,B,X,H  1291
ISN 0004      DOUBLE PRECISION Y,DY,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12 1292
ISN 0005      DOUBLE PRECISION 81,B2,B3,84                  1293
ISN 0006      IF (K1) 1000,10,1000                          1294
ISN 0007      IF IR 30,20,30
ISN 0008      J=1
ISN 0009      A1=.3333333333333333
ISN 0010      A2=.16666666666666667
ISN 0011      A3=.41666666666666667
ISN 0012      A4=.66666666666666667
ISN 0013      A5=.833333333333333310-1
ISN 0014      A6=1.*91666666666666667
ISN 0015      A7=1.*3333333333333333
ISN 0016      A8=.41666666666666667
ISN 0017      A9=.37500
ISN 0018      A10=.79166666666666667
ISN 0019      A11=.2083333333333333
ISN 0020      A12=.41666666666666667
ISN 0021      B1=2.*29166666666666667
ISN 0022      B2=2.*4583333333333333
ISN 0023      B3=1.*54166666666666667
ISN 0024      B4=.37500
ISN 0025      RETURN
ISN 0026      30
ISN 0027      X=A
ISN 0028      H=B
ISN 0029      GO TO 1100,150,100,150,100,100,3001,J
ISN 0030      CIND=X+H
ISN 0031      GO TO 200
ISN 0032      CIND=X
ISN 0033      J=J+1
ISN 0034      I=1
ISN 0035      RETURN
ISN 0036      CIND=X+H
ISN 0037      GO TO 250
ISN 0038      Y=A
ISN 0039      DY=B
ISN 0040      GO TO 1200,1300,1400,1500,1600,1700,1800,19001,J
ISN 0041      Y0111=Y
ISN 0042      DY0111=DY
ISN 0043      GO TO 3000
ISN 0044      CIND=Y0111+H*.5 DO *(DY+DY0111)
ISN 0045      DY1111=DY
ISN 0046      GO TO 3000
ISN 0047      Y0111=Y0111+H*I*DY*A1+DY1111*A2+.5D0*DY0111
ISN 0048      CIND=Y0111
ISN 0049      GO TO 3000
ISN 0050      CIND=Y0111+H*I*DY*1.5D0*DY0111

```

```

ISN 0051          DY1(I)=DY
ISN 0052          GO TO 3000
ISN 0053          Y0(I)=Y0(I)+H*(A3*DY+A4*DY1(I)-A5*DY0(I))
ISN 0054          CIND=Y0(I)
ISN 0055          GO TO 2000
ISN 0056          CIND=Y0(I)+H*(A6*DY-A7*DY1(I)+A8*)Y0(I)
ISN 0057          GO TO 2200
ISN 0058          CIND=Y0(I)+H*(A9*DY+A10*DY2(I)-A11*DY1(I)+A12*DY0(I))
ISN 0059          GO TO 2000
ISN 0060          CIND=Y0(I)+H*(B1*DY-B2+B3*DY1(I)-B4*DY0(I))
ISN 0061          DYC(I)=Y1(I)
ISN 0062          DY1(I)=DY2(I)
ISN 0063          DY2(I)=DY
ISN 0064          IF (I-50) 4000,5000,5000
ISN 0065          I=I+1
ISN 0066          RETURN
ISN 0067          STOP
ISN 0068          END

```

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 68-0677/17-16-51

```

CDMPILER OPTIONS - NAME= MAIN,OPT=00,LINFCNT=50,SNURCE,EBCDIC,NOLIST,NOEDIT,IN
ISN 0002      C DOUBLE PRECISION FUNCTION CORR (TEL)          1357
ISN 0002      C PROGRAM TO CALCULATE THE APPARENT RIGHT ASCENSION OF THE SJN 1358
ISN 0002      C AND THE TIME OF EPHemeris TRANSIT.          1359
ISN 0002      C WRITTEN BY CHAN CRICKER MIT LINCOLN LABORATORY 1360
ISN 0002      C THIS PROGRAM IS GOOD FOR THE YEAR 1967 AND ACCURATE TO ABOUT 1361
ISN 0002      C FEW SECONDS OF ARC ON THE RIGHT ASCENSION AND ABOUT 1.5 SEC 1362
ISN 0002      C ON THE EPHemeris TRANSIT 1363
ISN 0002      C DEFINITION OF SYMBOLS 1364
ISN 0002      C WS = MEAN LONGITUDE OF PERIGEE PAGE 50 AMFR. EPHemeris 1365
ISN 0002      C 3.3929 = DAY THAT PERIGEE OCCURS CALCULATED FROM THE TABLE 1366
ISN 0002      C ON PAGE 50 OF THE EPHemeris 1367
ISN 0002      C 105.726712328 = DAY THAT THE APPARENT SOLAR TIME AND 1368
ISN 0002      C EPHemeris TIME ARE THE SAME = APRIL 15 1369
ISN 0002      C .985600 = THE DAILY MOTION OF MEAN ANOMALY OF THE SUN 1370
ISN 0002      C 23.44358 = INCLINATION OF THE SUN'S ORBIT 1371
ISN 0002      C ESUN= ECCENTRICITY OF THE SUN'S ORBIT 1372
ISN 0002      C DAY = DAY OF THE YEAR 1373
ISN 0002      C RAM = RIGHT ASCENSION OF THE MEAN SUN 1374
ISN 0002      C 23. + 45.6*15./3600. = RIGHT ASCENSION OF THE MEAN SJN IN APRIL 15 1375
ISN 0002      C V= TRUE ANOMALY XM = MEAN ANOMALY 1376
ISN 0002      C                                         1377
ISN 0002      C                                         1378
ISN 0002      C IMPLICIT REAL * 9 (A-H,O-Z) 1379
ISN 0002      C DUMB IS A DUMMY VARIABLE IN COMMAND TO PREVENT AN ERROR MESSAGE 1380
ISN 0002      C COMMON/Z8/XLIM * CONST * XLI MX * CUT * XNUM * FUEL 1381
ISN 0002      C 1      XX   * WS   * COSI   * ESUN   * SINI   * DUMB   * JFK 1382
ISN 0002      C 2      LLL   * M    ; K    ; NUM4   ; 13    ; 15    ; 1383
ISN 0002      C DAY = TEL / 86400.00 1384
ISN 0002      C IF ((13.EQ.7 *OR. 13.EQ. 9) GO TO 110 1385
ISN 0002      C XM = (.985600 * DAY - 3.392301 * XX 1386
ISN 0002      C THE FOLLOWING EXPRESSION FROM MULIT PAGE 171 1387
ISN 0002      C V = XM + 2.00 * ESUN * DSIN(XM) + 5.00 * ESUN * ESUN * 1388
ISN 0002      C DSIN((2.0)*XM) / 4.00 1389
ISN 0002      C U = V + WS 1390
ISN 0002      C X = DCOS(U) 1391
ISN 0002      C Y = DSIN(U) * COSI 1392
ISN 0002      C RA = DATAN2(Y,X) 1393
ISN 0002      C IF (PA) 5.6*6 1394
ISN 0002      C RA = RA + 6.2*3.18531DD 1395
ISN 0016      C RAM = (.985600*(DAY-105.72671200)+23.0045.600*15.00/3600.00)*XX 1396
ISN 0017      C IF (RAM) 7.8,3 1397
ISN 0018      C RAM = RAM + 6.2*3.18531DD 1398
ISN 0019      C RAM = FESMOD(RAM,6.28315531DD) 1399
ISN 0020      C DIFF = RA - RAM 1400
ISN 0021      C IF (DABS(DIFF) - 3.00) 11,11,12 1401
ISN 0022      C DIFF = DIFF + 6.28318531DD 1402
ISN 0022      C NOTE 13 DOES NOT ATTAIN THE VALUE 4 IN THIS GO TO 1403
ISN 0022      C                                         NO Error 1404
ISN 0022      C                                         13,25

```

```

C      I3      = 3,6      RANDOM ERROR          1406
C      11   GO TO 120,21,22,20,21,22),I3      1407
C      20   CORR = OIFF      1408
C      RETURN      1409
C      22   DIFF = OIFF + (RAN(D) - .5) * XX      1410
C      21   DAY = ES MOD (DAY,365.DD)      1411
C      WHEN I3 = 5 OR 6      DO A STEP FUNCTION CORRECTION      1412
C      IF (I3 * EQ. 5 * OR. 13 * EQ. 6)      1413
C      IF (DAY-45.DD)100,100,101      1414
C      IFUF = .0800 * (DAY + 2.25DD) * XX      1415
C      GO TO 31      1416
C      101  IF (DAY - 129.DD) 102,102,103      1417
C      102  VALUE = -.0584.DD * (DAY-111.DD) * XX      1418
C      GO TO 31      1419
C      103  IF (DAY - 211.DD) 104,104,105      1420
C      104  VALUE = .0330.DD * (DAY - 162.DD) * XX      1421
C      GO TO 31      1422
C      105  IF (DAY - 310.DD) 106,106,107      1423
C      106  VALUE = -D.0584.DD*(DAY-240.DD)*XX      1424
C      GO TU 31      1425
C      107  VALUE = .0800 * (DAY-363.DD) * XX      1426
C      GO TO 31      1427
C      30   CALL ALVISE(DAY,VALUE)      1428
C      31   CORR = DIFF - VALUE      1429
C      RETURN      1430
C      11D  DAY = D MOD (DAY, 365.DD)      1431
C      CALL CURT (DAY,VALUE)      1432
C      IF (I3 * EQ. 8)  CORR = VALUE * XX + (RAN(D) - .5)*XX      1433
C      IF (I3 * EQ. 7)  CORR = VALUE * XX      1434
C      RETURN      1435
C      END      1436
ISN 0054      1437

```

LEVEL 2 FEB 67

DATE 68.067/17.16.56

OS/360 FORTRAN H

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINFCNT=50,SOURCE,ERCDIC,VLIST,NDFFCK,LNAM,MAP,NDEFIT,IN
LSN 0002 DOUBLE PRECISION FUNCTION DPNV(Y,DY) 1438
LSN 0003 DOUBLE PRECISION Y,DY 1439
LSN 0004 DOUBLE PRECISION CIND 1440
LSN 0005 DPNV=CIND(1,Y,DY) 1441
LSN 0006 RETURN 1442
LSN 0007 END 1443

LEVEL 2 FOR 67

OS/360 FORTRAN H

DATE 59.06717.16.59

```
COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCF,ERCDTC,NOLIST,NOECK,LNAD,MAP,NODEFT,IN
      DOUBLE PRECISION FUNCTION FINDV(X,H)
      DOUBLE PRECISION X,H
      DOUBLE PRECISION CIND
      FINDV=CIND(0,X,H)
      RETURN
END
ISN 0002
ISN 0003
ISN 0004
ISN 0005
ISN 0006
ISN 0007
```

LEVEL 2 FEB 67

OS/360 FORTRAN H DATE 68.067/17.17.02

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50, SOURCEF,EBCDIC,OBJLIST,LOAD,MAP+NODET,IO
ISN 0002 C DOUBLE PRECISION FUNCTION FLAM (XLAM) 1450
C THIS FUNCTION COMPUTES A FORCE WHEN THE INPUT PARAMETER IS JUST SIDE OF A SPECIFIED BAND. 1451
C
C THE FORCE IS LEFT ON FOR M+1 TIME INTERVALS --- THESE ARE DETERMINED BY THE MAIN PROGRAM AND SET IN COMMON 1452
C
C THE IMPULSES ARE ALLOWED ON ONE DAY INTERVALS, BUT DONG STARTED 1453
C
C NUM IMPULSES WILL BE GIVEN. 1454
C
C K IS A GATE WHICH ALLOWS PASSAGE INTO THE ROUTINE. IT IS 1455
C SET POSITIVE, BUT CHANGED TO NEGATIVE ONCE PER DAY. 1456
C IF IMPULSES ARE BEING GIVEN IT IS LEFT NEGATIVE UNTIL 1457
C ALL HAVE BEEN COMPLETED. 1458
C
C XLAM CURRENT POSITION 1459
C
C XLIM BOTTOM OF THE BAND 1460
C XLIMX TOP OF THE BAND 1461
C
C THUS, XLAM-XLIMX-GT. 0 ---XLAM IS ABOVE THE BAND 1462
C AND XLAM-XLIMX-LT. 0 ---XLAM IS BELOW THE BAND 1463
C
C LL IS CONTROLLED BY THE FORCE AND BY THE POSITION RELATIVE TO 1464
C THE BAND. IF LL IS NEGATIVE WF ARE INSIDE THE BAND OR TH 1465
C FORCF HAS BEEN COMPLETED. 1466
C
C M COUNTS THE NUMBER OF IMPULSES BEING ISSUED 1467
C
C IMPLICIT REAL * 8 (A-H,O-Z) 1468
C
C LOGICAL NEWFR 1469
C COMMON/FSL/NEWFR 1470
C COMMON/DOVE/XLIM 1471
C
C COMMON/ZR/XLIM * CONST * XLIMX * CUT * XNUM * FUEL * 1472
C 1 WS * COSI * ESUN * SINI * TEL * JFK * 1473
C 2 LLL * M * K * NUM * 13 * 15 * 1474
C
C ISN 0008 C COMMON/PARAMS/FITHRD,CONSTI, XLIMX, XDISP, XMASS * CUTBY * 1475
C 1 E * XI * XLAMO * XLAMD, TEND * ALONGR, DATE, 1476
C 2 FORCC,FORLM, AI * A? * H * 11 * 17, 1477
C 3 I3T * I4 * I5T * I6 * I7 * NUMIN 1478
C
C COMMON/SUPSTF/ ARI, AB, OFFSET, Iq 1479

```

C
C      WHEN I3 = 9 SAIL IS USFD. FORCE IS READ IN FROM SETUPS AS CONST
C      XLAM IS READ IN FROM SETUPS.
C      IF (I3,FG,9) GO TO 6
ISN 0010      C
C      IF (I3,FG,10) GO TO 3
ISN 0012      C
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      TEST TO SEE IF THE GATE IS OPEN
ISN 0014      C      5      IF(K) 8,8,9
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      THE GATE IS CLOSED
C      9      FLAM = 0.00
C      IF (I8,NE,0) GO TO 50
C      FORCE= 0.00
C      RETURN
ISN 0015      C
ISN 0016      C
ISN 0018      C
ISN 0019      C
ISN 0020      C
ISN 0021      C
ISN 0022      C
ISN 0023      C
ISN 0024      C
ISN 0025      C
ISN 0026      C      3      FLAM = CONST
ISN 0027      C      ABL = A1
ISN 0028      C      AB2 = A2
ISN 0029      C      IF (I6,NF,2) GO TO 12
ISN 0031      C      CALL TACK (XLAM)
ISN 0032      C      RETURN
ISN 0033      C      12      CALL CHAN (XLAM)
ISN 0034      C      RETURN
ISN 0035      C      COMPUTE THE FORCE TO BE APPLIED - INCLUDE THE CUT-DOWN.
ISN 0037      C      IF(M,NE,0) GO TO 4
ISN 0038      C      CON = CONST
ISN 0040      C      IF(I5,FG,1) GO TO 21
ISN 0041      C      42      CONTINUE
ISN 0042      C      IF(LL,GT,0) CON = CUT*CON
ISN 0043      C      COMPUTE THE CORRECTION FACTOR
ISN 0044      C      NOTE. IF I3,FG,4 THERE IS NO CORRECTION
ISN 0045      C      ?1 X = 0.00
ISN 0046      C
ISN 0047      C
ISN 0048      C

```

```

C      MODIFY THE INPUT AS A FUNCTION OF THE CORRECTION
C      AND DETERMINE WHERE IT LIFS RELATIVE TO THE BAND
C
ISN 0044          1549
ISN 0046          1550
C      IF(I3 .NE. 4) XTEMP = XLAM-X
C
ISN 0047          1551
ISN 0049          1552
ISN 0051          1553
C      IF WE ARE USING THE OLD FORCE CALCULATIONS JUST CONTINUE. IF NOT,
C      GO TO SUBROUTINE NEWFOR
C
ISN 0053          1554
ISN 0054          1555
C      K=1
C      GO TO 9
C
ISN 0055          1556
ISN 0056          1557
ISN 0058          1558
ISN 0059          1559
ISN 0060          1560
C      WE ARE OUTSIDE AND ABOVE, THRUSTING WITH THE ORBIT
C      11  FORCE=CON
C      IF (I8 .NE. 0) I8 = 1
C      OFFSET = -90.000 * XX
C      AB1 = A2
C      AB2 = A1
C      WE ARE OUTSIDE AND BELOW, THRUSTING AGAINST THE ORBIT
C      WE ARE OUTSIDE AND ABOVE, THRUSTING AGAINST THE ORBIT
C      GU TO 4
C
ISN 0061          1561
C      IF WE REACH HERE IT MEANS WE ARE USING THE NEW FRCF CALCULATIONS
C
ISN 0062          1562
ISN 0063          1563
ISN 0064          1564
ISN 0066          1565
ISN 0067          1566
ISN 0068          1567
ISN 0069          1568
C      CALL NEWFOR(XLIM0,XTEMP,CON)
C      300  CALL NEWFOR(XLIM0,XTEMP,CON)
C      1  FRCF = CON
C      IF (I8 .NE. 0) I8 = 2
C      OFFSFT= 0.00
C      A31 = A1
C      A32 = A2
C      M = M + 1
C
ISN 0070          1569
ISN 0072          1570
ISN 0074          1571
C      IF(I6.EQ.2) CALL TACK(XLAM)
C      43  IF (I6.EQ.1) CALL CHAN(XLAM)
C      IF(M .GT. NUM) GO TO 7
C
ISN 0076          1572
ISN 0077          1573
ISN 0079          1574
ISN 0080          1575
C      WE ARE STILL GIVING THE IMPULSES
C
ISN 0081          1576
C      FLAM = FORCE
C      IF (I6.EQ.1) FLAM = OABS (FORCE)
C      JFK = JFK + 1
C      RETURN
C
ISN 0082          1577
C      PUBLISH THE REPORT OF THE FUEL CONSUMED
C      WE ARE OUTSIDE AND BELOW
C
ISN 0083          1578
C      FUEL=FUEL+DABS(CON*NUM)
C
ISN 0084          1579
ISN 0085          1580
ISN 0086          1581
ISN 0087          1582
ISN 0088          1583
ISN 0089          1584
ISN 0090          1585
ISN 0091          1586
ISN 0092          1587
ISN 0093          1588
ISN 0094          1589
ISN 0095          1590
ISN 0096          1591
ISN 0097          1592
ISN 0098          1593
ISN 0099          1594
ISN 0100          1595
ISN 0101          1596
ISN 0102          1597
ISN 0103          1598

```

```

ISN 0082      IF (I6 .NE. 0) FUEL = 0.00
ISN 0084      C(0) = -X/XX
C
ISN 0085      WRITE(6,10) JFK,CORD,FORCE,FUEL
ISN C086      10 FORMAT(1X,I10,1FK,1D=*,1PE15.6,1PE15.6,
C                                         1'KG')
C
C           SET THE GATE TO CLOSED
C
ISN 0087      K     = 1
ISN 0088      M     = 0
ISN 0089      GO TO 9
ISN C090      END

```

LEVEL 2 FEB 67

05/360 FORTRAN H

DATAF 68.067/17.17.09

```
COMPILER OPTIONS - NAME= MAIN,OPT=00,LINFCNT=50,SOURCE,FRCDIC,NOLIST,NODECK,UDAN,MAP,NDEFIT,IN
```

```
ISN 0002 DOUBLE PRECISION FUNCTION FSQUD FX1,X?}
ISN 0003 IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004 IX=IDINT(X1/X2)
ISN 0005 FSMOD=X1-IX*X2
ISN 0006 KFTURN
ISN 0007 END
```

REFERENCES

1. M. J. Neufeld and B. M. Anzel, "Synchronous Satellite Stationkeeping," AIAA Paper No. 66-304.
2. J. H. Molitor and M. H. Kaplan, "Optimization of Ion Engine Control Systems for Synchronous Satellites," American Institute of Aeronautics and Astronautics, Summer Meeting, Paper No. 63-273.
3. R. R. Allan and G. E. Cook, "The Long-Period Motion of the Plane of a Distant Circular Orbit," Proc. Royal Society 280, p. 97 (7 July 1964).
4. W. L. Black, M. C. Crocker, and E. H. Swenson, "Stationkeeping a 24-Hour Satellite Using Solar Radiation Pressure," to be published in Journal of Spacecraft and Rockets.
5. L. Blitzer, E. M. Boughton, G. Kang and R. M. Page, "Effect of Ellipticity of the Equator on a 24-Hour Nearly Circular Satellite Orbits," J. of Geophysical Research 67, 329 (1962).
6. C. A. Wagner, "The Drift of a 24-Hour Equatorial Satellite Due To An Earth Gravity Field Through 4th Order," NASA TN D-3317 (1966).
7. C. A. Wagner, "Determination of the Ellipticity of the Earth's Equator from Observations on the Drift of the Syncom II Satellite," NASA TN D-2759 (May 1965).
8. Private communication.
9. Private communication.
10. B. J. Moriarty, "Position Error in Station-Keeping Satellite," Technical Note 1966-21, Lincoln Laboratory, M. I. T. (1 April 1966), DDC 633034.
11. Private communication.

ACKNOWLEDGMENT

Marie Roberts, Eileen Leiber, Chester Small, Harold Feinleib, and William Roberts have written different parts of the SSS program. Walter Morrow, Alvise Braga-Illa, William Black, and Roger Brockett have made some of the suggestions for automatic station-keeping that have been incorporated into the program.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)