CR 86 312

SCR 290 II

FINAL REPORT
FOR THE STUDY OF
SIMPLIFIED NAVIGATION
and
GUIDANCE SCHEMES

CONTRACT NAS 12-40

VOLUME II - SIMULATION DOCUMENTATION

United Aircraft Corporate Systems Center

DIVISION OF UNITED AIRCRAFT CORPORATION

FARMINGTON, CONNECTICUT 06032

SCR 290-II

FINAL REPORT

STUDY OF SIMPLIFIED NAVIGATION AND GUIDANCE SCHEMES

VOLUME II - SIMULATION DOCUMENTATION

Reported by: S.M. Vossler
S. M. Vossler

Approved by:
R. M. Grose
Chief, Systems Studies

Manoni
L. R. Manoni
Manager, Advanced Programs

UNITED AIRCRAFT
CORPORATE SYSTEMS CENTER
Division of United Aircraft Corporation
Farmington, Connecticut
October 31, 1966

FOREWORD

The work described in this report was performed by United Aircraft Corporate Systems Center for NASA Electronics Research Center as partial fulfillment of Contract No. NAS 12-40.

This volume, SCR 290, Volume II - Simulation Documentation, delineates the Compiler Programs for Interplanetary Space Navigation developed under this contract and provides the user with sufficient information for operation of the programs. Copies of each program and a sample test case for each program have been transmitted to NASA/ERC under the terms of this contract.

TABLE OF CONTENTS

Section **Page**

LIST OF TABLES

LIST OF ILLUSTRATIONS

I. INTRODUCTION

The two major objectives of the Study of Simplified Navigation and Guidance Schemes (NASA Contract 12-40) are 1) to define the navigation and guidance requirements, as applied to a small energetic kick stage, for an interplanetary mission in which the gravitational field of Jupiter is utilized to deflect the trajectory of a spacecraft from an orbit initially in the ecliptic plane to one which is perpendicular to the ecliptic plane, and 2) to extend the UACSC-conceived orbital navigation concept to interplanetary trajectories. To attain these objectives it was necessary to develop three major computer simulations in the general areas of trajectory analysis and navigation system performance. The purpose of Volume II is to document the simulations produced and to provide a complete guide for their use. All three computer programs are written in FORTRAN IV for the IBM 7094 Model II computer and may be described functionally as follows:

The N-Body Truth Model Program (N-B0DY) is a semi-double precision, Cowell Method trajectory program intended primarily for the simulation of interplanetary trajectories which pass in the neighborhood of Jupiter.

The Linear Optimum Filter Navigation Program (L0F) is a single precision program designed for the simulation of the performance of a linear optimum filter navigation technique over typical Jupiter swing-by interplanetary trajectories. The linear optimum filter scheme combines optical measurements of the line-of-sight vectors to nearby celestial bodies with a linear filter data processing technique using statistically optimized gains to bound position and velocity errors during the mission.

The Interplanetary Space Navigation Program (ISN) is a single precision program designed for the performance evaluation of the UACSC-conceived orbital navigation concept extended to interplanetary operation and applied to typical Jupiter swing-by trajectories. The ISN concept is a simple nonlinear technique which utilizes the filtered difference between measured and navigated estimates of the line-of-sight vector to nearby celestial bodies as feedback quantities in the navigation loops.

Because of functional similarities, the three computer programs described herein have many areas of common design. For example, each program has identical means for input and problem termination. In addition, all three programs use the point mass, Cartesian coordinate equations of motion to generate a reference "truth" trajectory. Each such trajectory is governed by a common gravitational force model. In order to provide a direct comparison between the results from the two navigation programs, both programs use the same error source model as input and are functionally designed to be equivalent. The major program characteristics are described in more detail in the following paragraphs. In order to avoid duplication of descriptive material, those characteristics common to two or more programs are discussed first.

A. Characteristics Common to All Three Programs

1. Each program contains logic which automatically adjusts the origin of the major coordinate frame in order to maintain computational accuracy when the vehicle passes through a planetary sphere of influence. The list of eligible coordinate centers is limited to the Sun and Jupiter for the two navigation programs and to the Earth, Sun and Jupiter for the N-BODY program.

2. The equations of motion contained in all three programs take into account the Newtonian gravitational fields of a maximum of seven celestial bodies chosen from a master list of ten celestial bodies. The particular set included at any one time is a function of the vehicle position relative to the positions of the planets, i.e., upon the particular origin of coordinates in use at the time. The group of planets associated with each coordinate center are given in Table 1-1. As indicated, the masses of Jupiter's moons are added to Jupiter's mass for those portions of the trajectory within the Earth's and Sun's spheres of influence. This is done to improve consistency in the total mass of the planetary system from one trajectory phase to another.

3. All planetary orbits are considered to be ellipses with constant orbital elements. Because the three programs are intended primarily as navigation and guidance system evaluation tools, this level of sophistication is sufficient to provide the desired level of comparison between truth and navigated trajectory estimates.

4. Each program utilizes a system of planet index codes as a means of referencing the celestial bodies. This system consists of a set of integers each corresponding to a given celestial body. The particular code for each body in the master list is given in Table 1-1.

5. All linear input-output variables are expressed in either kilometers or astronomical units (AU). In general, the metric system is used for those portions of the trajectory within the Earth's or Jupiter's spheres of influence and astronomical units are used in the Sun's sphere of influence. Velocities are always expressed in meters per second. Angular input-output variables are specified in degrees except in connection with sensor measurements where the expressed units are arc seconds. Internally all three programs use astronomical units, days, and radians as the units for length, time, and angular measurement, respectively.

TABLE I-I

PLANETARY MODEL SPECIFICATION

x included — not included

Planet Index Code	Celestial Body	Center of Coordinates		
		Earth*	Sun	Jupiter
1	Earth	x	x	x
2	Jupiter	-	-	x
2	Jupiter $+\sum_{i=1}^{4} J_{moon_i}$	x	x	-
3	Sun	x	x	x
4	Moon	x	x	-
5	Venus	x	x	-
6	Mars	x	x	-
7	Jupiter Moon I	-	-	x
8	Jupiter Moon II	-	-	x
9	Jupiter Moon III	-	-	x
10	Jupiter Moon IV	-	-	x

* N-B0DY Program only

B. Characteristics Common to the Navigation Programs

1. The two navigation programs are designed to simulate three possible sources of error in the generation of a navigated trajectory. The three pertinent sources of error are: 1) the use of an unrealistic force model caused by celestial body exclusion in the navigation system equations of motion, 2) errors in knowledge of the initial state, and 3) a composite measurement error applied at each measurement during a navigated trajectory. The measurement errors, expressed in terms of small changes in the inertial right ascension and declination of the observed planet, are intended to reflect the combined effects of sensor alignment errors, optical measurement errors, and stellar inertial self-calibration errors.

2. The user has the capability, via input, of specifying the number of celestial bodies to be considered in the navigated equations of motion. In this fashion the effect of eliminating certain force fields from the navigation system force model may be evaluated. The programs are set up such that the specification of less than the maximum number of planets possible will automatically eliminate the least significant celestial bodies from consideration.

3. Both navigation programs have the capability of generating a number of trajectories within the framework of one run for the convenience in applying a statistical analysis to the results. One run consists of a "truth" trajectory and possibly one or more navigated trajectories each starting from a given set of initial conditions. The navigated trajectories differ from the "truth" trajectory and from each other by an assumed set of errors in the knowledge of initial state and by the measurement errors applied at each input-specified measurement time. The errors in knowledge of initial state and the measurement errors are obtained from a random number generator subject to input-specified mean and standard deviation constraints.

4. A numerical data processing subroutine is included in each program for the purpose of calculating and printing the mean, standard deviation, and second moment about zero for each set of position and velocity errors at a given measurement time during the mission. The subroutine utilizes the data generated during the series of navigated trajectories in that particular run.

5. A maximum of 100 navigation measurements can be made during any one trajectory computation. These are specified via input by the selection of the desired measurement time and the planet index code representing the planet on which the measurement is to be made.

C. Specific Program Characteristics

1. N-Body Truth Model Program

a. The user has a choice of two numerical integration techniques:

   (1) Runge-Kutta fourth order with constant step size

   (2) Adams four-point with variable step size and automatic error control.

b. The initial trajectory conditions may be specified in either of two ways:

   (1) Cartesian coordinates

   (2) conic section orbital elements.

c. A maximum of four thrust maneuvers can be made during any one trajectory. The maneuvers are approximated by impulsive velocity additions and are defined by specifying the time at which the maneuver is to take place and the three components of impulsive velocity to be added.

d. In recognition of the varying integration accuracy requirements over the totality of one trajectory, a provision has been made for the input specification of six different integration step sizes. Three sizes may be specified for that part of the trajectory within the Earth's sphere of influence and two for that part within Jupiter's sphere of influence. Vehicle distance from the central body is the factor governing which is used at any given time. One step size is provided for that part of the trajectory within the Sun's sphere of influence. In addition, the user may specify the print frequency corresponding to each integration step size. The print frequency is herein defined to be the desired time interval of print expressed in terms of integral multiples of the integration step size minus one. In other words, if f is the print frequency, the program will print every (f+1) times the integration step size for that phase of the trajectory.

e. A maximum of twenty observation or measurement times may be specified by the user. At the specified times the program performs a standard print (see Section IV-A) and continues. The logic for accomplishing this task was originally included to allow the N-BODY program to function as a truth model base for the navigation system simulations.

Subsequent analysis of the navigation problem, however, indicated that a separate truth model program would be cumbersome, and that function was therefore incorporated into each navigation simulation separately. As a result, the option of specifying observation times will probably not be needed in the N-BØDY program for most applications. The logic, if not used, does not affect the execution time in any way. It is noted that the logic used in the measurement time calculation is shared with the logic used to simulate thrust maneuvers (see Section I. C. 1. c.) and cannot be eliminated without changing that function.

f. The N-BØDY program contains a built-in capability of generating a numerically integrated two-body trajectory for checkout purposes. Setting the input variable TEST equal to unity zeros out the masses of all celestial bodies except the initial dominant central body. Care must be taken, however, not to exceed the defined boundaries of the pertinent planetary sphere of influence.

2. Linear Optimum Filter Navigation Program

a. All navigation gains are statistically optimized to minimize the effects of measurement errors in a least squared sense and are generated within the navigation computation itself. Measurement data is fed back into the navigation loops only at the designated measurement times.

b. All navigation calculations are carried out using rectangular coordinates. The state vector is composed of three position and three velocity components.

c. A fourth order Runge-Kutta integration scheme is used to integrate the six state variables and a second order Runge-Kutta scheme is used to propagate the transition matrix elements.

3. Interplanetary Space Navigation Program

a. All navigation gains are assumed to be constants with respect to time. Measurement data is fed back into the navigation loops on continuous basis through a first order lag filter.

b. The navigation scheme is mechanized in terms of a set of orbit-referenced spherical coordinates. The integration of the state variables is performed in that system. However, the linear gravitational accelerations used in the equations of motion are generated first in rectangular components and then transformed to the corresponding spherical coordinate values. The definition of the vehicle state is in terms of the downrange angle, crossrange angle, the angular rates in the downrange and crossrange directions, the rate of change of the distance to the major attracting body, and the angular momentum per unit vehicle mass about that same body. Internally computed corrections to the angular momentum term are used to compensate in the navigation equations for the perturbations of other planetary bodies.

c. All program integrations, including the truth and navigated equations of motion and the navigation system filter equations, are performed using a fourth order Runge-Kutta integration scheme.

II. COORDINATE SYSTEMS, ASTROPHYSICAL CONSTANTS, AND CONVERSION FACTORS

A. Coordinate Systems

All three programs utilize the same basic system of inertial Cartesian coordinate frames. The coordinate frame in use at any given trajectory time is a function of the computed vehicle position relative to the existing planetary configuration at that time, i.e., the particular sphere of influence through which the vehicle is passing. Thus, a planetocentric, equatorial reference system is used whenever the vehicle passes sufficiently close to an eligible planetary coordinate center; otherwise a heliocentric, ecliptic system is employed. The heliocentric, ecliptic system (X_I, Y_I, and Z_I) is defined such that the X_I axis lies in the plane of the ecliptic and points in the direction of the Vernal Equinox on the Julian day of epoch 2439200.5. The Y_I axis is perpendicular to the X_I axis and lies in the plane of the ecliptic. The Z_I axis is orthogonal to the other two. All planetary orbital elements are defined relative to this system except those for Jupiter's moons which are defined relative to Jupiter's orbital plane. In general, a planetary equatorial system is defined such that the X_c axis is located along the intersection of the planet's equator and its orbital plane, the Y_c axis is perpendicular to the X_c axis and lies in the planet's equatorial plane, and the Z_c axis is orthogonal to both the X_c and Y_c axes. The quantities needed to specify this system are 1) i', the inclination of the planet's equator with respect to its orbit plane, and 2) Ω', the angle measured in the plane of the planet's orbit from the planetary ascending node to the intersection of the planet's equator with its orbit plane. Figure II-1 shows a typical planetary system as used herein.

In addition to the primary Cartesian reference frames, the ISN program also utilizes a secondary spherical coordinate frame in the mechanization of the navigation loops. This system is illustrated in Figure II-2. The nominal vehicle orbital plane is defined by using the center of coordinates at some given time. At the specified time, the reference orbital plane is equated to the vehicle's instantaneous orbital plane. The terms used to specify the reference orbit plane are 1) Ω_v, the orbital argument of nodes relative to the pertinent coordinate system, and 2) i_v, the vehicle's orbital inclination relative to the reference plane of the coordinate system in us

CENTER OF COORDINATES	Ω' RAD	i' RAD
EARTH	0	-.40926918
JUPITER	3.7751466	.05436699

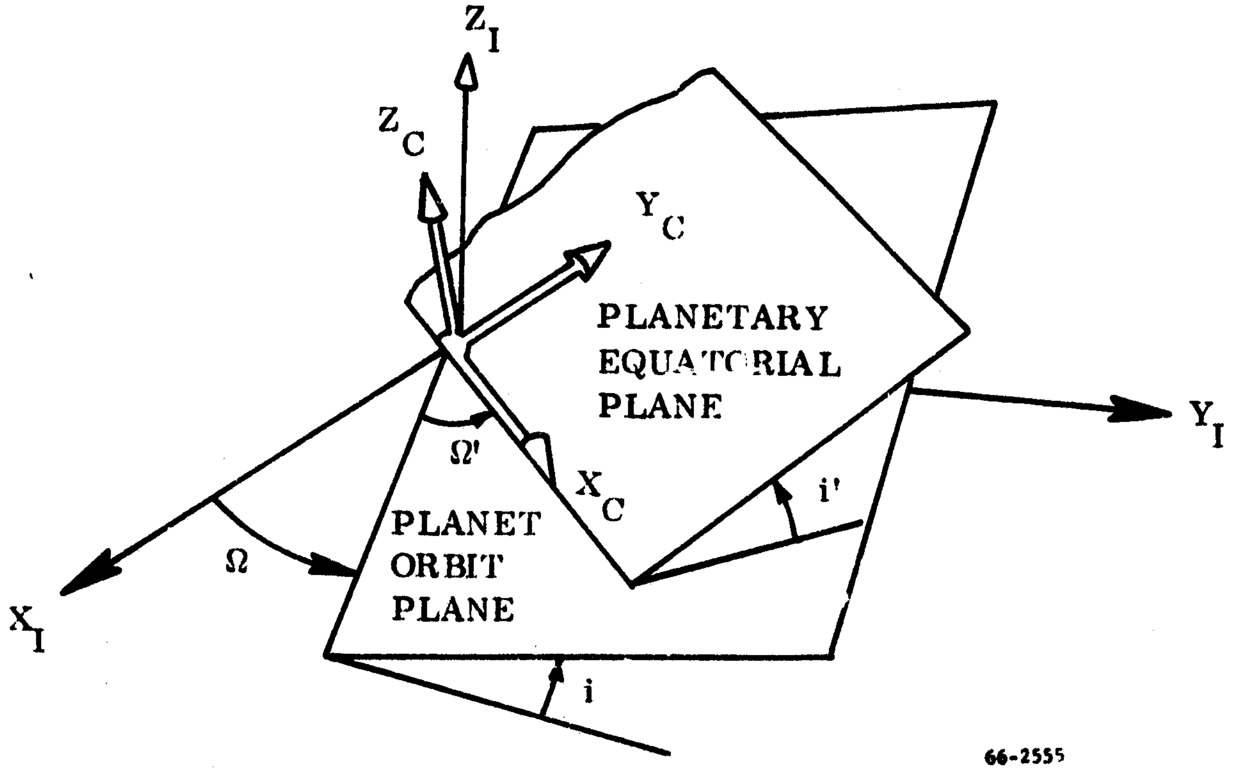


Figure II-1 Typical Planetary Equatorial Axis System

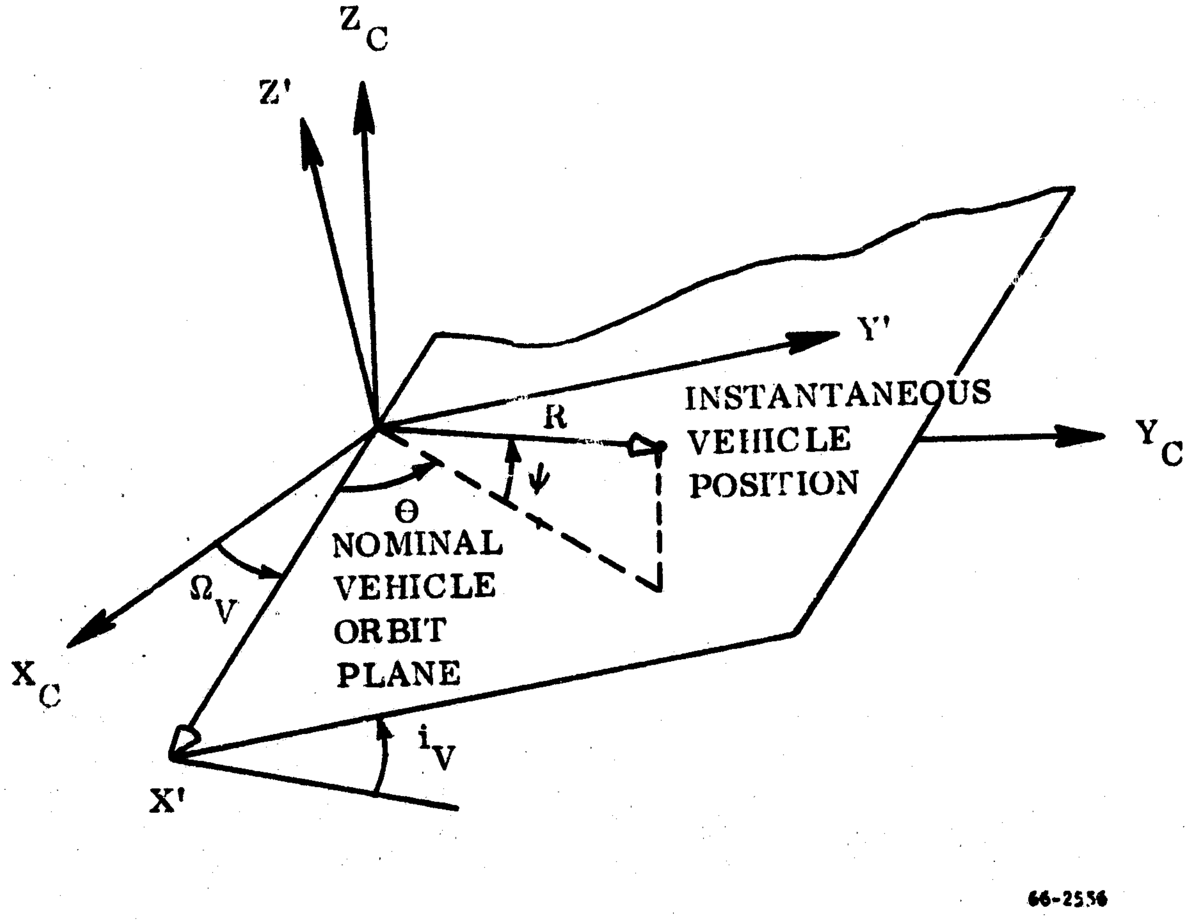


Figure II-2 Spherical Coordinate Frame Used in ISN Program

B. Astrophysical Constants

As mentioned in Section I.A.3, all planetary orbits are assumed to be ellipses with constant orbital elements. The numerical values for the orbital constants are essentially those corresponding to an epoch of March 16, 1966 (Julian date: 2439200.5) as given in Reference 1. Jupiter's four largest moons are assumed to be in circular equatorial orbits about Jupiter with orbital radii as listed in Reference 2. The mean longitudes at epoch for the four moons are derived from data given in Reference 1. Only the Earth and Jupiter are considered to be oblate spheroids insofar as the gravitational potential is concerned. All other celestial bodies in the master list of ten are assumed to have Newtonian force fields. The equations of motion include (where applicable) terms involving the second, third and fourth zonal harmonics for each of the oblate planets when it is the center of coordinates. Values for the harmonic coefficients were obtained from References 3 and 4. The astrophysical constants assumed in the three programs described herein are summarized in Table II-I.

C. Conversion Factors

The numerical values for all conversion factors used in the subject programs together with the corresponding program symbols are listed in Table II-II. All reverse transformations are obtained by dividing by the appropriate conversion factor.

TABLE II-II

PROGRAM CONVERSION FACTORS

	To Convert From	To	Multiply By	Program Symbol	Reference
Length	km	a.u.	6.6845813×10^{-9}	CKMAU	5
	a.u.	nm	8.0725861×10^{7}	AUNM	6
	a.u.	ft	$4.90498406 \times 10^{11}$	AUFT	6
	nm	ft	6076.1	XNMFT	6
Velocity	m/s	AU/day	5.7754783×10^{-7}	CMSAUD	5
	fps	AU/day	1.7614736×10^{-7}	FPSAUD	6
Angular Measurement	deg	radian	.01745329	DEGRAD	5
	arc sec	radian	4.8481368×10^{-6}	SECRAD	5
Time	days	sec	86400.	DAYSEC	5

TABLE II-I

ASTROPHYSICAL CONSTANTS

Body \ Constant	μ . . .A U ^{3}/day^2	a a.u.	e	i* rad	Ω rad	$\tilde{\omega}$ rad	Mean Longitude at epoch** rad	Mean Motion rad/day	Equatorial Radius nm
Sun	2.960075×10^{-4}	0	0	0	0	0		0	376100.
Earth	8.904704×10^{-10}	1	.0167272	0	0	1.6256378	2.8473681	.01702099	3443.93
Moon	1.092464×10^{-11}	.00257296	.05490	.0888372	1.0250494	2.5780288	4.92689716	.22997084	938.5
Mars	9.585665×10^{-11}	1.523691	.093371	.032286	.86039239	5.8544773	.105607937	.00914610	1844.
Venus	7.243555×10^{-10}	.723332	.006789	.0592417	1.330046	2.2880488	3.5606031	.02796244	3290.
Jupiter	2.826117×10^{-7}	5.202432	.0482227	.022794	1.7472157	.23527209	1.54122499	.00145037	38539.
J Moon I	1.081520×10^{-11}	.0028246	0	0	0	0	2.01672766	3.55155785	764.04
J Moon II	$6.9918943 \times 10^{-12}$	.0044940	0	0	0	0	2.85768529	1.76932274	850.43
J Moon III	$2.3051174 \times 10^{-11}$	.0071687	0	0	0	0	1.15249885	.87820794	1390.39
J Moon IV	1.442090×10^{-11}	.0126104	0	0	0	0	3.22478627	.37640276	1398.49

* **Inclinations listed are with respect to ecliptic plane except those given for Jupiter's moons which are with respect to Jupiter's equatorial plane**

** **epoch = 2439200.5**

Body \ Constant	Polar Radius nm	Flattening	J_2	J_3	J_4	Planetary Sphere of Influence a.u.
Earth	3432.38	.00335233	1.08228×10^{-3}	-2.3×10^{-6}	-2.12×10^{-6}	.0061852
Jupiter	36004.	.06578997	1.48×10^{-2}	0	-3.5×10^{-4}	.32262

$\pi = 3.14159265$ $2\pi = 6.28318530$

III. PROGRAM INPUTS

All inputs for the three computer programs are loaded using the following format on standard IBM 80-column punched cards:

Columns	1 - 2	3 - 5	6 - 20	21 - 35	36 - 50	51 - 65	66 - 80
	IWC	LØC	BUFFER(1)	BUFFER(2)	BUFFER(3)	BUFFER(4)	BUFFER(5)

where	IWC	- number of data words on each card ($1 \leq IWC \leq 5$)
	LØC	- the program location of the first variable listed on each card (all other variables on the card are read into successive locations.)
	BUFFER(1-5)	- floating point data words corresponding to each of the five possible input variables read in on any one card.

The end of a case is signaled to the program by the insertion of a negative word count (IWC) on the last regular input card for that case. The reading of input is terminated by the inclusion of a one card final case containing a negative run number.

Termination Card Format for Each Program:

Deck	Columns 1 - 2	3 - 5	6 - 20
N-BØDY	- 1	69	-1.
LØF	- 1	27	-1.
ISN	- 1	141	-1.

The procedure for performing a series of runs requires that only the data being changed from case to case need be input for all cases subsequent to the first. In addition, input values which are nominally zero do not have to be loaded except when they are loaded directly behind an input variable that is nonzero.

The required input data and corresponding program locations are given in Tables III-I, III-II, and III-III for the N-BØDY, LØF, and ISN programs, respectively.

TABLE III-I
PROGRAM INPUTS FOR THE N-BODY TRUTH MODEL PROGRAM (N-BØDY)

LOCATION	VARIABLE	UNITS	COMMENTS
1	XMTHD	-----	integration scheme indicator = 0. predictor-corrector integration method = 1. Runge-Kutta fourth order integration method
2	VARBH	-----	integration step indicator = 0. variable integration step size = 1. constant integration step size
3	PRIN1A	days	maximum integration step for Earth coordinate center $R < .0007$ A U
4	PRIN1B	days	maximum integration step for Earth coordinate center $.0007 \text{ A U} \leq R < .003$ a.u.
5	PRINT1	days	maximum integration step for Earth coordinate center $.003 \text{ A U} \leq R <$ RSPHE
6	PRINT2	days	maximum integration step for Sun coordinate center
7	PRINT3	days	maximum integration step for Jupiter coordinate center $.03 \text{ A U} < R <$ RSPHJ
8	DELTJ	days	maximum integration step for Jupiter coordinate center $R \leq .03$ A U
9	DATE	Julian days	Julian date at TIME = 0.
10	XN1	-----	initial coordinate system indicator = 1. geocentric, equatorial = 2. jovocentric, equatorial = 3. heliocentric, ecliptic
11	XMØDE	-----	input mode indicator = 1. input is in Cartesian coordinates = -1 input is in orbital elements
12	VX0	m/s	initial Cartesian velocity components in either equatorial or ecliptic system depending on XN1
13	VY0	m/s	
14	VZ0	m/s	
15	XC0	km if XN1 = 1., 2. A U = 3.	initial Cartesian position coordinates in either equatorial or ecliptic system depending on XN1
16	YC0	km if XN1 = 1., 2. A U = 3.	
17	ZC0	km if XN1 = 1., 2. A U = 3.	
Note: the next 6 items are alternatives to items 12-17 depending on XMØDE			
18	AV	A U	initial vehicle orbital semi-major axis
19	EV	-----	initial vehicle orbital eccentricity
20	YI0	deg	initial vehicle orbital inclination
21	ØM0	deg	initial vehicle orbital argument of nodes
22	ARGPR0	deg	initial vehicle orbital argument of pericenter
23	ETA0	deg	initial vehicle orbital true anomaly
24	THRIND	-----	number of thrust maneuvers (≤ 4)
25 ↓ 28	TTHR(1) ↓ TTHR(4)	days ↓	time measured from DATE for the initiation of each thrust maneuver
29 ↓ 32	DELVX(1) ↓ DELVX(4)	m/s ↓	impulsive velocity component along XC axis for each thrust maneuver

TABLE III-I (Continued)

PROGRAM INPUTS FOR THE N-BODY TRUTH MODEL PROGRAM (N-BØDY)

LOCATION	VARIABLE	UNITS	COMMENTS
33 ↓ 36	DELVY(1) ↓ DELVY(4)	m/s ↓	impulsive velocity component along YC axis for each thrust maneuver
37 ↓ 40	DELVZ(1) ↓ DELVZ(4)	m/s ↓	impulsive velocity components along ZC axis for each thrust maneuver
41	ØBSIND	-----	number of observation times ($\leq$ 20)
42 ↓ 66	TØB(1) ↓ TØB(25)	days ↓	table of observation times measured from DATE
67	TEST	-----	special function switch = 0. the program computes an n-body trajectory = 1. the program computes a two-body trajectory
68	ELEPR	-----	vehicle orbital element print switch = 0. orbital elements are printed only at TIME = TMAX = 1. orbital elements are printed every "standard" print
69	RUN	-----	case identification number
70	TRAMA	-----	switch controlling transition matrix calculation = 0. transition matrix calculation deleted = 1. transition matrix calculation included
71	TRAPRN	-----	transition matrix print switch = 0. transition matrix prints only at last calculated time = 1. transition matrix prints at every standard print
72	PINT	-----	integration print switch = 1. prints maximum integration step size and number of integration steps at last calculated time = 0. deletes above print
73	PCN1A	-----	print frequency for Earth's sphere of influence $R < .0007$ A U. The program prints every (PCN1A + 1.)*PRIN1A days
74	PCN1B	-----	print frequency for Earth's sphere of influence $.0007$ A U $\leq R < .003$ a.u. The program prints every (PCN1B + 1.)*PRIN1B days
75	PCN1	-----	print frequency for Earth's sphere of influence $.003 \leq R <$ RSPHE. The program prints every (PCN1 + 1.)*PRINT1 days
76	PCN2	-----	print frequency for the Sun's sphere of influence. The program prints every (PCN2 + 1.)*PRINT2 days
77	PCN3	-----	print frequency for Jupiter's sphere of influence $.03 < R <$ RSPHJ. The program prints every (PCN3 + 1.)*PRINT3 days
78	PCN4	-----	print frequency for Jupiter's sphere of influence $R \leq .03$ A U. The program prints every (PCN4 + 1.)*DELTJ days
79	T0	days	initial trajectory time relative to DATE
80	TMAX	days	maximum or final trajectory time relative to DATE

TABLE III-II

PROGRAM INPUTS FOR THE LINEAR OPTIMUM FILTER PROGRAM (LØF)

LOCATION	VARIABLE	UNITS	COMMENTS
1	DATAPR	-----	print control switch = 0. program prints standard output plus optional navigation data for each navigated trajectory = 1. program prints standard output only
2	BLNK	-----	unused variable
3	X0	km if XN1 = 2. A U = 3.	initial true position coordinates
4	Y0	km if XN1 = 2. A U = 3.	
5	Z0	km if XN1 = 2. A U = 3.	
6	VX0	m/s	initial true velocity components
7	VY0	m/s	
8	VZ0	m/s	
9	AAX	km	the statistical mean of the errors in knowledge of initial position
10	AAY	km	
11	AAZ	km	
12	AVX	m/s	the statistical mean of the errors in knowledge of initial velocity
13	AVY	m/s	
14	AVZ	m/s	
15	XN1	-----	initial coordinate system indicator = 2. jovocentric, equatorial = 3. heliocentric, ecliptic
16	XNUM	-----	number of trajectories to be generated in this run
17	XNMAX	-----	number of navigation measurement times per trajectory
18	CN1	-----	noise constant for covariance matrix calculation
19	XKØN	-----	number of integration steps between t_0 and the first measurement time.
20	XIPLAN	-----	number of celestial bodies other than the center of coordinates to be considered in the navigation force model (XIPLAN $\leq$ 6.)
21	SIGMU(1)	km	best estimate of error in knowledge of initial position-also, standard deviation of same quantities
22	SIGMU(2)	km	
23	SIGMU(3)	km	
24	SIGMU(4)	m/s	best estimate of error in knowledge of initial velocity - also, standard deviation of same quantities
25	SIGMU(5)	m/s	
26	SIGMU(6)	m/s	
27	RUN	-----	run identification number
28	XMNA10	arc sec	mean of measurement error in the inertial right ascension of the vehicle-observed planet vector while the vehicle is

TABLE III-II (Continued)
PROGRAM INPUTS FOR THE LINEAR OPTIMUM FILTER PROGRAM (LØF)

LOCATION	VARIABLE	UNITS		COMMENTS
30	XMND10	arc sec	mean	of measurement errors in the inertial declination of the vehicle-observed planet vector while the vehicle is in the Sun's sphere of influence
31	STDD10	arc sec	standard deviation	
32	XMNA20	arc sec	mean	of measurement errors in the inertial right ascension of the vehicle-observed planet vector while the vehicle is in Jupiter's sphere of influence
33	STDA20	arc sec	standard deviation	
34	XMND20	arc sec	mean	of measurement errors in the inertial declination of the vehicle-observed planet vector while the vehicle is in Jupiter's sphere of influence
35	STDD20	arc sec	standard deviation	
36	SIGP0	arc sec		one sigma estimates of measurement errors
37	SIGQ0	arc sec		
38	SIGPQ0	arc sec		cross correlation values for measurement errors
39	SIGQP0	arc sec		
40	DATE	Julian days		Julian date at TIME = 0.
41	T0	days		initial time measured from DATE
42	T1(1)	days		table of navigation measurement times measured from DATE
↓	↓	↓		
141	T1(100)			
142	XLAB(1)	-----		table of planet index codes specifying which body is being observed at each measurement time
↓	↓	↓		
241	XLAB(100)			

TABLE III-III
PROGRAM INPUTS FOR THE INTERPLANETARY SPACE NAVIGATION PROGRAM (ISN)

LOCATION	VARIABLE	UNITS		COMMENTS
1	XX	km if XN1 = 2. A U = 3.		initial true position coordinates
2	YY	km if XN1 = 2. A U = 3.		
3	ZZ	km if XN1 = 2. A U = 3.		
4	VXX	m/s		initial true velocity components
5	VYY	m/s		
6	VZZ	m/s		
7	DELTX	km		statistical mean of the errors in knowledge of initial position
8	DELTY	km		
9	DELTZ	km		
10	DELVX1	m/s		statistical mean of the errors in knowledge of initial velocity
11	DELVY1	m/s		
12	DELVZ1	m/s		
13	T0	days		initial trajectory time measured from DATE
14	XNMAX	-----		maximum number of measurement times
15	XMNA10	arc sec	mean	of measurement errors in inertial right ascension of the vehicle-observed planet vector (coordinate center = Sun)
16	STDA10	arc sec	standard deviation	
17	XMNA20	arc sec	mean	of measurement errors in inertial right ascension of the vehicle observed planet vector (coordinate center = Jupiter)
18	STDA20	arc sec	standard deviation	
19	XMND10	arc sec	mean	of measurement errors in inertial declination of the vehicle-observed planet vector (coordinate center = Sun)
20	STDD10	arc sec	standard deviation	
21	XMND20	arc sec	mean	of measurement errors in inertial declination of the vehicle-observed planet vector (coordinate center = Jupiter)
22	STDD20	arc sec	standard deviation	
23	XKM	A U ^{2}/day		navigation gain, momentum feedback
24	XKI	rad/day^3		navigation gain, $\dot{V}_{\Theta}$ integral feedback
25	XKTHE	rad/day		navigation gain, $\dot{\Theta}$ feedback
26	XKVTHE	rad/day^2		navigation gain, $\dot{V}_{\Theta}$ feedback
27	XKPSI	rad/day		navigation gain, $\dot{\psi}$ feedback
28	XKVPSI	rad/day^2		navigation gain, $\dot{V}_{\psi}$ feedback
29	XK0N	-----		number of integration steps between t_0 and the first measurement time

TABLE III-III (Continued)
PROGRAM INPUTS FOR THE INTERPLANETARY SPACE NAVIGATION PROGRAM (ISN)

LOCATION	VARIABLE	UNIT	COMMENTS
30	XL1	-----	navigation system initializer switch - incidates in which form the best estimates of initial position errors are input = 0. ΔΘ and Δψ (DETHE and DEPSI respectively) are input directly = 1. ΔX, ΔY, and ΔZ (DEX, DEY, and DEZ respectively) are input - ΔΘ and Δψ must be calculated
31	DETHE	deg	best estimate of error in knowledge of initial downrange position
32	DEPSI	deg	best estimate of error in knowledge of initial crossrange position
33	DEX	km	best estimates of errors in knowledge of initial position in Cartesian coordinates
34	DEY	km	
35	DEZ	km	
36	TAU1	days	filter time constants in generation of feedback quantities ΔΘ' and Δψ' respectively
37	TAU2	days	
38	XN1	-----	initial coordinate system indicator = 2. jovocentric, equatorial = 3. heliocentric, ecliptic
39	XIMAX	-----	maximum number of celestial bodies to be considered in navigation force model (XIMAX ≤ 7.)
40	XLMAX	-----	maximum number of trajectories to be generated (including the truth trajectory)
41	T(1)	days	table of navigation measurement times measured from DATE
↓	↓	↓	
140	T(100)		
141	RUN	-----	run identification number
142	DATE	Julian days	Julian date at TIME = 0.
143	TABLE(1)	-----	table of planet index codes signifying which celestial bodies are being observed in the navigation measurements
↓	↓	↓	
242	TABLE (100)		
243	TEST		special function switch = 0. the program computes an n-body trajectory = 1. the program computes a two body trajectory
244	SIGMU(1)	km	standard deviation of the errors in knowledge of initial position
245	SIGMU(2)	km	
246	SIGMU(3)	km	
247	SIGMU(4)	m/s	standard deviation of the errors in knowledge of initial velocity
248	SIGMU(5)	m/s	
249	SIGMU(6)	m/s	
250	DATAPR		print selector switch = 0. program prints standard output plus navigation data for each trajectory = 1. program prints standard output only

IV. PROGRAM OUTPUTS

All outputs for the three programs can be put into one of three classifications: 1) standard - outputs that automatically occur at input-specified times throughout a given program run, 2) optional - outputs (in addition to standard items) that may be selected or deleted by means of input specification, and 3) diagnostic - self-explanatory outputs that automatically occur in the event of certain computational troubles in the given programs.

In the output illustrative material that follows, the program variables that are printed as numerical values are distinguished by the enclosure of the complete variable name in parentheses. Program variable definitions are given in Section VIII.

A. Standard Outputs

1. N-Body Truth Model Program

The standard output format of the N-BØDY program is illustrated in Figure IV-1. Position and velocity data are printed at input-specified multiples of the current integration step size (see Section III) in the format shown in Figure IV-1a. In addition thrust maneuver velocity details are printed at each input-specified maneuver time as presented in Figure IV-1b.

2. Linear Optimum Filter Navigation Program

The standard print for the LØF program occurs in two different phases during one multi-trajectory run. The first print phase occurs at each measurement time during the first trajectory generated in that run (the truth trajectory) and consists of the pertinent linear optimum filter matrices and truth trajectory data as shown in Figure IV-2a. The second phase of output occurs at each measurement time during the last trajectory generated in that run and consists of a statistical summary of the navigation errors occurring in all the navigated trajectories at that measurement time. In particular, the second phase print gives the statistical mean, standard deviation, and second moment about zero for the errors in each component of the navigated position and velocity at a given measurement time (see Figure IV-2b).

3. Interplanetary Space Navigation Program

The standard output for the ISN program occurs in two phases during one multi-trajectory run. The first phase of output (see Figure IV-3a) occurs at each measurement time of the first trajectory generated in that run and consists of pertinent truth trajectory position and velocity data. The second phase of output occurs at each measurement time during the last trajectory generated in that run and consists of a statistical summary of the navigation errors occurring in all the navigated trajectories at that measurement time. The second phase print (see Figure IV-3b) gives the statistical mean, standard deviation, and second moment about zero for the errors in each rectangular coordinate component of navigated position and velocity at each measurement time.

```
ORBITAL CONDITIONS AT TIME = (TIME) DAYS      JULIAN DATE = (DATTE)
REFERENCE BODY = EARTH, SUN or JUPTER  BASIC PLANE = ECLIPTIC or EQUATORIAL

POSITION IN RECTANGULAR COORDINATES WITH RESPECT TO REFERENCE BODY
X = (XCC) {A.U. if Sun coordinate center   Y = (YCC) {A.U.        Z = (ZCC) {A.U.
          {KM if Earth or Jupiter center             {KM                    {KM

VELOCITY IN RECTANGULAR COORDINATES WITH RESPECT TO REFERENCE BODY
VX = (VXX)  M/S                 VY = (VYY)  M/S                 VZ = (VZZ)  M/S

OBSERVATION AND COMMUNICATION DATA
                          DIRECTION COSINES OF CELESTIAL BODY WITH RESPECT TO VEHICLE-ECLIPTIC SYSTEM

 BODY          DISTANCE(KM)                   L            M             N
(LABEL(N1))     (DCØM(N1))               (XL(N1))     (XM(N1))     (XN(N1))
(LABEL(N2))     (DCØM(N2))               (XL(N2))     (XM(N2))     (XN(N2))
(LABEL(N3))     (DCØM(N3))               (XL(N3))     (XM(N3))     (XN(N3))
(LABEL(N4))     (DCØM(N4))               (XL(N4))     (XM(N4))     (XN(N4))
(LABEL(N5))     (DCØM(N5))               (XL(N5))     (XM(N5))     (XN(N5))
(LABEL(N6))     (DCØM(N6))               (XL(N6))     (XM(N6))     (XN(N6))
(LABEL(N7))     (DCØM(N7))               (XL(N7))     (XM(N7))     (XN(N7))  *
 * this line printed only when Jupiter is center of coordinates
```

a. Position and Velocity, Observation and Communication Data

```
THRUST MANEUVER NO. (ITH) AT TIME = (TIME) DAYS

DELTA(VX) = (DELVX(ITH)) M/S    DELTA(VY) = (DELVY(ITH)) M/S    DELTA(VZ) = (DELVZ(ITH)) M/S
```

b. Thrust Maneuver Velocity Details

66-2616

Figure IV-1 N-BØDY Standard Output

```
TIME = (TIME) DAYS

                                   GEOMETRY MATRIX
               X(A.U.)      Y(A.U.)      Z(A.U.)     VX(A.U./DAY)      VY(A.U./DAY)     VZ(A.U./DAY)
ALPHA(RAD)     (HH(1, 1))   (HH(1, 2))   (HH(1, 3))   (HH(1, 4))        (HH(1, 5))       (HH(1, 6))
DELTA(RAD)     (HH(2, 1))   (HH(2, 2))   (HH(2, 3))   (HH(2, 4))        (HH(2, 5))       (HH(2, 6))

                                  COVARIANCE MATRIX
                 X            Y            Z            VX                 VY               VZ
X              (P(1, 1))    (P(1, 2))    (P(1, 3))    (P(1, 4))          (P(1, 5))        (P(1, 6))
Y              (P(2, 1))    (P(2, 2))    (P(2, 3))    (P(2, 4))          (P(2, 5))        (P(2, 6))
Z              (P(3, 1))    (P(3, 2))    (P(3, 3))    (P(3, 4))          (P(3, 5))        (P(3, 6))
VX             (P(4, 1))    (P(4, 2))    (P(4, 3))    (P(4, 4))          (P(4, 5))        (P(4, 6))
VY             (P(5, 1))    (P(5, 2))    (P(5, 3))    (P(5, 4))          (P(5, 5))        (P(5, 6))
VZ             (P(6, 1))    (P(6, 2))    (P(6, 3))    (P(6, 4))          (P(6, 5))        (P(6, 6))

                                  TRANSITION MATRIX
                  X                Y                Z                VX               VY               VZ

X          (PHIMAT(1, 1)) (PHIMAT(1, 2)) (PHIMAT(1, 3)) (PHIMAT(1, 4)) (PHIMAT(1, 5)) (PHIMAT(1, 6))
Y          (PHIMAT(2, 1)) (PHIMAT(2, 2)) (PHIMAT(2, 3)) (PHIMAT(2, 4)) (PHIMAT(2, 5)) (PHIMAT(2, 6))
Z          (PHIMAT(3, 1)) (PHIMAT(3, 2)) (PHIMAT(3, 3)) (PHIMAT(3, 4)) (PHIMAT(3, 5)) (PHIMAT(3, 6))
VX         (PHIMAT(4, 1)) (PHIMAT(4, 2)) (PHIMAT(4, 3)) (PHIMAT(4, 4)) (PHIMAT(4, 5)) (PHIMAT(4, 6))
VY         (PHIMAT(5, 1)) (PHIMAT(5, 2)) (PHIMAT(5, 3)) (PHIMAT(5, 4)) (PHIMAT(5, 5)) (PHIMAT(5, 6))
VZ         (PHIMAT(6, 1)) (PHIMAT(6, 2)) (PHIMAT(6, 3)) (PHIMAT(6, 4)) (PHIMAT(6, 5)) (PHIMAT(6, 6))

DETERMINANT  =  (DD)

TIME  =  (TIME) DAYS      CASE =  (CASE)
                             NAVIGATION PERFORMANCE DATA
              X(KM)          Y(KM)          Z(KM)          VX(M/S)        VY(M/S)        VZ(M/S)
ERR           (0.)           (0.)           (0.)           (0.)           (0.)           (0.)
TRUE          (XT1)          (YT1)          (ZT1)          (VXT1)         (VYT1)         (VZT1)
NAV1          (XC1)          (YC1)          (ZC1)          (VXC1)         (VYC1)         (VZC1)
NAV2          (XC2)          (YC2)          (ZC2)          (VC2)          (VYC2)         (VZC2)
CØRR          (0.)           (0.)           (0)            (0.)           (0.)           (0.)

                             WEIGHTING FUNCTION MATRIX
              X(A.U.)        Y(A.U.)        Z(A.U.)        VX(A.U./DAY) VY(A.U./DAY) VZ(A.U./DAY)
ALPHA(RAD)    (AK(1, 1))     (AK(2, 1))     (AK(3, 1))      (AK(4, 1))     (AK(5, 1))     (AK(6, 1))
DELTA(RAD)    (AK(1, 2))     (AK(2, 2))     (AK(3, 2))      (AK(4, 2))     (AK(5, 2))     (AK(6, 2))
```

a. First Print Phase

```
STATISTICAL DATA AT TIME  = (TIME) DAYS
NUMBER OF CASES  = (XNUM)

                X          Y          Z           VX           VY          VZ
MEAN        (XMEAN)   (YMEAN)   (ZMEAN)   (VXMEAN)   (VYMEAN)   (VZMEAN)
STD DEV     (SIGX)    (SIGY)    (SIGZ)    (SIGVX)    (SIGVY)    (SIGVZ)
STD DEV2    (SIGX)    (SIGY)    (SIGZ)    (SIGVX)    (SIGVY)    (SIGVZ)
```

b. Second Print Phase

Figure IV-2 LØF Standard Output

```
TRUTH TRAJECTORY DATA AT TIME  - (TIME) DAYS
REFERENCE BODY  =  SUN or JUPITER   BASIC PLANE = ECLIPTIC or EQUATORIAL

X(KM)      Y(KM)           Z(KM)     VX(M/S)         VY(M/S)           VZ(M/S)
(XT)        (YT)            (ZT)      (VXT)           (VYT)             (VZT)

R(KM)    THETA(DEG)     PSI(DEG)   VR(M/S)   VTHE(DEG/SEC)   VPSI(DEG/SEC)
 (RT)      (THET)        (PSIT)    (RDTT)       (THEDT)          (PSIDT)
```

a. First Print Phase

```
                  X          Y         Z          VX          VY          VZ
MEAN          (XMEAN)   (YMEAN)   (ZMEAN)   (VXMEAN)   (VYMEAN)   (VZMEAN)
STD DEV       (SIGX)    (SIGY)    (SIGZ)    (SIGVX)    (SIGVY)    (SIGVZ)
STD DEV2      (SIGX)    (SIGY)    (SIGZ)    (SIGVX)    (SIGVY)    (SIGVZ)

                                IV-3b
```

66-2614

b. Second Print Phase

Figure IV-3 ISN Standard Output

B. Optional Outputs

1. N-Body Truth Model Program

As a supplement to the position and velocity data printed as standard output, an equivalent set of orbital elements (see Figure IV-4a) may be obtained by setting the input variable ELEPR equal to unity. The transition matrix elements (see Figure IV-4b) may be obtained every time the program performs a standard print by specifying the variables TRAMA and TRAPRN equal to unity. A zero value for the variable TRAPRN in combination with a unit value for TRAMA gives the transition matrix print only at the final standard print time and a zero value for TRAMA deletes the transition matrix calculation entirely.

The number of integration steps and the maximum integration step size in use during the generation of the trajectory may be obtained just prior to the last standard print by the specification of the input variable PINT equal to unity.

2. Linear Optimum Filter Navigation Program

If the input variable DATAPR is equal to zero, the program prints navigation and truth trajectory data at each measurement time in all the navigated trajectories in that run. This data (see Figure IV-5) consists of: 1) the navigation errors in position and velocity after measurement updating, 2) the true position and velocity components at the given measurement time, 3) the corresponding navigated position and velocity components before measurement updating, 4) the corresponding position and velocity components after measurement updating, and 5) the measurement updating corrections. The above print is deleted by the specification of a nonzero value for DATAPR.

3. Interplanetary Space Navigation Program

The specification of the input variable DATAPR equal to zero results in the program printing navigation data at each measurement time in all the navigated trajectories in that run. This output (see Figure IV-6) consists of: 1) the true components of vehicle position and velocity, 2) the navigated estimates of the position and velocity components, and 3) the navigation errors in the position and velocity components. The data is presented first in terms of rectangular, inertial coordinates and then in terms of orbit-dependent, spherical coordinates. Because the true and navigated reference orbit planes differ due to errors in the knowledge of state at the time of reference plane definition, the orbit-referenced spherical coordinates given for the true and navigated trajectories are not directly comparable. Therefore, no error comparison is presented for the spherical coordinates. The above optional print is deleted by the specification of a nonzero value of DATAPR.

C. Diagnostic Output

All dignostic outputs are shown in Figure IV-7.

```
OSCULATING ORBITAL ELEMENTS                                                  {A.U.
ECCENTRICITY     =    (EP)                 SEMI-MAJOR AXIS       =     (A)    {KM
INCLINATION      =    (YIP)    DEG         ARG OF NODES          =    (ØMP)   DEG
ARG OF PERICEN   =   (PERIG)   DEG         TRUE ANOMALY          =   (ETAP)   DEG
MEAN MOTION      =    (XMM)    DEG/DAY     RADIUS                =     (R)   {A.U.
VELOCITY         =    (VEL)    M/S         FLIGHT PATH ANGLE     =   (THET)  {KM
                                                                              DEG
```

a. Orbital Element Print

```
        TRANSITION MATRIX RELATING STATE AT T1 TO STATE AT T2

          (PH1(1))  (PH1(2))  (PH1(3))  (PH1(4))  (PH1(5))  (PH1(6))
          (PH2(1))  (PH2(2))  (PH2(3))  (PH2(4))  (PH2(5))  (PH2(6))
          (PH3(1))  (PH3(2))  (PH3(3))  (PH3(4))  (PH3(5))  (PH3(6))
          (PH4(1))  (PH4(2))  (PH4(3))  (PH4(4))  (PH4(5))  (PH4(6))
          (PH5(1))  (PH5(2))  (PH5(3))  (PH5(4))  (PH5(5))  (PH5(6))
          (PH6(1))  (PH6(2))  (PH6(3))  (PH6(4))  (PH6(5))  (PH6(6))

  DETERMINANT = (DD)
```

b. Transition Matrix Print

66-2606

Figure IV-4 N-BØDY Optional Output

```
TIME    =  (TIME) DAYS    CASE = (KASE)

                              NAVIGATION PERFORMANCE DATA

                X(KM)      Y(KM)       Z(KM)       VX(M/S)      VY(M/S)       VZ(M/S)

ERR             (EX)       (EY)        (EZ)        (EVX)        (EVY)         (EVZ)
TRUE            (XT1)      (YT1)       (ZT1)       (VXT1)       (VYT1)        (VZT1)
NAV1            (XC1)      (YC1)       (ZC1)       (VXC1)       (VYC1)        (VZC1)
NAV2            (XC2)      (YC2)       (ZC2)       (VXC2)       (VYC2)        (VZC2)
CØRR            (CORR(1))  (CØRR(2))   (CØRR(3))   (CØRR(4))    (CØRR(5))     (CØRR(6))

SYSTEM ERRORS  DEL(ALPHA) = (DELAL) ARCSEC      DEL(DELTA) = (DELD) ARCSEC
OBSERVED BODY = (LABEL(L))

REFERENCE BODY TRUTH       = (LABEL(LL))     ALPHA(PRED) = (ALFPR)    DELTA(PRED) = (DLPPR)
REFERENCE BODY NAVIGATION = (LABEL(N1))      ALPHA(TRUE) = (ALFT)     DELTA(TRUE) = (DELLT)

                              WEIGHTING FUNCTION MATRIX

               X(A.U.)       Y(A.U.)       Z(A.U.)        VX(A.U./DAY)  VY(A.U./DAY)  VZ(A.U./DAY)

ALPHA(RAD)     (AKI(1,1,I))  (AKI(2,1,I))  (AKI(3,1,I))   (AKI(4,1,I))  (AKI(5,1,I))  (AKI(6,1,I))
DELTA(RAD)     (AKI(1,2,I))  (AKI(2,2,I))  (AKI(3,2,I))   (AKI(4,2,I))  (AKI(5,2,I))  (AKI(6,2,I))
```

Figure IV-5 LØF Optional Output

```
NAVIGATED TRAJECTORY NUMBER (J)
TIME =(TIME) DAYS
MEASUREMENT RESULTS   DEL(THE) = (DTHE) DEG      DEL(PSI) = (DPSI) DEG

              X(KM)        Y(KM)        Z(KM)        VX(KM)          VY(KM)           VZ(KM)

TRUE          (TYI(1))     (TYI(2))     (TYI(3))     (TYIDT(1))      (TYIDT(2))       (TYIDT(3))
NAV           (X)          (Y)          (Z)          (XDØT)          (YDØT)           (ZDØT)
ERRØR         (ERRI(1))    (ERRI(2))    (ERRI(3))    (ERRIDT(1))     (ERRIDT(2))      (ERRIDT(3))

              R(KM)        THETA(DEG)   PSI(DEG)     VR(M/S)         VTHE(DEG/SEC)    VPSI(DEG/SEC)

TRUE          (TR)         (TTHE)       (TP)         (TVR)           (TVT)            (TVP)
NAV           (RAUD)       (THT)        (PSIDEG)     (ARDØT)         (THTDØT)         (PSDT)

OBSERVED BODY = (LABEL(M))
TRUTH REFERENCE BODY = (LABEL (J))
NAVIGATED REFERENCE BODY = (LABEL(N1))
SYSTEM MEASUREMENT ERRORS
RIGHT ASCENSION ERROR (ARCSEC) = (DALF)  DECLINATION ERROR (ARCSEC) = (DDEL)
```

Figure IV-6 ISN Optional Output

PRØGRAM	ERROR	MESSAGE	ACTION
N-BØDY	NONE	NONE	NONE
LØF	1. The determinant of the intermediate matrix in the weighting function calculation is equal to zero	INVERSE MATRIX DØES NØT EXIST IN WEIGHTING FUNCTION CALCULATIØN	Run terminates
ISN	1. Denominator ($=X'^2 + Y'^2$)in $\Delta\Theta$ and $\Delta\psi$ calculation is equal to zero	MEASUREMENT NØ. (MTIME) MUST BE DISREGARDED DUE TØ ZERØ DETERMINANT IN DELTA THETA/ DELTA PSI CALCULATION	Program skips rest of measurement calculation and continues to next measurement time.

66-2615

Figure IV-7 Diagnostic Output

V. MAIN PROGRAM FLOW DIAGRAMS

The purpose of Section V is to illustrate the main program logic for each of the programs considered herein. Functional diagrams are given in Figures V-1, V-2 and V-3 for the N-BODY, LOF and ISN programs, respectively. Each figure is further detailed by a series of explanatory block diagrams placed immediately behind the appropriate figure. All block diagrams are referenced in the particular functional diagram that precedes them.

Some functions are performed using subroutines. These are detailed in Section VII and are not further discussed in the present section.

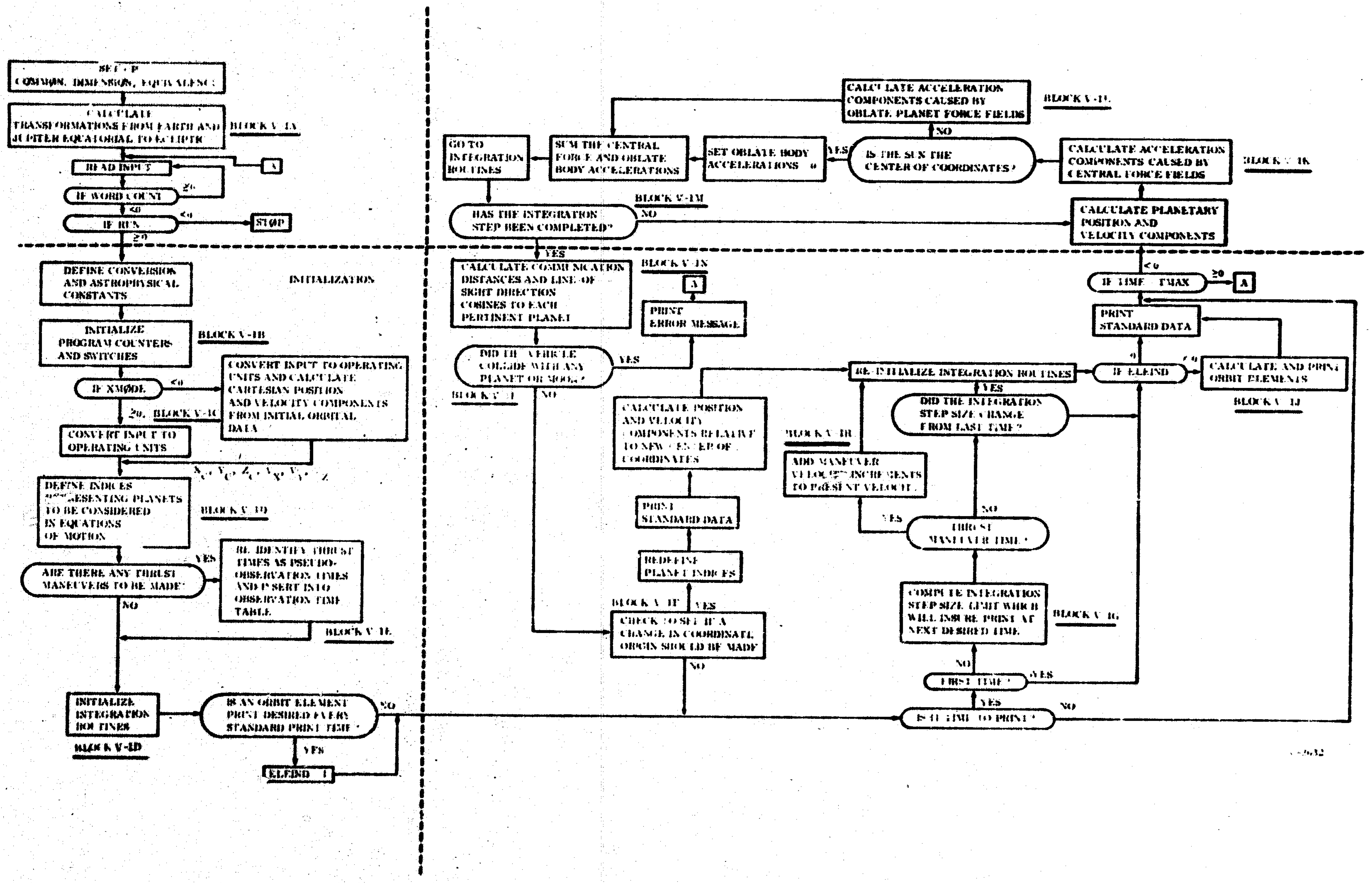
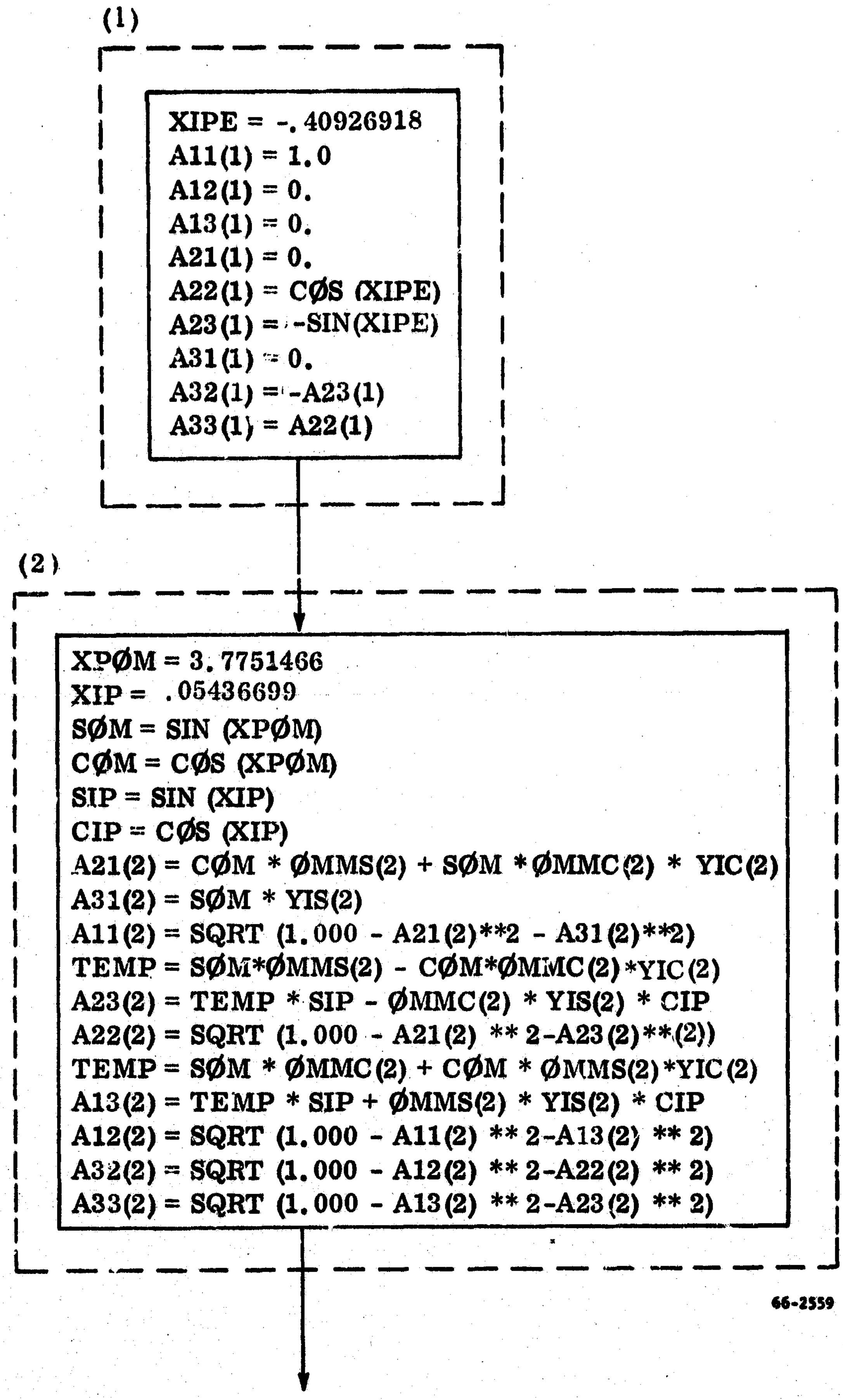


Figure V-1 N-Body Truth Model Computer Program

BLOCK V-1A CALCULATE TRANSFORMATION MATRICES FROM

(1) EARTH EQUATORIAL TO ECLIPTIC

(2) JUPITER EQUATORIAL TO ECLIPTIC



(1)

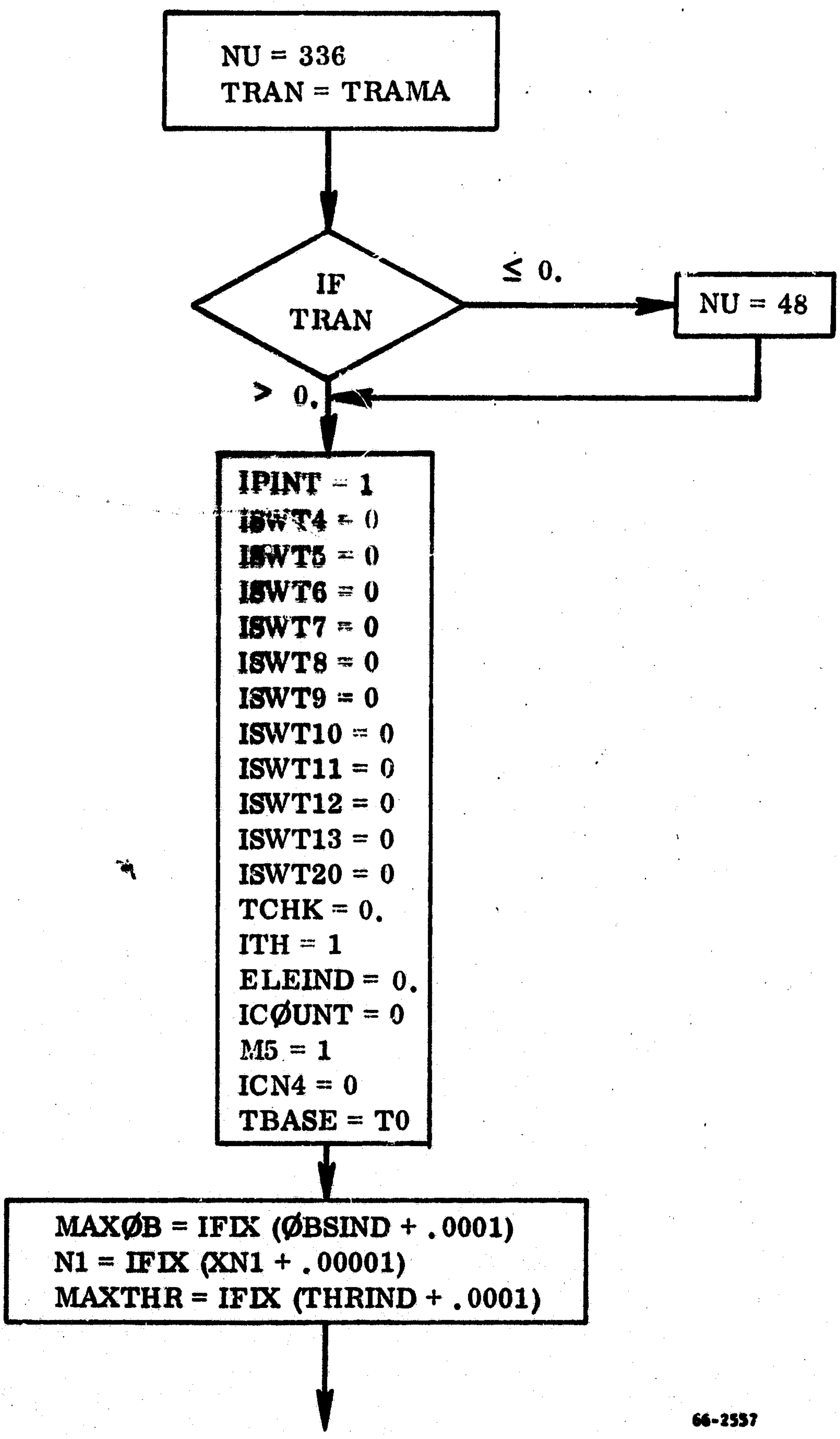
```
XIPE = -.40926918
A11(1) = 1.0
A12(1) = 0.
A13(1) = 0.
A21(1) = 0.
A22(1) = CØS (XIPE)
A23(1) = -SIN(XIPE)
A31(1) = 0.
A32(1) = -A23(1)
A33(1) = A22(1)
```

(2)

```
XPØM = 3.7751466
XIP = .05436699
SØM = SIN (XPØM)
CØM = CØS (XPØM)
SIP = SIN (XIP)
CIP = CØS (XIP)
A21(2) = CØM * ØMMS(2) + SØM *ØMMC(2) * YIC(2)
A31(2) = SØM * YIS(2)
A11(2) = SQRT (1.000 - A21(2)**2 - A31(2)**2)
TEMP = SØM*ØMMS(2) - CØM*ØMMC(2)*YIC(2)
A23(2) = TEMP * SIP - ØMMC(2) * YIS(2) * CIP
A22(2) = SQRT (1.000 - A21(2) ** 2-A23(2)**(2))
TEMP = SØM * ØMMC(2) + CØM * ØMMS(2)*YIC(2)
A13(2) = TEMP * SIP + ØMMS(2) * YIS(2) * CIP
A12(2) = SQRT (1.000 - A11(2) ** 2-A13(2) ** 2)
A32(2) = SQRT (1.000 - A12(2) ** 2-A22(2) ** 2)
A33(2) = SQRT (1.000 - A13(2) ** 2-A23(2) ** 2)
```

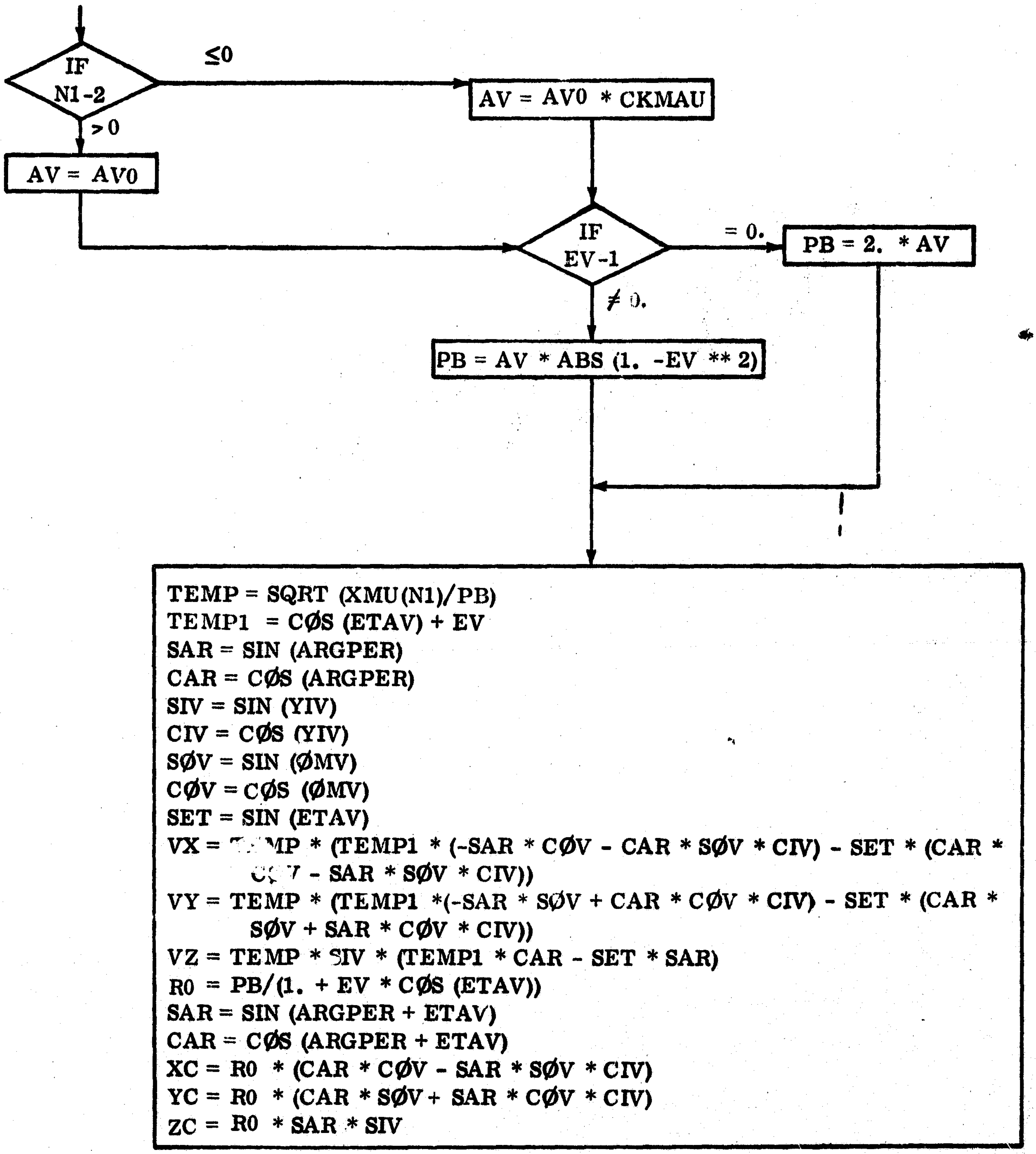
66-2559

BLOCK V-1B INITIALIZE PROGRAM COUNTERS AND SWITCHES



66-2557

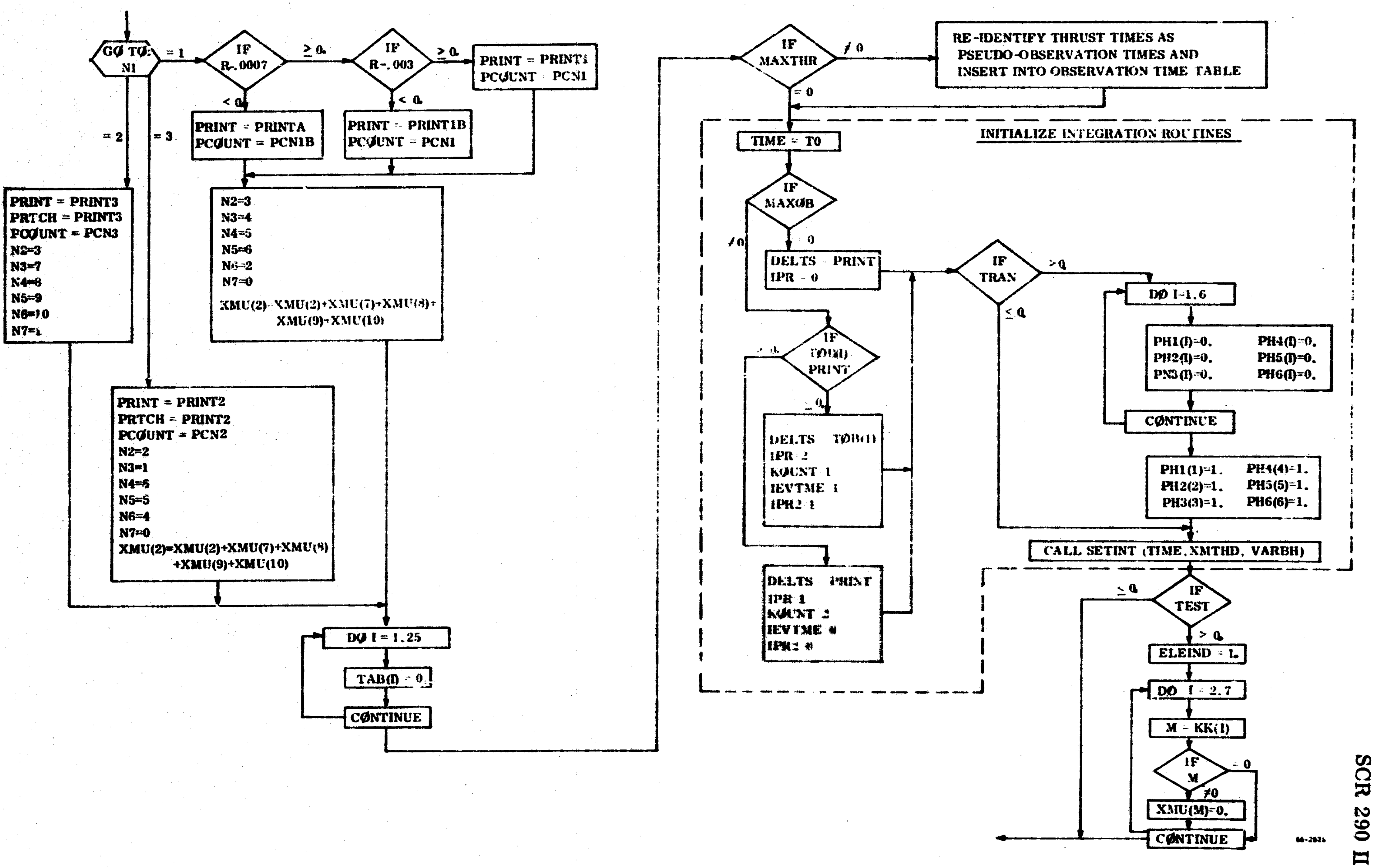
BLOCK V-1C CONVERT INPUT TO OPERATING UNITS AND CALCULATE CARTESIAN POSITION AND VELOCITY COMPONENTS FROM INITIAL ORBITAL DATA



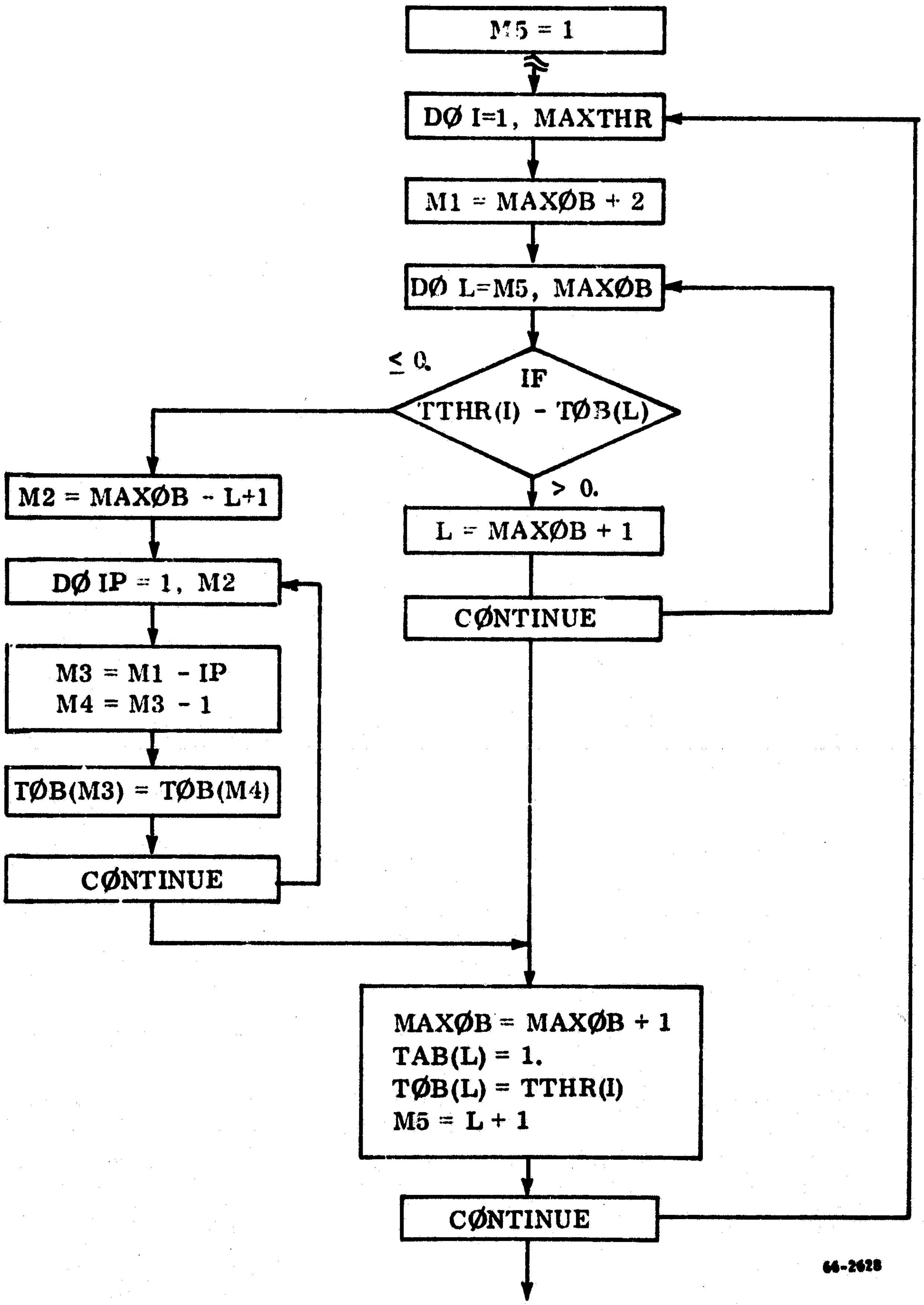
```
TEMP = SQRT (XMU(N1)/PB)
TEMP1  = CØS (ETAV) + EV
SAR = SIN (ARGPER)
CAR = CØS (ARGPER)
SIV = SIN (YIV)
CIV = CØS (YIV)
SØV = SIN (ØMV)
CØV = CØS (ØMV)
SET = SIN (ETAV)
VX = TEMP * (TEMP1 * (-SAR * CØV - CAR * SØV * CIV) - SET * (CAR *
        CØV - SAR * SØV * CIV))
VY = TEMP * (TEMP1 *(-SAR * SØV + CAR * CØV * CIV) - SET * (CAR *
        SØV + SAR * CØV * CIV))
VZ = TEMP * SIV * (TEMP1 * CAR - SET * SAR)
R0 = PB/(1. + EV * CØS (ETAV))
SAR = SIN (ARGPER + ETAV)
CAR = CØS (ARGPER + ETAV)
XC = R0  * (CAR * CØV - SAR * SØV * CIV)
YC = R0  * (CAR * SØV + SAR * CØV * CIV)
ZC = R0  * SAR * SIV
```

66-2576

BLOCK V-1D DEFINE INDICES REPRESENTING PLANETS TO BE CONSIDERED IN EQUATIONS OF MOTION AND INITIALIZE INTEGRATION ROUTINES

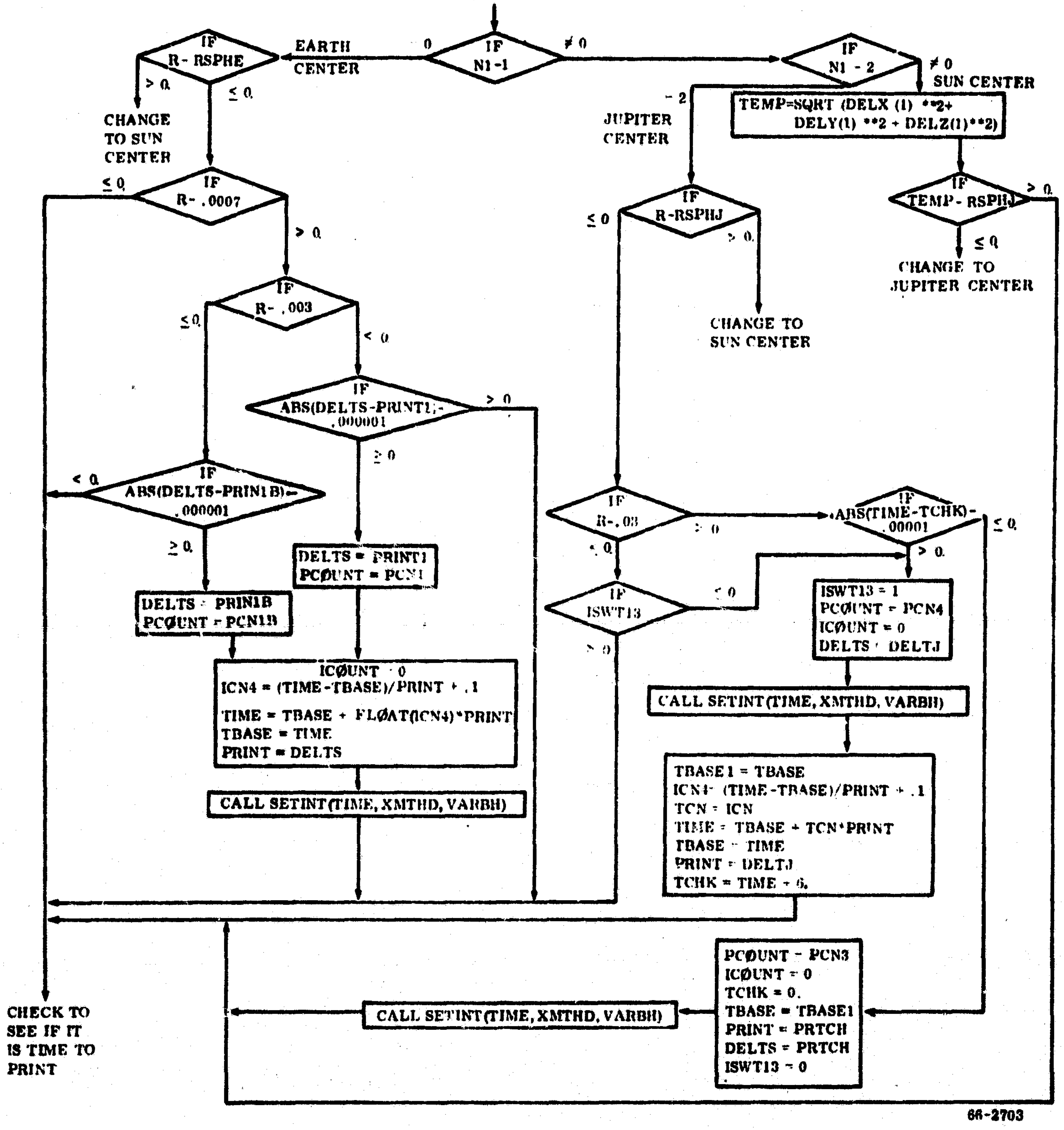


BLOCK V-1E RE-IDENTIFY THRUST AS PSEUDO-OBSERVATION TIMES AND INSERT INTO OBSERVATION TIME TABLE

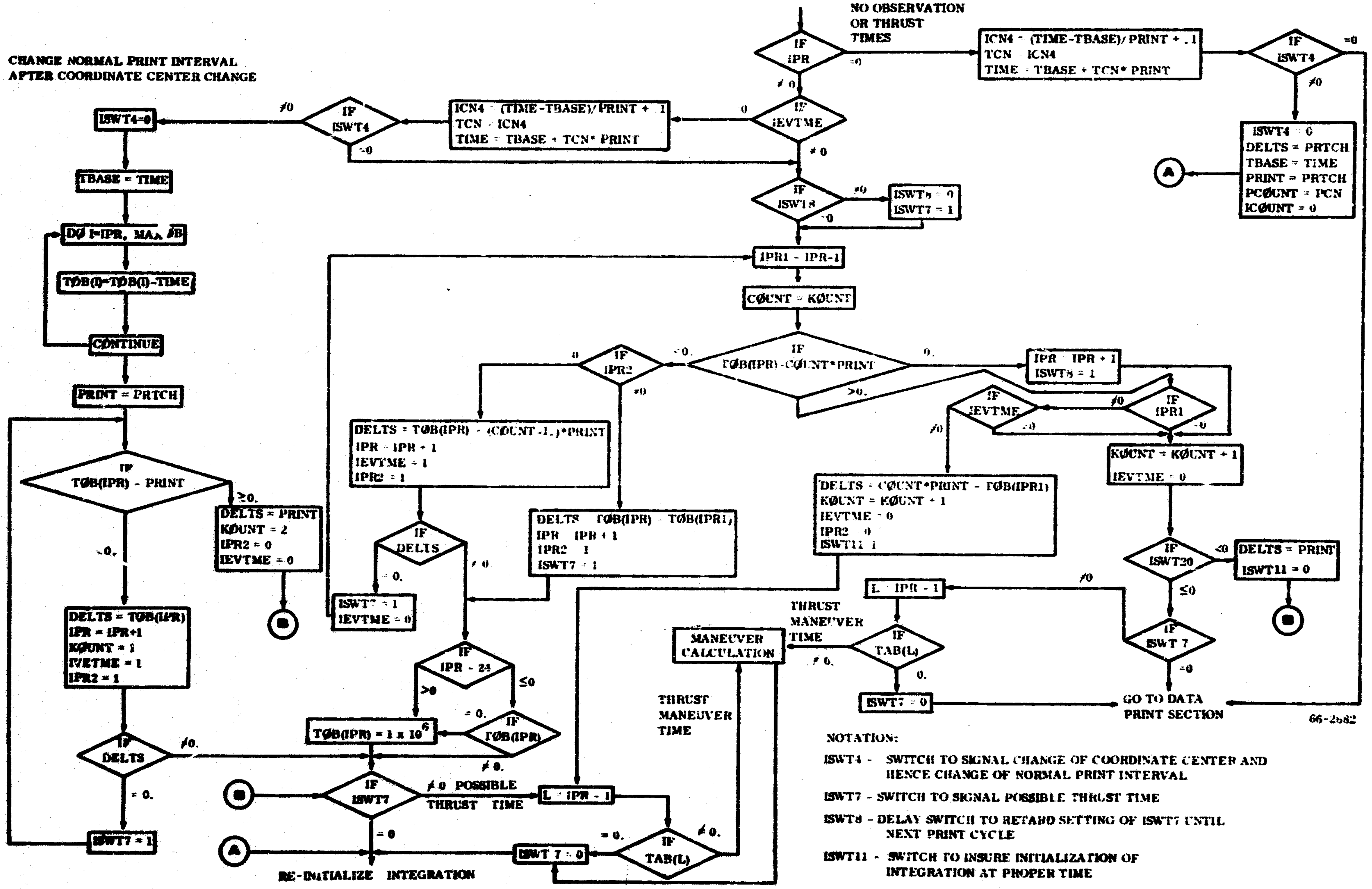


64-2628

BLOCK V-1F CHECK TO SEE IF A CHANGE IN COORDINATE ORIGIN SHOULD BE MADE



BLOCK V-1G COMPUTE INTEGRATION STEP SIZE LIMIT TO INSURE PRINT AT NEXT DESIRED TIME



NOTATION:

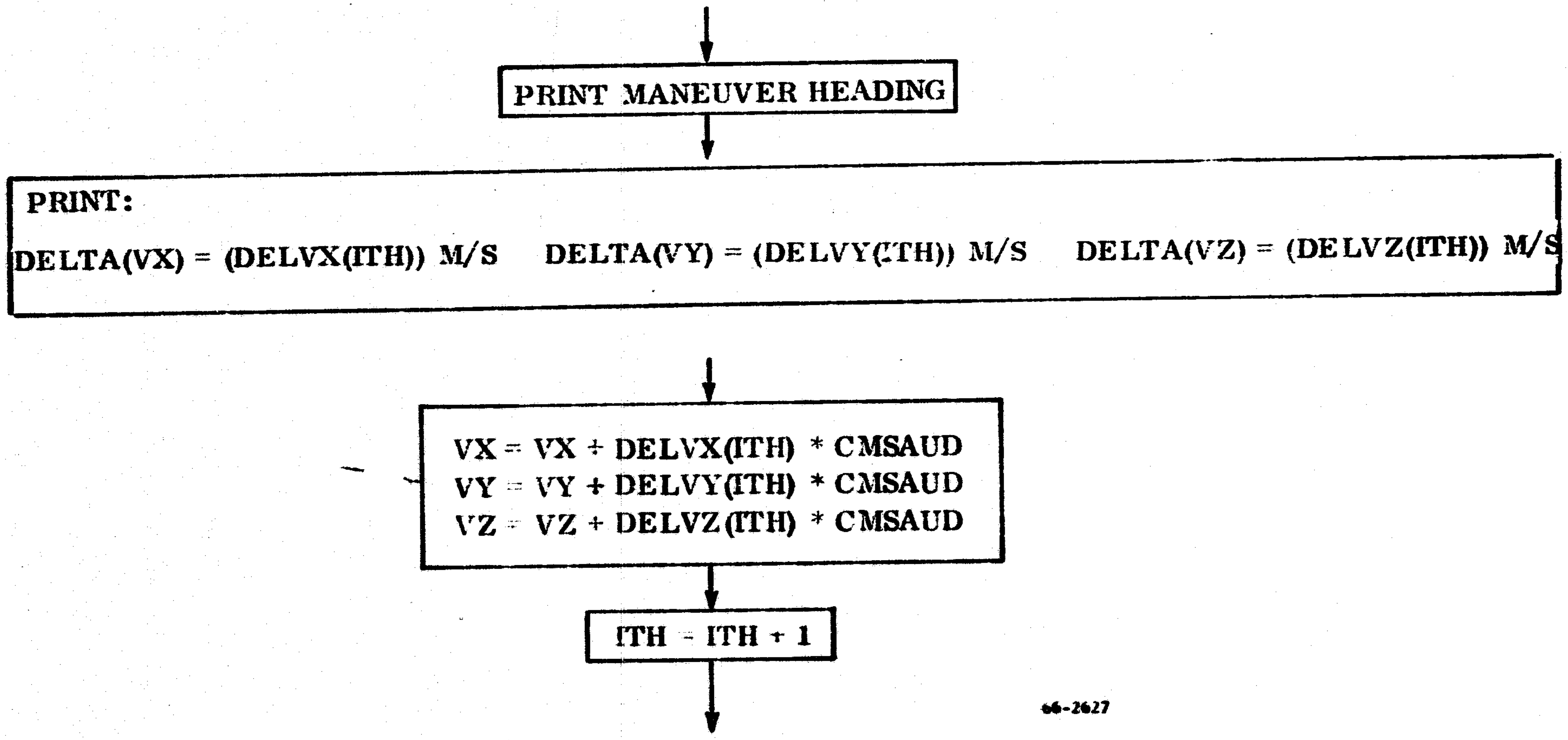
ISWT4 - SWITCH TO SIGNAL CHANGE OF COORDINATE CENTER AND HENCE CHANGE OF NORMAL PRINT INTERVAL

ISWT7 - SWITCH TO SIGNAL POSSIBLE THRUST TIME

ISWT8 - DELAY SWITCH TO RETARD SETTING OF ISWT7 UNTIL NEXT PRINT CYCLE

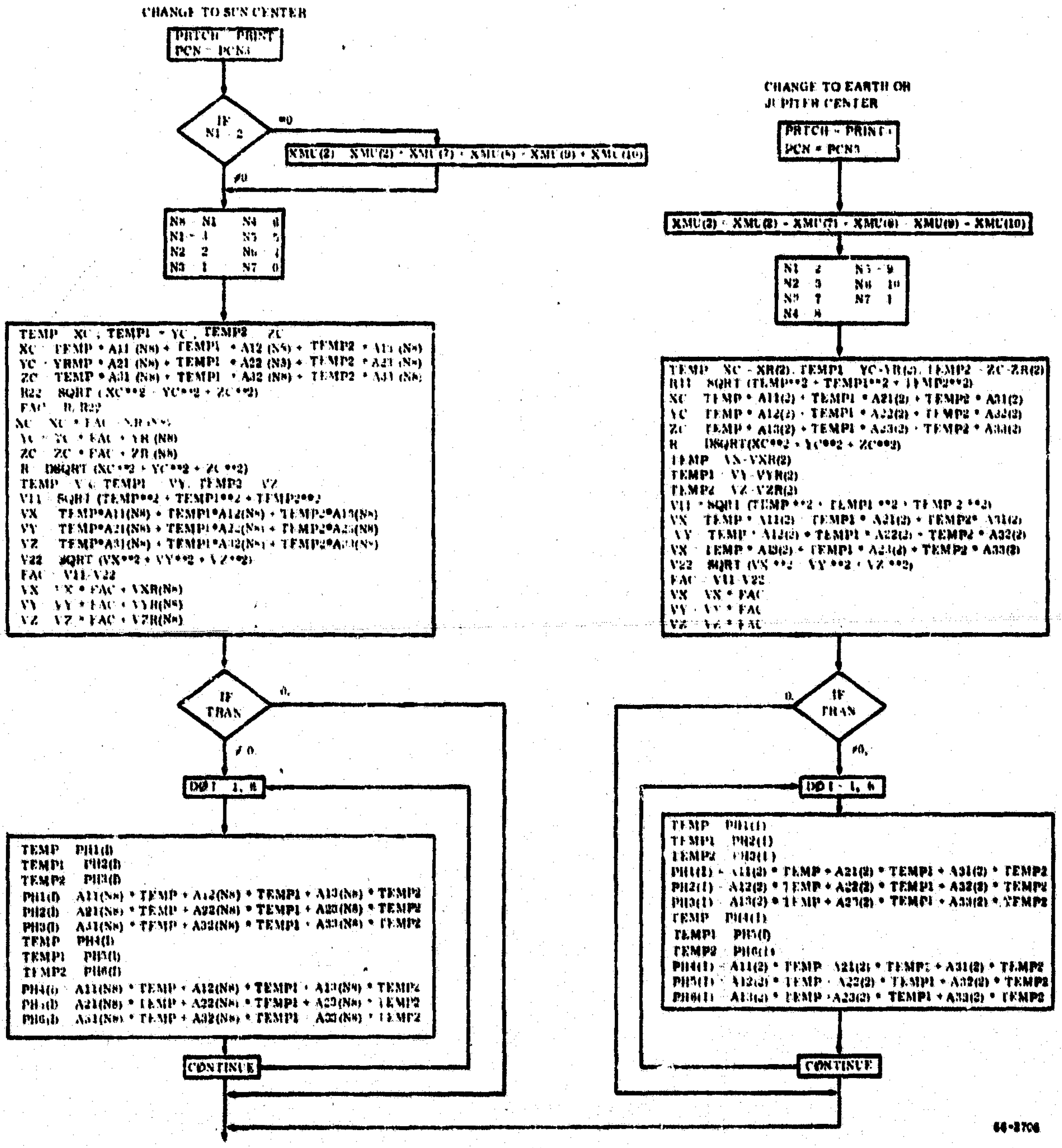
ISWT11 - SWITCH TO INSURE INITIALIZATION OF INTEGRATION AT PROPER TIME

BLOCK V-1H ADD MANEUVER VELOCITY INCREMENTS TO PRESENT VELOCITY



66-2627

BLOCK V-11 CALCULATE POSITION AND VELOCITY COMPONENTS RELATIVE TO NEW CENTER OF COORDINATES



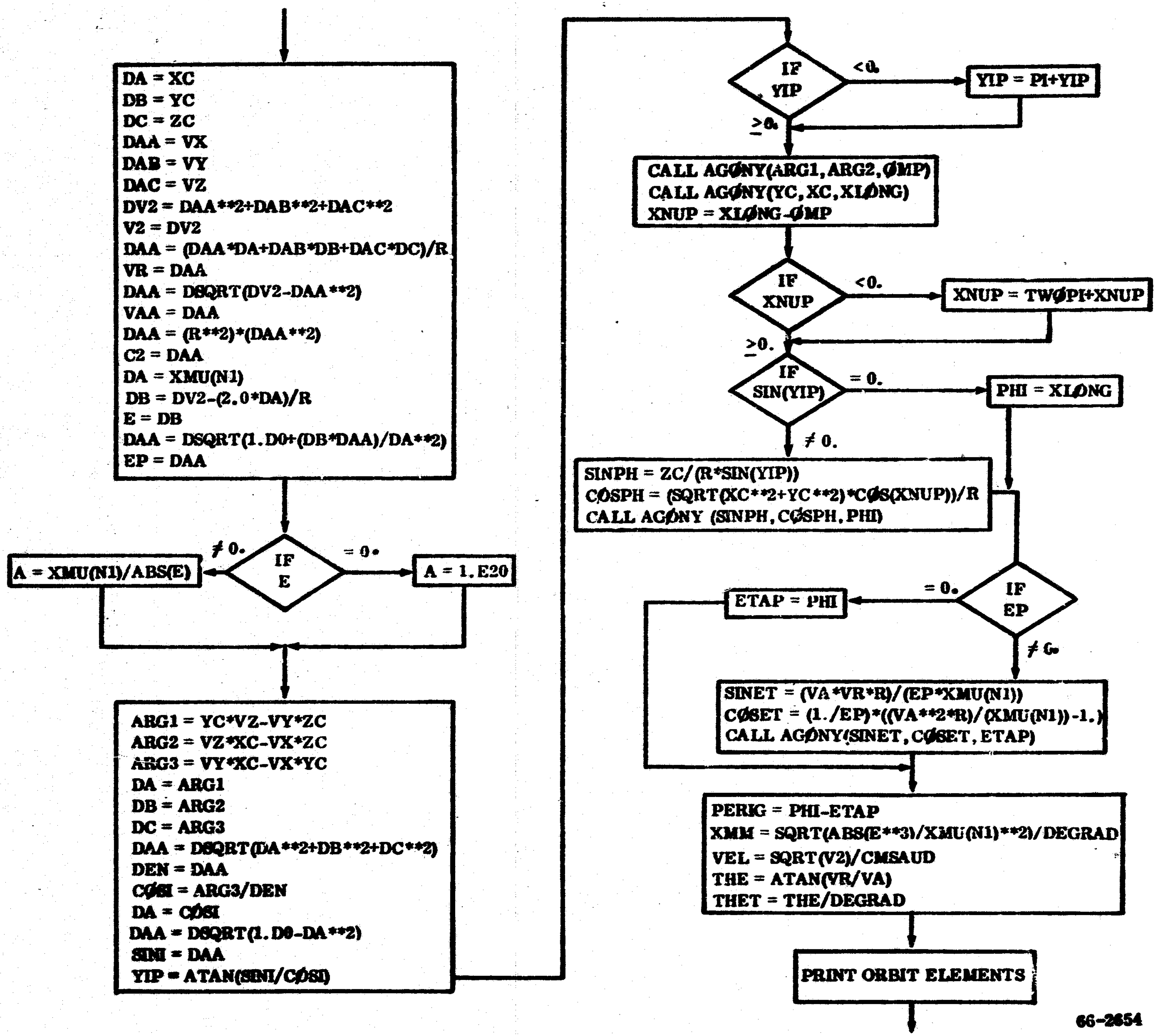
```
TEMP = XC , TEMP1 = YC , TEMP2 = ZC
XC = TEMP * A11 (N8) + TEMP1 * A12 (N8) + TEMP2 * A13 (N8)
YC = YRMP * A21 (N8) + TEMP1 * A22 (N8) + TEMP2 * A23 (N8)
ZC = TEMP * A31 (N8) + TEMP1 * A32 (N8) + TEMP2 * A33 (N8)
R22 = SQRT (XC**2 + YC**2 + ZC**2)
FAC = R/R22
XC = XC * FAC + XR (N8)
YC = YC * FAC + YR (N8)
ZC = ZC * FAC + ZR (N8)
R = DSQRT (XC**2 + YC**2 + ZC**2)
TEMP = VX, TEMP1 = VY, TEMP2 = VZ
V11 = SQRT (TEMP**2 + TEMP1**2 + TEMP2**2)
VX = TEMP*A11(N8) + TEMP1*A12(N8) + TEMP2*A13(N8)
VY = TEMP*A21(N8) + TEMP1*A22(N8) + TEMP2*A23(N8)
VZ = TEMP*A31(N8) + TEMP1*A32(N8) + TEMP2*A33(N8)
V22 = SQRT (VX**2 + VY**2 + VZ**2)
FAC = V11/V22
VX = VX * FAC + VXR(N8)
VY = VY * FAC + VYR(N8)
VZ = VZ * FAC + VZR(N8)
```

```
TEMP = PH1(I)
TEMP1 = PH2(I)
TEMP2 = PH3(I)
PH1(I) = A11(N8) * TEMP + A12(N8) * TEMP1 + A13(N8) * TEMP2
PH2(I) = A21(N8) * TEMP + A22(N8) * TEMP1 + A23(N8) * TEMP2
PH3(I) = A31(N8) * TEMP + A32(N8) * TEMP1 + A33(N8) * TEMP2
TEMP = PH4(I)
TEMP1 = PH5(I)
TEMP2 = PH6(I)
PH4(I) = A11(N8) * TEMP + A12(N8) * TEMP1 + A13(N8) * TEMP2
PH5(I) = A21(N8) * TEMP + A22(N8) * TEMP1 + A23(N8) * TEMP2
PH6(I) = A31(N8) * TEMP + A32(N8) * TEMP1 + A33(N8) * TEMP2
```

```
TEMP = XC - XR(2), TEMP1 = YC-YR(2), TEMP2 = ZC-ZR(2)
R11 = SQRT (TEMP**2 + TEMP1**2 + TEMP2**2)
XC = TEMP * A11(2) + TEMP1 * A21(2) + TEMP2 * A31(2)
YC = TEMP * A12(2) + TEMP1 * A22(2) + TEMP2 * A32(2)
ZC = TEMP * A13(2) + TEMP1 * A23(2) + TEMP2 * A33(2)
R = DSQRT(XC**2 + YC**2 + ZC**2)
TEMP = VX-VXR(2)
TEMP1 = VY-VYR(2)
TEMP2 = VZ-VZR(2)
V11 = SQRT (TEMP **2 + TEMP1 **2 + TEMP 2 **2)
VX = TEMP * A11(2) + TEMP1 * A21(2) + TEMP2* A31(2)
VY = TEMP * A12(2) + TEMP1 * A22(2) + TEMP2 * A32(2)
VX = TEMP * A13(2) + TEMP1 * A23(2) + TEMP2 * A33(2)
V22 = SQRT (VX **2 + VY **2 + VZ **2)
FAC = V11/V22
VX = VX * FAC
VY = VY * FAC
VZ = VZ * FAC
```

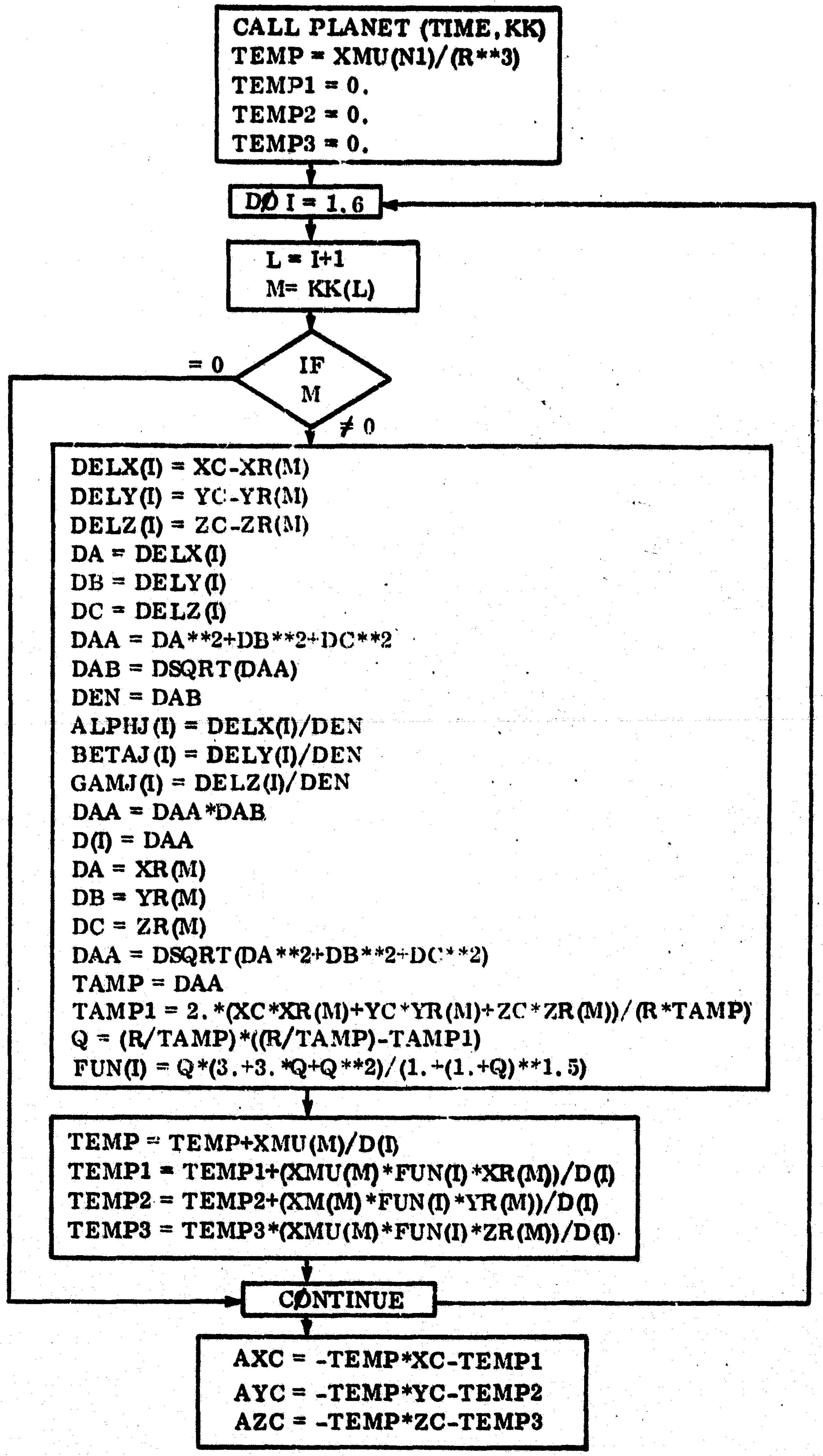
```
TEMP = PH1(I)
TEMP1 = PH2(I)
TEMP2 = PH3(I)
PH1(I) = A11(2) * TEMP + A21(2) * TEMP1 + A31(2) * TEMP2
PH2(I) = A12(2) * TEMP + A22(2) * TEMP1 + A32(2) * TEMP2
PH3(I) = A13(2) * TEMP + A23(2) * TEMP1 + A33(2) * TEMP2
TEMP = PH4(I)
TEMP1 = PH5(I)
TEMP2 = PH6(I)
PH4(I) = A11(2) * TEMP + A21(2) * TEMP1 + A31(2) * TEMP2
PH5(I) = A12(2) * TEMP + A22(2) * TEMP1 + A32(2) * TEMP2
PH6(I) = A13(2) * TEMP + A23(2) * TEMP1 + A33(2) * TEMP2
```

BLOCK V-1J CALCULATE AND PRINT ORBIT ELEMENTS



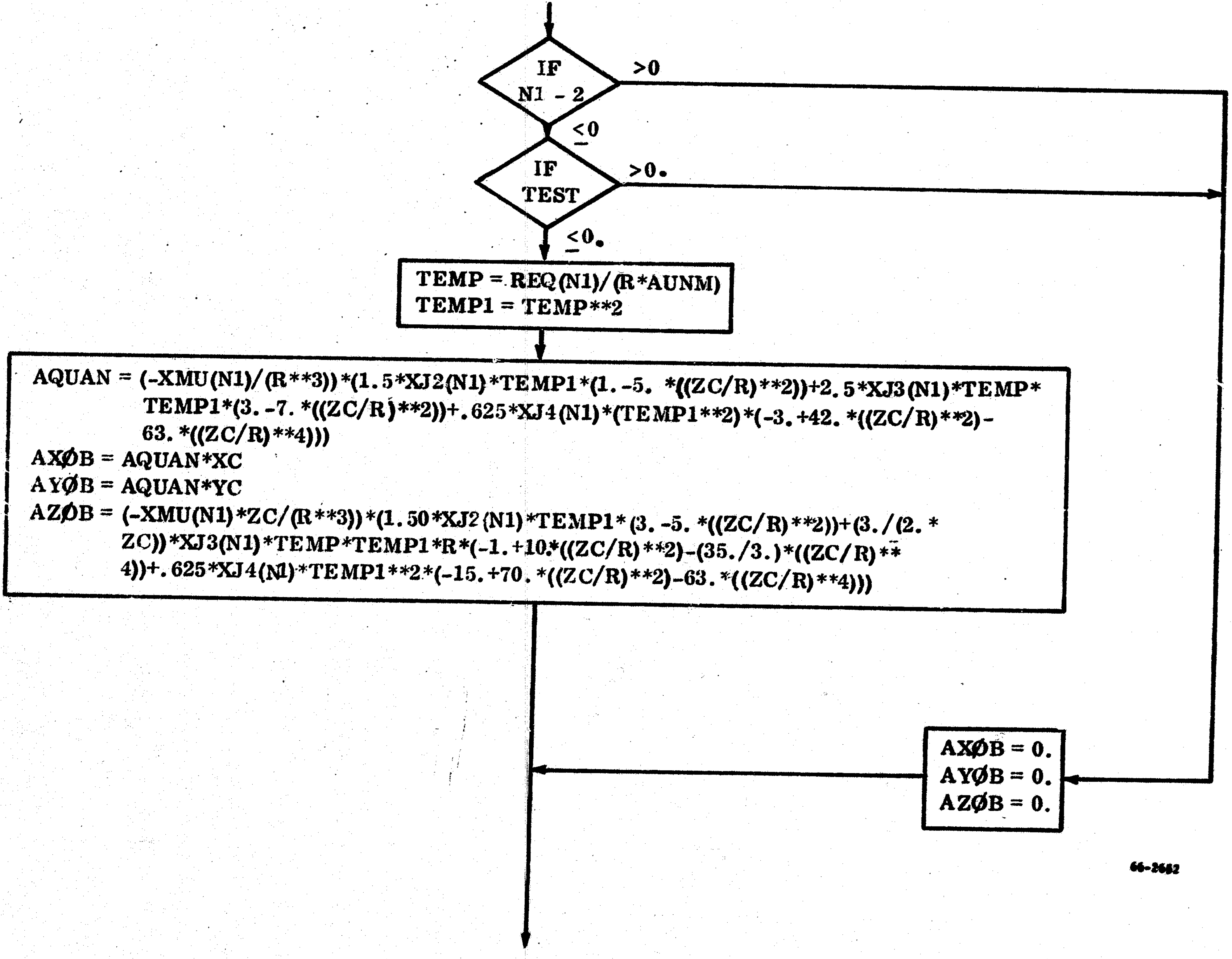
66-2654

BLOCK V-1K COMPUTF ACCELERATION COMPONENTS CAUSED BY CENTRAL FORCE FIELDS



```
CALL PLANET (TIME,KK)
TEMP = XMU(N1)/(R**3)
TEMP1 = 0.
TEMP2 = 0.
TEMP3 = 0.

DØ I = 1.6

L = I+1
M= KK(L)

DELX(I) = XC-XR(M)
DELY(I) = YC-YR(M)
DELZ(I) = ZC-ZR(M)
DA = DELX(I)
DB = DELY(I)
DC = DELZ(I)
DAA = DA**2+DB**2+DC**2
DAB = DSQRT(DAA)
DEN = DAB
ALPHJ(I) = DELX(I)/DEN
BETAJ(I) = DELY(I)/DEN
GAMJ(I) = DELZ(I)/DEN
DAA = DAA*DAB
D(I) = DAA
DA = XR(M)
DB = YR(M)
DC = ZR(M)
DAA = DSQRT(DA**2+DB**2+DC**2)
TAMP = DAA
TAMP1 = 2.*(XC*XR(M)+YC*YR(M)+ZC*ZR(M))/(R*TAMP)
Q = (R/TAMP)*((R/TAMP)-TAMP1)
FUN(I) = Q*(3.+3.*Q+Q**2)/(1.+(1.+Q)**1.5)

TEMP = TEMP+XMU(M)/D(I)
TEMP1 = TEMP1+(XMU(M)*FUN(I)*XR(M))/D(I)
TEMP2 = TEMP2+(XM(M)*FUN(I)*YR(M))/D(I)
TEMP3 = TEMP3*(XMU(M)*FUN(I)*ZR(M))/D(I)

CØNTINUE

AXC = -TEMP*XC-TEMP1
AYC = -TEMP*YC-TEMP2
AZC = -TEMP*ZC-TEMP3
```

BLOCK V-1L CALCULATE ACCELERATION COMPONENTS CAUSED BY OBLATE PLANET FORCE FIELDS



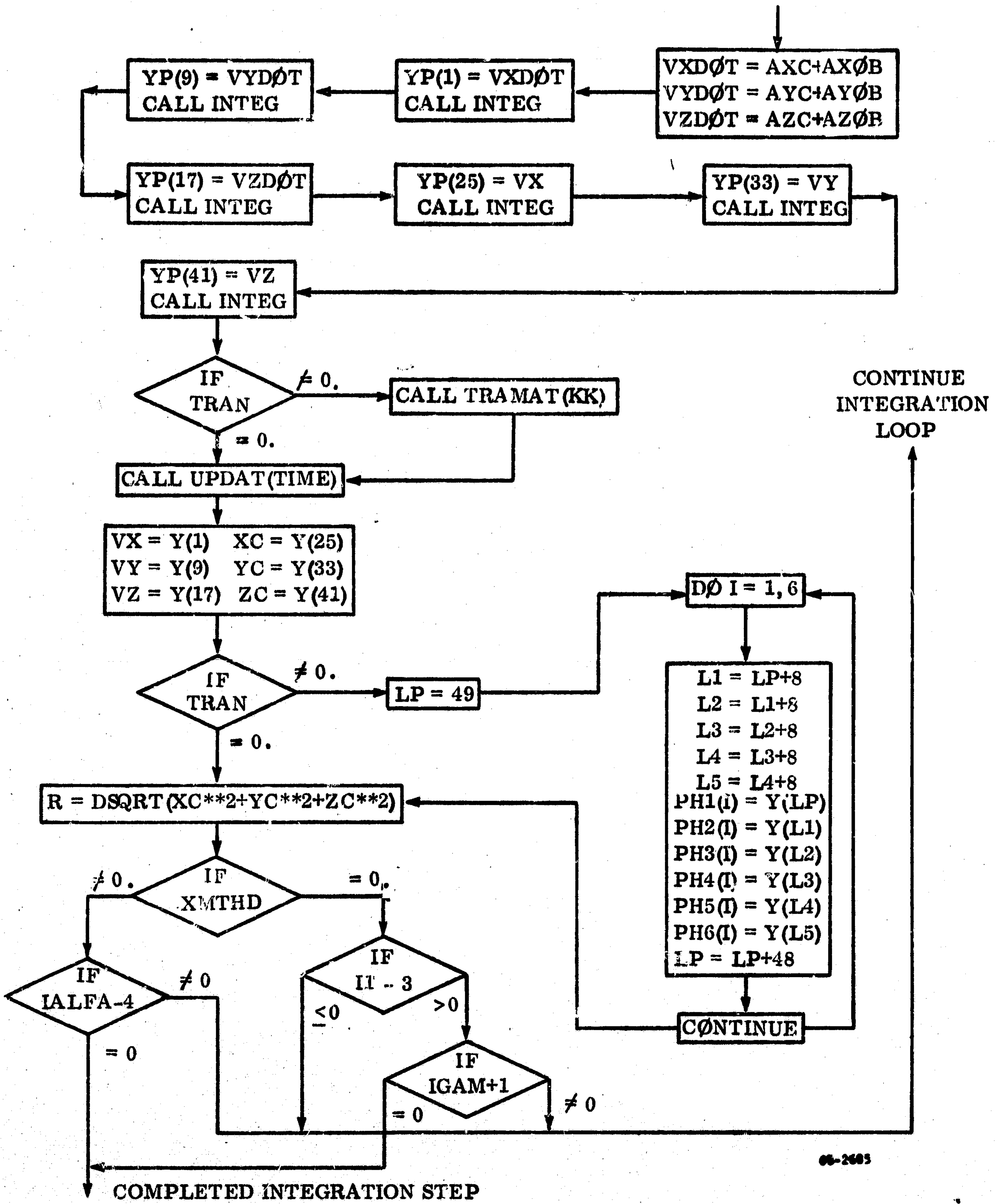
```
TEMP = REQ(N1)/(R*AUNM)
TEMP1 = TEMP**2
```

```
AQUAN = (-XMU(N1)/(R**3))*(1.5*XJ2(N1)*TEMP1*(1.-5. *((ZC/R)**2))+2.5*XJ3(N1)*TEMP*
          TEMP1*(3.-7. *((ZC/R)**2))+.625*XJ4(N1)*(TEMP1**2)*(-3.+42. *((ZC/R)**2)-
          63. *((ZC/R)**4)))
AXØB = AQUAN*XC
AYØB = AQUAN*YC
AZØB = (-XMU(N1)*ZC/(R**3))*(1.50*XJ2(N1)*TEMP1*(3.-5. *((ZC/R)**2))+(3./(2. *
        ZC))*XJ3(N1)*TEMP*TEMP1*R*(-1.+10.*((ZC/R)**2)-(35./3.)*((ZC/R)**
        4))+.625*XJ4(N1)*TEMP1**2*(-15.+70. *((ZC/R)**2)-63. *((ZC/R)**4)))
```

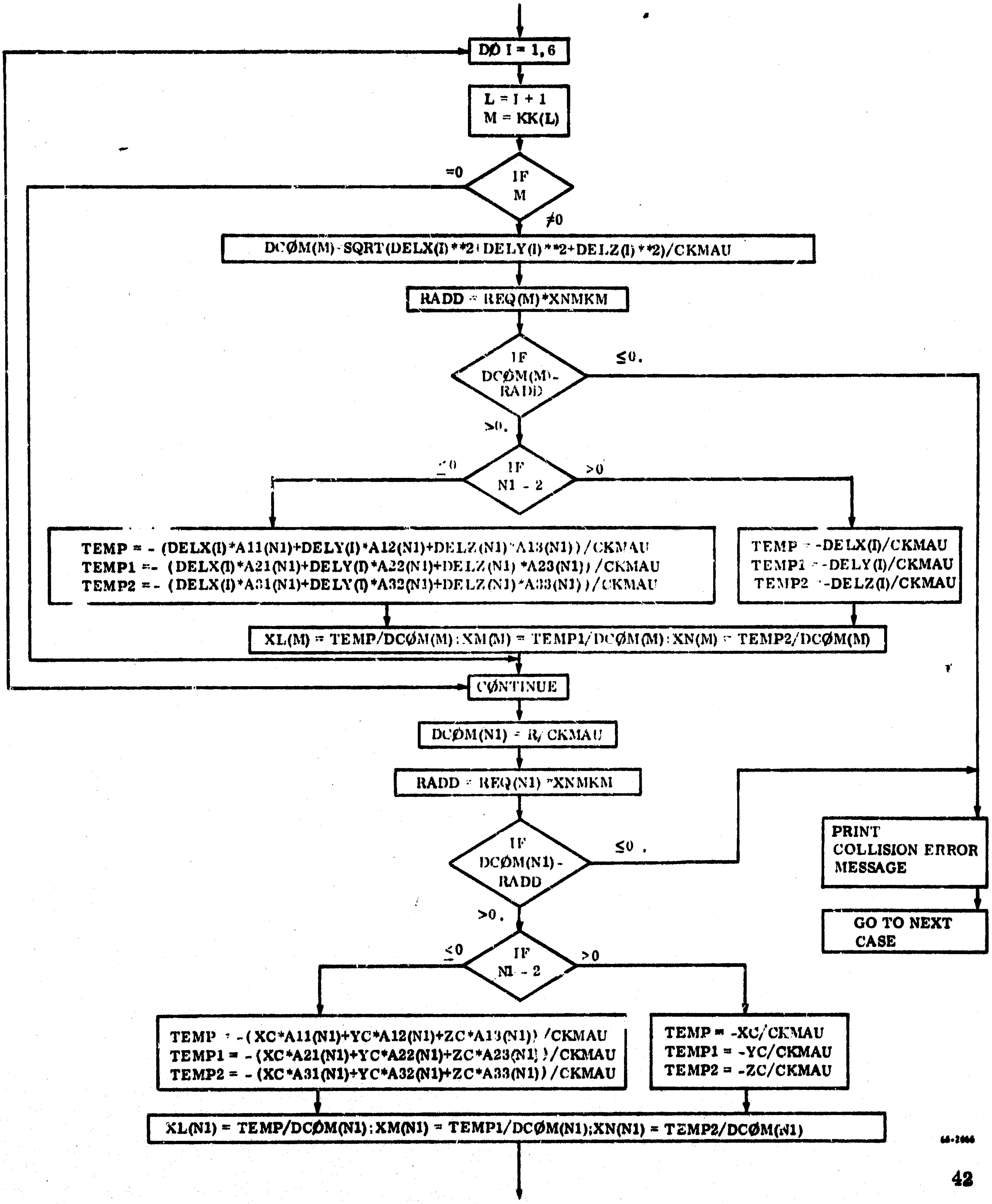
```
AXØB = 0.
AYØB = 0.
AZØB = 0.
```

66-2682

BLOCK V-1M GO TO INTEGRATION ROUTINES AND CHECK FOR COMPLETED INTEGRATION STEP



BLOCK V-1N CALCULATE COMMUNICATION DISTANCES AND LINE-OF-SIGHT DIRECTION COSINES TO EACH PERTINENT PLANET



66-2066

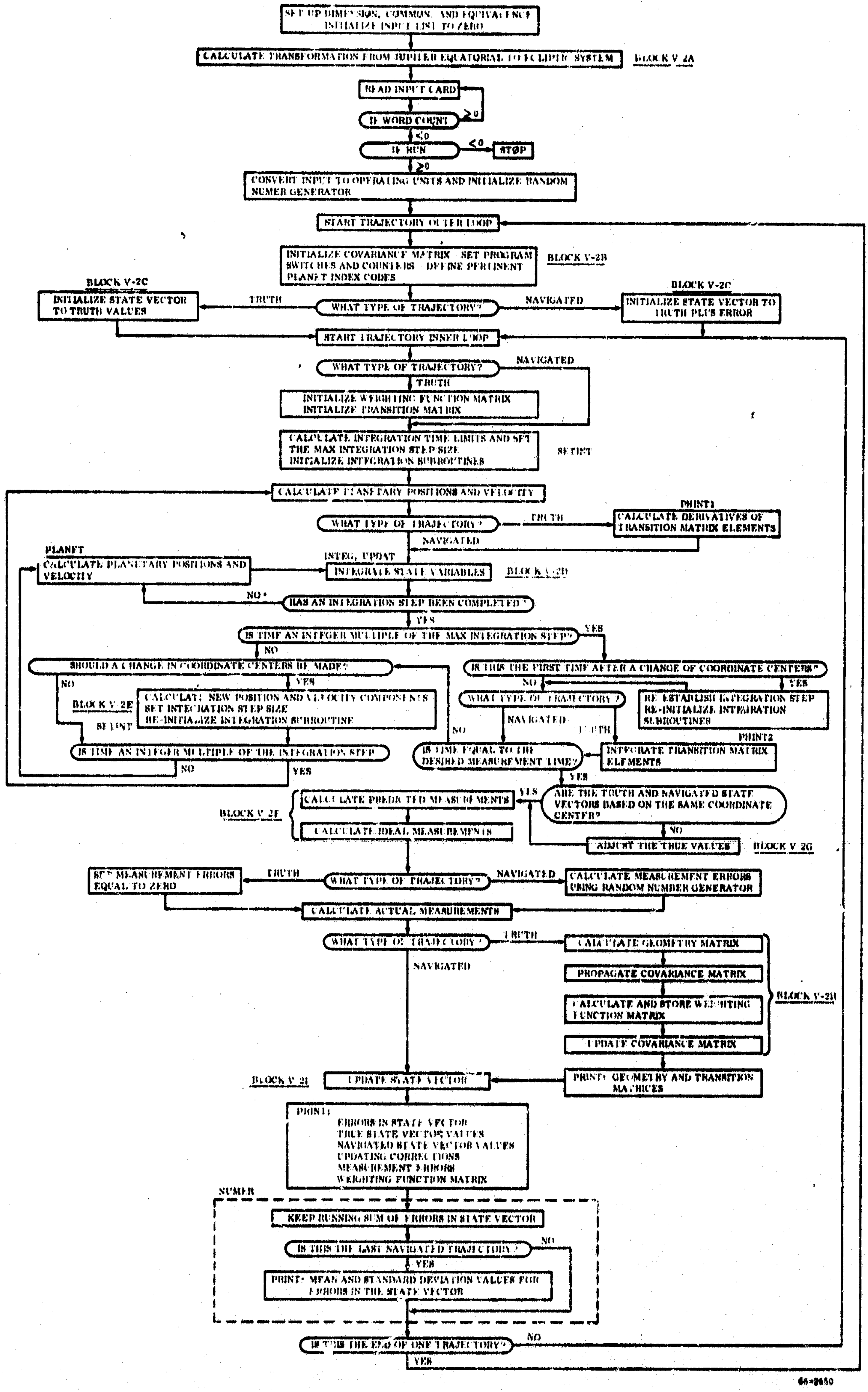
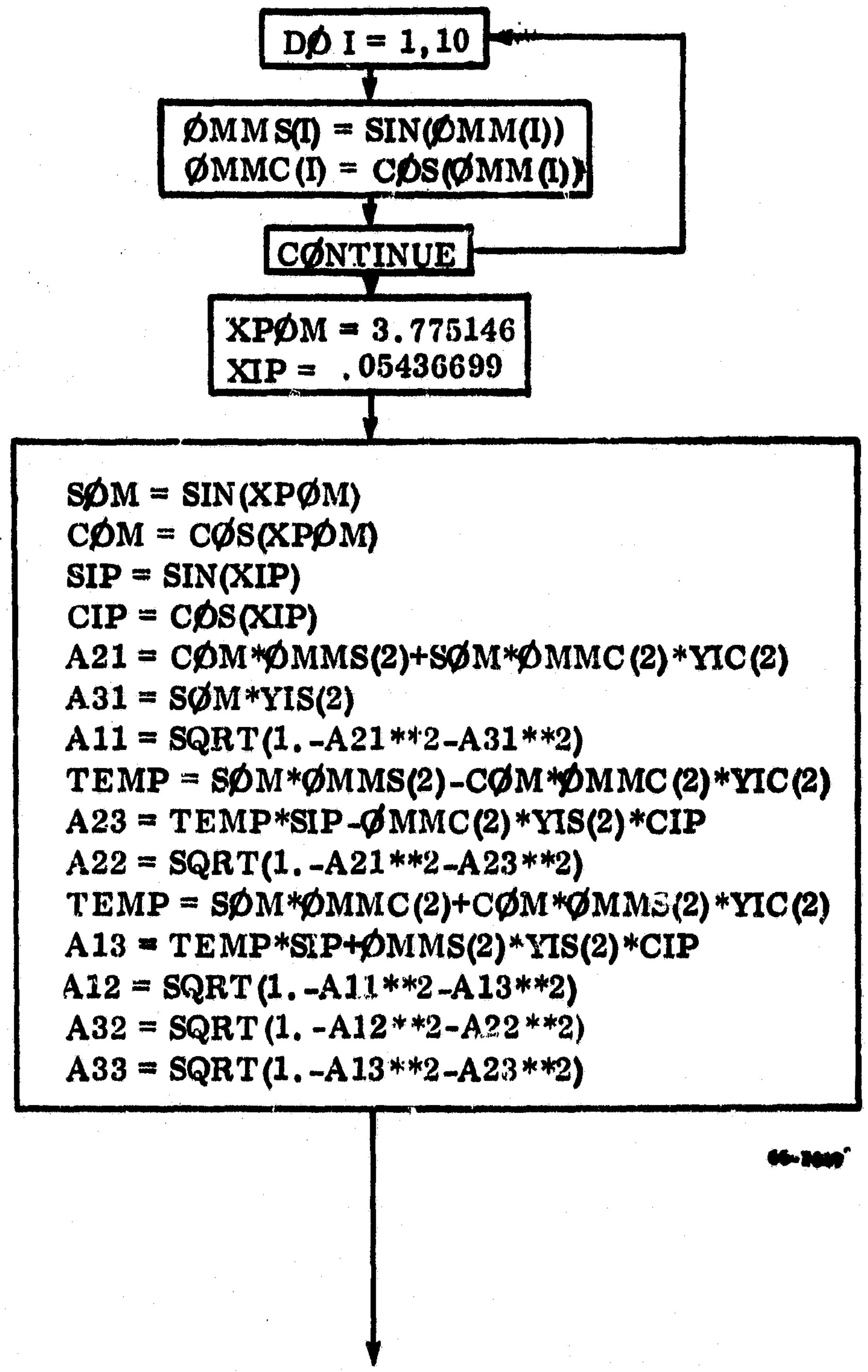


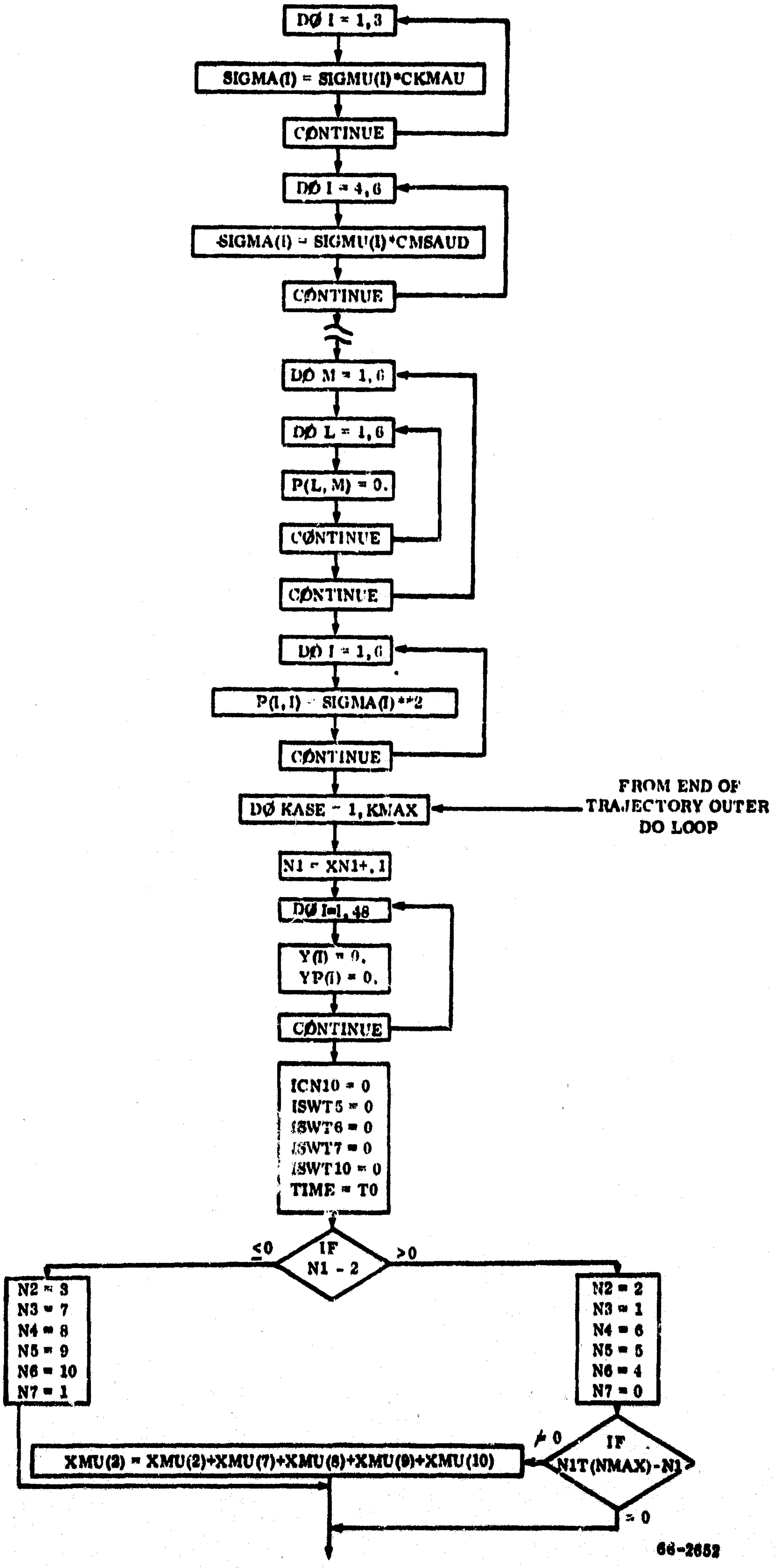
Figure V-2 Linear Optimum Filter Navigation Program

BLOCK V-2A CALCULATE TRANSFORMATION FROM JUPITER EQUATORIAL COORDINATE SYSTEM TO ECLIPTIC SYSTEM

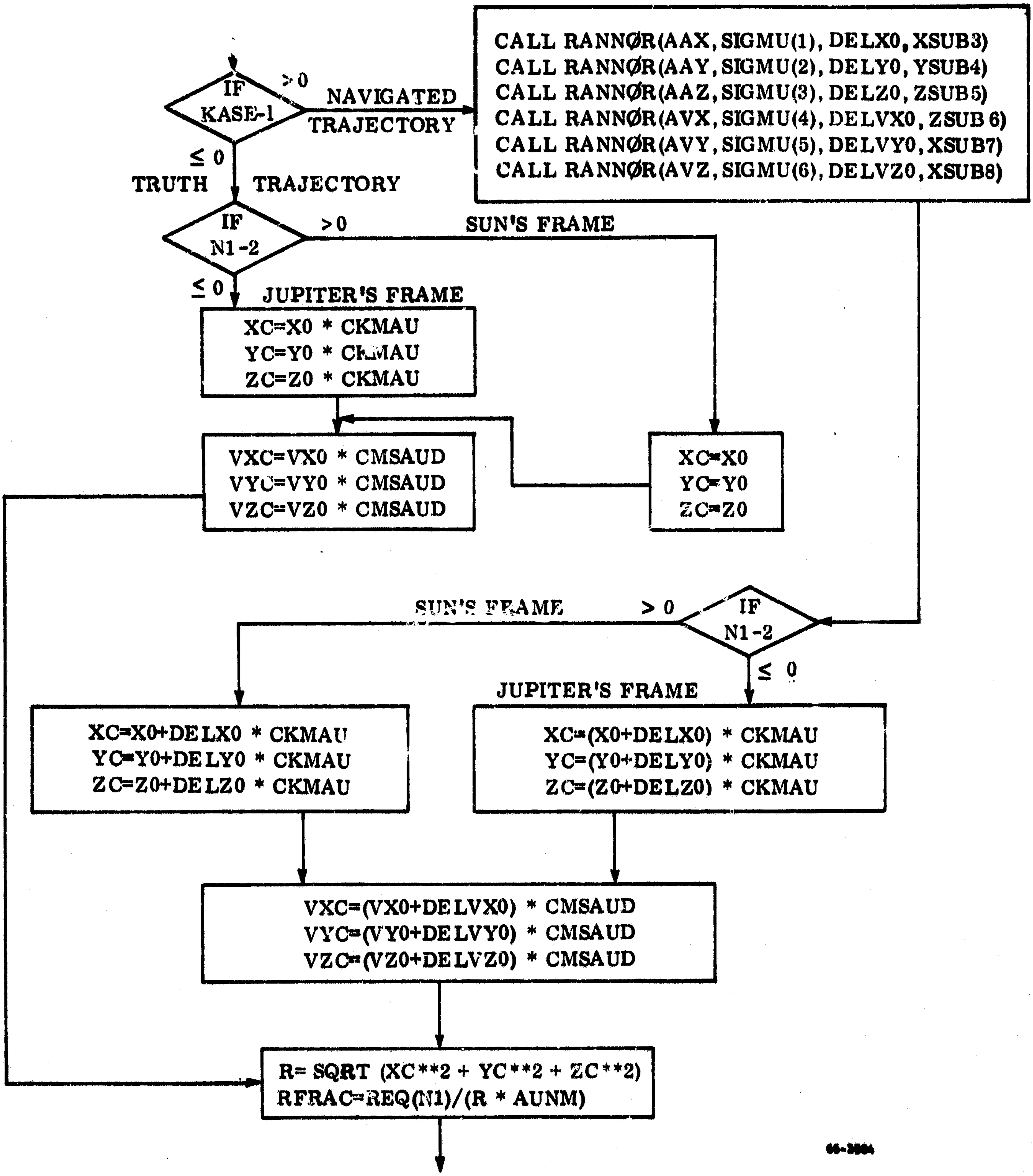


```
DØ I = 1,10
ØMMS(I) = SIN(ØMM(I))
ØMMC(I) = CØS(ØMM(I))
CØNTINUE
XPØM = 3.775146
XIP = .05436699
SØM = SIN(XPØM)
CØM = CØS(XPØM)
SIP = SIN(XIP)
CIP = CØS(XIP)
A21 = CØM*ØMMS(2)+SØM*ØMMC(2)*YIC(2)
A31 = SØM*YIS(2)
A11 = SQRT(1.-A21**2-A31**2)
TEMP = SØM*ØMMS(2)-CØM*ØMMC(2)*YIC(2)
A23 = TEMP*SIP-ØMMC(2)*YIS(2)*CIP
A22 = SQRT(1.-A21**2-A23**2)
TEMP = SØM*ØMMC(2)+CØM*ØMMS(2)*YIC(2)
A13 = TEMP*SIP+ØMMS(2)*YIS(2)*CIP
A12 = SQRT(1.-A11**2-A13**2)
A32 = SQRT(1.-A12**2-A22**2)
A33 = SQRT(1.-A13**2-A23**2)
```

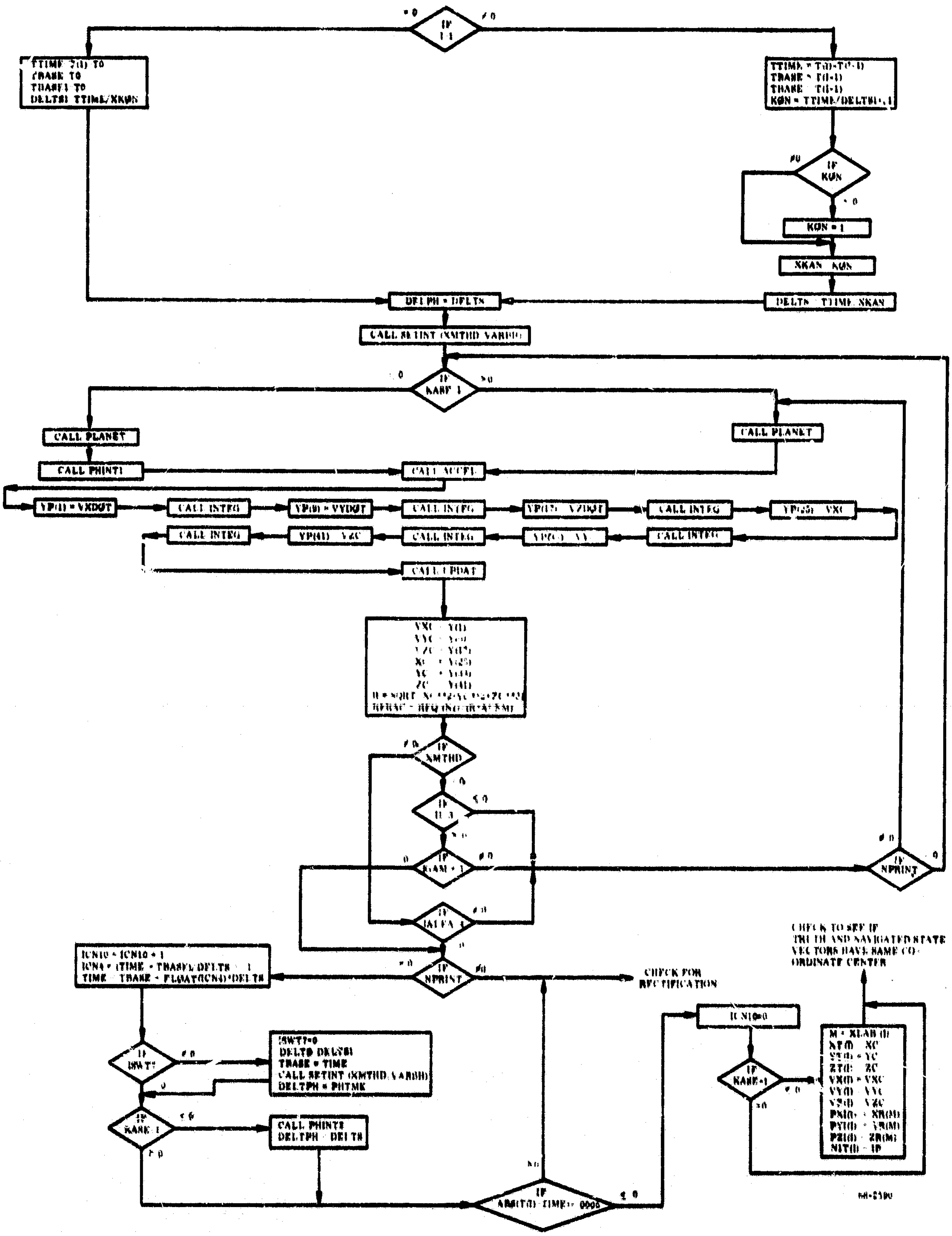
BLOCK V-2B INITIALIZE: COVARIANCE MATRIX, PROGRAM SWITCHES, AND INDEX CODES



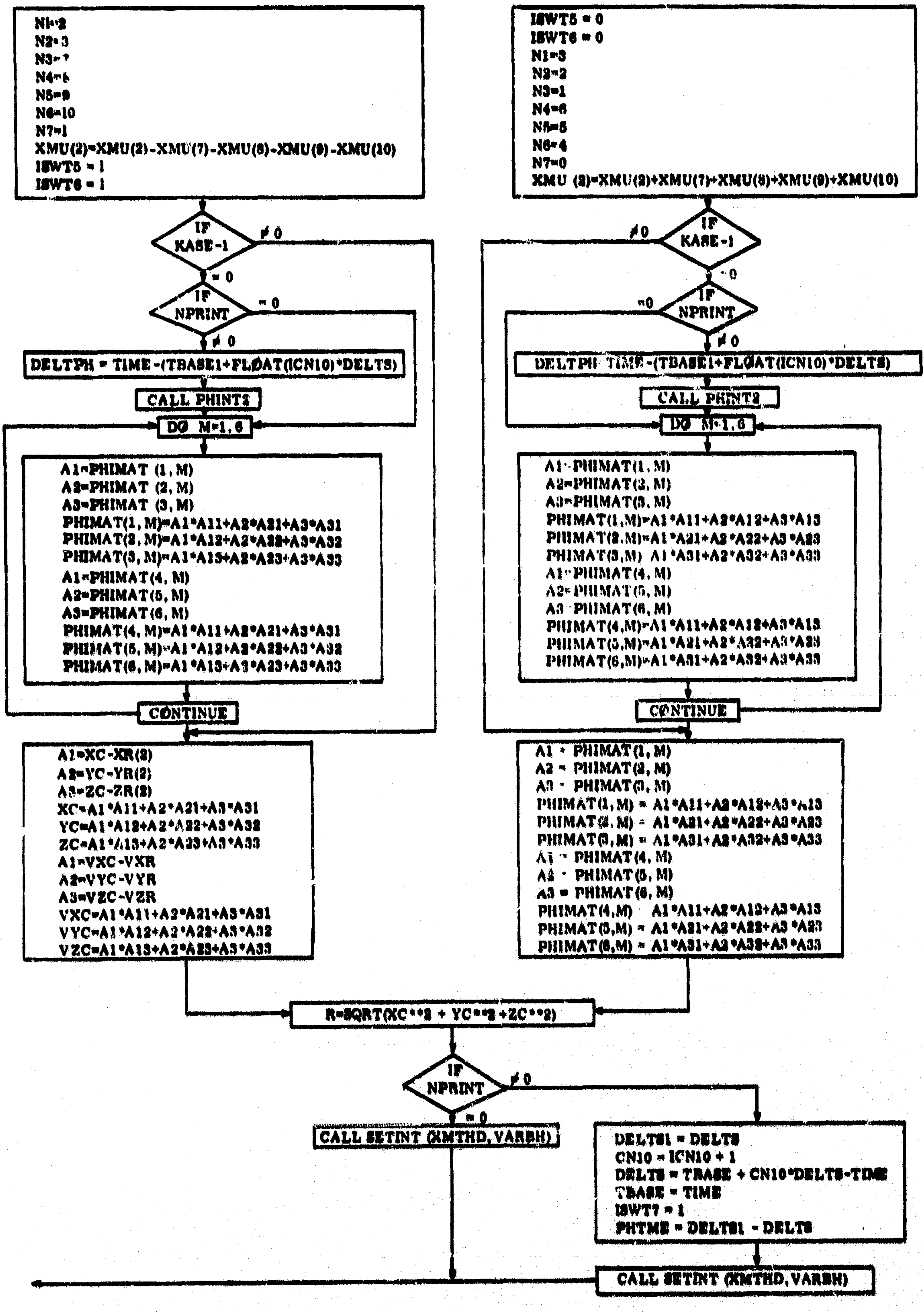
BLOCK V-2C INITIALIZE STATE VECTOR



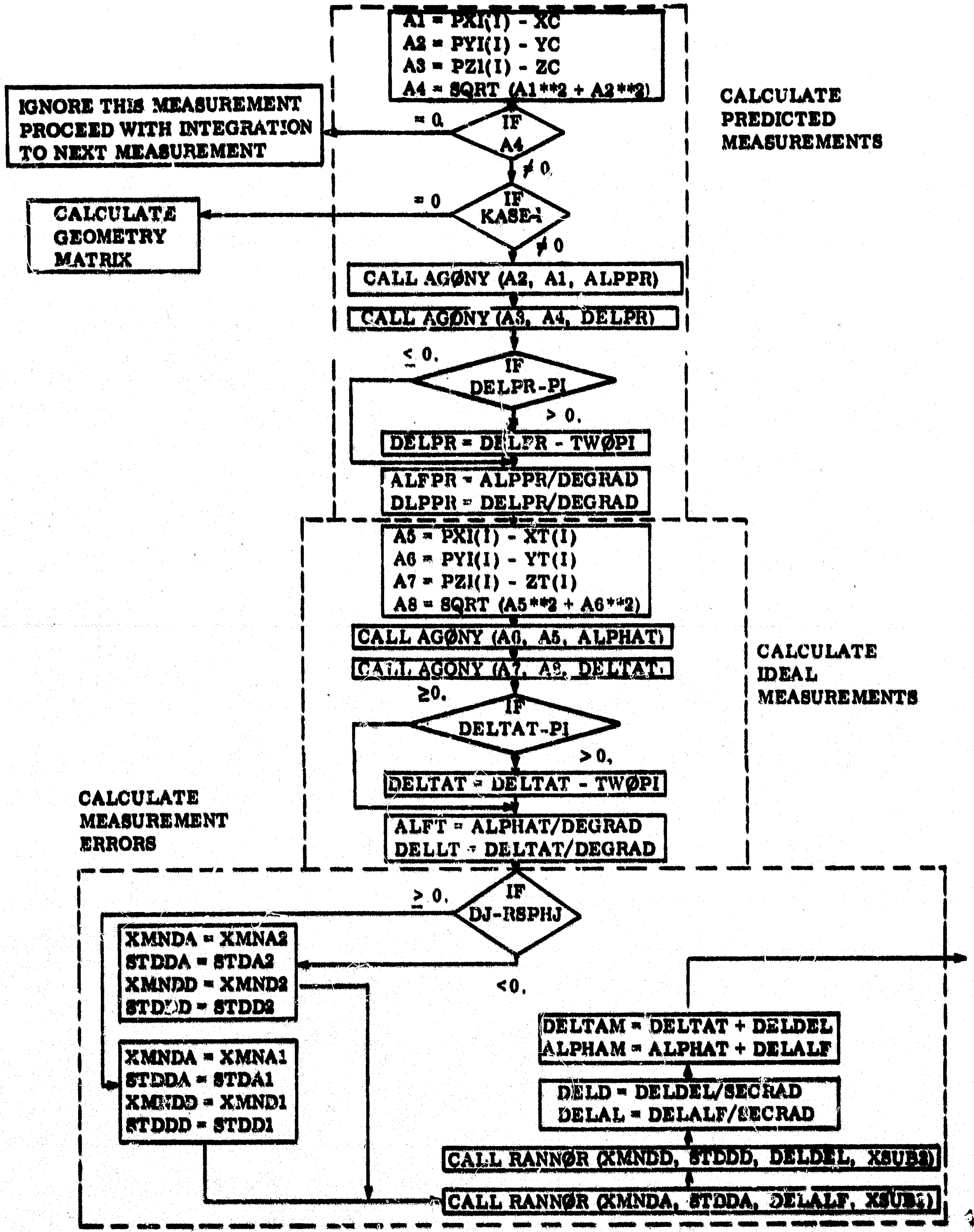
BLOCK V-2D INTEGRATE STATE VARIABLES AND TRANSITION MATRIX ELEMENTS



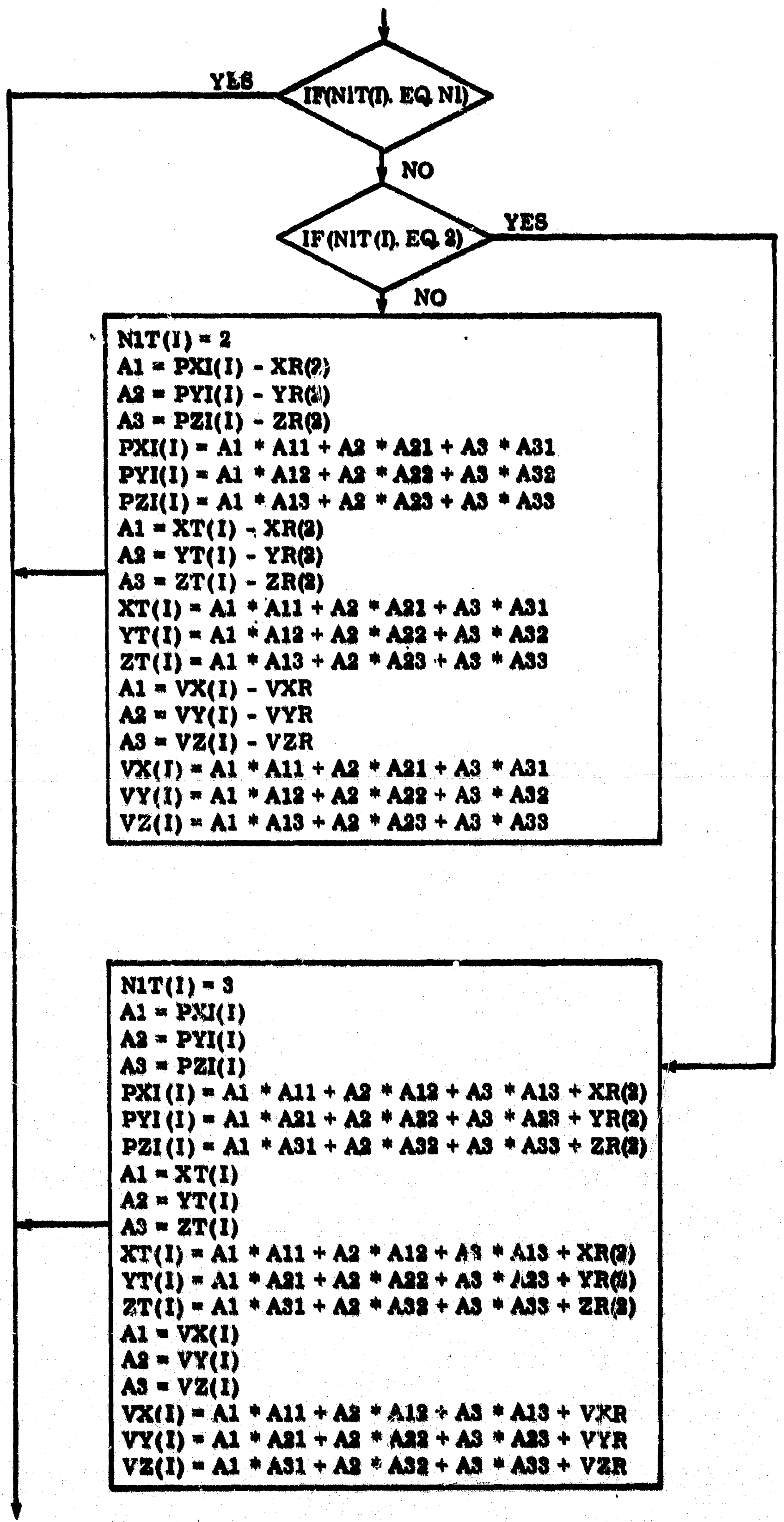
BLOCK V-2E CALCULATE NEW POSITION AND VELOCITY COMPONENTS



BLOCK V-2F CALCULATE PREDICTED AND IDEAL MEASUREMENTS



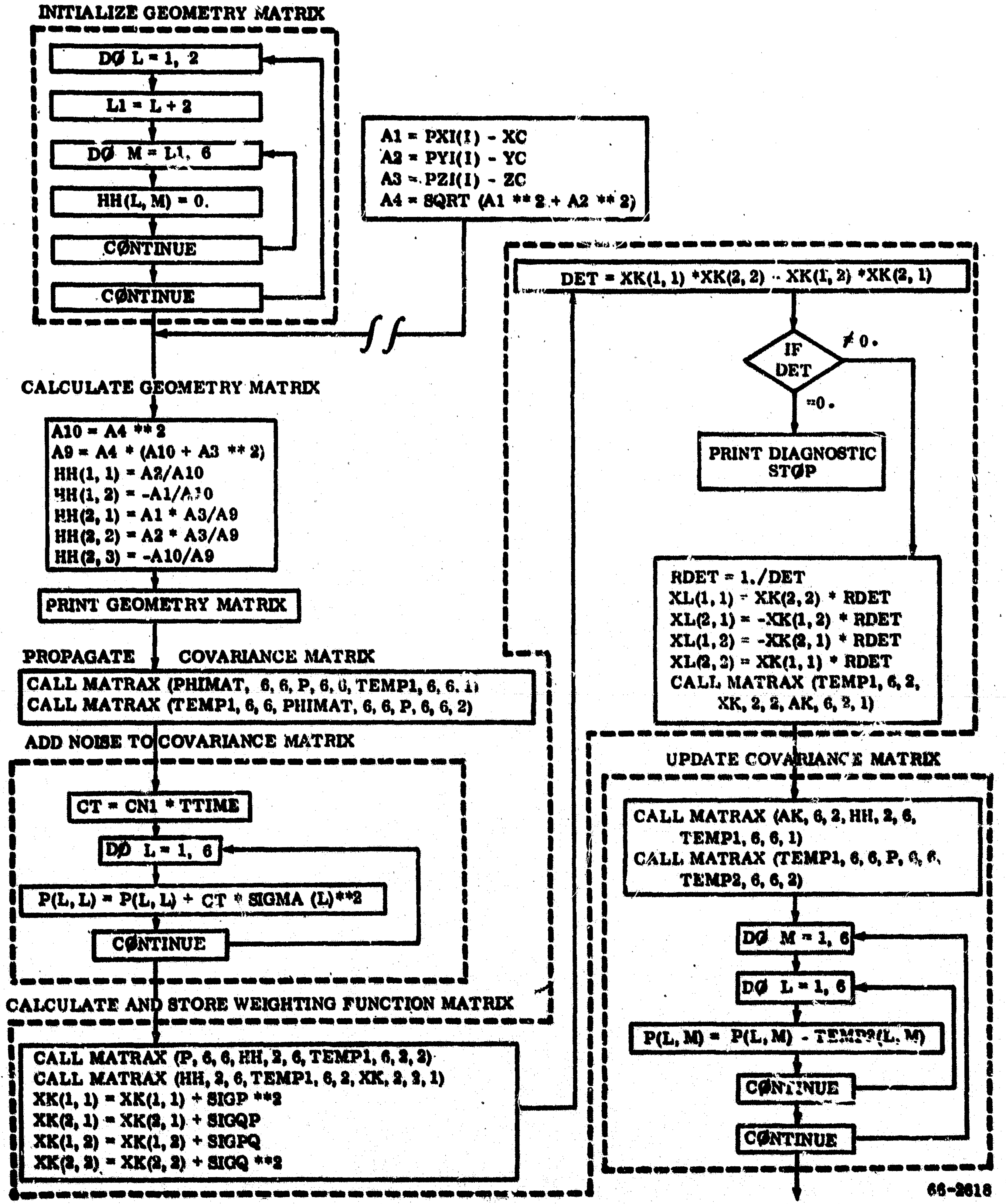
BLOCK V-2G CHECK TO SEE IF TRUTH AND NAVIGATED STATE VECTORS HAVE SAME REFERENCE COORDINATE CENTER - CHANGE TRUTH IF NECESSARY



```
N1T(I) = 2
A1 = PXI(I) - XR(2)
A2 = PYI(I) - YR(2)
A3 = PZI(I) - ZR(2)
PXI(I) = A1 * A11 + A2 * A21 + A3 * A31
PYI(I) = A1 * A12 + A2 * A22 + A3 * A32
PZI(I) = A1 * A13 + A2 * A23 + A3 * A33
A1 = XT(I) - XR(2)
A2 = YT(I) - YR(2)
A3 = ZT(I) - ZR(2)
XT(I) = A1 * A11 + A2 * A21 + A3 * A31
YT(I) = A1 * A12 + A2 * A22 + A3 * A32
ZT(I) = A1 * A13 + A2 * A23 + A3 * A33
A1 = VX(I) - VXR
A2 = VY(I) - VYR
A3 = VZ(I) - VZR
VX(I) = A1 * A11 + A2 * A21 + A3 * A31
VY(I) = A1 * A12 + A2 * A22 + A3 * A32
VZ(I) = A1 * A13 + A2 * A23 + A3 * A33
```

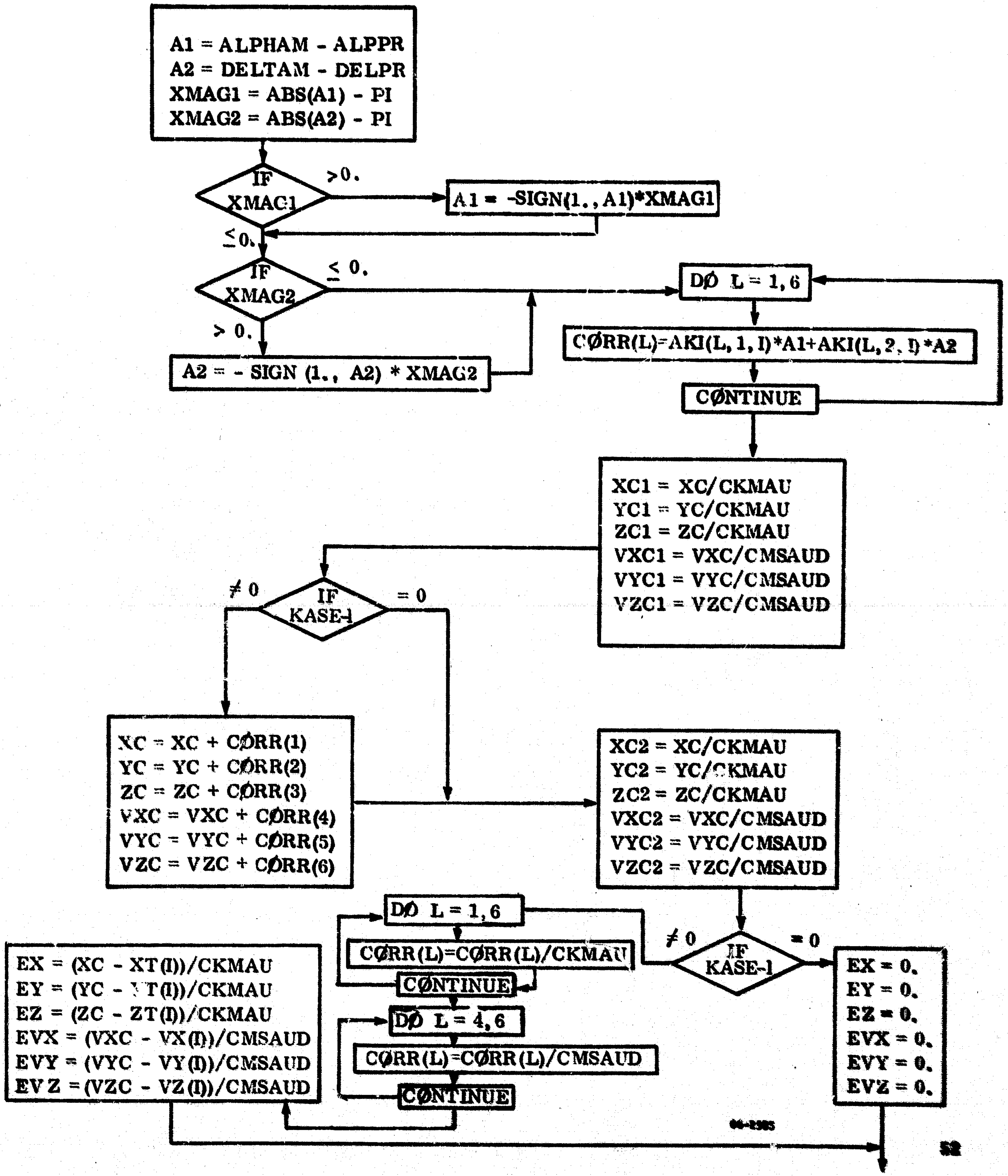
```
N1T(I) = 3
A1 = PXI(I)
A2 = PYI(I)
A3 = PZI(I)
PXI (I) = A1 * A11 + A2 * A12 + A3 * A13 + XR(2)
PYI (I) = A1 * A21 + A2 * A22 + A3 * A23 + YR(2)
PZI (I) = A1 * A31 + A2 * A32 + A3 * A33 + ZR(2)
A1 = XT(I)
A2 = YT(I)
A3 = ZT(I)
XT(I) = A1 * A11 + A2 * A12 + A3 * A13 + XR(2)
YT(I) = A1 * A21 + A2 * A22 + A3 * A23 + YR(2)
ZT(I) = A1 * A31 + A2 * A32 + A3 * A33 + ZR(2)
A1 = VX(I)
A2 = VY(I)
A3 = VZ(I)
VX(I) = A1 * A11 + A2 * A12 + A3 * A13 + VXR
VY(I) = A1 * A21 + A2 * A22 + A3 * A23 + VYR
VZ(I) = A1 * A31 + A2 * A32 + A3 * A33 + VZR
```

BLOCK V-2H. CALCULATE GEOMETRY AND CØVARIANCE MATRICES



66-2618

BLOCK V-2I UPDATE STATE VECTOR AND COMPUTE NAVIGATION ERRORS



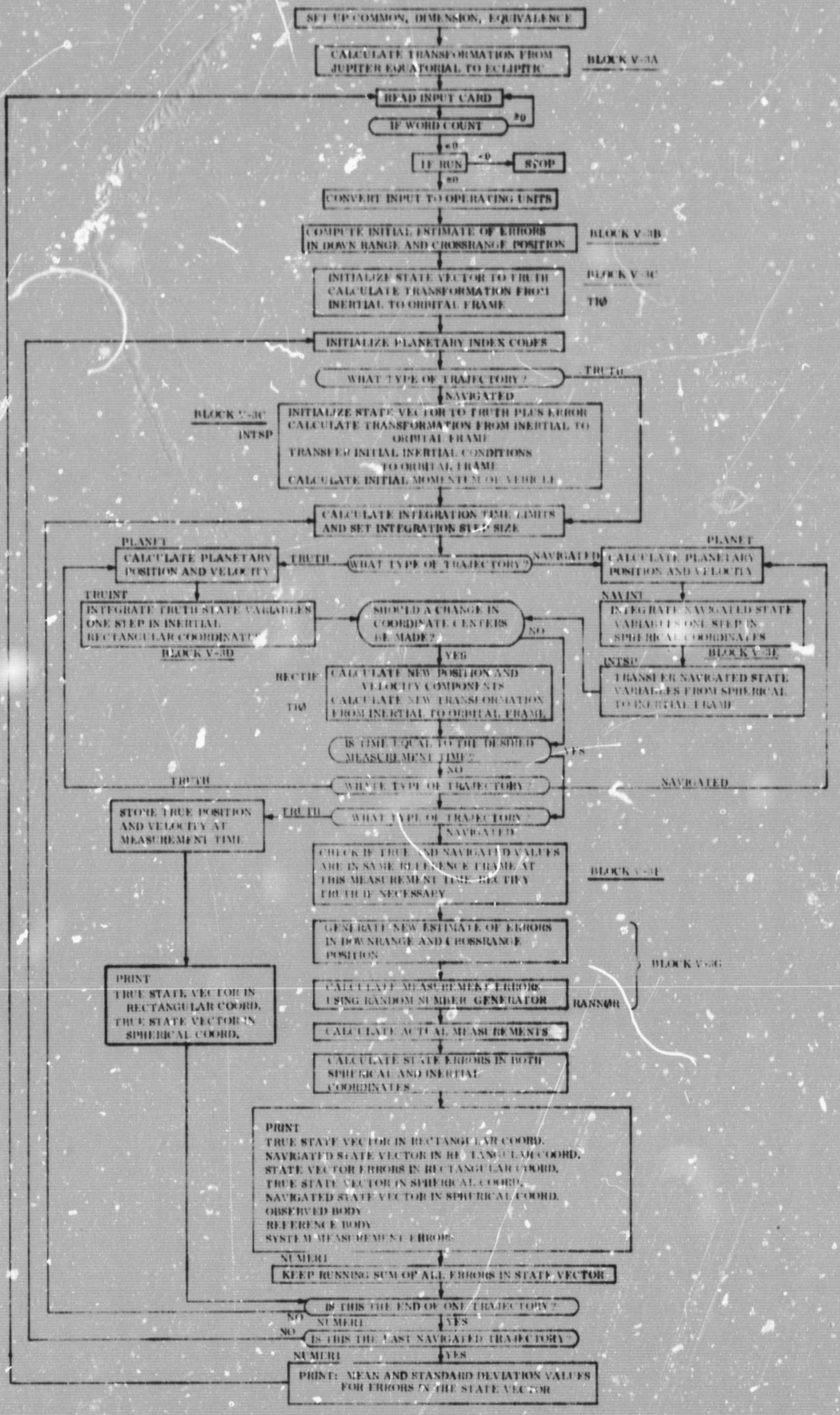
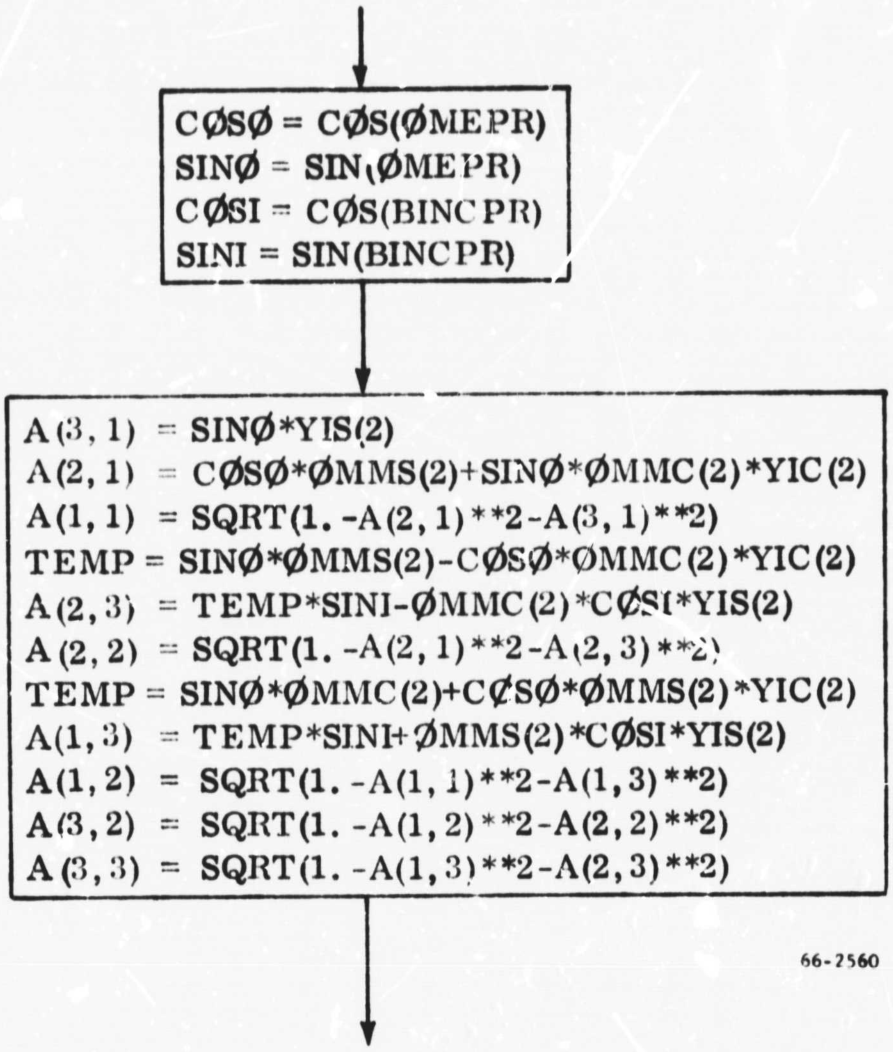


Figure V-3 Interplanetary Space Navigation Program

BLOCK V-3A CALCULATE TRANSFORMATION FROM JUPITER EQUATORIAL COORDINATE SYSTEM TO ECLIPTIC SYSTEM

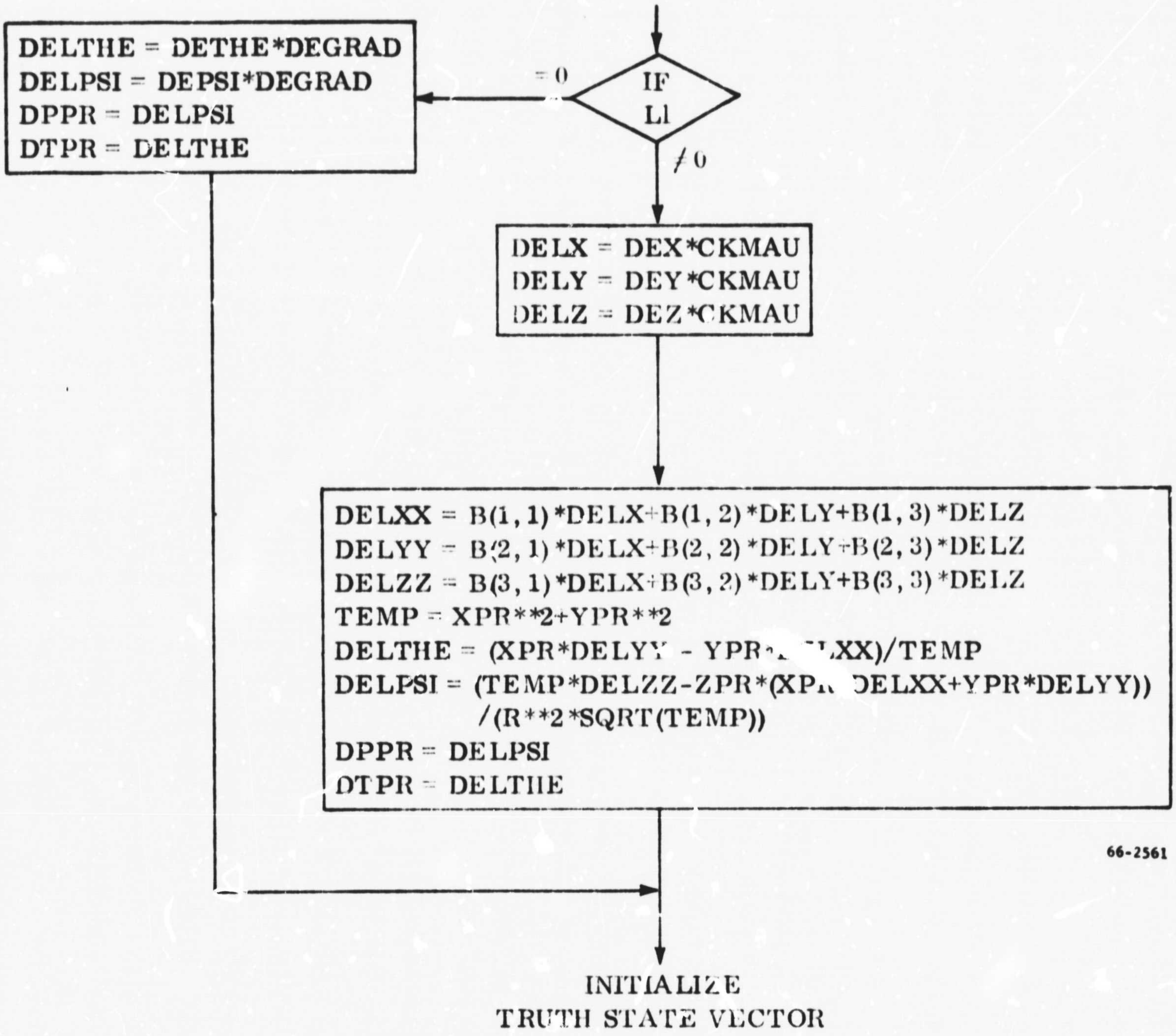


```
CØSØ = CØS(ØMEPR)
SINØ = SIN(ØMEPR)
CØSI = CØS(BINCPR)
SINI = SIN(BINCPR)
```

```
A(3,1)  = SINØ*YIS(2)
A(2,1)  = CØSØ*ØMMS(2)+SINØ*ØMMC(2)*YIC(2)
A(1,1)  = SQRT(1.-A(2,1)**2-A(3,1)**2)
TEMP = SINØ*ØMMS(2)-CØSØ*ØMMC(2)*YIC(2)
A(2,3)  = TEMP*SINI-ØMMC(2)*CØSI*YIS(2)
A(2,2)  = SQRT(1.-A(2,1)**2-A(2,3)**2)
TEMP = SINØ*ØMMC(2)+CØSØ*ØMMS(2)*YIC(2)
A(1,3)  = TEMP*SINI+ØMMS(2)*CØSI*YIS(2)
A(1,2)  = SQRT(1.-A(1,1)**2-A(1,3)**2)
A(3,2)  = SQRT(1.-A(1,2)**2-A(2,2)**2)
A(3,3)  = SQRT(1.-A(1,3)**2-A(2,3)**2)
```

66-2560

BLOCK V-3B COMPUTE INITIAL ESTIMATES OF ERRORS IN DOWNRANGE AND CROSSRANGE POSITION



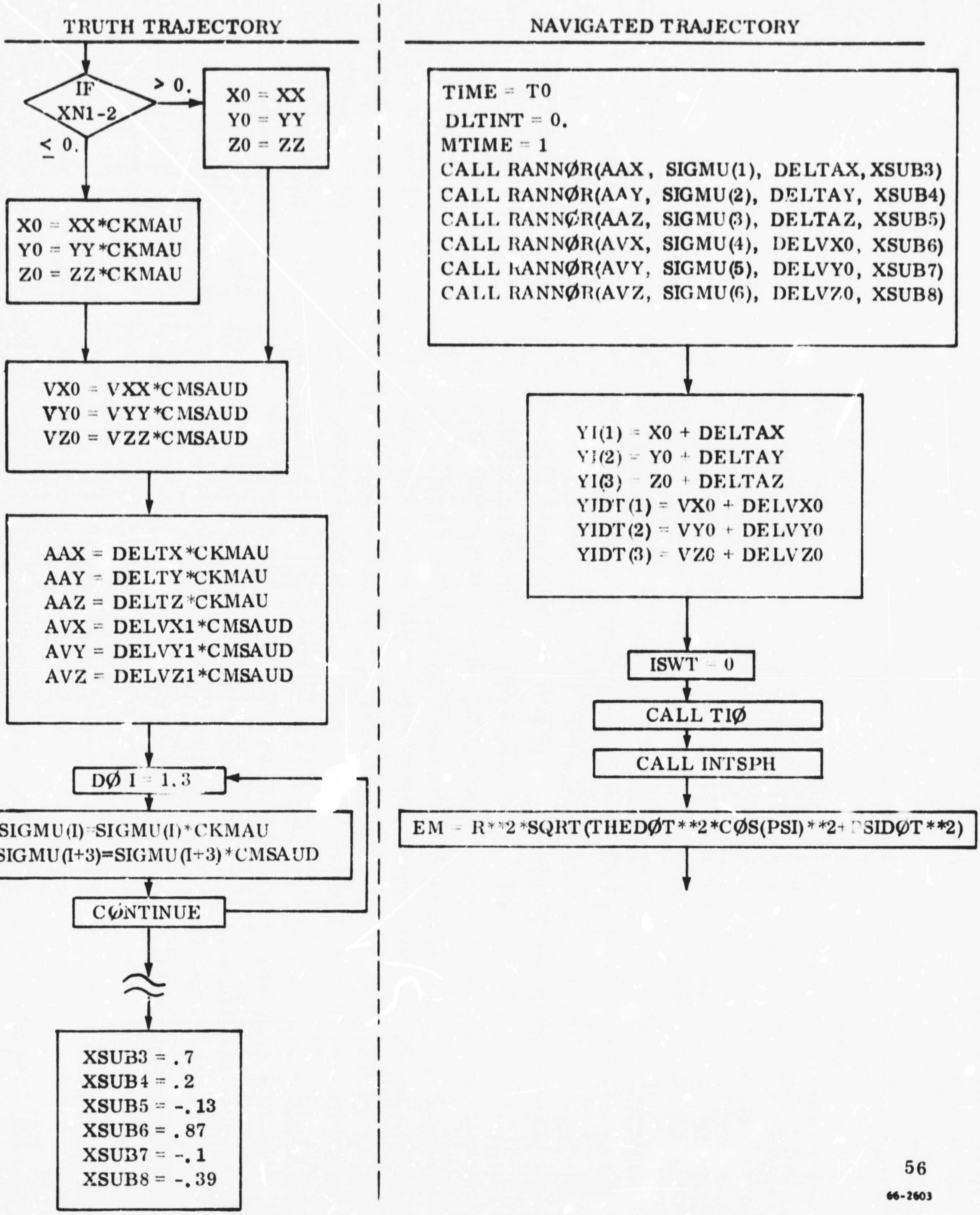
```
DELTHE = DETHE*DEGRAD
DELPSI = DEPSI*DEGRAD
DPPR = DELPSI
DTPR = DELTHE
```

```
DELX = DEX*CKMAU
DELY = DEY*CKMAU
DELZ = DEZ*CKMAU
```

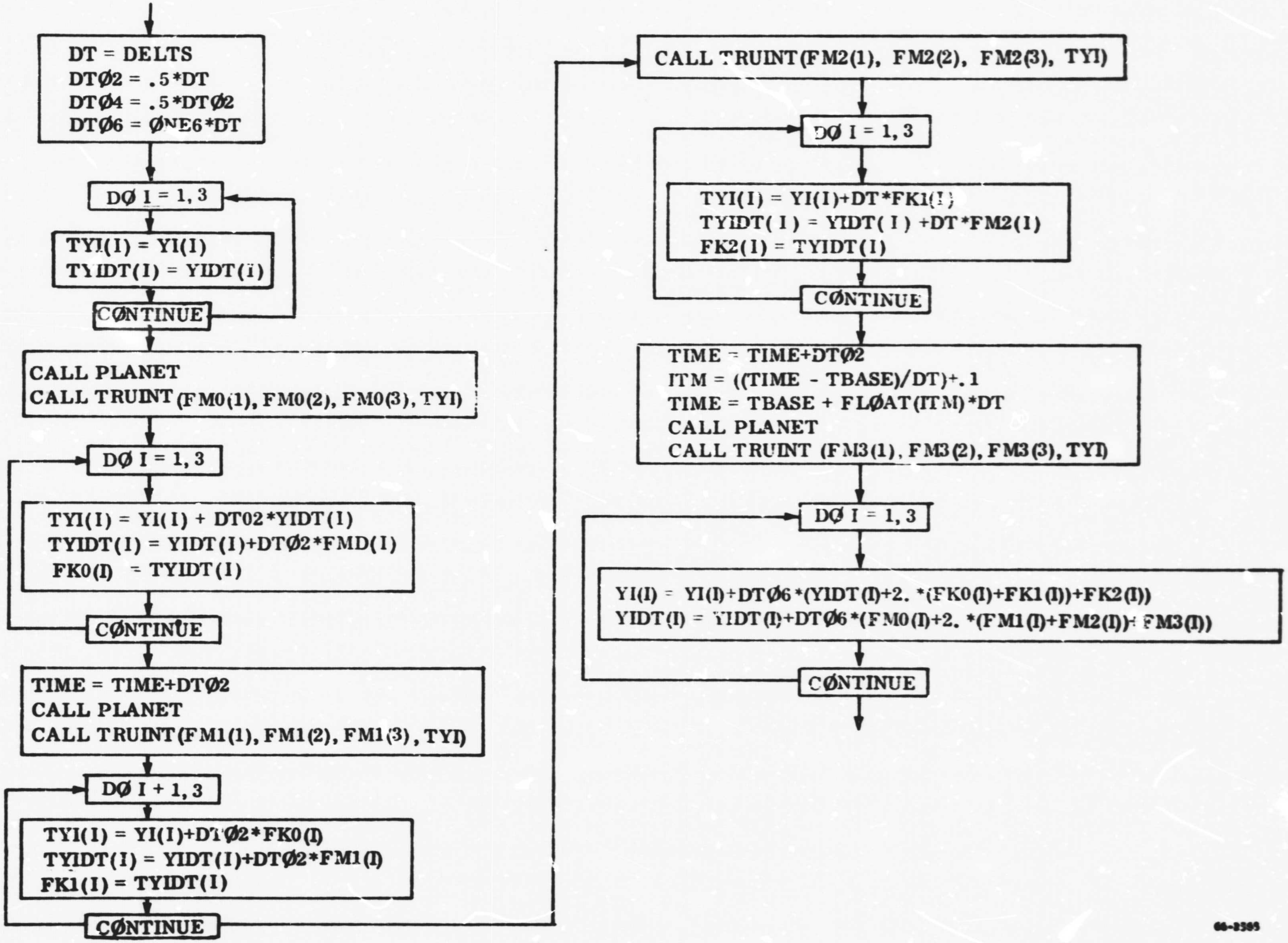
```
DELXX = B(1, 1)*DELX+B(1, 2)*DELY+B(1, 3)*DELZ
DELYY = B(2, 1)*DELX+B(2, 2)*DELY+B(2, 3)*DELZ
DELZZ = B(3, 1)*DELX+B(3, 2)*DELY+B(3, 3)*DELZ
TEMP = XPR**2+YPR**2
DELTHE = (XPR*DELYY - YPR*DELXX)/TEMP
DELPSI = (TEMP*DELZZ-ZPR*(XPR*DELXX+YPR*DELYY))
         /(R**2*SQRT(TEMP))
DPPR = DELPSI
DTPR = DELTHE
```

66-2561

BLOCK V-3C INITIALIZE STATE VECTOR

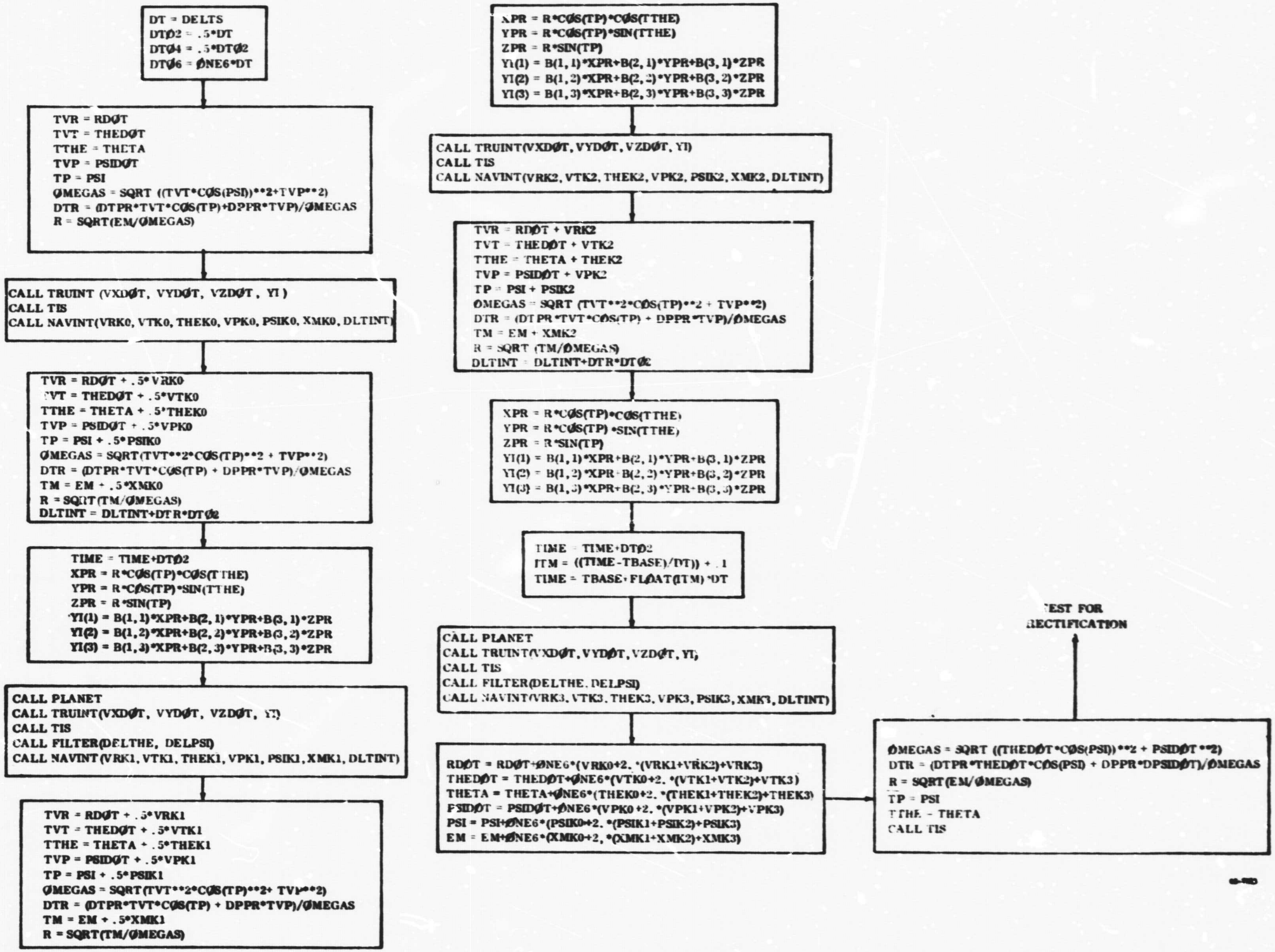


BLOCK V-3D INTEGRATE TRUTH STATE VARIABLES



66-2305

BLOCK V-3E INTEGRATE NAVIGATED STATE VARIABLES



```
DT = DELTS
DTØ2 = .5*DT
DTØ4 = .5*DTØ2
DTØ6 = ØNE6*DT
```

```
TVR = RDØT
TVT = THEDØT
TTHE = THETA
TVP = PSIDØT
TP = PSI
ØMEGAS = SQRT ((TVT*CØS(PSI))**2+TVP**2)
DTR = (DTPR*TVT*CØS(TP)+DPPR*TVP)/ØMEGAS
R = SQRT(EM/ØMEGAS)
```

```
CALL TRUINT (VXDØT, VYDØT, VZDØT, YI )
CALL TIS
CALL NAVINT(VRK0, VTK0, THEK0, VPK0, PSIK0, XMK0, DLTINT)
```

```
TVR = RDØT + .5*VRK0
TVT = THEDØT + .5*VTK0
TTHE = THETA + .5*THEK0
TVP = PSIDØT + .5*VPK0
TP = PSI + .5*PSIK0
ØMEGAS = SQRT(TVT**2*CØS(TP)**2 + TVP**2)
DTR = (DTPR*TVT*CØS(TP) + DPPR*TVP)/ØMEGAS
TM = EM + .5*XMK0
R = SQRT(TM/ØMEGAS)
DLTINT = DLTINT+DTR*DTØ2
```

```
TIME = TIME+DTØ2
XPR = R*CØS(TP)*CØS(TTHE)
YPR = R*CØS(TP)*SIN(TTHE)
ZPR = R*SIN(TP)
YI(1) = B(1,1)*XPR+B(2,1)*YPR+B(3,1)*ZPR
YI(2) = B(1,2)*XPR+B(2,2)*YPR+B(3,2)*ZPR
YI(3) = B(1,3)*XPR+B(2,3)*YPR+B(3,3)*ZPR
```

```
CALL PLANET
CALL TRUINT(VXDØT, VYDØT, VZDØT, YI)
CALL TIS
CALL FILTER(DELTHE, DELPSI)
CALL NAVINT(VRK1, VTK1, THEK1, VPK1, PSIK1, XMK1, DLTINT)
```

```
TVR = RDØT + .5*VRK1
TVT = THEDØT + .5*VTK1
TTHE = THETA + .5*THEK1
TVP = PSIDØT + .5*VPK1
TP = PSI + .5*PSIK1
ØMEGAS = SQRT(TVT**2*CØS(TP)**2+ TVP**2)
DTR = (DTPR*TVT*CØS(TP) + DPPR*TVP)/ØMEGAS
TM = EM + .5*XMK1
R = SQRT(TM/ØMEGAS)
```

```
XPR = R*CØS(TP)*CØS(TTHE)
YPR = R*CØS(TP)*SIN(TTHE)
ZPR = R*SIN(TP)
YI(1) = B(1,1)*XPR+B(2,1)*YPR+B(3,1)*ZPR
YI(2) = B(1,2)*XPR+B(2,2)*YPR+B(3,2)*ZPR
YI(3) = B(1,3)*XPR+B(2,3)*YPR+B(3,3)*ZPR
```

```
CALL TRUINT(VXDØT, VYDØT, VZDØT, YI)
CALL TIS
CALL NAVINT(VRK2, VTK2, THEK2, VPK2, PSIK2, XMK2, DLTINT)
```

```
TVR = RDØT + VRK2
TVT = THEDØT + VTK2
TTHE = THETA + THEK2
TVP = PSIDØT + VPK2
TP = PSI + PSIK2
ØMEGAS = SQRT (TVT**2*CØS(TP)**2 + TVP**2)
DTR = (DTPR*TVT*CØS(TP) + DPPR*TVP)/ØMEGAS
TM = EM + XMK2
R = SQRT (TM/ØMEGAS)
DLTINT = DLTINT+DTR*DTØ2
```

```
XPR = R*CØS(TP)*CØS(TTHE)
YPR = R*CØS(TP) *SIN(TTHE)
ZPR = R*SIN(TP)
YI(1) = B(1,1)*XPR+B(2,1)*YPR+B(3,1)*ZPR
YI(2) = B(1,2)*XPR+B(2,2)*YPR+B(3,2)*ZPR
YI(3) = B(1,3)*XPR+B(2,3)*YPR+B(3,3)*ZPR
```

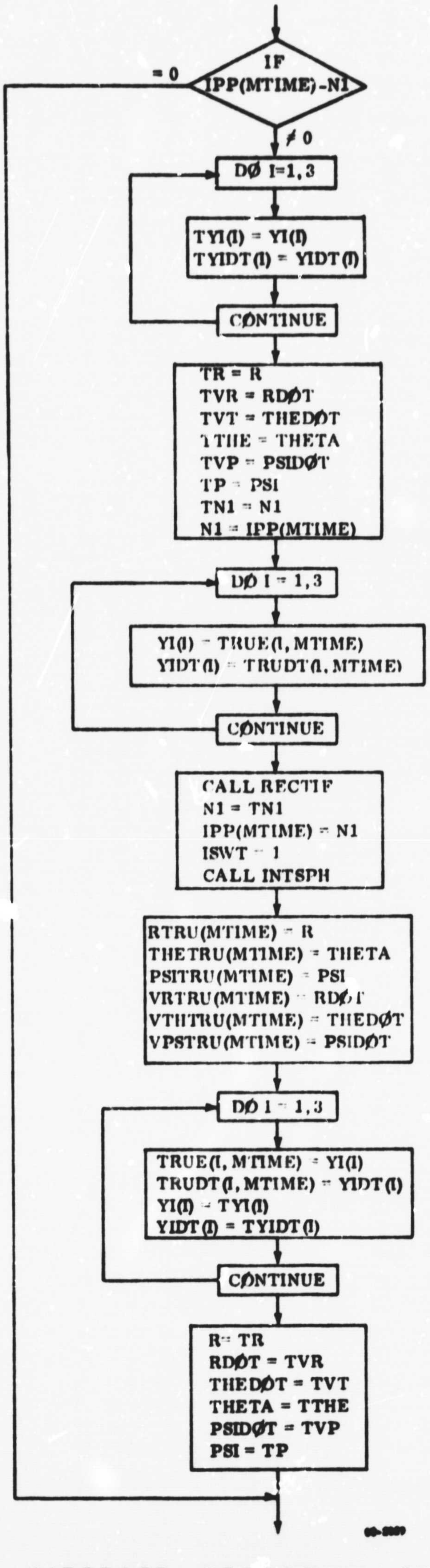
```
TIME = TIME+DTØ2
ITM = ((TIME-TBASE)/DT)) + .1
TIME = TBASE+FLØAT(ITM)*DT
```

```
CALL PLANET
CALL TRUINT(VXDØT, VYDØT, VZDØT, YI)
CALL TIS
CALL FILTER(DELTHE, DELPSI)
CALL NAVINT(VRK3, VTK3, THEK3, VPK3, PSIK3, XMK3, DLTINT)
```

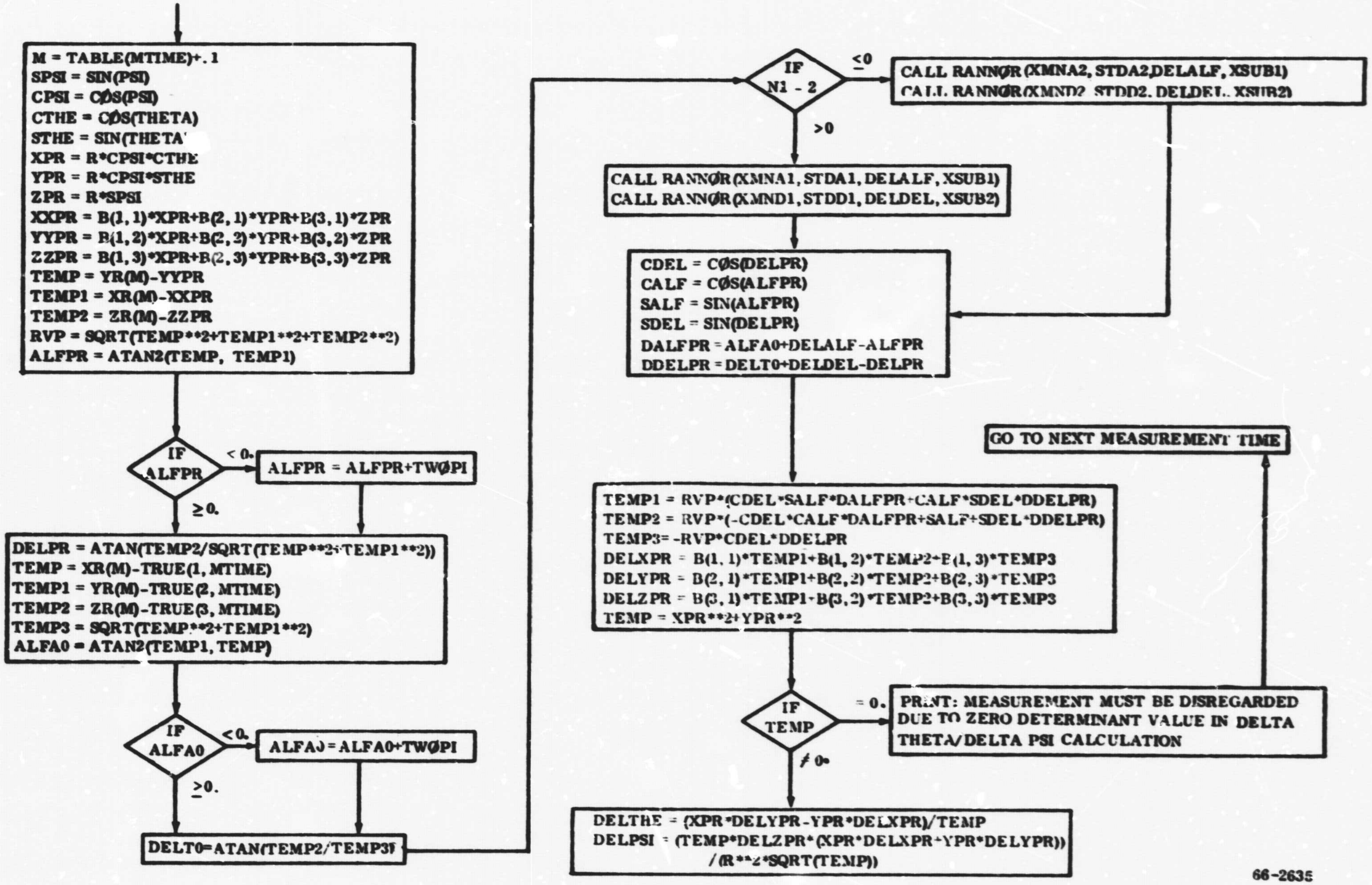
```
RDØT = RDØT+ØNE6*(VRK0+2.*(VRK1+VRK2)+VRK3)
THEDØT = THEDØT+ØNE6*(VTK0+2.*(VTK1+VTK2)+VTK3)
THETA = THETA+ØNE6*(THEK0+2.*(THEK1+THEK2)+THEK3)
PSIDØT = PSIDØT+ØNE6*(VPK0+2.*(VPK1+VPK2)+VPK3)
PSI = PSI+ØNE6*(PSIK0+2.*(PSIK1+PSIK2)+PSIK3)
EM = EM+ØNE6*(XMK0+2,*(XMK1+XMK2)+XMK3)
```

```
ØMEGAS = SQRT ((THEDØT*CØS(PSI))**2 + PSIDØT**2)
DTR = (DTPR*THEDØT*CØS(PSI) + DPPR*DPSIDØT)/ØMEGAS
R = SQRT(EM/ØMEGAS)
TP = PSI
TTHE = THETA
CALL TIS
```

BLOCK V-3F CHECK IF TRUE AND NAVIGATED STATE VECTORS HAVE THE SAME REFERENCE - CORRECT TRUTH IF NECESSARY



BLOCK V-3G GENERATE NEW ESTIMATE OF ERRORS IN DOWNRANGE AND CROSSRANGE POSITION



66-2635

VI. COMMON

All three programs described herein used labeled common data storage areas to make more efficient use of the data handling capabilities of the computer. Each program has one labeled common area. The variables associated with each area are listed in Table VI-I. Definitions for each variable are given in the respective program directories (Section VII).

TABLE VI-I

LABELED CØMMØN VARIABLES

Program	Label	Variables			
A. N-BØDY	Z	1. XC	19. TM1	37. ØMM(10)	55. A32(2)
(N-BØDY)		2. YC	20. J	38. ØMMS(10)	56. A33(2)
DATA		3. ZC	21. N	39. ØMMC(10)	57. XNN(10)
PLANET		4. VX	22. K	40. CRPP(10)	58. RAQ
SETINT		5. VY	23. IALFA	41. YIS(10)	59. ALPHJ(6)
INTEG		6. VZ	24. NPRINT	42. YIC(10)	60. BETAJ(6)
UPDAT		7. Y(336)	25. ICN1	43. DELTS	61. GAMJ(6)
DETERM		8. YP(336)	26. ICN2	44. IBETA	62. D(6)
TRAMAT		9. R	27. ICN3	45. RKTME	63. PH1(6)
		10. HØ	28. IGAM	46. NU	64. PH2(6)
		11. H	29. XR(10)	47. I1	65. PH3(6)
		12. W(42)	30. YR(10)	48. A11(2)	66. PH4(6)
		13. INDR	31. ZR(10)	49. A12(2)	67. PH5(6)
		14. ISWT1	32. VXR(2)	50. A13(2)	68. PH6(6)
		15. ISWT2	33. VYR(2)	51. A21(2)	69. XJ2(2)
		16. ISWT3	34. VZR(2)	52. A22(2)	70. XJ3(2)
		17. EPS(10)	35. AA(10)	53. A23(2)	71. XJ4(2)
		18. XMU(10)	36. EE(10)	54. A31(2)	72. TRAN
B. LØF	Z	1. XC	22. A22	43. IGAM	64. ISWT2
(MAIN)		2. YC	23. A23	44. NU	65. ISWT3
DETERM		3. ZC	24. A31	45. Y(48)	66. HØ
PLANET		4. R	25. A32	46. YP(48)	67. H
NUMER		5. VXC	26. A33	47. W(6)	68. DELTS
INTEG		6. VYC	27. XR(10)	48. TM1	69. PHIMAT(6, 6)
DATA		7. VZC	28. YR(10)	49. EX	70. PHDØT(6, 6)
ACCEL		8. XMU(10)	29. ZR(10)	50. EY	71. DELTPH
PHINT1		9. EPS(10)	30. VXR	51. EZ	72. VXDØT
PHINT2		10. XNN(10)	31. VYR	52. EVX	73. VYDØT
SETINT		11. EE(10)	32. VZR	53. EVY	74. VZDØT
UPDAT		12. ØMM(10)	33. IPLAN	54. EVZ	75. IBETA
		13. CRPP(10)	34. KMAX	55. KK(7)	76. RKTME
		14. ØMMS(10)	35. I	56. T(100)	77. ISWT10
		15. ØMMC(10)	36. NMAX	57. TIME	78. AA(10)
		16. YIS(10)	37. KASE	58. NPRINT	79. AUNM
		17. YIC(10)	38. J	59. ICN1	80. XJ2
		18. A11	39. N	60. ICN2	81. XJ4
		19. A12	40. K	61. ICN3	82. REFRAC
		20. A13	41. I1	62. INDR	83. REQ(10)
		21. A21	42. IALFA	63. ISWT1	
C. ISN	CØM	1. YI(3)	20. ZPR	39. APSI	58. LMAX
(MAIN)		2. ZR(10)	21. XMU(10)	40. AA(10)	59. EPSI
NUMER1		3. KK(7)	22. DTPR	41. ØMMC(10)	60. YR(10)
FILTER		4. THEDØT	23. TVP	42. TM1	61. A(3, 3)
INSTPH		5. YPR	24. ATHE	43. MTIME	62. RDØT
TISS		6. IP	25. XJ4	44. ETHE	63. XPR
PLANET		7. DUMMY(250)	26. ØMMS(10)	45. ISWT	64. IMAX
TIØ		8. TTHE	27. EPS(10)	46. XR(10)	65. AZSP
NAVINT		9. AR	28. TIME	47. VZR	66. TVT
RECTIF		10. XJ2	29. ER	48. PSI	67. DT
TRUINT		11. CRPP(10)	30. EVPSI	49. C(3, 3)	68. REQ(10)
		12. XNN(10)	31. LP	50. TWØPI	69. ØMM(10)
		13. VZDØT	32. VYR	51. AYSP	70. YIC(10)
		14. ERRIDT(3)	33. THETA	52. TVR	71. VYDØT
		15. EVTHE	34. B(3, 3)	53. TM	72. ERRI(3)
		16. YIDT(3)	35. IND	54. ØMEGAS	73. EVR
		17. VXR	36. AXSP	55. EE(10)	
		18. R	37. DPPR	56. YIS(10)	
		19. PSIDØT	38. TP	57. VXDØT	

66-2647

VII. SUBROUTINES

A. General Description

The mathematical subroutines used by the three programs are listed in Table VII-I together with the associated calling sequence for each. Details and flow charts for each subroutine are given in Section VII-B. It is assumed that most of the subroutine writeups in Section VII-B are self-explanatory and do not need further clarification. However, in some instances, additional descriptive material may be helpful in the understanding of the individual roles of those subroutines which are collectively used to perform a single function or of those in which the relationship of the subroutine function to the main problem may not be clear. The following paragraphs provide such supplementary information.

In the N-BODY and LOF programs, the function of numerical integration is spread over three subroutines (as opposed to only one part of the ISN main program). As designed, the three subroutines (SETINT, INTEG, and UPDAT) together with associated main program logic offer an input option of either a Runge-Kutta integration scheme with constant step size or an Adams 4-point predictor-corrector scheme with either a constant step size or a variable step size with automatic error control. The Adams 4-point integration formulas are easily adapted to procedures for estimating the step-by-step truncation error. The error control portion of the integration process compares the predicted and corrected values obtained from such formulas and changes the integration interval according to the results of the comparison. This is done to minimize round-off error and computing time and still maintain a low level of truncation error. When the Adams 4-point formulas are used, the differential equations are solved only twice per integration step as compared with four times for the Runge-Kutta scheme. However, the Adams formulas require four past values of the dependent variables in order to advance one integration step; whereas, the Runge-Kutta scheme needs only two. Consequently, the Adams 4-point method is not self-starting and the first four points must be generated using the Runge-Kutta technique.

In general, the subroutine INTEG contains the integration formulas and UPDAT stores the variables and places them in the correct locations for the next integration step. INTEG is called once for each dependent variable during any given pass through the time loop, whereas UPDAT is called only once at the completion of the time loop.

It is noted that twice as many values as are normally required in the Adams 4-point method are reserved in the integrated variable tables, Y(I) and YP(I), for each variable to be integrated, i.e., 8 spaces. The reason for this is that if the integration step size is to be doubled, only every other value in the table is used to provide the required past values. A comparable situation does not exist, however, if integration step size is to be halved. In that event, the routine re-initializes and starts over again using Runge-Kutta formulas.

Two main tables in UPDAT are allocated for data storage. Each table is composed of blocks of eight spaces with the current value of a particular variable appearing in the first location of the eight spaces reserved for it. The Y(I) table contains blocks of eight spaces reserved for the integrated variables and the YP(I) table contains the first derivatives of the variables. The value of the first derivative and the variable for any given time are found at corresponding values of the index I in each table.

TABLE VII-I

PROGRAM SUBROUTINES AND CALLING SEQUENCES

N-BØDY	LØF	ISN
1. ZPRN(DUMMY)	ZPRN(DUMMY)	1. DPRINT(X)
2. AGØNY(ARG1, ARG2, ANS)	2. AGØNY(ARG1, ARG2, ANS)	2. RECTIF
3. PLANET(TIME, KK)	3. PLANET	3. PLANET
4. SETINT(TIME, XMTHD, VARBH)	4. SETINT(XMTHD, VARBH)	4. TIS
5. INTEG	5. INTEG	5. TIØ
6. UPDAT(TIME)	6. UPDAT	6. NAVINT(VR, VT, TH, VP, PPSI, XM, DLTINT)
7. DETERM	7. DETERM	7. TRUINT(VXDT, VYDT, VZDT, Y)
8. TRAMAT(KK)	8. NUMER	8. NUMER1
	9. RANNØR(XMN, STDV, ANS, XSUBI)	9. RANNØR(XMN, STDV, ANS, XUSBI)
	10. RANDØM(XSUBI)	10. RANDØM(XSUBI)
	11. PHINT1	11. INTSPH
	12. PHINT2	12. SPHINT
	13. ACCEL	13. FILTER(DELTHE, DELPSI)
	14. MATRAX(A, K1, K2, B, K3, K4, C, K5, K6, M)	

Most of the initialization of INTEG and UPDAT is done in the subroutine, SETINT. However, in addition to the initialization contained in subroutine SETINT, the integration weighting factors, W(K), corresponding to each of the integrated variables are set equal to unity in the pertinent main programs. These weighting factors $(0. < W(K) \leq 1.)$ determine the relative effectiveness of each integrated variable in the error control process. Variables with low weighting factors affect the error control process less and hence have a greater variation of acceptable accuracy. The computer multiplies the minimum of the percentage difference and the absolute difference between the predicted and corrected integrated value of each variable by the corresponding weighting factor to obtain a measure of the truncation error for the error control process (see the flow diagram for INTEG in Section VII-B).

If it becomes desirable at any time to add integrated variables to either the N-BODY or LOF programs, an integration accuracy tolerance must be provided for each new variable. The tolerances should be given in terms of allowable absolute differences for those variables with expected values less than unity, and in terms of percentage differences for those with expected values greater than unity. Once supplied with the appropriate tolerances, the programmer can determine the correct W(K) from the following equation:

$$W(K) = \frac{.001}{\text{tolerance}(K)}$$

Because the Adams 4-point integration scheme is the more complex of the two integration options and therefore less likely to be incorporated into a flight computer, the option of choosing it was eliminated from both navigation programs. This was accomplished in the LOF program by internally initializing the pertinent integration package control variables and in the ISN program by mechanizing only the fourth order Runge-Kutta integration scheme. See Reference 7 for the mathematical details of the indicated integration techniques.

Although not part of the essential problem logic, subroutine DETERM was included in both the N-BODY and LOF programs to serve as a means of checking the numerical integration accuracy of the transition matrix calculation. The subroutine calculates and prints the determinant of the transition matrix each time it is called by the main program. As shown by Stern in Reference 8, the determinant of the transition matrix should always be unity for those situations where the matrix of partial derivatives of the vehicle's acceleration with respect to position is symmetrical. To the degree of sophistication of the subject programs, this condition is true everywhere except in the Earth's sphere of influence where the J_3 zonal harmonic of the Earth's gravitational field causes the partial derivative matrix to deviate slightly from a symmetrical condition. The technique for computing the determinant of the 6 x 6 matrix was obtained from Reference 9.

B. Subroutine Flow Diagrams and Logic

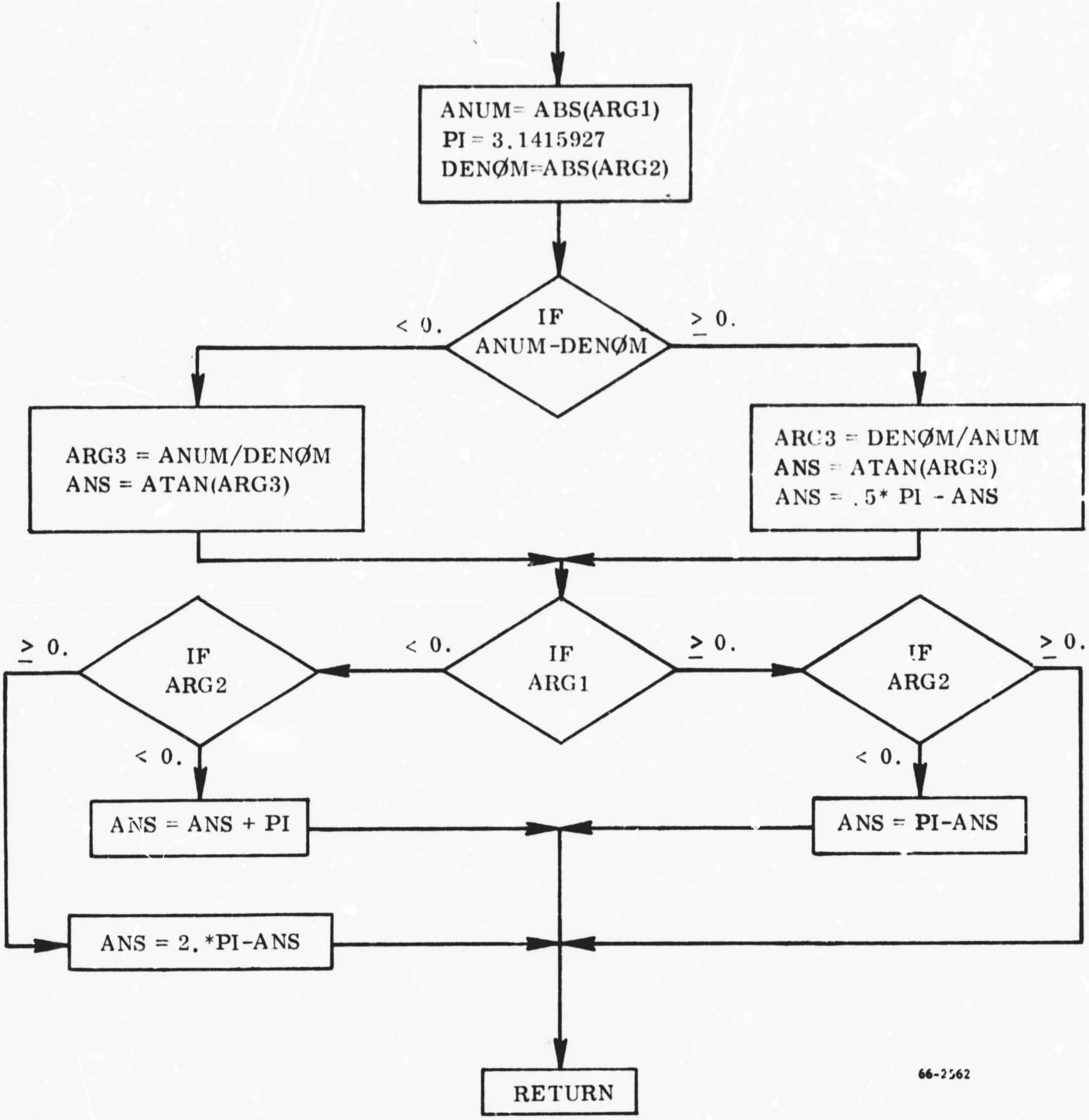
TABLE OF CONTENTS

Subroutine Page No.

SUBRØUTINE AGØNY (ARG1, ARG2, ANS)

Purpose: To calculate $ANS = \tan^{-1}\left[\frac{ARG1}{ARG2}\right]$ where $0 \le ANS \le 2\pi$

Program(s): N-BØDY, LØF

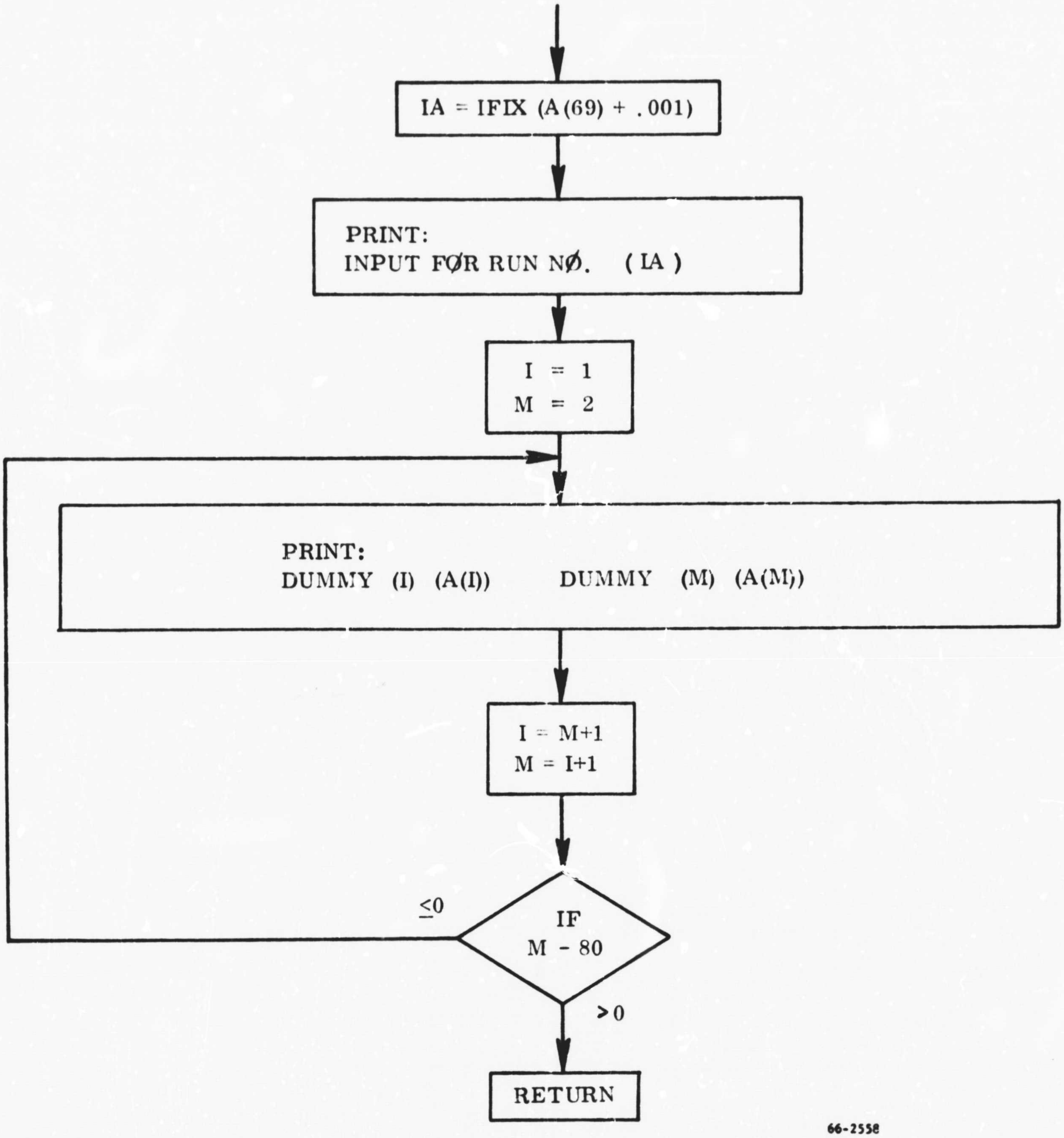


66-2562

SUBRØUTINE ZPRN(A)

Purpose: To print input data

Program(s): N-BØDY



66-2558

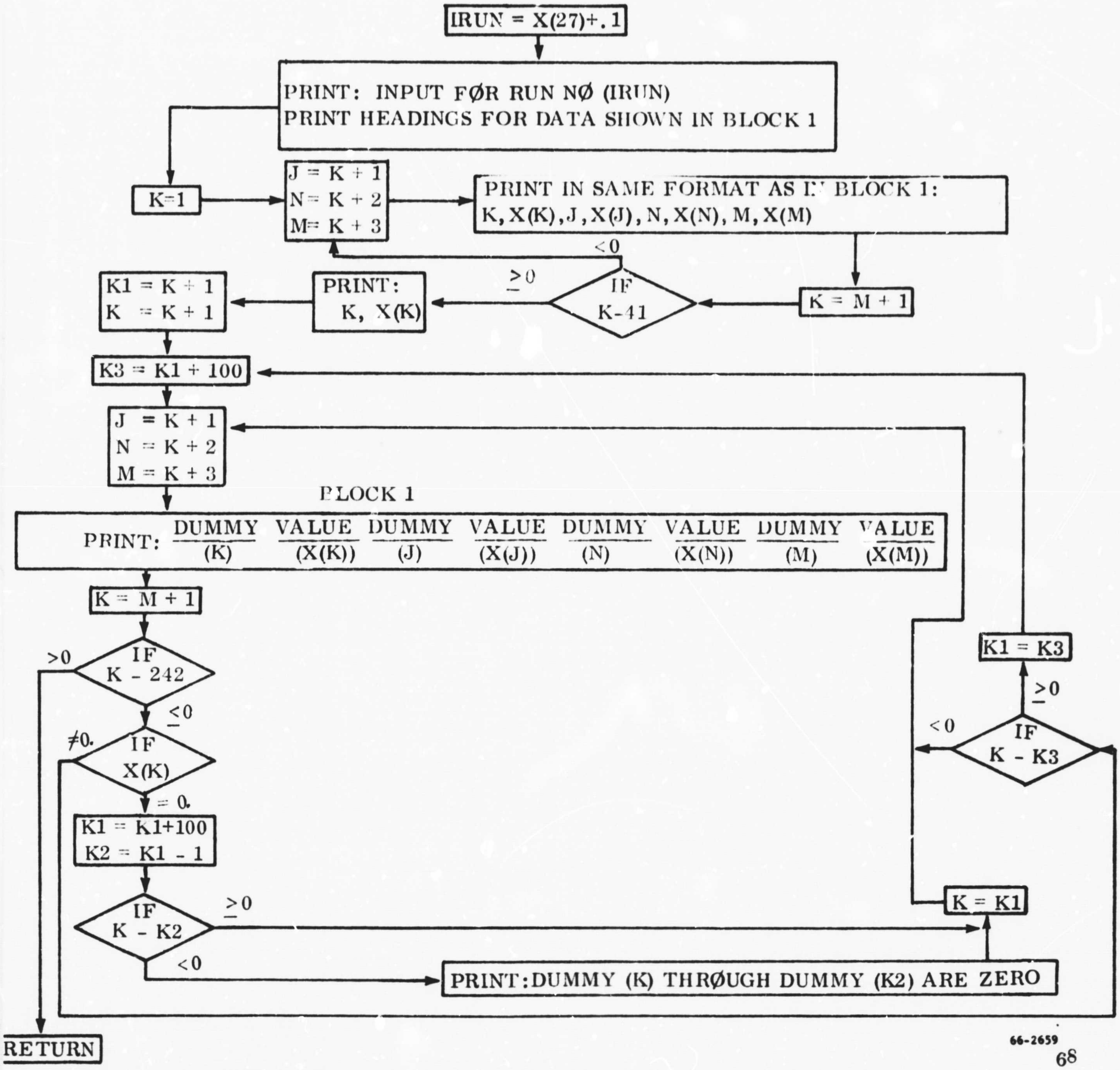
SUBROUTINE ZPRIN(X)

Purpose: To Print Input Data

Program(s): L0F

Flow Diagram Key:

Block 1 - Format for Printing Out the Input



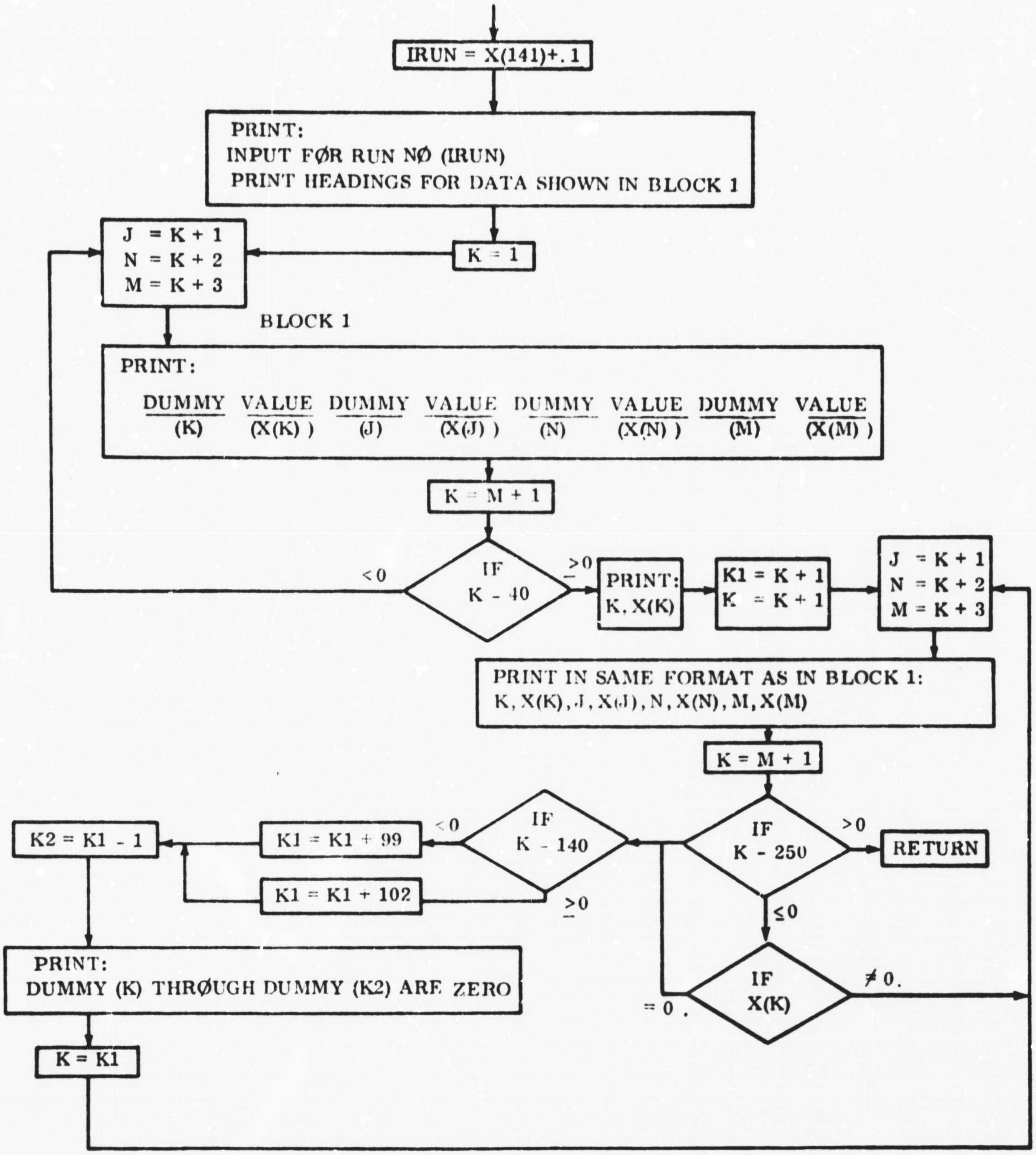
SUBRØUTINE DPRINT(X)

Purpose: To print out all the input in one block

Program(s): ISN

Flow Diagram Key:

Block 1 - Format for printing out the input.



66-2646

SUBRØUTINE PLANET (TIME, KK) (N-BØDY)
SUBRØUTINE PLANET (LØF, ISN)

Purpose:	To compute the planetary positions and velocities as functions of time.
Program(s):	LØF, N-BØDY, ISN

Flow Diagram Key:

Block 1 - Compute planetary eccentric anomalies.

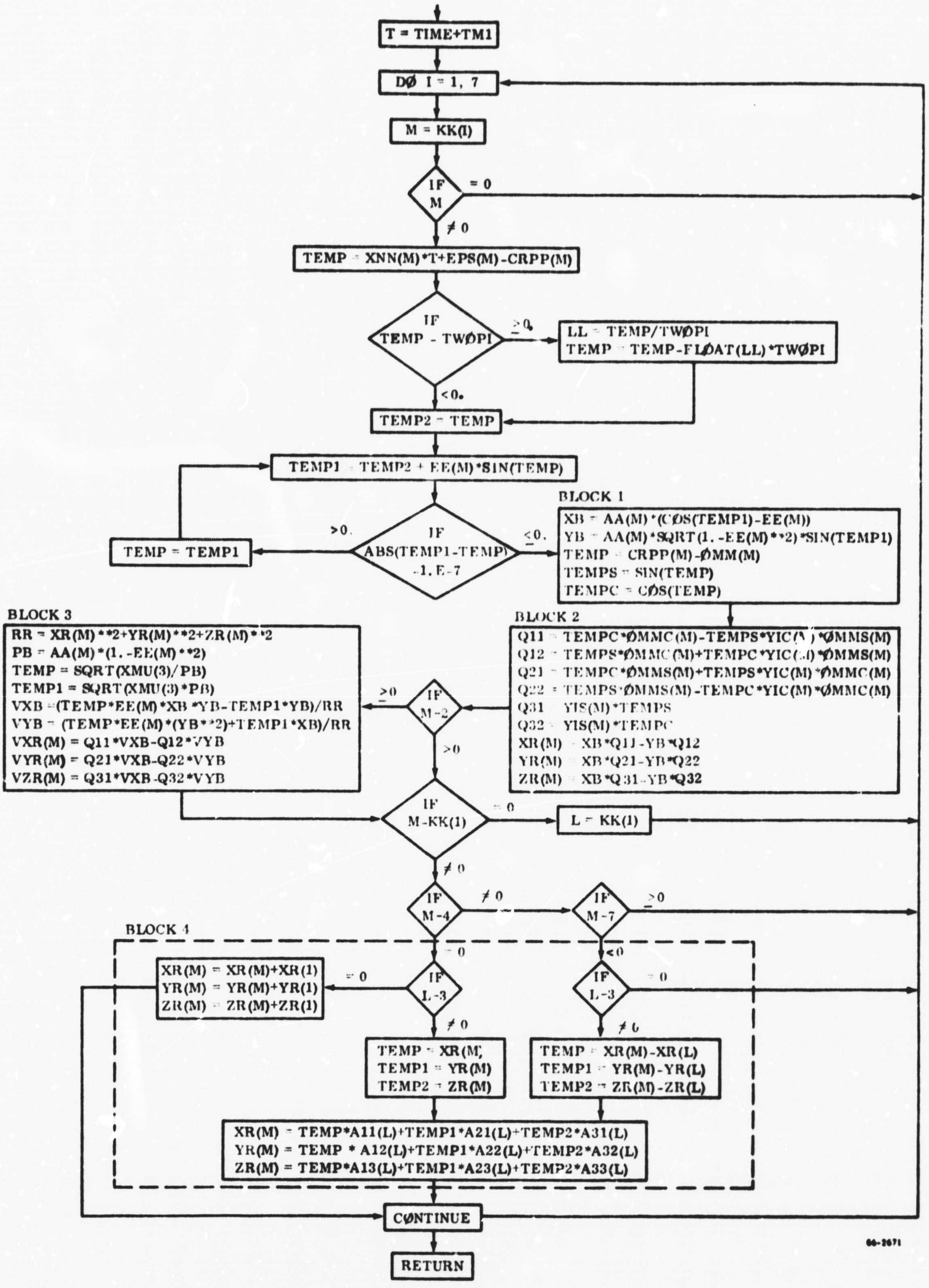
Block 2 - Calculate position relative to Sun.

Block 3 - Calculate planetary velocity.

Block 4 - Calculate position relative to central body.

SUBRØUTINE PLANET (TIME,KK)

SUBRØUTINE PLANET



SUBROUTINE SETINT (TIME, XMTHD, VARBH)

Purpose: To initialize integration subroutines

Programs: N-BODY, LOF

Flow Diagram Key:

Block 1 - Initialize switches and counters common to both Adams 4-point and Runge-Kutta schemes

Block 2 - Initialize integration tables for initial state vector

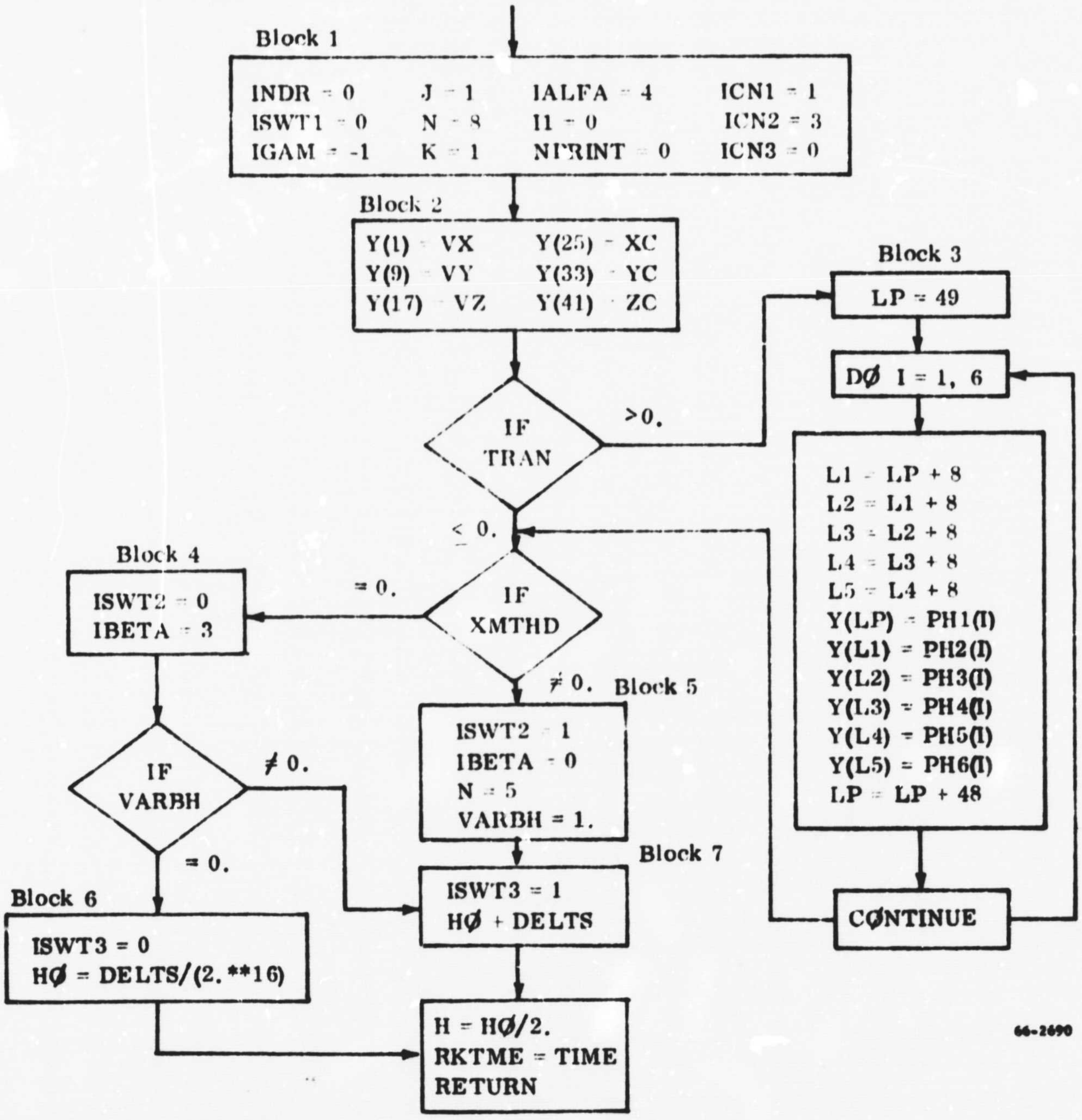
Block 3 - Initialize integration tables for transition matrix elements (omitted in LOF)

Block 4 - Initialize switches for Adams 4-point method

Block 5 - Initialize switches for Runge-Kutta method

Block 6 - Initialize switches for variable step size

Block 7 - Initialize switches for constant step size



SUBRØUTINE INTEG

Purpose: To perform the numerical integration of variables

Program(s): N-BØDY, LØF

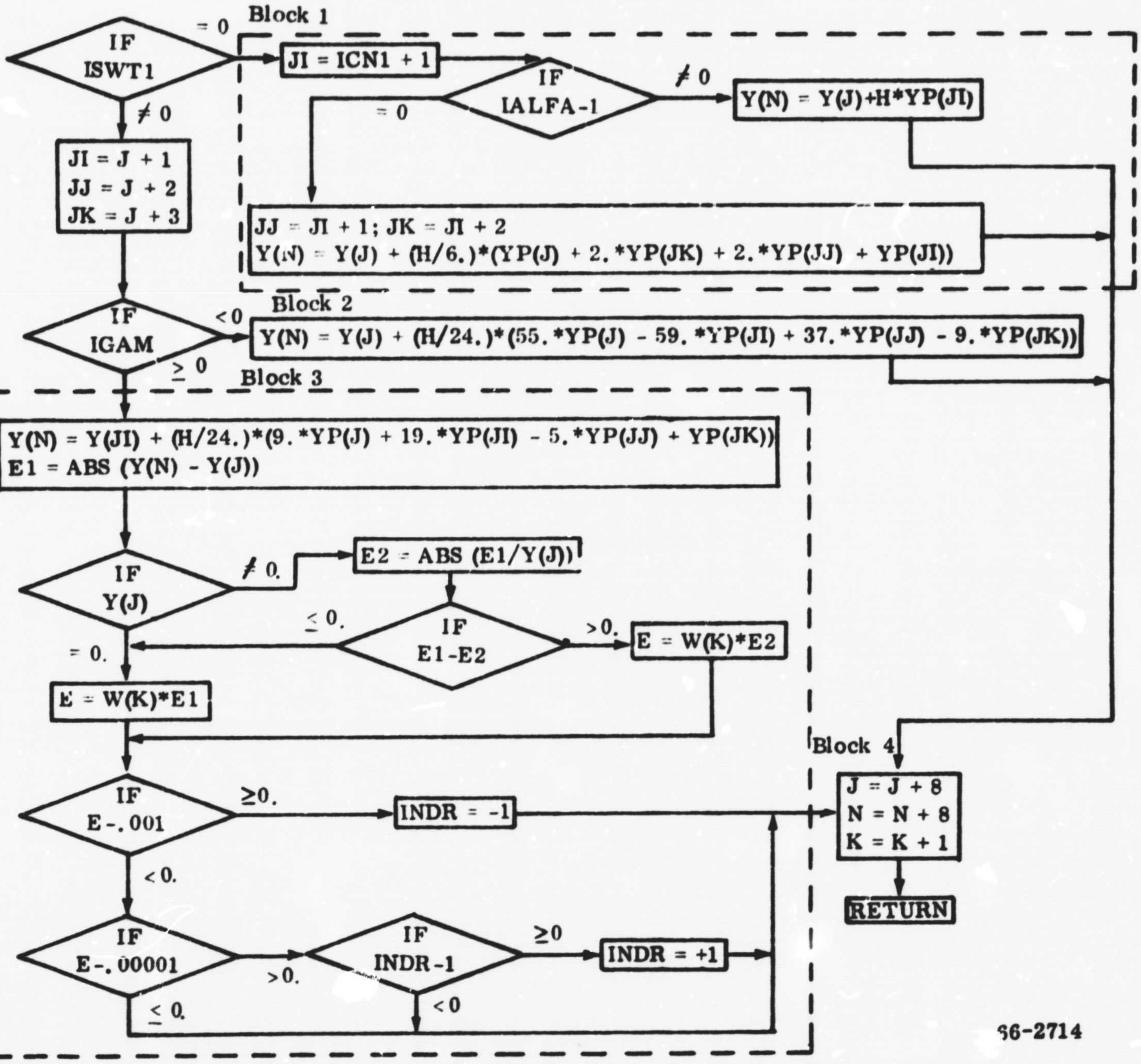
Flow Diagram Key:

Block 1 - Perform fourth-order Runge-Kutta integration

Block 2 - Calculate "predictor" estimate of predictor-corrector integration scheme

Block 3 - Calculate "corrector" estimate of predictor-corrector integration scheme and check to see if error is within preset tolerances

Block 4 - Set indices for next integrated variable



SUBR0UTINE UPDAT (TIME) (N-B0DY)
SUBR0UTINE UPDAT (L0F)

Purpose: To store and manipulate data in integration tables, control integration step size changes, and signal valid print times to main program

Program(s): N-B0DY, L0F

Flow Diagram Key:

Block 1 - Control integration switching for Adams 4-point integration scheme and provide for doubling the integration step size

Block 2 - Control integration switching for Runge-Kutta integration scheme

Block 3 - Reset integration tables to start of last initial Runge-Kutta point - prepare to halve integration step

Block 4 - Reset integration tables to last integrated point - prepare to halve integration step

Block 5 - Rearrange integration tables to contain every second data point - step has been doubled

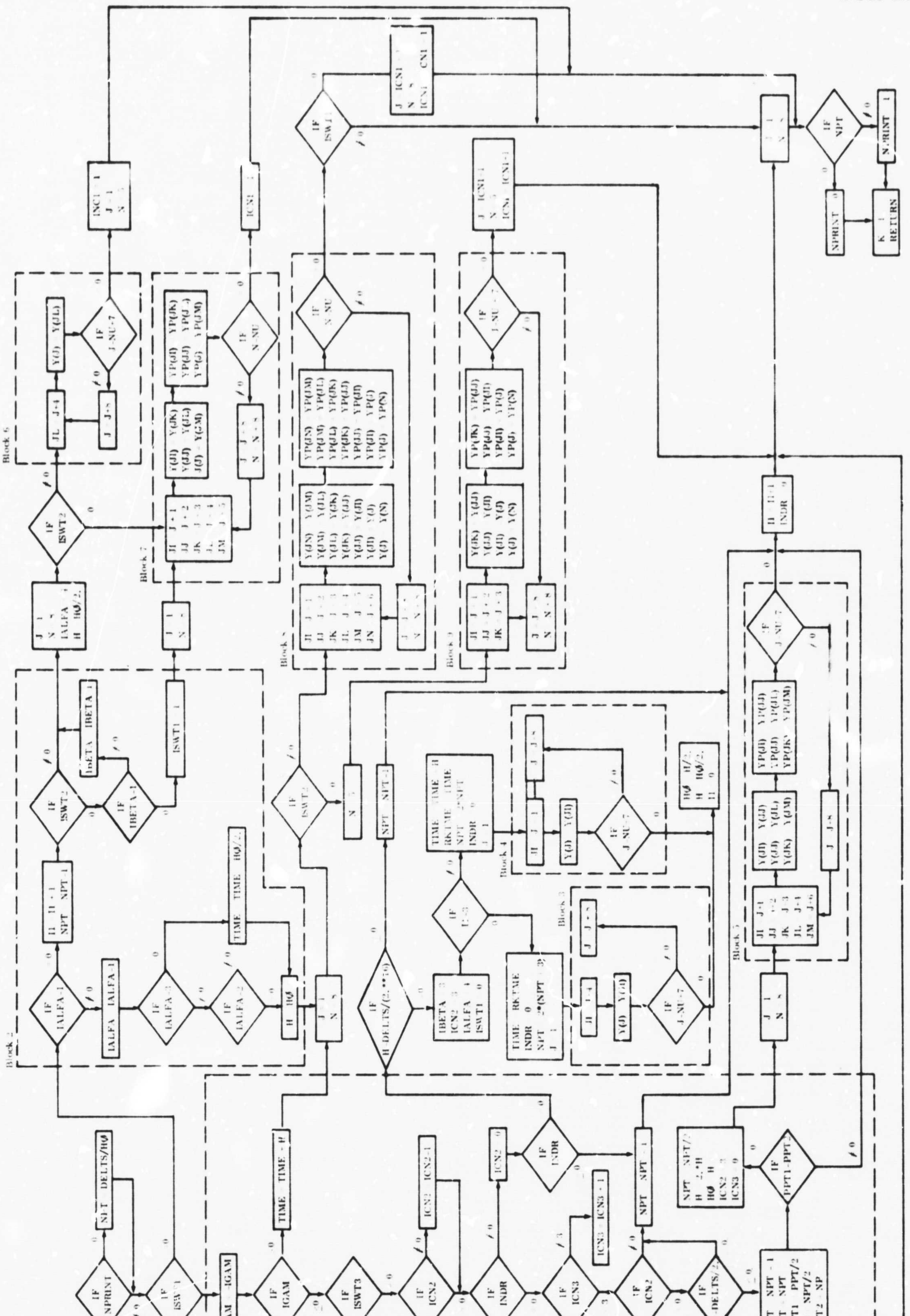
Block 6 - Update integration tables for fourth order Runge-Kutta scheme-final value

Block 7 - Set up integration tables in preparation for starting Adams 4-point integration scheme

Block 8 - Update integration tables for "normal" Adams 4-point scheme operation

Block 9 - Update integration tables for fourth order Runge-Kutta scheme-intermediate values

SUBRØUTINE UPDAT(TIME) ,
SUBRØUTINE UPDAT



SUBRØUTINE DETERM

Purpose: To calculate the determinant of a 6 x 6 matrix

Program(s): N-BØDY, LØF *

Flow Diagram Key:

Block 1 - Initialize A(I, J) matrix

*Note: In the LØF program the A(I, J) matrix is set equal to PHIMAT(I, J)

Block 2 - Find the largest element in the A(I, J) matrix

Block 3 - Interchange rows to put largest element in first row

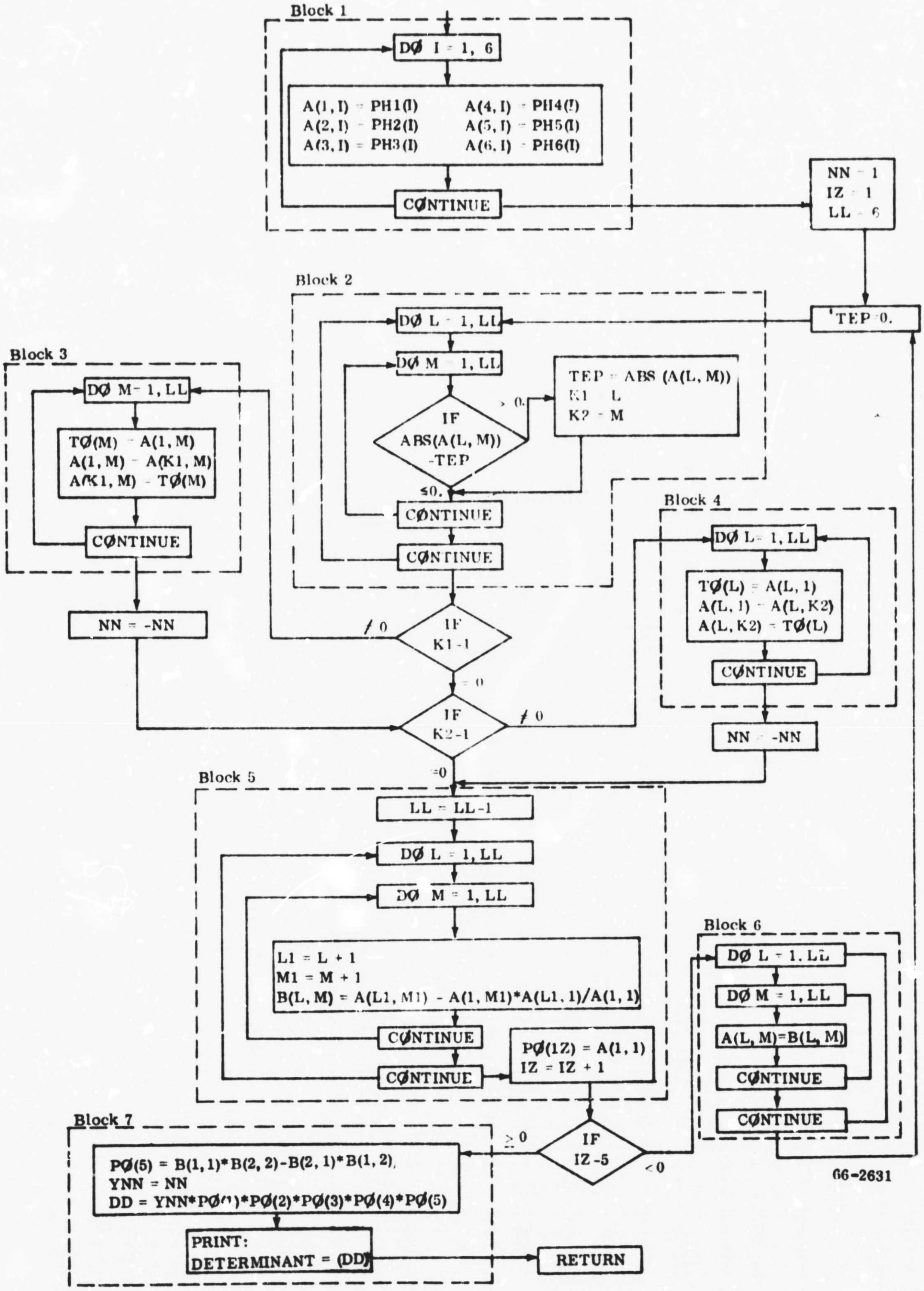
Block 4 - Interchange columns to put largest element in first column

Block 5 - Reduce matrix order by pivotal method

Block 6 - Set A(I, J) matrix equal to reduced order B(I, J) matrix

Block 7 - Calculate determinant and print results

SUBROUTINE DETERM



66-2631

SUBR0UTINE TRAMAT (KK)

Purpose:	To calculate the elements of the transition matrix
Program(s):	N-B0DY

Flow Diagram Key:

Block 1 - Calculate the central body contribution to the element time derivatives

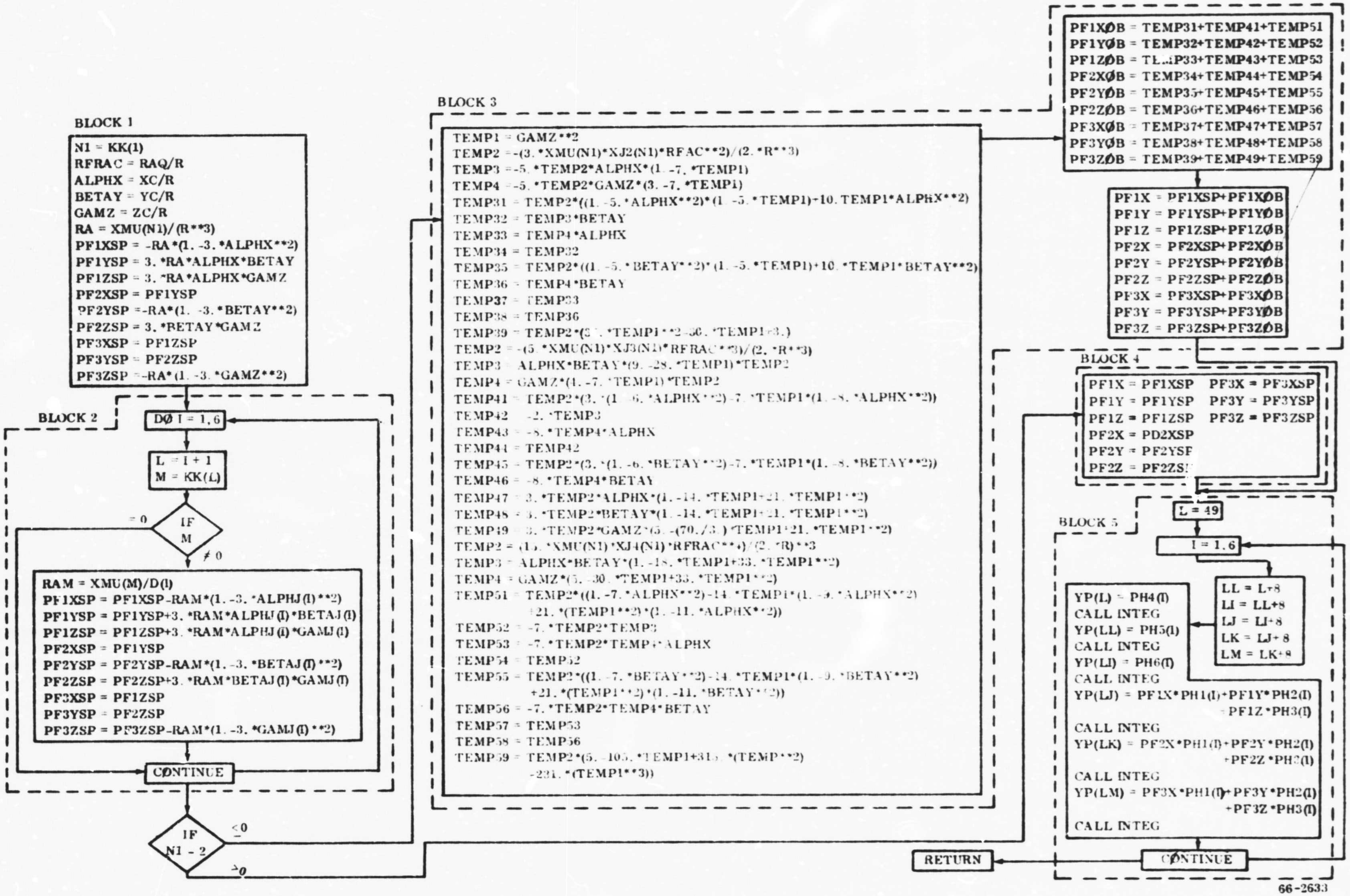
Block 2 - Calculate the contribution of the other Newtonian force fields to the element time derivatives

Block 3 - Calculate the central body oblateness contribution to the element time derivatives and combine with the Newtonian force contributions to form the total element time derivatives

Block 4 - Set the total element time derivatives equal to the Newtonian force field contributions

Block 5 - Integrate the total element time derivatives.

SUBROUTINE TRAMAT (KK)



BLOCK 1

```
N1 = KK(1)
RFRAC = RAQ/R
ALPHX = XC/R
BETAY = YC/R
GAMZ = ZC/R
RA = XMU(N1)/(R**3)
PF1XSP = -RA*(1. -3. *ALPHX**2)
PF1YSP = 3. *RA*ALPHX*BETAY
PF1ZSP = 3. *RA*ALPHX*GAMZ
PF2XSP = PF1YSP
PF2YSP =-RA*(1. -3. *BETAY**2)
PF2ZSP = 3. *BETAY*GAMZ
PF3XSP = PF1ZSP
PF3YSP = PF2ZSP
PF3ZSP =-RA*(1. -3. *GAMZ**2)
```

BLOCK 2

```
RAM = XMU(M)/D(I)
PF1XSP = PF1XSP-RAM*(1. -3. *ALPHJ(I)**2)
PF1YSP = PF1YSP+3. *RAM*ALPHJ(I)*BETAJ(I)
PF1ZSP = PF1ZSP+3. *RAM*ALPHJ(I)*GAMJ(I)
PF2XSP = PF1YSP
PF2YSP = PF2YSP-RAM*(1. -3. *BETAJ(I)**2)
PF2ZSP = PF2ZSP+3. *RAM*BETAJ(I)*GAMJ(I)
PF3XSP = PF1ZSP
PF3YSP = PF2ZSP
PF3ZSP = PF3ZSP-RAM*(1. -3. *GAMJ(I)**2)
```

BLOCK 3

```
TEMP1 = GAMZ**2
TEMP2 =-(3. *XMU(N1)*XJ2(N1)*RFAC**2)/(2. *R**3)
TEMP3 =-5. *TEMP2*ALPHX*(1. -7. *TEMP1)
TEMP4 =-5. *TEMP2*GAMZ*(3. -7. *TEMP1)
TEMP31 = TEMP2*((1. -5. *ALPHX**2)*(1. -5. *TEMP1)-10. TEMP1*ALPHX**2)
TEMP32 = TEMP3*BETAY
TEMP33 = TEMP4*ALPHX
TEMP34 = TEMP32
TEMP35 = TEMP2*((1. -5. *BETAY**2)*(1. -5. *TEMP1)+10. *TEMP1*BETAY**2)
TEMP36 = TEMP4*BETAY
TEMP37 = TEMP33
TEMP38 = TEMP36
TEMP39 = TEMP2*(35. *TEMP1**2-30. *TEMP1+3.)
TEMP2 = -(5. *XMU(N1)*XJ3(N1)*RFRAC**3)/(2. *R**3)
TEMP3 = ALPHX*BETAY*(9. -28. *TEMP1)*TEMP2
TEMP4 = GAMZ*(1. -7. *TEMP1)*TEMP2
TEMP41 = TEMP2*(3. *(1. -6. *ALPHX**2)-7. *TEMP1*(1. -8. *ALPHX**2))
TEMP42 = -2. *TEMP3
TEMP43 = -8. *TEMP4*ALPHX
TEMP44 = TEMP42
TEMP45 = TEMP2*(3. *(1. -6. *BETAY**2)-7. *TEMP1*(1. -8. *BETAY**2))
TEMP46 = -8. *TEMP4*BETAY
TEMP47 = 3. *TEMP2*ALPHX*(1. -14. *TEMP1+21. *TEMP1**2)
TEMP48 = 3. *TEMP2*BETAY*(1. -14. *TEMP1+21. *TEMP1**2)
TEMP49 = 3. *TEMP2*GAMZ*(5. -(70./3.) *TEMP1+21. *TEMP1**2)
TEMP2 = (15. *XMU(N1) *XJ4(N1) *RFRAC**4)/(2. *R)**3
TEMP3 = ALPHX*BETAY*(1. -18. *TEMP1+33. *TEMP1**2)
TEMP4 = GAMZ*(5. -30. *TEMP1+33. *TEMP1**2)
TEMP51 = TEMP2*((1. -7. *ALPHX**2)-14. *TEMP1*(1. -9. *ALPHX**2)
         +21. *(TEMP1**2)*(1. -11. *ALPHX**2))
TEMP52 = -7. *TEMP2*TEMP3
TEMP53 = -7. *TEMP2*TEMP4*ALPHX
TEMP54 = TEMP52
TEMP55 = TEMP2*((1. -7. *BETAY**2)-14. *TEMP1*(1. -9. *BETAY**2)
         +21. *(TEMP1**2)*(1. -11. *BETAY**2))
TEMP56 = -7. *TEMP2*TEMP4*BETAY
TEMP57 = TEMP53
TEMP58 = TEMP56
TEMP59 = TEMP2*(5. -105. *TEMP1+315. *(TEMP**2)
         -231. *(TEMP1**3))
```

```
PF1XOB = TEMP31+TEMP41+TEMP51
PF1YOB = TEMP32+TEMP42+TEMP52
PF1ZOB = TEMP33+TEMP43+TEMP53
PF2XOB = TEMP34+TEMP44+TEMP54
PF2YOB = TEMP35+TEMP45+TEMP55
PF2ZOB = TEMP36+TEMP46+TEMP56
PF3XOB = TEMP37+TEMP47+TEMP57
PF3YOB = TEMP38+TEMP48+TEMP58
PF3ZOB = TEMP39+TEMP49+TEMP59
```

```
PF1X = PF1XSP+PF1XOB
PF1Y = PF1YSP+PF1YOB
PF1Z = PF1ZSP+PF1ZOB
PF2X = PF2XSP+PF2XOB
PF2Y = PF2YSP+PF2YOB
PF2Z = PF2ZSP+PF2ZOB
PF3X = PF3XSP+PF3XOB
PF3Y = PF3YSP+PF3YOB
PF3Z = PF3ZSP+PF3ZOB
```

BLOCK 4

```
PF1X = PF1XSP   PF3X = PF3XSP
PF1Y = PF1YSP   PF3Y = PF3YSP
PF1Z = PF1ZSP   PF3Z = PF3ZSP
PF2X = PD2XSP
PF2Y = PF2YSP
PF2Z = PF2ZSP
```

BLOCK 5

```
LL = L+8
LI = LL+8
LJ = LI+8
LK = LJ+8
LM = LK+8
```

```
YP(L) = PH4(I)
CALL INTEG
YP(LL) = PH5(I)
CALL INTEG
YP(LI) = PH6(I)
CALL INTEG
YP(LJ) = PF1X*PH1(I)+PF1Y*PH2(I)
                    +PF1Z*PH3(I)
CALL INTEG
YP(LK) = PF2X*PH1(I)+PF2Y*PH2(I)
                    +PF2Z*PH3(I)
CALL INTEG
YP(LM) = PF3X*PH1(I)+PF3Y*PH2(I)
                    +PF3Z*PH3(I)
CALL INTEG
```

66-2633

SUBR0UTINE NUMER

Purpose:	To collect error statistics and calculate the statistical mean, standard deviation, and second moment about zero
Program(s):	L0F

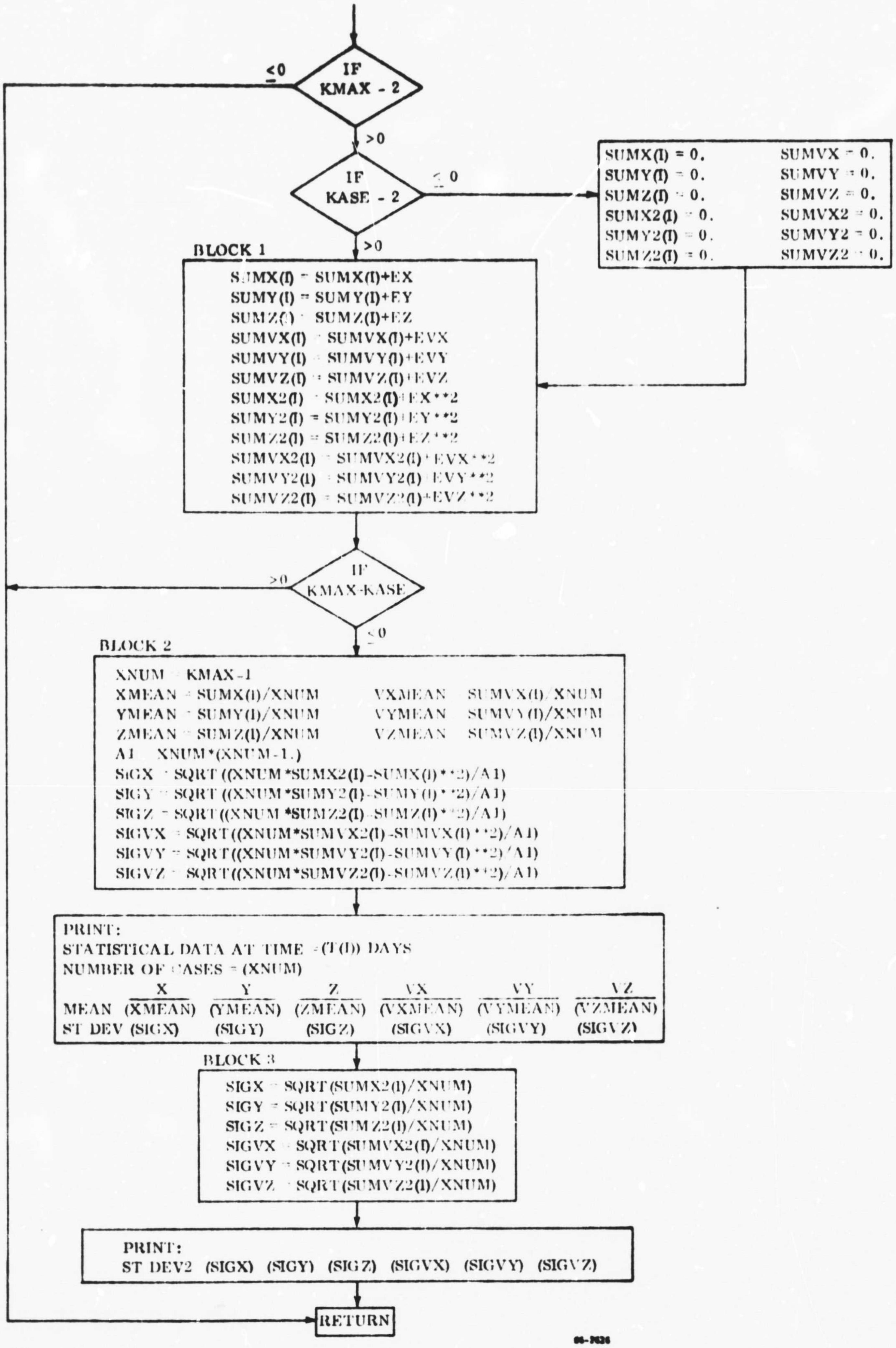
Flow Diagram Key:

Block 1 - Sum the errors and square of the errors in navigated position and velocity for each navigation measurement time

Block 2 - Calculate the mean and standard deviation for each error component

Block 3 - Calculate the second moment about zero for each error component

SUBROUTINE NUMER



```
SUMX(I) = 0.          SUMVX = 0.
SUMY(I) = 0.          SUMVY = 0.
SUMZ(I) = 0.          SUMVZ = 0.
SUMX2(I) = 0.         SUMVX2 = 0.
SUMY2(I) = 0.         SUMVY2 = 0.
SUMZ2(I) = 0.         SUMVZ2 = 0.
```

```
SUMX(I) = SUMX(I)+EX
SUMY(I) = SUMY(I)+EY
SUMZ(I) = SUMZ(I)+EZ
SUMVX(I) = SUMVX(I)+EVX
SUMVY(I) = SUMVY(I)+EVY
SUMVZ(I) = SUMVZ(I)+EVZ
SUMX2(I) = SUMX2(I)+EX**2
SUMY2(I) = SUMY2(I)+EY**2
SUMZ2(I) = SUMZ2(I)+EZ**2
SUMVX2(I) = SUMVX2(I)+EVX**2
SUMVY2(I) = SUMVY2(I)+EVY**2
SUMVZ2(I) = SUMVZ2(I)+EVZ**2
```

```
XNUM = KMAX-1
XMEAN = SUMX(I)/XNUM         VXMEAN = SUMVX(I)/XNUM
YMEAN = SUMY(I)/XNUM         VYMEAN = SUMVY(I)/XNUM
ZMEAN = SUMZ(I)/XNUM         VZMEAN = SUMVZ(I)/XNUM
A1 = XNUM*(XNUM-1.)
SIGX = SQRT ((XNUM*SUMX2(I)-SUMX(I)**2)/A1)
SIGY = SQRT ((XNUM*SUMY2(I)-SUMY(I)**2)/A1)
SIGZ = SQRT((XNUM *SUMZ2(I)-SUMZ(I)**2)/A1)
SIGVX = SQRT((XNUM*SUMVX2(I)-SUMVX(I)**2)/A1)
SIGVY = SQRT((XNUM*SUMVY2(I)-SUMVY(I)**2)/A1)
SIGVZ = SQRT((XNUM*SUMVZ2(I)-SUMVZ(I)**2)/A1)
```

```
PRINT:
STATISTICAL DATA AT TIME =(T(I)) DAYS
NUMBER OF CASES = (XNUM)
               X         Y         Z          VX          VY          VZ
MEAN     (XMEAN)   (YMEAN)   (ZMEAN)   (VXMEAN)   (VYMEAN)   (VZMEAN)
ST DEV    (SIGX)    (SIGY)    (SIGZ)    (SIGVX)    (SIGVY)    (SIGVZ)
```

```
SIGX = SQRT(SUMX2(I)/XNUM)
SIGY = SQRT(SUMY2(I)/XNUM)
SIGZ = SQRT(SUMZ2(I)/XNUM)
SIGVX = SQRT(SUMVX2(I)/XNUM)
SIGVY = SQRT(SUMVY2(I)/XNUM)
SIGVZ = SQRT(SUMVZ2(I)/XNUM)
```

```
PRINT:
ST DEV2  (SIGX)  (SIGY)  (SIGZ)  (SIGVX)  (SIGVY)  (SIGVZ)
```

66-7626

SUBRØUTINE NUMER1

Purpose:	To calculate mean and standard deviation values for the position and velocity errors; it is used when more than one navigated trajectory has been generated
Program(s):	ISN

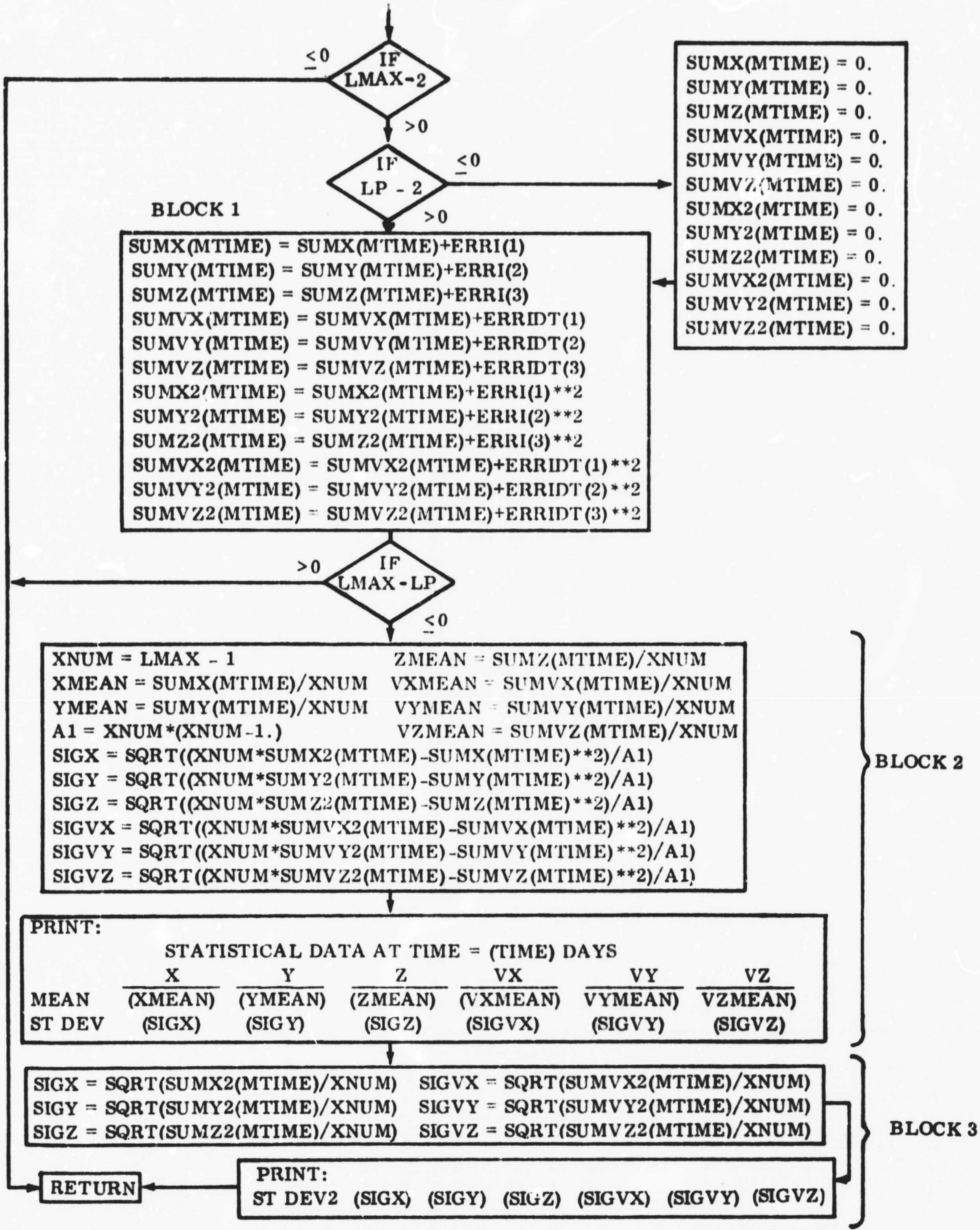
Flow Diagram Key:

Block 1 - Sum the errors and the squares of the errors at each measurement time.

Block 2 - Calculate and print the mean and standard derivation of the errors at the last measurement time.

Block 3 - Calculate and print the second moment about zero at the last measurement time.

SUBRØUTINE NUMER1



```
SUMX(MTIME) = 0.
SUMY(MTIME) = 0.
SUMZ(MTIME) = 0.
SUMVX(MTIME) = 0.
SUMVY(MTIME) = 0.
SUMVZ(MTIME) = 0.
SUMX2(MTIME) = 0.
SUMY2(MTIME) = 0.
SUMZ2(MTIME) = 0.
SUMVX2(MTIME) = 0.
SUMVY2(MTIME) = 0.
SUMVZ2(MTIME) = 0.
```

```
SUMX(MTIME) = SUMX(MTIME)+ERRI(1)
SUMY(MTIME) = SUMY(MTIME)+ERRI(2)
SUMZ(MTIME) = SUMZ(MTIME)+ERRI(3)
SUMVX(MTIME) = SUMVX(MTIME)+ERRIDT(1)
SUMVY(MTIME) = SUMVY(MTIME)+ERRIDT(2)
SUMVZ(MTIME) = SUMVZ(MTIME)+ERRIDT(3)
SUMX2(MTIME) = SUMX2(MTIME)+ERRI(1)**2
SUMY2(MTIME) = SUMY2(MTIME)+ERRI(2)**2
SUMZ2(MTIME) = SUMZ2(MTIME)+ERRI(3)**2
SUMVX2(MTIME) = SUMVX2(MTIME)+ERRIDT(1)**2
SUMVY2(MTIME) = SUMVY2(MTIME)+ERRIDT(2)**2
SUMVZ2(MTIME) = SUMVZ2(MTIME)+ERRIDT(3)**2
```

```
XNUM = LMAX - 1                   ZMEAN = SUMZ(MTIME)/XNUM
XMEAN = SUMX(MTIME)/XNUM          VXMEAN = SUMVX(MTIME)/XNUM
YMEAN = SUMY(MTIME)/XNUM          VYMEAN = SUMVY(MTIME)/XNUM
A1 = XNUM*(XNUM-1.)               VZMEAN = SUMVZ(MTIME)/XNUM
SIGX = SQRT((XNUM*SUMX2(MTIME)-SUMX(MTIME)**2)/A1)
SIGY = SQRT((XNUM*SUMY2(MTIME)-SUMY(MTIME)**2)/A1)
SIGZ = SQRT((XNUM*SUMZ2(MTIME)-SUMZ(MTIME)**2)/A1)
SIGVX = SQRT((XNUM*SUMVX2(MTIME)-SUMVX(MTIME)**2)/A1)
SIGVY = SQRT((XNUM*SUMVY2(MTIME)-SUMVY(MTIME)**2)/A1)
SIGVZ = SQRT((XNUM*SUMVZ2(MTIME)-SUMVZ(MTIME)**2)/A1)
```

```
PRINT:
              STATISTICAL DATA AT TIME = (TIME) DAYS
            X          Y          Z          VX          VY          VZ
MEAN     (XMEAN)    (YMEAN)    (ZMEAN)    (VXMEAN)    VYMEAN)     VZMEAN)
ST DEV   (SIGX)     (SIGY)     (SIGZ)     (SIGVX)     (SIGVY)     (SIGVZ)
```

```
SIGX = SQRT(SUMX2(MTIME)/XNUM)    SIGVX = SQRT(SUMVX2(MTIME)/XNUM)
SIGY = SQRT(SUMY2(MTIME)/XNUM)    SIGVY = SQRT(SUMVY2(MTIME)/XNUM)
SIGZ = SQRT(SUMZ2(MTIME)/XNUM)    SIGVZ = SQRT(SUMVZ2(MTIME)/XNUM)
```

```
PRINT:
ST DEV2  (SIGX)  (SIGY)  (SIGZ)  (SIGVX)  (SIGVY)  (SIGVZ)
```

66-2636

SUBRØUTINE PHINT1

Purpose:	To calculate the derivatives of the transition matrix elements
Program(s):	LØF

Flow Diagram Key:

Block 1 - Calculate the central body contribution (Newtonian) to the transition matrix derivatives.

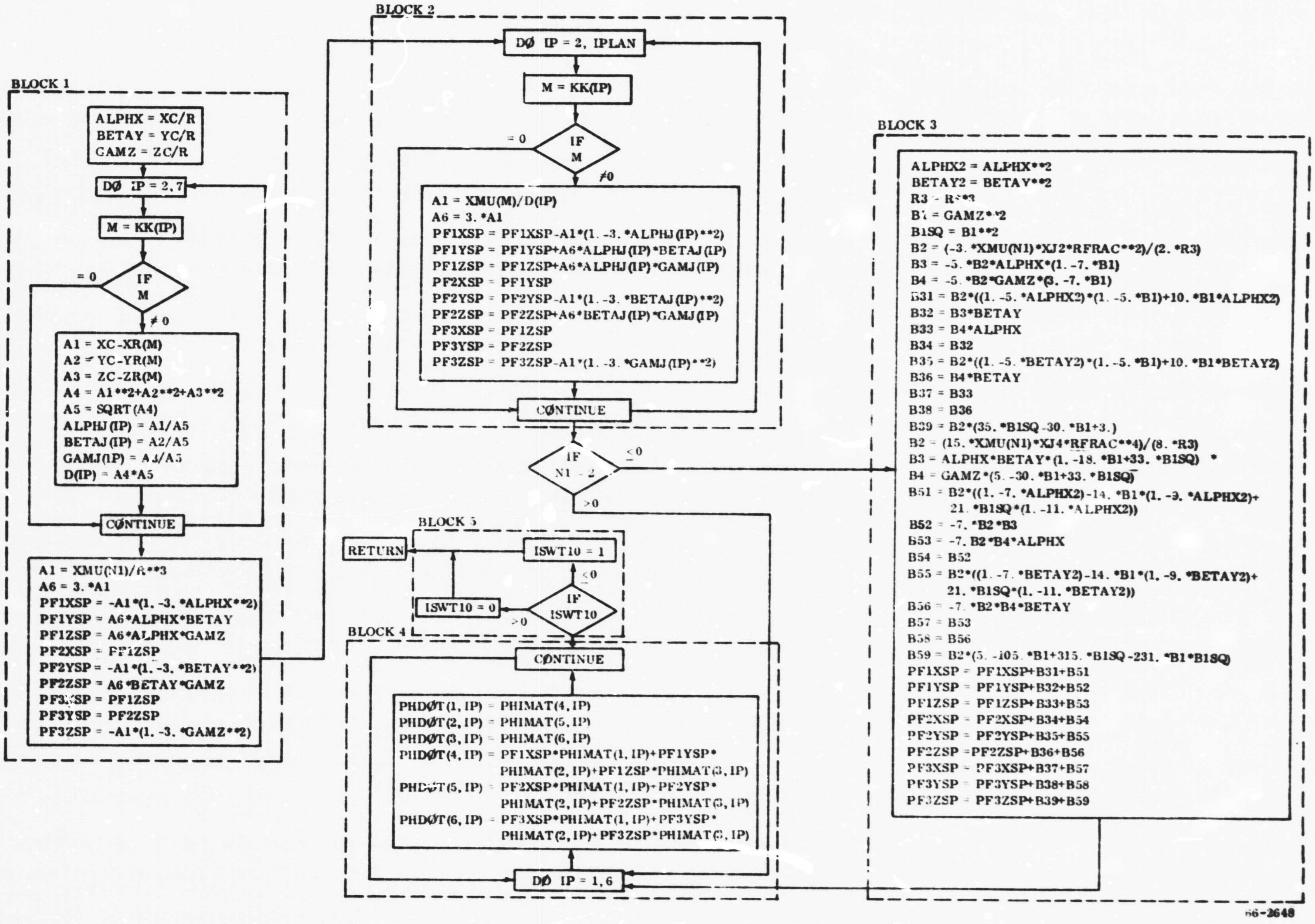
Block 2 - Calculate the contribution of the n^{th} celestial body to the transition matrix derivatives.

Block 3 - Calculate the central body contribution (oblate body) to the transition matrix derivatives.

Block 4 - Calculate the transition matrix derivatives.

Block 5 - Adjust flow control switches to be used in PHINT2 to assure correct integration of the transition matrix derivatives.

SUBRØUTINE PHINT1



```
ALPHX = XC/R
BETAY = YC/R
GAMZ = ZC/R
```

```
A1 = XC-XR(M)
A2 = YC-YR(M)
A3 = ZC-ZR(M)
A4 = A1**2+A2**2+A3**2
A5 = SQRT(A4)
ALPHJ(IP) = A1/A5
BETAJ(IP) = A2/A5
GAMJ(IP) = A3/A5
D(IP) = A4*A5
```

```
A1 = XMU(N1)/R**3
A6 = 3. *A1
PF1XSP = -A1*(1. -3. *ALPHX**2)
PF1YSP = A6*ALPHX*BETAY
PF1ZSP = A6*ALPHX*GAMZ
PF2XSP = PF1ZSP
PF2YSP= -A1*(1. -3. *BETAY**2)
PF2ZSP = A6*BETAY*GAMZ
PF3XSP = PF1ZSP
PF3YSP = PF2ZSP
PF3ZSP = -A1*(1. -3. *GAMZ**2)
```

```
A1 = XMU(M)/D(IP)
A6 = 3. *A1
PF1XSP = PF1XSP-A1*(1. -3. *ALPHJ(IP)**2)
PF1YSP = PF1YSP+A6*ALPHJ(IP)*BETAJ(IP)
PF1ZSP = PF1ZSP+A6*ALPHJ(IP)*GAMJ(IP)
PF2XSP = PF1YSP
PF2YSP = PF2YSP-A1*(1. -3. *BETAJ(IP)**2)
PF2ZSP = PF2ZSP+A6*BETAJ(IP)*GAMJ(IP)
PF3XSP = PF1ZSP
PF3YSP = PF2ZSP
PF3ZSP = PF3ZSP-A1*(1. -3. *GAMJ(IP)**2)
```

```
PHDØT(1, IP) = PHIMAT(4, IP)
PHDØT(2, IP) = PHIMAT(5, IP)
PHDØT(3, IP) = PHIMAT(6, IP)
PHDØT(4, IP) = PF1XSP*PHIMAT(1, IP)+PF1YSP*
               PHIMAT(2, IP)+PF1ZSP*PHIMAT(3, IP)
PHDØT(5, IP) = PF2XSP*PHIMAT(1, IP)+PF2YSP*
               PHIMAT(2, IP)+PF2ZSP*PHIMAT(3, IP)
PHDØT(6, IP) = PF3XSP*PHIMAT(1, IP)+PF3YSP*
               PHIMAT(2, IP)+PF3ZSP*PHIMAT(3, IP)
```

```
ALPHX2 = ALPHX**2
BETAY2 = BETAY**2
R3 = R**3
B1 = GAMZ**2
B1SQ = B1**2
B2 = (-3. *XMU(N1)*XJ2*RFRAC**2)/(2. *R3)
B3 = -5. *B2*ALPHX*(1. -7. *B1)
B4 = -5. *B2*GAMZ*(3. -7. *B1)
B31 = B2*((1. -5. *ALPHX2)*(1. -5. *B1)+10. *B1*ALPHX2)
B32 = B3*BETAY
B33 = B4*ALPHX
B34 = B32
B35 = B2*((1. -5. *BETAY2)*(1. -5. *B1)+10. *B1*BETAY2)
B36 = B4*BETAY
B37 = B33
B38 = B36
B39 = B2*(35. *B1SQ-30. *B1+3.)
B2 = (15. *XMU(N1)*XJ4*RFRAC**4)/(8. *R3)
B3 = ALPHX*BETAY*(1. -18. *B1+33. *B1SQ)  *
B4 = GAMZ*(5. -30. *B1+33. *B1SQ)
B51 = B2*((1. -7. *ALPHX2)-14. *B1*(1. -9. *ALPHX2)+
      21. *B1SQ*(1. -11. *ALPHX2))
B52 = -7. *B2*B3
B53 = -7. B2*B4*ALPHX
B54 = B52
B55 = B2*((1. -7. *BETAY2)-14. *B1*(1. -9. *BETAY2)+
      21. *B1SQ*(1. -11. *BETAY2))
B56 = -7. *B2*B4*BETAY
B57 = B53
B58 = B56
B59 = B2*(5. -105. *B1+315. *B1SQ-231. *B1*B1SQ)
PF1XSP = PF1XSP+B31+B51
PF1YSP = PF1YSP+B32+B52
PF1ZSP = PF1ZSP+B33+B53
PF2XSP = PF2XSP+B34+B54
PF2YSP = PF2YSP+B35+B55
PF2ZSP =PF2ZSP+B36+B56
PF3XSP = PF3XSP+B37+B57
PF3YSP = PF3YSP+B38+B58
PF3ZSP = PF3ZSP+B39+B59
```

SUBRØUTINE PHINT2

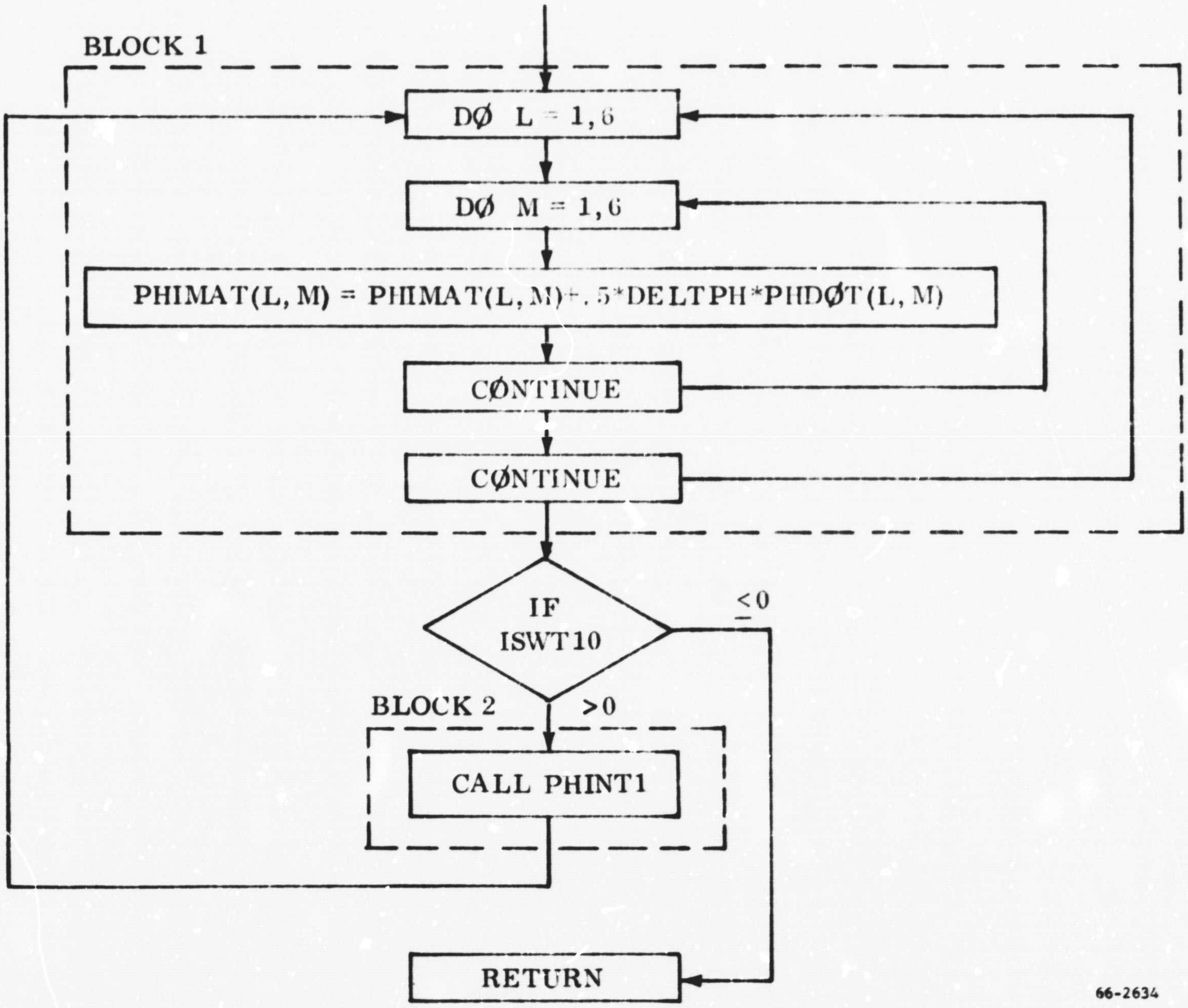
Purpose: To integrate the time derivatives of the transition matrix elements (second order Runge-Kutta scheme)

Program(s): LØF

Flow Diagram Key:

Block 1 - Calculate transition matrix elements

Block 2 - Obtain time derivatives at time = $t_n + \Delta t$

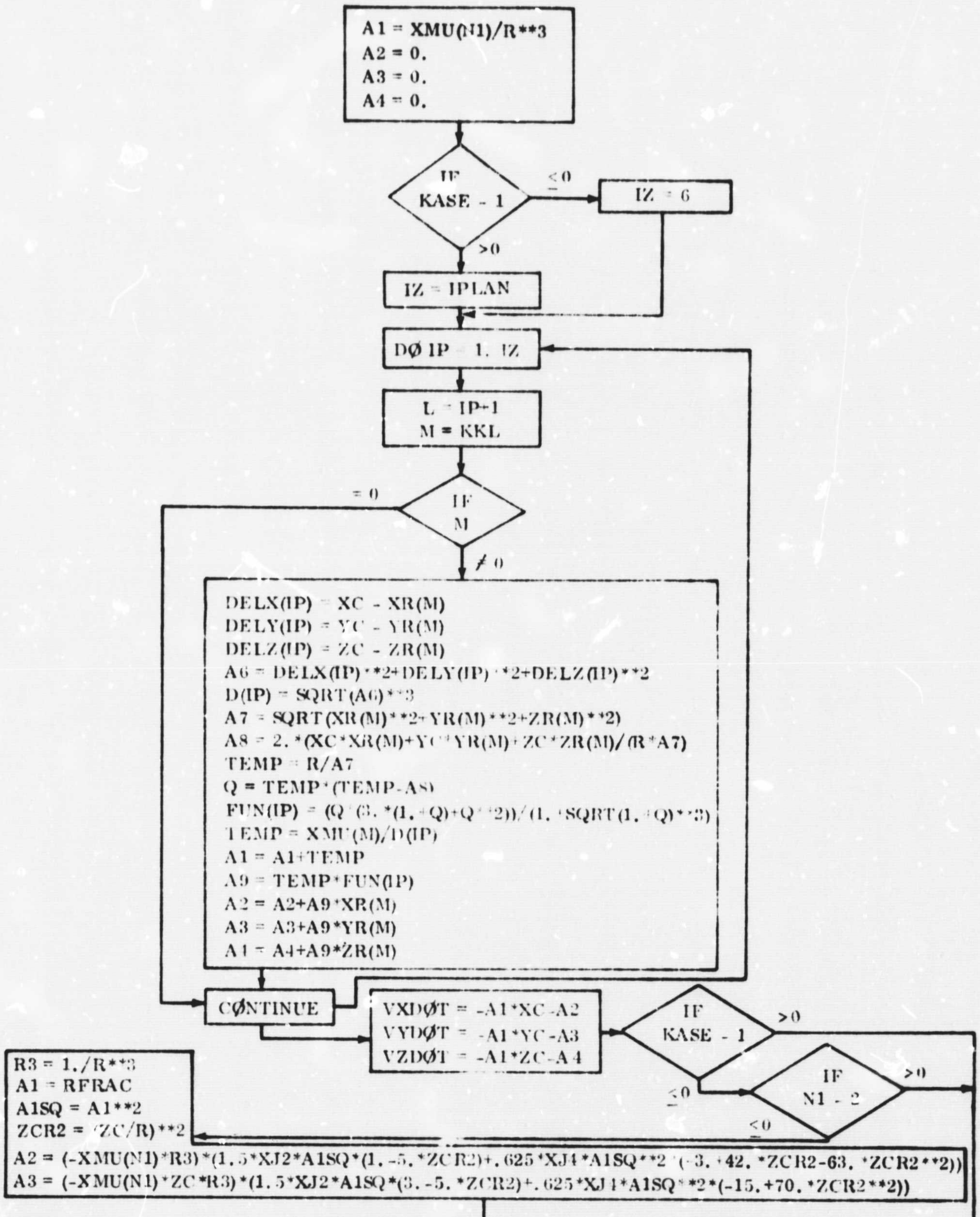


66-2634

SUBRØUTINE ACCEL

Purpose: To calculate the components of vehicle acceleration with respect to the center of coordinates

Program(s): LØF



SUBRØUTINE MATRAX (A, I1, I2, B, J1, J2, C, K1, K2, M1)

Purpose: To multiply two matrices $A_{I1,\ I2}$ and $B_{J1,\ J2}$ in the following ways:

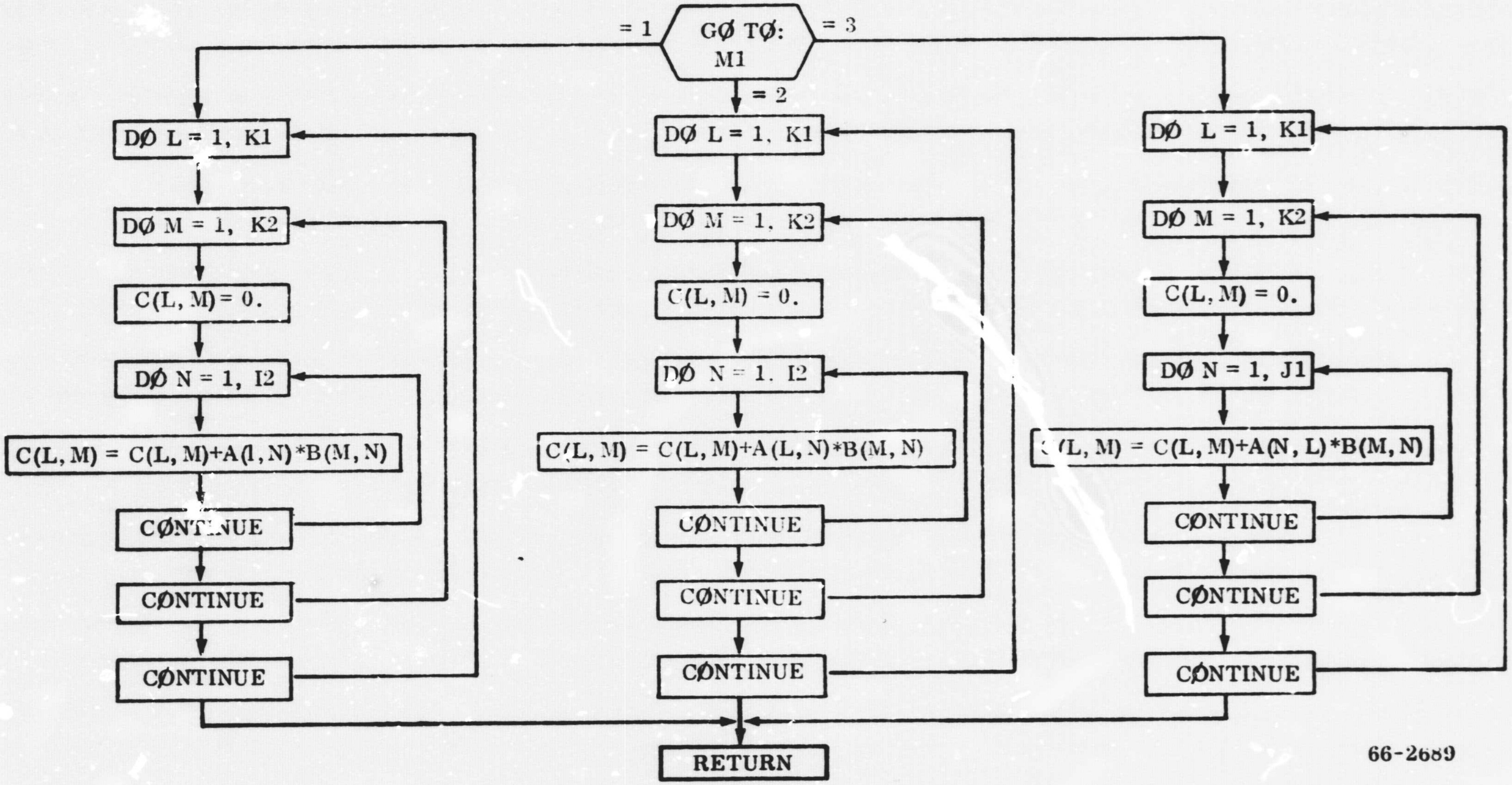
$C_{K1,\ K2} = A_{I1,\ I2} B_{J1,\ J2}$ M1 = 1

$C_{K1,\ K2} = A_{I1,\ I2} B^{T}_{J1,\ J2}$ M1 = 2

$C_{K1,\ K2} = A^{T}_{I1,\ I2} B_{J1,\ J2}$ M1 = 3

where the superscript T represents the transpose of the indicated matrax

Programs: LØF



66-2689

SUBRØUTINE RECTIF

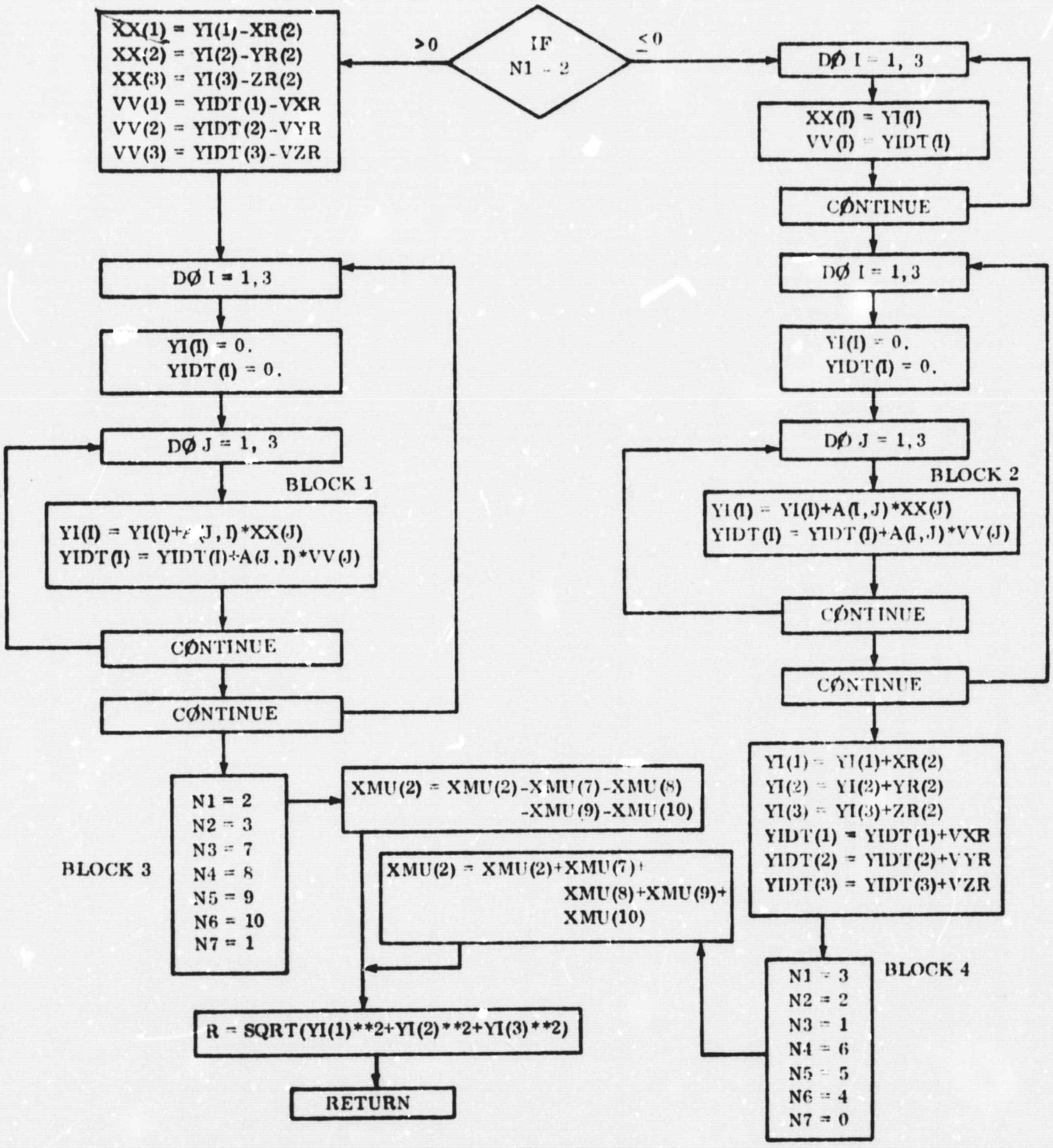
Purpose: To change coordinate centers when the vehicle has passed from one sphere of influence to another.

Program(s): ISN

Flow Diagram Key:

Block 1 & 2 - Calculate position and velocity with respect to new reference body.

Block 3 & 4 - Change planetary index codes corresponding to the particular celestial bodies to be considered in force model.



66-2670

SUBRØUTINE TIS

Purpose: To calculate the transformation from the inertial to spherical frame; to calculate the spherical acceleration components used in integrating the navigation equations.

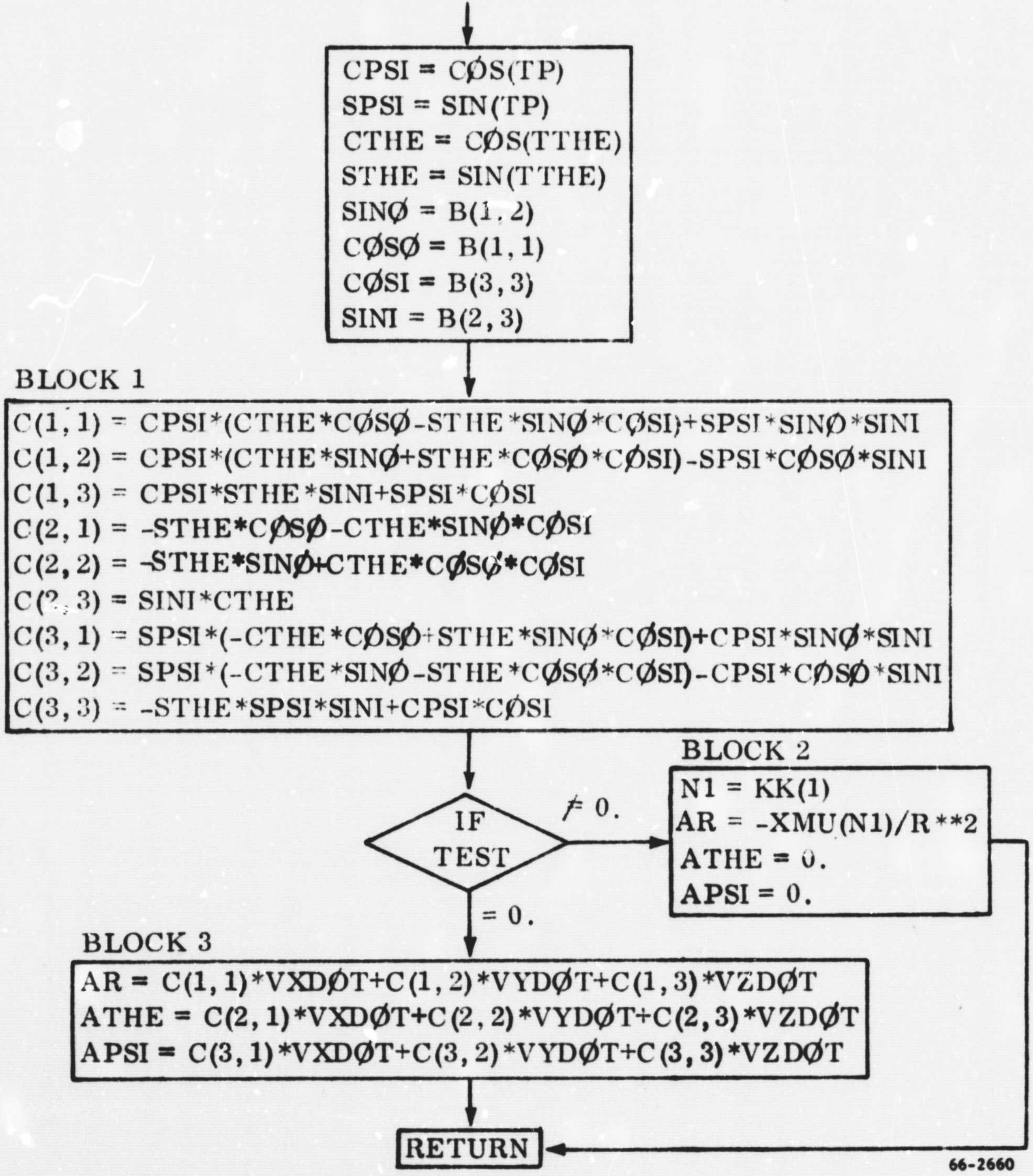
Program(s): ISN

Flow Diagram Key:

Block 1 - Calculate the transformation matrix from inertial to spherical frame.

Block 2 - Calculate spherical acceleration for two-body case.

Block 3 - Calculate spherical acceleration for n-body case.



```
CPSI = CØS(TP)
SPSI = SIN(TP)
CTHE = CØS(TTHE)
STHE = SIN(TTHE)
SINØ = B(1,2)
CØSØ = B(1,1)
CØSI = B(3,3)
SINI = B(2,3)
```

BLOCK 1

```
C(1,1) = CPSI*(CTHE*CØSØ-STHE*SINØ*CØSI)+SPSI*SINØ*SINI
C(1,2) = CPSI*(CTHE*SINØ+STHE*CØSØ*CØSI)-SPSI*CØSØ*SINI
C(1,3) = CPSI*STHE*SINI+SPSI*CØSI
C(2,1) = -STHE*CØSØ-CTHE*SINØ*CØSI
C(2,2) = -STHE*SINØ+CTHE*CØSØ*CØSI
C(2,3) = SINI*CTHE
C(3,1) = SPSI*(-CTHE*CØSØ+STHE*SINØ*CØSI)+CPSI*SINØ*SINI
C(3,2) = SPSI*(-CTHE*SINØ-STHE*CØSØ*CØSI)-CPSI*CØSØ*SINI
C(3,3) = -STHE*SPSI*SINI+CPSI*CØSI
```

BLOCK 2

```
N1 = KK(1)
AR = -XMU(N1)/R**2
ATHE = 0.
APSI = 0.
```

BLOCK 3

```
AR = C(1,1)*VXDØT+C(1,2)*VYDØT+C(1,3)*VZDØT
ATHE = C(2,1)*VXDØT+C(2,2)*VYDØT+C(2,3)*VZDØT
APSI = C(3,1)*VXDØT+C(3,2)*VYDØT+C(3,3)*VZDØT
```

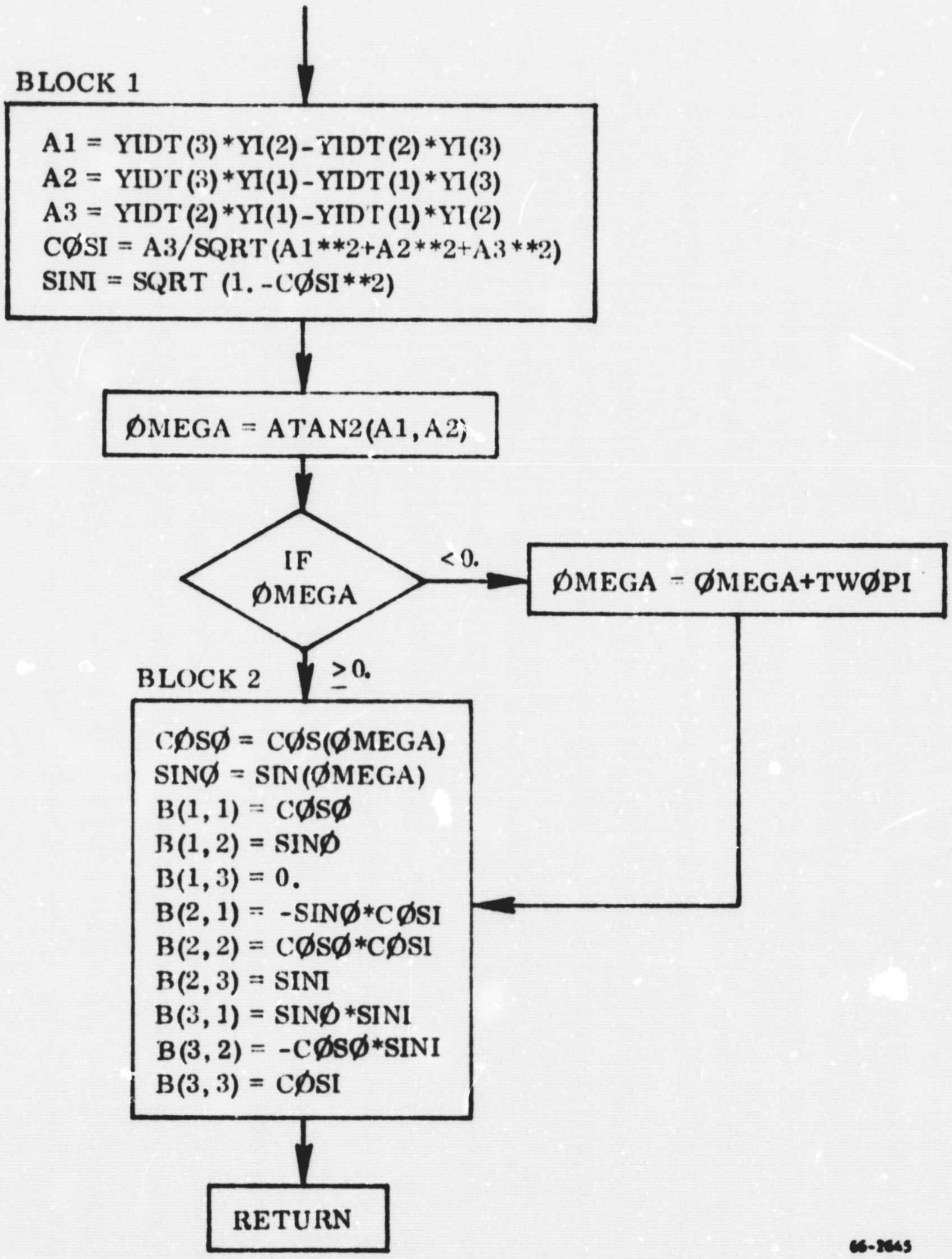
SUBRØUTINE TIØ

Purpose:	To calculate the transformation from the inertial to orbital frame. This is done in the beginning of the program and each time after rectification takes place.
Program(s):	ISN

Flow Diagram Key:

Block 1 - Calculate the sine and cosine of the orbital inclination angle.

Block 2 - Calculate the transformation matrix from the inertial to orbital frame.



```
A1 = YIDT(3)*YI(2)-YIDT(2)*YI(3)
A2 = YIDT(3)*YI(1)-YIDT(1)*YI(3)
A3 = YIDT(2)*YI(1)-YIDT(1)*YI(2)
CØSI = A3/SQRT(A1**2+A2**2+A3**2)
SINI = SQRT (1.-CØSI**2)
```

```
ØMEGA = ATAN2(A1,A2)
```

```
ØMEGA = ØMEGA+TWØPI
```

```
CØSØ = CØS(ØMEGA)
SINØ = SIN(ØMEGA)
B(1,1) = CØSØ
B(1,2) = SINØ
B(1,3) = 0.
B(2,1) = -SINØ*CØSI
B(2,2) = CØSØ*CØSI
B(2,3) = SINI
B(3,1) = SINØ*SINI
B(3,2) = -CØSØ*SINI
B(3,3) = CØSI
```

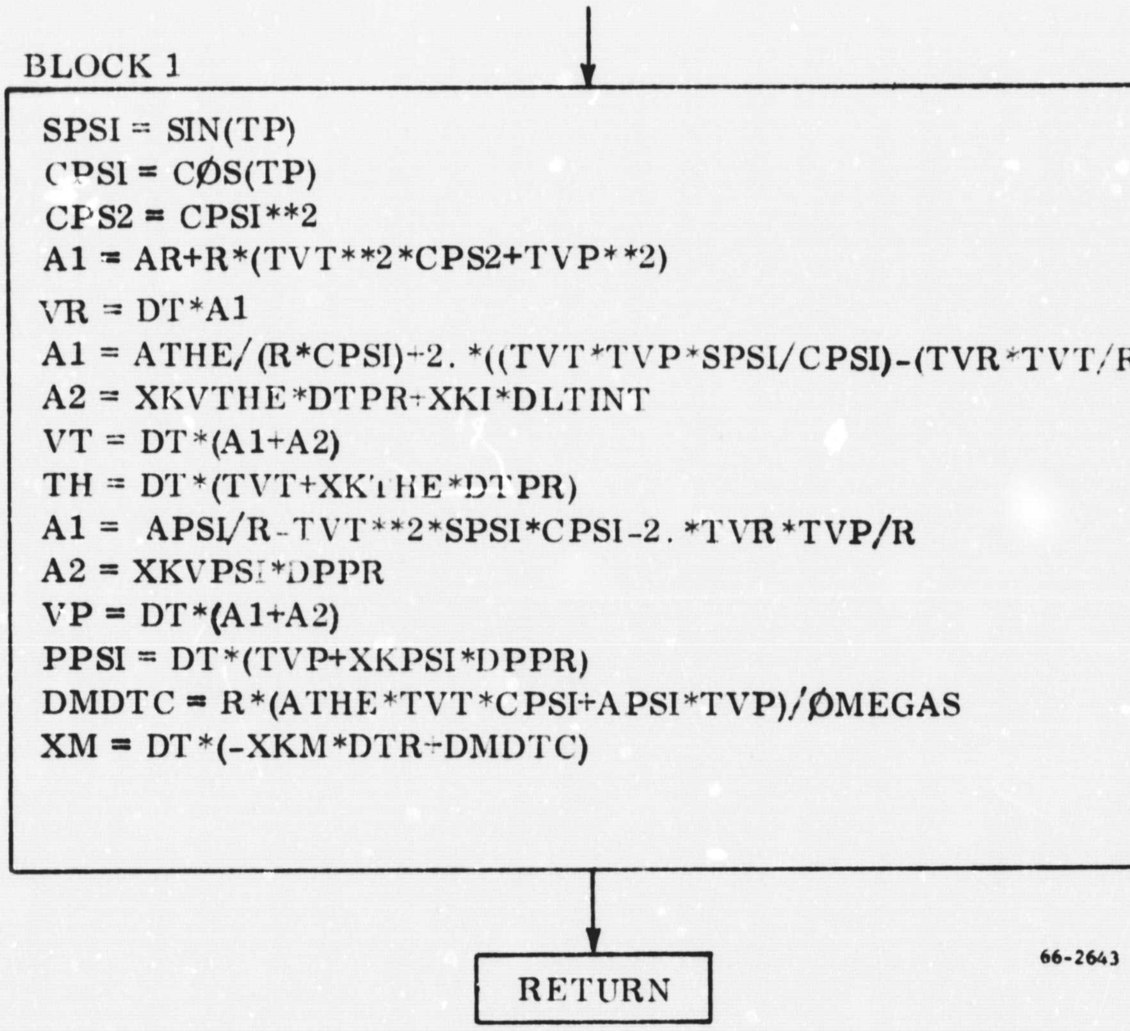
SUBRØUTINE NAVINT(VR, VT, TH, VP, PPSI, SM, DLTINT)

Purpose: To perform the intermediate stages of calculation in integrating the navigation equations.

Program(s): ISN

Flow Diagram Key:

Block 1 - Calculate the values of the navigation equations at a particular time step.



```
SPSI = SIN(TP)
CPSI = CØS(TP)
CPS2 = CPSI**2
A1 = AR+R*(TVT**2*CPS2+TVP**2)
VR = DT*A1
A1 = ATHE/(R*CPSI)+2.*((TVT*TVP*SPSI/CPSI)-(TVR*TVT/R))
A2 = XKVTHE*DTPR+XKI*DLTINT
VT = DT*(A1+A2)
TH = DT*(TVT+XKTHE*DTPR)
A1 =  APSI/R-TVT**2*SPSI*CPSI-2.*TVR*TVP/R
A2 = XKVPSI*DPPR
VP = DT*(A1+A2)
PPSI = DT*(TVP+XKPSI*DPPR)
DMDTC = R*(ATHE*TVT*CPSI+APSI*TVP)/ØMEGAS
XM = DT*(-XKM*DTR+DMDTC)
```

SUBRØUTINE TRUINT (VXDT, VYDT, VZDT, Y)

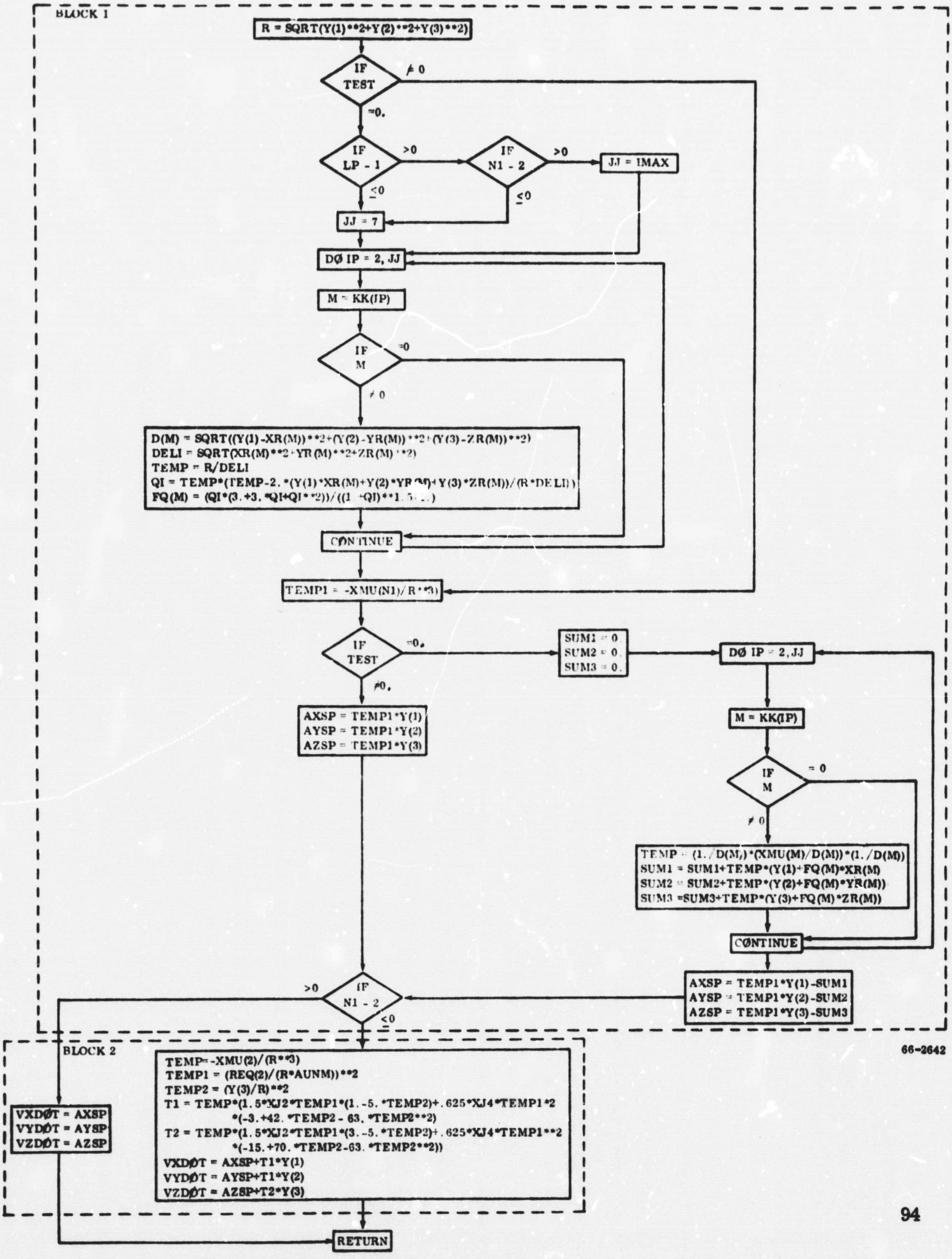
Purpose:	To calculate the gravitational force acceleration components and the oblate planet accelerations
Program(s):	ISN

Flow Diagram Key:

Block 1 - Calculate the gravitational force acceleration components.

Block 2 - Calculate the vehicle acceleration components.

SUBRØUTINE TRUINT (VXDT, VYDT, VZDT, Y)



66-2642

SUBRØUTINE INTSPH

Purpose: To calculate position and velocity in spherical coordinates given the equivalent Cartesian components

Program(s): ISN

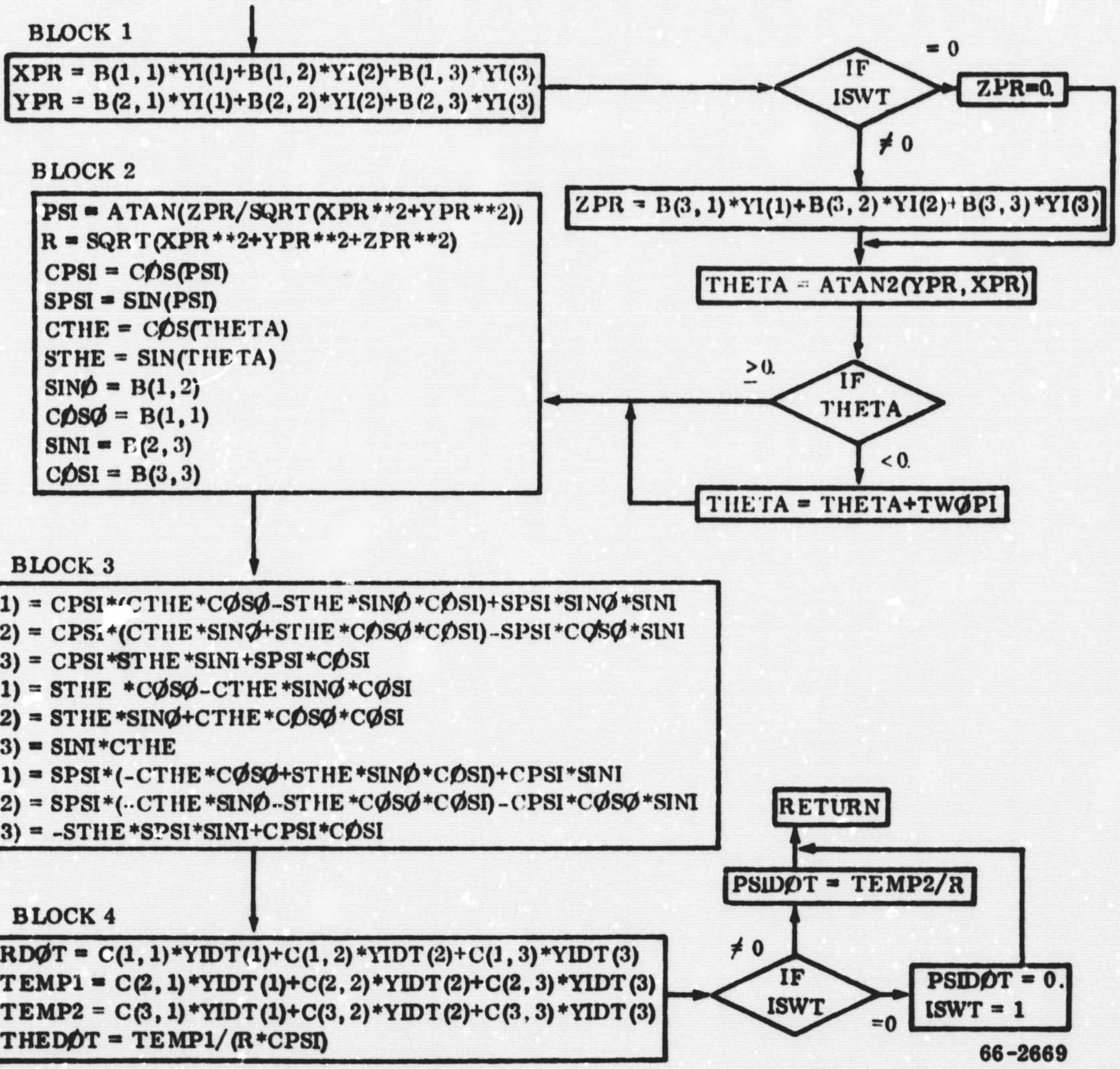
Flow Diagram Key:

Block 1 - Transform initial conditions to orbital frame.

Block 2 - Convert Cartesian position coordinates to spherical coordinates

Block 3 - Calculate the transformation matrix from inertial to spherical frame.

Block 4 - Calculate spherical coordinate derivatives.



```
XPR = B(1, 1)*YI(1)+B(1, 2)*YI(2)+B(1, 3)*YI(3)
YPR = B(2, 1)*YI(1)+B(2, 2)*YI(2)+B(2, 3)*YI(3)
```

```
ZPR=0.
```

```
ZPR = B(3, 1)*YI(1)+B(3, 2)*YI(2)+B(3, 3)*YI(3)
```

```
THETA = ATAN2(YPR, XPR)
```

```
THETA = THETA+TWØPI
```

```
PSI = ATAN(ZPR/SQRT(XPR**2+YPR**2))
R = SQRT(XPR**2+YPR**2+ZPR**2)
CPSI = CØS(PSI)
SPSI = SIN(PSI)
CTHE = CØS(THETA)
STHE = SIN(THETA)
SINØ = B(1,2)
CØSØ = B(1,1)
SINI = B(2,3)
CØSI = B(3,3)
```

```
C(1, 1) = CPSI*(CTHE*CØSØ-STHE*SINØ*CØSI)+SPSI*SINØ*SINI
C(1, 2) = CPSI*(CTHE*SINØ+STHE*CØSØ*CØSI)-SPSI*CØSØ*SINI
C(1, 3) = CPSI*STHE*SINI+SPSI*CØSI
C(2, 1) = STHE *CØSØ-CTHE*SINØ*CØSI
C(2, 2) = STHE*SINØ+CTHE*CØSØ*CØSI
C(2, 3) = SINI*CTHE
C(3, 1) = SPSI*(-CTHE*CØSØ+STHE*SINØ*CØSI)+CPSI*SINI
C(3, 2) = SPSI*(-CTHE*SINØ-STHE*CØSØ*CØSI)-CPSI*CØSØ*SINI
C(3, 3) = -STHE*SPSI*SINI+CPSI*CØSI
```

```
RDØT = C(1, 1)*YIDT(1)+C(1, 2)*YIDT(2)+C(1, 3)*YIDT(3)
TEMP1 = C(2, 1)*YIDT(1)+C(2, 2)*YIDT(2)+C(2, 3)*YIDT(3)
TEMP2 = C(3, 1)*YIDT(1)+C(3, 2)*YIDT(2)+C(3, 3)*YIDT(3)
THEDØT = TEMP1/(R*CPSI)
```

```
PSIDØT = TEMP2/R
```

```
PSIDØT = 0.
ISWT = 1
```

SUBRØUTINE SPHTIN

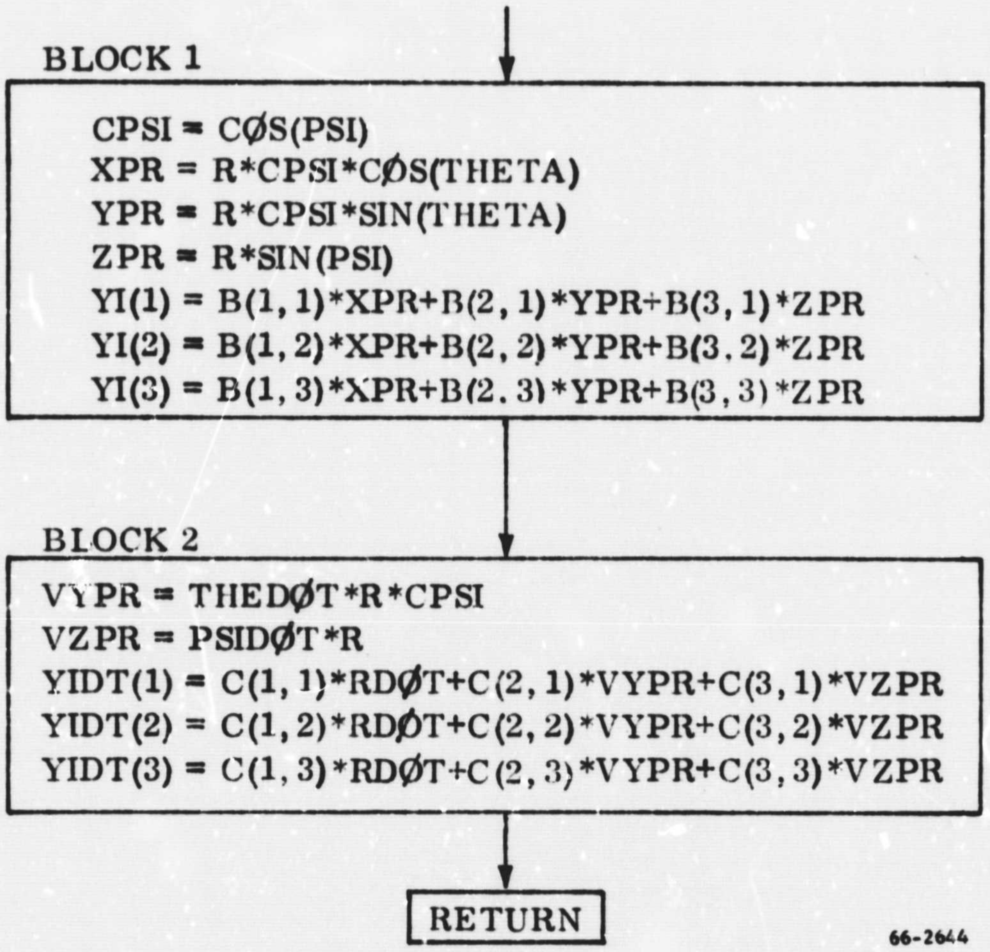
Purpose: To calculate position and velocity in Cartesian coordinates given the equivalent spherical coordinate components

Program(s): ISN

Flow Diagram Key:

Block 1 - Calculate inertial position coordinates

Block 2 - Calculate inertial velocity components



BLOCK 1

```
CPSI = CØS(PSI)
XPR = R*CPSI*CØS(THETA)
YPR = R*CPSI*SIN(THETA)
ZPR = R*SIN(PSI)
YI(1) = B(1, 1)*XPR+B(2, 1)*YPR+B(3, 1)*ZPR
YI(2) = B(1, 2)*XPR+B(2, 2)*YPR+B(3. 2)*ZPR
YI(3) = B(1, 3)*XPR+B(2. 3)*YPR+B(3, 3)*ZPR
```

BLOCK 2

```
VYPR = THEDØT*R*CPSI
VZPR = PSIDØT*R
YIDT(1) = C(1, 1)*RDØT+C(2, 1)*VYPR+C(3, 1)*VZPR
YIDT(2) = C(1, 2)*RDØT+C(2, 2)*VYPR+C(3, 2)*VZPR
YIDT(3) = C(1, 3)*RDØT+C(2, 3)*VYPR+C(3, 3)*VZPR
```

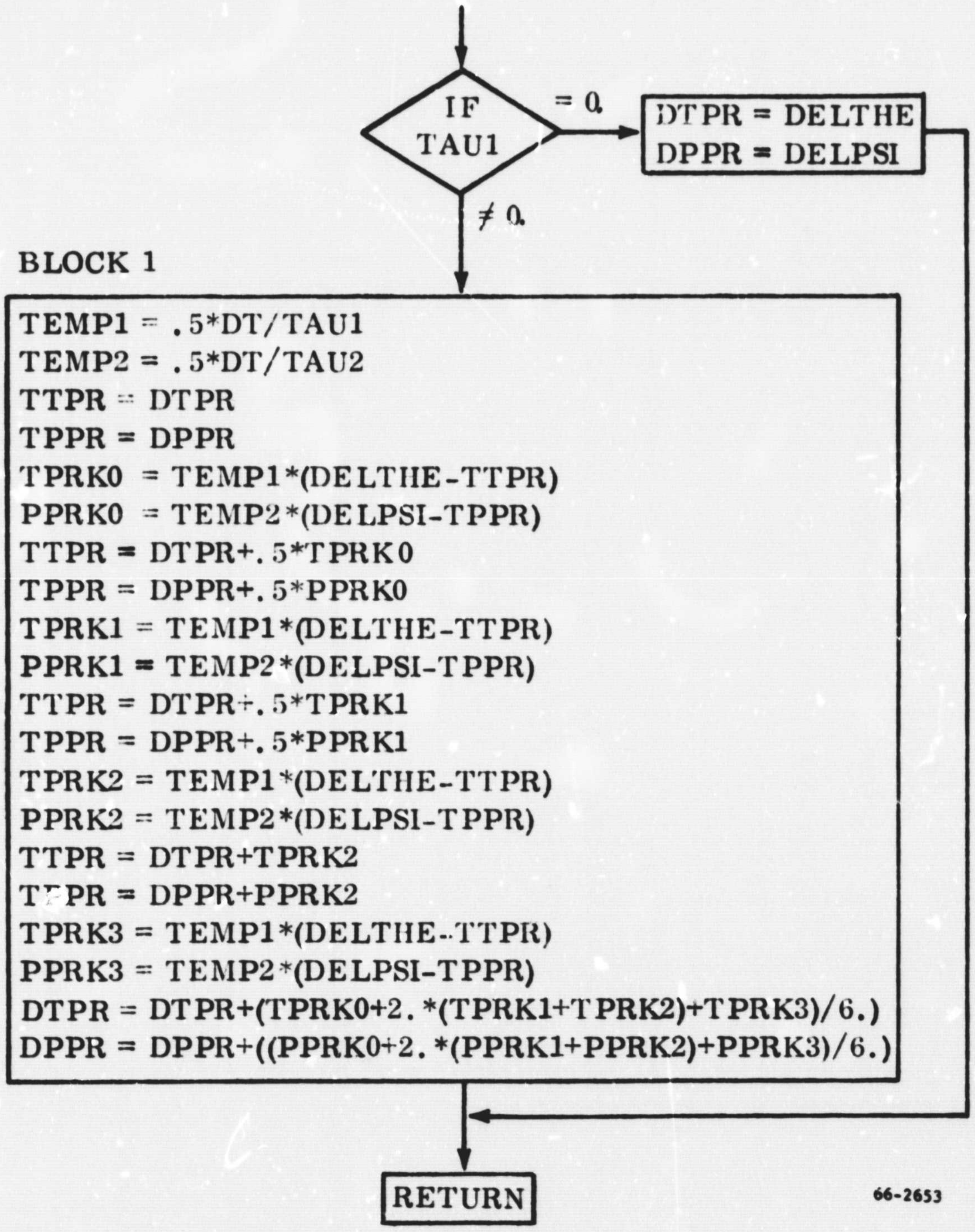
66-2644

SUBRØUTINE FILTER (DELTHE, DELPSI)

Purpose:	To generate a first-order lag in the feedback quantities
Program(s):	ISN

Flow Diagram Key:

Block 1 - Integrate the filter equations using fourth order Runge-Kutta.

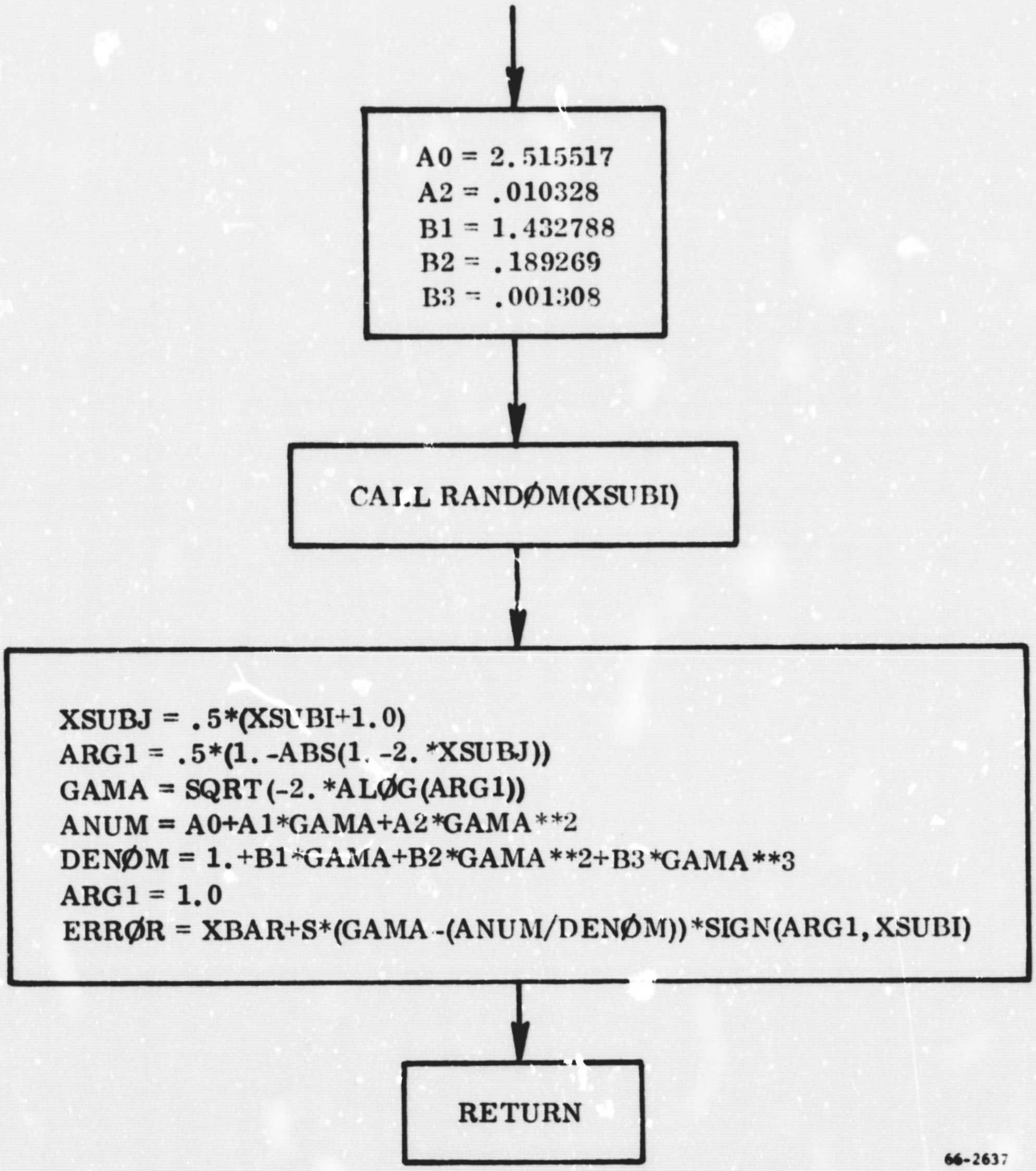


```
TEMP1 = .5*DT/TAU1
TEMP2 = .5*DT/TAU2
TTPR = DTPR
TPPR = DPPR
TPRK0 = TEMP1*(DELTHE-TTPR)
PPRK0 = TEMP2*(DELPSI-TPPR)
TTPR = DTPR+.5*TPRK0
TPPR = DPPR+.5*PPRK0
TPRK1 = TEMP1*(DELTHE-TTPR)
PPRK1 = TEMP2*(DELPSI-TPPR)
TTPR = DTPR+.5*TPRK1
TPPR = DPPR+.5*PPRK1
TPRK2 = TEMP1*(DELTHE-TTPR)
PPRK2 = TEMP2*(DELPSI-TPPR)
TTPR = DTPR+TPRK2
TPPR = DPPR+PPRK2
TPRK3 = TEMP1*(DELTHE-TTPR)
PPRK3 = TEMP2*(DELPSI-TPPR)
DTPR = DTPR+(TPRK0+2.*(TPRK1+TPRK2)+TPRK3)/6.)
DPPR = DPPR+((PPRK0+2.*(PPRK1+PPRK2)+PPRK3)/6.)
```

66-2653

SUBRØUTINE RANNØR (XBAR, S, ERRØR, XSUBI)

Purpose:	**To calculate a normally distributed random number with specified mean and standard deviation**
Program(s):	**LØF, ISN**



```
A0 = 2.515517
A2 = .010328
B1 = 1.432788
B2 = .189269
B3 = .001308
```

```
CALL RANDØM(XSUBI)
```

```
XSUBJ = .5*(XSUBI+1.0)
ARG1 = .5*(1.-ABS(1.-2.*XSUBJ))
GAMA = SQRT(-2.*ALØG(ARG1))
ANUM = A0+A1*GAMA+A2*GAMA**2
DENØM = 1.+B1*GAMA+B2*GAMA**2+B3*GAMA**3
ARG1 = 1.0
ERRØR = XBAR+S*(GAMA-(ANUM/DENØM))*SIGN(ARG1,XSUBI)
```

```
RETURN
```

66-2637

SUBRØUTINE RANDØM(XSUBI)
MAP LISTING

Purpose: To calculate a uniformly distributed random number with a magnitude $-1. \leq \text{XSUBI} \leq 1.$

Program(s): RANNØR

```
LOCATION  OPERATION  ADDRESS, TAG, DECREMENT
$IBMAP    RNDM
          ENTRY      RANDØM
RANDØM    CLA*       3,4
          FAD        =1.
          FDP        =2.
          XCA
          UFA        =Ø200000000000
          ANA        =Ø777777777
          ØRA        =1
          XCL
          MPY        K
          XCA
          SSP
          ANA        =Ø777777777
          ØRA        =Ø200000000000
          FAD        =Ø200000000000
          FSB        =.5
          XCA
          FMP        =2.
          STØ*       3,4
          TRA        1,4
K         DEC        1220703125
          END
```

VIII. PROGRAM VARIABLE DIRECTORIES

The purpose of Section VIII is to define all program variables. These are listed alphabetically for the N-BØDY, LØF, and ISN programs in Sections VIII.A., VIII.B., and VIII.C., respectively.

The column labeled SUBROUTINE APPEARANCE is used to indicate (where applicable) the subroutines in which a particular variable is used. If the variable is used in one of the main programs only, this column is left blank. Variables located in the labeled common areas (see Section VI) are indicated by the expression "(CØMMØN)" in the SUBROUTINE APPEARANCE column.

A few variables in the given programs are used as temporary storage locations for intermediate results. These variables take on several meanings depending on the place where they appear. To avoid confusion, they are designated "scratch pad variables." Similarly, fixed point variables which are used as indices in more than one or two locations are termed "general index variables."

A. N-BØDY PROGRAM

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
A11(I) A12(I) A13(I) A21(I) A22(I) A23(I) A31(I) A32(I) A33(I)	I=1, 2	Elements of transformation matrix relating planet equatorial axis system to ecliptic axis system	(CØMMØN)
A(I)	I=1, 10	Planetary semi-major axis	
A(I, J)	I=1, 6 J=1, 6	Scratch pad matrix	DETERM
ALPHX	The ratio XC/R		TRAMAT
ALPHJ(I)	I=1, 6	X direction cosine of the vehicle to planet vector	(CØMMØN)
ANUM	Scratch pad variable		AGØNY
AQUAN	Intermediate term in oblate-body acceleration calculation		
ARG1 ARG2 ARG3	X, Y, Z components of $\overline{R} \times \overline{V}$ calculation		
ARGPER ARGPR0	Vehicle argument of pericenter		
A	Vehicle orbital semi-major axis		
AUFT	Conversion from A U to feet		
AUNM	Conversion from A U to nm		
AV0 AV	Initial vehicle orbital semi-major axis		

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
AXC	X component of Newtonian force acceleration		
AXØB	X component of oblate-body acceleration		
AYC	Y component of Newtonian force acceleration		
AYØB	Y component of oblate-body acceleration		
AZC	Z component of Newtonian force acceleration		
AZØB	Z component of oblate-body acceleration		
BETAY	The ratio YC/R		TRAMAT
BETAJ(I)	I=1, 6	Y direction cosine of the vehicle to planet vector	(CØMMØN)
B(I, J)	I=1, 5 J=1, 5	Scratch pad matrix	DETERM
BUFFER(I)	I=1, 5	Intermediate input data storage area	
C2	Intermediate term in eccentricity calculation		
CAR	Cosine of the orbital argument of perigee		
CIV	Cosine of the initial orbital inclination angle		
CKMAU	Conversion from km to AU		
CMSAUD	Conversion from m/s to AU/day		
CØM	Cosine of angle between Jupiter's equatorial line of nodes and its orbital line of nodes		
CØSET	Cosine of the orbital true anomaly		
CØSI	Cosine of the orbital inclination angle		
CØSPH	Cosine of the angle between $\bar{R}$ and the orbital line of nodes		

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
CØUNT	Counter used in change-print calculations		
CØV	Cosine of orbital argument of nodes		
CRPP(I)	I=1, 10	Planetary mean longitude of perihelion	(CØMMØN)
DATE	Julian date at TIME = 0.		
DATTE	Julian date at any given time		
DAYHR	Conversion from days to hours		
DAYSEC	Conversion from days to seconds		
DCØM(I)	I=1, 10	Distance from vehicle to each planet	
DD	The determinant of the calling matrix		DETERM
DEGRAD	Conversion from degrees to radians		
DELTS	Maximum integration step size		(CØMMØN)
DELVX(I)	I=1, 4	X component of maneuver velocity increment	
DELVY(I)	I=1, 4	Y component of maneuver velocity increment	
DELVZ(I)	I=1, 4	Z component of maneuver velocity increment	
DELX(I)	I=1, 6	X component of DCØM(I)	
DELY(I)	I=1, 6	Y component of DCØM(I)	
DELZ(I)	I=1, 6	Z component of DCØM(I)	
D(I)	I=1, 6	Distance from each planet to central body cubed	(CØMMØN)

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
DELTJ	Integration step size in Jupiter's sphere of influence	
DEN	Scratch pad variable	
DENØM	Scratch pad variable	AGØNY
DUMMY(I)	I=1, 69 Intermediate input table	
EE(I)	I=1, 10 Planetary orbit eccentricities	(CØMMØN)
ELEIND	Program-set element print indicator	
ELEPR	Input-set element print switch	
EP	Vehicle orbit eccentricity	
EPS(I)	I=1, 10 Planetary mean longitude at epoch	(CØMMØN)
E	Two-body orbital constant $= V^2 - 2\mu/R$	
E1	Weighted measure of percentage integration error per variable	INTEG
E2	Weighted measure of percentage integration error per variable	INTEG
ETA0	Initial vehicle true anomaly in degrees	
ETAP	True anomaly	
ETAV	Initial vehicle true anomaly in radians	
EV	Initial vehicle orbit eccentricity	
FLAT(I)	I=1, 3 Planetary flattening constant for oblate bodies	
FPSAUD	Conversion from fps to AU /day	
FUN(I)	I=1, 6 Intermediate term in Newtonian acceleration calculation	
GAMJ(I)	I=1, 6 Z direction cosine of vehicle-to-planet vector	(CØMMØN)

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
GAMZ	The ratio ZC/R	
HGEØC	Planetocentric altitude	
HGEØD	Planetodetic altitude	
HØ	Integration step size	(CØMMØN)
HRSEC	Conversion from hours to seconds	
H	Integration step-size intermediate variable	(CØMMØN)
I1	Integration counter-number of integrations	(CØMMØN)
IA	Scratch pad variable-run identification number	ZPRN
IALFA	Integration counter-number of times through R-K	(CØMMØN)
IBETA	Integration counter-insures four R-K points before starting Adams method	(CØMMØN)
ICN1	Integration counter-used to update J index in referencing last complete integrated point	(CØMMØN)
ICN2 ICN3	Integration counters to assure that enough points are available to allow doubling the step size	(CØMMØN)
ICN4	Number of "PRINT" - sized time increments since TIME = TBASE	
ICØUNT	Output switch-used to signal program when it is time to print	
IERR	Diagnostic indicator	

A. N-BODY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
IEVTME	Change-print constant - signifies normal print time	
IGAM	Integration switch - indicates predictor or corrector phase	(CØMMØN)
INDR	Integration switch - indicates when doubling is possible	(CØMMØN)
IPINT	Internally-set print indicator - fixed point version of the input variable PINT	
IPR1	Observation time index - indicates previous observation time	
IPR2	Observation time switch	
IPR	Observation time index - indicates current observation time	
IP	General index variable	
I	General index variable	
ISWT1	Integration switch - indicates R-K or Adams 4-point scheme	(CØMMØN)
ISWT2	Integration switch- indicates input choice of R-K or Adams 4-point scheme	(CØMMØN)
ISWT3	Integration switch - indicates input choice of constant or variable step size	(CØMMØN)
ISWT4	Rectification indicator switch	
ISWT5	Print return selector switch - channels flow after printing	
ISWT6	First time switch - indicates initial computations	

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ISWT7	Thrust time indicator - signals possible thrust times	
ISWT8	Change print delay switch	
ISWT9	Last time indicator - signals final computations	
ISWT10	Pre-rectification and post-rectification print switch	
ISWT11	Change-print switch	
ISWT12	Switch used to specify a post-rectification print	
ISWT13	Switch used to control integration near closest approach to Jupiter	
ITH	Thrust maneuver index variable	
IWC	Input word count per card	
IZ	General index variable	DETERM
J	Integration counter - specifies first location of each variable in integration tables	(CØMMØN)
JI JJ JK JL JM	General index variables	INTEG, UPDAT INTEG, UPDAT INTEG, UPDAT UPDAT UPDAT
K1	Scratch pad variable - indicates the row of the largest element in the A(I, J) matrix	DETERM
K2	Scratch pad variable - indicates the column of the largest element in the A(I, J) matrix	DETERM

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
KK(I)	I=1, 7	Table of planet index codes representing the current set of celestial bodies considered in the equations of motion	
KL	Location in the DUMMY table of the given input variable being initialized		
KØUNT	Change-print counter		
K	Integration routine index-references a particular integrated variable		(CØMMØN)
LABEL(I)	I=1, 10	Hollerith statements containing planet names	
L1	General index variables		N-BØDY, DETERM, SETINT
L2			
L3			
L4			
L5			
LI			TRAMAT
LJ			
LK			
LL			DETERM, TRAMAT
LM			TRAMAT
LØC	Location in DUMMY table of first data word on an input card		
LP	General index variables		N-BØDY, DETERM, SETINT
L			
M1	Thrust time-to-observation time conversion indices		
M2			
M3			
M4			
M5			
MAXØB	Number of observation times		

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
MAXTHR	Number of thrust maneuvers	
M	General index variable	
N1 N2 N3 N4 N5 N6 N7 N8	Indices representing particular planets being considered in the equations of motion	
NI	General index variable	UPDAT
NN	General index variable	UPDAT
NN	Switch to adjust the sign of the required determinant	DETERM
NP	General index variable	UPDAT
NPRINT	Integration step indicator-signals when a time multiple of DELTS has been reached	(CØMMØN)
NPT	Integration counter - controls the setting of NPRINT	UPDAT
NU	Maximum number of spaces in the integration tables (8 spaces per integrated variable)	(CØMMØN)
N	Integration counter - indicates last location of each variable in integration tables	(CØMMØN)
ØBSIND	Number of observation times	
ØM0	Initial vehicle orbital argument of nodes in degrees	
ØMMC(I)	I=1, 10 Cosines of the planetary argument of nodes	(CØMMØN)

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ØMM(I)	I=1, 10 Planetary argument of nodes	(CØMMØN)
ØMMS(I)	I=1, 10 Sines of the planetary argument of nodes	(CØMMØN)
ØMP	Vehicle orbital argument of nodes	
ØMV	Initial orbital argument of nodes in radians	
PB	Vehicle orbital semi-latus rectum	
PCN1A PCN1B PCN1	Print frequencies within Earth's sphere of influence	
PCN2	Print frequency within Sun's sphere of influence	
PCN3 PCN4	Print frequencies within Jupiter's sphere of influence	
PCN	A general print frequency variable	
PERIG	Vehicle orbital argument of pericenter	
PF1X	Partial derivative of the X acceleration component with respect to XC	TRAMAT
PF1XØB PF1XSP	Oblate body and Newtonian force contributions to PFIX	TRAMAT
PF1Y	Partial derivative of the X acceleration component with respect to YC	TRAMAT
PF1YØB PF1YSP	Oblate body and Newtonian force contributions to PF1Y	TRAMAT
PF1Z	Partial derivative of the X acceleration component with respect to ZC	TRAMAT
PF1ZØB PF1ZSP	Oblate body and Newtonian force contributions to PF1Z	TRAMAT

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
PF2X	Partial derivative of the Y acceleration component with respect to XC	TRAMAT
PF2XØB PF2XSP	Oblate body and Newtonian force contributions to PF2X	TRAMAT
PF2Y	Partial derivative of the Y acceleration component with respect to YC	TRAMAT
PF2YØB PF2YSP	Oblate body and Newtonian force contributions to PF2Y	TRAMAT
PF2Z	Partial derivative of the Y acceleration component with respect to ZC	TRAMAT
PF2ZØB PF2ZSP	Oblate body and Newtonian force contributions to PF2Z	TRAMAT
PF3X	Partial derivative of the Z acceleration component with respect to XC	TRAMAT
PF3XØB PF3XSP	Oblate body and Newtonian force contributions to PF3X	TRAMAT
PF3Y	Partial derivative of the Z acceleration component with respect to YC	TRAMAT
PF3YØB PF3YSP	Oblate body and Newtonian force contributions to PF3Y	TRAMAT
PF3Z	Partial derivative of the Z acceleration component with respect to ZC	TRAMAT
PF3ZØB PF3ZSP	Oblate body and Newtonian force contributions to PF3Z	TRAMAT
PHI	Angle between $\overline{R}$ and orbital line of nodes	
PH1(1)	I=1, 6 First row of the transition matrix	(CØMMØN)

A. N-BODY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
PH2(I)	I=1, 6	Second row of the transition matrix	(COMMON)
PH3(I)	I=1, 6	Third row of the transition matrix	(COMMON)
PH4(I)	I=1, 6	Fourth row of the transition matrix	(COMMON)
PH5(I)	I=1, 6	Fifth row of the transition matrix	(COMMON)
PH6(I)	I=1, 6	Sixth row of the transition matrix	(COMMON)
PINT	Input switch - signals program whether or not to print number of integration steps and maximum integration step size - see Section III		
PI	The constant π		
PO(I)	I=1, 6	Scratch pad variable	DETERM
PPT1 PPT2 PPT	Scratch pad variables		UPDAT
PRIN1A PRIN1B PRINT1 PRINT2 PRINT3	Integration step sizes - see Section III		
PRINT	Instantaneous value of DELTS		
PRTCH	Intermediate integration step size variable		

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
Q11 Q12 Q21 Q22 Q31 Q32	Elements of the transformation matrix from a two-dimensional, orbit-referenced axis system to the three-dimensional, ecliptic reference frame	PLANET
Q	Mathematical term in Newtonian acceleration calculation	
RAM	Scratch pad variable	TRAMAT
RA	Scratch pad variable	TRAMAT
RADD	Planetary equatorial radius in km	
RAQ	The ratio REQ(N1)/AUNM	(CØMMØN)
REQ(I)	I=1, 10 Planetary equatorial radius	(CØMMØN)
RKTME	Time of last R-K calculation in integration routine	(CØMMØN)
RPØL(I)	I=1, 2 Planetary polar axis (Earth and Jupiter, respectively)	
R	Instantaneous vehicle orbital radius measured from dominant central body	(CØMMØN)
R0	Initial vehicle orbital radius	
RFRAC	The ratio RAQ/R	TRAMAT
RR	Planetary heliocentric orbital radius squared	PLANET
RSPHE	Radius of Earth's sphere of influence	
RSPHJ	Radius of Jupiter's sphere of influence	
RUN	Run identification number	

A. N-B0DY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
SAR	Sine of ARGPER	
SET	Sine of ETAV	
SINET	Sine of ETAP	
SINPH	Sine of PHI	
SIP	Sine of the inclination angle of Jupiter's equator to the ecliptic	
SIV	Sine of YIV	
SL0TMX	Maximum integration step size used during the generation of a given trajectory	
S0V	Sine of 0MV	
T0	Initial time measured from DATE	
TAB(I)	I=1, 25 Thrust maneuver time indicator table	
TAMP1	Intermediate term in Newtonian acceleration calculation	
TAMP	Intermediate term in Newtonian acceleration calculation	
TBASE	Time corresponding to the last time the normal print increment was changed	
TBASE1	Temporary storage for TBASE	
TCHK	Jupiter passage time constant - equals the time at which the program should re-establish PRINT3 as the maximum integration step size after the vehicle's closest approach to Jupiter	
TCN	Floating point version of ICN4	

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
TEMP1 TEMP2 TEMP3 TEMP4 TEMP	Scratch pad variables	
TEMP31	Oblate body J_2 contribution to PF1XØB	TRAMAT
TEMP32	Oblate body J_2 contribution to PF1YØB	TRAMAT
TEMP33	Oblate body J_2 contribution to PF1ZØB	TRAMAT
TEMP34	Oblate body J_2 contribution to PF2XØB	TRAMAT
TEMP35	Oblate body J_2 contribution to PF2YØB	TRAMAT
TEMP36	Oblate body J_2 contribution to PF2ZØB	TRAMAT
TEMP37	Oblate body J_2 contribution to PF3XØB	TRAMAT
TEMP38	Oblate body J_2 contribution to PF3YØB	TRAMAT
TEMP39	Oblate body J_2 contribution to PF3ZØB	TRAMAT
TEMP41	Oblate body J_3 contribution to PF1XØB	TRAMAT
TEMP42	Oblate body J_3 contribution to PF1YØB	TRAMAT
TEMP43	Oblate body J_3 contribution to PF1ZØB	TRAMAT
TEMP44	Oblate body J_3 contribution to PF2XØB	TRAMAT
TEMP45	Oblate body J_3 contribution to PF2YØB	TRAMAT
TEMP46	Oblate body J_3 contribution to PF2ZØB	TRAMAT
TEMP47	Oblate body J_3 contribution to PF3XØB	TRAMAT
TEMP48	Oblate body J_3 contribution to PF3YØB	TRAMAT

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
TEMP49	Oblate body J_3 contribution to PF3ZØB	TRAMAT
TEMP51	Oblate body J_4 contribution to PF1XØB	TRAMAT
TEMP52	Oblate body J_4 contribution to PF1YØB	TRAMAT
TEMP53	Oblate body J_4 contribution to PF1ZØB	TRAMAT
TEMP54	Oblate body J_4 contribution to PF2XØB	TRAMAT
TEMP55	Oblate body J_4 contribution to PF2YØB	TRAMAT
TEMP56	Oblate body J_4 contribution to PF2ZØB	TRAMAT
TEMP57	Oblate body J_4 contribution to PF3XØB	TRAMAT
TEMP58	Oblate body J_4 contribution to PF3YØB	TRAMAT
TEMP59	Oblate body J_4 contribution to PF3ZØB	TRAMAT
TEMPC	Cosine of the planetary argument of perihelion	PLANET
TEMPS	Sine of the planetary argument of perihelion	PLANET
TEP	Scratch pad variable	DETERM
TEST	Indicator used to specify two-body trajectory - see Section III	
THE	Flight path angle in degrees	
THET	Flight path angle in radians	
THRIND	Number of thrust maneuvers	
TIME	Time measured from DATE	
TM1	Time from epoch to launch	(CØMMØN)

A. N-BODY PRØGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
TMAX	Maximum trajectory time		
TØ(I)	I=1, 6	Scratch pad variable list	DETERM
TØB(I)	I=1, 26	Table specifying the times at which observations are to be taken	
TTHR(I)	I=1, 4	Table specifying the times at which thrust maneuvers are to be made	
TRAMA	Input switch to control transition matrix calculation		
TRAN	Internal switch to control transition matrix calculation		(CØMMØN)
TRAPRN	Input switch to control transition matrix print - see Section III		
T	Trajectory time measured from epoch		PLANET
TWØPI	2π		
V2	Velocity squared		
VARBH	Variable integration step size indicator		
VA	Local horizontal velocity component		
VEL	Velocity		
VR	Radial velocity component		
VX0	Initial X component of velocity		
VXDØT	First time derivative of X velocity component		
VXR(I)	I=1, 2	Planetary X component of velocity	(CØMMØN)

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
VX	X component of velocity		(CØMMØN)
VXB VYB	Orbit-referenced components of planetary velocity		PLANET
VXX	Printout X component of velocity		
VY0	Initial Y component of velocity		
VYDØT	First time derivative of Y velocity component		
VYR(I)	I=1, 2	Planetary Y component of velocity	(CØMMØN)
VY	Y component of velocity		(CØMMØN)
VYY	Printout Y component of velocity		
VZ0	Initial Z component of velocity		
VZDØT	First time derivative of Z velocity component		
VZR(I)	I=1, 2	Planetary Z component of velocity	(CØMMØN)
VZ	Z component of velocity		(CØMMØN)
VZZ	Printout Z velocity component		
W(I)	I=1, 42	Weighting factors used in integration error control process	(CØMMØN)
XB	Orbit-referenced component of planetary position		PLANET
XC0	Initial X position component		
XCC	Printout X position component		
XC	X position component		(CØMMØN)
XICN4	Floating point version of ICN4		
XIPE	Inclination angle of Earth's equator to the ecliptic		

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XIP	Inclination angle of Jupiter's equator to the ecliptic	
XJ2(I)	I=1, 2 Second harmonic coefficient in oblate body component	(CØMMØN)
XJ3(I)	I=1, 2 Third harmonic coefficient in oblate body acceleration	(CØMMØN)
XJ4(I)	I=1, 2 Fourth harmonic coefficient in oblate body acceleration	(CØMMØN)
XLØNG	Vehicle inertial longitude	
XL(I)	I=1, 10 X direction cosines of vehicle-planet line of sight	
XMM	Vehicle orbital mean motion	
XMØDE	Input mode selector switch - see Section III	
XM(I)	I=1, 10 Y direction cosine of vehicle-planet line of sight	
XMTHD	Integration scheme indicator - see Section III	
XMU(I)	I=1, 10 Planetary gravitational constants	(CØMMØN)
XN1	Input value for N1	
XNMFT	Conversion from nm to ft	
XNN(I)	I=1, 10 Planetary mean motion values	(CØMMØN)
XNMKM	Conversion factor from nm to km	
XN(I)	I=1, 10 Z direction cosine of vehicle-planet line of sight	
XNUP	Vehicle longitude measured from ascending node	

A. N-BØDY PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
XR(I)	I=1, 10	Planetary X position component	(CØMMØN)
YB	Orbit-referenced component of planetary position		PLANET
YC0	Initial Y component of position		
YCC	Printout Y position component		
YC	Y component of position		(CØMMØN)
YI0	Initial vehicle orbital inclination in degrees		
YIC(I)	I=1, 10	Cosine of the planetary orbital inclination angles	(CØMMØN)
YIP	Vehicle orbital inclination		
YIS(I)	I=1, 10	Sine of the planetary orbital inclination angles	(CØMMØN)
YIV	Initial vehicle orbital inclination in radians		
YNN	Floating point version of NN		DETERM
YP(I)	I=1, 336	First derivative table for integration routines	(CØMMØN)
YR(I)	I=1, 10	Planetary Y component of position	(CØMMØN)
Y(I)	I=1, 336	Variable table for integration routines	(CØMMØN)
ZC0	Initial Z position component		
ZCC	Printout Z position component		
ZC	Z position component		
ZR(I)	I=1, 10	Planetary Z position component	(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
A10	Scratch pad variable	
A11	First row of elements in transformation matrix from Jupiter equatorial frame to ecliptic frame	(CØMMØN)
A12		(CØMMØN)
A13		(CØMMON)
A1	Scratch pad variable	MAIN, NUMER, ACCEL, PHINT1
A1SQ	Scratch pad variable	ACCEL, PHINT1
A21	Second row of elements in transformation matrix from Jupiter equatorial frame to ecliptic frame	(CØMMØN)
A22		(CØMMØN)
A23		(COMMON)
A2	Scratch pad variable	MAIN, PHINT1, ACCEL
A31	Third row of elements in transformation matrix from Jupiter equatorial frame to ecliptic frame	(CØMMØN)
A32		(CØMMØN)
A33		(CØMMØN)
A3	Scratch pad variables	MAIN, ACCEL, PHINT1
A4		MAIN, ACCEL, PHINT1
A5		MAIN, PHINT1
A6		MAIN, ACCEL, PHINT1
A7		MAIN, ACCEL
A8		MAIN, ACCEL
A9		MAIN, ACCEL
AA(I)	I=1, 10 Planetary semi-major axes	(CØMMØN)
AAX	Statistical mean of errors in knowledge of initial position and velocity components	
AAY		
AAZ		
AVX		
AVY		
AVZ		
A(I, J)	I=1, 6 J=1, 6 Scratch pad matrix	DETERM

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
AKI(I, J, K)	I=1, 6 J=1, 2 K=1, 100	Composite table of all weighting function matrices	
AK(I, J)	I=1, 6 J=1, 2	Navigation weighting function matrix	
ALPHAM	Measured right ascension of observed vehicle-planet vector		
ALPHAT	True right ascension of observed vehicle-planet vector		
ALPHJ(I)	I=1, 7	X direction cosine of observed planet	PHINT1
ALPHX2	The ratio $(XC/R)^2$		PHINT2
ALPHX	The ratio XC/R		PHINT2
ALPPR	Predicted right ascension of observed planet		
ANUM	Scratch pad variable		AGØNY
ARG3	Scratch pad variable		AGØNY
AUNM	Conversion factor from AU to nm		(CØMMØN)
B1	Scratch pad variables		PHINT1
B1SQ			PHINT1
B2			PHINT1
B31	Oblate body J_2 contribution to the partial derivative of the X acceleration with respect to XC		PHINT1
B32	Oblate body J_2 contribution to the partial derivative of the X acceleration with respect to YC		PHINT1

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
B33	Oblate body J_2 contribution to the partial derivative of the X acceleration with respect to ZC	PHINT1
B34	Oblate body J_2 contribution to the partial derivative of the Y acceleration with respect to XC	PHINT1
B35	Oblate body J_2 contribution to the partial derivative of the Y acceleration with respect to YC	PHINT1
B36	Oblate body J_2 contribution to the partial derivative of the Y acceleration with respect to ZC	PHINT1
B37	Oblate body J_2 contribution to the partial derivative of the Z acceleration with respect to XC	PHINT1
B38	Oblate body J_2 contribution to the partial derivative of the Z acceleration with respect to YC	PHINT1
B39	Oblate body J_2 contribution to the partial derivative of the Z acceleration with respect to ZC	PHINT1
B3 B4	Scratch pad variables	PHINT1 PHINT1
B51	Oblate body J_4 contribution to the partial derivative of the X acceleration with respect to XC	PHINT1
B52	Oblate body J_4 contribution to the partial derivative of the X acceleration with respect YC	PHINT1

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
B53	Oblate body J_4 contribution to the partial derivative of the X acceleration with respect to ZC		PHINT1
B54	Oblate body J_4 contribution to the partial derivative of the Y acceleration with respect to XC		PHINT1
B55	Oblate body J_4 contribution to the partial derivative of the Y acceleration with respect to YC		PHINT1
B56	Oblate body J_4 contribution to the partial derivative of the Y acceleration with respect to ZC		PHINT1
B57	Oblate body J_4 contribution to the partial derivative of the Z acceleration with respect XC		PHINT1
B58	Oblate body J_4 contribution to the partial derivative of the Z acceleration with respect to YC		PHINT1
B59	Oblate body J_4 contribution to the partial derivative of the Z acceleration with respect to ZC		PHINT1
B(I,J)	I=1,5 J=1,5	Scratch pad matrix	DETERM
BETAJ(I)	I=1,7	Y direction cosine of the observed planet	PHINT1
BETAY2	The ratio $(YC/R)^2$		PHINT1
BETAY	The ratio YC/R		PHINT1
BUFFER(I)	I=1,5	Temporary input table	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
CIP	Cosine of the inclination angle between Jupiter's equatorial plane and its orbital plane	
CKMAU	Conversion factor from km to A U	
CMSAUD	Conversion factor from m/s to A U /day	
CN10	The number of completed integration steps since TIME=TBASE plus one	
CØM	Cosine of XPØM	
CØRR(I)	I=1, 6 Navigated state vector corrections	
CRPP(I)	I=1, 10 Planetary mean longitude of perihelion	(CØMMØN)
CT	Noise term introduced into covariance matrix calculation	
DATAPR	Data print option switch	
DATE	Julian date at TIME=0.	
DD	The determinant of the calling matrix	DETERM
DEGRAD	Conversion factor from degrees to radians	
DELALF	Measurement error in the right ascension of the vehicle-observed planet vector	
DELAL	DELALF converted to arc sec	
DELDEL	Measurement error in declination of the observed vehicle-planet vector	
DELD	DELDEL converted to arc sec	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
DELPR	Predicted declination of the observed planet based on navigated vehicle position	
DELTAM	Measured declination of the observed vehicle-planet vector	
DELTAT	True declination of the observed vehicle-planet vector	
DELTPH	Integration step size for transition matrix	(CØMMØN)
DELTS1	Temporary storage for the variable DELTS	
DELTS	Integration step size for state variables	(CØMMØN)
DELVX0 DELVY0 DELVZ0	Error in knowledge of initial velocity	
DELX0 DELY0 DELZ0	Error in knowledge of initial position	
DELX(I) DELY(I) DELZ(I)	I=1, 6 I=1, 6 I=1 [illegible] Inertial components of planet-to-vehicle vector	ACCEL
DENØM	Scratch pad variable	AGØNY
DET	Determinant of intermediate matrix in weighting function calculation	
DJ	Distance between vehicle and Jupiter	
DUMMY(I)	I=1, 80 Intermediate input table	
D(I)	I=1, 7 Cubed distance between vehicle and i^{th} planet	ACCEL, PHINT1

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
E1	Weighted estimate of actual integration error per variable		INTEG
E2	Weighted estimate of percentage integration error per variable		INTEG
EE(I)	I=1, 10	Planetary orbit eccentricity	(CØMMØN)
EPS(I)	I=1, 10	Planetary mean longitude at epoch	(CØMMØN)
EVX EVY EVZ	Error in navigated estimate of velocity		(CØMMØN)
EX EY EZ	Error in navigated estimate of position		(CØMMØN)
FUN(I)	I=1, 6	Intermediate step in acceleration calculation	ACCEL
GAMJ(I)	A direction cosine of observed planet		PHINT1
GAMZ	The ratio ZC/R		PHINT1
HH(I, J)	I=1, 2 J=1, 6	Geometry matrix	
HØ	Current integration step size		(CØMMØN)
H	Intermediate integration step size variable		(CØMMØN)
I1	Integration counter-number of integrations		(CØMMØN)
IALFA	Integration counter-number of times through R-K		(CØMMØN)
IBETA	Integration counters-insures four R-K points before starting Adams method		(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ICN10	Counter used to tabulate number of integration steps performed	
ICN1	Integration counter-used to update J index in referencing last complete integrated data point	(CØMMØN)
ICN2 ICN3	Integration counters to insure that enough points are available to allow doubling the step size	(CØMMØN)
ICN4	Integer number of integration steps since TIME=TBASE	
IGAM	Integration switch-signals predictor or corrector phase of scheme operation	(CØMMØN)
II	General index variable	UPDAT
INDR	Integration switch-indicates when doubling the step size is possible	(CØMMØN)
IPLAN	The number of celestial bodies (other than the center of coordinates) to be included in the navigation force model	(CØMMØN)
IP	General index variable	PLANET, ACCEL, PHINT1, PHINT2, SETINT, INTEG, UPDAT
I	Index referencing particular measurement time	(CØMMØN)
IRUN	Case number for printout purposes	ZPRIN
ISWT10	Switch set in PHINT1 to control transition matrix integration in PHINT2	(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ISWT1	Integration switch-indicates R-K or Adams 4-point scheme	(CØMMØN)
ISWT2	Integration switch-indicates input choice of R-K or Adams 4-point	(CØMMØN)
ISWT3	Integration switch-indicates input choice of constant or variable step size	(CØMMØN)
ISWT5	Switches used in rectification logic to keep track of current coordinate center	
ISWT6		
ISWT7	Switch to signal when rectification has occurred	
IWC	Number of data words on one input card	
IZ	General index variables	DETERM, ACCEL
JI		INTEG, UPDAT
JJ		INTEG, UPDAT
JK		INTEG, UPDAT
JL		UPDAT
JM		UPDAT
J	Special index used to indicate first location of each variable in integration tables	(CØMMØN)
K1	Scratch pad variable-indicates the row of the largest element in the A(I, J) matrix	DETERM, ZPRIN
K2	Scratch pad variable-indicates the column of the largest element in the A(I, J) matrix	DETERM, ZPRIN
K3	General index variable	ZPRIN
KASE	Index referencing a particular trajectory	(CØMMØN)
KK(I)	I=1, 7 Set of planet index codes in use at any given time	(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
KL	Index referencing the location of an input variable in the DUMMY(I) list	
KMAX	Maximum number of trajectories to be generated in one run	(CØMMØN)
KØN	Number of integration steps between measurement times	
K	Index referencing a particular integrated variable	(CØMMØN)
L1	General index variable	MAIN, DETERM
LABEL(M)	M=1, 10 List of planet names for printout purposes	MAIN, DETERM, PLANET
LL	General index variable	DETERM
LØC	Location in the DUMMY(I) table of the first data word on an input card	
L	General index variable	MAIN, DETERM, MATRAX, ACCEL, PHINT2, PLANET
M1	General index variable	MAIN, DETERM, ZPRIN, PLANET, MATRAX, ACCEL, PHINT1, PHINT2
N1	Planet index code representing the dominant central force field	MAIN, PLANET, ACCEL PHINT1
N1T(I)	I=1, 100 Table of planet index codes indicating the truth reference body at a given measurement time	
N2 N3 N4 N5 N6 N7	Planet index codes representing the celestial bodies (other than the dominant one) to be considered in the equations of motion	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
NMAX	Maximum number of measurement times	(CØMMØN)
NN	Scratch pad variable	DETERM, UPDAT
NPRINT	Print switch-signals when a time multiple of DELTS has been reached	(CØMMØN)
NP	General index variable	UPDAT
NPT	Integration counter-controls setting of NPRINT	UPDAT
N	Special index-indicates last location of each variable in integration tables	(CØMMØN)
NU	Maximum number of locations used in the integration tables (= 8 x the number of integrated variables)	(CØMMØN)
ØMMC(I)	I=1, 10 Cosine of the planetary argument of nodes	(CØMMØN)
ØMM(I)	I=1, 10 Planetary argument of nodes	(CØMMØN)
ØMMS(I)	I=1, 10 Sine of the planetary argument of nodes	(CØMMØN)
PB	Planetary semi-latus rectum	PLANET
PF1XSP PF1YSP PF1ZSP PF2XSP PF2YSP PF2ZSP PF3XSP PF3YSP PF3ZSP	Spherical force field contribution to the partial derivatives of the acceleration components with respect to the inertial position components	PHINT1
PHDØT(I, J)	I=1, 6 J=1, 6 Matrix of the derivatives of the transition matrix elements	(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
PHIMAT(I, J)	I=1, 6 J=1, 6	Transition matrix	(CØMMØN)
PHTME	Intermediate transition matrix integration step size - used when rectification occurs between integral multiples of DELTPH		
PI	Mathematical constant π		MAIN, ACØNY
PØ(I)			DETERM
PPT1	I = 1, 6 Scratch pad variable		UPDAT
PPT2	Scratch pad variables		UPDAT
PPT			UPDAT
P(I, J)	I=1, 6 J=1, 6	Covariance matrix	
PXI(I) PYI(I) PZI(I)	I=1, 100	Inertial position of the observed planet at a given measurement time	
Q11 Q12 Q21 Q22 Q31 Q32	Elements of the transformation matrix from a two-dimensional orbit-referenced axis frame to the inertial three-dimensional ecliptic frame		PLANET
Q	Intermediate variable in acceleration calculation		ACCEL
R3	R^3		ACCEL, PHINT1
RDET	Scratch pad variable		
REQ(I)	I=1, 10	Planetary equatorial radii	(CØMMØN)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
RFRAC	The ratio REQ(N1)/(R * AUNM)	(CØMMØN)
RKTME	The starting time of the last R-K integrated point	(CØMMØN)
RR	Planetary heliocentric orbital radius squared	PLANET
R	Distance from vehicle to the dominant central body	(CØMMØN)
RSPHJ	Radius of the sphere of influence of Jupiter	
RUN	Run identification number	
SECRAD	Conversion factor from arc sec to radians	
SIGMA(I)	I=1, 6 Standard deviation of the errors in knowledge of the initial position and velocity components also navigation system estimates of the same in metric units	
SIGMU(I)	I=1, 6 SIGMA(I) converted to internal units	
SIGP0	Standard deviation of error in p measurement in arc sec	
SIGPQ0	Correlation factor between p and q measurements in arc sec	
SIGPQ	SIGPQ0 converted to radians	
SIGP	SIGP0 converted to radians	
SIGQ0	Standard deviation of error in q measurement in arc sec	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
SIGQP0	Correlation factor between q and p measurements in arc sec	
SIGQP	SIGQP0 converted to radians	
SIGQ	SIGQ0 converted to radians	
SIGVX SIGVY SIGVZ	Standard deviation of navigated velocity components and second moment about zero for same data	NUMER
SIGX SIGY SIGZ	Standard deviation of navigated position components and second moment about zero for same data	NUMER
SIP	Sine of the inclination of Jupiter's equatorial plane with respect to its orbital plane	
SØM	Sine of XPØM	
STDA10	Standard deviation of the applied measurement errors in right ascension of the vehicle-observed planet vector in arc sec (coordinate center = Jupiter)	
STDA20	Standard deviation of the applied measurement errors in right ascension of the vehicle-observed planet vector in arc sec (coordinate center = Sun)	
STDA2	STDA20 converted to radians	
STDD10	Standard deviation of the applied measurement errors in declination of the vehicle-observed planet vector in arc sec (coordinate center = Jupiter)	
STDD1	STDD10 converted to radians	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
STDD20	Standard deviation of the applied errors in declination of the vehicle-observed planet vector in arc sec (coordinate center = Sun)	
STDD2	STDD20 converted to radians	
STDDA	Current standard deviation of the measurement error in right ascension of the observed planet	
STDDD	Current standard deviation of the measurement error in declination of the observed planet	
SUMVX2(I)	I=1,100 Sum of the square of the navigated errors in VX at any given measurement time	NUMER
SUMVX(I)	I=1,100 Sum of the navigated errors in VX at any given measurement time	NUMER
SUMVY2(I)	I=1,100 Sum of the square of navigated errors in VY at any given measurement time	NUMER
SUMVY(I)	I=1,100 Sum of the navigated errors in VY at any given measurement time	NUMER
SUMVZ2(I)	I=1,100 Sum of the square of navigated errors in VZ at any given measurement time	NUMER
SUMVZ(I)	I=1,100 Sum of the navigated errors in VZ at any given measurement time	NUMER

B. LINEAR 0PTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
SUMX2(I)	I=1, 100	Sum of the square of the navigated errors in XC at any given measurement time	NUMER
SUMX(I)	I=1, 100	Sum of the navigated errors in XC at any given measurement time	NUMER
SUMY2(I)	I=1, 100	Sum of the square of the navigated errors in YC at any given measurement time	NUMER
SUMY(I)	I=1, 100	Sum of the navigated errors in YC at any given measurement time	NUMER
SUMZ2(I)	I=1, 100	Sum of the square of the navigated errors in ZC at any given measurement time	NUMER
SUMZ(I)	I=1, 100	Sum of the navigated errors in ZC at any given measurement time	NUMER
T0	Initial time measured from DATE		
T1(I)	I=1, 100	Input table of measurement times	
TBASE1	Temporary storage for TBASE		
TBASE	Time at which integration step size was last changed		
TEMP1(I, J)	I=1, 6 J=1, 6	Scratch pad matrix	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
TEMP1	Scratch pad variable	PLANET
TEMP2(I, J)	I=1, 6 J=1, 6 Scratch pad matrix	
TEMP2	Scratch pad variable	PLANET
TEMPC	Cosine of the planetary argument of perihelion	PLANET
TEMP	Scratch pad variable	MAIN, PLANET, ACCEL
TEMPS	Sine of the planetary argument of perihelion	PLANET
TEP	Scratch pad variable	DETERM
TIME	Trajectory time measured from DATE	(CØMMØN)
TM1	Number of days from epoch to DATE	(CØMMØN)
TME	Time measured from epoch	PLANET
TØ(I)	I=1, 6 Scratch pad variable	DETERM
T(I)	I=1, 100 Table of measurement times	(CØMMØN)
TTIME	Time between navigation measurements	
TWØPI	2π	MAIN, PLANET
VARBH	Integration switch-specifies constant or variable step size	
VX0	Initial X component of velocity	
VXB	Intermediate variable; orbit-referenced X component of planetary velocity	PLANET

B. LINEAR 0PTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
VXC1	Navigated value of VXC before updating at a measurement time	
VXC2	Navigated value of VXC after updating at a measurement time	
VXC	X component of vehicle velocity	(C0MM0N)
VXD0T	X component of vehicle acceleration	(C0MM0N)
VXMEAN	Statistical mean of navigated errors in VXC	NUMER
VXR	Inertial X component of Jupiter's velocity	(C0MM0N)
VX(I)	I=1,100 Stored value of the true X component of vehicle velocity at any given measurement time	
VXT1	True vehicle X velocity component at any given measurement time	
VY0	Initial Y component of vehicle velocity	
VYB	Intermediate variable; orbit-referenced component of Jupiter's velocity	PLANET
VYC1	Navigated value of VYC before updating at a measurement time	
VYC2	Navigated value of VYC after updating at a measurement time	
VYC	Inertial Y component of vehicle velocity	(C0MM0N)
VYD0T	Y component of vehicle acceleration	(C0MM0N)
VYMEAN	Statistical mean of navigated errors in VYC	NUMER
VYR	Inertial Y component of Jupiter's velocity	(C0MM0N)

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
VY(I)	I=1,100 Stored value of the true Y component of vehicle velocity at any given measurement time	
VYT1	True vehicle Y velocity component at any given measurement time	
VZ0	Initial Z component of vehicle velocity	
VZC1	Navigated value of VZC before updating at a measurement time	
VZC2	Navigated value of VZC after updating at a measurement time	
VZC	Z component of vehicle velocity	(CØMMØN)
VZDØT	Z component of vehicle acceleration	(CØMMØN)
VZMEAN	Statistical mean in navigated errors in VZC	NUMER
VZR	Inertial Z component of Jupiter's velocity	(CØMMØN)
VZ(I)	I=1,100 Stored value of the true Z component of vehicle velocity at any given measurement time	
VZT1	True vehicle velocity component at any given measurement time	
W(I)	I=1,6 Integration error weighting factors	(CØMMØN)
X0	Initial X vehicle position	
XB	Intermediate variable; orbit-referenced X component of planetary position	PLANET

B. LINEAR OPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XC1	Navigated value of XC before updating at a measurement time	
XC2	Navigated value of XC after updating at a measurement time	
XC	X component of vehicle position	(CØMMØN)
XIPLAN	Floating point version of IPLAN	
XIP	Inclination of Jupiter's equator with respect to its orbital plane	
XJ2	Second zonal harmonic coefficient of Jupiter's gravitational field	(CØMMØN)
XJ4	Fourth zonal harmonic coefficient of Jupiter's gravitational field	(CØMMØN)
XKAN	The number of integration steps between any two measurement times	
XKØN	The number of integration steps between the initial time and the first measurement time	
XK(I,J)	I=1,2 J=1,2 Scratch pad matrix	
XLAB(I)	I=1,100 Table of planet index codes specifying the measurement schedule	
XL(I,J)	I=1,2 J=1,2 Scratch pad matrix	
XMEAN	The statistical mean of the navigated error in XC at any measurement time	NUMER

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XMNA10	Mean value of the measurement error in right ascension of the observed planet (Sun's frame) in arc sec	
XMNA1	XMNA10 converted to radians	
XMNA20	Mean value of the measurement error in right ascension of the observed planet (Jupiter's frame) in arc sec	
XMNA2	XMNA20 converted to radians	
XMND10	Mean value of the measurement error in declination of the observed planet (Sun's frame) in arc sec	
XMND1	XMND10 converted to radians	
XMND20	Mean value of the measurement error in declination of the observed planet (Jupiter's frame) in arc sec	
XMND2	XMND20 converted to radians	
XMNDA	Current mean of the measurement error in right ascension of the observed planet	
XMNDD	Current mean of the measurement error in the declination angle of the observed planet	
XMTHD	Integration switch-specifies R-K or Adams 4-point method	(CØMMØN)
XMU(I)	I=1, 10 Planetary gravitational constants	(CØMMØN)
XN1	Planet index code representing initial center of coordinates	
XNMAX	Total number of measurements per trajectory	

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE		DEFINITION	SUBROUTINE APPEARANCE
XNN(I)	I=1, 10	Planetary mean motion	(CØMMØN)
XNUM		Total number of trajectories Total number of navigated trajectories	NUMER
XPØM		Angle measured in Jupiter's orbital plane between the intersection of Jupiter's equatorial and orbit planes and the intersection of Jupiter's equatorial and the ecliptic plane	
XR(I)	I=1, 10	X component of planetary position	(CØMMØN)
XSUB1		A uniformly distributed random number (-1. < XSUB1 < 1.) associated with measurement errors in right ascension	
XSUB2		A uniformly distributed random number (-1. < XSUB2 < 1.) associated with measurement errors in declination	
XSUB3		A uniformly distributed random number (-1. < XSUB3 < 1.) associated with errors in knowledge of the initial X position component	
XSUB4		A uniformly distributed random number (-1. < XSUB4 < 1.) associated with errors in knowledge of the initial Y position component	
XSUB5		A uniformly distributed random number (-1. < XSUB5 < 1.) associated with errors in knowledge of the initial Z position component	
XSUB6		A uniformly distributed random number (-1. < XSUB6 < 1.) associated with errors in knowledge of the initial X velocity component	

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XSUB7	A uniformly distributed random number (-1. < XSUB7 < 1.) associated with errors in knowledge of the initial Y velocity component	
XSUB8	A uniformly distributed random number (-1. < XSUB8 < 1.) associated with errors in knowledge of the initial Z velocity component	
XT1	True X position of the vehicle at any measurement time (printout)	
XT(I)	I=1,100 True X position of the vehicle at any measurement time	
Y0	Initial Y component of vehicle position	
YB	Intermediate variable; orbit-referenced Y component of planetary position	PLANET
YC1	Navigated value of YC before updating at a measurement time	
YC2	Navigated value of YC after updating at a measurement time	
YC	Y component of vehicle position	(COMMON)
YIC(I)	I=1,10 Cosine of planetary inclination angle	(COMMON)
YIS(I)	I=1,10 Sine of planetary inclination angle	(COMMON)
YMEAN	The statistical mean of the navigated error in YC at any measurement time	NUMER
YNN	Floating point version of NN	DETERM

B. LINEAR ØPTIMUM FILTER PROGRAM (continued)

VARIABLE		DEFINITION	SUBROUTINE APPEARANCE
YP(I)	I=1,48	Integration table containing first derivatives	(CØMMØN)
YR(I)	I=1,10	Y component of planetary position	(CØMMØN)
Y(I)	I=1,48	Integration table containing the integrated variables	(CØMMØN)
YT1		True Y position of the vehicle at any measurement time (printout)	
YT(I)	I=1,100	True Y position of the vehicle at any measurement time	
Z0		Initial Z component of vehicle position	
ZC1		Navigated value of ZC before updating at a measurement time	
ZC2		Navigated value of ZC after updating at a measurement time	
ZCR2		The ratio $(ZC/R)^2$	ACCEL
ZC		Z component of vehicle position	(CØMMØN)
ZMEAN		The statistical mean of the navigated error in ZC at any measurement time	NUMER
ZR(I)	I=1,10	Z component of planetary position	(CØMMØN)
ZT1		True Z position of the vehicle at any measurement time (printout)	
ZT(I)	I=1,100	True Z position of the vehicle at any measurement time	

C. INTERPLANETARY SPACE NAVIGATIØN PROGRAM

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
A(I, J)	I=1, 3 J=1, 3 Transformation matrix from Jupiter equatorial frame to ecliptic frame	(CØMMØN)
A1 A2 A3	Scratch pad variables	TIØ1, NUMER1 TIØ TIØ
AA(I)	I=1, 10 Planetary semi-major axes	(CØMMØN)
AAX AAY AAZ AVX AVY AVZ	Statistical mean of errors in knowledge of initial position and velocity components	
ALFA0	True inertial right ascension of the observed planet	
ALFPR	Predicted right ascension of the vehicle-observed planet vector	
ARDØT	RDØT in m/s	
AR ATHE APSI	Spherical coordinate components of vehicle acceleration	(CØMMØN)
AXSP AYSP AZSP	Newtonian force contributions to the total gravitational acceleration components	(CØMMØN)
B(I, J)	I=1, 3 J=1, 3 Transformation matrix from the inertial frame to the orbital frame	(CØMMØN)
BINCPR	Inclination of Jupiter's equatorial plane with respect to the ecliptic	
C(I, J)	I=1, 3 J=1, 3 Transformation matrix from the pertinent inertial frame to an intermediate Cartesian, R-Θ-ψ oriented axis system	(CØMMØN)
CALF CDEL	Scratch pad variables	

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
CKMAU	Conversion factor from km to AU	
CMSAUD	Conversion factor from m/s to AU /day	
CØSI	Scratch pad variables	TIS, TIØ, INTSPH
CØSØ		TIS, TIØ, INTSPH
CPS2		NAVINT
CPSI	Cosine of the crossrange angle	TIS, NAVINT, INTSPH, SPHTIN
CRPP(I)	I=1, 10 Planetary mean longitude of perihelion	(CØMMØN)
D(I)	I=1, 7 Planetary distances from vehicle	TRUINT
DALFPR	Difference between measured and predicted right ascension of the vehicle-observed planet vector	
DALF	DELALF in arc sec	
DATAPR	Optional print switch - see Section III	
DATE	Julian date at TIME = 0.	
DDELPR	Difference between measured and predicted declination of the vehicle-observed planet vector	
DDEL	DELDEL in arc sec	
DEGRAD	Conversion factor from degrees to radians	
DEGSEC	Conversion factor from degrees	
DELALF	Measurement error in right ascension of the vehicle-observed planet vector	
DELDEL	Measurement error in declination of the vehicle-observed planet vector	
DELI	Distance from each of the celestial bodies from the center of coordinates	TRUINT

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
DELPR	Predicted declination of the vehicle-observed planet vector	
DELPSI	Estimated error in crossrange position	FILTER
DELT0	True inertial declination of the observed planet	
DELTAX DELTAY DELTAZ	Errors in knowledge of initial position in AU	
DELTHE	Estimated error in downrange position	FILTER
DELTS	Integration step size for state variables	
DELTS1	Temporary storage for the variable DELTS	
DELTX DELTY DELTZ	Statistical mean of errors in knowledge of initial position in km	
DELVX1 DELVY1 DELVZ1	Statistical mean of errors in knowledge of initial velocity in m/s	
DELVX0	DELVX1 converted to AU/day	
DELVY0	DELVY1 converted to AU/day	
DELVZ0	DELVZ1 converted to AU/day	
DELX DELY DELZ	Best estimate of errors in knowledge of initial position in AU	
DELXPR DELYPR DELZPR	Intermediate variables used in the calculation of the measurement quantities DELTHE AND DELPSI	
DELXX DELYY DELZZ	Initial X, Y and Z position errors transformed from the inertial to the orbital frame	
DEN	Scratch pad variable	

C. INTERPLANETARY SPACE NAVIGATIØN PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
DEPSI	Best estimate of initial error in crossrange position	
DETHE	Best estimate of initial error in downrange position	
DEX DEY DEZ	Best estimate of errors in knowledge of initial position in km	
DMDTC	Intermediate variable used in the integration of vehicle momentum	NAVINT
DPPR	Crossrange filter measurement result	(CØMMØN)
DPSI	DELPSI converted to degrees	
DT	Integration step size	(CØMMØN)
DTØ2	The ratio (DT/2)	
DTØ4	The ratio (DT/4)	
DTØ6	The ratio (DT/6)	
DTPR	Downrange filter measurement result	(CØMMØN)
DTR	Intermediate variable in navigation integration	NAVINT
DUMMY(I)	I=1, 250 Intermediate input table	(CØMMØN)
EE(I)	I=1, 10 Planetary orbit eccentricities	(CØMMØN)
EM	Momentum of vehicle	
EPS(I)	I=1, 10 Planetary mean longitude at epoch	(CØMMØN)
EPSI	Navigated error in PSI at the end of a measurement time	(CØMMØN)
ER	Navigated error in R at the end of a measurement time	(CØMMØN)
ETHE	Navigated error in THETA at the end of a measurement time	(CØMMØN)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
ERRI(1)	Navigated error in YI(1) at the end of a measurement time		(CØMMØN)
ERRI(2)	Navigated error in YI(2) at the end of a measurement time		(CØMMØN)
ERRI(3)	Navigated error in YI(3) at the end of a measurement time		(CØMMØN)
ERRIDT(1)	Navigated error in YIDT(1) at the end of a measurement time		(CØMMØN)
ERRIDT(2)	Navigated error in YIDT(2) at the end of a measurement time		(CØMMØN)
ERRIDT(3)	Navigated error in YIDT(3) at the end of a measurement time		(CØMMØN)
EVPSI	Navigated error in PSIDØT at the end of a measurement time		(CØMMØN)
EVR	Navigated error in RDØT at the end of a measurement time		(CØMMØN)
EVTHE	Navigated error in THEDØT at the end of a measurement time		(CØMMØN)
FK0(I)	I=1, 3	Intermediate variables used in Runge-Kutta integration of truth state equations	
FK1(I)	I=1, 3		
FK2(I)	I=1, 3		
FM0(I)	I=1, 3	Intermediate variables used in Runge-Kutta integration of truth state equations	
FM1(I)	I=1, 3		
FM2(I)	I=1, 3		
FM3(I)	I=1, 3		
FQ(I)	I=1, 7	Intermediate variables used in the Newtonian acceleration calculation	TRUINT
I	General index variable		RECTIF
IMAX	Maximum number of spherical force fields to be considered in navigation force model		(CØMMØN)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
IND	Unused variable	(CØMMØN)
INØ	Absolute value of NØ	
IP	General index variable	(CØMMØN)
IPP(I)	I=1,100 Table of planet index codes indicating the truth reference body at a given measurement time	
ISWT	Code used to indicate the time at which a new orbit reference frame has been defined	(CØMMØN)
ITM	The number of integer multiples of DT which have occurred since TIME=TBASE	
J	General index variable	MAIN, DPRINT, RECTIF
JJ	General index variable	TRUINT
K K1 K2	General index variables	DPRINT
KK(I)	I=1,7 Set of planet index codes in use at any given time	(CØMMØN)
L	General index variable	PLANET
L1	Switch specifying whether estimates of initial position error have been input in spherical coordinate downrange and cross-range components or equivalent Cartesian components	
LABEL(I)	I=1,10 List of planet names for printout purposes	
LL	General index variable	PLANET
LMAX	Total number of trajectories to be generated (including truth)	(CØMMØN)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
L0C	The location in the DUMMY table of the first data word on an input card	
LP	Index referencing a particular trajectory in a given run	(C0MM0N)
M	General index variable	DPRINT, PLANET, TRUINT
MTIME	Index referencing a particular measurement time in a trajectory	(C0MM0N)
N	General index variable	DPRINT
N1	Planet index code of dominant central body	RECTIF, TIS, PLANET, TRUINT
N2 N3 N4 N5 N6 N7	Planet index codes for celestial bodies other than the dominant one to be considered in the equations of motion	TRUINT, RECTIF
NMAX	Number of navigation measurements to be made during any one trajectory	
N0	Variable indicating the number of data words on an input card	
0MEGA	Orbital argument of nodes	TI0
0MEGAS	Instantaneous orbital angular velocity	(C0MM0N)
0MEPR	The angle measured in Jupiter's orbital plane between Jupiter's equatorial line of nodes and its orbital line of nodes	
0MM(I)	I=1, 10 Planetary argument of nodes	(C0MM0N)
0MMC(I)	I=1, 10 Cosine of the planetary argument of nodes	(C0MM0N)
0MMS(I)	I=1, 10 Sine of the planetary argument of nodes	(C0MM0N)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ØNE6	The ratio (1./6.)	
PB	Planetary semi-latus rectum	PLANET
PPRK0 PPRK1 PPRK2 PPRK3	Intermediate variables used in Runge-Kutta integration of DPPR	FILTER
PSDT	PSIDØT converted to degrees	
PSI	Crossrange angle	(CØMMØN)
PSIDEG	PSI converted to deg	
PSIDØT	First time derivative of PSI	(CØMMØN)
PSIK0 PSIK1 PSIK2 PSIK3	Intermediate variables used in Runge-Kutta integration of the crossrange angle PSI	
PSITRU(I)	I=1,100 True value of PSI at any given measurement time	
PSMEAN	Statistical mean of the navigated error in PSI at any measurement time	NUMER1
PSVMN	Statistical mean of the navigated error in PSIDØT at any measurement time	NUMER1
QI	Mathematical term in the Newtonian acceleration calculation	TRUINT
Q11 Q12 Q21 Q22 Q31 Q32	Elements of transformation matrix from a two-dimensional orbit-referenced frame to the inertial three-dimensional ecliptic frame	PLANET
R	Distance from vehicle to dominant central body	(CØMMØN)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
RAUD	R converted to km	
RDØT	First time derivative of R	(CØMMØN)
REQ(I)	I=1,10 Planetary equatorial radii	(CØMMØN)
RMEAN	Statistical mean of the navigated error in R at any measurement time	NUMER1
RR	The planetary heliocentric orbital radius	PLANET
RSPHJ	Radius of the sphere of influence of Jupiter	
RTRU(I)	I=1,100 The true value of R at any given measurement time	
RUN	Run identification number	
RVMN	The statistical mean of the navigated error in RDØT at any measurement time	NUMER1
RVP	Distance between the vehicle and the observed planet	
SALF SDEL	Scratch pad variables	
SECRAD	Conversion factor from arc sec to radians	
SGPSV SGRV SGTHV	Standard deviation of the errors in the navigated spherical coordinate derivatives (PSIDØT, RDØT, THEDØT) and the second moment about zero for the same data	NUMER1
SIGPSI SIGR SIGTHE	Standard deviation of the errors in the navigated spherical coordinates (PSI, R, THETA) and the second moment about zero for the same data	NUMER1
SIGMU(I)	I=1,6 Standard deviation of the errors in knowledge of initial position and velocity	

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
SIGVX SIGVY SIGVZ	Standard deviation of the errors in the navigated Cartesian velocity components (YIDT(I)) and the second moment about zero for the same data		NUMER1
SIGX SIGY SIGZ	Standard deviation of the errors in the navigated position components (YI(I)) and the second moment about zero for the same data		NUMER1
SINI SINØ	Scratch pad variables		INTSPH, TIS, TIØ
SMPSV2(I)	I=1, 100	Sum of the square of the navigated errors in PSIDØT at any given measurement time	NUMER1
SMRV2(I)	I=1, 100	Sum of the square of the navigated errors in RDØT at any given measurement time	NUMER1
SMTHV2(I)	I=1, 100	Sum of the square of the navigated errors in THEDØT at any given measurement time	NUMER1
SPSI	Sine of the crossrange angle		INTSPH, TIS
STDA10	Standard deviation of the applied measurement errors in right ascension of the vehicle-observed planet vector in arc sec (coordinate center = Jupiter)		
STDA1	STDA10 converted to radians		
STDA20	Standard deviation of the applied measurement errors in right ascension of the vehicle-observed planet vector in arc sec (coordinate center = Sun)		
STDA2	STDA20 converted to radians		
STDD10	Standard deviation of the applied measurement errors in declination of the vehicle-observed planet vector in arc sec (coordinate center = Jupiter)		

C. INTERPLANETARY SPACE NAVIGATION PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
STDD1	STDD10 converted to radians		
STDD20	Standard deviation of the applied errors in declination of the vehicle-observed planet vector in arc sec (coordinate center = Sun)		
STDD2	STDD20 converted to radians		
STHE	Sine of the downrange angle		INTSPH, TIS
SUM1 SUM2 SUM3	Intermediate variables used in the calculation of the gravitational force acceleration components		TRUINT
SUMPS2(I)	I=1, 100	Sum of the square of the navigated errors in PSI at any given measurement time	NUMER1
SUMPSI(I)	I=1, 100	Sum of the navigated errors in PSI at any given measurement time	NUMER1
SUMPSV(I)	I=1, 100	Sum of the navigated errors in PSIDOT at any given measurement time	NUMER1
SUMR(I)	I=1, 100	Sum of the navigated errors in R at any given measurement time	NUMER1
SUMR2(I)	I=1, 100	Sum of the square of the navigated errors in R at any given measurement time	NUMER1
SUMRV(I)	I=1, 100	Sum of the navigated errors in RDOT at any given measurement time	NUMER1
SUMTH2(I)	I=1, 100	Sum of the square of the navigated errors in THETA at any given measurement time	NUMER1
SUMTHE(I)	I=1, 100	Sum of the navigated errors in THETA at any given measurement time	NUMER1

C. INTERPLANETARY SPACE NAVIGATI0N PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
SUMTHV(I)	I=1, 100	Sum of the navigated errors in THED0T at any given measurement time	NUMER1
SUMVX(I)	I=1, 100	Sum of the navigated errors in YIDT(1) at any given measurement time	NUMER1
SUMVX2(I)	I=1, 100	Sum of the square of the navigated errors in YIDT(1) at any given measurement time	NUMER1
SUMVY(I)	I=1, 100	Sum of the navigated errors in YIDT(2) at any given measurement time	NUMER1
SUMVY2(I)	I=1, 100	Sum of the square of the navigated errors in YIDT(2) at any given measurement time	NUMER1
SUMVZ(I)	I=1, 100	Sum of the navigated errors in YIDT(3) at any given measurement time	NUMER1
SUMVZ2(I)	I=1, 100	Sum of the square of the navigated errors in YIDT(3) at any given measurement time	NUMER1
SUMX(I)	I=1, 100	Sum of the navigated errors in YI(1) at any given measurement time	NUMER1
SUMX2(I)	I=1, 100	Sum of the square of the navigated errors in YI(1) at any given measurement time	NUMER1
SUMY(I)	I=1, 100	Sum of the navigated errors in YI(2) at any given measurement time	NUMER1
SUMY2(I)	I=1, 100	Sum of the square of the navigated errors in YI(2) at any given measurement time	NUMER1

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
SUMZ(I)	I=1,100	Sum of the navigated errors in YI(3) at any given measurement time	NUMER1
SUMZ2(I)	I=1,100	Sum of the square of the navigated errors in YI(3) at any given measurement time	NUMER1
T(I)	I=1,100	Table specifying the time at which navigation measurements should be made	
T0	Initial time measured from DATE		
T1 T2	Intermediate variables used in the calculation of the vehicle acceleration components		TRUINT
TABLE(I)	I=1,100	Table of planet index codes signifying which planet is being observed in a particular navigation measurement	
TAU1 TAU2	ISN filter time constants		FILTER
TBASE	Time at which integration step size was last changed		
TEMP	Scratch pad variables		TRUINT, PLANET
TEMP1			FILTER, TRUINT, INTSPH, PLANET
TEMP2			FILTER, TRUINT, INTSPH, PLANET
TEMP3			
TEMPC			PLANET
TEMPS			PLANET
TEST	Indicator used to specify a two-body case - see Section III		TRUINT, TIS
THEDØT	First time derivative of THETA		(CØMMØN)

C. INTERPLANETARY SPACE NAVIGATIØN PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
THEK0 THEK1 THEK2 THEK3	Intermediate variables used in the Runge-Kutta integration of the downrange angle THETA	
THETA	Downrange angle	(CØMMØN)
THETRU(I)	I=1, 100 The true value of THETA at any given measurement time	
THMEAN	Statistical mean of navigated errors in THETA at any measurement time	NUMER1
THT	THETA converted to degrees	
THTDØT	THEDØT converted to degrees	
THVMN	Statistical mean of navigated errors in THEDØT at any measurement time	NUMER1
TIME	Trajectory time measured from DATE	(CØMMØN)
TM	Temporary value of EM	(CØMMØN)
TM1	Number of days from epoch to DATE	(CØMMØN)
TME	Time measured from epoch	PLANET
TP	Temporary value of PSI	(CØMMØN)
TPPR	Temporary value of DPPR	FILTER
TPRK0 TPRK1 TPRK2 TPRK3	Intermediate variables used in the Runge-Kutta integration of DTPR	FILTER
TR	Temporary value of R	
TRUDT(I, J)	I=1, 3 J=1, 100 Stored values of the vehicle's true Cartesian velocity components at any given measurement time	

C. INTERPLANETARY SPACE NAVIGATIØN PROGRAM (continued)

VARIABLE	DEFINITION		SUBROUTINE APPEARANCE
TRUE(I, J)	I=1,3 J=1,100	Stored values of the vehicle's true Cartesian position components at any given measurement time	
TTHE	Temporary values of THETA		(CØMMØN)
TTIME	Time between navigation measurements		
TTPR	Temporary value of DTPR		FILTER
TVP	Temporary value of PSIDØT		(CØMMØN)
TVR	Temporary value of RDØT		(CØMMØN)
TVT	Temporary value of THEDØT		(CØMMØN)
TWØPI	2π		
TYI(I)	I=1,3	Temporary values of YI(I) I=1,3	
TYIDT(I)	I=1,3	Temporary values of YIDT(I) I=1,3	
VPK0 VPK1 VPK2 VPK3	Intermediate variables used in the Runge-Kutta integration of PSIDØT		
VPSTRU(I)	I=1,100	The true value of PSIDØT at any given measurement time	
VRK0 VRK1 VRK2 VRK3	Intermediate variables used in the Runge-Kutta integration of RDØT		
VTHTRU(I)	I=1,100	True value of THEDØT at any given measurement time	
VTK0 VTK1 VTK2 VTK3	Intermediate variables used in the Runge-Kutta integration of THEDØT		

C. INTERPLANETARY SPACE NAVIGATION PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
VV(I)	I=1,3 Intermediate variables used in calculating the rectified velocity components	RECTIF
VX0 VY0 VZ0	Initial X, Y and Z components of velocity in AU/day	
VXB VYB	Intermediate variables; orbit-referenced X and Y components of Jupiter's velocity	PLANET
VXDOT VYDOT VZDOT	Inertial Cartesian components of vehicle acceleration	(COMMON)
VXMEAN	Statistical mean of navigated errors in YIDT(1) at any measurement time	NUMER1
VYMEAN	Statistical mean of navigated errors in YIDT(2) at any measurement time	NUMER1
VZMEAN	Statistical mean of navigated errors in YIDT(3) at any measurement time	NUMER1
VYPR VZPR	Intermediate variables used in calculating inertial velocity components	SPHTIN
VXR VYR VZR	Inertial X, Y and Z components of Jupiter's velocity	(COMMON)
VXX VYY VZZ	Initial true vehicle Cartesian velocity components in m/s	
X	YI(1) converted to km	
X(I)	I=1,250 Intermediate values of DUMMY variables for the purpose of printing out the input	DPRINT
X0	Initial X vehicle position in AU	

C. INTERPLANETARY SPACE NAVIGATIØN PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XB	Intermediate variable; orbit-referenced X component of planetary position	PLANET
XDØT	YIDT(1) converted to m/s	
XIMAX	Floating point version of IMAX	
XJ2	Second zonal harmonic coefficient of Jupiter's gravitational field	(CØMMØN)
XJ4	Fourth zonal harmonic coefficient of Jupiter's gravitational field	(CØMMØN)
XKAN	The number of integration steps between any two measurement times	
XKØN	The number of integration steps between the initial time and the first measurement time	
XKM	Navigation gain, momentum feedback	NAVINT
XKPSI	Navigation gain, $\dot{\psi}$ feedback	NAVINT
XKI	Navigation gain, $\dot{V}_{\Theta}$ feedback	NAVINT
XKTHE	Navigation gain, $\dot{\Theta}$ feedback	NAVINT
XKVPSI	Navigation gain, V_{ψ} feedback	NAVINT
XKVTHE	Navigation gain, V_{Θ} feedback	NAVINT
XL1	Floating point version of L1	
XLMAX	Floating point version of LMAX	
XMEAN	The statistical mean of the navigated error in YI(1) at any measurement time	NUMER1
XMK0 XMK1 XMK2 XMK3	Intermediate variables used in the Runge-Kutta integration of EM	

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XMDD20	Mean value of the measurement error in declination of the observed planet (Sun's frame) in arc sec	
XMNA10	Mean value of the measurement error in right ascension of the observed planet (Jupiter's frame) in arc sec	
XMNA1	XMNA10 converted to radians	
XMNA2	XMNA20 converted to radians	
XMNA20	Mean value of the measurement error in right ascension of the observed planet (Sun's frame) in arc sec	
XMND10	Mean value of the measurement error in declination of the observed planet (Jupiter's frame) in arc sec	
XMND1	XMND10 converted to radians	
XMND2	XMDD20 converted to radians	
XMU(I)	I=1,10 Planetary gravitational constants	(CØMMØN)
XN1	Planetary index code representing initial center of coordinates	
XNMAX	Floating point version of NMAX	
XNN(I)	I=1,10 Planetary mean motion	(CØMMØN)
XNUM	Total number of navigated trajectories	NUMER1
XPR	The orbital frame X component of navigated position	(CØMMØN)
XR(I)	I=1,10 X component of planetary position	(CØMMØN)
XSUB1	A uniformly distributed random number $(-1. < XSUB2 \leq 1.)$ associated with measurement errors in right ascension	

C. INTERPLANETARY SPACE NAVIGATION PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
XSUB2	A uniformly distributed random number ($-1. <$ XSUB2$\leq 1.$) associated with measurement errors in declination	
XSUB3	A uniformly distributed random number ($-1. <$ XSUB3$< 1.$) associated with errors in knowledge of the initial X position component	
XSUB4	A uniformly distributed random number ($-1. <$ XSUB4$< 1.$) associated with errors in knowledge of the initial Y position component	
XSUB5	A uniformly distributed random number ($-1. <$ XSUB5$< 1.$) associated with errors in knowledge of the initial Z position component	
XSUB6	A uniformly distributed random number ($-1. <$ XSUB6$< 1.$) associated with errors in knowledge of the initial X velocity component	
XSUB7	A uniformly distributed random number ($-1. <$ XSUB7 $< 1.$) associated with errors in knowledge of the initial Y velocity component	
XSUB8	A uniformly distributed random number ($-1. <$ XSUB8$< 1.$) associated with errors in knowledge of the initial Z velocity component	
XX	Initial X component of vehicle position in km or AU	
XX(I)	I=1, 3 Intermediate variable used in the calculation of the rectified position components	RECTIF
XXPR	Inertial X component of the navigated vehicle position	
Y	YI(2) converted to km	
Y(I)	I=1, 3 The vehicle position components as used in the truth integration scheme	TRUINT

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
Y0	Initial Y component of vehicle position in AU	
YB	Intermediate variable; orbit-referenced Y component of planetary position	PLANET
YDØT	YIDT(2) converted to m/s	
YIC(I)	I=1, 10 Cosine of planetary inclination angle	(CØMMØN)
YI(1)	X component of vehicle position	
YI(2)	Y component of vehicle position	(CØMMØN)
YI(3)	Z component of vehicle position	
YIDT(1)	X component of vehicle velocity	
YIDT(2)	Y component of vehicle velocity	(CØMMØN)
YIDT(3)	Z component of vehicle velocity	
YIS(I)	I=1, 10 Sine of planetary inclination angle	(CØMMØN)
YMEAN	The statistical mean of the navigated error in YI(2) at any measurement time	NUMER1
YPR	The orbital Y component of navigated position	(CØMMØN)
YR(I)	I=1, 10 Y component of planetary position	(CØMMØN)
YY	Initial Y component of vehicle position in km or AU	
YYPR	Inertial Y component of navigated vehicle position	
Z	YI(3) converted to km	
Z0	Initial Z component of vehicle position in AU	

C. INTERPLANETARY SPACE NAVIGATI0N PROGRAM (continued)

VARIABLE	DEFINITION	SUBROUTINE APPEARANCE
ZD0T	YIDT(3) converted to m/s	
ZMEAN	The statistical mean of the navigated error in YI(3) at any measurement time	NUMER1
ZPR	The orbital Z component of navigated position	(C0MM0N)
ZR(I)	I=1, 10 Z component of planetary position	(C0MM0N)
ZZ	Initial Z component of vehicle position in km or AU	
ZZPR	The inertial Z component of navigated vehicle position	

IX. REFERENCES

1. The American Ephemeris and Nautical Almanac For the Year 1966, U. S. Government Printing Office, 1964.

2. Ehricke, K.A., Space Flight; Vol. I, "Environment and Celestial Mechanics", D. Van Nostrand Company, Inc., Princeton, New Jersey; Toronto; New York; London, 1960. pgs. 118-127.

3. Jensen, J., Townsend, G.E., Kraft, J.D., Kork, J., Design Guide To Orbital Flight, McGraw Hill Book Company Inc., New York, Toronto, London, 1962, ch. III.

4. Wolverton, R.W., Flight Performance Handbook For Orbital Operations, John Wiley and Sons, Inc., New York and London, 1963, Appendix B.

5. Mechtly, E.A., The International System of Units, "Physical Constants and Conversion Factors", NASA SP-7012, National Aeronautics and Space Administration, Washington, D.C., 1964.

6. Brown, R.C., Brulle, R.V., Griffin, G.D., Six-Degree-of-Freedom Flight Path Study Generalized Computer Program, "Part I - Problem Formulation", McDonnel Aircraft Corporation, St. Louis, Missouri, May 1961, Pg. 215.

7. Hildebrand, F.B., Introduction to Numerical Analysis, McGraw-Hill Book Company, Inc., New York, Toronto, London, 1956.

8. Stern, R.G., Interplanetary Midcourse Guidance Analysis, Vol. II Appendices, Doctor of Science Thesis at the Massachusetts Institute of Technology, Cambridge, Massachusetts.

9. Faddeeva, V.N., Computational Methods of Linear Algebra, Dover Publications, Inc., New York 1959, Pgs. 72-75.