DOUGLAS REPORT SM-49250

NUMERICAL RESIDUAL PERTURBATION SOLUTION APPLIED TO AN EARTH SATELLITE INCLUDING LUNI-SOLAR EFFECTS

SEPTEMBER 1965

Prepared under Contract No. NASw-901
by Douglas Aircraft Company, Inc.
Missile and Space Systems Division
Santa Monica, California
for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

DOUGLAS REPORT SM-49250

NUMERICAL RESIDUAL PERTURBATION SOLUTION APPLIED TO AN EARTH SATELLITE INCLUDING LUNI-SOLAR EFFECTS

SEPTEMBER 1965

By
J.T. Martin
M.C. Eckstein

Prepared under Contract No. NASw-901
by Douglas Aircraft Company, Inc.
Missile and Space Systems Division
Santa Monica, California
for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

The purpose of this report is to demonstrate a new method of numerical residual perturbation solution as applied to the problem of an earth satellite including luni-solar effects. Cowell demonstrated a method of numerically solving the total differential equations of motion of an orbiting object. The variation of parameters and Encke's methods take advantage of the known analytic solution to the two-body problem and numerically handle only the perturbations to the orbit. This report demonstrates the use of an analytic series perturbation solution of the oblateness problem as a reference orbit (rather then using conics as a reference) with numerical solution of the residual perturbation equations of motion including neglected higher order effects as well as perturbations not included in the analytic model. Results obtained from this demonstration program were compared with both single precision and double precision Cowell programs, and showed significant accuracy improvements over the single precision program as well as reducing computing time by a factor of four over the double precision program. Further refinements were suggested in order to obtain the maximum benefit from the technique for a production program. This work was supported by contract NASw-901.

CONTENTS

Section 1

INTRODUCTION

This program was developed primarily as a research program to investigate the practicality of a generalized Encke-type solution to the motion of an earth satellite, in satisfaction of portions of research contract NASw-901. The program uses an approximate analytic solution of the oblateness problem (ref. 1) for a reference orbit, and numerically integrates using the Runge-Kutta method to find the contribution of the neglected higher-order analytic terms as well as other perturbations not included in the analytical model. The analytical model considers only the perturbations of the second and fourth zonal harmonics of the potential. The program is designed to consider additional zonal, tesseral, and sectorial harmonics up to and including the coefficients C_{66} and S_{66}, and to also consider lunar and solar perturbations if desired.

Section 2

SYMBOLS

NOTE: In the definition of symbols, the numbers in parentheses represent the numbers assigned to equations throughout the rest of the report; the names or letters in parentheses represent the titles of specific subroutines.

A — Symbol for $\frac{p_a}{\cos i_{oo}}$, initial total angular momentum (non-dimensional), or array of dimension 3 in subroutine EXPERT to store the sum of the sun and moon accelerations (non-dimensional) or FORTRAN floating point variable for L = N-1 in subroutine GPOT

A_1 — $\frac{p}{\cos i}$ = total angular momentum at any time (non-dimensional), (FORTRAN symbol A1)

a_f — Acceleration in local geocentric south direction (non-dimensional with respect to g),(FORTRAN symbol AF)

AA — $\frac{\cos 2w - \cos 2\omega}{\bar{S}_o + \text{sign}(\bar{S}_o)\sqrt{\bar{\kappa}_o - \kappa_1 \cos 2\omega}}$, (APSOL)

AB — $\frac{\cos^8 i_{oo}}{2p^8}$, (CONST)

a_g — Acceleration in local geocentric east direction (non-dimensional with respect to g),(FORTRAN symbol AG)

a_h — Acceleration outward along the local geocentric vertical (non-dimensional with respect to g),(FORTRAN symbol AH)

a_r	Accelerations in the outward radial direction not considered in original analytical model (non-dimensional) (96)
AC	$\dfrac{\cos^5 i_{oo}}{p^4}$, (CONST)
ACC	$\dfrac{5 \cos i_{oo}}{A^4 S_1'}$, (CONST)
ACS	$\dfrac{\cos^5 i_{oo} \sin i_{oo}}{2p^5}$, (CONST)
ACSS	$\dfrac{-\varepsilon \cos^5 i_{oo} \sin i_{oo}}{2p^4}$, (CONST)
ACS2	$\dfrac{\cos^5 i_{oo} \sin i_{oo}}{2p^4}$, (CONST)
ACS32	$\dfrac{5^{3/2} \cos^4 i_{oo} \sin i_{oo}}{p^4}$, (CONST)
AC32	$\dfrac{\varepsilon^{3/2} \cos^5 i_{oo}}{p^5}$, (CONST)
AD	Array of derivatives of the approximate solutions in subroutine APSOL. $AD(1) = \dfrac{dp_a}{d\phi} = 0$, $AD(2) = \dfrac{d\Omega_a}{d\phi}$, $AD(3) = \dfrac{di_a}{d\phi}$, $AD(4) = \dfrac{dq_a}{d\phi}$, $AD(5) = H$, $AD(6) = \dfrac{dt_a}{d\phi}$
AF	a_f
AF1	a_{f_1}
AF2	a_{f_2}

AG	a_g
AG1	a_{g_1}
AG2	a_{g_2}
AH	a_h
AH1	a_{h_1}
AH2	a_{h_2}
AJ2	The J_2 zonal coefficient of the potential, (MAIN)
AJ4	The J_4 zonal coefficient of the potential, (MAIN)
AMP	$\sqrt{\frac{\bar{\kappa}_o-\kappa_1}{\bar{\kappa}_o+\kappa_1}}$ or $\sqrt{\frac{\kappa_1-\bar{\kappa}_o}{\bar{\kappa}_o+\kappa_1}}$ depending on perigee case number IC, (CONST)
AM4	$\frac{\cos^4 i_{oo}}{p^4}$, (CONST)
ANG	$\sqrt{2\kappa_1}\,(\bar{\phi}_T-\bar{\phi}_o)$ or $\sqrt{2\kappa_1}\,\phi_T$, auxiliary angle to find ω ϕ_T = total ϕ (not modded), (APSOL)
ANG2	$\frac{\phi_i-w}{2}$ = angle to determine quadrant of TANG in subroutine CONST (rad) or $= \frac{\phi-\omega}{2} =$ angle to determine quadrant of TANG in subroutine APSOL
AOR	Array of $\frac{J_n}{\underline{r}^{n+2}}$ in subroutine GPOT. AOR(9) maximum

AS	Array of approximate solutions in subroutine APSOL. AS(1) = p_a, AS(2) = Ω_a, AS(3) = i_a, AS(4) = q_a, AS(5) = u_a, AS(6) = t_a
AU2	$\frac{A_1}{u^2}$, (ENCKE)
A1	$A_1 = \frac{p}{\cos i}$, (ENCKE)
A3	$\frac{p^3}{\cos^3 i_{oo}}$, (CONST)
A3E	$\frac{p^3}{\cos^3 i_{oo} (1-e_o^2)}$, (CONST)
A6	$\frac{1}{2A^6}$, (CONST)
b	Angle measured from ascending node to satellite's meridian along the equator (rad), (104). FORTRAN symbol B, (EXPERT)
B	b, (EXPERT)
B	M, (GPOT)
BEW	Integer multiples of the longitude, (GPOT)
B2E2	$2\epsilon^2 B_2^*$, (CONST)
B2E	$\epsilon^2 B_2^*$, (CONST)
B2S	B_2^*, (21)

B2SP	$B_2^{*'}$, (20)
C	c = ratio of $\frac{D}{J^2}$ where D and J are coefficients of the second and first zonals of the potential, respectively. $c \approx 4/7$
C	Two-dimensional array (6 x 6) for the C_{nm}, (GPOT)
C_{nm}	Coefficients for computation of tesserals and sectorials of the earth's potential, used in subroutine GPOT
CA	Array of reduced moduli k obtained by decreasing Landen transformation in subprogram ELIPE
CAP	Array of reduced modified moduli k' obtained by descending Landen transformation in subprogram ELIPE $k' = \sqrt{1-k^2}$
CAS	k_{ir} • SNVE, (ELIF)
CAℓ	k = dummy variable for the modulus, (ELIPE)
CB	cos b, (EXPERT)
CBE	Array of cos(n • EW) in subroutine GPOT. CBE(6) maximum
CC	Two-dimensional array (6 x 6) of coefficients used in computation of complete potential, (GPOT)
CC7	$(3-7 \cos^2\theta)$, (ENCKE)
CHI1	Intermediate angle needed to find CHI1S

CHI1S	χ_1^* = angle used to find $\overline{\phi}_o$ in subroutine CONST, case 1
CHI2	Intermediate angle to find CHI2S
CHI2S	χ_2 = angle used to find $\overline{\phi}_o$ in subroutine CONST, case 3
CI	cos i in subroutine ENCKE, or cos i_{oo} in subroutine CONST
CI2	$\cos^2 i$ in subroutine ENCKE, or $\cos^2 i_{oo}$ in subroutine CONST
CI3	$\cos^3 i$ in subroutine ENCKE
CI31	$1-3\cos^2 i_{oo}$ in subroutine CONST
CI4	$\cos^4 i$ in subroutine ENCKE, or $\cos^4 i_{oo}$ in subroutine CONST
CIOC	cos i_{oc}, (APSOL)
CMAX	Maximum value of the absolute value of the Runge-Kutta increments over two complete steps.
CMC	$\cos 2w - \cos 2\omega$, (APSOL)
CN	Elliptic function cn, (APSOL)
COEFF	Array of coefficients for the potential. Contains $J_0 \rightarrow J_8$, $C_{1,1} \rightarrow C_{6,6}$, $S_{1,1} \rightarrow S_{6,6}$, N1 and N2
COSP	cos(PHIS(I-1)), (ELI)
CP	$\cos \phi$
CPMW	$\cos(\phi_i - w)$, (CONST) $\cos(\phi - \omega)$, (APSOL)

CPPW	$\cos(\phi+\omega)$, (APSOL)
CQ	$\cos q$, (EXPERT)
CRD	"Critical divisor term" $1-5\cos^2 i_{oo}$, (CONST)
CS	$\cos i_{oo} \sin i_{oo}$, (CONST), or two-dimensional array (6 x 6) of coefficients in computation of complete potential, (GPOT)
CT	$\cos\theta$
CT2	$\cos^2\theta$, (ENCKE)
CVE	cos (VEO), (ELIF)
CW	$\cos w$, (CONST)
CXW	$\cos\omega$
C2IOC	$\cos^2 i_{oc}$, (APSOL)
C2P	$\cos 2\phi$
C2S	C_2^*, (23)
C2SP	$C_2^{*\prime}$, (22)
C2E	$\varepsilon^2 C_2^*$,(CONST)
C2PMW	$\cos^2(\phi-\omega)$
C2T	$\cos 2\theta$, (ENCKE)

C2W	$\cos 2w^*$, (CONST)
C2XW	$\cos 2\omega$
C22E	$\frac{2\epsilon^2 C_2^*}{p^2}$, (CONST)
C3PMW	$\cos(3\phi-\omega)$
C4E	$\epsilon^4 (C_2^*)^2$, (CONST)
C4PMW	$\cos(4\phi-2\omega)$
D	N, (GPOT)
DAFDPH	$\frac{da_f}{d\phi}$, (ENCKE)
DAGDPH	$\frac{da_g}{d\phi}$, (ENCKE)
DA1PHI	$\frac{dA_1}{d\phi}$, (ENCKE)
DEA	$\frac{de_a}{d\phi}$, (APSOL)
DELI	Angle representing the total even number of revolutions of PHILT in radians. (ELI)
DELPHI	Original guess at computing interval, input in degrees used internally in radians. (MAIN)
DELPHO	Angle used to find $\Delta\phi$ in ELI
DELS	Modded change in ϕ, (ELI)
DELU	Δu used to approximate $\frac{d}{d\phi}(\frac{d\phi}{dt})$, (ENCKE)

DENK	Array of three coefficients used to compute the total energy
DENOM	$p^2u^2\sin^2 i \sin\theta + \cos^4 i \cos\theta F$, (ENCKE)
DFDPHI	$\frac{dF}{d\phi}$, (ENCKE)
DIDPHI	$\frac{di}{d\phi}$, (ENCKE)
DM1	D-1, (GPOT)
DODPHI	$\frac{d\Omega}{d\phi}$, (ENCKE)
DOMEO	$\frac{1}{\varepsilon^{1/2}} \frac{d\Omega_{oo}}{d\phi}$, (APSOL)
DOME12	$\frac{d\Omega_o^{1/2}}{d\phi}$, (APSOL)
DOME32	$\frac{d\Omega^{3/2}}{d\phi}$, (APSOL)
DP	Array of coefficients in subroutine GPOT, DP(10)
DPDPHI	$\frac{dp}{d\phi}$, (ENCKE)
DPH	Test ratio to determine when the limit of $\frac{\phi_n}{2^n}$ has been reached (ELI)
DPHI	$\Delta\phi$ used to find approximation for $\frac{d}{d\phi}(\frac{d\phi}{dt})$, (ENCKE)
DPHIDT	$\frac{d\phi}{dt}$, (ENCKE)
DQO	$\frac{dq_o}{d\phi}$, (APSOL)

DQ1	$\frac{\epsilon dq_1}{d\phi}$, (APSOL)
DSIOC	$\frac{di_{oc}}{d\phi} \sin i_{oc}$, (APSOL)
DT	Change in time since entering subroutine EXPERT (initially = 0). Also used as the step size in subroutine RKTOM
DTA	$\frac{dt_a}{d\phi}$, (APSOL)
DTDPHI	$\frac{dt}{d\phi}$, (ENCKE)
DTHETP	$\frac{d\theta}{d\phi}$, (ENCKE)
DTIP2	$n\rho_1$, (GPOT)
DTI21	$(2n-1)\rho_1$, (GPOT)
DTM	Multiplicative input parameter to increase the step size if the estimated computing error is too small
DTSAVE	Saved value of time to compute change in time since entering subroutine ENCKE, (MAIN)
DT2	Half the step-size in subroutine RKTOM
DT3	Step-size over 3 in subroutine RKTOM
DUDSI	$\frac{\partial U}{\partial \psi}$, (ENCKE)
DU1	$\frac{du_1}{d\phi}$, (APSOL)

DVT	Array of 6 which represents the sum of the approximate numerical values of the dependent variables at any time. (MAIN)
DW	$\frac{d\omega}{d\phi} = \omega'$, (APSOL)
DWB	$\frac{d\omega}{d\bar{\phi}}$, (APSOL)
D2EA	$\frac{d^2 e_a}{d\phi^2}$, (APSOL)
D2W	$\frac{d^2\omega}{d\phi^2}$, (APSOL)
E	e_o or e_o^* in subroutine CONST, or dummy output array of (6) giving evaluation of the derivatives for numerical integration (ENCKE) $E(1) = \frac{dp_n}{d\phi}$, $E(2) = \frac{d\Omega_n}{d\phi}$, $E(3) = \frac{di_n}{d\phi}$, $E(4) = \frac{dq_n}{d\phi}$, $E(5) = \frac{du_n}{d\phi}$, $E(6) = \frac{dt_n}{d\phi}$
EA	e_a, (APSOL)
EALL	Allowable error computed in subroutine RKTOM
EE	Dummy name for the array of 6 stored in labeled common /ENERG/. Used in main program to obtain quantities to compute the total energy
EF	$\frac{\sqrt{1-e_o^2}}{1+e_o}$, (CONST)
EM	k_n^2 where k_n is the last reduced modulus, (ELIF)

EMAX	Input value to program and to subroutine RKTOM, which is a measure of the maximum allowable accuracy desired
EMIN	Input value to program and to subroutine RKTOM, which is a measure of the minimum allowable accuracy desired
EM2	$(1-e_o^2)$, (CONST)
EM212	$\frac{2}{\sqrt{1-e_o^2}}$, (CONST)
EM22	$\sqrt{1-e_o^2}$, (CONST)
ENK	Array of 6 which is $\frac{dp_n}{d\phi}$, $\frac{d\Omega_n}{d\phi}$, $\frac{di_n}{d\phi}$, $\frac{dq_n}{d\phi}$, $\frac{du_n}{d\phi}$, $\frac{dt_n}{d\phi}$
EO2	$\frac{e_o}{2}$, (CONST)
EO6	$\frac{e_o}{6}$, (CONST)
EO3	$\frac{e_o}{3}$, (CONST)
EPD2IO	$\varepsilon^{1/2} \frac{d^2 i^*_{o1/2}}{d\phi^2}$, (APSOL)
EPS	$\varepsilon = J$ = non-dimensional coefficient of the second zonal harmonic of the potential $\approx 1.623 \times 10^{-3}$
EPS12	$\varepsilon^{1/2}$, (CONST)
EPS2	ε^2, (CONST)

EPS3	ε^3, (CONST)
EPS32	$\varepsilon^{3/2}$, (CONST)
ERMIN	Minimum allowable error computed in subroutine RKTOM
EROT	Input rotation rate of the earth in rad/hour, but used internally as a non-dimensional rate. (EXPERT)
ESTER	Estimated computing error in subroutine RKTOM
ES12	$e^*_{1/2}$, (APSOL)
EW	East earth longitude of the satellite, (EXPERT)
EWOG	Longitude of Greenwich measured from 1950.0 equinox at any time. $0 \leq \text{EWOG} \leq 2\pi$. Initially input as the value at t_i
E12	$E_{1/2}$, (A.23)
E2	e_o^2, (CONST)
E2C	$e_o^2 c$, (CONST)
E3K	$\varepsilon^3 \kappa_1$, (CONST)
F	Symbol for the term $\frac{\partial U}{\partial \phi} + \tan i \frac{\cos \phi}{\sin \phi} \frac{\partial U}{\partial \psi}$ (non-dimensional) same in FORTRAN, (ENCKE)
FDT	Multiplying input factor used in selecting the optimum computing interval

GAM1	Symbol for γ_1 = constant for cases 2 and 3 eccentricity calculations, (CONST)
GAPOB	$\bar{\kappa}_o$, (CONST) (27)
GAP1	κ_1, (CONST) (29)
GAP1P	κ_1', (CONST) (28)
GM	Array of 2 where GM(1) = GM_{sun} $(\frac{km^3}{sec^2})$, GM(2) = $GM_{moon}(\frac{km^3}{sec^2})$, (EXPERT)
GOK	$\frac{\bar{\kappa}_o}{\kappa_1}$, (CONST)
GP(I,J)	Two-dimensional array storing perturbative accelerations of the sun and moon. I = 1, 2, 3,; J = 1 (sun), J = 2 (moon) (km^3/sec^2), (EXPERT)
GO	$-(1 - 3\cos^2 i_{oo}) - \frac{e_o^2}{2}(1 - 5\cos^2 i_{oo})$, coefficient in u_1, (CONST)
G1	$\frac{e_o^2}{4}(1 - 3\cos^2 i_{oo})$, coefficient in u_1, (CONST)
G2	$-(\frac{\sin^2 i_{oo}}{3} - \frac{e_o^2}{3} + \frac{5}{6} e_o^2 \sin^2 i_{oo})$, coefficient in u_1, (CONST)
G3	$\frac{e_o^2}{6}(1 - 9\cos^2 i_{oo})$, coefficient in u_1, (CONST)
G4	$-\frac{e_o}{12}(5-11\cos^2 i_{oo})$, coefficient in u_1, (CONST)

G5	$-\frac{e_o^2}{12}(1-3\cos^2 i_{oo})$, coefficient in u_1, (CONST)
H	All small terms in $\frac{dq_o}{d\phi}$ (non-dimensional), (81) and (82), (FORTRAN symbol = H)
H	FORTRAN symbol for theoretical H, (APSOL)
HAH	Array of dependent variables and their derivatives HAH(12). (MAIN)
HS	Array of six values of the Simpson's rule increments over two complete computing intervals
i	Inclination (rad)
IC	Flag which gives the case number for perigee calculation IC = 1, 2, or 3, (CONST)
IE	Flag to determine case number for $e^*_{1/2}$ calculations IE = 1, 2, or 3, (CONST)
IMI1	Counter for I-1, (ELI)
IMI	I-1, (ELIPE)
IP	Initial point flag = 1 for first point, = 2 thereafter, (MAIN)
IPRINT	Print flag - calculations for print only and printing are done if IPRINT = 1; if IPRINT = 2, this is suppressed. (MAIN)
IR	NOT2 - I, used to determine last k value to be used in computing the elliptic function in subprogram ELIF

IW — Quadrant of the angle W in subprogram QUAD1. IW = 1, 2, 3 or 4

IWC — Flag which tells if ω = constant in case 2 perigee calculation. If IWC = 1, $\omega = w$. If IWC = 2, ω is a variable, (CONST)

K — Quarter-period of the elliptic integral $F(\phi,k)$ in subroutine (CONST) or = N-1 in subroutine GPOT

k_1 — Modulus of elliptic function (non-dimensional), (50)

k_2 — Modulus of elliptic function (non-dimensional), (58)

KC — Simpson's rule flag in subroutine RKTOM; when KC = 1, no Simpson's rule calculation is made; when KC = 2 (two complete steps of Runge-Kutta have been completed), the Simpson's rule calculation is made to check the accuracy

KDER — Input flag that indicates the model considered. Input KDER = 1 if the model is the same as the analytical model (J and D terms only and no sun or moon). Input KDER = 2 if any other perturbations are considered

KF — Intermediate failure counter in subroutine RKTOM

KFAIL — Total failure counter in subroutine RKTOM

KHALT — Halt flag. KHALT = 3 is normal halt upon completion; KHALT = 2 is halt due to failure of computing interval selection. (MAIN)

KR — Runge-Kutta flag indicating the Runge-Kutta cycle. (KR = 5 indicates two complete integration steps have been completed). (MAIN)

K1OR3	Flag to determine point about which perigee oscillates. Input Quantity = 1 if w closer to $\frac{\pi}{2}$, or 2 if w closer to $\frac{3\pi}{2}$
L	Quadrant of angle Z1 in subprogram QUAD2. L = 1, 2, 3, or 4; L = N-1, in subroutine GPOT
L_o	Initial angle of the ascending node to order ε (rad)
LS	Luni-solar flag. LS = 1 means consider luni-solar perturbation. LS = 2 means ignore luni-solar perturbation
MFAIL	Maximum number of failures in computing interval selection input to the program and to subroutine RKTOM
N	Counter in subprogram ELI which equals the last value of I done in the loop
NN1	N1+1, (GPOT)
NOT	Variable that counts the number of times the Landen transformation is used in subprogram ELIPE, also gives the index of the last calculated member of the arrays CA and CAP
NOT2	NOT + 2
NPTWO	Used to generate $2^{(i-1)}$ in subprogram ELI
N1	Degree of highest zonal harmonic to be considered, $N1 \leq 9$
N2	Degree of highest tesseral harmonic to be considered; $N2 \leq 6$

OMEG	Ω = longitude of ascending node (rad)
OMEGA	Ω_a = approximate Ω ($\Omega = \Omega_a + \Omega_n$), (rad) or = dummy variable for Ω in subroutine EXPERT or = dummy angular variable in subprogram QUAD1 which represents the angle that is to be placed in the proper quadrant
OMEGN	Ω_n = numerical correction to Ω (rad) $\dot{\Omega} = \dot{\Omega}_a + \dot{\Omega}_n$
OMEGT	$\Omega = \Omega_a + \Omega_n$, (ENCKE)
OMEOO	Ω_{oo}, (APSOL)
OMEO12	$\Omega_{o1/2}$, (APSOL)
OME32	$\Omega_{3/2}$, (APSOL)
OSK	$\frac{\sqrt{\kappa_1}}{\bar{s}_o}$, (CONST)
OSK2	$\frac{\kappa_1}{\bar{s}_0^2}$, (CONST)
OTD	Dimensional Ω_{TOTAL} for output, (degrees), (MAIN)
P	p = component of angular momentum along the polar axis (non-dimensional) or array of coefficients ρ_n in subroutine GPOT, P(10)
PA	p_a = approximate solution for p, (APSOL)
PHI	ϕ = independent variable, angle from node to satellite, (rad)

PHIB	$\bar{\phi} = \varepsilon^{3/2}\ \phi$, (APSOL)
PHIBT	$\bar{\phi}_{TOTAL} = \varepsilon^{3/2}\ \phi_{TOTAL}$, (APSOL)
PHIIB	$\bar{\phi}_i$
PHIK	Angle which is the number of complete revolutions of ϕ multiplied by 2π, (ELI)
PHILT	Array of angles ϕ used in decreasing Landen transformation. (ELI). Maximum dimension (10). Total angle not modded (rad)
PHIO	$\bar{\phi}_o$ = constant angle needed to calculate approximate perigee in case 1 or case 3. (CONST)
PHIS	Array of modded angles ϕ in subprogram ELI, maximum dimension (10). (rad)
PHISTP	Stopping condition for ϕ, input in degrees, used internally in radians. (MAIN)
PHIT	ϕ_{TOTAL} = total accumulated angle to compute secular terms, (MAIN)
PHITD	Total ϕ in degrees for output. (MAIN)
PHI1	ϕ_1 used to approximate $\frac{d}{d\phi}\left(\frac{d\phi}{dt}\right)$, (ENCKE)
PHI2	ϕ_2 used to approximate $\frac{d}{d\phi}\left(\frac{d\phi}{dt}\right)$, (ENCKE)
PI	π, (CONST)
PIO2	$\frac{\pi}{2}$, (CONST)

PN	p_n, (ENCKE)
PP	Dummy name for first three elements of labeled common array /EX/. Used in main program to eliminate changing values
PR	Variable used to accumulate the product in subprogram ELIPE
PT	Total $p = p_a + p_n$, (ENCKE)
PT2	p^2, (ENCKE)
P2	p_a^2, (CONST)
P4	p_a^4, (CONST)
q	$\frac{du}{d\phi}$ (non-dimensional) used to change second-order differential equation to two first-order differential equations
Q	Dummy variable for $\sin\theta$, (GPOT)
QA	q_a, (APSOL)
QN	q_n, (ENCKE)
QPER	Quarter-period of elliptic functions or integrals with modulus k_1 or k_2 in subroutine CONST Dummy variable in subprogam QUAD1 for same as above or $\frac{\pi}{2}$
QQ	Dummy name for last three elements of array stored in labeled common /EX/. Used in main program to prevent changing values that are stored there
QT	Total $q = q_a + q_n$, (ENCKE)

R	Mean equatorial radius (n. mi.) FORTRAN symbol for non-dimensional radius vector = $\frac{1}{u}$, (ENCKE)
r	Radius to satellite (non-dimensional with respect to R)
RAD	Conversion factor from degrees to radians. (MAIN)
RD	Dimensional r in subroutine EXPERT, (km)
REST	Dummy name for last 12 elements in labeled common array /APS/. Used in main program to prevent changing values that are stored in that part of the array
RK	$\sqrt{2\kappa_1}$ or $\sqrt{\kappa_o+\kappa_1}$ in subroutine CONST depending on perigee case number IC
RK1 RK2 RK3	Arrays of 6 which represent the Runge-Kutta parameters for each of the six dependent variables
RKINC	Array of 6 to compute the common increment used in HAH and SR
RMK	$\sqrt{\kappa_o-\kappa_1}$, quantity needed for case 3 perigee calculations, (CONST)
RR	Dummy storage array of dimension (125) for reference run usage in main program
RUM	$\frac{F}{DENOM}$, (ENCKE)
RX	Array of r^{n+2} (non-dimensional) in subroutine GPOT. RX(9) maximum

R1	r_1 Used to approximate $\frac{d}{d\phi}(\frac{d\phi}{dt})$, (ENCKE)
R2	r_2 Used to approximate $\frac{d}{d\phi}(\frac{d\phi}{dt})$, (ENCKE)
S	Array of 14 in which values of dependent variables, their derivatives, the time, and ϕ are saved for ordinary Runge-Kutta use, or two-dimensional array (6 x 6) of coefficients in computation of complete potential, (GPOT)
S_{nm}	Coefficients for computation of tesserals and sectorials of the earth's potential, used in subroutine GPOT
SBE	Array of sin (n • EW) in subroutine GPOT. SBE(6) maximum
SC	Two-dimensional array (6 x 6) of coefficients in computation of complete potential, (GPOT)
SI	$\sin i_{oo}$, (CONST), or $\sin i$, (ENCKE, EXPERT)
SI2	$\sin^2 i_{oo}$, (CONST), or $\sin^2 i$, (ENCKE)
SINP	sin (PHIS(I-1)), (ELI)
SIOC	$\sin i_{oc}$, (APSOL)
SN	Elliptic function sn, (APSOL)
SNVE	Quantity used recursively to find sn, (ELIF)
SOK	$\frac{\bar{S}_o}{\sqrt{\kappa_1}}$, (CONST)
SP	$\sin \phi$, (APSOL)

SPMW	$\sin(\phi_i - w)$ in subroutine CONST or $\sin(\phi - \omega)$ in subroutine APSOL
SPPW	$\sin(\phi + \omega)$, (APSOL)
SQ	$\sqrt{\bar{\kappa}_o - \kappa_1 \cos 2\omega}$, (APSOL) or $\sin q$, (EXPERT)
SQ1	$\sqrt{\frac{\bar{\kappa}_o}{\kappa_1} \cos 2\omega}$, (APSOL)
SR	Runge-Kutta increments over two complete computing intervals; SR(6)
SS	Array of 14 in which values of dependent variables, their derivatives, the time, and ϕ are saved for Simpson's rule use and in case of computing interval selection failure
ST	$\sin \theta$
SVE	sin (VEO), (ELIF)
SW	$\sin(w)$, (CONST)
SXW	$\sin \omega$
SOB	$\bar{S}_o$, (A.20)
SOBS	$\bar{S}_o^2$, (CONST)
S1	S_1, (25)
S1P	S_1', (24)

S1S	S_1^2, (CONST)
S2P	sin 2 ϕ, (APSOL)
S2PMW	sin 2 $(\phi-\omega)$, (APSOL)
S2T	sin 2θ, (ENCKE)
S2XW	sin 2ω
S3PMW	sin $(3\phi-\omega)$, (APSOL)
S4PMW	sin $(4\phi-2\omega)$, (APSOL)
t	Time (non-dimensional with respect to $\sqrt{\frac{R^3}{\mu}}$)
T	Total time since 1950.0 equinox = $t_a + t_n$, (ENCKE), also the dummy name for the independent variable in subroutine RKTOM
t_a	Approximate analytic time
t_n	Numerical correction to the time
TA	t_a = approximate solution for time (non-dimensional), (APSOL)
TAB1	Tape control array to read data from JPL ephemeris tapes, (EXPERT)
TAB2	Tape control array to read data from JPL ephemeris tapes, (EXPERT)

TANG	Value of $\tan^{-1}$ expression for time constant in subroutine CONST (rad), or value of $\tan^{-1}$ expression for t_a in subroutine APSOL
TD	Dimensional time in subroutine EXPERT (hours)
TF	Dummy variable in input array of subroutine RKTOM which represents the maximum desired value of the independent variable
THETA	FORTRAN symbol for θ
TI	$\tan i_{oo}$, (CONST), or $\tan i$, (ENCKE)
TILT	Dummy variable for inclination in subroutine EXPERT
TN	"Next time" after Runge-Kutta step would be completed
TOTE	Total energy which is computed and printed when only J and D perturbations are considered
TS	Place to accumulate double sum of tesserals and sectorials for a_f in subroutine GPOT.
TS1	Place to accumulate double sum of tesserals and sectorials for a_g in subroutine GPOT
TS2	Place to accumulate double sum of tesserals and sectorials for a_h in subroutine GPOT.
TWON	2^n accumulation in subroutine ELI
TWOPI	2π, (CONST)
TW2	$\tan \frac{w}{2}$, (CONST)

TO	Initial time (non-dimensional)
TO1	Constant used in approximation for time, (CONST), (non-dimensional)
U	Earth's potential (non-dimensionalized), or in FORTRAN a symbol for u = reciprocal of non-dimensionalized radius (divided by R), or dummy angular variable in subprogram ELIF which is the argument of sn (rad), or two-dimensional array of coefficients for perturbative accelerations in subroutine GPOT. U(6, 6) maximum
U_{nm}	$\sec\phi \cdot \rho_n^m$, (GPOT)
U_1	Small terms in the radial acceleration (non-dimensional), (96)
UA	u_a = approximate u (non-dimensional), $u = u_a + u_n$
UN	u_n = numerical correction to u (non-dimensional), $u = u_a + u_n$
UT	Total $u = u_a + u_n$, (ENCKE)
UOO	u_o, (APSOL)
UO1	Quantity to store zero in the location for zero index in array U in subroutine GPOT
U1	εu_1, (APSOL)
U2	u_t^2, (ENCKE)
U3	u_t^3, (ENCKE)

U5	u_t^5, (ENCKE)
V_o	All small terms in $\frac{d\phi}{dt}$ (non-dimensional), (87)
V_1	All small terms in $\frac{d}{d\phi}(\frac{d\phi}{dt})$ (non-dimensional), (88) and (89)
V_3	Small terms in $\frac{dq}{d\phi}$ (non-dimensional), (97) and (98)
VEO	$\frac{U}{PR}$ = last reduced argument, (ELIF)
VU2	$\frac{V_o}{A_1 u^2}$, (ENCKE)
VO	V_o, (ENCKE)
VO2	VU2 (2 + VU2), (ENCKE)
V1	V_1, (ENCKE)
V22	$(1 + \frac{V_o}{A_1 u^2})^2$, (ENCKE)
V3P	V_3', (ENCKE)
W	Dummy variable in subroutine CONST for $w^* = w$ = initial angle of perigee (rad), or dummy variable for angle which determines the quadrant in subprogram QUAD1, or two dimensional (6 x 6) array for the W_{nm} in subroutine GPOT
W_{nm}	$\cos\phi \cdot \rho_n^{m'}$, (GPOT)
WO2	$\frac{w}{2}$, (CONST)
XI	i = inclination (rad) $= i_{oo} = i_{oo}^*$ in subroutine CONST
XIA	i_a = approximate inclination, $i = i_a + i_n$, (rad)

XIN	i_n = numerical correction to inclination $i = i_a + i_n$, (rad)
XINCI	Initial value of the inclination in degrees, (APSOL)
XIT	Total inclination = $i_a + i_n$, (ENCKE)
XITD	Total inclination in degrees for output, (MAIN)
XIOC	i_{oc}, (APSOL)
XI1	ϵi_1, (APSOL)
XI12	$i^*_{o1/2}$, (APSOL)
XLO	L_o (constant related to Ω_i) when input, changed to $L_o + L_{1/2}$ in subroutine APSOL to make initial $L_o \approx \Omega_i$
XMOD	Modulus of elliptic functions and integrals = k_1 or k_2 depending on perigee case number IC. (CONST)
XNODEI	Initial value of ascending node in degrees, (APSOL)
XW	Analytic value for the **osculating argument of perigee, (rad),** (APSOL or CONST)
Z	Place to accumulate the sum $\sum_{n=2}^{N1} (\frac{J_n}{r^{n+2}}) \rho'_n$ in subroutine GPOT. Also name of input array of dimension (125) in main program
Z1	Dummy angular variable in subprogram QUAD2 used to determine the quadrant of the first argument
Z1	Place to accumulate the sum $\sum_{n=2}^{N1} (n+1) (\frac{J_n}{r^{n+2}}) \rho_n$ in subroutine GPOT

z_1	Angle used to find quadrant of ω, (rad), (53)
z_2	$\sqrt{\overline{\kappa}_o + \kappa_1}\ (\overline{\phi} - \overline{\phi}_o)$ angle used to find the quadrant of ω, (rad)
Z2	z_2, (APSOL)
ZD	$1+\cos(\phi_i - w)$ in subroutine CONST or $1+\cos(\phi-\omega)$ in subroutine APSOL
γ_1	Constant defined in (48), (non-dimensional)
ε	Coefficient of the second zonal harmonic of the earth's potential, (non-dimensional)
θ	Complement of the latitude, (rad)
$\overline{\kappa}_o$	Constant defined in (A.21)
κ_1 and κ_1'	Constants defined in (28) and (29)
λ	Instantaneous East longitude of the satellite measured from Greenwich (FORTRAN symbol EW)
λ_G	Instantaneous longitude of Greenwich measured for equinox of 1950.0 (FORTRAN symbol EWOG)
λ_{OG}	Longitude of Greenwich measured from equinox of 1950.0 at $t = 0$
μ	GM_{earth} $\left(\frac{n.mi.^3}{hr.^2}\right)$

ρ_n, ρ_n^m, ρ_n', etc	Coefficients used for calculation of the complete potential in subroutine GPOT. Defined in (111) ff
ϕ	Angle from ascending node to satellite, (rad)
$\bar{\phi}$	$\varepsilon^{3/2}\ \phi$, "slow variable", (rad)
$\bar{\phi}_o$	Constant of integration defined by (51) or (59)
χ_1^*	Angle used to find constant of integration for ω solution, (52)
χ_2	Angle used to find constant of integration for ω solution, (60)
ψ	Longitude (rad)
Ω	Longitude of the ascending node (rad) measured from equinox of 1950.0
ω	Argument of perigee, (rad)
ω_E	Mean rotation rate of the earth, (non-dimensional)

SUBSCRIPTS

a	Approximate
n	Numerical
t or T	Total
o, o 1/2, oo 1, 1.2, oc	Denote various orders of the approximate solution

Section 3

SOURCES OF EQUATIONS

3.1 FORMULATION OF THE PROBLEM AND THE APPROXIMATE SOLUTION

In general, it is the purpose of this program to solve a set of simultaneous differential equations by a combination of numerical and analytical methods which might be called a modified-Encke solution. Thus, for the problem:

$$\ddot{X} = f(X,t),$$

$$\ddot{X}_n = f(X_a + X_n,t) - \ddot{X}_a,$$

where X_a is an approximate solution, X_n is the correction obtained by solving the latter differential equation numerically, and the complete solution is then $X = X_a + X_n$. In the normal Encke method, the approximate solution is taken as the two-body solution (a fixed Keplerian ellipse). In the modified-Encke approach, the approximate solution will be a solution of the oblateness problem considering the first, second, and fourth zonal harmonics of the potential. The approximate solution differs from reality for two reasons. First, the mathematical model is necessarily simplified from the actual physical case, and second, the solution only approximates the true solution of the simplified problem. The numerical solution accounts for both of these discrepancies. For this program, the complete model will include zonal, tesseral, and sectorial harmonics of the potential up to and including the coefficients C_{66} and S_{66}, in addition to luni-solar perturbations.

The general equations of motion, nomenclature, and approximate solution to the oblateness problem as described in reference 1 are used as a framework for this program. For convenience, all equations taken directly from this reference will be given the original numbering at the left in addition to consecutive numbering for this report on the right.

The complete set of differential equations is given in equation (3.5) of the reference and consists of four first-order equations and one second-order equation. In this formulation, the independent variable is the angle ϕ between the ascending node and the radius vector, and the dependent variables are p (component of angular momentum along the polar axis), Ω (argument of the ascending node), i (instantaneous inclination of the orbital plane), u (reciprocal of the radius), and t (time). These equations are:

(3.5a)
$$\frac{dp}{d\phi} = \frac{\frac{\partial U}{\partial \psi}}{\frac{pu^2}{\cos i} + \frac{\cos^3 i \cos \theta}{p \sin^2 i \sin \theta} F} , \tag{1}$$

(3.5b)
$$\frac{d\Omega}{d\phi} = \frac{- \cos^3 i \cos \theta \; F}{p^2 u^2 \sin^2 i \sin \theta + \cos^4 i \cos \theta \; F} , \tag{2}$$

(3.5c)
$$\frac{di}{d\phi} = \frac{- \sin^2 i \cos^3 i \cos \phi \; F}{p^2 u^2 \sin^2 i \sin \theta + \cos^4 i \cos \theta \; F} , \tag{3}$$

(3.5d)
$$\frac{d^2 u}{d\phi^2} - \frac{2}{u} \left(\frac{du}{d\phi}\right)^2 + \frac{\frac{du}{d\phi} \frac{d}{d\phi} \left(\frac{d\phi}{dt}\right)}{\frac{d\phi}{dt}} = - \frac{\frac{p^2 u^5}{\cos^2 i} + u^2 \frac{\partial U}{\partial r}}{\left(\frac{d\phi}{dt}\right)^2} , \tag{4}$$

(3.5g)
$$\frac{dt}{d\phi} = \left[\frac{pu^2}{\cos i} + \frac{\cos^3 i \cos \theta \; F}{p \sin^2 i \sin \theta}\right]^{-1} , \tag{5}$$

where the co-latitude θ is related to i and ϕ by

(3.5h)
$$\cos \theta = \sin i \sin \phi , \tag{6}$$

$$\sin \theta = + \sqrt{1 - \cos^2 \theta} , \tag{7*}$$

---

* Since $0 \leq \phi \leq 180°$

where U is the potential of the central body, and

$$F \equiv \frac{\partial U}{\partial \theta} + \tan i \, \frac{\cos \phi}{\sin \theta} \frac{\partial U}{\partial \psi} . \tag{8}$$

These equations of motion are exact for any satellite orbiting around a central body of potential U. To consider additional perturbations, the equations can be kept unaltered by including the appropriate components of the perturbative accelerations in the quantities $\frac{\partial U}{\partial \psi}$, $\frac{\partial U}{\partial r}$, and F. The equations would then still be exact.

Defining the accelerations a_f, a_g, and a_h in a local orthogonal frame with a_h outward along the geocentric vertical, a_f directed south, and a_g directed east:

$$a_f = \frac{1}{r} \frac{\partial U}{\partial \theta} , \tag{9}$$

$$a_g = \frac{1}{r \sin \theta} \frac{\partial U}{\partial \psi} , \tag{10}$$

and

$$a_h = \frac{\partial U}{\partial r} . \tag{11}$$

Then, from equations (8), (9), and (10):

$$F = \frac{\partial U}{\partial \theta} + \tan i \, \frac{\cos \phi}{\sin \theta} \frac{\partial U}{\partial \psi}$$

$$= \frac{1}{u} (a_f + \tan i \cos \phi \, a_g). \tag{12}$$

The analytical solution of reference 1 only includes the first, second, and fourth zonal harmonics of the earth's potential. To include more terms of the potential, the a_f, a_g, and a_h accelerations will be used directly from reference 2, pages 4-97 and 4-98. (Repeated in this report, equations 108, 109, and 110.)

It is also desirable to change the original equations of motion into six first-order equations rather than having one second-order equation. Equation (4) then is replaced by the following two equations:

$$\frac{du}{d\phi} \equiv q \ , \tag{13}$$

and

$$\frac{dq}{d\phi} = \frac{2}{u} q^2 - \frac{q \frac{d}{d\phi} \left(\frac{d\phi}{dt}\right)}{\frac{d\phi}{dt}} - \frac{\frac{p^2 u^5}{\cos^2 i} + u^2 \frac{\partial U}{\partial r}}{\left(\frac{d\phi}{dt}\right)^2} \ . \tag{14}$$

These two equations need special consideration when finding the numerical differential equations. Using subscripts a and n for approximate and numerical solutions and defining the right side of equation (14) as the function $G(p, q, u, i, \phi)$ leads to:

$$\frac{dq_n}{d\phi} = G\left(p_a + p_n, q_a + q_n, u_a + u_n, i_a + i_n, \phi\right) - \frac{d^2 u_a}{d\phi^2} \tag{15}$$

and

$$\frac{du_n}{d\phi} = q_n, \ \frac{du_a}{d\phi} = q_a \ . \tag{16}$$

Now the appropriate approximate solutions will be selected from reference 1. From equation (3.6a):

$$p_a = \text{const.} = \text{initial } p. \tag{17}$$

From equations (3.71), (3.73), and (3.76):

$$\Omega_a = \frac{\Omega_{oo}}{\varepsilon^{1/2}} + \Omega_{o\ 1/2} + \varepsilon\ \Omega_{3/2} + L_o. \tag{18}$$

Before writing the expressions for Ω_{oo}, etc., it should be mentioned that some numerical difficulties would be experienced by using the results of reference 1 exactly as written. The results of the reference are algebraically correct and pose no analytical ambiguities. However, in certain cases there are apparent indeterminacies which a computer cannot handle. Most of these can be eliminated by minor modifications of constants, but several quantities will still require two or more different forms for accurate and correct numerical evaluation.

Let

$$A \equiv \frac{p}{\cos i_{oo}} \tag{19}$$

designate the total initial angular momentum.

From equation (A.11), define:

$$B_2{*}' \equiv \frac{1}{2A^8} \{(\frac{1}{12} - \frac{3}{2} c) (1 - e_o^2)$$

$$+ \cos^2 i_{oo} [(1 - e_o^2) 12c - \frac{7}{3} + \frac{4}{3} e_o^2]$$

$$+ \cos^4 i_{oo} [- \frac{21}{2} c (1 - e_o^2) + \frac{5}{4} (5 - e_o^2)]\}, \tag{20}$$

so

$$B_2{*} \equiv e_o B_2{*}'. \tag{21}$$

Equation (20) uses equation (3.38), i.e.,

$$i_{oo}{*} = i_{oo}, \; e_o{*} = e_o, \; w{*} = w.$$

From equation (A.14), define:

$$C_2{*}' = \frac{1}{4A^8} [- \frac{1}{6} + 3c + (\frac{5}{2} - 21c) \cos^2 i_{oo}], \tag{22}$$

so

$$C_2^* \equiv e_o^{\,2} \cos i_{oo} \sin i_{oo} C_2^{*'} . \tag{23}$$

From equation (A.18), define:

$$S_1' = \frac{1}{2A^4} (2 - 15 \cos^2 i_{oo}), \tag{24}$$

so

$$S_1 \equiv \tan i_{oo} S_1' . \tag{25}$$

(A.20)

$$\overline{S}_o \equiv - \frac{1}{2\, \varepsilon^{1/2} A^4} (1 - 5 \cos^2 i_{oo}). \tag{26}$$

(A.21)

$$\overline{\kappa}_o = \overline{S}_o^{\,2} + \kappa_1 \cos 2w. \tag{27}$$

Equation (27) assumes $j_{1/2}^* = 0$. In reference 1 it is assumed that constants of integration can be expressed in series form. For the program, the leading term will be taken as accurately as desired, and all the higher order terms will then be zero except for $L_{1/2}$. To make the constant L_o approximately equal to the initial value of the ascending node, the $L_{1/2}$ constant is chosen to make $\Omega_{o1/2} = 0$ initially.

From equation (A.22), define:

$$\kappa_1' = S_1' C_2^{*'}, \tag{28}$$

so

$$\kappa_1 = e_o^{\,2} \sin^2 i_{oo} \kappa_1' . \tag{29}$$

From equation (A.23):

$$E_{1/2} = - \frac{B_2^* \overline{S}_o}{\kappa_1} = - \frac{B_2^{*'} \overline{S}_o}{e_o \sin^2 i_{oo} \kappa_1'} . \tag{30}$$

Now we can return to writing the approximate expressions.

(3.79) $$\Omega_{oo} = - \frac{\cos i_{oo}}{A^4} \bar{\phi}. \qquad (31)$$

(3.82) $$\Omega_{o1/2} = - \frac{5 \cos i_{oo}}{A^4 S_1'} [\bar{S}_o \bar{\phi} - \omega_o] + L_{1/2}. \qquad (32)$$

(3.80) $$\Omega_{3/2} = - \frac{\cos i_{oo}}{A^4} [- \frac{\sin 2\phi}{2} + e_o \sin (\phi - \omega_o)$$

$$- \frac{e_o}{2} \sin (\phi + \omega_o) - \frac{e_o}{6} \sin (3\phi - \omega_o)] \qquad (33)$$

As stated in equation (18), $\Omega_a = \frac{\Omega_{oo}}{\varepsilon^{1/2}} + \Omega_{o1/2} + \varepsilon\, \Omega_{3/2} + L_o$. From equations (3.43) and (3.48b):

$$i_a = i_{oo} + \varepsilon^{1/2}\, i_o{}^*{}_{1/2} + \varepsilon\, i_1 , \qquad (34)$$

since $i_{1/2} = 0$ from reference 1, page 27.

$$i_{oo} = \text{initial inclination to order } \varepsilon. \qquad (35)$$

The equation for $i_{o\,1/2}{}^*$ is given in equation (3.33) but two forms are required for the numerical evaluation.

When $62° \leq i_{oo} \leq 65°$ (Limit on i_{oo} satisfied when $|1 - 5 \cos^2 i_{oo}| \leq 0.106$), use:

(3.33) $$i_{o\,1/2}{}^* = \frac{1}{S_1} [\pm (\bar{\kappa}_o - \kappa_1 \cos 2\omega)^{1/2} - \bar{S}_o]. \qquad (36)^*$$

---

* The sign of the square root must agree with the sign of the numerical value of $\bar{S}_o$.

Otherwise, use the form:

$$i_{o\,1/2}{}^{*} = C_2{}^{*} \frac{[\cos 2w^{*} - \cos 2\omega]}{\overline{S}_o \pm\sqrt{\overline{\kappa}_o - \kappa_1 \cos 2\omega}} . \qquad (37)^{*}$$

Equations (36) and (37) are algebraically equivalent with equation (37) coming from equation (36) by multiplying and dividing by $[\sqrt{\overline{\kappa}_o - \kappa_1 \cos 2\omega} + \overline{S}_o]$ and by using equations (A.21) and (A.22) for $\overline{\kappa}_o$ and κ_1, and equations (23) and (25) for $C_2{}^{*}$ and S_1.

The final expression required for i_a is then:

(3.15a)

$$i_1 = \frac{\cos i_{oo} \sin i_{oo}}{2 A^4} [\cos 2\phi + e_o \cos (\phi + \omega) + \frac{e_o}{3} \cos (3\phi - \omega)]. \qquad (38)$$

Since the differential equation for u was changed from one second-order equation to two first-order equations in u and $\frac{du}{d\phi}$, the approximate values for both these quantities are required.

Repeating the equation from reference 1:

(3.48a)

$$u_a = \frac{\cos^2 i_{oc}}{p^2} [1 + e_a \cos (\phi - \omega)] + \varepsilon\, u_1, \qquad (39)$$

(since $u_{1/2} = 0$ from reference 1, page 27).

From equation (3.43):

$$i_{oc} = i_{oo}{}^{*} + \varepsilon^{1/2}\, i_{o\,1/2}{}^{*}. \qquad (40)$$

---

* The sign of the square root must agree with the sign of the numerical value of $\overline{S}_o$.

To compute u_a, the expressions e_a, ω, and u_1 are needed.

From equation (3.41):

$$e_a = e_o^* + \varepsilon^{1/2} e_{1/2}^* . \tag{41}$$

$$e_o^* = \text{constant}, \tilde{=} \text{ initial } e. \tag{42}$$

The variable $e_{1/2}^*$ is given in equation (3.35), but again different forms are required for numerical evaluation by computer. As in the development of $i^*_{o\ 1/2}$, if i_{oo} is not between 62° and 65°, use:

$$e_{1/2}^* = \frac{B_2^* (\cos 2w^* - \cos 2\omega)}{\bar{S}_o \pm \sqrt{\bar{\kappa}_o - \kappa_1 \cos 2\omega}} . \tag{43)*$$

If $62° \leq i_{oo} \leq 65°$, then use:

$$e_{1/2}^* = \gamma_1 \left[\pm \sqrt{\frac{\bar{\kappa}_o}{\kappa_1} - \cos 2\omega} - \frac{\bar{S}_o}{\sqrt{\kappa_1}}\right] \tag{44)*$$

providing $\bar{S}_o^{\,2} \leq \kappa_1$. Otherwise, use:

$$e_{1/2}^* = \gamma_1 \frac{\sqrt{\kappa_1}}{\bar{S}_o} \frac{(\cos 2w^* - \cos 2\omega)}{1 + \sqrt{1 + \frac{\kappa_1}{\bar{S}_o^{\,2}} (\cos 2w^* - \cos 2\omega)}} . \tag{45}$$

Again, all these formulas are mathematically identical, but they are required because of possible ambiguities in computer calculations. In equations (44) and (45), the quantity γ_1 is defined by:

* The sign of the square root must agree with the sign of the numerical value of $\bar{S}_o$.

$$\gamma_1 \equiv \frac{B_2{}^{*'}}{\sin i_{oo} \sqrt{\kappa_1{}'}} . \qquad (46)$$

The solution for ω also requires three forms, given as three cases in reference 1 depending upon the relative values of $\bar{\kappa}_o$ and $\bar{\kappa}_1$.

For case 1 when $-\kappa_1 < \bar{\kappa}_o < \kappa_1$:

(3.55)
$$\omega^* = \pm \tan^{-1} \{\frac{\kappa_1 - \bar{\kappa}_o}{\kappa_1 + \bar{\kappa}_o} (1 + tn^2 [\sqrt{2 \kappa_1} (\bar{\phi} - \bar{\phi}_o)])\}^{1/2}$$

$$= \pm \tan^{-1} \{[\frac{\kappa_1 - \bar{\kappa}_o}{\kappa_1 + \bar{\kappa}_o}]^{1/2} \cdot \frac{1}{cn [\sqrt{2 \kappa_1} (\bar{\phi} - \bar{\phi}_o)]}\} \qquad (47)$$

where the modulus of tn or cn is k_1 and

(3.54c)
$$k_1 = [\frac{\bar{\kappa}_o + \kappa_1}{2 \kappa_1}]^{1/2} . \qquad (48)$$

From equations (3.54a) and (3.54b):

$$\bar{\phi}_i - \bar{\phi}_o \equiv \pm (2 \kappa_1)^{-1/2} F (\chi_1{}^*, k_1) \qquad (49)$$

($\pm$ same sign as $\bar{S}_o$ which is sign of $\frac{d\omega}{d\phi}$ at $\omega = w$)

and

$$\chi_1{}^* = \pm \tan^{-1} [\frac{\kappa_1 + \bar{\kappa}_o}{\kappa_1 - \bar{\kappa}_o} \tan^2 w^* - 1]^{1/2}, \qquad (50)$$

and the sign is chosen so w^* and $\chi_1{}^*$ are in the same quadrant. In these expressions, $F(\chi_1{}^*, k_1)$ is an elliptic integral of the first kind and tn and cn are elliptic functions. To determine the quadrant of ω^* from equation (47), a new angle and K (the quarter-period of the elliptic functions cn or tn) are used. Let

$$z_1 \equiv \sqrt{2\,\kappa_1}\,(\bar{\phi} - \bar{\phi}_o), \qquad (51)$$

now the quadrant of ω^* can be related to the quadrant (defined by K) of z_1.

z_1	ω^* (when w^* nearer $\pi/2$)	ω^* (when w^* nearer $3\pi/2$)
0 - K	$0 - \pi/2$	$\pi - 3\pi/2$
K - 2K	$\frac{\pi}{2} - \pi$	$3\pi/2 - 2\pi$
2K - 3K	$\frac{\pi}{2} - \pi$	$3\pi/2 - 2\pi$
3K - 4K	$0 - \frac{\pi}{2}$	$\pi - 3\pi/2$
4K - 5K	$0 - \frac{\pi}{2}$	$\pi - 3\pi/2$
etc.		

For case 2, when $\bar{\kappa}_o = \kappa_1$, there are two possibilities. If $w^* = 0$ or π, then ω^* always equals 0 or π. If w^* has any other value, ω^* is given by the formula:

$$\tan\frac{\omega^*}{2} = e^{\pm\sqrt{2\kappa_1}\,(\bar{\phi} - \bar{\phi}_i)}\,\tan\frac{w^*}{2}\,. \qquad (52)$$

Here the $\pm$ sign is determined from the sign of the quantity, $\bar{S}_o + S_1\, i^*_{o1/2}$, since

(3.27c) $$\frac{d\omega^*}{d\bar{\phi}} \equiv \bar{S}_o + S_1 \, i^*_{o1/2} \qquad (53)$$

(Can use only $\bar{S}_o$ since derivative always has the same sign in this case.) Also the quadrant of ω^* is determined by the fact that the quadrant of $\frac{\omega^*}{2}$ is the same as the quadrant of $\frac{w^*}{2}$. Physically this means that for this case, the perigee either starts at 0 or π and remains there or approaches one of these values as the time becomes very large. The limit to which the perigee travels is not determined by the nearest of the two values, but by the sign of the derivative given in equation (53). Equation (52) replaces equation (3.59) of reference 1. This is done because the integral (3.58) should read

$$\bar{\phi} - \bar{\phi}_o = \int_o^{\omega^*} \frac{d\varepsilon}{(2\kappa_1)^{1/2} \sin \varepsilon}$$

rather than with $\cos \varepsilon$ replacing $\sin \varepsilon$ as shown in the reference.

Case 3 occurs when $\bar{\kappa}_o > \kappa_1$. In this case,

(3.65) $$\omega^* = \tan^{-1}\{[\frac{\bar{\kappa}_o - \kappa_1}{\bar{\kappa}_o + \kappa_1}]^{1/2} \operatorname{tn}[(\bar{\kappa}_o + \kappa_1)^{1/2} (\bar{\phi} - \bar{\phi}_o)]\}$$

$$= \tan^{-1}\{[\frac{\bar{\kappa}_o - \kappa_1}{\bar{\kappa}_o + \kappa_1}]^{1/2} \frac{\{1 - \operatorname{cn}^2[(\bar{\kappa}_o + \kappa_1)^{1/2} (\bar{\phi} - \bar{\phi}_o)]\}^{1/2}}{\operatorname{cn}[(\bar{\kappa}_o + \kappa_1)^{1/2}(\bar{\phi} - \bar{\phi}_o)]}\}, \qquad (54)$$

where tn and cn are elliptic functions with modulus k_2 and

$$k_2 = [\frac{2\kappa_1}{\kappa_1 + \bar{\kappa}_o}]^{1/2} . \quad (55)$$

This is the correct modulus and replaces the k_2 given in equation (3.64b) of reference 1.

As in case 1, the quadrant of $z_2 = (\bar{\kappa}_o + \kappa_1)^{1/2}(\bar{\phi} - \bar{\phi}_o)$, determined by K, provides the quadrant of ω^*. In this case, the quadrants are equal, i.e.,

z_2	ω^*
0-K	$0-\pi/2$
K-2K	$\pi/2 - \pi$
etc.	

The quantity $\bar{\phi}_o$ must be determined from equation (3.64a) and (3.64c) of reference 1:

(3.64a) $$\bar{\phi}_i - \bar{\phi}_o = \pm(\bar{\kappa}_o + \kappa_1)^{-1/2} F(\chi_2,k_2), \quad (56)$$

($\pm$ same sign as $\bar{S}_o$)

(3.64c) $$\chi_2 = \pm \tan^{-1}\{[\frac{\bar{\kappa}_o + \kappa_1}{\bar{\kappa}_o - \kappa_1}]^{1/2} \tan w^*\} . \quad (57)$$

In equation (57), the sign and quadrant are chosen such that χ_2 and w^* are in the same quadrant.

Finally, all that is required for u_a is the expression for u_1. This comes directly from reference 1.

(3.15b) $$u_1 = \frac{1}{2A^6}\{-1 + 3\cos^2 i_{oo} - \frac{e_o^2}{2}(1 - 5\cos^2 i_{oo})$$

$$+ \frac{e_o^2}{4}(1 - 3\cos^2 i_{oo})\cos 2\omega - (\frac{1}{3}\sin^2 i_{oo} - \frac{e_o^2}{3} + \frac{5}{6}e_o^2 \sin^2 i_{oo}) \cdot \cos 2\phi$$

$$+ \frac{e_o^2}{6}(1 - 9\cos^2 i_{oo})\cos 2(\phi - \omega) - \frac{e_o}{12}(5 - 11\cos^2 i_{oo}) \cdot \cos(3\phi - \omega)$$

$$- \frac{e_o^2}{12}(1 - 3\cos^2 i_{oo})\cos(4\phi - 2\omega)\}. \quad (58)$$

Now,

$$q_a \equiv \frac{du_a}{d\phi} . \quad (59)$$

From equations (39) and (40):

$$q_a = -\frac{2\cos i_{oc}\sin i_{oc}}{p^2}[1 + e_a\cos(\phi - \omega)]\varepsilon^{1/2}\frac{di^*_{o1/2}}{d\phi}$$

$$+ \frac{\cos^2 i_{oc}}{p^2}[\frac{de_a}{d\phi}\cos(\phi - \omega) - e_a(1 - \frac{d\omega}{d\phi})\sin(\phi - \omega)] + \varepsilon\frac{du_1}{d\phi} . \quad (60)$$

All these derivatives will be given in the following section.

3.2 DERIVATIVES REQUIRED FOR THE GENERALIZED ENCKE SOLUTION

Derivatives of all the approximate solutions must be taken to find the differential equations to be numerically integrated. These derivatives are taken rather than using the original derivatives of the theory since in some cases approximations are made to carry out the integration.

From equation (3.6a),

$$\frac{dp_a}{d\phi} = 0 \; . \tag{61}$$

From equation (18),

$$\frac{d\Omega_a}{d\phi} = \frac{1}{\varepsilon^{1/2}} \left[\frac{d\Omega_{oo}}{d\phi} + \varepsilon^{1/2} \frac{d\Omega_{o1/2}}{d\phi} + \varepsilon^{3/2} \frac{d\Omega_{3/2}}{d\phi}\right] . \tag{62}$$

From equation (3.79)

$$\frac{d\Omega_{oo}}{d\phi} = \frac{d\Omega_{oo}}{d\bar{\phi}} \frac{d\bar{\phi}}{d\phi} = - \frac{\cos i_{oo}}{A^4} (\varepsilon^{3/2}) \tag{63}$$

From (3.82) and (25):

$$\frac{d\Omega_{o\,1/2}}{d\phi} = - \frac{5}{A^4 \, S_1'} \cos i_{oo} \, \varepsilon^{3/2} \left(\bar{S}_o - \frac{d\omega}{d\bar{\phi}}\right). \tag{64}$$

Equation (3.27c) will always be used for $\frac{d\omega}{d\bar{\phi}}$, i.e.,

(3.27c)

$$\frac{d\omega}{d\bar{\phi}} = \bar{S}_o + S_1 \, i^*_{o\,1/2} \; . \tag{65}$$

Then from equations (64), (65), and (25)

$$\frac{d\Omega_{o\,1/2}}{d\phi} = \frac{5 \, \varepsilon^{3/2}}{A^4} \sin i_{oo} \, i^*_{o\,1/2} \; . \tag{66}$$

From equation (3.80):

$$\frac{d\,\Omega_{3/2}}{d\phi} = - \frac{\cos i_{oo}}{A^4} \{- \cos 2\phi + e_o \cos (\phi - \omega) (1 - \omega')$$

$$- \frac{e_o}{2} \cos (\phi + \omega) (1 + \omega') - \frac{e_o}{6} \cos (3\phi - \omega) (3 - \omega')\}, \tag{67}$$

where from equation (3.21c):

$$\omega' \equiv \frac{d\omega}{d\phi} = \frac{d\omega}{d\bar{\phi}} \frac{d\bar{\phi}}{d\phi} = \varepsilon^{3/2} \frac{d\omega}{d\bar{\phi}} , \tag{68}$$

and $\frac{d\omega}{d\bar{\phi}}$ is given in equation (65).

Next the derivatives of i_a will be given from equation (34):

$$\frac{di_a}{d\phi} = \varepsilon^{1/2} \frac{di^*_{o1/2}}{d\phi} + \varepsilon \frac{di_1}{d\phi}, \tag{69}$$

since $\frac{di_{oo}}{d\phi} = 0.$

From equations (3.33), (3.34), and (A.22):

$$\frac{di^*_{o1/2}}{d\phi} = \varepsilon^{3/2} C^*_2 \sin 2\omega. \tag{70}$$

This agrees with equation (3.29a), so that (3.29a) was integrated exactly. Also note that there is only one form for $\frac{di^*_{o1/2}}{d\phi}$ while $i^*_{o1/2}$ required two different algebraic forms for computation.

From equation (3.15a):

$$\frac{di_1}{d\phi} = - \frac{\cos i_{oo} \sin i_{oo}}{2 A^4} [2 \sin 2\phi + e_o (1 + \omega') \sin (\phi + \omega)$$

$$+ \frac{e_o}{3} (3 - \omega') \sin (3\phi - \omega)] \tag{71}$$

Equation (63) gave an expression for q_a, but some derivatives were required and they will be formed here.

From equation (3.41):

$$\frac{de_a}{d\phi} = \varepsilon^{1/2} \frac{de^*_{1/2}}{d\phi} \tag{72}$$

$$\frac{de^*_{1/2}}{d\phi} = \varepsilon^{3/2} B^*_2 \sin 2\omega, \tag{73}$$

Thus, equation (3.29b) was integrated exactly, and no special cases are required for the derivative of $e^*_{1/2}$.

Now the derivative of u_1 is needed. From equation (3.15b):

$$\begin{aligned}\frac{du_1}{d\phi} = \frac{1}{2A^6} \{ &- \frac{e_o^2}{2} \omega' (1 - 3 \cos^2 i_{oo}) \sin 2\omega \\ &+ 2 \left(\frac{\sin^2 i_{oo}}{3} - \frac{e_o^2}{3} + \frac{5 e_o^2 \sin^2 i_{oo}}{6}\right) \sin 2\phi \\ &- \frac{e_o^2}{3} (1 - 9 \cos^2 i_{oo}) (1 - \omega') \sin 2 (\phi - \omega) \\ &+ \frac{e_o}{12} (5 - 11 \cos^2 i_{oo}) (3 - \omega') \sin (3\phi - \omega) \\ &+ \frac{e_o^2}{12} (1 - 3 \cos^2 i_{oo}) (4 - 2\omega') \sin (4\phi - 2\omega)\}\end{aligned} \tag{74}$$

This completes q_a and now $\frac{dq_a}{d\phi}$ is required. The form for this derivative will be chosen to allow analytic cancellation of the terms of order unity when forming the modified-Encke equations of motion. If this were not done, accuracy would be lost trying to find numerically the small difference between two large numbers. Define:

$$\frac{dq_a}{d\phi} \equiv \frac{dq_o}{d\phi} + \varepsilon \frac{dq_1}{d\phi} ,$$

where

$$q_o = \frac{du_o}{d\phi} , \quad q_1 = \frac{du_1}{d\phi} . \tag{75}$$

q_o can be found from equation (60) and q_1 from equation (74). From equation (60) by differentiation:

$$\frac{dq_o}{d\phi} = - u_o + \frac{\cos^2 i_{oc}}{p_a^2} + H, \tag{76}$$

where

$$H = - \frac{2 u_o}{\cos^2 i_{oc}} \left[\varepsilon \left(\frac{d\, i^*_{o1/2}}{d\phi} \right)^2 \cos^2 i_{oc} + \frac{\varepsilon^{1/2}}{2} \frac{d^2\, i^*_{o1/2}}{d\phi^2} \sin 2i_{oc} \right]$$

$$+ \frac{\cos i_{oc}}{p_a^2} \{ - 4 \frac{di_{oc}}{d\phi} \frac{de}{d\phi} \sin i_{oc} \cos (\phi - \omega)$$

$$+ 2 \left[e_a \frac{di_{oc}}{d\phi} \sin i_{oc} - \cos i_{oc} \frac{de_a}{d\phi} \right] (1 - \omega') \sin (\phi - \omega)$$

$$+ \left[\cos i_{oc} \frac{d^2 e_a}{d\phi^2} + e_a \cos i_{oc} \;\; \omega' (2 - \omega') \right] \cos (\phi - \omega)$$

$$+ e_a \cos i_{oc} \;\; \omega'' \sin (\phi - \omega) \}, \tag{77}$$

and from equations (72) and (73):

$$\frac{d^2 e_a}{d\phi^2} = 2 \varepsilon^2 B_2^* \;\; \omega' \cos 2\omega , \tag{78}$$

from equations (65), (68), and (70):

$$\omega'' = \frac{d^2\omega}{d\phi^2} = \varepsilon^{3/2} S_1 \frac{di^*_{o1/2}}{d\phi} = \varepsilon^3 \kappa_1 \sin 2\omega, \tag{79}$$

from equation (70):

$$\frac{d^2 i^*_{o1/2}}{d\phi^2} = 2\, \varepsilon^{3/2} C^*_2 (\cos 2\omega)\, \omega' \, . \tag{80}$$

Then to compute $\frac{dq_a}{d\phi}$, $\frac{dq_1}{d\phi} = \frac{d^2 u_1}{d\phi^2}$ is required.

From equation (74):

$$\frac{d^2 u_1}{d\phi^2} = \frac{1}{2 A^6} \{ - \frac{e_o^2}{2} (1 - 3 \cos^2 i_{oo}) (\omega'' \sin 2\omega + 2\omega'^2 \cos 2\omega)$$

$$+4 \left(\frac{\sin^2 i_{oo}}{3} - \frac{e_o^2}{3} + \frac{5 e_o^2 \sin^2 i_{oo}}{6}\right) \cos 2\phi$$

$$\frac{-e_o^2}{3} (1 - 9 \cos^2 i_{oo})[2(1 - \omega')^2 \cos 2(\phi - \omega) - \omega'' \sin 2(\phi - \omega)]$$

$$\frac{+e_o}{12} (5 - 11 \cos^2 i_{oo})[(3 - \omega')^2 \cos (3\phi - \omega) - \omega'' \sin (3\phi - \omega)]$$

$$\frac{+e_o^2}{12} (1 - 3 \cos^2 i_{oo})[(4 - 2\omega')^2 \cos (4\phi - 2\omega) - 2\omega'' \sin (4\phi - 2\omega)] \tag{81}$$

Before finding the modified-Encke equations, the quantity $\frac{d}{d\phi}\left(\frac{d\phi}{dt}\right)$ must be developed in an ordered fashion.

From equations (3.5g) and (8):

$$\frac{d\phi}{dt} = \frac{pu^2}{\cos i} + \frac{\cos^3 i \cos \theta}{p \sin^2 i \sin \theta} F \equiv A_1 u^2 + V_o . \tag{82}$$

Differentiating:

$$\frac{d}{d\phi}\left(\frac{d\phi}{dt}\right) = 2A_1uq + u^2\frac{dA_1}{d\phi} + V_1, \tag{83}$$

where

$$V_1 = \frac{\cot^2 i \cot\theta}{A_1}\left[\frac{dF}{d\phi} - F\left(\frac{1}{A_1}\frac{dA_1}{d\phi} + \frac{2}{\sin i\cos i}\frac{di}{d\phi} + \frac{1}{\cos\theta\sin\theta}\frac{d\theta}{d\phi}\right)\right], \tag{84}$$

and from equation (6):

$$\frac{d\theta}{d\phi} = \frac{-1}{\sin\theta}\left[\cos i\sin\phi\frac{di}{d\phi} + \cos\phi\sin i\right], \tag{85}$$

From equation (83):

$$\frac{dA_1}{d\phi} = \frac{d}{d\phi}\left(\frac{p}{\cos i}\right) = \frac{1}{\cos i}\frac{dp}{d\phi} + \frac{\sin i}{\cos^2 i}\frac{di}{d\phi} \tag{86}$$

from equation (12):

$$\frac{dF}{d\phi} = \frac{d}{d\phi}\left\{\frac{1}{u}\left[a_f + \tan i\cos\phi\, a_g\right]\right\} = \frac{-F}{u}\frac{du}{d\phi} + \frac{1}{u}\left[\frac{da_f}{d\phi} + \tan i\cos\phi\frac{da_g}{d\phi}\right.$$

$$\left. + a_g\left(\frac{\cos\phi}{\cos^2 i}\frac{di}{d\phi} - \tan i\sin\phi\right)\right]. \tag{87}$$

Since the a_f and a_g are quite complex for the general problem, the derivatives of a_f and a_g will be approximated for perturbations other than the analytical model by the quantities

$$\frac{da_f}{d\phi} \approx \frac{a_{f_2} - a_{f_1}}{\phi_2 - \phi_1} \text{ and } \frac{da_g}{d\phi} \approx \frac{a_{g_2} - a_{g_1}}{\phi_2 - \phi_1}, \tag{88}$$

where ϕ_2 and ϕ_1 are values of ϕ close to and on each side of the value of ϕ at which the derivative is required. For example, if $\frac{da_f}{d\phi}$ is desired when $\phi = 30°$, take $\phi_2 = 31°$, $\phi_1 = 29°$. Then a_{f_2} and a_{g_2} will be found as a function of ϕ_2, Ω, i, t, and u_2 where Ω, i, and t are the values when $\phi = 30°$, ϕ_2 is $31°$, and u_2 is given by:

$$u_2 \approx u + \frac{du}{d\phi}\,\Delta\phi, \qquad (89)$$

where u is the total reciprocal radius at $\phi = 30°$ and $\frac{du}{d\phi}$ is taken as the q_a at $\phi = 30°$.

3.3 ADDITIONAL DEVELOPMENTS

In addition to the straightforward development to this point, a number of less obvious considerations were necessary before formulation of the computer code. These topics are the elimination of taking differences between two large, nearly equal numbers (with a resultant loss of accuracy) in finding the $\frac{dq_n}{d\phi}$ equations, the orientation of the rotating earth beneath the satellite, the treatment of the time, the formulation of the disturbances from the complete potential, the formulation of the disturbances due to luni-solar effects, and the development of the Runge-Kutta formulation.

3.3.1 Elimination of Large Quantities from the Encke Equation for q_n

Substituting the expressions for $\frac{d\phi}{dt}$ and $\frac{d}{d\phi}\left(\frac{d\phi}{dt}\right)$ from equations (82) and (83) into the differential equation for q in equation (14), and multiplying by $\left(\frac{d\phi}{dt}\right)^2$ yields:

$$\left(\frac{dq}{d\phi} - \frac{2}{u}\,q^2\right)\left[A_1{}^2u^4 + 2A_1u^2\,V_o + V_o{}^2\right] + q\left(A_1u^2 + V_o\right)\left[2A_1uq + \frac{dA_1}{d\phi}\,u^2 + V_1\right]$$

$$= -A_1{}^2u^5 + \left(u^2 + U_1\right)\,u^2 \qquad (90)$$

where

$$U_1 = -\left(u^2 + \frac{\partial U}{\partial r}\right) = \varepsilon u^4\left(1 - 3\cos^2\theta\right) + c\varepsilon^2u^6\left(35\cos^4\theta - 30\cos^2\theta + 3\right) + a_r \qquad (91)$$

(a_r represents accelerations in the outward radial direction in addition to those given in the analytical model).

Using the abbreviation:

$$V_3 = -V_o \frac{dq}{d\phi} \left[\frac{2}{A_1 u^2} + \frac{V_o}{A_1^2 u^4}\right] + \left[2 \frac{V_o}{A_1 u^3} q^2 - \frac{V_1 q}{A_1 u^2} - q \frac{dA_1}{d\phi} \frac{1}{A^1}\right]\left(1 + \frac{V_o}{A_1 u^2}\right) + \frac{u_2 U_1}{A_1^2 u^4} \tag{92}$$

one obtains after dividing equation (90) by $A_1^2 u^4$:

$$\frac{dq}{d\phi} = -u + \frac{1}{A_1^2} + V_3. \tag{93}$$

Using the expression for $\frac{dq_o}{d\phi}$ from equation (76) and adding and subtracting $\frac{dq_o}{d\phi}$ gives:

$$\frac{dq_n}{d\phi} \equiv \frac{dq}{d\phi} - \frac{dq_o}{d\phi} + \frac{dq_o}{d\phi} - \frac{dq_a}{d\phi} = -u + u_o + \frac{\cos^2 i}{p^2} - \frac{\cos^2 i_{oc}}{p_a^2} + V_3 - H$$
$$+ \frac{dq_o}{d\phi} - \frac{dq_a}{d\phi}. \tag{94}$$

Note that

$$\frac{1}{p^2} - \frac{1}{p_a'^2} = \frac{p_a + p}{p^2 p_a^2} (p_a - p) = - \frac{p_n}{p^2 p_a^2} (p_a + p) \tag{95}$$

and from reference 4, equation (401.13):

$$\cos^2 i - \cos^2 i_{oc} = -\sin (i + i_{oc}) \sin (i - i_{oc})$$
$$= -\sin (i + i_{oc}) \sin (i_n + \varepsilon i_1). \tag{96}$$

After substitution, the result is then:

$$\frac{dq_n}{d\phi} = -u_n - \varepsilon u_1 + \frac{\cos^2 i}{p^2} - \frac{\cos^2 i_{oc}}{p^2} + \frac{\cos^2 i_{oc}}{p^2} - \frac{\cos^2 i_{oc}}{p_a^2}$$

$$+ V_3 - H - \varepsilon \frac{dq_1}{d\phi}$$

$$= -u_n - \varepsilon u_1 - \frac{1}{p^2} \sin(i + i_{oc}) \sin(i_n + \varepsilon i_1)$$

$$-\cos^2 i_{oc} \frac{p_a + p}{p_a^2 p^2} p_n + V_3 - H - \varepsilon \frac{dq_1}{d\phi} \tag{97}$$

$$\frac{du_n}{d\phi} = q_n \tag{98}$$

All terms occurring in these equations are numerically small. However, this is not the completed form, since V_3 contains the term $\frac{dq}{d\phi}$.

From equation (92), let

$$V_3 \equiv - \frac{V_o}{A_1 u^2} \frac{dq_n}{d\phi} (2 + \frac{V_o}{A_1 u^2}) + V_3' \tag{99}$$

Then

$$V_3' \equiv - \frac{V_o}{A_1 u^2} \frac{dq_a}{d\phi} (2 + \frac{V_o}{A_1 u^2}) + q(1 + \frac{V_o}{A_1 u^2})[\frac{2V_o q}{A_1 u^3}$$

$$- \frac{V_1}{A_1 u^z} - \frac{1}{A_1} \frac{dA_1}{d\phi}] + \frac{U_1}{A_1^2 u^2}. \tag{100}$$

Note that $U_1 = -a_h$ if the coefficient of the leading term of the potential is zeroed. From these two equations and equation (97), the final form for dq_n is:

$$\frac{dq_n}{d\phi} = [-u_n - \varepsilon u_1 - \frac{1}{p^2} \sin(i + i_{oc}) \sin(i_n + \varepsilon i_1)$$

$$-\cos^2 i_{oc} \frac{p_a + p}{p_a^2 \, p^2} p_n + V_3' - H - \varepsilon \frac{dq_1}{d\phi}] \cdot (1 + \frac{V_o}{A_1 u^2})^{-2} \qquad (101)$$

3.3.2 Orientation of the Earth Beneath the Satellite

To find the effects of the tesseral and sectorial harmonics of the potential, the longitude of the satellite above the rotating earth must be known. Denoting the east longitude of the satellite as λ:

$$\lambda = \Omega + b - \lambda_G, \qquad (102)$$

where

$$\lambda_G = \lambda_{oG} + \omega_E (t - t_o) \qquad (103)$$

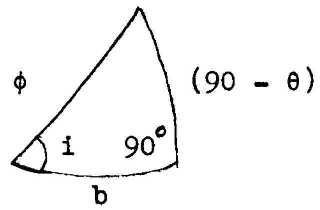
and

λ_{oG} = longitude of Greenwich measured from the equinox of 1950 at t_o.

ω_E = mean rational rate of the earth

Ω = longitude of the node measured from the equinox of 1950.

b is given by the following sketch



From spherical trigonometry:

$$\cos b = \frac{\cos \phi}{\sin \theta} \tag{104}$$

and b is in the same quadrant as ϕ except when $i = 90°$. In that case $b = 0$ always.

3.3.3 Treatment of Time

The only reason that time is calculated in this program is to find the orientation of the rotating earth and the location of the sun and moon. Therefore, extreme accuracy in the time is not required. A simple approximate solution which includes the predominant effects will be used so that the numerical differential equation for time will be of the same order as those for the other parameters. This is done to keep the computing interval as large as possible for the complete system of equations. Since great accuracy is not necessary, no attempt will be made to analytically cancel the terms of order unity in the numerical differential equation.

The approximate solution chosen for the time is:

$$t_a = \frac{p^3}{\cos^3 i_{oo} (1 - e_o^2)} \left\{ \frac{-e_o \sin (\phi - \omega)}{(1 + e_o \cos(\phi - \omega)} + \frac{2}{\sqrt{1 - e_o^2}} \tan^{-1} \left[\frac{\sqrt{1 - e_o^2}}{1 + e_o} \tan \frac{(\phi - \omega)}{2} \right] \right\} + t_{ol} \tag{105}$$

where the $\tan^{-1}$ [ ] is in the same quadrant as $\frac{(\phi - \omega)}{2}$.

The derivative of the approximate solution is simply:

$$\frac{dt_a}{d\phi} = \frac{p^3 (1 - \omega')}{\cos^3 i_{oo} [1 + e_o \cos (\phi - \omega)]^2}. \tag{106}$$

Now t_{ol} is given by:

$$t_{ol} = \frac{p^3}{\cos^3 i_{oo}(1 - e_o^2)} \{\frac{e_o \sin(\phi_i - w)}{1 + e_o \cos(\phi_i - w)}$$

$$\frac{-2}{\sqrt{1 - e_o^2}} \tan^{-1} [\frac{\sqrt{1 - e_o^2}}{1 + e_o} \tan (\frac{\phi_i - w}{2})]\} + t_o \quad (107)$$

3.3.4 Development of the Perturbative Accelerations Due to the Complete Potential

This description determines the perturbative gravitational acceleration of a spacecraft by means of the zonal, sectorial, and tesseral harmonic equations found in reference 2 (pages 4-97, 4-98). These equations are as follows:

$$a_f = \cos\phi \sum_{n=2}^{N1} (J_n r^{-n-2})\rho_n' + \sum_{m=2}^{N2} m r^{-m-2} \sin\phi\, (\sec\phi\, \rho_m^m)(C_{mm} \cos m\lambda + S_{mm} \sin m\lambda)$$

$$- \sum_{m=1}^{N3} \sum_{n=m+1}^{N3} r^{-n-2} (\cos\phi\, \rho_n^{m'})\, (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \quad (108)$$

$$a_g = - \sum_{m=2}^{N2} m r^{-m-2} (\sec\phi\, \rho_m^m)\, (C_{mm} \sin m\lambda - S_{mm} \cos m\lambda)$$

$$- \sum_{m=1}^{N3} m \sum_{n=m+1}^{N3} r^{-n-2} (\sec\phi\, \rho_n^m)\, (C_{nm} \sin m\lambda - S_{nm} \cos m\lambda) \quad (109)$$

$$a_h = \sum_{n=2}^{N1} (n+1)(J_n r^{-n-2})\rho_n - \cos\phi[\sum_{m=2}^{N2} (m+1) r^{-m-2}(\sec\phi\,\rho_m^m)(C_{mm}\cos m\lambda + S_{mm}\sin m\lambda)$$

$$+ \sum_{m=1}^{N1} \sum_{n=m+1}^{N3} (n+1) r^{-n-2} (\sec\phi\,\rho_n^m)(C_{nm}\cos m\lambda + S_{nm}\sin m\lambda)] \qquad (110)$$

where:

$$\rho_n = [(2n-1)\sin\phi\ \rho_{n-1} - (n-1)\rho_{n-2}]/n$$

$$\rho_o = 1 \qquad (111)$$

$$\rho_1 = \sin\phi$$

$$\rho_n' = \sin\phi\,\rho'_{n-1} + n\rho_{n-1}$$

$$\rho_1' = 1 \qquad (112)$$

$$(\sec\phi\,\rho_m^m) = (2m-1)\cos\phi\,(\sec\phi\,\rho_{m-1}^{m-1})$$

$$(\sec\phi\,\rho_1^1) = 1$$

$$(\sec\phi\,\rho_n^m) = [(2n-1)\sin\phi\,(\sec\phi\,\rho_{n-1}^m) - (n+m-1)(\sec\phi\,\rho_{n-2}^m)]/(n-m)$$

$$(\sec\phi\,\rho_{m-1}^m) = 0 \qquad (113)$$

$$(\cos\phi\,\rho_m^{m'}) = -m\sin\phi\,(\sec\phi\,\rho_m^m)$$

$$(\cos\phi\,\rho_n^{m'}) = -n\sin\phi\,(\sec\phi\,\rho_n^m) + (n+m)(\sec\phi\,\rho_{n-1}^m) \qquad (114)$$

It is noted that the components of the acceleration are non-dimensional and in a local rectangular system (f, g, h) with h along the outward geocentric vertical, f directed south, and g directed east. Also, the recursion equations may be recognized as the Legendre polynomials, the rhos being the zonal set, and the secant rho and cosine rho comprising the sectorial and tesseral set.

The equations may be written in a more convenient form by substituting U_{nm} for $(\sec\phi\ \rho_n^m)$, W_{nm} for $(\cos\phi\ \rho_n^{m'})$, and V_{mm} for $(\sec\phi\ \rho_m^m)$; also $m \sin\phi\ (\sec\phi\ \rho_m^m)$ may be replaced by $-(\cos\phi\ \rho_m^{m'})$ in the sectorial term of a_f. Finally, if the degree of the highest sectorial harmonic (N2) is taken equal to the degree of the highest tesseral harmonic (N3), the sectorial and tesseral terms may be combined with the summation scheme being set at: $\sum_{n=2}^{N2} \sum_{m=1}^{n}$. The equations may then be written:

$$a_f = \cos\phi \sum_{n=2}^{N1} \left(\frac{J_n}{r^{n+2}}\right)\rho_n' - \sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{1}{r^{n+2}} W_{nm} (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \quad (115)$$

$$a_g = -\sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{m}{r^{n+2}} U_{nm} (C_{nm} \sin m\lambda - S_{nm} \cos m\lambda) \quad (116)$$

$$a_h = \sum_{n=2}^{N1} (n + 1) \left(\frac{J_n}{r^{n+2}}\right) \rho_n - \cos\phi \sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{n+1}{r^{n+2}} U_{nm} (C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \quad (117)$$

where the ρ_n's and ρ_n''s are given in equations (111) and (112) and:

$$U_{mm} = (2m - 1) \cos\phi\ U_{m-1,\ m-1}$$

$$U_{11} = 1 \quad (118)$$

$$U_{nm} = \frac{1}{n-m} [(2n - 1) \sin \phi \, U_{n-1,m} - (n + m - 1) \, U_{n-2,m}]$$

$$U_{m-1,m} = 0 \tag{119}$$

$$W_{mm} = - m \sin \phi \, U_{mm}$$

$$W_{nm} = - n \sin \phi \, U_{nm} + (n + m) \, U_{n-1,m}$$

3.3.5 Development of Luni-Solar Perturbations

The most difficult part of obtaining luni-solar perturbations would normally be encountered in obtaining the relative positions of the earth, moon, sun, and satellite at any particular time. This problem has been circumvented by utilizing the JPL Ephemeris Tapes and their associated tape-reading routines to determine the positions of the earth, moon, and sun. These routines are described in detail in reference 5, and will not be discussed here.

The remaining problem is that of expressing the perturbative accelerations in the a_f, a_g, a_h reference frame adopted for the earth potential perturbations.

3.3.6 Development of Runge-Kutta Equations and Self-Computing Interval Scheme

The Runge-Kutta method is used for the numerical solution of the differential equations. The method is a simple extension of the methods for second-order and first-order simultaneous equations given by Hildebrand (ref. 6, page 237) which are:

Given the simultaneous first-order equations:

(6.16.7)* $$\frac{dy}{dx} = F(x,y,u),$$

$$\frac{du}{dx} = G(x,y,u) \tag{120}$$

* These numbers are equation numbers from Hildebrand.

the solution may be written as:

(6.16.8)
$$y_{n+1} = y_n + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3) + O(h^5)$$
$$u_{n+1} = u_n + \frac{1}{6}(m_0 + 2m_1 + 2m_2 + m_3) + O(h^5) \qquad (121)$$

where

(6.16.9)
$$k_0 = hF(x_n, y_n, u_n),$$
$$k_1 = hF(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_0,\ u_n + \frac{1}{2}m_0), \qquad (122)$$
$$k_2 = hF(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_1,\ u_n + \frac{1}{2}m_1),$$
$$k_3 = hF(x_n + h,\ y_n + k_2,\ u_n + m_2),$$

and

(6.16.10)
$$m_0 = hG(x_n, y_n, u_n).$$
$$m_1 = hG(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_0,\ u_n + \frac{1}{2}m_0), \qquad (123)$$
$$m_2 = hG(x_n + \frac{1}{2}h,\ y_n + \frac{1}{2}k_1,\ u_n + \frac{1}{2}m_1),$$
$$m_3 = hG(x_n + h,\ y_n + k_2,\ u_n + m_2).$$

Given the second-order equation:

(6.16.11)
$$\frac{d^2y}{dx^2} = G(x, y, y'), \qquad (124)$$

The above equation can be written as two simultaneous first-order differential equations as:

$$\frac{dy}{dx} = u \qquad (125)$$

and

$$\frac{du}{dx} = G(x, y, u) \qquad (126)$$

Then equation (6.16.9) gives:

$$k_0 = hy'_n, \quad k_1 = hy'_n + \frac{h}{2} m_0, \quad k_2 = hy'_n + \frac{h}{2} m_1, \quad k_3 = hy'_n + hm_2,$$

and hence equations (6.16.8) and (6.16.10) give:

(6.16.12)

$$y_{n+1} = y_n + hy'_n + \frac{h}{6}(m_0 + m_1 + m_2) + O(h^5),$$

$$y'_{n+1} = y'_n + \frac{1}{6}(m_0 + 2m_1 + 2m_2 + m_3) + O(h^5), \qquad (127)$$

where

(6.16.13)

$$m_0 = hG(x_n, y_n, y'_n),$$

$$m_1 = hG(x_n + \frac{1}{2} h,\ y_n + \frac{1}{2} hy'_n,\ y'_n + \frac{1}{2} m_0),$$

$$m_2 = hG(x_n + \frac{1}{2} h,\ y_n + \frac{1}{2} hy'_n + \frac{1}{4} hm_0,\ y'_n + \frac{1}{2} m_1), \qquad (128)$$

$$m_3 = hG(x_n + h,\ y_n + hy'_n + \frac{1}{2} hm_1,\ y'_n + m_2).$$

After integration over two intervals of equal size, the results for the velocity components are compared with an integration over the same intervals using Simpson's rule which is also of fourth order accuracy. Simpson's rule is given on page 73 of reference 6 as:

$$\int_{x_o}^{x_2} f(x)dx = \frac{h}{3}(f_o + 4f_1 + f_2) - \frac{h^5 f^{IV}(\xi)}{90} \qquad (129)$$

where $x_o < \xi < x_2$,

$$f_o = f(x_o), \quad f_1 = f(x_o + \frac{h}{2}), \quad \text{and} \quad f_2 = f(x_2)$$

By virtue of the comparison between the two integrated results, decisions are made by the program concerning the accuracy of the integration, and the computing interval for the next two intervals is chosen. The logic underlying these program decisions will now be explained using one first-order differential equation as an example.

Let the differential equation to be solved be of the form:

$$\dot{x} = \dot{x}(t, x) \tag{130}$$

If this equation is integrated over an interval, h, by Runge-Kutta methods of fourth order, then the numerical value of that function corresponds to a Taylor series expansion with an error term of $O(h^5)$, i.e.:

$$x_{n+1} = x_n + \dot{x}_n h + \ddot{x}_n \frac{h^2}{2!} + \frac{\dddot{x}\, h^3}{3!} + \frac{x^{IV} h^4}{4!} + O(h^5) \tag{131}$$

The complete functional form of the coefficient of the error term is unknown, but it is known to contain x^V. For the purposes of this program, the coefficient of the fifth order term is assumed to be the next term in the Taylor series $\frac{x}{5!}$ and x^V is assumed to be a slowly varying function. The coefficient of the fifth order term in Simpson's rule is known to be $-\frac{x^V}{90}$. Thus, if we let x_c be the correct value of x at the end of the two equal intervals, and let x_{RK} and x_{SR} be the Runge-Kutta and Simpson's rule integrated values respectively, we may write:

$$x_c = x_{RK} + 2\left(\frac{x^V h^5}{5!}\right) \tag{132}$$

$$x_c = x_{SR} - \frac{x^V h^5}{90} \tag{133}$$

Eliminating x_c between these two equations and solving for x^V results in:

$$x^V = \frac{36(x_{SR} - x_{RK})}{h^5} \tag{134}$$

From equations (132) through (134) the error in the Runge-Kutta solution is estimated to be:

$$\delta x = \frac{3}{5}(x_{SR} - x_{RK}) \tag{135}$$

A factor of $\frac{3}{5}$ is dropped in the use of this equation because an arbitrary constant is introduced at this point.

Letting $\Delta\dot{x}$, $\Delta\dot{y}$, $\Delta\dot{z}$ be the changes in the $\dot{x}$, $\dot{y}$, $\dot{z}$ values over the double interval, then what is required in the program is that:

$$E = \text{maximum}\ (|\delta\dot{x}|, |\delta\dot{y}|, |\delta\dot{z}|) < E_{all} = \text{maximum}\ (|W_8|C_{max}), \tag{136}$$

$$10^{-9}\ \text{maximum}\ (|\dot{x}|, |\dot{y}|, |\dot{z}|)$$

where

$$C_{max} = \text{maximum}\ (|\Delta\dot{x}|, |\Delta\dot{y}|, |\Delta\dot{z}|) \tag{137}$$

and W_8 is an input number designed to require a series truncation greater than number truncation but as small as possible. An error which is less than 10^{-9} of the maximum of the absolute values of $\dot{x}$, $\dot{y}$, and $\dot{z}$ is always acceptable since it will be lost in the first addition anyway because of the limits of machine word length.

If $E \leq E_{all}$, the computation proceeds. If $E > E_{all}$, the last two steps are done over.

If E is greater than an input minimum error $E_{min} \cdot C_{max}$, then Δt is computed by:

$$\Delta t_{new} = FDT \bullet \Delta t_{old} \left(\frac{E_{all}}{E}\right)^{.25} \tag{138}$$

If it assumed that $E_{all} = W_8 C_{max} = K\Delta t$ where K is some constant (since x is roughly proportional to Δt and C_{max} is normally proportional to Δx)

and FDT = 1, then by equations (132), (135), and (136), Δt_{new} would result in an error of exactly E_{all}. FDT is an input number < 1 to prevent Δt_{new} from resulting in an error $E > E_{all}$ due to number truncation or changes of x^V over the two new intervals as compared to the x^V of the previous two intervals.

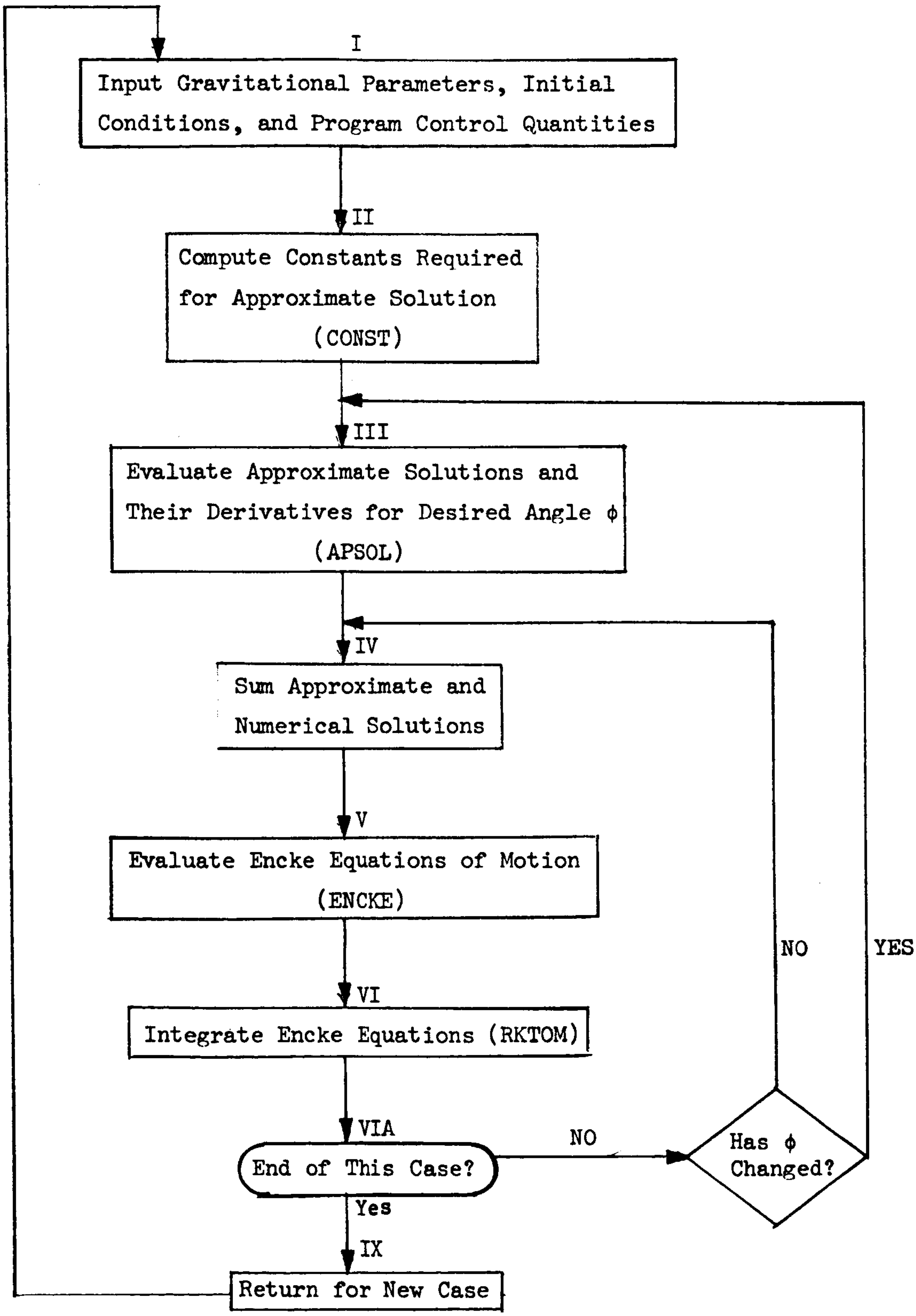
If $E < E_{min} \cdot C_{max}$, then it is assumed that the error in x_{RK} is primarily due to number truncation in the computations. In this case equation (132) does not apply. The new computing interval is then computed by:

$$\Delta t_{new} = \Delta t_{min} \cdot \Delta t_{old} \tag{139}$$

where Δt_{min} is an input quantity > 1.

Section 4

BLOCK FLOW CHART



Section 5

EQUATIONS IN ORDER OF SOLUTION, DETAIL FLOW CHARTS, AND PROGRAM LISTING

5.0 GENERAL

In the following sections, certain equations will not be repeated in the equations in order of solution due to their length. In this event, the equation number for the expression in Section 3, Sources of Equations, will be given. This avoids lengthy repetition and also links Sections 3 and 5.

In order to relate the FORTRAN coding and the analytical formalism, both the FORTRAN variable name and the equivalent algebraic expression are presented throughout Section 5. In some cases indices of an array are also used in the FORTRAN style using parentheses, i.e. $A(1) = A_1$, $A(I) = A_i$, etc.

5.1 MAIN PROGRAM

The main program serves mainly as a control program to convert input quantities to non-dimensional units for internal purposes, to control the flow to and from subroutines, and to print the results in the desired dimensions.

5.1.1 Equations in Order of Solution

I. Call Input Data, Store, and Modify for Internal Computation.

A. Non-dimensionalize input quantities and store necessary constants. Start clock to time the case (call TIKTOK). Read input array into storage using INPUT 1 routine with reference run capability. Store J_2 and J_4 in AJ2 and AJ4 for permanent use. Store non-dimensional values of w, ϕ_T, i_{oo}, L_o, ϕ_{stop}, $\Delta\phi$, longitude of Greenwich, t_o and ω_E = EROT. Compute

$$c = -\frac{J_4}{J_2^2}\left(\frac{5}{18}\right)$$

Compute

$$\epsilon^{1/2} = \sqrt{\frac{3}{2} J_2}$$

Place the coefficients of the potential in common by filling the array COEFF. Set the initial conditions of the numerical solution equal to zero. (HAH(i) = 0, i = 1, 6). Save the initial time in DTSAVE.

B. Print the input array and the format heading for the regular output during operation.

C. Set initial values of flags.

Set Runge-Kutta flag = 1 for first cycle of Runge-Kutta.

Set IP = 1 for the first point of the trajectory.

Set KHALT = 1 to show no halt.

Set IPRINT = 2 to initialize print flag.

II. Compute the Constants Required for the Approximate Solution and Its Derivatives.

Store the computed constants in labeled common /CON/ by calling subroutine CONST. Compute

$$\bar{\phi}_i = \varepsilon^{3/2} \phi_i,$$

$$\text{DENK}(1) = 2\varepsilon,$$

$$\text{DENK}(2) = 2\varepsilon c,$$

$$\text{DENK}(3) = 2\varepsilon^2 c$$

III. Evaluate the Approximate Solutions and Their Derivatives for the Current Value of ϕ.

Store the approximate solutions as the array AS(6) and the derivatives of the approximate solutions as the array AD(6) in labeled common /APS/ by calling subroutine APSOL.

IV. Sum Numerical and Approximate Solutions and Find the Change in Time.

Store the sums of the numerical and approximate solutions for the six dependent variables in the array DVT(6).

Find

$$DT = t_T - DTSAVE$$

Save the total time in DTSAVE.

V. Evaluate the Encke Equations of Motion and Test for Print

Store the values of the differential equations of motion in the array ENK(6) by calling subroutine ENCKE.

A. If the Runge-Kutta flag (KR) is 1, go to VB, otherwise, call the Runge-Kutta routine at VI.

B. If the halt flag (KHALT) is 3, go to VC for print computations and print, otherwise call the Runge-Kutta routine at VI.

C. Compute ϕ_T, i_T, Ω_T, in degrees. Compute t_T in hours and r_T in kilometers.

D. Check if energy print is desired.

If KDER is 2, go to VF; otherwise go to VE.

E. Calculate the total energy and print.

$$\text{TOTE} = -u\left(2 - \frac{p^2 u}{\cos^2 i}\right) - \frac{2}{3}\varepsilon u^3 (1 - 3\cos^2\theta)$$
$$-c\varepsilon^2 u^5 [\cos^2\theta(14\cos^2\theta - 12) + 1.2] + \left[\frac{q}{u^2}\frac{d\phi}{dt}\right]^2$$

Print in three rows of six columns the approximate solutions AS(6), the numerical solutions HAH(6), the ϕ_{TOTAL} (deg), time (hours), radius (km), Ω (deg), i (deg), total energy (non-dimensional), e_a, and ω_a (non-dimensional). Go to VG.

F. Print in three rows of six columns, the approximate solutions AS(6), the numerical solutions HAH(6), and the dimensional values of ϕ_T, t_T, r_T, Ω_T, i_T, and the values e_a, and ω_a (non-dimensional).

G. Test halt flag.

If the halt flag (KHALT) is 1 or 2, go to VI.

If the halt flag is 3, go to I to start a new case.

H. Test print flag.

If the print flag (IPRINT) is 1, set it equal to 2 and proceed to VC for print computation.

If the print flag (IPRINT) is 2, set it equal to 1 and proceed as in VG.

VI. Call Runge-Kutta Routine and Test for Direction after Exit.

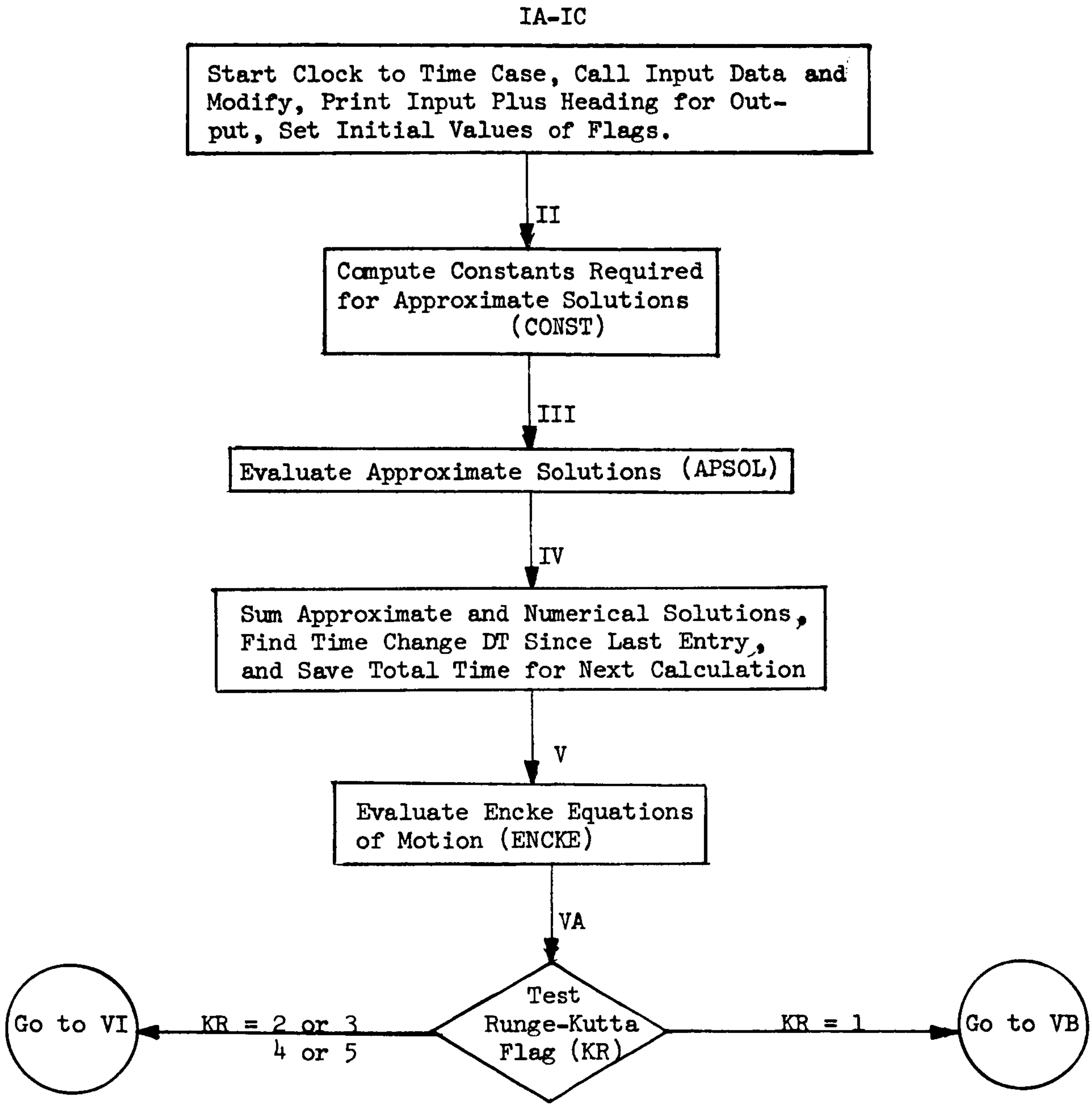
Find new values of the numerical solution and the independent variable, ϕ, by calling the Runge-Kutta routine RKTOM.

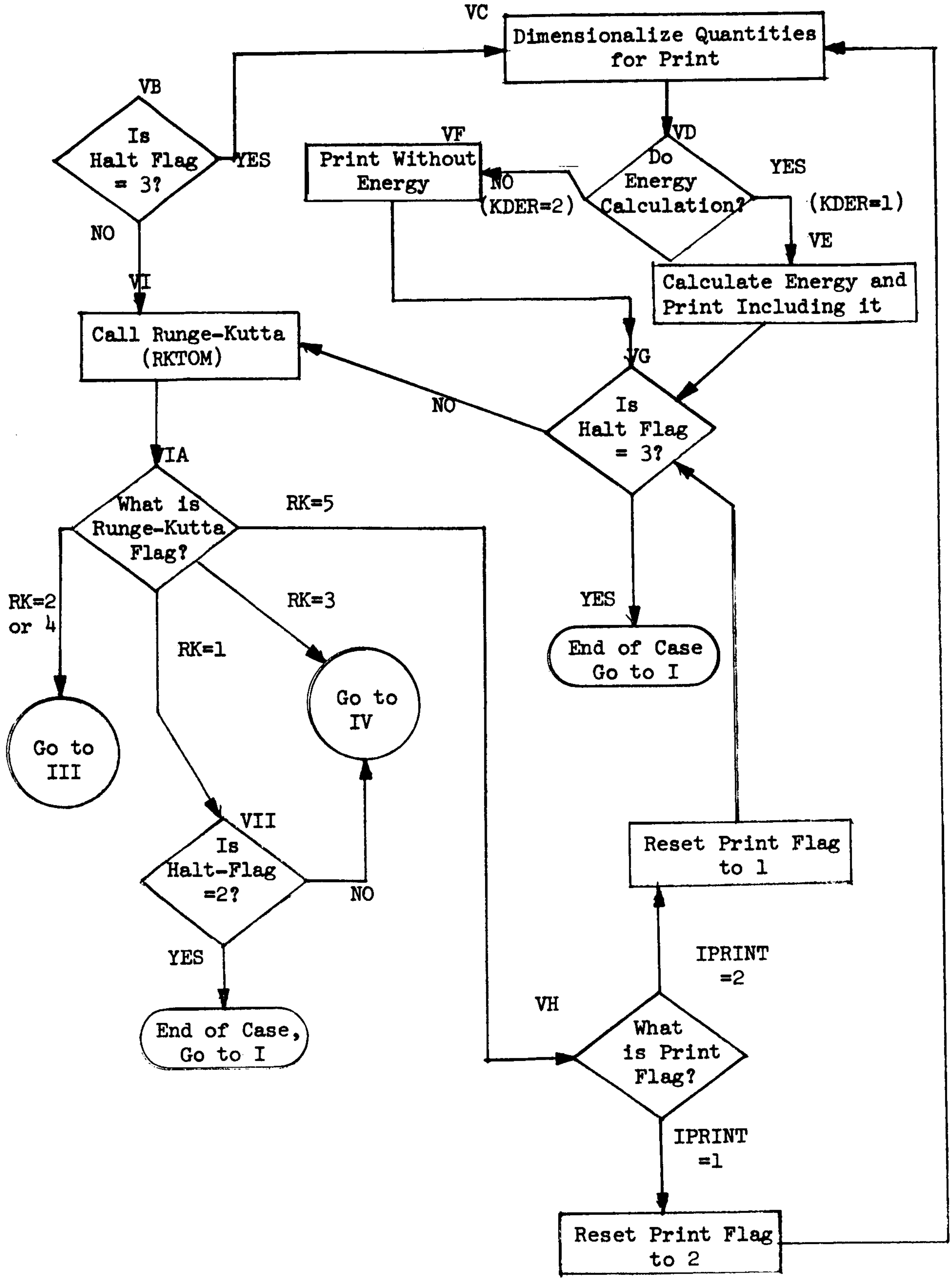
After exit:

A. If the Runge-Kutta flag (KR) is 1, go to VII.
   If the Runge-Kutta flag (KR) is 2 or 4, go to III.
   If the Runge-Kutta flag (KR) is 3, go to IV.
   If the Runge-Kutta flag (KR) is 5, go to VH.

VII. If the Halt Flag (KHALT) is 2, start a new case by going to I. Otherwise, continue by going to IV.

5.1.2 <u>Detail Flow Chart</u>




VC
Dimensionalize Quantities for Print
VB
Is Halt Flag = 3?
YES
NO
VF
Print Without Energy
VD
Do Energy Calculation?
NO (KDER=2)
YES (KDER=1)
VE
Calculate Energy and Print Including it
VI
Call Runge-Kutta (RKTOM)
VG
Is Halt Flag = 3?
NO
YES
End of Case Go to I
IA
What is Runge-Kutta Flag?
RK=5
RK=2 or 4
RK=3
RK=1
Go to III
Go to IV
VII
Is Halt-Flag =2?
NO
YES
End of Case, Go to I
Reset Print Flag to 1
IPRINT =2
VH
What is Print Flag?
IPRINT =1
Reset Print Flag to 2

Section 5.1.3 Program Listing

The following pages give the listing of the MAIN program.

```
C PROGRAM TO COMPUTE SATELLITE MOTION ABOUT A NON-SPHERICAL CENTRAL
C BODY INCLUDING LUNI-SOLAR PERTURBATIONS.  MODIFIED ENCKE APPROACH
C USING KEVORKIAN OBLATE PLANET SOLUTION AS THE REFERENCE ORBIT.
      DIMENSION Z(125),RR(125),FNK(6),HAH(12),DVT(6)
      EQUIVALENCE (Z(84),P),(Z(85),E),(Z(93),MFAIL),(Z(94),FMAX),
     1(Z(95),EMIN),(Z(96),DTM),(Z(99),LS),(Z(100),K1OR3),(ENK,HAH(7))
     2,(Z(101),FDT),(Z(82),N1),(Z(83),N2) ,(Z(102),KDFR)
      COMMON  /CON/ CI,CI2,CI4,SI,SI2,CS,TI,E2,E2C,EM2,P4,AB
     1,CRD,C2W,EPS32,EPS2,EPS3,CI31,EO2,EO6,EO3,AM4,AC,ACS,A
     2 CS32,AC32,G0,G1,G2,G3,G4,G5,C2SP,C2S,B2SP,B2S,S1P,S1,
     3SOB,GAP1P,GAP1,GAPOB,C2E,B2E,B2E2,E3K,C22E,PI,TWOPI,
     4PIO2,IC,RK,XMOD,AMP,CW,QPER,PHIO,TW2,RMK,IE,GAM1,GOK,
     5SOK,OSK2,OSK,E12,S1S,SOBS ,ACSS ,C4E,A3,EPS
     6,ACS2,EM22,EM212,EF,A3F,T01,P2,ACC,A6,XW,IWC,WO2
      COMMON/CPOT/COEFF(81),N1,N2/APS/AS(6),REST(12)
      COMMON/EX/QQ(3),EWOG,FPOT,PP(3)
      COMMON/ENERG/EE(6 ) /DERIV/DENK(3)
C I CALL INPUT DATA, STORE AND MODIFY FOR INTERNAL COMPUTATIONS
C IA NON DIM INPUT QUANTITIES AND FORM NECESSARY CONSTS.
    1 CALL TIKTOK
        CALL INPUT1(Z,Z(125),RR)
      AJ2=Z(2)
      AJ4=Z(4)
      RAD=.17453293 E-01
      W=Z(86)*RAD
      PHI=Z(88)*RAD
      PHIT=PHI
      XI=Z(89)*RAD
      XLO=Z(90)*RAD
      PHISTP=Z(91)*RAD
      DELPHI=Z(92)*RAD
      EWOG=Z(97)*RAD
      TO=Z(87)/.22411493
       EROT=Z(98)*.22411493
      C=-.27777777*Z(4)/(Z(2)*Z(2))
      EPS12=SQRT(1.5*Z(2))
      DO 10 I=1,81
   10 COEFF(I)=Z(I)
      DO 21 I=1,6
   21 HAH(I)=0.0
      DTSAVE=TO
C IB PRINT INPUT ARRAY AND OUTPUT HEADING
      WRITE (6,11)Z
   11 FORMAT(5E19.8)
      WRITE(6,12)
   12 FORMAT( 21H        OUTPUT  FORMAT //
     140H  APPROX. SOLUTIONS  AS(6)  (NON-DIM)              /
     240H  NUMERICAL SOLUTIONS HAH(6)(NON-DIM)              /
     348H TOTAL PHI(DEG) T(HRS) R(KM) NODE(DEG) INC(DEG)    )
C IC SET INITIAL VALUES OF FLAGS
   20 KR=1
```

```
      IP=1
      KHALT=1
      IPRINT=2
C II COMPUTE CONSTANTS REQRD. FOR APPROX. SOL. AND DERIVATIVES
      CALL CONST(E,P,XI,C,EPS12,W,PHI,TO,LS)
      PHIIB=PHI*EPS32
      DENK(1)= 2.*EPS
      DENK(2)=DENK(1)*C
      DENK(3)= DENK(2)*EPS
C III EVALUATE  APPROX. SOLS. AND THEIR DERIVATIVES FOR PHI
   30 CALL APSOL(PHI,PHIT,IP,  XI,XLO,TO,W,E,K1OR3,PHIIB,EPS12)
C IV SUM NUM. AND APPROX. SOLS. AND FIND DT
   40 DO 41 I=1,6
   41 DVT(I)=HAH(I)+AS(I)
      DT=DVT(6)-DTSAVE
      DTSAVE=DVT(6)
C V EVALUATE ENCKE EQS. OF MOTION
   50 CALL ENCKE(DVT(1),DVT(2),DVT(3),DVT(4),DVT(5),DVT(6),
     1 PHI,LS,DT,N2,HAH(3),HAH(1),HAH(5),P2,P,HAH(4),ENK,AJ2 ,AJ4 ,KDER)
C VA TEST RUNGE-KUTTA FLAG
      GO TO(51,60,60,60,60),KR
C VB TEST HALT FLAG
   51 GO TO (60,60,52),KHALT
C VH  CHECK PRINT FLAG
   54 GO TO(55,56),IPRINT
   55 IPRINT=2
C VC  COMPUTATION FOR PRINT AND PRINT
C   CONVERT TO DIMENSIONAL QUANTITIES
   52 PHITD=PHIT/RAD
      XITD=DVT(3)/RAD
      OTD=DVT(2)/RAD
      TD=DVT(6)*.22411493
      RD=   6378.1521/DVT(5)
CVD   CHECK IF ENERGY PRINT IS DESIRED
      GO  TO (57,58),KDER
C VE  CALCULATE ENERGY AND PRINT INCLUDING ENERGY
   57 TOTE=-DVT(5)*(2.-P2*DVT(5)/EE(1))-.66666667*EPS*
     1EE(2)*(1.-3.*EE(3))-EPS2*C*EE(4)*(EE(3)*(14.*EE(3)-12.
     2)+1.2)+(DVT(4)*EE(5)/EE(6))**2
      WRITE(6,53) AS,(HAH(I),I=1,6),PHITD,TD,RD,OTD,XITD ,TOTE
     1,REST(12),XW
      GO TO 59
   53 FORMAT(6E15.8)
C VF  PRINT WITHOUT ENERGY
   58 WRITE(6,53) AS,(HAH(I),I=1,6),PHITD,TD,RD,OTD,XITD,REST(12),XW
CVG   TEST HALT FLAG
   59 GO TO (60,60,1),KHALT
C   VI  RUNGE-KUTTA
   60 CALL RKTOM(KR,IP,KHALT,PHISTP,HAH,EMIN,EMAX,MFAIL,FDT,DTM,DELPHI,
     1PHIT,   PHI)
C   TEST RUNGE-KUTTA FLAG
```

```
      GO TO (70,30,40,30,54),KR
C VII TEST HALT FLAG
   70 GO TO (40,1,40),KHALT
   56 IPRINT=1
      GO TO (60,60,1),KHALT
      END
```

5.2 SUBROUTINE CONST (E, P, XI, C, EPS12, W, PHI, TO, LS)

Calculates constants which depend only on initial conditions and stores them in labeled common /CON/. Inputs are e_o, p, i_{oo}, c, $\varepsilon^{1/2}$, w*, ϕ_o, t_o, and LS.

5.2.1 Equations in Order of Solution

I. Calculate Combinations of Constants Needed Frequently.

A. $P4 = p^4$

$CI = \cos i_{oo}$

$SI = \sin i_{oo}$

$$AM4 = \frac{\cos^4 i_{oo}}{p^4}$$

$CI2 = \cos^2 i_{oo}$

$CI4 = \cos^4 i_{oo}$

$$AB = \frac{\cos^8 i_{oo}}{2p^8}$$

$SI2 = \sin^2 i_{oo}$

$E2 = e_o^2 c$

$CS = \cos i_{oo} \sin i_{oo}$

$TI = \tan i_{oo}$

$CRD = 1 - 5 \cos^2 i_{oo}$

$EM2 = 1 - e_o^2$

$C2W = \cos 2w$

$EPS = \varepsilon$

$EPS32 = \varepsilon^{3/2}$

$EPS2 = \varepsilon^2$

$EPS3 = \varepsilon^3$

$PHIB = \bar{\phi}_i$

$C131 = 1 - 3 \cos^2 i_{oo}$

$$E02 = \frac{e_o}{2}$$

$$E06 = \frac{e_o}{6}$$

$$E03 = \frac{e_o}{3}$$

$$AC = \frac{\cos^5 i_{oo}}{p^4}$$

$$ACS = \frac{\cos^5 i_{oo} \sin i_{oo}}{2p^5}$$

$$ACS32 = \frac{5\varepsilon^{3/2}\sin i_{oo}\cos^4 i_{oo}}{p^4}$$

$$AC32 = \frac{\varepsilon^{3/2}\cos^5 i_{oo}}{p^5}$$

$$G0 = -1+3\cos^2 i_{oo} - \frac{e_o^2}{2}(1-5\cos^2 i_{oo})$$

$$G1 = \frac{e_o^2}{4}(1-3\cos^2 i_{oo})$$

$$G2 = -(\frac{\sin^2 i_{oo}}{3} - \frac{e_o^2}{3} + \frac{5}{6}e_o^2\sin^2 i_{oo})$$

$$G3 = \frac{e_o^2}{6}(1-9\cos^2 i_{oo})$$

$$G4 = -\frac{e_o}{12}(5-11\cos^2 i_{oo})$$

$$G5 = -\frac{e_o^2}{12}(1-3\cos^2 i_{oo})$$

$ACS2 = ACS \cdot P$

$C2SP = C_2^{*\prime}$

$C2S = C_2^*$

$S1P = S_1'$

$S1 = S_1$

$B2SP = B_2^{*\prime}$

$B2S = B_2^*$

$SOB = \bar{S}_o$

$GAP1P = \kappa_1'$

$GAP1 = \kappa_1$

$E12 = E_{1/2}$

$S1S = S_1^2$

$SOBS = \bar{S}_o^2$

$GAPOB = \bar{\kappa}_o$

$C2E = \varepsilon^2 C_2^*$

$B2E = \varepsilon^2 B_2^*$

$B2E2 = 2\varepsilon^2 B_2^*$

$E3K = \varepsilon^3 \kappa_1$

$P2 = p^2$

$$C22E = \frac{2\varepsilon^2 C_2^*}{p^2}$$

$PI = \pi$

$TWOPI = 2\pi$

$PIO2 = \frac{\pi}{2}$

$$ACC = \frac{5\cos i_{oo}}{A^4 S_1'}$$

$$A6 = \frac{1}{2A^6}$$

$$EM22 = \sqrt{1-e_o^2}$$

$$EM212 = \frac{2}{\sqrt{1-e_o^2}}$$

$$EF = \frac{\sqrt{1-e_o^2}}{1+e_o}$$

$$A3E = \frac{p^3}{\cos^3 i_{oo}\ (1-e_o^2)}$$

$$ACSS = -\frac{\varepsilon \cos^5 i_{oo} \sin i_{oo}}{2p^4}$$

$$C4E = (C2E)^2$$

$$A3 = p^3/\cos^3 i_{oo}$$

B. Calculate constants for time approximation.

$$T01 = t_{o1} = \frac{p^3}{\cos^3 i_{oo}\ (1-e_o^2)} \left\{\frac{e_o \sin(\phi_i - w)}{1+e_o \cos(\phi_i - w)}\right.$$

$$\left.\frac{-2}{\sqrt{1-e_o^2}} \tan^{-1}\left[\frac{\sqrt{1-e_o^2}}{1+e_o} \tan\left(\frac{\phi_i - w}{2}\right)\right]\right\} + t_o ,$$

where

$$\tan\left(\frac{\phi_i - w}{2}\right) = \frac{\sin(\phi_i - w)}{1+\cos(\phi_i - w)}$$

The $\tan^{-1}$ is taken as the positive principal value and corrected to be in the same quadrant as $\frac{\phi_1 - w}{2}$ by subprogram QUAD1.

If $1+\cos(\phi_1 - w) = 0$, $\tan^{-1}$ is set equal to $\frac{\pi}{2}$ and corrected for quadrant by subprogram QUAD1.

C. If luni-solar perturbations are to be considered, set tape control arrays.

If LS = 1, set TAB1, TAB2, and GM arrays, and continue.
If LS = 2, go to II.

II. Check Case Number for Perigee Calculation.

If $-\kappa_1 < \bar{\kappa}_o < \kappa_1$, go to IIIA.

If $\kappa_1 = \bar{\kappa}_o$, go to IIIB.

If $\bar{\kappa}_o > \kappa_1$, go to IIIC.

III. Set Case Flag for Perigee Calculation and Evaluate Necessary Constants for Case in Question.

A. Set

$$IC = 1$$

Calculate

$$RK = \sqrt{2\kappa_1}$$

$$XMOD = k_1 = \sqrt{\frac{\bar{\kappa}_o + \kappa_1}{2\kappa_1}}$$

$$AMP = \sqrt{\frac{\kappa_1 - \bar{\kappa}_o}{\kappa_1 + \bar{\kappa}_o}}$$

$$CW = \cos(w)$$

If $\cos w = 0$, set CHI1S = w, and go to A.2. Otherwise, continue.

1. $$\text{CHI1} = \tan^{-1}\left[\frac{\kappa_1-\bar{\kappa}_o}{\kappa_1+\bar{\kappa}_o}\tan^2 w-1\right]^{1/2}$$

$$\text{CHI1S} = \chi_1^* = \text{CHI1 adjusted for quadrant}$$

(function QUAD1)

2. QPER = K (quarter-period of elliptic function)

$$\text{PHIO} = \bar{\phi}_o = \pm\frac{F(\chi_1^*, k_1)}{\sqrt{2\kappa_1}} + \bar{\phi}_i$$

(sign is chosen opposite sign of $\bar{S}_o$)

Go to IV.

B. Set IC = 2

If $w = 0$ or π, ω = const. = w, and was stored in subroutine CONST, set flag IWC = 1 and go to IV. Otherwise,

calculate

$$\text{RK} = \sqrt{2\kappa_1}$$

$$\text{TW2} = \left|\tan\frac{w}{2}\right| = \left|\frac{\sin w}{1+\cos w}\right|$$

Set flag

$$\text{IWC} = 2,\quad \text{W02} = \frac{w}{2}$$

Go to IV.

C. Set

$$IC = 3$$

Calculate

$$RK = \sqrt{\bar{\kappa}_o + \kappa_1}$$

$$XMOD = k_2 = \left[\frac{2\kappa_1}{\kappa_1 + \bar{\kappa}_o}\right]^{1/2}$$

$$RMK = \sqrt{\bar{\kappa}_o - \kappa_1}$$

$$AMP = \left[\frac{\bar{\kappa}_o - \kappa_1}{\bar{\kappa}_o + \kappa_1}\right]^{1/2}$$

If cos w = 0, set CHI2S = w, and go to C.2. Otherwise, continue.

1. $$CHI2 = \tan^{-1}\left[\left(\frac{\bar{\kappa}_o + \kappa_1}{\bar{\kappa}_o - \kappa_1}\right)^{1/2} \tan w\right]$$

$CHI2S = \chi_2$ = CHI2 adjusted for quadrant

(function QUADI)

2. QPER = K (quarter-period of elliptic function)

(function ELIPE)

$$PHIO = \bar{\phi}_o = \pm (\bar{\kappa}_o + \kappa_1)^{-1/2} F(\chi_2, k_2) + \bar{\phi}_i$$

(sign chosen opposite sign of $\bar{S}_o$)

IV. Determine Which Form Will Be Used to Find e_a. Set Flag and Evaluate Necessary Constants.

A. If $|1-5\cos^2 i_{oo}| \leq 0.106$, go to IVC; otherwise go to IVB.

B. Set flag to use first form for e_a (IE = 1).

Go to V.

C. Calculate

$$GAM1 = \gamma_1 = \frac{B_2^{*\prime}}{\sin i_{oo} \sqrt{\kappa_1'}}$$

D. If $\bar{S}_o^2 \leq \kappa_1$, continue; otherwise go to IVF.

E. Set flag to use second form for e_a (IE = 2).

Calculate

$$GOK = \frac{\bar{\kappa}_o}{\kappa_1}$$

$$SOK = \frac{\bar{S}_o}{\sqrt{\kappa_1}}$$

Go to V.

F. Set flag to use third form for e_a (IE = 3).

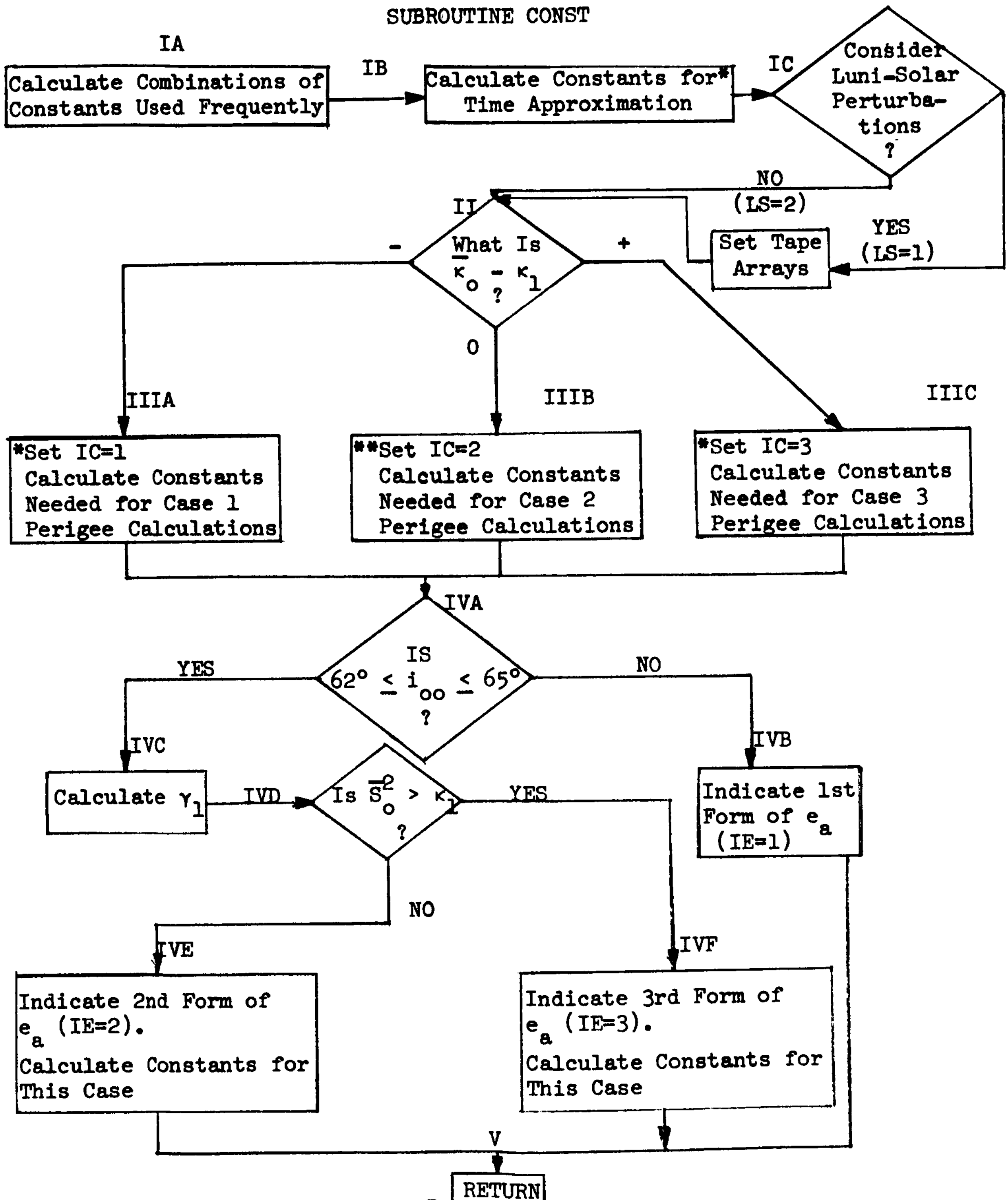
Calculate

$$OSK2 = \frac{\kappa_1}{\bar{S}_o^2} \quad OSK = \frac{\sqrt{\kappa_1}}{\bar{S}_o}$$

V. Return to Main Program.

5.2.2 Detail Flow Chart

** Has transfer for w=0 or π, then ω=w=Const.



* Has Transfer for Arctan (∞) = $\frac{\pi}{2}$

Section 5.2.3 <u>Program Listing</u>

The following pages give the listing of subroutine CONST.

```
      SUBROUTINE  CONST(E,P,XI,C,EPS12,W,PHI,TO,LS)
      COMMON  /CON/ CI,CI2,CI4,SI,SI2,CS,TI,E2,E2C,EM2,P4,AB
     1,CRD,C2W,EPS32,EPS2,EPS3,CI31,EO2,EO6,EO3,AM4,AC,ACS,A
     2 CS32,AC32,G0,G1,G2,G3,G4,G5,C2SP,C2S,B2SP,B2S,S1P,S1,
     3SOB,GAP1P,GAP1,GAP0B,C2E,B2E,B2E2,E3K,C22E,PI,TWOPI,
     4PIO2,IC,RK,XMOD,AMP,CW,QPER,PHIO,TW2,RMK,IE,GAM1,GOK,
     5SOK,OSK2,OSK,E12,S1S,SOBS ,ACSS ,C4F,A3,EPS
     6,ACS2,EM22,EM212,EF,A3E,TO1,P2,ACC,A6,XW,IWC,WO2
      COMMON /TABLE/ TAB1(36),TAB2(13),GP(3,2),GM(2)
      COMMON /APS/AS(6),AD(6),C2IOC,U1,XIOC,DQ1,XI1
C IA   STORE     CONSTANTS USED FREQUENTLY
      P4 = P** 4
      CI = COS (XI)
      SI = SIN  (XI)
      CI2 = CI*CI
      CI4 = CI2* CI2
      AM4 = CI4 /P4
      AB = .5* AM4*AM4
      SI2 = SI*SI
      E2 = E*E
      E2C =E2*C
      CS = CI * SI
      TI = SI/CI
      CRD = 1. -5.* CI2
      EM2= 1.- E2
      C2W = COS (2.*W)
      EPS=EPS12*EPS12
      EPS32= EPS12**3
      EPS2= EPS32* EPS12
      EPS3= EPS32* EPS32
       PHIB=PHI*EPS32
      CI31 = 1.- 3.*CI2
      EO2 = E/2.
      EO6 = E/6.
      EO3= E/3.
      AC = CI * AM 4
      ACS = .5* SI*AC/P
      ACS32 = 5.*EPS32* AM4*SI
      AC32 = EPS32 *AC/P
      G0 = -CI31-E2*CRD*.5
      G1 = .25* E2* CI31
      G2 =(E2-SI2)/3. -.83333333* E2*SI2
      G3 = E2*(.16666666-1.5*CI2)
      G4 = E *(11.* CI2 -5.)/12.
      G5 = - E2* CI31/12.
      ACS2 = ACS*P
      C2SP =AB*(1.5*C-.83333333E-01  +(1.25-10.5*C)*CI2)
      C2S = E2* CS* C2SP
      S1P=AM4*(1.-7.5*CI2)
      S1 = TI * S1P
      B2SP = AB*((.83333333E-1-1.5*C)*EM2 + CI2*(EM2*12.*C
```

```
     1-2.3333333 +1.3333333* E2) +CI4*( -10.5*C* EM2 + 1.25
     2*( 5.- E2)))
      B2S = E* B2SP
      SOB = - CRD* AM4/(2.*EPS12)
      GAP1P = S1P* C2SP
      GAP1 = E2* SI2 *GAP1P
      E12 = -B2S*SOB/GAP1
      S1S = S1*S1
      SOBS = SOB* SOB
      GAPOB = SOBS + GAP1* C2W
      C2E = EPS 2 *C2S
      B2E = EPS 2 *B2S
      B2E2 = 2.* B2E
      E3K = EPS3* GAP1
      P2 =P*P
      C22E = 2.*C2E /P2
      PI = 3.1415927
      TWOPI =6.2831853
      PIO2 =1.5707963
      ACC = 5.*AC/S1P
      A6 = .5*CI*AC/P2
      EM22 = SQRT(EM2)
      EM212= 2./EM22
      EF = EM22/(1.+ E)
      A3E = CI2/(AC*P*EM2)
      ACSS=-EPS*ACS*P
      C4F=C2F*C2F
       A3=P2*P/(CI2*CI)
C IB  CALCULATION OF CONST. FOR TIME APPROXIMATION
C    CHECK  IF  ARC TAN IS  PI/2
      CPMW = COS(PHI -W)
      SPMW = SIN(PHI-W)
      ZD = 1. + CPMW
      IF (ZD)  1,2,1
    2  TANG = PIO2
      GO  TO  3
    1 TANG = ATAN(EF*ABS(SPMW)/ZD )
    3 ANG2 = .5*(PHI -W)
      TANG = QUAD1(TANG,ANG2,PIO2,PI,TWOPI )
      TO1 =A3E*( E* SPMW/(1.+E *CPMW) -EM212* TANG)+ TO
C  IC  IF LUNI-SOLAR PERT. CONSIDERED, SET TAPE CONTROL ARRAYS
      GO TO (4,20),LS
C GLOSSARY
C  TAB1(7) = GM(EARTH) IN KM**3/SEC**2
C  TAB1(21)= GM(SUN)   IN KM**3/SEC**2
C  TAB1(23)= GM(MOON)  IN KM**3/SEC**2
C  TAB1(25)= A.U.      IN KM
C  TAB1(27)= CONV.FACTOR FOR LUNAR COORDS.(KM)  (FICT.EARTH RADIUS)
C  TAB1(33)= SECONDS/ MEAN SOLAR DAY
C (TAB2 CONTAINS EPHEM.TAPE OUTPUT CONTROL FLAGS)
    4 DO 100 K = 1,36
```

```
 100   TAB1(K) = 0.
       TAB1(7) = 398603.2
       TAB1(21)= 1.3271544 E11
       TAB1(23)= 4902.7779
       TAB1(25)= 1.49599 E8
       TAB1(27)= 6378.327
       TAB1(33)= 86400.
       DO 101 K = 1,13
 101   TAB2(K) = 0.
       TAB2(3) = 1.
       TAB2(10)= 1.
       TAB2(11)= 1.
       GM(1) = TAB1(21)
       GM(2) = TAB1(23)
C II   CHECK CASE NO. FOR PERIGEE  CALC.
   20 IF (GAP1- GAPOB) 32,31,30
C IIIA COMPUTE  CONSTANTS FOR CASE 1
   30 IC =1
      RK = SQRT(2.* GAP1)
      XMOD = SQRT(GAP1 +GAPOB)/RK
      AMP = SQRT((GAP1-GAPOB)/(GAP1+GAPOB))
      CW = COS(W)
      IF(CW) 34,33,34
   33 CHI1S= W
      GO TO  35
C III A1
   34 CHI1= ATAN(SQRT((SIN(W)/(AMP*CW))**2 -1.))
      CHI1S = QUAD1(CHI1,W,PIO2,PI,TWOPI)
C III A2
   35 QPER = ELIPE (XMOD)
      PHI0 = - ELI(CHI1S,     QPER)/SIGN(RK,SOB)+PHIB
      IWC= 2
      GO TO  40
C IIIB COMPUTE CONSTANTS FOR  CASE  2
   31 IC = 2
C     IF  PERIGEE INITIALLY  0 OR PI, IT IS CONSTANT
      SW = SIN(W)
      IF (SW)  311,310,311
  310 XW = W
      IWC = 1
      GO TO 40
  311 RK = SQRT(2.*GAP1)
      TW2=ABS(SW/(1.+COS(W)))
      WO2 = W/2.
      GO  TO  40
C IIIC COMPUTE  CONSTANTS FOR CASE  3
   32 IC  = 3
      RK = SQRT(GAPOB+GAP1)
      XMOD= SQRT(2.*GAP1)/RK
      RMK = SQRT(GAPOB- GAP1)
      AMP= RMK/RK
```

```
      CW = COS(W)
      IF (CW)37,36,37
C III C1
   36 CHI2S = W
      GO  TO  38
   37 CHI2 =ATAN( ABS(SIN(W)/(AMP*CW)))
      CHI2S = QUAD1(CHI2,W,PIO2,PI,TWOPI)
C III C2
   38 QPER = FLIPE (XMOD)
      PHIO = -ELI (CHI2S,       QPER)/ SIGN(RK,SOB) +PHIB
C IV    DETERMINE  FORM FOR FA,EVALUATE CONSTANTS
C IV A
   40 IF (ABS(CRD)-.106)  42,42,41
C IV B
   41 IE =1
      GO TO 50
C IV C
   42 GAM1= B2SP/(SI *SQRT(GAP1P))
C IV D
      IF (SOBS- GAP1) 43,43,44
C IV E
   43 IE = 2
      GOK = GAPOB/GAP1
      SOK = SOB/SQRT(GAP1)
      GO TO 50
C IV F
   44 IE = 3
      OSK2 = GAP1/SOBS
      OSK = SQRT(OSK2)
C  V
C  STORE APPROX. SOL. AND DERIV. FOR P
   50  AS(1)=P
       AD(1)=0.0
      RETURN
      END
```

5.3 SUBROUTINE APSOL (PHI, PHIT, IP, XIO, XLO, TO, W, E, K1OR3, PHIIB, EPS12)

Calculates approximate solutions and necessary derivatives for the desired angle ϕ. Inputs are ϕ (modded to 2π each time it is stepped), ϕ_T (total Φ unmodded), 1st point flag (= 1 if 1st point = 2, otherwise); initial values of p, i_{oo}, L_o, t, ω, and e; flag to determine perigee center of oscillation, $\bar{\phi}_i$, and $\varepsilon^{1/2}$. Outputs are in common /APS/ as arrays AS(6), AD(6) for approximate solutions and derivatives. AS(1) = p_a, AS(2) = Ω_a, AS(3) = i_a, AS(4) = q_a, AS(5) = u_a, AS(6) = t_a. AD(1) = $\frac{dp_a}{d\phi} = 0$, AD(2) = $\frac{d\Omega_a}{d\phi}$, AD(3) = $\frac{di_a}{d\phi}$. AD(4) = $\frac{dq_a}{d\phi}$, AD(5) = H, AD(6) = $\frac{dt_a}{d\phi}$. Other outputs are C2IOC = $\cos^2 i_{oc}$, XI1 = εi_1, U1 = εu_1, XIOC = i_{oc}. DQ1 = $\frac{\varepsilon dq_1}{d\phi}$. Uses as input labeled common /CON/ to provide all the constants obtained in CONST.

5.3.1 Equations in Order of Solution

Calculate

$$\text{PHIB} = \bar{\phi}$$

$$\text{PHIBT} = \bar{\phi}_T$$

I. Determine if This is the First Point of the Trajectory.

A. If this is the first point of the trajectory (IP = 1), go to I B; otherwise (IP = 2) go to II.

B. Set some of the approximate solutions equal to the initial conditions.

$$t_a = t_i,\ i_{o1/2} = 0,\ e_{1/2} = 0,\ \omega = w.$$

Combine L_o and $L_{1/2}$ constants and store in L_o location.

Go to III.

II. Determine the Case Number for Calculating the Perigee (IC is the Case Number).

If IC = 1, go to IIA.
If IC = 2, go to IIB.
If IC = 3, go to IIC.

A. Calculate ω from case 1 formula.

If cn = 0, set $\omega = \frac{\pi}{2}$ and go to QUAD2; otherwise,

$$XW = \omega = \tan^{-1} \left\{ \left[\frac{\kappa_1 - \bar{\kappa}_o}{\kappa_1 + \bar{\kappa}_o} \right]^{1/2} \frac{1}{cn[\sqrt{2\kappa_1}\,(\bar{\phi}-\bar{\phi}_o)]} \right\}$$

Adjust this ω to the proper quadrant by using QUAD2.

$$\omega = \text{QUAD2}\ (\omega,\ z_1,\ K,\ \text{K1OR3},\ \pi)$$

Go to III.

B. If $w = 0$ or π, (IWC = 0); go to III. Otherwise, calculate ω from case 2 formula.

$$XW = \omega = 2 \tan^{-1}\{ e^{\text{sign}\ (\bar{S}_o)\ \sqrt{2\kappa_1}\ (\bar{\phi}-\bar{\phi}_i)} \tan \frac{w}{2} \}$$

Adjust the quadrant of ω using QUAD1.

$$\omega = 2\ \text{QUAD1}\ (\frac{\omega}{2},\ \frac{w}{2},\ \frac{\pi}{2},\ \pi,\ 2\pi)$$

Go to III.

C. Calculate ω from case 3 formula. If $cn = 0$, set $\omega = \frac{\pi}{2}$ and skip calculation. Otherwise, calculate

$$\omega = \tan^{-1}\left\{\left[\frac{\bar{\kappa}_o - \kappa_1}{\bar{\kappa}_o + \kappa_1}\right]^{1/2} \frac{[1-cn^2(z_2)]^{1/2}}{cn\ (z_2)}\right\}$$

Adjust ω to correct quadrant using QUAD1 and the elliptic function quarter-period K.

$$\omega = \text{QUAD1}\ (\omega,\ z_2,\ K,\ \pi,\ 2\pi)$$

Reduction of Entries to Trig Functions for Approximate Solutions and Derivatives.

$$CP = \cos\phi$$
$$SP = \sin\phi$$
$$CXW = \cos\omega$$
$$SXW = \sin\omega$$
$$S2P = \sin 2\phi = 2(CP)(SP)$$
$$C2P = \cos 2\phi = 2(CP)^2-1$$
$$S2XW = \sin 2\omega = 2(CXW)(SXW)$$
$$C2XW = \cos 2\omega = 2(CXW)^2-1$$
$$CPPW = \cos(\phi+\omega) = (CP)(CXW) - (SP)(SXW)$$
$$SPPW = \sin(\phi+\omega) = (SP)(CXW) + (CP)(SXW)$$
$$CPMW = \cos(\phi-\omega) = (CP)(CXW) + (SP)(SXW)$$
$$SPMW = \sin(\phi-\omega) = (SP)(CXW) - (CP)(SXW)$$
$$C2PMW = \cos 2(\phi-\omega) = 2\cos^2(\phi-\omega)-1 = 2(CPMW)^2-1$$
$$S2PMW = \sin 2(\phi-\omega) = 2\sin(\phi-\omega)\cos(\phi-\omega) = 2(CPMW)(SPMW)$$
$$C3PMW = \cos(3\phi-\omega) = \cos 2\phi\cos(\phi-\omega) - \sin 2\phi\sin(\phi-\omega)$$
$$= (C2P)(CPMW) - (S2P)SPMW)$$
$$S3PMW = \sin(3\phi-\omega) = \sin 2\phi\cos(\phi-\omega) + \cos 2\phi\sin(\phi-\omega)$$
$$= (S2P)(CPMW) + (C2P)(SPMW)$$

$$C4PMW = \cos (4\phi-2\omega)$$
$$= \cos (3\phi-\omega) \cos (\phi-\omega) - \sin (3\phi-\omega) \sin (\phi-\omega)$$
$$= (C3PMW)(CPMW) - (S3PMW)(SPMW)$$
$$S4PMW = \sin (4\phi-2\omega)$$
$$= \sin (3\phi-\omega) \cos (\phi-2\omega) + \cos (3\phi-2\omega) \sin (\phi-\omega)$$
$$= (S3PMW)(CPMW) + (C3PMW)(SPMW)$$

III. Calculate approximate nodal solution.

$$OMEOO = \Omega_{oo} \tag{31}$$

$$OMEO12 = \Omega_{o\ 1/2} \tag{32}$$

$$OME32 = \Omega_{3/2} \tag{33}$$

$$\Omega_a = \frac{1}{\varepsilon^{1/2}} [\Omega_{oo} + \varepsilon^{1/2} \Omega_{o1/2} + \varepsilon^{3/2} \Omega_{3/2}] + L_o$$

Calculate

$$XI1 = \varepsilon i_1 \tag{38}$$

and

$$U1 = \varepsilon u_1 \tag{58}$$

Calculate

$$TA = t_a = \frac{p^3}{\cos^3 i_{oo}(1-e_o^2)} \{\frac{-e_o \sin (\phi-\omega)}{1+e_o \cos(\phi-\omega)} + \frac{2}{\sqrt{1-e_o^2}} \tan^{-1} [\frac{\sqrt{1-e_o^2}}{1+e_o} \tan(\frac{\phi-\omega}{2})] + t_{o1}$$

providing $\tan (\frac{\phi-\omega}{2}) \neq \infty$. If it does, take $\tan^{-1} (\infty) = \frac{\pi}{2}$.

Take the positive principal $\tan^{-1}$ and find correct quadrant using QUAD1. Save the number of complete revolutions and add this to the QUAD1 result.

IV. Decide Which Equations Will be Used to Calculate $i_{o\ 1/2}$ and $e_{1/2}$.

If IE = 1, go to IVA.
If IE = 2, go to IVB.
If IE = 3, go to IVC.

A. Calculate

$$AA = \frac{[\cos 2w - \cos 2\omega]}{\overline{S}_o + \mathrm{sign}\,(\overline{S}_o)\sqrt{\overline{\kappa}_o - \kappa_1 \cos 2\omega}}$$

then

$$ES12 = e_{1/2} = B_2^{*} \cdot AA$$

$$XI12 = i_{o\ 1/2} = C_2^{*} \cdot AA$$

Go to V.

B. Calculate

$$i_{o\ 1/2} = \frac{1}{S_1} [\mathrm{sign}\,(\overline{S}_o)\ (\overline{\kappa}_o - \kappa_1 \cos 2\omega)^{1/2} - \overline{S}_o]$$

$$e_{1/2} = \gamma_1\ [\mathrm{sign}\,(\overline{S}_o)\sqrt{\frac{\overline{\kappa}_o}{\kappa_1} \cos 2\omega} - \frac{\overline{S}_o}{\sqrt{\kappa_1}}]$$

Go to V.

C. Calculate

$$i_{o\ 1/2} = \frac{1}{S_1} [\mathrm{sign}\,(\overline{S}_o)\ (\overline{\kappa}_o - \kappa_1 \cos 2\omega)^{1/2} - \overline{S}_o]$$

$$e_{1/2} = \gamma_1 \frac{\sqrt{\kappa_1}}{\bar{S}_o} \frac{(\cos 2w - \cos 2\omega)}{1 + \sqrt{1 + \frac{\bar{\kappa}_1}{\bar{S}_o^2}(\cos 2w - \cos 2\omega)}}$$

V. Calculate

$$XIOC = i_{oc} = i_{oo} + \varepsilon^{1/2}\, i_{o\,1/2}$$

$$XIA = i_a = i_{oc} + \varepsilon i_1$$

$$EA = e_a = e_o + \varepsilon^{1/2}\, e_{1/2}$$

$$UA = u_a = u_o + \varepsilon u_1$$

VI. Calculate the Derivatives and Second-Derivatives of the Approximate Solutions Which are Necessary to Find the Modified-Encke Equations.

These are:

$$DOMEO = \frac{1}{\varepsilon^{1/2}} \frac{d\Omega_{oo}}{d\phi} \tag{63}$$

$$DWB = \frac{d\omega}{d\bar{\phi}} \tag{65}$$

$$DW = \frac{d\omega}{d\phi} = \varepsilon^{3/2} \frac{d\omega}{d\bar{\phi}}$$

$$DOME12 = \frac{d\Omega_{o\,1/2}}{d\phi} \tag{66}$$

$$DOME32 = \frac{d\Omega_{3/2}}{d\phi} \tag{67}$$

$$AD(2) = \frac{d\Omega_a}{d\phi} \qquad (62)$$

$$AD(3) = \frac{di_a}{d\phi} \qquad (69),(70),(71)$$

$$DEA = \frac{de_a}{d\phi} \qquad (72),(73)$$

$$DU1 = \frac{du_1}{d\phi} \qquad (74)$$

$$D2W = \frac{d^2\omega}{d\phi^2} \qquad (79)$$

$$D2EA = \frac{d^2e_a}{d\phi^2} \qquad (78)$$

$$EPD2IO = \varepsilon^{1/2} \frac{d^2 i^*_{o\ 1/2}}{d\phi^2} \qquad (80)$$

$$AS(4) = q_a \qquad (60)$$

$$DQ1 = \varepsilon \frac{dq_1}{d\phi} \qquad (81)$$

$$DSIOC = \sin i_{oc} \frac{di_{oc}}{d\phi} = \varepsilon^{1/2} \sin i_{oc} \frac{di_{o\ 1/2}}{d\phi} \qquad (70)$$

$$H = H \qquad (77)$$

$$DTA = \frac{dt_a}{d\phi} \qquad (106)$$

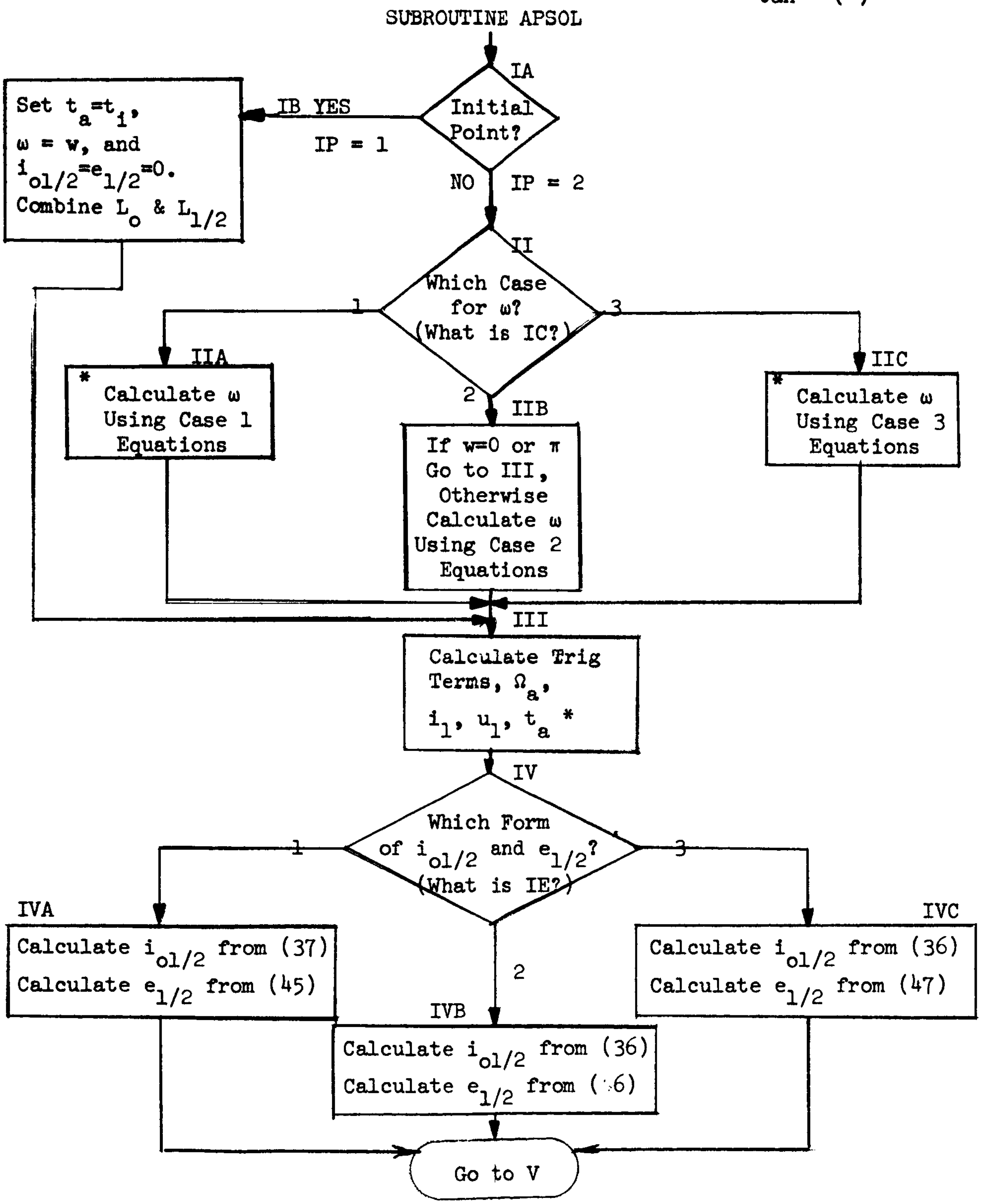
$$DQ0 = \frac{dq_o}{d\phi} \qquad (76)$$

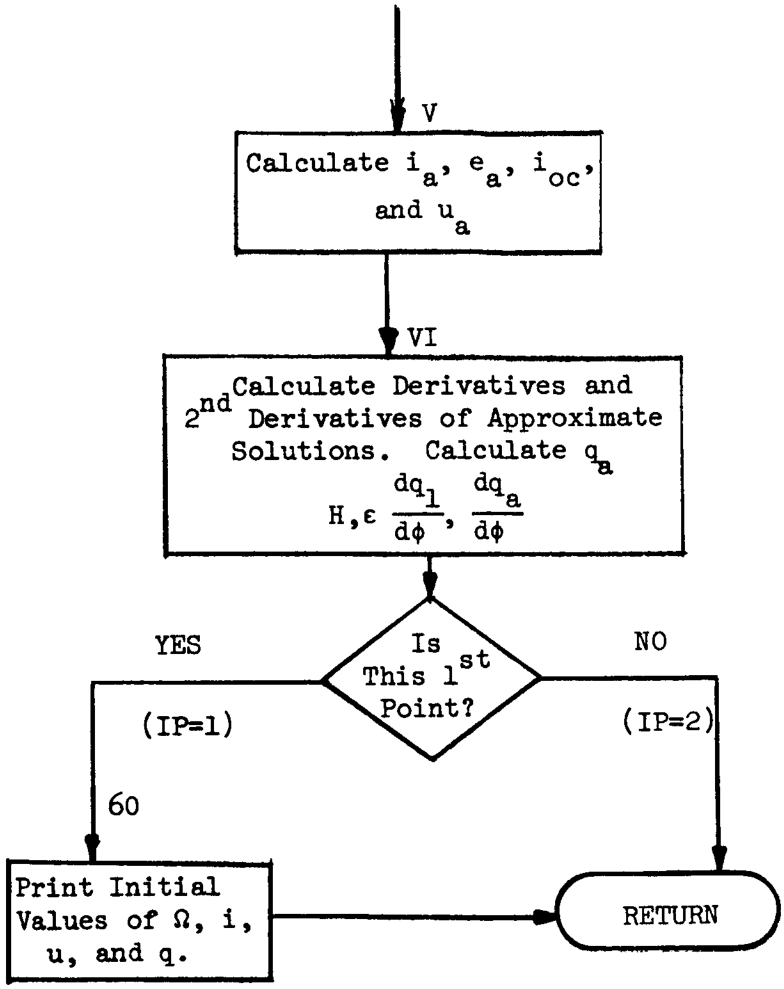
If this is first point of trajectory (IP = 1), print initial values of Ω (deg), i (deg), u, and q and return.

If this is not the first point (IP = 2), skip the print and return.

5.3.2 Detail Flow Chart

* Includes Transfer to Obtain $\frac{\pi}{2}$ for $\tan^{-1}(\infty)$




V
Calculate i_a, e_a, i_{oc}, and u_a
VI
Calculate Derivatives and 2nd Derivatives of Approximate Solutions. Calculate q_a
H, ε $\frac{dq_1}{d\phi}$, $\frac{dq_a}{d\phi}$
Is This 1st Point?
YES
(IP=1)
60
NO
(IP=2)
Print Initial Values of Ω, i, u, and q.
RETURN

Section 5.3.3 Program Listing

The following pages give the listing of subroutine APSOL.

```
      SUBROUTINE APSOL(PHI,PHIT,IP,XIO,XLO,TO,W,F,K1OR3,PHIIB,EPS12)
      COMMON  /CON/ CI,CI2,CI4,SI,SI2,CS,TI,F2,F2C,FM2,P4,AB
     1,CRD,C2W,EPS32,EPS2,EPS3,CI31,EO2,FO6,FO3,AM4,AC,ACS,A
     2 CS32,AC32,G0,G1,G2,G3,G4,G5,C2SP,C2S,B2SP,B2S,S1P,S1,
     3SOB,GAP1P,GAP1,GAPOB,C2E,B2E,B2E2,E3K,C22E,PI,TWOPI,
     4PIO2,IC,RK,XMOD,AMP,CW,QPER,PHIO,TW2,RMK,IE,GAM1,GOK,
     5SOK,OSK2,OSK,E12,S1S,SOBS ,ACSS ,C4E,A3,EPS
     6,ACS2,EM22,EM212,EF,A3E,TO1,P2,ACC,A6,XW,IWC,WO2
      COMMON /APS/AS(6),AD(6),C2IOC,U1,XIOC,DQ1,XI1,EA
      PHIB = PHI* EPS32
      PHIBT = PHIT *EPS32
C I   IS THIS 1ST POINT
      GO TO (10,20 ),IP
C IB  SET APPROX. SOLS.= INITIAL CONDITIONS
   10 TA = TO
      XI12=0.0
       ES12=0.0
      XW=W
C  COMBINE LO AND L1/2 CONSTANTS
      XLO=XLO+ACC*(SOB*PHIBT-XW)
      GO  TO  30
CII   DETERMINE PERIGEE CASE NO. (IC)
   20 GO TO (21,25,28), IC
C IIA  CALCULATE  PERIGEE  BY CASE 1 FORMULAS
   21 ANG = RK* (PHIBT-PHIO)
      CN = SQRT(1. -(ELIF(ANG          ))**2)
      IF (CN)  23,22,23
C  IF PERIGEE SHOULD BE  PI/2 , ELIMINATE  ARC TAN
   22 XW = PIO2
      GO TO 24
   23 XW = ATAN( AMP/CN)
   24 XW = QUAD2(XW,ANG,QPER,K1OR3,PI)
      GO TO  30
C IIB  CALCULATE  PERIGEE  BY CASE 2 FORMULAS
C   CHECK IWC TO SEE IF PERIGEE IS CONSTANT
   25  GO TO (30,26),IWC
   26 ANG = RK*(PHIBT    -PHIIB)
      XW  =     ATAN(    TW2 *EXP(SIGN(ANG,SOB)))
      XW = 2.*QUAD1(XW,WO2,PIO2,PI,TWOPI)
      GO TO  30
C IIC  CALCULATE PERIGEE  BY CASE 3 FORMULAS
   28 Z2 = RK*(PHIBT-PHIO)
      SN = ABS(ELIF(Z2))
      CN = SQRT(1. - SN*SN)
C  IF PERIGEE SHOULD BE  PI/2 ,ELIMINATE ARC TAN
      IF (CN)290,29,290
   29 XW = PIO2
      GO TO  280
  290 XW = ATAN( AMP*SN/CN)
  280 XW = QUAD1(XW,Z2,QPER,PI,TWOPI)
C III
```

```
C   CALCULATE  TRIG TERMS FOR  SOLS. AND DERIVATIVES
 30 SP = SIN(PHI)
    CP= COS(PHI)
    CXW = COS(XW)
    SXW = SIN(XW)
    S2P=2.*CP*SP
    C2P= 2.* CP*CP -1.
    S2XW = 2.*SXW*CXW
    C2XW = 2.*CXW*CXW-1.
    CPPW =(CP*CXW)-(SP*SXW)
    SPPW =(SP*CXW)+(CP* SXW)
    CPMW = (CP* CXW)+(SP*SXW)
    SPMW =(SP*CXW)-(CP*SXW)
    C2PMW = 2.* CPMW*CPMW-1.
    S2PMW = 2.*CPMW *SPMW
    C3PMW = C2P*CPMW - S2P*SPMW
    S3PMW = S2P*CPMW +C2P* SPMW
    C4PMW = C3PMW*CPMW -S3PMW *SPMW
    S4PMW = S3PMW *CPMW +C3PMW *SPMW
C   CALCULATE APPROX. NODAL SOLUTION
    OMEOO = - AC*PHIBT
    OMEO12= -ACC*(SOB* PHIBT-XW)
    OME32 =-AC*(-.5*S2P+E*SPMW-EO2*SPPW -EO6*S 3PMW)
    OMEGA = OMEOO/EPS12 +OMEO12 +EPS*OME32+XLO
C   CALCULATE  I1 APPROXIMATION
    XI1 =  ACS2*(C2P+E*CPPW +EO3* C3PMW)*EPS
C   CALCULATE   U1 APPROXIMATION
    U1=A6*(G0+G1*C2XW+G2*C2P+G3*C2PMW+G4*C3PMW+G5*C4PMW)
   1  *EPS
    GO TO (50,300),IP
C   CALCULATE  APPROX. TIME ,TA
C    CHECK FOR ZERO DIVISOR IN ARC TAN
300 ZD = 1.+ CPMW
    IF (ZD)  31,32,31
 32 TANG = PIO2
    GO  TO  33
 31 TANG = ATAN(EF* ABS(SPMW)/ZD)
 33 ANG2 =(PHIT-XW)/2.
    ANG3=ANG2
    TANG = AINT(ANG3/TWOPI)*TWOPI +QUAD1(TANG,ANG2,PIO2,
   1PI,TWOPI)
    TA =A3E*(-E*SPMW/(1.+E*CPMW)+EM212*TANG)+ TO1
C IV  CHECK FORM OF  E AND  I  EQUATIONS
    SQ = SQRT(GAPOB- GAP1*C2XW)
    GO TO (40,41,42), IF
C IVA  CASE  1
 40 AA = (C2W - C2XW)/(SOB+SIGN(SQ,SOB))
    ES12= B2S* AA
    XI12= C2S* AA
    GO TO  50
C IBV  CASE  2
```

```
   41 SQ1= SQRT(GOK *C2XW )
      XI12 =(SIGN(SQ,SOB)-SOB)/S1
      ES12 = GAM1*(SIGN(SQ1,SOB)- SOK)
      GO TO 50
C  IVC  CASE 3
   42 XI12 =(SIGN(SQ,SOB)- SOB)/S1
      CMC = C2W - C2XW
      ES12= GAM1*OSK*CMC/(1.+SQRT(1.+OSK2*CMC))
C  V
   50 XIOC = XIO +EPS12 *XI12
      XIA  = XIOC +XI1
      EA = E + EPS12* ES12
      CIOC = COS(XIOC)
      SIOC = SIN(XIOC)
      C2IOC = CIOC* CIOC
      U00=  C2IOC          *(1.+EA* CPMW)/P2
      UA = U00+U1
C  VI   CALCULATE  DERIVATIVES
      DOMEO = -AC* EPS
      DWB = SOB + S1*XI12
      DW = EPS32* DWB
      DOME12 = ACS32* XI12
      DOME32 = -AC* (-C2P+E*(1.- DW)*CPMW -EO2* (1.+DW)*CPPW
     1- E06*(3.-DW)*C3PMW)
      AD(2)=  DOMEO + DOME12 +EPS  *DOME32
      AD(3)= C2E * S2XW +ACSS*(2.*S2P +E*(1.+DW)*SPPW +EO3*
     1(3.- DW)*S3PMW)
      DEA = B2E* S2XW
      DU1= A6*(  ( G1*DW*S2XW + G2*S2P +G3*(1.-DW)*S2PMW)
     1*(-2.)- G4*(3.-DW)* S3PMW -2.*G5*(2.- DW)*S4PMW)
      D2W = E3K* S2XW
      D2EA= B2E2* DW* C2XW
      EPD2IO = 2.*C2E*DW*C2XW
      AS(4)=- C22E*S2XW*CIOC*SIOC*(1.+ EA* CPMW) +C2IOC
     1    *(DEA*CPMW -EA*(1.-DW)*SPMW)/P2 + EPS*DU1
      DQ1 = A6*( -2.*G1*( D2W* S2XW+ 2.*DW*DW*C2XW)-4.*G2
     1*C2P -2.*G3*(2.*C2PMW*(1.-DW)**2 -D2W*S2PMW)- G4*(
     2C3PMW*(3.-DW)**2 - D2W*S3PMW )- G5*(C4PMW*4.*(2. -
     3DW)**2 -2.*D2W*S4PMW))*EPS
      DSIOC= C2E*S2XW*SIOC
      H = -2.*U00*(C4E *S2XW*S2XW*(2.*C2IOC-1.)+ SIOC*CIOC*
     1EPD2IO)/C2IOC+CIOC*(-4.*DSIOC    *DEA         *CPMW +2.
     2*(EA* DSIOC -CIOC*DEA)*(1.-DW)*SPMW +CIOC*(D2EA +EA*DW
     3*(2.-DW))*CPMW+ EA*CIOC*D2W*SPMW  )/P2
      DTA = A3*(1.-DW)/(1.+E*CPMW)**2
       DQ0=-U00+C2IOC/P2+H
      AD(6)= DTA
      AD(4)= DQ1+DQ0
      AD(5)= H
      AS(2)= OMEGA
      AS(3)= XIA
```

```
      AS(5)= UA
      AS(6)= TA
      GO TO (60,70 ),IP
   60 XNODEI=OMEGA/.17453293E-01
      XINCI=XIA   /.17453293E-01
      VEL=AS(1)*UA*7.90535872/COS(XIA)
      WRITE(6,61)XNODEI,XINCI,UA,AS(4) ,VEL
   61 FORMAT(32H INITIAL VALUES OF NODE,INC.,U,Q //4E18.8)
   70 RETURN
      END
```

5.4 SUBROUTINE EXPERT (LS, OMEGA, TILT, PHI, R, T, DT, AF, AG, AH, N2)

Subroutine EXPERT calculates the nondimensional accelerations a_f, a_g, and a_h due to the earth's potential and due to the sun and moon if the luni-solar flag LS = 1. Other inputs are Ω, i, ϕ, r, t, Δt, and N2. (Δt is the difference in time since the last entry to this routine).

5.4.1 Equations in Order of Solution

Store quantities needed for SOLUN and GPOT routines.

$$SP = \sin\phi,\ CP = \cos\phi,\ SI = \sin i,$$

$$CT = \cos\theta,\ ST = \sin\theta.$$

Check if longitude is required.

If N2 = 0 (no tesseral or sectorial harmonics), longitude not needed, go to IB.

If N2 $\neq$ 0, longitude needed; go to I.

I. Find longitude (λ) of the Satellite.

$$\text{compute } \cos b = \frac{\cos\phi}{\sin\theta} \text{ and}$$

$$b = \cos^{-1}(\cos b).$$

This gives the principal value. To find desired angle, check $\cos\theta$.

If $\cos\theta \geq 0$, principal value is correct, go to IA.

If $\cos\theta < 0$, replace b with $2\pi - b$ and continue.

A. Find longitude of Greenwich at this time by replacing previous value with the previous value plus amount the earth has rotated. If the longitude of Greenwich exceeds 2π, reduce it by 2π.

Calculate longitude of satellite:

$$\lambda = \Omega + b - \lambda_G$$

B. Find accelerations due to the earth.

Call subroutine GPOT.

II. Consider Luni-Solar Perturbations.

If luni-solar flag (LS) is 1, go to III and prepare to calculate luni-solar perturbations.

If LS is 2, return to the calling program.

III. Convert R to km and T to Hours Before Entering Luni-Solar Routine.

Calculate accelerations due to moon and sun (subroutine SOLUN). Sum lunar and solar contributions. Convert accelerations from km/sec^2 to nondimensional units.

A. Rotate these accelerations into the desired AF, AG, and AH frame.

Compute necessary trig functions for the rotations.

$$SQ = \sin q = \sin i \cos b$$

$$CQ = \cos q = + \sqrt{1 - \sin^2 q} \text{ (since } q \leq 90°\text{)}$$

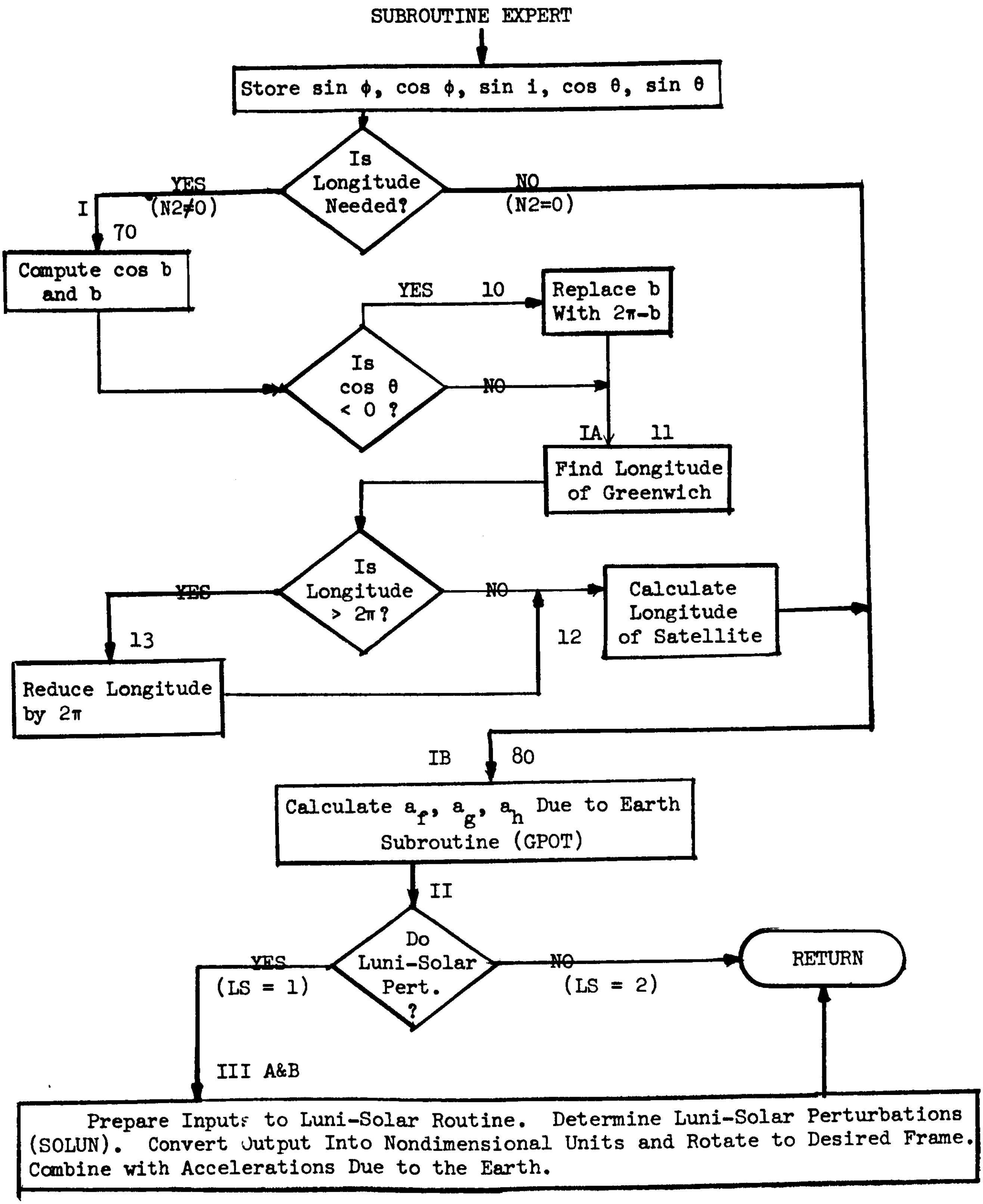
$$a_f = GP\ (1,\ J) \cos q - GP\ (2,\ J) \sin q$$

$$a_g = GP\ (1,\ J) \sin q + GP\ (2,\ J) \cos q$$

$$a_h = GP\ (3,\ J)$$

B. Sum lunar, solar, and earth's potential contributions and return.

5.4.2 Detail Flow Chart



Section 5.4.3 Program Listing

The following page gives the listing of subroutine EXPERT.

```
      SUBROUTINE EXPERT(LS,OMEGA,TILT,PHI,R,T,DT,AF,AG,AH,N2
     1)
      DIMENSION  A(3)
      COMMON /TABLE/ TAB1(36),TAB2(13),GP(3,2),GM(2)
      COMMON /EX/ CB,CT,ST,FWOG,FROT,SP,CP,SI
C     STORE QUANTITIES NEEDED FOR SOLUN AND GPOT
      SP = SIN(PHI)
      CP=  COS(PHI)
      SI = SIN(TILT)
      CT = SI* SP
      ST = SQRT(1. -CT* CT)
C     CHECK IF LONGITUDE NEEDED
      IF(N2) 70,80,70
C   I  FIND  EARTH LONGITUDE OF SATELLITE
   70 CB  =CP/ST
      B = ACOS (CB)
      IF  (CT) 10,11,11
   10 B = 6.2831853    - B
C IA
   11 EWOG = EWOG + FROT * DT
      IF (EWOG -6.3) 12,13,13
   13 EWOG = EWOG - 6.2831853
   12 EW = OMEGA +B - EWOG
C IB  FIND  ACCELERATIONS DUE TO THE EARTH
   80 CALL GPOT(ST,CT,EW,R,AF,AG,AH)
C  II CONSIDER  LUNI-SOLAR  PERTURBATIONS
      GO TO (30,20),LS
C III  PREPARE FOR  LUNI-SOLAR ROUTINE
C   FIND DIMENSIONAL R AND T
   30 RD = R * 6378.1521
      TD = T * .22411493
      CALL  SOLUN (OMEGA,TILT,PHI,RD,TD)
C   SUM LUNAR AND SOLAR PERT. AND NON-DIM.
C  GP(K,1),K=1,3,ARE THE PERT.ACCELS.DUE TO THE SUN (KM3/SEC2)
C  GP(K,2),K=1,3,ARE THE PERT.ACCELS.DUE TO THE MOON(KM3/SEC2)
      DO  31 I=1,3
   31 A(I) =(GP(I,1)+GP(I,2))/.97983068 E-02
C IIIA ROTATE  ACCELERATIONS  TO  AF, AG, AH FRAME  AND SUM
      SQ = SI* CB
      CQ = SQRT(1.- SQ* SQ)
      AF = AF + A(1)*CQ- A(2)*SQ
      AG = AG + A(1)*SQ +A(2)* CQ
      AH = AH + A(3)
   20 RETURN
      END
```

5.5 SUBROUTINE ENCKE (PT, OMEGT, XIT, QT, UT, T, PHI, LS, DT, N2, XIN, PN, UN, P2, PA, QN, E, AJ2, AJ4, KDER)

Subroutine ENCKE evaluates Encke equations of motion for the Runge-Kutta subroutine. Inputs to ENCKE are p, Ω, i, q, u, t, ϕ, LS, DT, N2, i_n, p_n, u_n, p_a^2, p_a, q_n, J_2, J_4, and KDER. Other inputs come from subroutine APSOL through labeled common /APS/. Output is the array E(6) where:

$$E(1) = \frac{dp_n}{d\phi},\ E(2) = \frac{d\Omega_n}{d\phi},\ E(3) = \frac{di_n}{d\phi},\ E(4) = \frac{dq_n}{d\phi},$$

$$E(5) = \frac{du_n}{d\phi},\ \text{and}\ E(6) = \frac{dt_n}{d\phi}.$$

5.5.1 Equations in Order of Solution

I. Compute and Store Useful Quantities.

Find r, $\cos i$, $\sin i$, $\tan i$, $\cos^2 i$, $\sin^2 i$, $\cos^3 i$, $\cos^4 i$, A_1, u^2, p^2, u^5, $\sin\phi$, $\cos\phi$, $\cos\theta$, and $\sin\theta$.

II. Find Perturbative Accelerations, a_f, a_g, a_h (EXPERT).

Calculate

$$\frac{\partial U}{\partial \psi}\ (10), \qquad F\ (8), \qquad \frac{d\phi}{dt}\ (5), \qquad \frac{dt}{d\phi}, \frac{dp}{d\phi}\ (1).$$

Calculate

$$\text{DENOM} = p^2 u^2 \sin^2 i \sin\theta + F \cos^4 i \cos\theta$$

$$\text{RUM} = \frac{F}{\text{DENOM}}\ .$$

Calculate

$$\text{DODPHI} = \frac{d\Omega}{d\phi} \qquad (2)$$

A. Zero J_2 and J_4 since they have already been accounted for.

$$\text{DPHI} = 1^\circ \text{ in radians}$$

$$\phi_1 = \phi - \text{DPHI}$$

$$\phi_2 = \phi + \text{DPHI}$$

$$\text{DELU} = \Delta u = \frac{du_a}{d\phi}\,\text{DPHI}$$

$$\text{R1} = \frac{1}{u-\Delta u}$$

$$\text{R2} = \frac{1}{u+\Delta u}$$

Find a_{f_1}, a_{g_1}, a_{h_1}, and a_{f_2}, a_{g_2}, a_{h_2} (EXPERT).

Calculate:

$$\text{DAFAP} = \frac{da_f}{d\phi} \text{ approximate}$$

$$\text{DAGAP} = \frac{da_g}{d\phi} \text{ approximate} \qquad (88)$$

Set the total derivatives equal to the sum of the exact portion and the approximate portion.

Restore the values of J_2 and J_4 in the working array for subroutine GPOT.

IV. Complete the Evaluation of the Encke Equations.

Compute:

$$\text{DFDPHI} = \frac{dF}{d\phi} \qquad (87)$$

$$V1 = V_1 \tag{84}$$

$$AU2 = A_1 u^2, \; VU2 = \frac{V_o}{AU2}, \; VO2 = VU2\ (2 + VU2).$$

$$V3P = V_3' \tag{100}$$

$$V22 = (1 + VU2)^2$$

Calculate

$$E(4) = \frac{dq_n}{d\phi}. \tag{101}$$

$$E(5) = q_n$$

$$E(1) = \frac{dp_n}{d\phi} = \frac{dp}{d\phi}$$

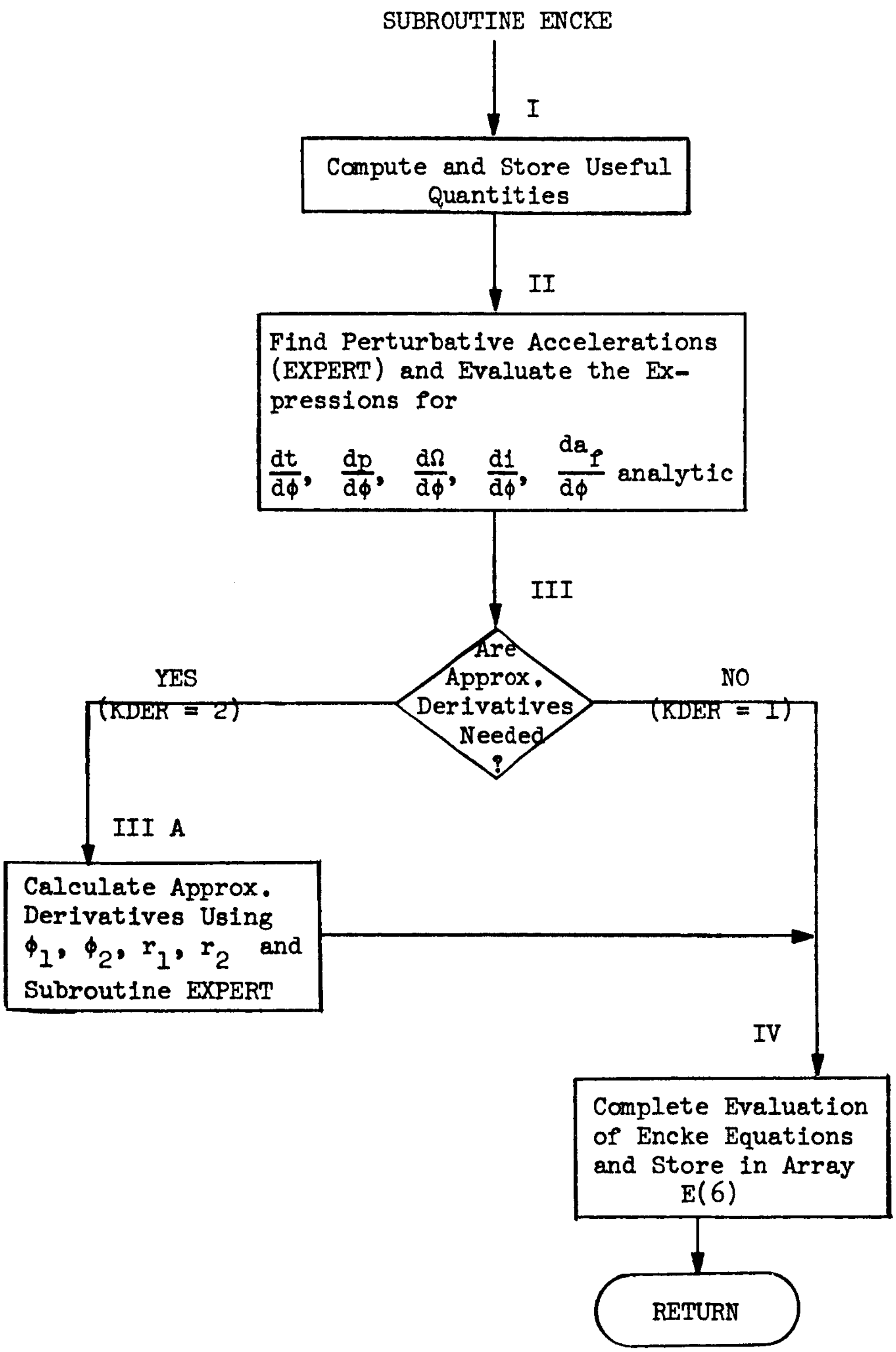
$$E(2) = \frac{d\Omega_n}{d\phi} = \frac{d\Omega}{d\phi} - \frac{d\Omega_a}{d\phi}$$

$$E(3) = \frac{di_n}{d\phi} = \frac{di}{d\phi} - \frac{di_a}{d\phi}$$

$$E(6) = \frac{dt_n}{d\phi} = \frac{dt}{d\phi} - \frac{dt_a}{d\phi}.$$

Return

5.5.2 Detail Flow Chart



Section 5.5.3 Program Listing

The following pages give the listing of subroutine ENCKE.

```
      SUBROUTINE  ENCKE(PT,OMEGT,XIT,QT,UT,T,PHI,LS,DT,N2
     1,XIN,PN,UN,P2,PA,QN,E,AJ2,AJ4,KDER)
      DIMENSION  E(6)
      COMMON /APS/AS(6),AD(6),C2IOC,U1,XIOC,DQ1,XI1/DERIV/DENK(3)
      COMMON/ENERG/      CI2,U3,CT2,U5,DPHIDT, U2/CPOT/COEFF(83)
C   I  COMPUTE AND STORE USEFUL QUANTITIES
      R= 1./UT
      CI = COS(XIT)
      SI = SIN(XIT)
      TI = SI/CI
      CI2= CI* CI
      SI2= SI* SI
      CI3= CI2* CI
      CI4 = CI3* CI
      A1 = PT/CI
      U2= UT*UT
      PT2 = PT* PT
      U5= UT** 5
      SP= SIN(PHI)
      CP= COS(PHI)
      CT = SI* SP
      ST = SQRT(1.-CT*CT)
C   II  FIND PERTURBATIVE ACCELERATIONS
      CALL EXPERT (LS,OMEGT,XIT,PHI,R,T,DT,AF,AG,AH,N2)
      DUDSI = R* ST* AG
      F = R* AF + TI* CP*DUDSI/ST
      DPHIDT = PT* U2/CI   +F*CI3*CT/(PT* SI2*ST)
      DTDPHI = 1./DPHIDT
      DPDPHI = DUDSI/ DPHIDT
      DENOM = PT2* U2*SI2*ST + CI4*CT* F
      RUM = F/DENOM
      DODPHI = -CI3 *CT*RUM
      DIDPHI = -SI2*CI3*CP* RUM
      VO = F*CI3*CT/(PT*SI2*ST)
      DA1PHI =(DPDPHI      +TI*DIDPHI)/CI
      DTHETP = -(CI*SP*DIDPHI +CP*SI)/ST
      CT2=CT*CT
      C2T=2.*CT2-1.
      S2T=2.*ST*CT
      CC7=3.-7.*CT2
      U3=UT*U2
      DAFDPH=DENK(1)    *U3*(UT*C2T*DTHETP+2.*QT*
     1S2T)*(1.+DENK(2)*U2*CC7)+DENK(3)     *U5*S2T*(2.*
     2CC7+7.*UT*S2T*DTHETP)
      DAGDPH=0.0
C III CHECK IF APPROX. VALUES FOR DAF + DAG ARE REQUIRED
      GO  TO (20,21),KDER
C III A CALC. APPROX. VALUES FOR DERIVATIVES
   21 COEFF(2) = 0.0
      COEFF(4) = 0.0
      DPHI = .17453293 E-01
```

```
      PHI1 = PHI -DPHI
      PHI2 = PHI +DPHI
      DELU = AS(4)* DPHI
      R1   = 1./( UT - DELU)
      R2   = 1./(UT + DELU)
      CALL               EXPERT (LS, OMEGT,XIT ,PHI1,R1,T,0,
     1 AF1, AG1, AH1, N2)
      CALL               EXPERT (LS, OMEGT, XIT, PHI2, R2,T,0
     1,AF2, AG2, AH2, N2)
      DAFAP  = ( AF2 - AF1)/  .34906586E -01
      DAGAP  = ( AG2 - AG1)/  .34906586 E-01
      DAFDPH = DAFDPH +DAFAP
      DAGDPH = DAGAP
      COEFF(2) = AJ2
      COEFF(4) = AJ4
C IV  COMPLETE THE EVALUATION OF ENCHE EQS.
   20 DFDPHI =(- F*QT      +  DAFDPH + DAGDPH *CP * TI
     1  + AG * (CP*DIDPHI/CI2 - TI *SP))/UT
      V1 = CT*(DFDPHI - F*(DA1PHI/A1 +2.*DIDPHI/(SI*CI)
     1+DTHETP/(CT*ST)))/(A1*ST*TI*TI)
      AU2 = A1*U2
      VU2 = V0/AU2
      V02 = VU2*(2.+ VU2)
      V3P=-(V02*AD(4)+AH/(A1*AU2))+QT*(1.+VU2)*(2.*QT*VU2/UT-V1/AU2-
     1 DA1 PHI/A1)
       V22=(1.+VU2)**2
      E(4)=(- UN - U1 -SIN(XIT+XIOC)*SIN(XIN+XI1)/PT2
     1- C2IOC*PN*(PA+PT)/(P2*PT2) +V3P -AD(5)- DQ1)/V22
      E(5)= QN
      E(1)= DPDPHI
      E(2)    = DODPHI -AD(2)
      E(3) = DIDPHI - AD(3)
      E(6) =DTDPHI - AD(6)
      RETURN
      END
```

5.6 SUBROUTINE RKTOM (KR, IP, KHALT, TF, HAH, EMIN, EMAX, MFAIL, FDT, DTM, DT, T, PHI)

Subroutine RKTOM Calling Statement

KR	Runge-Kutta flag
IP	Initial point flag
KHALT	Halt flag
TF	Run stop time
HAH	Array of dependent variables and their derivatives; HAH(1) through HAH(6) are dependent variables HAH(7) through HAH(12) are their derivatives
EMIN	Input minimum error allowed
EMAX	Input maximum error allowed
MFAIL	Maximum failures allowed
FDT	Multiplier to decrease computing interval
DTM	Multiplier to increase computing interval
DT	Current value of computing interval
T	Current value of independent variable
PHI	Current value of the angle ϕ which is always kept $\leq 2\pi$.

5.6.1 Equations in Order of Solution

Test Runge-Kutta flag, KR.

If KR = 1, continue below.

If KR = 2, go to IV.

If KR = 3, go to V.

If KR = 4, go to VI.

If KR = 5, go to IB.

I. Test Initial Point Flag, IP.

If IP = 1, continue below.

If IP = 2, go to IC.

A. Initial point calculations.

IP = 2	Increment initial point flag
KHALT = 1	Set halt flag to continue run
KC = 1	Set Simpson's rule flag to signal first cycle computations
KF = 0 KFAIL = 0	Set intermediate and total failure counters to zero
SR(i) = 0	Set Runge-Kutta increments to zero
i = 1,2,..6	

B. Save quantities for Simpson's rule calculations and for use if computing interval selection fails.

Set

$$SS(13) = T$$
$$SS(14) = \phi$$
$$SS(i) = HAH(i)$$
$$\text{for } i = 1, 2, \ldots 12$$

Go to ID.

C. Test Simpson's rule flag, KC.

If KC = 1, set KC = 2 and continue below.
If KC = 2, go to III.

D. Save quantities for ordinary Runge-Kutta use.

Set

$$S(13) = T$$
$$S(14) = \phi$$
$$S(i) = HAH(i)$$
$$\text{for } i = 1, 2, \ldots 12$$

E. Compute the next value of time and determine if it exceeds run stop time.

$$T_n = S(13) + \Delta T$$

If $T_n > TF$, continue below.

If $T_n = TF$, go to IG.

If $T_n < TF$, go to IH.

F. Set $\Delta T = TF - S(13)$.

G. Set halt flag.

$$KHALT = 3$$

H. Complete first pass of Runge-Kutta.
Compute

$$\Delta T_2 = \Delta T/2$$

$$T = S(13) + \Delta T_2$$

Compute Runge-Kutta parameters.

$$RK1(i) = \Delta T \cdot S(i+6)$$

for i = 1, 2, ... 6

Compute new values for quantities.

$$HAH(i) = S(i) + 1/2 \ RK1(i)$$

for i = 1, 2, ... 6

Increment Runge-Kutta flag.

$$KR = 2$$

II. Exit from Subroutine (Return).

III. Perform Accuracy Tests on Integrated Values.
Reset Simpson's rule flag.

$$KC = 1$$

Compute

$$\Delta T_3 = \Delta T/3$$

Set

$$HS(i) = \Delta T_3[SS(i+6)+4S(i+6) + HAH(i+6)]$$

$$\text{for } i = 1, 2, \ldots\ 6$$

Compute estimated and allowable errors.

$$C_{max} = \text{Maximum of } |SR(i)|,\ i = 1, 2, \ldots\ 5$$

$$E_{est} = \text{Maximum of } |SR(i)-HS(i)|,\ i = 1, 2, \ldots\ 5$$

Set Runge-Kutta increments to zero.

$$SR(i) = 0,\quad i = 1, \ldots\ 6$$

$$E_{all} = \text{Maximum of } [E_{max}\, C_{max} \text{ or } 10^{-9} \text{ times the maximum } HAH(i)]$$

$$i = 1, 2, \ldots\ 5$$

$$E_{rmin} = E_{min}\, C_{max}$$

Print the values of T, ΔT, number of intermediate failures,

E_{all}, E_{est}, and E_{rmin}.

Test estimated error versus maximum allowable error.

If $E_{est} > E_{all}$, continue below.

If $E_{est} \leq E_{all}$, go to III D.

A. Increment total failure counter.

KFAIL = KFAIL + 1

Test total failures against maximum allowed.

If KFAIL $\geq$ MFAIL, continue below.

If KFAIL < MFAIL, go to III C.

B. Set halt flag to stop run.

KHALT = 2

Write "computing interval selection fails," exit subroutine at II.

C. Increment intermediate failure counter.

KF = KF+1

Set halt flag to 1.

KHALT = 1

Go to III H.

D. Test estimated error against minimum allowed.

If $E_{est} \leq E_{rmin}$, continue below.

If $E_{est} > E_{rmin}$, go to III G.

E. Increment total failure counter, KFAIL.

$$KFAIL = KFAIL + 1$$

Test total failures against maximum allowed.

If $KFAIL \geq MFAIL$, go to IIIB.

If $KFAIL < MFAIL$, continue below.

F. Increment intermediate failure counter.

$$KF = KF + 1$$

Set halt flag to 1.

$$KHALT = 1$$

Increase ΔT by input multiplier.

$$\Delta T_{new} = DTM \cdot \Delta T_{old}$$

Restore values saved at IB to the ordinary Runge-Kutta values.

$$S(i) = SS(i), \quad i = 1, 2, \ldots. 14$$

Go to IE.

G. Set intermediate failure counter to zero.

H. Compute new allowable computing interval.

$$\Delta T_{new} = (FDT)(\Delta T_{old})[E_{all}/E_{est}]^{1/4}$$

Test $\frac{\Delta T}{T}$ against 10^{-8}.

If $\Delta T/T \leq 10^{-8}$, print "Computing interval = (ΔT)," and go to III B.

If $\Delta T/T > 10^{-8}$, continue below.

J. Test intermediate failure counter, KF.

If KF $\leq$ 0, continue below.

If KF > 0, go to III L.

K. Set KR = 5, and exit to print at II.

L. Restore values saved at IB to ordinary Runge-Kutta values.

S(i) = SS(i), i = 1, 2, ... 13

Go to IH.

IV. Second Pass of Runge-Kutta.

Increment Runge-Kutta flag.

KR = 3

Compute Runge-Kutta parameters and new values of dependent variables.

RK2(i) = (ΔT)(HAH(i+6))

HAH(i) = S(i) + 1/2 RK2(i)

i = 1, 2, ... 6.

Exit subroutine at II.

V. Third Pass of Runge-Kutta.

Increment Runge-Kutta flag.

$$KR = 4$$

Compute new time.

$$T = S(13) + \Delta T$$

$$\phi = \mathrm{mod}(S(14) + \Delta\phi, 2)$$

Compute Runge-Kutta parameters and new values of dependent variables.

$$RK3(i) = (\Delta T)\,(HAH(i+6))$$

$$HAH(i) = S(i) + RK3(i)$$
$$i = 1, 2, \ldots 6$$

VI. Fourth Pass of Runge-Kutta.

Reset Runge-Kutta flag.

$$KR = 1$$

Compute Runge-Kutta integrated values and increments.

$$RKINC(i) = \{RK1(i)+2[RK2(i)+RK3(i)] + (\Delta T)[HAH(i+6)]\}/6$$

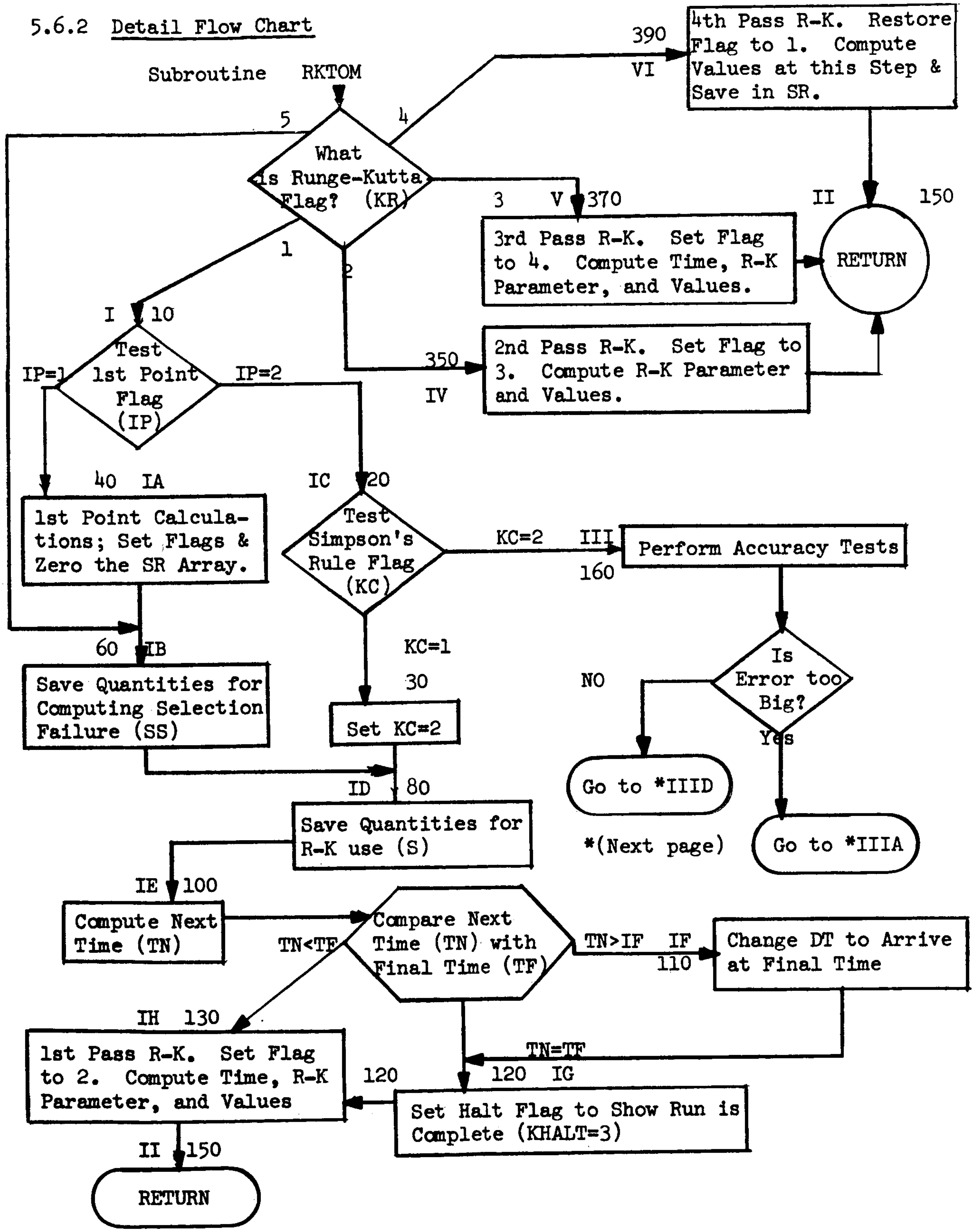
$$SR(i)_{new} = SR(i)_{old} + RKINC(i)$$

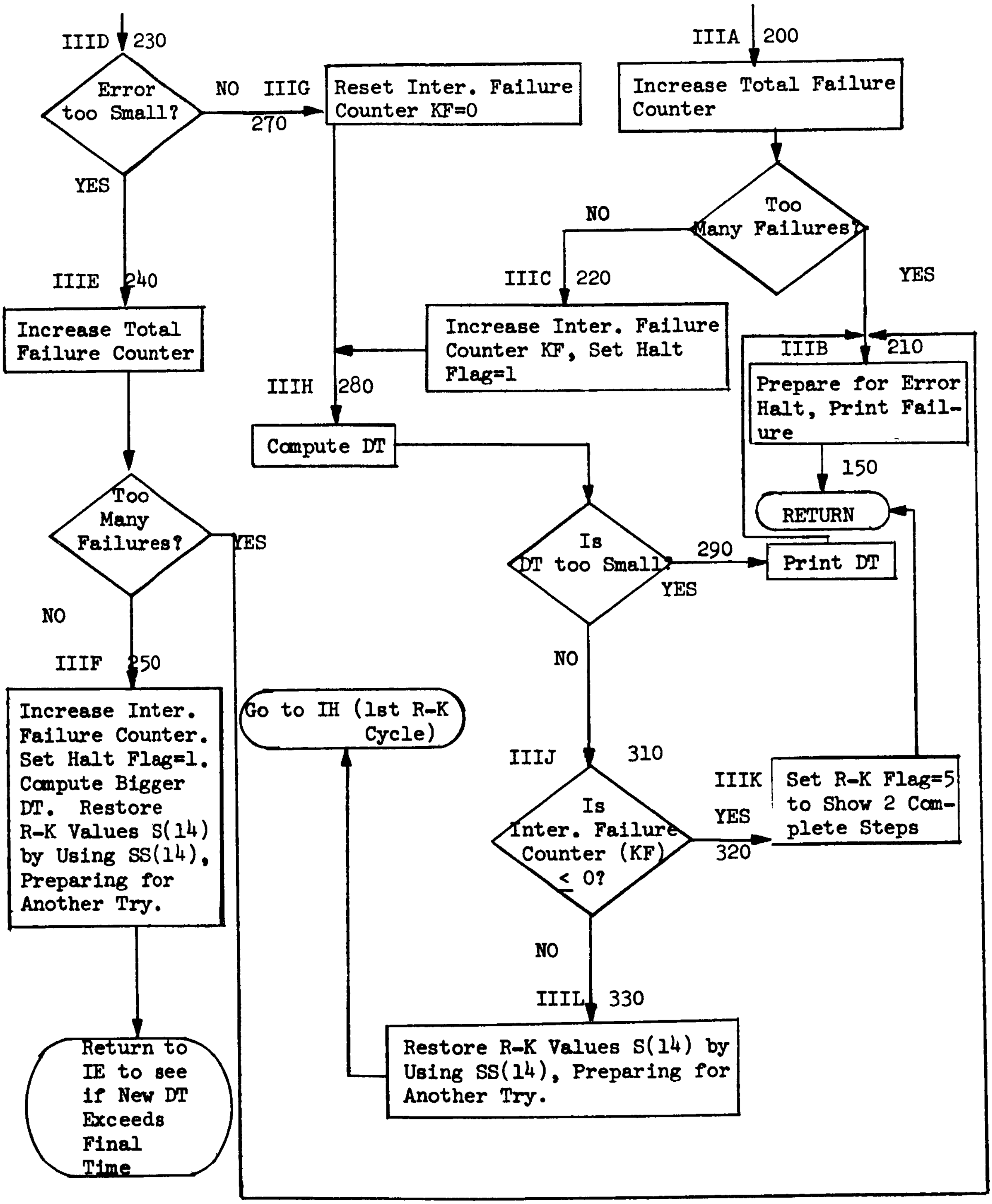
$$HAH(i) = S(i) + RKINC(i)$$

$$\text{for } i = 1, 2, \ldots 6$$

Exit subroutine at II.

5.6.2 Detail Flow Chart




IIID 230
Error too Small?
NO IIIG 270
Reset Inter. Failure Counter KF=0
IIIA 200
Increase Total Failure Counter
Too Many Failures?
NO
YES
IIIE 240
Increase Total Failure Counter
IIIC 220
Increase Inter. Failure Counter KF, Set Halt Flag=1
IIIB 210
Prepare for Error Halt, Print Failure
IIIH 280
Compute DT
150
RETURN
Too Many Failures?
YES
Is DT too Small?
290
Print DT
YES
NO
NO
IIIF 250
Increase Inter. Failure Counter. Set Halt Flag=1. Compute Bigger DT. Restore R-K Values S(14) by Using SS(14), Preparing for Another Try.
Go to IH (1st R-K Cycle)
IIIJ 310
Is Inter. Failure Counter (KF) ≤ 0?
IIIK
YES 320
Set R-K Flag=5 to Show 2 Complete Steps
NO
IIIL 330
Restore R-K Values S(14) by Using SS(14), Preparing for Another Try.
Return to IE to see if New DT Exceeds Final Time

Section 5.6.3 <u>Program Listing</u>

The following pages give the listing of subroutine RKTOM.

```
      SUBROUTINE RKTOM (KR, IP, KHALT, TF, HAH, FMIN, FMAX,
     1 MFAIL,      FDT, DTM, DT, T,    PHI)
      DIMENSION HAH(12), S(14), SS(14), SR(6), HS(6), RK1(6),RKINC(6)
     1,RK2(6), RK3(6)
C     TEST RUNGE-KUTTA FLAG
      GO TO (10, 350, 370, 390, 60), KR
C I   TEST FIRST POINT FLAG
   10 GO TO ( 40, 20), IP
C IC  TEST SIMPSONS RULE FLAG
   20 GO TO (30, 160), KC
   30 KC = 2
      GO TO 80
C IA  FIRST POINT CALCULATIONS
   40 IP = 2
      KHALT = 1
      KC = 1
      KF = 0
      KFAIL = 0
      DO 50 I = 1, 6
   50 SR(I) = 0.
C IB  SAVE QUANTITIES USED IF COMP INT SELECTION FAILS
   60 SS(13) = T
      SS(14)=PHI
      DO 70 I = 1, 12
   70 SS(I) = HAH(I)
C ID  SAVE QUANTITIES FOR ORDINARY RUNGE-KUTTA USE
   80 S(13) = T
       S(14)=PHI
      DO 90 I = 1, 12
   90 S(I) = HAH(I)
C IE  COMPUTE NEXT TIME AND DETERMINE IF IT EXCEEDS STOP TIME
  100 TN = S(13) + DT
      IF (TN - TF) 130, 120, 110
C IF
  110 DT = TF - S(13)
C IG
  120 KHALT = 3
C IH  COMPLETE 1ST R-K PASS, COMPUTE NEW TIME AND POSITIONS
  130 DT2 = DT / 2.
      T = S(13) + DT2
      PHI=S(14)+DT2
      DO 140 I = 1, 6
      RK1(I) = DT * S   (I+6)
  140 HAH(I) = S(I) + .5 * RK1(I)
      KR = 2
C II
  150 RETURN
C III PERFORM ACCURACY TESTS ON INTEGRATED VALUES
  160 KC = 1
      DT3 = DT / 3.
C     COMPUTE SIMPSONS RULE INTEGRATED VALUES
```

```
      DO 170 I = 1, 6
  170 HS(I) = DT3 * (SS(I+6) + 4. * S(I+6) + HAH(I+6))
C     COMPUTE ESTIMATED AND ALLOWABLE ERRORS
      CMAX = AMAX1 (ABS (SR(1)), ABS (SR(2)), ABS (SR(3)),
     1                ABS (SR(4)), ABS (SR(5)))
      ESTER = AMAX1 (ABS (SR(1) - HS(1)), ABS (SR(2) - HS(2)
     1 ), ABS (SR(3) - HS(3)), ABS (SR(4) - HS(4)),
     2    ABS (SR(5) - HS(5)))
 1000 DO 180 I = 1, 6
  180 SR(I) = 0.
      EALL = AMAX1 (EMAX * CMAX, 1.E-9 * AMAX1 (ABS (HAH(1))
     1,ABS (HAH(2)), ABS (HAH(3)), ABS (HAH(4)),
     2 ABS (HAH(5))))
      ERMIN = EMIN * CMAX
      WRITE (6, 190) S(13),DT, KF, EALL, ESTER, ERMIN
  190 FORMAT (1H 2E20.8, I12, E27.8, 2E20.8)
      IF (ESTER - EALL) 230, 230, 200
C IIIA
  200 KFAIL = KFAIL + 1
      IF (KFAIL - MFAIL) 220, 210, 210
C III B  EXIT TO HALT RUN
  210 KHALT = 2
      WRITE (6, 215)
  215 FORMAT (1H0,35H COMPUTING INTERVAL SELECTION FAILS)
      GO TO 150
C IIIC
  220 KF = KF + 1
      KHALT = 1
      GO TO 280
C IIID
  230 IF (ESTER - ERMIN) 240, 240, 270
C IIIE
  240 KFAIL = KFAIL + 1
      IF (KFAIL - MFAIL) 250, 210, 210
C IIIF
  250 KF = KF + 1
      KHALT = 1
      DT = DTM * DT
      DO 260 I = 1, 14
  260 S(I) = SS(I)
      GO TO 100
C IIIG
  270 KF = 0
C III H   COMPUTE NEW ALLOWABLE COMPUTING INTERVAL
  280 DT = FDT * DT * (EALL / ESTER) ** 0.25
      IF (DT / T - 1.E-8) 290, 290, 310
  290 WRITE (6, 300) DT
  300 FORMAT(1H0,16H COMP INTERVAL = E17.8)
      GO TO 210
C IIIJ
  310 IF (KF) 320, 320, 330
```

```
C III K   EXIT TO PRINT
  320 KR = 5
      GO TO 150
C IIIL
  330 DO 340 I = 1, 14
  340 S(I) = SS(I)
      GO TO 130
C IV  2ND PASS OF RUNGE-KUTTA
  350 KR = 3
      DO 360 I = 1, 6
      RK2(I) = DT * HAH(I+6)
  360 HAH(I) = S(I) + .5 * RK2(I)
      GO TO 150
C V   3RD PASS OF RUNGE-KUTTA
  370 KR = 4
      T = S(13) + DT
       PHI=AMOD(S(14)+DT,6.2831853)
      DO 380 I = 1, 6
      RK3(I) = DT * HAH(I+6)
  380 HAH(I) = S(I) + RK3(I)
      GO TO 150
C VI  4TH PASS OF RUNGE-KUTTA
  390 KR = 1
      DO 400 I = 1, 6
      RKINC(I)=(RK1(I)+2.*(RK2(I)+RK3(I))+DT*HAH(I+6))/6.
      HAH(I)=S(I)+RKINC(I)
  400 SR(I)=SR(I)+RKINC(I)
      GO TO 150
      END
```

5.7 FUNCTION ELIPE

5.7.1 Equations in Order of Solution

The quarter-period K of the elliptic integral $F(\phi,k)$ is evaluated by sucessive application of the decreasing Landen Transformation. From reference 3, equation 17.5.7 and 17.5.1:

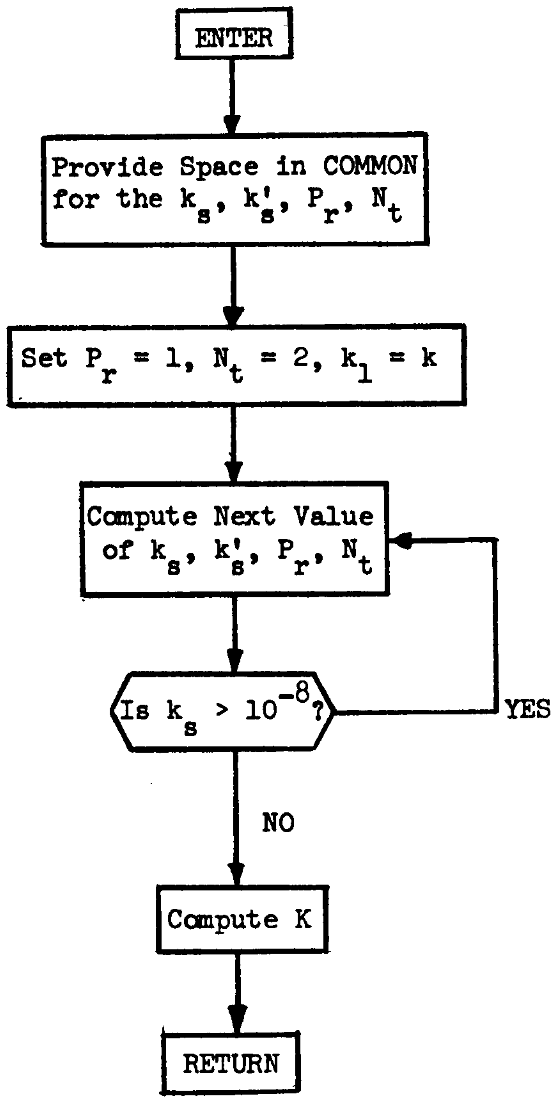
$$K = \frac{1}{2}\pi \prod_{s=1}^{\infty} (1 + k_s)$$

$$k_{s+1} = \left(\frac{k_s}{1+\sqrt{1-k_s^2}}\right)^2$$

where $\sin \alpha_s$ in reference 3 is replaced by k_s. The k_s are decreasing very rapidly. Even for $k_o = .99995$, seven steps are sufficient to make k_7 less than 10^{-8}. Therefore a maximum of 10 steps is suggested. The k_s, $k_s' = \sqrt{1-k_s^2}$, $Pr = \prod_{s=1}^{N_t} (1+k_s)$, and N_t are stored in COMMON because they will be needed in the computation of the elliptic integral $F(\phi,k)$. (N_t = Number of transformations.)

5.7.2 Detail Flow Chart

FUNCTION ELIPE (k)



Section 5.7.3 Program Listing

The following page gives the listing of function subprogram ELIPE.

```
      FUNCTION   FLIPE ( CAY)
C     FUNCTION ELIPE  TO COMPUTE THE QUARTER PERIOD
      COMMON /QUART/  CAP (10),PR,NOT,CA(10)
      PR = 1.
        NOT =2
      CA(1) = CAY
      DO 300  I = 2,10
      NOT = I
      IM1  =   I - 1
      CAP(IM1) =  SQRT (1. - CA(IM1)* CA (IM1))
      CA (I) = ( CA(IM1) / ( 1. + CAP (IM1))) ** 2
      PR=PR+PR*CA(I)
C      TEST  LAST FACTOR OF THE PRODUCT
      IF (CA(I)-.1E-07) 400,400,300
  300  CONTINUE
  400  ELIPE  = 1.5707963  * PR
      RETURN
      END
```

5.8 FUNCTION ELI

5.8.1 Equations in Order of Solution

The eliptic integral $F(\phi,k)$ is evaluated by successive application of Landen's decreasing transformation. From reference 3, equations 17.5.8, 17.5.6, and 17.5.2:

$$F(\phi,k) = \frac{2}{\pi} \cdot K \cdot \lim_{n\to\infty} \frac{\phi_n}{2^n}$$

$$\phi_{n+1} = \phi_n + \arctan\left(\sqrt{1+k_n^2} \cdot \tan\phi_n\right)$$

The k_n are stored in COMMON and K is known from function ELIPE.

The quantity $\Delta\phi = \phi_{n+1} - \phi_n = \arctan\left(\sqrt{1+k_n^2} \cdot \tan\phi_n\right)$ is computed at each step and added to ϕ_n. The quadrant of $\Delta\phi$ is the same as the quadrant of ϕ_n. To accomplish this, $\Delta\phi$ is written as

$$\Delta\phi = \Delta_s + \Delta_i$$

where Δ_i is 2π-times the number of revolutions completed by ϕ_n, and Δ_s is the remainder, determined by QUAD1 so that

$$\tan\Delta\phi = \sqrt{1-k^2}\tan\phi_n$$

After adding this value of $\Delta\phi$ to ϕ_n, the total is modded with 2π giving ϕ_{si} which preserves small arguments for the next step.

According to the above limit approach, the iteration process is halted when

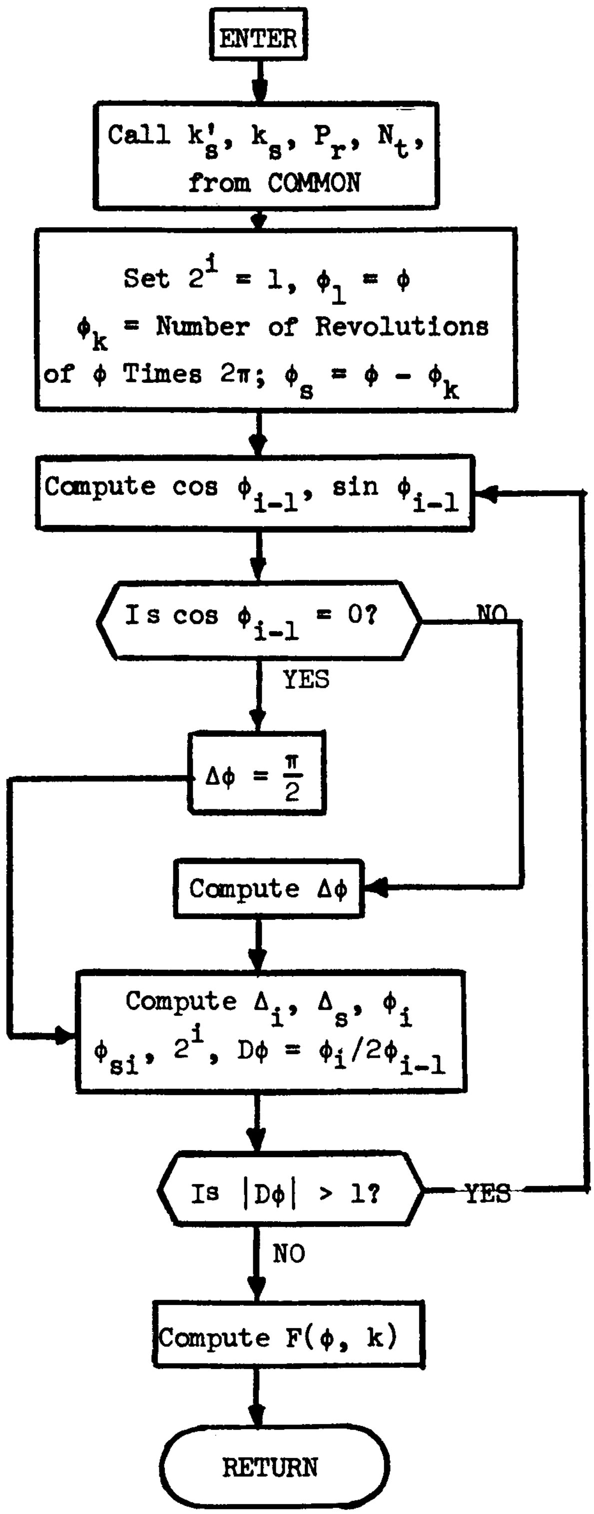
$$\phi_{n+1} = 2\phi_n$$

or when

$$D\phi = \left|\phi_{n+1}/2\phi_n\right| - 1 = 0.$$

5.8.2 Detail Flow Chart

FUNCTION ELI (ϕ, K)



Section 5.8.3 Program Listing

The following page gives the listing of function subprogram ELI.

```
      FUNCTION   ELI ( PHI       , QP )
C     FUNCTION  ELI  TO COMPUTE THE VALUE OF THE ELLIPTIC
C     INTEGRAL
      DIMENSION  PHILT (10) , PHIS(10)
      COMMON/QUART/CAP(10),PR,NOT,CA(10)
      TWOPI =6.2831853
      PIO2 = 1.5707963
      PI =3.1415927
      NPTWO  = 1
      PHILT(1)     = PHI
      PHIK      =AINT(PHI/TWOPI)*TWOPI
      PHIS(1) = PHI - PHIK
      DO 600 I=2,NOT
      IMI1 =  I-1
      COSP = COS (PHIS  (IMI1))
      SINP = SIN (PHIS  (IMI1))
      IF(COSP)  580,590,580
  590 DELPHO  =  PIO2
      GO TO  550
  580 DELPHO  =  ATAN(ABS(CAP(IMI1)* (SINP / COSP )))
  550 DELI  = AINT (PHILT(IMI1)/ TWOPI) * TWOPI
      DELS  = QUAD1( DELPHO , PHILT(IMI1), PIO2 ,PI,TWOPI)
      PHILT(I) =  PHILT(IMI1) + DELS + DELI
      PHIS(I)  =  PHIS(IMI1) + DELS
      PHIS(I) = PHIS(I) -AINT(PHIS(I)/TWOPI)*TWOPI
      NPTWO= NPTWO * 2
      DPH = ABS  (PHILT(I) /(PHILT(IMI1) * 2.))
      IF (DPH - 1.)   600 , 500 , 500
  600 CONTINUE
  500 TWON  = NPTWO
      N=IMI1+1
       ELI=PHILT(N)*QP*.63661977/TWON
      RETURN
      END
```

5.9 FUNCTION ELIF

5.9.1 Equations in Order of Solution

The evaluation of the elliptic function $sn(u,k)$ is accomplished by the use of formulae (16.12.1) and (16.12.2) of reference 3:

$$sn(u,m) = \frac{(1+\mu^{1/2})\ sn(v,\mu)}{1+\mu^{1/2}\ sn^2(v,\mu)}$$

$$\mu = \left(\frac{1-\sqrt{1-k^2}}{1+\sqrt{1-k^2}}\right)^2 = \left(\frac{k}{1+\sqrt{1-k^2}}\right)^4 \quad , \quad v = \frac{u}{1+\mu^{1/2}}$$

The above transformation from v,μ to u,m is repeated until the modulus is zero. Thus, we have in general:

$$sn(u_{n-1}, k_{n-1}) = \frac{(1+k_n)\ sn(u_n,k_n)}{1+k_n\ sn^2(u_n,k_n)} \quad , \ n = 1,\ 2,\ \ldots.$$

where the modulus k is used rather than m $(k^2=m)$

$$u_n = \frac{u_o}{\prod_{i=1}^{\infty} (1+k_i)}$$

and the k_i have been calculated in function ELIPE and are stored in COMMON. The procedure of computing $sn(u_o,k_o)$ is as follows.

The number of transformations (NOT) is chosen such that k_n is sufficiently small to permit the approximation:

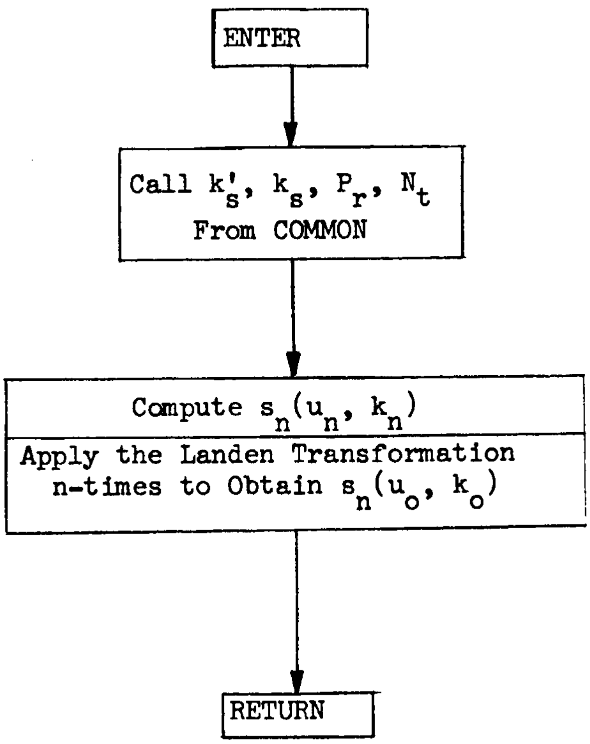
$$sn(u_n,k_n) = \sin u_n - \frac{1}{4} k_n^2(u_n - \sin u_n \cos u_n) \cos u_n$$

(equation 16.13.1, reference 3)

Then starting with $sn(u_n,k_n)$, the recursive formula is applied n-times. After the n^{th} step, the value of $sn(u_o,k_o) = sn(u,k)$ is obtained.

5.9.2 Detail Flow Chart

FUNCTION ELIF (u)



Section 5.9.3 Program Listing

The following page gives the listing of function subprogram ELIF.

```
      FUNCTION  ELIF ( U )
C     FUNCTION ELIF  TO CALCULATE  THE ELLIPTIC FUNCTION SN
      COMMON / QUART/CAP(10) ,PR   , NOT ,CA(10)
      EM = CA(NOT)**2
      VEO  =  U/PR
      SVE = SIN(VEO)
      CVE = COS(VEO)
      SNVE =  SVE     -.25 * EM     *(VEO -SVE *CVE)*CVE
      NOT2 = NOT +2
      DO 150  I=  2 , NOT
      IR = NOT2 -I
      CAS = CA(IR)* SNVE
  150 SNVE = (SNVE +CAS)/(1.+ CAS*SNVE)
      ELIF  =  SNVE
      RETURN
      END
```

5.10 SUBROUTINE GPOT (Q, CT, EW, R, AF, AG, AH)

This subroutine calculates perturbative accelerations a_f, a_g, and a_h due to the earth's potential. The inputs are sin θ, cos θ, longitude of satellite, and non-dimensional radius. The subroutine obtains coefficients of the potential from labeled common /CPOT/.

5.10.1 Equations in Order of Solution

I. Set Original Recursion Values.

$$P(1) = \rho_o = 1$$

$$DP(2) = \rho_1' = 1$$

U01 = 0 [Stores zero before array U to become U(0,1)]

$$U(1,1) = U_{11} = 1$$

$$RX(1) = r^3$$

$$P(2) = \rho_1 = \cos\theta$$

II. Set Sum Limits and Zero Original Sum Quantities.

$$NN1 = N1 + 1$$

Zero locations to gather sums of zonal coefficients for a_f, zonal coefficients for a_h, tesseral and sectorial contribution to a_f, tesseral and sectorial contribution to a_g, and tesseral and sectorial contribution to a_h. These are, respectively, Z = 0, Z1 = 0, TS = 0, TS1 = 0, and TS2 = 0.

Calculate the arrays for ρ_n and ρ'_n (P and DP).

Calculate the array of $r^{(n+2)}$, (RX).

Calculate the ratio-array $\frac{J_n}{r^{(n+2)}}$, (AOR).

Find the sum:

$$Z = \sum_{n=2}^{N1} \left(\frac{J_n}{r^{n+2}}\right) \rho_n'$$

and

$$Z1 = \sum_{n=2}^{N1} (n+1) \left(\frac{J_n}{r^{n+2}}\right) \rho_n$$

III. Are Tesserals Required?

If the limit on the tesserals (N2) is less than 2, tesserals are not required, go to VI; otherwise, continue.

IV. Calculate Quantities for Tesseral and Sectorial Sums.

Calculate arrays for sine and cosine of $n \cdot$ longitude (SBE and CBE). Calculate and store the arrays for U_{nm} and W_{nm}.

V. Sum Appropriate Tesserals and Sectorials.

Calculate and store arrays for:

$$CC(N,M) = C_{nm} \cos(m\,\lambda) \equiv CC_{nm}$$

$$CS(N,M) = C_{nm} \sin(m\,\lambda) \equiv CS_{nm}$$

$$SC(N,M) = S_{nm} \cos(m\,\lambda) \equiv SC_{nm}$$

$$SS(N,M) = S_{nm} \sin(m\,\lambda) \equiv SS_{nm}$$

Find the sums:

$$TS = -\sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{W_{nm}}{r^{n+2}} (CC_{nm} + SS_{nm})$$

$$TS1 = \sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{m}{r^{n+2}} U_{nm} (CS_{nm} - SC_{nm})$$

$$TS2 = -\sum_{n=2}^{N2} \sum_{m=1}^{n} \frac{n+1}{r^{n+2}} U_{nm} (CC_{nm} + SS_{nm})$$

VI. Calculate perturbative accelerations.

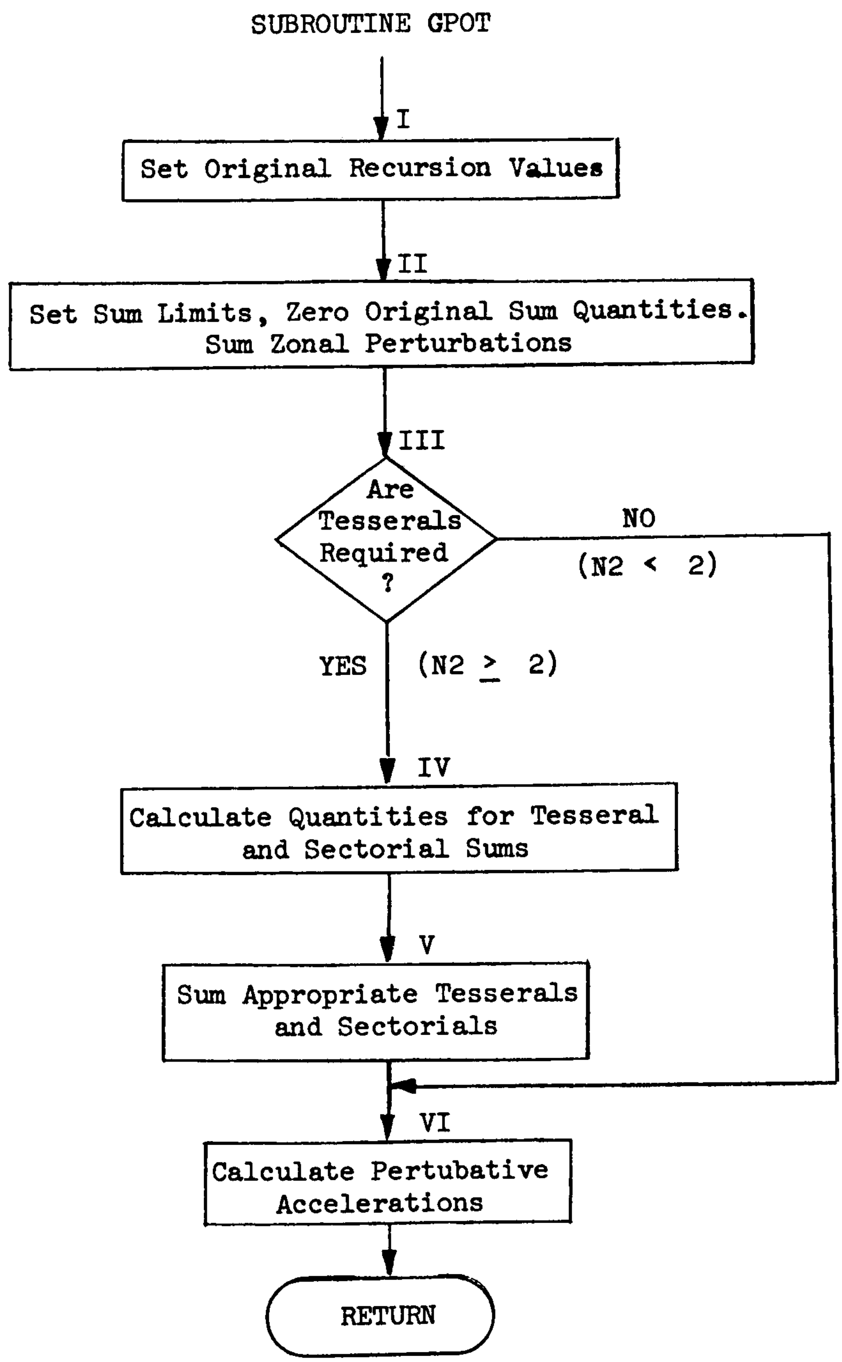
$$a_f = AF = Z \cos \theta + TS$$

$$a_g = -TS1 = AG$$

$$a_h = AH = Z1 + \cos \theta \; TS2$$

Return.

5.10.2 Detail Flow Chart



Section 5.10.3 Program Listing

The following pages give the listing of subroutine GPOT.

```
      SUBROUTINE  GPOT(Q ,CT,FW ,R   ,AF,AG,AH)
C     SUBROUTINE  GPOT  TO COMPUTE THE ACCELERATIONS DUE TO
C     THE HIGHER HARMONICS ,TESSERALS AND SECTORIALS OF THE
C     EARTHS POTENTIAL
      COMMON / CPOT/ AJ(9),C(6,6), S(6,6), N1,N2
     1 /G/U01,U(6,6)
      DIMENSION  P(10),DP(10),CBE(6),   W(6,6), CC(6,6),
     1CS(6,6), SC(6,6), SS(6,6),SBE(6) ,RX(9),AOR(9)
C     I
C  SET ORIGINAL RECURSION VALUES
      P(1) = 1.
      DP(2) = 1.
      U01=0.0
      U(1,1) = 1.
      RX(1)=R**3
      P(2)=CT
C     II
C  SET SUM LIMITS AND ZERO ORIGINAL SUM QUANTITIES
      NN1 = N1 + 1
  143 Z = 0.
      Z1= 0.
      TS = 0.
      TS1= 0.
      TS2= 0.
C  CALC. RHO,RHO- AND ZONAL SUMS
      DO 125  N = 3,NN1
      D = N
      L = N - 1
      A = L
      P(N) = ((2.*A-1.)*P(2)*P(L)-(A-1.)*P(N-2))/A
      DP(N) = P(2)*DP(L) + A*P(L)
      RX(L)=RX(L-1)*R
      AOR(L)=AJ(L)/RX(L)
      Z=Z+  AOR(L)*DP(N)
  125  Z1=Z1+D*AOR(L)*P(N)
C III ARE TESSERALS REQUIRED
      IF(N2-1)30,30,40
C IV CALCULATE QUANTITIES FOR TESSERAL AND SECTORIAL SUMS
   40 SBE(1)=SIN(FW)
      CBE(1)=COS(FW)
      DO 126 N=2,N2
      K=N-1
      D=N
      BEW=D*FW
      CBE(N)=COS(BEW)
   10 SBE(N)=SIN(BEW)
      U(N,N) = (2.*D-1.)*Q*U(K,K)
      U(K,N) = 0.
      W(N,N) = -D*P(2)*U(N,N)
      DM1=D-1.
      DTI21=(2.*D-1.)*P(2)
```

```
      DTIP2=D*P(2)
      DO 126  M=1,K
      B = M
      U(N,M)=(DTI21*U(K,M)-(DM1+B)*U(N-2,M))/(D-B)
  126 W(N,M) = -DTIP2 *U(N,M) +(B+D)* U(K,M)
CV SUM TESSERALS AND SECTORIALS
      DO  242  N=2,N2
      D = N
      DO  242  M=1,N
      B = M
      CC(N,M) = C(N,M) * CBF(M)
      CS(N,M) = C(N,M) * SBF (M)
      SC(N,M) = S(N,M) * CBF(M)
  135 SS(N,M) = S(N,M) * SBF(M)
  228 TS = TS-(W(N,M)/RX(N)    )*(CC(N,M)+SS(N,M))
  232 TS1=TS1+(B/RX(N)    )*U(N,M)*(CS(N,M)-SC(N,M))
  242 TS2=TS2-  ((D+1.)/RX(N)    )*U(N,M)*(CC(N,M)+SS(N,M))
C VI  CALCULATE PERTURBATIVE ACCELERATIONS
   30 AF =Z*Q+ TS
      AG = -TS1
      AH = Z1+ TS2*Q
      RETURN
      END
```

5.11 FUNCTION QUAD1 (OMEGA, W, QPER, PI, TWOPI)

QUAD1 is the angle which is in the same quadrant as W (with respect to QPER) and |tan (QUAD1)| = tan (OMEGA). All inputs and outputs in radians.

5.11.1 Equations in Order of Solution

I. Adjust W so $-4 \cdot QPER \leq W \leq 4 \cdot QPER$.
(Mod W with 4·QPER)

II. If W is negative, go to IIA; otherwise go to IIB.

A. Make W the equivalent positive angle by adding the total period 4·QPER.

B. Find IW, which is the quadrant of W with respect to QPER.

$$IW = \text{Integer part of } \left(\frac{W}{QPER}\right) + 1$$

IW is then 1, 2, 3, or 4.

III. If W is in the first quadrant (IW = 1), go to IIIA.

If W is in the second quadrant (IW = 2), go to IIIB.

If W is in the third quadrant (IW = 3), go to IIIC.

If W is in the fourth quadrant (IW = 4), go to IIID.

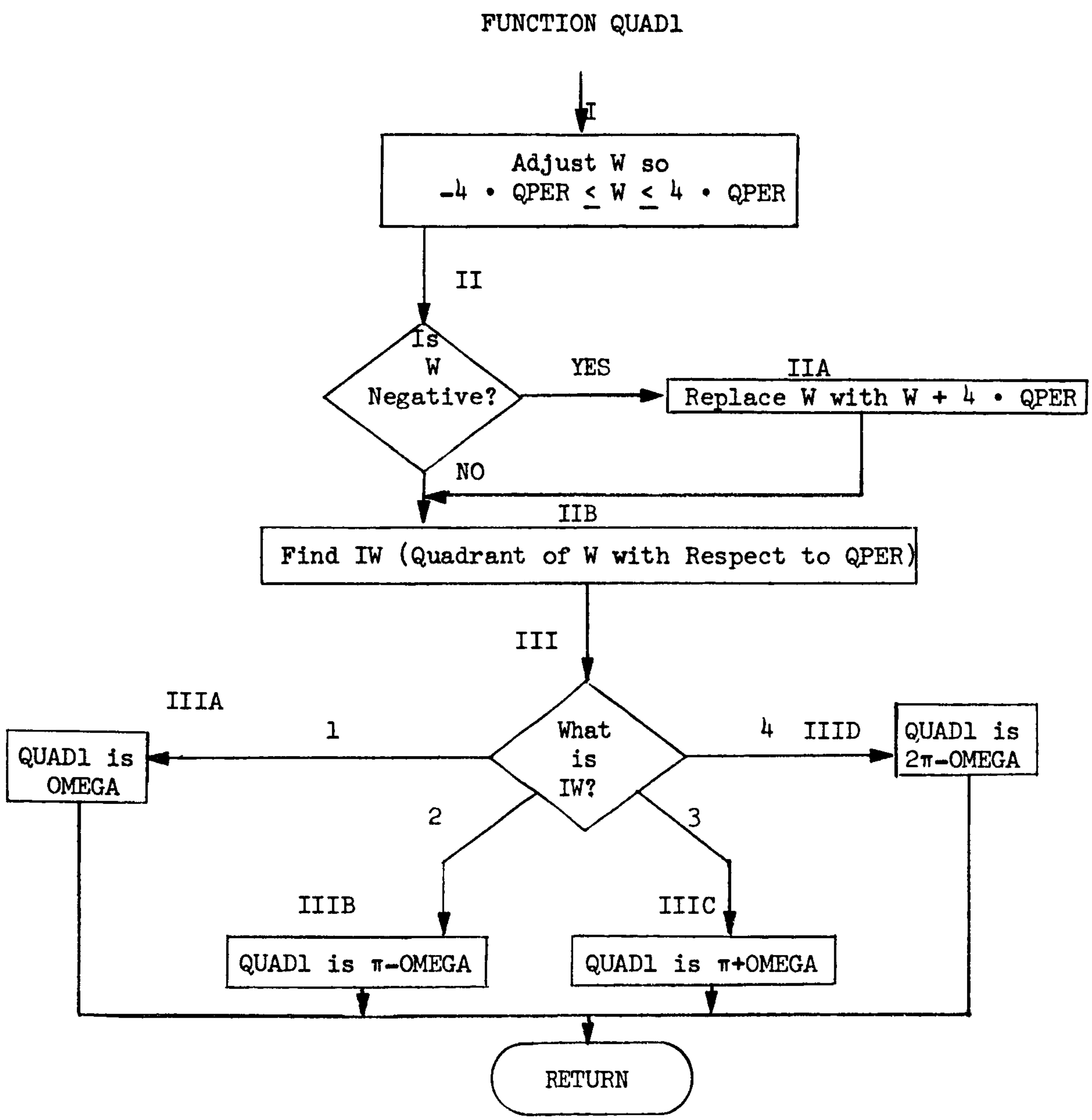
A. Set QUAD1 = OMEGA, and return.

B. Set QUAD1 = π - OMEGA, and return.

C. Set QUAD1 = π + OMEGA, and return.

D. Set QUAD1 = 2π - OMEGA, and return.

5.11.2 Detail Flow Chart



Section 5.11.3 Program Listing

The following page gives the listing of function subprogram QUAD1.

```
      FUNCTION  QUAD1(OMEGA,W,QPER,PI,TWOPI)
C I   ADJUST  W
      W = AMOD (W,(4.*QPER))
C II  CHECK  W  SIGN
      IF (W)  20,21,21
C IIA MAKE  W  EQUIVALENT POSITIVE  ANGLE
   20 W = W +(4.*QPER)
C IIB FIND QUADRANT OF W
   21 IW = IFIX (W/QPER) +1
C III CHECK  QUADRANT
      GO TO (31,32,33,34),IW
C IIIA
   31 QUAD1= OMEGA
      RETURN
C IIIB
   32 QUAD1= PI- OMEGA
      RETURN
C IIIC
   33 QUAD1= PI+ OMEGA
      RETURN
C IIID
   34 QUAD1= TWOPI -OMEGA
      RETURN
      END
```

5.12 FUNCTION QUAD2 (XW,Z1,K,K1OR3,PI) - Adjusts the Quadrant of XW to Agree with the Conditions of Case 1 for the Perigee Calculation

Inputs are ω, angle z_1, quarter-period K, flag that indicates whether ω oscillates around $\frac{\pi}{2}$ or $\frac{3\pi}{2}$, and π. All angles are in radians.

5.12.1 Equations in Order of Solution

I. Adjust z_1 so $-4K \leq z_1 \leq 4K$.
(Mod z_1 with 4K)

II. If z_1 is then negative, go to IIA; otherwise go to IIB.

A. Make z_1 the equivalent positive angle by adding the total period 4K.

B. Find L, which is the quadrant of z_1 with respect to K.

$$L = \text{Integer part of } \left(\frac{z_1}{K}\right) + 1$$

L is then 1, 2, 3, or 4.

III. If L=1 or 4, no change is necessary in ω; go to IIIB.
If L=2 or 3, go to IIIA.

A. Place ω in the second quadrant by replacing ω with $\pi-\omega$, since the magnitudes of the tangents of the two angles must be equal.

B. The input quantity K1OR3 determines if ω oscillates around $\frac{\pi}{2}$ or $\frac{3\pi}{2}$. (K1OR3 = 1 or 2, respectively).

If $\frac{\pi}{2}$ is the oscillation point, go to V.

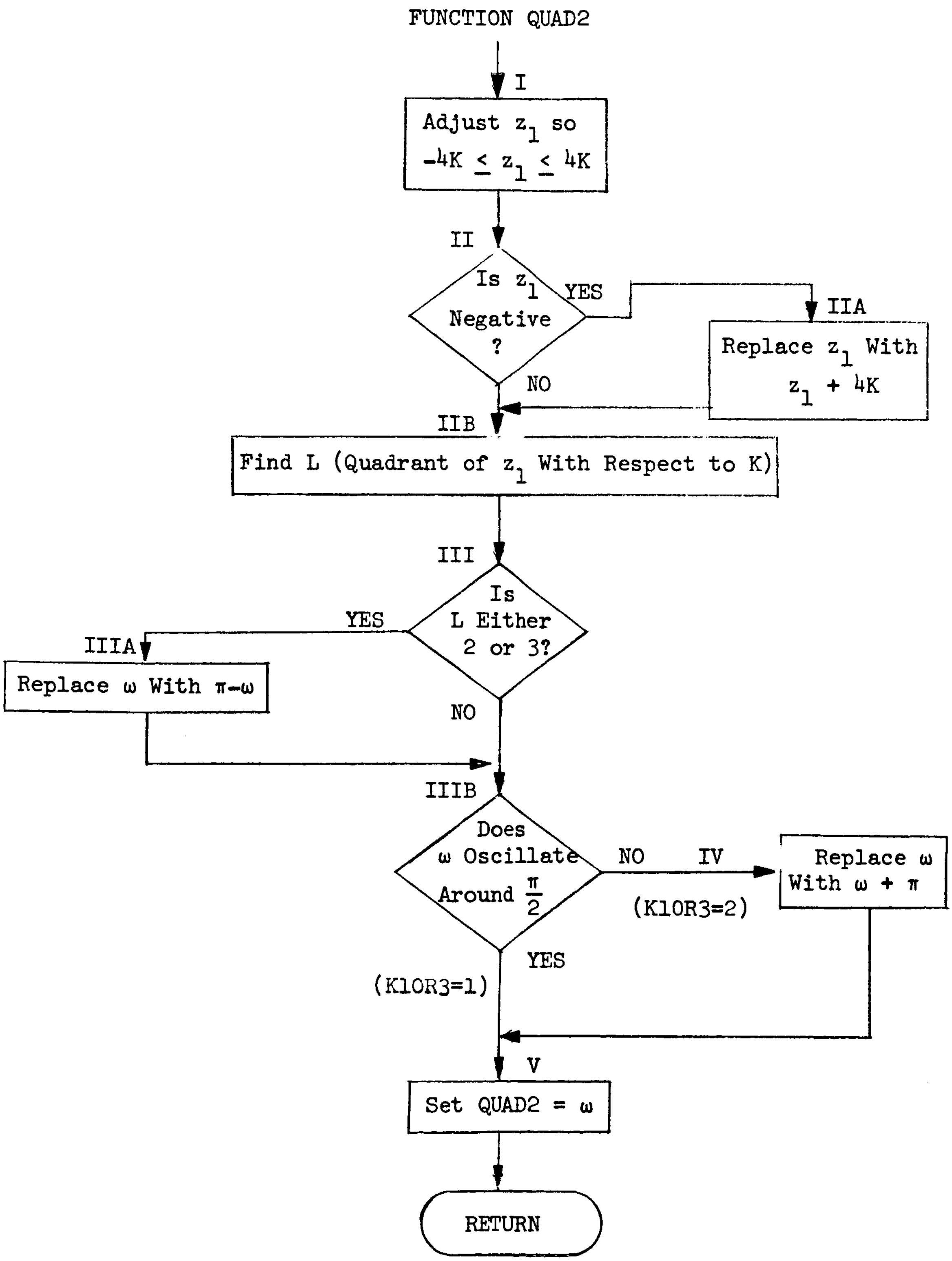
If $\frac{3\pi}{2}$ is the oscillation point, go to IV.

IV. Replace ω by $\pi + \omega$ so the oscillation will be around $\frac{3\pi}{2}$.

V. Set QUAD2 = ω.

Return.

5.12.2 Detail Flow Chart



Section 5.12.3 Program Listing

The following page gives the listing of function subprogram QUAD2.

```
      FUNCTION  QUAD2(XW,Z1,QPER,K1OR3,PI)
C     I  ADJUST  Z1
      Z1=AMOD (Z1,(4.*QPER))
C     II CHECK  Z1 SIGN
      IF (Z1) 20,21,21
C     II A   MAKE Z1 EQUIVALENT POSITIVE ANGLE
   20 Z1 = Z1 + (4.* QPER)
C     II  B  FIND QUADRANT OF  Z1
   21 L=  IFIX(Z1/QPER) + 1
C     III   CHECK  QUADRANT
      GO TO (31,30,30,31), L
C     III  A  PLACE OMEGA IN QUADRANT  2
   30 XW = PI - XW
C     III B  CHECK OSCILLATION  POINT
   31 GO  TO (50,40),K1OR3
C     IV   MAKE  OMEGA  OSCILLATE AROUND  3PI/2
   40 XW = XW + PI
C     V   PREPARE  FOR  RETURN
   50 QUAD2 = XW
      RETURN
      END
```

5.13 MODES OF INPUT AND OUTPUT

5.13.1 Input

This program has only load sheet input through subroutine INPUT 1. The card format is:

column 1 - a one

columns 2 through 6 - location number of piece of data

columns 7 through 15 - input number

columns 16 and 17 - location of decimal place from beginning of field positive if to the right

Three other pieces of data may be entered on the card. The location numbers are punched in columns 18-22, 34-38, and 50-54. The data are punched in columns 23-31, 39-47, and 55-63. The exponents, as explained above, are punched in columns 32-33, 48-49, and 64-65, respectively. The remaining information required is:

columns 66-68, zeros

columns 69-70, reference run number

columns 71-73, case number.

This routine allows identification on the card of each piece of input data by relative location number; only-non-zero numbers need be entered. It has a "Reference Run," "Case" setup. If the case number (card columns 71 to 73) is non-zero, but the reference run number (card columns 69, 70) is zero, then the data on the load sheet are assumed to be sufficient and the case is computed. If the case number is zero and the reference run number is non-zero, the data are stored in array RR and no case is attempted. If the following load sheets with non-zero case numbers have also the reference run number of the stored array, then a case is run using the input of array RR as modified by the new load sheet.

The order of stacking cases is then:

1. All cases with zero reference run number
2. First reference run (zero case number)
3. All cases with first reference run number and non-zero case number
4. Second reference run (zero case number)

•

•

•

The total input array utilizes 102 locations. The locations and quantities are listed below. All input quantities are non-dimensional unless otherwise noted.

Location	Quantity	Remarks
1 - 9	J_n	Leave $J_1 = 0$
10-45	$C_{m,n}$	Arranged in column-sort in 6x6 array
46-81	$S_{m,n}$	Arranged in column-sort in 6x6 array
82	No. of zonals, N1	Integer, $0 \leq N1 \leq 9$
83	No. of tesserals, N2	Integer, $0 \leq N2 \leq 6$, If set = 0 or 1, no tesserals or sectorials are considered
84	Initial value of polar component of angular momentum, p	
85	Initial eccentricity, e_o	

86	Initial argument of perigee, w	In degrees
87	Initial time, t_o	In hours
88	Initial ϕ	In degrees
89	Approximate initial inclination, i_{oo}	In degrees
90	Approximate initial argument of node, L_o	In degrees
91	Total ϕ desired	In degrees
92	Initial guess at computing interval, DELPHI	In degrees
93	Maximum failures allowed for computing interval selection, MFAIL	Positive integer
94	Maximum error allowable, EMAX	
95	Minimum error allowable, EMIN	
96	Factor to increase computing interval, DTM	
97	Longitude of Greenwich with respect to 1950.0 equinox at initial time, EWOG	In degrees

98	Rotation rate of the earth, EROT	In $\frac{rad}{hr}$
99	Luni-solar flag, LS	Integer; if = 1, consider luni-solar; if = 2, omit
100	Perigee flag, K1OR3	Integer, set = 1 if initial perigee closest to $\frac{\pi}{2}$ or = 2 if closest to $\frac{3\pi}{2}$
101	Multiplier to compute new computing interval, FDT	
102	Derivative flag,	Integer; set = 1 if J_2 and J_4 are the only perturbations; otherwise, set = 2

5.13.2 Output

At the beginning of each case, the entire input array is printed in floating point. There are 25 rows of 5 columns, with locations 1 through 5 printed in the first row, etc.

The next printed values are the initial values of:

Ω (deg.) i (deg.) u (non-dim.) q (non-dim.) velocity (non-dim.)

At the attempted completion of each two computing steps, the following information is printed from the Runge-Kutta routine:

Total ϕ (rad)	Computing Interval (rad)	Intermediate Failure Counter (Integer)	Maximum Allowable Error	Estimated Error	Minimum Allowable Error

At the completion of each four successful computing steps, either Format 1 or Format 2 is printed.

Format 1

p_a	Ω_a	i_a	u_a	q_a	t_a	(All non-dim.)
p_n	Ω_n	i_n	u_n	q_n	t_n	
ϕ(deg)	t(hrs)	r(km)	Ω(deg)	i(deg)	Energy (non-dim.)	
e_a	ω_a(rad)					

Format 2 differs only in that the energy is not printed and the approximate eccentricity is printed in its place, and the approximate argument of perigee then appears in the first column.

Format 1 is printed if the only perturbations are J_2 and J_4. If any other perturbations are considered, Format 2 is used.

After a case has been completed successfully, a start time, stop time, and total time for reading the input data and doing all the computations are printed in minutes.

Error Prints

If the computing interval becomes too small, it is printed along with the comment - COMPUTING INTERVAL SELECTION FAILS, and the case halts.

If the number of computing interval selection failures exceeds the maximum value which is input, the case halts with the comment the same as above.

Section 6

DISCUSSION OF RESULTS

In order to evaluate the effectiveness of the modified Encke approach, comparisons were made between three programs. These programs were the modified Encke program described here, and two existing programs based on a Cowell formulation of the problem and using Runge-Kutta integration. The two Cowell programs were essentially the same except that one performed operations using single-precision arithmetic, and the other used double-precision. The modified Encke program used single-precision arithmetic exclusively.

Three representative orbits were chosen for comparison. These were:

Orbit 1: A low altitude, moderate eccentricity orbit which considered the same perturbations as the analytic model (second and fourth zonal harmonics only). The initial osculating elements were 30° inclination, 0° argument of perigee, .03117 eccentricity, and 6928.2255 kilometers for the semi-major axis. This orbit was chosen so that known integrals of the motion could be used as indications of the accuracies of the three programs.

Orbit 2: A very high altitude, low inclination, nearly circular orbit which considered the second and fourth zonal harmonics of the potential in addition to luni-solar perturbations. The initial osculating elements were 5° inclination, 0° argument of perigee, .0001 eccentricity, and 41,138.154 kilometers for the semi-major axis. This orbit was chosen because orbits of this type are of interest for communications networks for example, and because luni-solar perturbations are significant at these altitudes.

Orbit 3: A highly eccentric, low inclination orbit considering the second and fourth zonal harmonics of the potential in addition to luni-solar perturbations. The initial osculating elements were 5° inclination, 0° argument of perigee, .723 eccentricity, and 23,963.206 kilometers for the semi-major axis. This orbit was chosen because orbits of this type are of interest for environment sampling, and because the oblateness perturbations predominate at perigee while the luni-solar perturbations become significant at apogee.

Figure 1 shows the variation in the polar component of angular momentum (p) for orbit 1 for 20 revolutions of ϕ. This is plotted non-dimensionalized as $\frac{(p-p_i)}{p_i}$. In this case, p should remain constant or Δp should be zero. The modified Encke program satisfies this condition identically, since p is one of the dependent variables, however, the error is shown on the plot as 10^{-9}. It can be seen from Figure 1, that the single precision Cowell solution drifts off monotonically with increasing angle until the error is greater than 10^{-5} after 20 revolutions of the angle ϕ. The double precision Cowell solution oscillates, but the error is never as large as 10^{-7}.

Figure 2 shows the variation in the total energy for orbit 1 for 20 revolutions of ϕ. This is also plotted non-dimensionally as $\frac{\text{Energy-Energy (initial)}}{\text{Energy (initial)}}$. Again this quantity should be constant and zero, but it can be seen that the single precision Cowell solution builds up the error monotonically to approximately $.5 \times 10^{-5}$ after ϕ reaches 7200 degrees. The errors for the double precision Cowell solution and for the modified Encke solution undergo oscillations with the double precision results varying between 10^{-7} and 10^{-9} and the modified Encke results not exceeding $.5 \times 10^{-7}$. This clearly shows that the modified Ecnke approach can improve accuracy while using only single precision arithmetic. A further improvement in accuracy could be achieved by analytic cancellation of all terms of order epsilon when forming the Encke equations of motion. This is theoretically possible and allows the maximum accuracy available with this approach, but it was not deemed feasible within the limits of the present study.

Finally, the positional error was analyzed for all three representative orbits. This was done by taking the double precision $r-\phi$ history as correct and plotting $\frac{\text{r-r (double precision)}}{\text{r (double precision)}}$ vs. a function of ϕ during the 20th

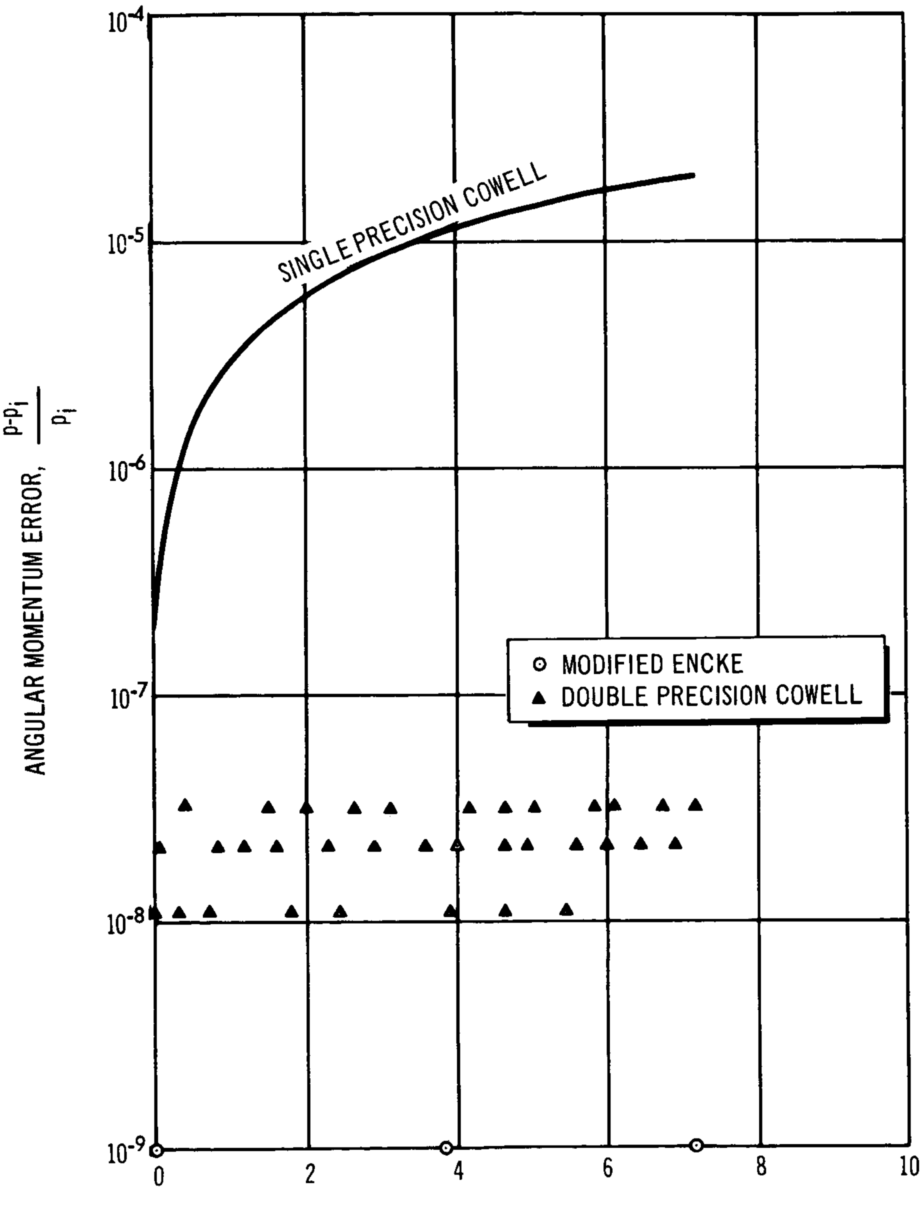


Figure 1. Angular Momentum Error vs. Argument of Latitude
$a \approx 6928$ km., $e \approx .03$, $i \approx 30°$, J_2 and J_4 Perturbations Only

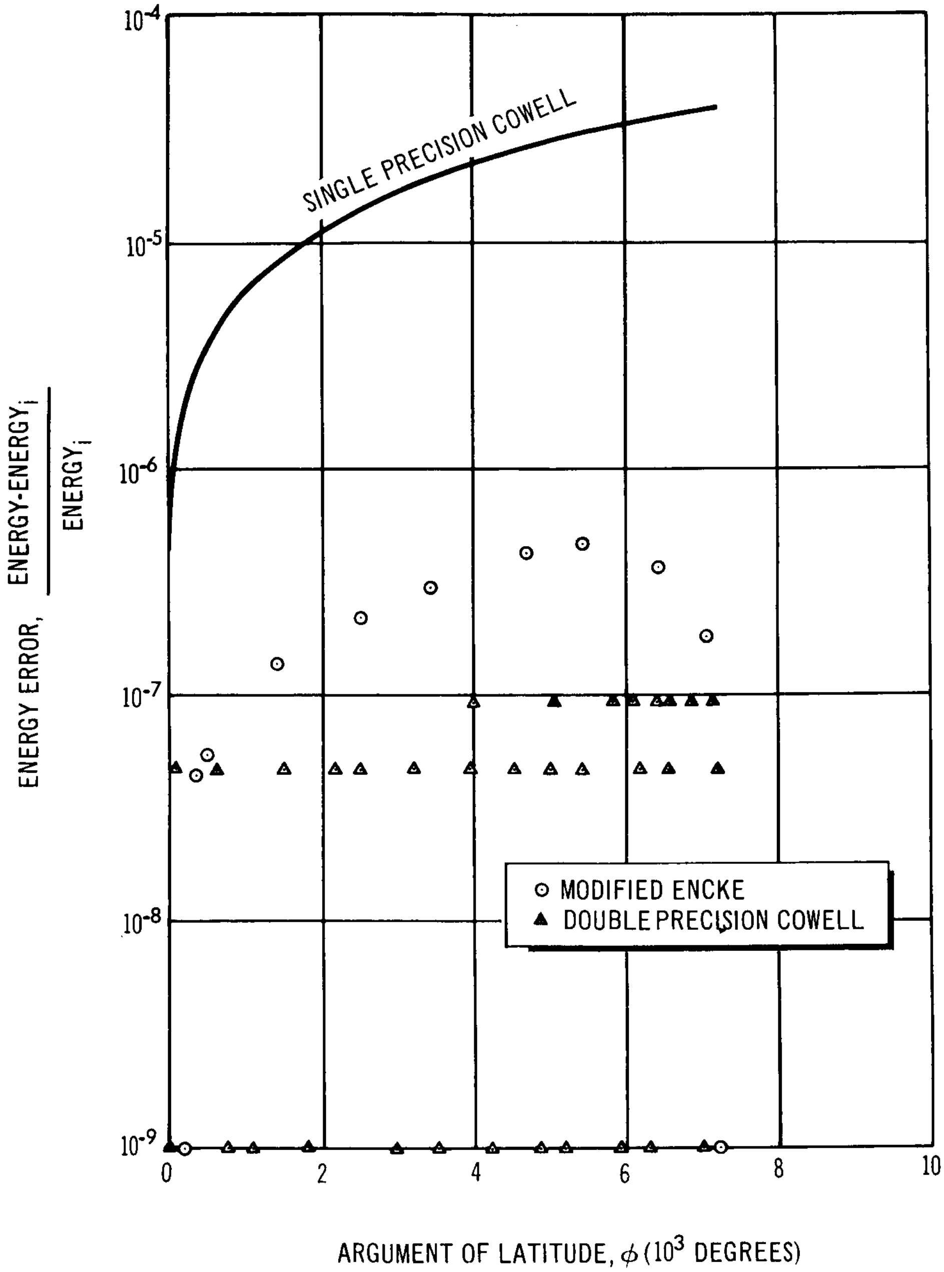


Figure 2. Energy Error vs. Argument of Latitude
a ≈ 6928 km., e ≈ .03, i ≈ 30°. J_2 and J_4 Perturbations Only

revolution of ϕ and comparing the single precision Cowell results and the modified Encke results for all three representative orbits. Figure 3 shows this result for orbit 1, while Figures 4 and 5 represent orbit 2 and orbit 3, respectively.

Figure 3 shows that both the modified Encke and the single precision Cowell solution show reduced errors in the radius near the apogee during the 20th revolution. In general the radius error follows the trend of the error in energy plotted in Figure 2. That is that the single precision Cowell error is nearly 2 orders of magnitude larger.

Figure 4 presents the non-dimensionalized error in radius for the high altitude, nearly circular orbit. The single precision Cowell solution exhibits a smaller error at apogee with a high error near perigee. The error from the modified Encke solution is somewhat erratic, but it remains nearly two orders of magintude below the single precision Cowell solution near perigee. In general the modified Encke solution would have shown a bigger improvement if rectification was included, since oblateness perturbations and luni-solar perturbations are of equal magnitudes at this altitude.

Figure 5 represents the largest error for both the single precision Cowell solution and the modified Encke solution. It can be seen that the modified Encke error is nearly constant and generally below the single precision Cowell error. However, the single precision Cowell error drops very low around the apogee. This can be interpreted to mean that the modified Encke solution should have been rectified before this time, since the luni-solar perturbations are significant and are not included in the analytic model. It also shows that the error in the single precision Cowell solution is due mainly to an error in the time-history of the angle ϕ, and the radius is not sensitive to small time errors in the vicinity of apogee.

In conclusion it can be stated that the modified Encke approach can be used to increase the accuracy of solutions without resorting to double precision arithmetic. In the comparisons made, the more lengthy calculations

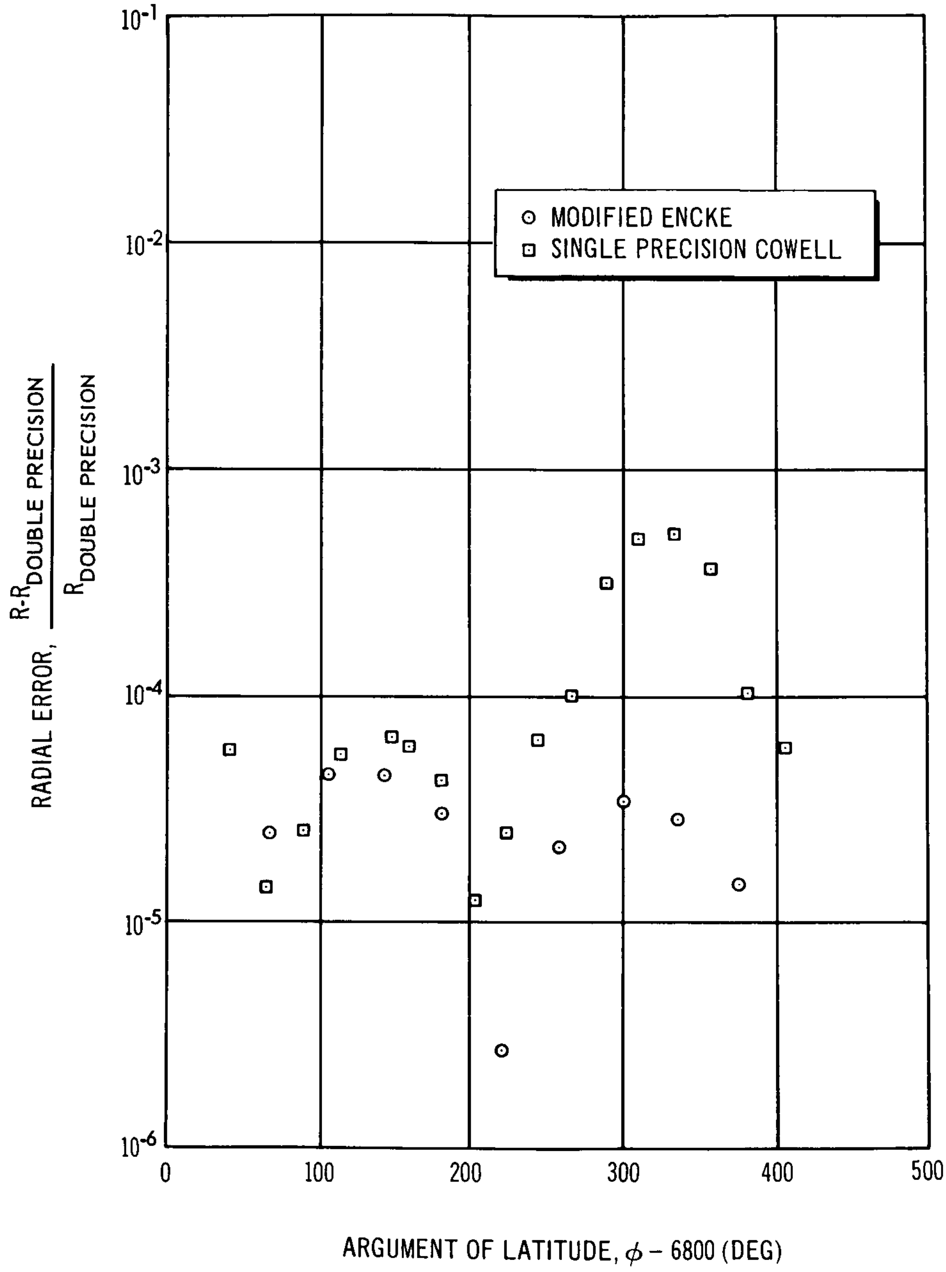


Figure 3. Radial Error vs. Argument of Latitude
$a \approx 6928$ km., $e \approx .03$, $i \approx 30°$ J_2 and J_4 Perturbations Only

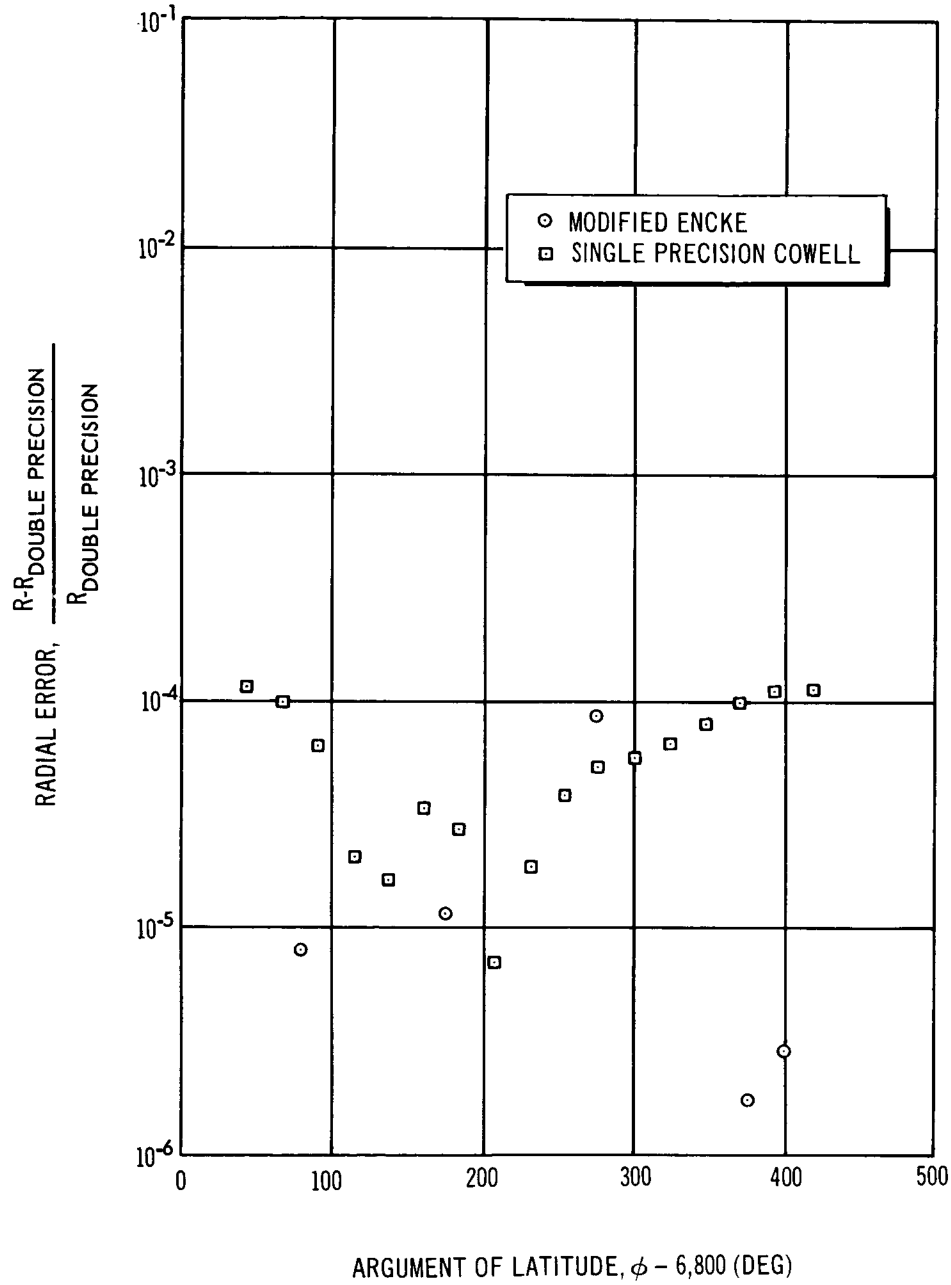


Figure 4. Radial Error vs. Argument of Latitude
$a \approx 41{,}138$ km., $e \approx .0001$, $i \approx 5^\circ$ J_2, J_4, and Luni-Solar Perturbations

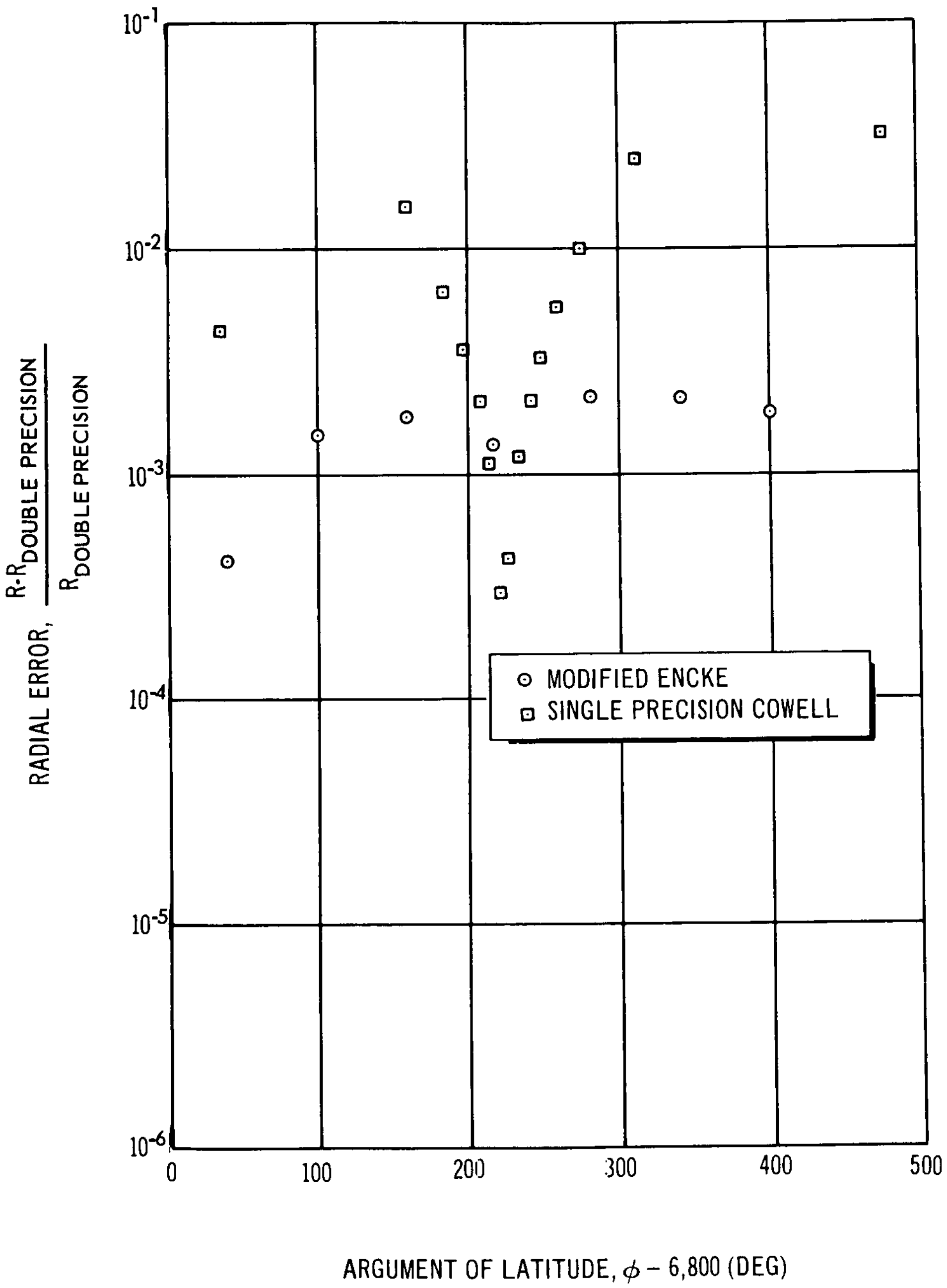


Figure 5. Radial Error vs. Argument of Latitude
$a \approx 23{,}963$ km., $e \approx .723$, $i \approx 5^{\circ}$ J_2, J_4, and Luni-Solar Perturbations

per step were offset by the larger allowable step-size, so that running time was reduced by nearly a factor of four over the double precision Cowell program and was essentially the same as the single precision Cowell program. To achieve the utmost accuracy from such a program for production purposes, the analytic solution should be cancelled analytically to order epsilon when forming the Encke equations, or this portion of the calculation should be done in double precision. Furthermore, for long time predictions a rectification capability would be a necessity.

Section 7

REFERENCES

1. Eckstein, M.; Shi, Y.; and Kevorkian, J.: Satellite Motion for All Inclinations Around an Oblate Planet. Douglas Paper No. 3078, August, 1964.

2. Anon,: ESPOD Mathematical and Subroutine Description. TRW Space Technology Laboratories Report No. 8497-6065-RU000, June, 1964.

3. U.S. Dept. of Commerce, National Bureau of Standards: Handbook of Mathematical Functions. Applied Mathematics Series .55. June, 1964.

4. Dwight, H. B.: Tables of Integrals and Other Related Data. Fourth ed., The Macmillan Co., 1961.

5. Peabody, P. R.; Scott, J. F.; and Oroyco, E. G.: User's Description of JPL Ephemeris Tapes. Jet Propulsion Laboratory, Technical Report No. 32-580, March 2, 1964.

6. Hildebrand, F. B.: Introduction to Numerical Analysis. McGraw-Hill Book Co., Inc., 1956.