

AD-A218 799

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

DTIC
ELECTE
MAR 08 1990
S B D

ORBITAL MECHANICS
A LEARNING TOOL ON THE MAIN FRAME

by

Anthony A. Vraa

September 1989

Thesis Advisor

E.A. Milne

Approved for public release; distribution is unlimited.

00 03 0 05 0

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified		1b Restrictive Markings	
2a Security Classification Authority		3 Distribution Availability of Report Approved for public release; distribution is unlimited.	
2b Declassification Downgrading Schedule		5 Monitoring Organization Report Number(s)	
4 Performing Organization Report Number(s)		7a Name of Monitoring Organization Naval Postgraduate School	
7a Name of Performing Organization Naval Postgraduate School	7b Office Symbol (if applicable) 366	7c Address (city, state, and ZIP code) Monterey, CA 93943-5000	
8a Name of Funding Sponsoring Organization	8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number	
8c Address (city, state, and ZIP code)		10 Source of Funding Numbers	
		Program Element No	Project No
		Task No	Work Unit Accession No
11 Title (include security classification) ORBITAL MECHANICS A LEARNING TOOL ON THE MAIN FRAME			
12 Personal Author(s) Anthony A. Vraa			
13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report (year, month, day) September 1984	15 Page Count 99
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
17 Cross Codes			
Field	Group	Subgroup	18 Subject Terms (continue on reverse if necessary and identify by block number)
			orbital mechanics, unperturbed and perturbed orbit, FORTRAN program, DISSPLA graphics. (EG)
19 Abstract (continue on reverse if necessary and identify by block number)			
This thesis consists of an interactive program that enables the student to study the orbital motion of satellites around the earth. The student can investigate the shape of a variety of orbits by varying the initial position and velocity of the satellite, or by supplying select orbital parameters i.e. initial orbital radius, eccentricity, and inclination. Satellite maneuvers can also be studied, like transfer orbits and inclination changes, by command velocity changes at any location in the orbit. Also the effects of the perturbing forces due to the oblateness of the earth, drag for low earth orbits, and gravitational attraction from the sun and moon can be investigated. The orbits are displayed in either the perifocal coordinate system around a model of the earth, or the ground track can be displayed on a map of the world. Orbital data is displayed below the orbital plot. The display is enabled by the use of display integrated software system and plotting language (DISSPLA) subroutines.			
20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users		21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual E.A. Milne		22b Telephone (include Area code) (408) 646-2886	22c Office Symbol 61Mn

DD FORM 1473, MAR

85 APR edition may be used until exhausted
All other editions are obsolete

Security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Orbital Mechanics
A Learning Tool On The Main Frame

by

Anthony A. Vraa
B.S., University of Minnesota, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(SPACE SYSTEMS OPERATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
September 1989

Author:

Anthony A. Vraa

Anthony A. Vraa

Approved by:

Edmund A. Milne

E.A. Milne, Thesis Advisor

D. A. Danielson

D.A. Danielson, Second Reader

R. Panholzer

Rudolph Panholzer, Chairman,
Department of Space Systems

ABSTRACT

This thesis consists of an interactive program that enables the student to study the orbital motion of satellites around the earth. The student can investigate the shape of a variety of orbits by varying the initial position and velocity of the satellite, or by supplying select orbital parameters i.e. initial orbital radius, eccentricity, and inclination. Satellite maneuvers can also be studied, like transfer orbits and inclination changes, by command velocity changes at any location in the orbit. Also the effects of the perturbing forces due to the oblateness of the earth, drag for low earth orbits, and gravitational attraction from the sun and moon can be investigated. The orbits are displayed in either the perifocal coordinate system around a model of the earth, or the ground track can be displayed on a map of the world. Orbital data is displayed below the orbital plot. The display is enabled by the use of display integrated software system and plotting language (DISSPLA) subroutines.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification _____	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I. INTRODUCTION	1
II. PROGRAM DESIGN	3
III. UNPERTURBED ORBIT	7
IV. PERTURBED ORBIT	10
A. ORBITAL ELEMENTS	10
B. COMPUTE PERTURBING FORCES	11
1. NON-SPHERICAL EARTH	11
2. ATMOSPHERIC DRAG	13
3. PERTURBING FORCE DUE TO HEAVENLY BODY	14
a. SUN'S POSITION	16
b. MOON'S POSITION	16
C. RATE-OF-CHANGE OF ORBITAL ELEMENTS	17
D. NEW ORBITAL ELEMENTS	19
V. VELOCITY CHANGES	20
VI. GRAPHICAL PLOTS	22
A. PERIFOCAL PLOT	23
B. GROUND TRACK	23
C. DATA	23
VII. CONCLUSIONS AND RECOMMENDATIONS	25
APPENDIX A. ORBIT PROGRAM	26
APPENDIX B. COORDINATE SYSTEMS	77
A. 'IJK': GEOCENTRIC - EQUATORIAL	77
B. 'PQW': PERIFOCAL	78

C. "RSW": ORBITAL	79
D. COORDINATE TRANSFORMATIONS	80
APPENDIX C. ORBITAL ELEMENTS	82
APPENDIX D. SAMPLE ORBITS	86
LIST OF REFERENCES	91
INITIAL DISTRIBUTION LIST	92

I. INTRODUCTION

A visual aid for students new to orbital mechanics is required to comprehend fully the dynamics of orbital motion. This program is an interactive time step simulation program that calculates and plots either unperturbed or perturbed elliptical orbits. The program interacts with the student in developing the initial orbit. Also the program enables the student with the ability to change the velocity of the satellite at a specific location in the orbit. This feature will permit the student to investigate the effects of commanded velocity changes as in perigee kicks, apogee kicks and inclination changes. The user can also modify the initial position and velocity of the satellite at the completion of any orbit.

The student is given an opportunity to investigate the effects of perturbing forces on the satellites orbit by choosing to have the program calculate the orbit with or without perturbing forces. The variation of parameters method, as seen in [Ref. 1: pp. 396-407], is used in calculating the perturbing orbit. The perturbing forces taken into consideration are the following:

1. the oblateness of the earth
2. drag for low earth orbits
3. gravitational force of the moon
4. gravitational force of the sun

In order to review fully the operation of the program (included in appendix A) and to uncover any problems or limitations that plagued the programming, the program has been divided up as follows:

1. program design
2. unperturbed orbit
3. perturbed orbit
4. velocity changes
5. graphical plots

The programming approach and equations used in each of the above sections will be examined in there respective chapters. A review of the coordinate systems used and their

transformations between them are included in appendix B. Since all the equations used in the calculation of the orbital elements are from reference 1, they will not be reviewed in each chapter but will be included in appendix C for a quick reference. Equations from other sources will be referenced in their respective chapters.

Examples of perturbed and unperturbed orbital plots for a variety of initial orbital parameters are included in appendix D. Included are plots of low earth orbits, transfer orbits and geosynchronous orbits.

II. PROGRAM DESIGN

In designing this program an attempt was made to make it not only as user friendly as possible, but also to make the program as simple as possible to understand. To achieve these goals, the program would have to be written in a logical manner, in a computer language that is easy to follow, the program would have to run on terminals readily available to students (at the Naval Postgraduate School (NPS)), and the program would have to be easily used by students with a minimum amount of computer or orbital mechanics knowledge.

FORTRAN was chosen as the programming language since it is a widely used scientific language and it allows for very structured programming. By programming in a structured format, the program can be expanded in the future with a minimum amount of time required to understand the programming code. FORTRAN also allows for double precision numbers to be used in the calculation of the orbit. This is critical when round off error in single precision could be greater than the actual change that one is trying to model. The equations in the descriptions of the program might not exactly match the equations in the listings because of special programming techniques which must be included in most computer programs to handle such problems as "division by zero".

The display integrated software system and plotting language (DISSPLA) package available on the mainframe computer at NPS was used to enable a variety of graphical displays with a minimum amount of programming. DISSPLA has a set of subroutines that the programmer calls to display data contained in arrays. This requirement forces the program to load arrays with the satellites position in order for it to be plotted. The TEC618 computer terminal and associative plotter was used for ease of gaining hard copy plots of the orbits and the diversity of locations that are available here at NPS. In order to run a program in DISSPLA the user must first define storage space of 1500k and designate temporary disk space, and then call DISSPLA with the program name. This is accomplished with the following commands:

1. DEFINE STORAGE 1500K

2. I CMS
3. TDISK 4 DIS
4. DISPLA ORBIT

To make the program user friendly, the user is prompted for inputs via the keyboard. The entry is usually a number. A yes or no response can be entered by typing "Y" or a "N". In most cases the program does a check to see if the input is appropriate. In order to make it as easy as possible for the student to get the desired orbit displayed, the program requires only the initial position and velocity of the satellite. The initial position and velocity of the satellite is supplied by the user in one of two ways. The user can input the position and velocity of the satellite, using the perifocal coordinate system (IJK), or the user can let the program place the satellite on the "I" axis of the IJK system at the radius of perigee (RP) distance supplied by the user. This latter choice gives the initial location of the satellite, but to get the velocity the program will prompt the user for one of the following:

1. the actual velocity in the IJK system.
2. the eccentricity (e) of the orbit. In which case the velocity is calculated from the following equations:

$$a = \frac{RP}{1 - e} = \text{semi-major axis}$$

$$ENR = -\frac{\mu}{2a} = \text{energy mass}$$

Where $\mu = MG$

M = mass of earth

G = Universal gravitational constant

$$v = \sqrt{2(ENR + \frac{\mu}{RP})}$$

3. the radius of apogee (RA) The velocity is calculated by first calculating the eccentricity (e) from the following:

$$e = \frac{RA - RP}{RA + RP}$$

With the eccentricity the same equations used above are used to calculate the velocity.

In order to give the velocity a direction the inclination (i) of the orbit is required from the user. The following equations are used to calculate the velocity vector:

$$v_j = 0.0$$

$V_x = 1.23456$

$V_y = 1.23456$

The program will check to ensure that the orbital eccentricity is less than 1.0, if it is not then the program will reject the inputs. After the initial input are accepted, the program will do calculations for the six orbital elements required to describe the size, shape and orientation of the orbit, and to pinpoint the position of the satellite along the orbit at a particular time. This classical set of six orbital elements are as follows:

1. a, semi-major axis.
2. e, eccentricity.
3. i, inclination.
4. Ω , longitude of the ascending node.
5. ω , argument of perigee passage.
6. T, time of perigee passage.

The program actually calculates more orbital elements than the six classical elements required to plot the orbit, this is done in an effort to make the program as robust as possible. This will add in the ability to expand the program in the future.

If the satellite is not initially at the perigee point then the satellite must first be stepped around to the perigee point. The program then enters a loop that calculates the orbit from the perigee point through one complete orbit around the earth and back to the perigee point. The orbit is calculated in steps of 2 times pi divided by an integer, i.e., 2 times pi divided by 50. This step size was used to ensure a smooth orbit for display purposes and also to get within adequate distance to the perigee point or other location for a velocity change. After the loop is completed, the program will offer the user a choice of the following plots to check the orbit:

1. perifocal
2. groundtrack

The program then goes into a loop offering the user the following choices until the user decides to end the program:

1. plot another view of the same orbit.

If the user wishes to plot another view of the same orbit then the user may use this choice to reenter the display portion of the program.

2. plot the next orbit (perturbed or unperturbed).
To plot the next orbit the satellite is stepped around the complete orbit either with or without perturbing forces effecting the satellite.
3. change the initial conditions.
The program goes to the beginning of the program and allows the user to change the initial position and velocity of the satellite.
4. change the velocity at a specific location
Step the satellite around to a specific true anomaly and make a velocity change at that location.
5. clear the previous orbits from the plot.
Clear the memory of all the previous orbits and only retain the current location and velocity as the initial position and velocity.

Before each new orbit, the orbital elements are recalculated.

There are several common assumptions and constants used throughout the program i.e. all bodies are considered to be spherically symmetric (this allows these bodies to be treated as though their masses are concentrated at their centers (point masses)). other assumptions will be covered in their respective chapters.

III. UNPERTURBED ORBIT

The subroutines that calculate the unperturbed orbit are the most widely used subroutines in the entire program. These subroutines are called to step the satellite around to the perigee point from the user supplied initial position and velocity, to calculate the next unperturbed orbit, and for any velocity change. No matter which of these sources supply the initial position and velocity the program calculates the unperturbed orbit in the same manner. The only difference is where in the orbit the satellite is initially when these subroutines are called. Before the unperturbed subroutines are called, the orbital elements are calculated.

The unperturbed subroutines are called by a single subroutine 'UNPERT' which has the following basic algorithm:

1. Increment time by the time step size (DT). The time step was chosen as the period divided by fifty to give a smooth plot, but more importantly to ensure that the satellite is within an acceptable distance from a specific location for a velocity change. The angular error caused by the step size can be as much as $\pm 150^\circ$ from the desired point for a circular orbit and will increase for more eccentric orbits. This error becomes a factor when the user is making velocity changes, and therefore it will be covered in that chapter in further detail.
2. Calculate the new elements. The calculation of the new elements is the heart of this algorithm. The size, shape and orientation of the orbit remains unchanged. What is required is the position of the satellite along the orbit as a function of time. The problem becomes a matter to solve "the Kepler problem"-predicting the future position and velocity of an orbiting object as a function of some known initial position and velocity and the time of flight [Ref. 1: p. 181]. An algorithm using these principles will follow:
 - a. A time step (DT) is added to the time of flight(TF), time of flight is the elapsed time since the satellite passed the perigee point.
$$TF = TF + DT$$
 - b. The new mean anomaly (MA) is calculated from the new time of flight, and the mean motion (MM).
$$MA = MM \times TF$$
 - c. With the new mean anomaly the new eccentric anomaly (EA) is calculated. Because the solution to the Kepler problem ($MA = EA - e \times \sin(EA)$) is transcendental, an iterative solution based on the Newton method of root finding is used. The root in question is a solution to the equation $(MA - EA + e \times \sin(EA) = 0)$. This algorithm takes the form of [Ref. 1: p. 222]:
 - 1) $MA_1 = EA_1 - e \times \sin(EA_1)$

2.

$$E.A._{n+1} = E.A._n - \frac{(M.A. - M.A._n)}{(1 - e \times \cos(E.A._n))}$$

Where this equation is applied initially to $E.A. = M.A$ and then reapplied until the difference between $M.A$ and $M.A$, becomes small enough to be ignored.

d. The new true anomaly (v_n) is calculated from:

$$v_n = \frac{\cos^{-1}(e - \cos(E.A.))}{e \cos(E.A.) - 1}$$

3. Calculate the new position and velocity. The position and velocity are calculated in the perifocal coordinate system (PQW). The PQW system uses the orbit as its fundamental plane and therefore requires only two coordinate to specify the satellite's position and velocity. The z_u coordinate is by definition always equal to zero. The position of the satellite is calculated as:

$$x_u = r \cos v$$

$$y_u = r \sin v$$

$$z_u = 0$$

The velocity of the satellite is calculated as:

$$v_x = \sqrt{\frac{\mu}{p}} (-\sin v_u)$$

$$v_y = \sqrt{\frac{\mu}{p}} (e + \cos v_u)$$

$$v_z = 0$$

4. Store position and elements in arrays for plotting. In order for the program to plot the orbit the radius, true anomaly, inclination, and argument of perigee must be stored in arrays. The use of these arrays to plot the orbit will be explained in chapter 6.

5. The process is repeated until the satellite is at the perigee point and the true anomaly is two pi.

The procedure used to calculate the unperturbed orbit leave very little to be modified by a programmer. The only choices that had to be made concerned step size, how to tell the UNFRET subroutine that the perigee point had been reached, and a value of acceptable error for newtons method. For the unperturbed orbit, the step size just had to be small enough to produce a smooth plot of the orbit. Two indicators for perigee were used, one was that the true anomaly was greater than 6.21 radians (two pi equals 6.28 radians) and the time from the previous perigee point will be greater then the period. The two indicators were logically 'and' together to ensure the perigee point was reached.

The disparity between two π and 6.21 radians is due to the error produced by the satellite not beginning the orbit at exactly the perigee point and the step size used go around the orbit. The acceptable size of error for newtons method was set at 1.0×10^{-10} , because for an unperturbed orbit this would be the major contributor to any error in the orbit and the magnitude of this error would be acceptable. However, in a perturbed orbit there are other factors contributing to determining the acceptable error, and these will be discussed in the next chapter.

IV. PERTURBED ORBIT

The perturbed orbit uses the same basic routines as the unperturbed orbit in stepping the satellite around the earth with one major difference, the perturbing forces produce a time rate of change of the orbital elements that must be applied at each step. The variation of parameters method is used to determine this influence of the perturbing forces on the orbital elements. The analysis is simplified by using the orbital coordinate system 'RSW', as explained in appendix B. The basic algorithm is as follows [Ref. 1: p. 407]:

1. At $t = t_0$ calculate six orbital elements.
2. Compute the perturbing forces and transform it at $t = t_0$ to the 'RSW' SYSTEM.
3. Compute the time rate-of-change of the elements.
4. Calculate the change of elements for one time step, and add the changes to the old values at each step to get the new elements.
5. From the new values of the orbital elements, calculate a position and velocity.
6. Go to the step 2 and repeat until the final time is reached.

The steps in the algorithm will be explained in the following sections:

A. ORBITAL ELEMENTS

The standard orbital elements a , e , i , Ω , ω and T (or M) will be used, where

a = semi-major axis

e = eccentricity

i = inclination

Ω = longitude of ascending node

ω = argument of perigee

T = time of perigee passage

(M_0 = mean anomaly at epoch = $M - n(t - t_0)$). The elements are calculated only at the beginning of the orbit from the initial position and velocity vectors. The elements are then changed continuously throughout the orbit by adding the changes due to the perturbing forces. For the perturbed orbit, the satellite will always begin at the perigee point. This is done so one complete orbit is from perigee point to perigee point.

B. COMPUTE PERTURBING FORCES

The variation of parameters method requires that the perturbing forces be calculated at each step in the orbit. In order to do this a model of each perturbing force must be developed. The following perturbing forces were used in calculating the total perturbing force effecting the satellite:

1. oblateness of the earth
2. atmospheric drag
3. gravitational attraction of the sun
4. gravitational attraction of the moon

The magnitudes of these forces have an enormous range of values and are dependent on the distance the satellite is from the perturbing body. Figure 1 on page 12 shows a graphical representation of the magnitude of the perturbing forces in a log-log plot of perturbing forces per unit mass [Ref. 2: p. IV-61]. The model of each of these forces follows:

1. NON-SPHERICAL EARTH

The earth is not perfectly spherical, but bulges around the equator. The polar and equatorial diameters are 12713.0 Km and 12756.3 Km, respectively. The oblateness results in a perturbing force per unit mass with these components in the 'RSW' coordinate system [Ref. 3: p. 51]:

$$F_r = \frac{(-3\mu J_2 r_e^2)}{2r^3} (1 - 3 \sin^2(i) \sin^2(u_0))$$

$$F_s = \frac{(-3\mu J_2 r_e^2)}{r^3} (\sin^2(i) \sin(u_0) \cos(u_0))$$

$$F_w = \frac{(-3\mu J_2 r_e^2)}{r^3} (\sin(i) \cos(i) \sin(u_0))$$

The variable and constants of these equations are defined below:

i. Variables:

- a. u_0 = the argument of latitude and is equal to the true anomaly v_0 plus the argument of perigee ω .

$$u_0 = v_0 + \omega$$

- b. r = the radius from the center of the earth to the satellite.

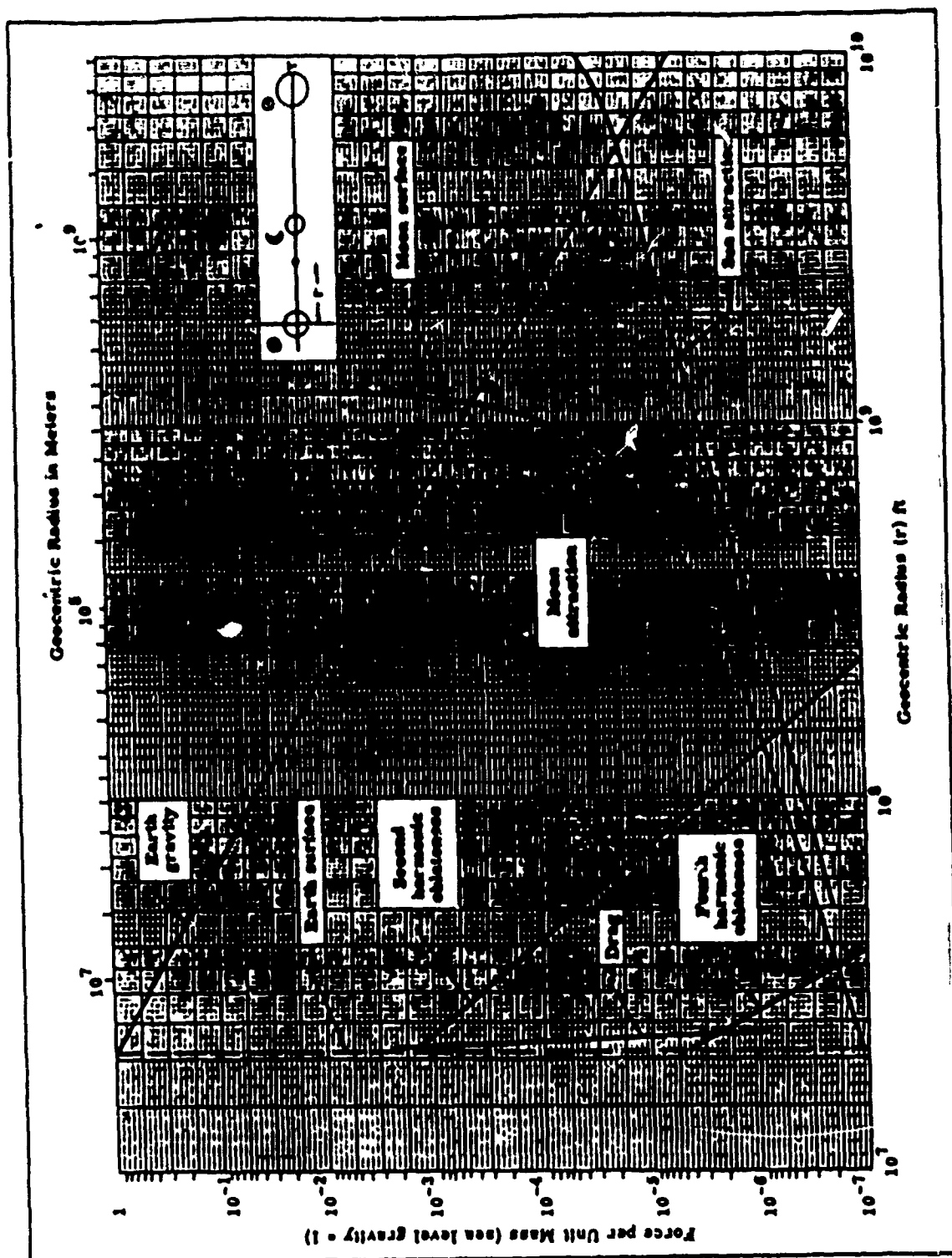


Figure 1. Comparison of perturbation magnitudes.

$$\vec{r} = \vec{r}$$

2. Constants:

- a. μ = the gravitational parameter of the earth.

$$\mu = 398601.2 \frac{km^3}{s^2}$$

- b. J_2 = the second harmonic of oblateness coefficient, determined by experimental observations.

$$J_2 = 1.0823E-3$$

- c. r_e = the mean radius of the earth,

$$r_e = 6.3782E3 Km$$

2. ATMOSPHERIC DRAG

The formulation of atmospheric drag equations are plagued with uncertainties of atmospheric fluctuations, frontal areas of orbiting object (if not constant), the drag coefficient, and other parameters. A fairly simple formulation will be given here. Drag, by definition, will be opposite to the velocity of the vehicle relative to the atmosphere. Thus, the perturbing force is

$$\vec{F} = -\left(\frac{1}{2m}\right) \cdot CD \cdot AR \cdot DEN \cdot v \cdot \vec{v}$$

The velocity vector is in the 'IJK' system so the resulting force is also in the 'IJK' system. Therefore a transformation to the 'RSW' system is required.

The variables and constants of this equation are defined below:

1. Variables:

- a. v = speed of vehicle.

- b. CD = the dimensionless drag coefficient. The drag coefficient CD has a value between 1 and 2. It takes a value near 1 when the mean free path of the atmospheric molecules is small compared with the satellite size, and takes a value close to 2 when the mean free path is large compared with the size of the satellite. The drag coefficient will be modeled with $CD = 2$ when the satellites altitude is greater than 550km and equal to 1 otherwise. [Ref. 4: p. 295]

- c. DEN = atmospheric density at the vehicle's altitude. The density is spherically symmetric, and will be modeled using exponential steps using the parameters in Table 1 on page 14 and the following formula [Ref. 1: pp. 423-424]:

$$\delta(z) = \delta_0 e^{a_1 z}$$

Table 1. ATMOSPHERIC PARAMETERS AND VALUES

Altitude (km)	ρ	μ	γ	δ
0.150	1.225E-02	4.74E-02	0.0	1.2225E-02
			150	1.0E-03
150.550	1.79846E-01	4.3614E-02	550	3.0E-8
550	1.015484E-07	2.21698E-07	1500	3.65E-09
			2100	1.0E-12

2. Constants set to typical values:

- m = mass of the satellite, set equal to 100kg.
- AR = the sectional area of the vehicle perpendicular to the direction of motion.

3. PERTURBATION DUE TO HEAVENLY BODY

The satellite is subject to perturbation forces due to the gravitational effects of the sun and the moon. The perturbation force from a perturbing body is the difference between the gravitational force due to the perturbing body at the satellite and the gravitational force the satellite would experience if it were at the center of the earth. From Figure 2 on page 15, the perturbing force per unit mass of the satellite is

$$\vec{f}_p = \mu_p \frac{\vec{r}_p \vec{r}_p - r_p^2 \vec{r}}{(\vec{r}_p - \vec{r})^3} - \frac{\mu_p \vec{r}_p}{r_p^3}$$

The variable and constants are defined below:

1. Variables:

- r_p = distance from the earth center for the perturbing body
- $\vec{r}_p$ = unit vector from the earth to the perturbing body
- r = distance from earth center to the satellite
- $\vec{r}$ = unit vector from the earth to the satellite

2. Constants:

- μ_p = gravitational constant of the perturbing body = $M_p G$

The subscript p is to be replaced by s if the perturbing body is the sun, and by m if the perturbing body is the moon. We will assume that $r \ll r_p$, then the equation above becomes

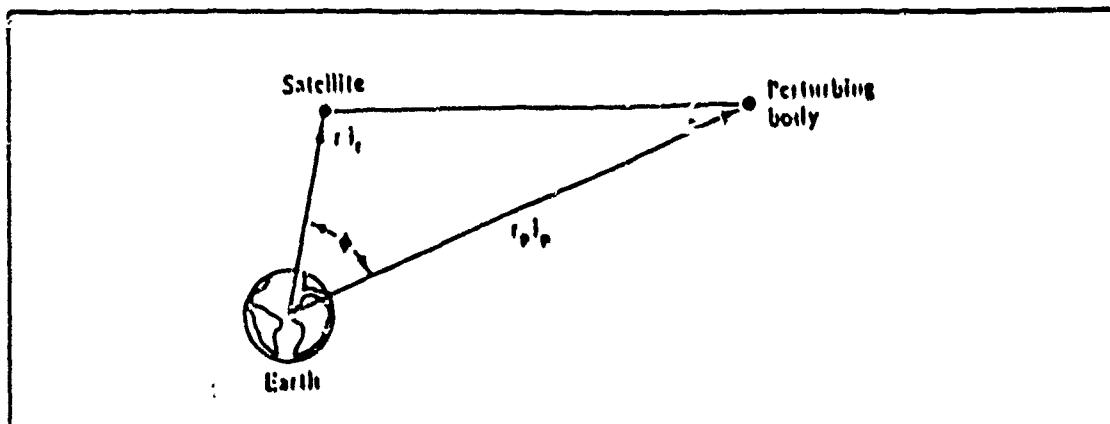


Figure 2. Perturbation forces.

$$\vec{F}_p = \left(\frac{\mu_p}{r_p^2} \right) \left(\frac{r}{r_p} \right) (3(\vec{i}_r \vec{i}_p) \vec{i}_p - \vec{i}_r)$$

The unit vectors $\vec{i}_r$ and $\vec{i}_p$ can be written in terms of the 'IJK' system as:

$$\vec{i}_r = (\cos(\Omega) \cos(u_0) - \sin(\Omega) \cos(i) \sin(u_0)) \vec{I} + (\cos(u_0) \sin(\Omega) + \cos(\omega) \cos(i) \sin(u_0)) \vec{J} + (\sin(i) \sin(u_0)) \vec{K}$$

$$\vec{i}_p = (\cos(\Omega_p) \cos(u_{0p}) - \sin(\Omega_p) \cos(i_p) \sin(u_{0p})) \vec{I} + (\cos(u_{0p}) \sin(\Omega_p) + \cos(\omega_p) \cos(i_p) \sin(u_{0p})) \vec{J} + (\sin(i_p) \sin(u_{0p})) \vec{K}$$

where Ω , i , and u_0 are the orbital elements of the satellites and Ω_p , i_p , and u_{0p} are the orbital elements of the perturbing body. The formulas above use the 'IJK' system, and as such the resultant forces must be transformed to the 'RSW' system. Models of the sun and moon orbits are required to calculate $\vec{r}_s$ and $\vec{i}_p$. The models used in the program for the sun and moon's orbits follows: [Ref. 3: pp. 73-74]

a. SUN'S POSITION

In order to model the sun's orbit, a number of simplifications had to be made in the actual parameters of the sun's orbit. First the sun will be assumed to be in a circular orbit. This means that the radius (r) to the sun will be constant, and the eccentricity (e) will equal 0.0 instead of its true value of 0.017. The other assumption will

be to place the sun on the 'I' axis of the 'IJK' system at the beginning of the program and have it progress through its orbit as the program runs. These changes will not effect the perturbing force in any noticeable magnitude.

The following variables and constants were used in the program to model the sun's orbit after applying the simplifications: [Ref. 3: pp. 75-78]

1. Constants:

a. Gravitational Constant: $G = 6.67E-11 \frac{(Nm^2)}{kg^2}$

b. Sun's Mass: $m_s = 1.99E30 kg$

c. Sun's Gravitational parameter:

$$\mu_s = 1.32733E20 \frac{Nm^2}{kg}$$

d. Sun's eccentricity: $e_s = 0.0$

e. Radius of orbit, assume sun is in circular orbit: $r_s = 1.49E11 m$

f. Sun's inclination: $si = 23.45 \text{ deg.} = 4.09279709d-01 \text{ radians}$

g. Longitude of ascending node: $\Omega_s = 0.0$

h. Argument of perigee: $\omega_s = 0.0$

2. Variables:

a. The true anomaly of the sun's position as a function of the time the satellite has been in orbit:

$$v_s(TT) = \frac{2\pi}{350 \times 24 \times 3600} TT$$

Where TT = true time, the time the satellite has been in orbit (sec)

b. Sun's Position vector: $\vec{r} = r \cos v_s \vec{P} + r \sin v_s \vec{Q}$

c. Unit vector from the earth to the sun: $\vec{i}_s = \frac{\vec{r}_s}{|\vec{r}_s|}$

b. MOON'S POSITION

In modeling the orbit of the moon, similar assumptions were used as with the sun. The moon's orbit will be assumed to be circular, actually the eccentricity is equal to 0.055. By placing the moon initially on the 'I' axis of the 'IJK' system along with the sun, the gravitational forces of the two bodies will combine to a maximum. However; since the moon's orbital period is only 27.3 days, the moon will not stay in this alignment and the magnitude of the combined forces will vary with time. The inclination of the moon's orbit is not constant, but drifts between 18.3 and 28.6 degrees in ten years.

Also the longitude of the ascending node (Ω) oscillates between 13 and -13 degrees. To simplify this the inclination will be chosen as a constant 23.5 degrees and the longitude of the ascending node as 0.0 degrees. For the time period involved in calculating the perturbed orbit, these assumptions will not make any significant difference.

The following variables and constants were used in the program to model the moons orbit, after applying the simplifications:

1. Constants:

- a. Gravitational Constant: $G = 6.67E-11 \frac{(Nm^2)}{kg^2}$
- b. Moon's Mass: $m_m = 7.35E22 kg$
- c. Moon's Gravitational Parameter: $\mu_m = GM_m = 4.90E12 \frac{(Nm^2)}{kg}$
- d. Moon's eccentricity: $e_m = 0.0$
- e. Radius of orbit, assume moon is in circular orbit: $r_m = 3.844E8 km$
- f. Moon's inclination: $i = 23.5 \text{ deg.} = 4.10152374E-1 \text{ radians}$
- g. Moon's longitude of ascending node: $\Omega_m = 0.0$
- h. Moon's argument of perigee: $\omega_m = 0.0$
- i. Moon's period: $T = 27.3 \text{ days [period]}$

2. Variables:

- a. The true anomaly of the moon's position as a function of the time the satellite has been in orbit: $v_m(TT) = \frac{2\pi}{27.3 \times 24 \times 3600} TT$
- b. Moon's position Vector: $\vec{r} = r \cos v_m \vec{P} + r \sin v_m \vec{Q}$
- c. Unit vector from earth to moon: $\vec{i}_m = \frac{\vec{r}_m}{|\vec{r}_m|}$

The models of the sun and moons orbit calculates the position vector in the 'PQW' system and therefore the position vector must be transformed to the 'IJK' system.

C. RATE-OF-CHANGE OF ORBITAL ELEMENTS

The derivations and equations of the rates-of-change of the orbital elements are contained in reference 1 pages 398 to 406. Therefore; only a summary of the actual analytic expressions for the rate-of-change of the parameters in terms of the perturbations will follow:

1. Rate-of-change of the semi-major axis:

$$\frac{dr}{dt} = \left[\frac{2e \sin v}{n' \sqrt{1-e^2}} \right] F_r = \left[\frac{2e \sqrt{1-e^2}}{n' r} \right] F_r$$

Where n' is the mean motion of the satellites orbit.

$$n' = \sqrt{\frac{\mu}{a^3}}$$

2. Rate-of-change of the eccentricity:

$$\frac{de}{dt} = \left[\frac{\sqrt{1-e^2} \sin v_0}{n' a} \right] F_r + \left[\frac{\sqrt{1-e^2}}{n' a^2 e} \right] \left[\frac{a^2(1-e^2)}{r} - r \right] F_r$$

3. Rate-of-change of the inclination:

$$\frac{di}{dt} = \left[\frac{r \cos u_3}{n' a^2 \sqrt{1-e^2}} \right] F_w$$

4. Rate-of-change of the longitude of the ascending node:

$$\frac{d\Omega}{dt} = \left[\frac{r \sin u_3}{n' a^2 \sqrt{1-e^2} \sin i} \right] F_w$$

5. Rate-of-change of the argument of perigee:

$$\frac{d\omega}{dt} = \left(\frac{d\omega}{dt} \right)_r + \left(\frac{d\omega}{dt} \right)_t + \left(\frac{d\omega}{dt} \right)_w$$

Where,

$$\left(\frac{d\omega}{dt} \right)_r = \left[\frac{-\sqrt{1-e^2} \cos v_1}{n' a e} \right] F_r$$

$$\left(\frac{d\omega}{dt} \right)_t = \left[\frac{p}{eh} \right] \left[\sin v_0 \left(1 + \frac{1}{1+e \cos v_0} \right) \right] F_r$$

$$\left(\frac{d\omega}{dt} \right)_w = \left[\frac{-r \cot i \sin u_3}{n' a^2 \sqrt{1-e^2}} \right] F_w$$

6. Rate-of-change of the eccentric anomaly:

$$\frac{dE.A}{dt} = \frac{1}{\sin(E.A)} \frac{[(\sin v_0 + \frac{de}{dt})(1+e \cos v_0) - (\cos v_0 + e)(\frac{de}{dt} \cos v_0 + e \sin v_0)]}{[1+e \cos v_0]^2}$$

7. Rate-of-change of the mean anomaly:

$$\frac{dM.A}{dt} = \frac{dE.A}{dt} - \frac{de}{dt} \sin(E.A) - e \times \cos \frac{(E.A)dE.A}{dt} - \frac{dn'}{dt} (t - t_0)$$

This equation reduces to the following for circular and ecliptic orbits
 $(e = 0, i = 0)$

$$\frac{dM}{dt} = \frac{-1}{a^3} \left[\frac{2r}{a} - \frac{1(1-e^2)}{e} \cos v \right] F_r - \left[\frac{1-e^2}{a^3 a e} \right] \left[1 - \frac{r}{a(1-e^2)} \right] \sin v F_t = r \frac{dn'}{dt}$$

Where the Rate-of-change of the mean motion:

$$\frac{dn'}{dt} = \left[\frac{-3\mu}{2a'^3} \right] \frac{da}{dt}$$

[ref. 1 p. 396-407]

D. NEW ORBITAL ELEMENTS

The change of each element is calculated by multiplying the rate-of-change of the element by the time step (DT). The change in the orbital elements are then added to the current values of the elements to give the new orbital elements. With the new elements calculated, the satellite is stepped forward and the new position and velocity are calculated in the same manner as the unperturbed orbit (chapter 3). Also as with the unperturbed orbit, the process is repeated until the satellite is at the perigee point, indicated by the time of flight (TF) equal to the period of the perturbed orbit.

V. VELOCITY CHANGES

The ability of the student to change the velocity of the satellite at any position in the orbit is a vital element in this program. With velocity changes the student can investigate the effects of varying the satellites velocity as in transfer orbits and inclination changes. In order to simplify the program the unperturbed orbit is used throughout this routine. The velocity change algorithm used in the program follows:

1. Rotate to velocity change location.

The user is given the choice of changing the velocity of the satellite at the perigee, apogee or at any true anomaly. If the user chooses perigee or apogee as the change locations, the true anomaly is set equal to zero or π radians respectively. With the location of the velocity change, the satellite is first stepped around to the desired true anomaly. The stepping is identical with the unperturbed orbit with the exception that the stepping terminates when the true anomaly is greater or equal to the desired true anomaly. With a step size of one fiftieth of the period, the satellite is actually stepped around to a location near the desired location. This variance can be reduced by decreasing the step size but this would increase the computation time. This error will be a major factor in precise calculations of transfer orbits, or any other orbital maneuver where precise velocity changes are required. However, this program is not a tool to calculate precise orbital maneuvers, but rather a learning tool for the student to get a feel for the results of velocity changes in a satellite's orbit.

2. Change the velocity.

With the satellite at the desired location, the program calculates and displays for the user the satellite's current velocity, escape velocity and circular velocity (the velocity required to circularize the orbit). The program will not allow velocities greater than or equal to the escape velocity. The user is given the option to enter a new velocity in the 'LJK' system or to change the magnitude of the velocity in the orbital plane. If the user chooses to change the velocity in the orbital plane, the program will prompt the user for the magnitude of the velocity change, and multiply this change by a unit vector in the direction of the satellites velocity. This velocity change vector is then added to the satellites velocity vector, to calculate the new velocity vector.

3. Calculate new elements.

The orbital elements are calculated with the new velocity vector and the satellite's position vector.

4. Complete the orbit.

The program will complete the orbit to the new perigee point using the satellite's position, new velocity and new elements. There are a number of problems that arise if the satellite is just stepped around to the perigee point. For example, with velocity changes in the orbital plane the apogee and perigee directions can physically swap. This is a problem when plotting with the perifocal coordinate system because the X_p axis points toward perigee. To avoid problems like this the arrays used in plotting the orbit must be cleared and the satellite's current position

and velocity be treated as initial conditions. However, to compare the old and new orbits there is a desire to retain as much of the previous orbit as possible. The velocity changes were divided into the following four cases to handle these problems:

- a. Change velocity in the orbital plane at the perigee point with the new velocity greater than the circular velocity. The perigee point will remain the same so the satellite is stepped around using the unperturbed subroutines.
- b. Change velocity in the orbital plane at the perigee point with the new velocity less than or equal to the circular velocity. The perigee and apogee directions will switch so the plotting arrays are first cleared and stored with the current location data. Because the satellite is now at the apogee point the satellite is stepped around to the perigee point storing the second half of the orbit. The entire next orbit is calculated and stored to get a complete orbit.
- c. Change velocity in the orbital plane at the apogee point with the new velocity less than the circular velocity. The perigee and apogee directions will remain the same, so the satellite is stepped around to the perigee point completing the orbit.
- d. This last case catches all the following velocity changes: velocity change in the orbital plane at the apogee point with the new velocity greater than or equal to the circular velocity, velocity changes at any other true anomaly in the orbital plane, and any velocity change out of the orbital plane. The plotting arrays are cleared and stored with the current location data. No matter where in the orbit the satellite is, the satellite is first stepped around to the perigee point, and to ensure a complete orbit is plotted the entire next orbit is also calculated and stored.

VI. GRAPHICAL PLOTS

The program provides two types of graphical displays of the orbit, a display in the perifocal coordinate system and a display of the satellite's ground track. Each display type is useful in observing different aspects of the orbit. The perifocal display will allow the user to see how certain orbital parameters change with different initial positions and velocities, and also how the parameters change with velocity changes at varying positions in the orbit. The ground track will enable the user to gain an appreciation for the physical location of the satellite above the earth, and see how the orbital parameter affects the path of the satellite. The ground track will also display the precession of a sequence of orbits. Both displays plot the position steps to give the user an understanding of how the satellite speeds up at perigee and slows down around apogee.

The DISSPLA package on the mainframe computer was used to enable the plotting of the orbits. The versatility of plotting subroutines of DISSPLA makes the actual programming of the orbit a simple matter of initializing DISSPLA for the type of monitor being used, setting up the plotting area, initializing the axis and axis scale, and then plotting the desired curve from points contained in arrays. This is a simplified explanation of DISSPLA, but for further details on DISSPLA programming refer to the DISSPLA user's manual [Ref. 5]. DISSPLA also supplies subroutines to draw a variety of projections of the world and fill the projections with coast lines, latitude lines and longitude lines. There are a couple of DISSPLA requirements that did require special handling in the program. The requirement that the data be supplied in arrays forced the program to load arrays with the required position and parameters and to keep a counter for the number in the arrays. The array format requires the size of the array be specified in the beginning of the program. The array size needs to be large enough to hold a number of orbits, but not so large as to waste storage space. The program will continue to add orbital data to the arrays until the user chooses to delete the previous orbits. If a new initial position and velocity is entered or if the arrays will overflow with the next orbit the arrays will automatically delete all previous orbits. DISSPLA also requires that all data be in single precision format. The program calculates all orbits in double precision in order to limit the effect of round-off error, but by using the single precision data for plotting will not affect the accuracy of the plot in any way.

The subroutines used to display the orbits will be covered in the following three sections:

A. PERIFOCAL PLOT

The plotting of the orbit in the perifocal coordinate system is the easier of the two types of plots. Since the perifocal coordinate system has the orbital plane as the fundamental plane, the only requirements to describe the orbit in the perifocal coordinate system are arrays with the true anomaly and the radius to the satellite. To give the user a sense of the size of the plot, the axis length varies with the eccentricity and semi-major axis length. Also a plot of the earth is plotted to the same scale, with the pole or center of the plot on the origin of the axis. The latitude of the earth at the center of the plot will vary with the inclination of the orbit. This plot will allow the user to see a relative view of the satellite's coverage in the minus 'Z' axis direction of the perifocal coordinate system.

B. GROUND TRACK

The ground track plot is a very complex subroutine compared with the perifocal plot. Because the ground track is not a continuous curve a procedure to handle the satellite ending at one end of the plot and wrapping around to the other end was developed. The wrap around problem is avoided in most orbits by plotting the orbit in segments with the following two rules. Each segment begins at the beginning of a new plot or at the edge of the plot area, and ending when the satellite would wrap around to the other side of the plot. At the beginning of a segment if the position of the satellite is within five degrees of the edge of the plot, that position and any other positions within that five degree boundary will not be plotted. The segment will end when the satellite is within ten degrees of the edge of the plot. The above restrictions imposed on the segments of the plot will not substantially affect the interpretation or usefulness of the plot. The ground track is plotted on top of a cylindrical equidistant projection of the world, with the world coast lines and a longitude-latitude grid for reference.

C. DATA

Information concerning the orbit is displayed on the lower half of the plot. The information is designed to supply the user with enough of the basic orbital elements and other parameters affecting the orbit to be able to evaluate what basic type of orbit the satellite is in, and the effects of velocity changes and perturbing forces have on the orbit. The following data are plotted: inclination(i), semi-major axis (a), eccentricity (e), period

(per), apogee and perigee velocity and radius, average time rate-of-change of orbital elements, and the average magnitude of perturbing forces per unit mass.

VII. CONCLUSIONS AND RECOMMENDATIONS

The program supplies the student with an interactive tool to study the orbital motion of satellites around the earth. The student can investigate a variety of orbits by varying the orbital parameters, command velocity changes, and observe the effects of perturbing forces.

The student is provided with two options for entering the initial position and velocity of the satellite. The program could be expanded to provide the student with the additional options of entering either orbital parameters or a ground observation data and have the program calculate the initial position and velocity from this data. Also the student is limited to orbits with eccentricities less than one (elliptic orbits). The program could be also be expanded to include more eccentric orbit for Lunar, interplanetary, and missile trajectories. The perturbing orbit is calculated for orbits around the earth with relatively small perturbing forces in relation to the earths gravitational force. This fact will cause the program to produce false results if the student tries to calculate lunar trajectories. Special routines would have to be employed when the perturbing force (the moons gravitational attraction) is comparable to the earths gravitational attraction. This will not become a factor for studying current satellite orbits out to the geosynchronous radius of 42241.1km.

The velocity change subroutines move the satellite to a location close to the desired location before a velocity change is imposed. By reducing the step size in the velocity change subroutine, this error could be reduced. Precise orbital transfer maneuvers can be modeled by reducing this error caused by the positioning of the satellite prior to changing the velocity. The program will currently provide the student with useful plots for gaining experience with various transfer orbits by varying the magnitude and location of the velocity changes.

The output of the calculations of the orbit are arrays loaded with the satellite's position and select orbital parameters. The `DISPLA` subroutines that plot the points are not unique. The program would become portable to personal computers with these graphics subroutines written in `FORTRAN` and included in the program.

A final recommendation is that the display of the ground track could be modified to show ground coverage, number of satellites in a constellation, and other elements necessary for planning a real-world artificial satellite application.

APPENDIX A. ORBIT PROGRAM

*	PROGRAM ORBIT	ORB00010
*	THIS PROGRAM IS AN INTERACTIVE TIME STEP SIMULATION OF	ORB00020
*	SATELLITES AROUND THE EARTH. PERTURBED AND UNPERTURBED ORBITS	ORB00030
*	ARE CALCULATED AND PLOTTED. VELOCITY CHANGES ARE ALSO PERMITTED	ORB00040
*	AT SPECIFIED TRUE ANOMALIES.	ORB00050
*	A LIST OF VARIABLES USED BY THE MAIN PROGRAM FOLLOWS.	ORB00060
*	A = SEMI-MAJOR AXIS	ORB00070
*	AL = ARGUMENT OF LONGITUDE	ORB00080
*	AP = ARGUMENT OF PERIGEE	ORB00090
*	CHTA = VELOCITY CHANGE LOCATION TRUE ANOMALY	ORB00100
*	DT = TIME STEP	ORB00110
*	E = ECCENTRICITY	ORB00120
*	EA = ECCENTRIC ANOMALY	ORB00130
*	EI = I VECTOR OF ECCENTRICITY	ORB00140
*	EJ = J VECTOR OF ECCENTRICITY	ORB00150
*	EK = K VECTOR OF ECCENTRICITY	ORB00160
*	FR = R VECTOR OF TOTAL FORCE	ORB00170
*	FS = S VECTOR OF TOTAL FORCE	ORB00180
*	FW = W VECTOR OF TOTAL FORCE	ORB00190
*	H = ANGULAR MOMENTUM	ORB00200
*	HI = I VECTOR OF ANGULAR MOMENTUM	ORB00210
*	HJ = J VECTOR OF ANGULAR MOMENTUM	ORB00220
*	HK = K VECTOR OF ANGULAR MOMENTUM	ORB00230
*	I = INCLINATION	ORB00240
*	IOPT1 = PERTURBED OR UNPERTURBED OPTION	ORB00250
*	IOPT2 = OPTIONS: PLOT NEXT ORBIT, CHANGE INITIAL VALUES,	ORB00260
*	CHANGE VELOCITY, PLOT ANOTHER VIEW OF ORBIT, QUIT	ORB00270
*	LAN = LONGITUDE OF ASCENDING NODE	ORB00280
*	LP = LONGITUDE OF PERIGEE	ORB00290
*	MA = MEAN ANOMALY	ORB00300
*	MM = MEAN MOTION	ORB00310
*	MU = GRAVITATIONAL PARAMETER	ORB00320
*	N = ASCENDING NODE	ORB00330
*	NI = I VECTOR OF ASCENDING NODE	ORB00340
*	NJ = J VECTOR OF ASCENDING NODE	ORB00350
*	NK = K VECTOR OF ASCENDING NODE	ORB00360
*	NUM = STEP COUNTER	ORB00370
*	P = SEMI-LATUS RECTUM	ORB00380
*	PER = PERIOD OF ORBIT	ORB00390
*	PI = PI	ORB00400
*	RA = RADIUS OF APOGEE	ORB00410
*	RE = RADIUS OF EARTH	ORB00420
*	R = ORBITAL RADIUS	ORB00430
*	RI = I VECTOR OF ORBITAL RADIUS	ORB00440
*	RJ = J VECTOR OF ORBITAL RADIUS	ORB00450
*	RK = K VECTOR OF ORBITAL RADIUS	ORB00460
*	T = TIME COUNTER IN ORBIT	ORB00470
*	TA = TRUE ANOMALY	ORB00480
*	TDA = TOTAL CHANGE IN SEMI-MAJOR AXIS	ORB00490
*	TDAP = TOTAL CHANGE IN ARGUMENT OF PERIGEE	ORB00500
		ORB00510

★	TDE = TOTAL CHANGE IN ECCENTRICITY	ORB00520
★	TDM = TOTAL CHANGE IN ANGULAR MOMENTUM	ORB00530
★	TDI = TOTAL CHANGE IN INCLINATION	ORB00540
★	TDMA = TOTAL CHANGE IN MEAN ANOMALY	ORB00550
★	TDMM = TOTAL CHANGE IN MEAN MOTION	ORB00560
★	TDLAN = TOTAL CHANGE IN LONGITUDE OF ASCENDING NODE	ORB00570
★	TF = TIME OF FLIGHT	ORB00580
★	TFDRA = TOTAL FORCE OF DRAG	ORB00590
★	TFEA = TOTAL FORCE OF EARTH'S OBLATENESS	ORB00600
★	TFMO = TOTAL FORCE FROM MOON	ORB00610
★	TFSU = TOTAL FORCE FROM SUN	ORB00620
★	TL = TRUE Longitude AT EPOCH	ORB00630
★	TT = TRUE TIME SINCE SATELLITE HAS BEEN IN ORBIT	ORB00640
★	V = SATELLITE VELOCITY	ORB00650
★	VI = I VECTOR OF SATELLITE VELOCITY	ORB00660
★	VJ = J VECTOR OF SATELLITE VELOCITY	ORB00670
★	VK = K VECTOR OF SATELLITE VELOCITY	ORB00680
		ORB00690
★	A LIST OF THE ARRAYS USED FOLLOWS:	ORB00700
		ORB00710
★	AINRAY = INCLINATION	ORB00720
★	APRAY = ARGUMENT OF PERIGEE	ORB00730
★	RARRAY = RADIUS	ORB00740
★	RIRAY = I VECTOR OF RADIUS	ORB00750
★	RJRAY = J VECTOR OF RADIUS	ORB00760
★	RKRAY = K VECTOR OF RADIUS	ORB00770
★	TARAY = TRUE ANOMALY	ORB00780
★	TIMRAY = TIME	ORB00790
		ORB00800
★	A LIST OF SUBROUTINES CALLED BY THE MAIN PROGRAM WILL FOLLOW:	ORB00810
		ORB00820
★	CALCEL = CALCULATES THE ORBITAL ELEMENTS	ORB00830
★	CHGVEL = ALLOW THE USER TO CHANGE THE VELOCITY OF THE SATELLITE	ORB00840
★	INPUTS = PROMPTS USER FOR INITIAL POSITION AND VELOCITY	ORB00850
★	INTSUM = INITIALIZES THE SUMS IN THE ARRAYS	ORB00860
★	NEWELT = CALCULATE NEW ORBITAL ELEMENTS FROM TIME STEP	ORB00870
★	NEWPOS = CALCULATE NEW POSITION VECTOR	ORB00880
★	NEWVEL = CALCULATE NEW VELOCITY VECTOR	ORB00890
★	OPTION = GIVE THE USER THE OPTIONS Permitted IN THE PROGRAM	ORB00900
★	PLOTS = PLOTS THE ORBITS	ORB00910
★	PRETUR = CALCULATES THE PERTURBED ORBIT	ORB00920
★	STORE = STORE THE POSITION DATA IN ARRAYS	ORB00930
★	UNPRET = CALCULATE THE UNPERTURBED ORBIT	ORB00940
		ORB00950
★	BEGIN MAIN PROGRAM	ORB00960
		ORB00970
	DOUBLE PRECISION PI,MU,RI,RJ,RK,R,VI,VJ,VK,V,HI,HJ,HK,H,	ORB00980
+	NI,NJ,NK,N,P,EI,EJ,EK,E,A,I,LAN,AP,TA,AL,LP,TL,PER,EA,	ORB00990
+	MM,MA,T,DT,TF,FR,FS,FW,TT,CHTA,RA,VA,TEMPTA,RE	ORB01000
		ORB01010
	DIMENSION TARAY(500),RARRAY(500),RIRAY(500),RJRAY(500),RKRAY(500),	ORB01020
+	AINRAY(500),APRAY(500),TIMRAY(500)	ORB01030
		ORB01040
	CHARACTER*1,LOOP,YORN,ORLOOP	ORB01050
		ORB01060
	PI = 3.141592653589794	ORB01070

	MU = 3.986012D+05	ORB01080
	RE = 6.378145D+03	ORB01090
*	USER INTRO TO PROGRAM	ORB01100
	CALL INTRO	ORB01110
*	ENTERED MAIN PROGRAM LOOP	ORB01120
	LOOP = 'Y'	ORB01130
10	IF (LOOP .EQ. 'Y') THEN	ORB01140
		ORB01150
*	INITIALIZE STEP COUNTER AND TRUE TIME	ORB01160
20	NUM = 1	ORB01170
	TT = 0.0	ORB01180
		ORB01190
*	PROMPT USER FOR INITIAL POSITION AND VELOCITY	ORB01200
	CALL INPUTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,LOOP,PI)	ORB01210
		ORB01220
*	EXIT PROGRAM	ORB01230
	IF (LOOP .EQ. 'N') THEN	ORB01240
	GOTO 10	ORB01250
	ENDIF	ORB01260
		ORB01270
*	CALCULATE AND STORE ORBITAL ELEMENTS	ORB01280
	CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,	ORB01290
+	LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB01300
	CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,	ORB01310
+	NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB01320
		ORB01330
*	PRINT DATE FOR USER TO REVIEW	ORB01340
	PRINT*, 'VI =', VI, ' KM/S'	ORB01350
	PRINT*, 'VJ =', VJ, ' KM/S'	ORB01360
	PRINT*, 'VK =', VK, ' KM/S'	ORB01370
	PRINT*, 'V =', V, ' KM/S'	ORB01380
	PRINT*, 'RI =', RI, ' KM'	ORB01390
	PRINT*, 'RJ =', RJ, ' KM'	ORB01400
	PRINT*, 'RK =', RK, ' KM'	ORB01410
	PRINT*, 'R =', R, ' KM'	ORB01420
	PRINT*, 'ECCENTRICITY =', E	ORB01430
	DEGI = SNGL((180.0/PI)*I)	ORB01440
	PRINT*, 'INCLINATION =', DEGI, ' DEGREES'	ORB01450
	PERHRS = SNGL(PER/3600.0)	ORB01460
	PRINT*, 'PERIOD =', PERHRS, ' HOURS'	ORB01470
	PRINT*, 'ARE THESE VALUES CORRECT? ENTER "Y" OR "N" :'	ORB01480
	READ*, YORN	ORB01490
	CALL EXCMS('CLRSCRN')	ORB01500
	IF (.NOT. YORN .EQ. 'Y') THEN	ORB01510
	GOTO 20	ORB01520
	ENDIF	ORB01530
		ORB01540
*	CALCULATE TIME STEP AND SET TIMER TO ONE TIME STEP	ORB01550
	DT = PER/50	ORB01560
	T = DT	ORB01570
		ORB01580
*	STEP SATELLITE TO PERIGEE POINT AND RECORD	ORB01590
50	IF ((TA.GT. 0.063).AND. (TA.LT. 6.21)) THEN	ORB01600
	TT = TT + DT	ORB01610
		ORB01620
		ORB01630

	CALL NEWELT(MM,MA,E,EA,TA,TF,PI,PER)	ORB01640
	CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E)	ORB01650
	CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU)	ORB01660
	NUM = NUM + 1	ORB01670
	CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,	ORB01680
	NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB01690
	T = T + DT	ORB01700
	GOTO 50	ORB01710
	ENDIF	ORB01720
*	CALCULATE ELEMENTS FROM PERIGEE POINT	ORB01730
	CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,	ORB01740
+	LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB01750
		ORB01760
		ORB01770
	DT = PER/30	ORB01780
	T = DT	ORB01790
		ORB01800
*	STORE FIRST Unperturbed ORBIT	ORB01810
	CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB01820
+	MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,T,NUM,RIRAY,RJRAY,	ORB01830
+	RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB01840
		ORB01850
*	INITIALIZE SUMS FOR FORCE AND ORBITAL ELEMENT CHANGES TO ZERO	ORB01860
	CALL INTSUM(TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDM,TDMA,TDLAN,	ORB01870
+	TDH,TDAP)	ORB01880
		ORB01890
*	PLOT FIRST UNPERTURBED ORBIT	ORB01900
70	CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM,PI,I,LP,A,E,TF,	ORB01910
+	AINRAY,APRAY,TIMRAY,TFEA,TFSU,TFMO,TFDRA,PER,TDI,TDA,	ORB01920
+	TDE,TDM,TDMA,TDLAN,TDH,TDAP,MM,MA,LAN,H,AP,R,V)	ORB01930
		ORB01940
*	BEGIN NEW ORBIT OPTIONS	ORB01950
*	IOFT1 = 1. Unperturbed ORBIT	ORB01960
*	= 2. Perturbed ORBIT	ORB01970
*	= 3. QUIT	ORB01980
*	IOFT2 = 1. PLOT NEXT ORBIT	ORB01990
*	= 2. CHANGE INITIAL VALUES	ORB02000
*	= 3. CHANGE VELOCITY AT A SPECIFIC TRUE Anomaly	ORB02010
*	= 4. PLOT ANOTHER VIEW OF SAME ORBIT	ORB02020
*		ORB02030
*	ALSO ASKED IF WANT TO CLEAR ALL PREVIOUS ORBITS	ORB02040
		ORB02050
*	CALCULATE ELEMENTS AT PERIGEE	ORB02060
80	CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,	ORB02070
+	LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB02080
		ORB02090
*	CHECK FOR POSSIBLE ARRAY OVERFLOW	ORB02100
	IF (NUM .GT. 425) THEN	ORB02110
	PRINT*, 'ARRAYS ARE FULL'	ORB02120
	PRINT*, 'PREVIOUS ORBITS WILL BE ERASED!'	ORB02130
	NUM = 1	ORB02140
	CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,	ORB02150
+	NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB02160
	ENDIF	ORB02170
		ORB02180
*	PROMPT USER FOR DESIRED OPTION	ORB02190

	CALL OPTION(IOPT1, IOPT2, NUM, RIRAY, RJRAY, RKRAY, RARAY,	ORB02200
+	TARAY, AINRAY, APRAY, TIMRAY)	ORB02210
		ORB02220
*	Initialize SUMS FOR FORCE AND ORBITAL ELEMENT CHANGES TO ZERO	ORB02230
	CALL INTSUM(TFEA, TFSU, TFMO, TFDRA, TDI, TDA, TDE, TDMM, TDMA, TDLAN,	ORB02240
+	TDH, TDAP)	ORB02250
		ORB02260
*	SET TIME COUNTER TO ONE TIME STEP	ORB02270
	T = DT	ORB02280
		ORB02290
*	OPTION: PLOT THE NEXT ORBIT	ORB02300
	IF (IOPT2 .EQ. 1) THEN	ORB02310
		ORB02320
*	CALCULATE AND PLOT UNPERTURBED ORBIT	ORB02330
	IF (IOPT1 .EQ. 1) THEN	ORB02340
	CALL UNPRET(DT, PER, AL, LAN, AP, I, RI, RJ,	ORB02350
+	RK, R, VI, VJ, VK, V, MU, PI, H, A,	ORB02360
+	E, N, TA, P, MM, MA, EA, TF, T, NUM, RIRAY, RJRAY, RKRAY,	ORB02370
+	RARAY, TARAY, AINRAY, APRAY, TIMRAY, TT)	ORB02380
	CALL PLOTS(RIRAY, RJRAY, RKRAY, RARAY, TARAY, NUM,	ORB02390
+	PI, I, LP, A, E, TF, AINRAY, APRAY, TIMRAY,	ORB02400
+	TFEA, TFSU, TFMO, TFDRA, PER,	ORB02410
+	TDI, TDA, TDE, TDMM, TDMA, TDLAN, TDH, TDAP,	ORB02420
+	MM, MA, LAN, H, AP, R, V)	ORB02430
		ORB02440
*	CALCULATE AND PLOT PERTURBED ORBIT	ORB02450
	ELSEIF (IOPT1 .EQ. 2) THEN	ORB02460
	CALL PRETUR(DT, PER, AL, LAN, AP, I,	ORB02470
+	RI, RJ, RK, R, VI, VJ, VK, V, FR, FS, FW,	ORB02480
+	MU, PI, H, A, E, N, TA, P, MM, MA, EA, TF, T, NUM,	ORB02490
+	RIRAY, RJRAY, RKRAY, RARAY, TARAY, AINRAY, APRAY,	ORB02500
+	TIMRAY, TT, TFEA, TFSU, TFMO, TFDRA,	ORB02510
+	TDI, TDA, TDE, TDMM, TDMA, TDLAN, TDH, TDAP)	ORB02520
	CALL PLOTS(RIRAY, RJRAY, RKRAY, RARAY, TARAY, NUM,	ORB02530
+	PI, I, LP, A, E, TF, AINRAY, APRAY, TIMRAY,	ORB02540
+	TFEA, TFSU, TFMO, TFDRA, PER,	ORB02550
+	TDI, TDA, TDE, TDMM, TDMA, TDLAN, TDH, TDAP,	ORB02560
+	MM, MA, LAN, H, AP, R, V)	ORB02570
	ENDIF	ORB02580
		ORB02590
*	GOTO THE BEGINNING OF THE PROGRAM TO CHANGE THE INITIAL VALUES	ORB02600
	ELSEIF (IOPT2 .EQ. 2) THEN	ORB02610
	GOTO 20	ORB02620
		ORB02630
*	CHANGE VELOCITY AT A SPECIFIC TRUE ANOMALY AND	ORB02640
*	PLOT THE NEW ORBIT	ORB02650
	ELSEIF (IOPT2 .EQ. 3) THEN	ORB02660
	CALL CHGVEL(DT, PER, AL, LAN, AP, I, RI, RJ, RK, R,	ORB02670
+	VI, VJ, VK, V, MU, PI,	ORB02680
+	H, A, E, N, TA, P, MM, MA, EA, TF, T, NUM, RIRAY,	ORB02690
+	RJRAY, RKRAY, RARAY, TARAY, AINRAY, APRAY,	ORB02700
+	TIMRAY, TT, EI, EJ, EK, LP, HI, HJ, IOPT1,	ORB02710
+	TFEA, TFSU, TFMO, TFDRA, TDI, TDA, TDE, TDMM,	ORB02720
+	TDMA, TDLAN, TDH, TDAP)	ORB02730
	CALL PLOTS(RIRAY, RJRAY, RKRAY, RARAY, TARAY, NUM,	ORB02740
+	PI, I, LP, A, E, TF, AINRAY, APRAY, TIMRAY,	ORB02750

+	TFEA,TFSU,TFMO,TFDRA,PER,	ORB02760
+	TDI,TDA,TDE,TDMX,TDMA,TDLAN,TDH,TDAP,	ORB02770
+	MI,MA,LAN,H,AP,R,V)	ORB02780
☆	PLOT ANOTHER VIEW OF THE SAME ORBIT	ORB02790
	ELSEIF (IOPT2 .EQ. 4) THEN	ORB02800
	CALL PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM,	ORB02810
+	PI,I,LP,A,E,TF,AINRAY,APRAY,TIMRAY,	ORB02820
+	TFEA,TFSU,TFMO,TFDRA,PER,	ORB02830
+	TDI,TDA,TDE,TDMX,TDMA,TDLAN,TDH,TDAP,	ORB02840
+	MI,MA,LAN,H,AP)	ORB02850
		ORB02860
		ORB02870
☆	STOP THE PROGRAM	ORB02880
	ELSEIF (IOPT2 .EQ. 5) THEN	ORB02890
	GOTO 90	ORB02900
	ELSE	ORB02910
	PRINT*, 'INVALID ENTRY!'	ORB02920
	GOTO 80	ORB02930
	ENDIF	ORB02940
☆	CHECK IF SATELLITE Impacted THE EARTH AND GO TO THE BEGINNING	ORB02950
	IF (R .LE. 6450.0) THEN	ORB02960
	PRINT*, 'SATELLITE WILL IMPACT THE EARTH!!!'	ORB02970
	PRINT*, 'PROGRAM WILL RESET TO THE BEGINNING'	ORB02980
	GOTO 20	ORB02990
	ENDIF	ORB03000
☆	GOTO THE TOP OF THE OPTION LOOP	ORB03010
	GOTO 60	ORB03020
☆	GIVE THE USER A CHANCE TO RECOVER THE PROGRAM	ORB03030
90	PRINT*, 'THIS IS YOUR LAST CHANCE!'	ORB03040
	PRINT*, 'DO YOU WANT TO CONTINUE?'	ORB03050
	PRINT*, 'AND GOTO THE Beginning OF THE PROGRAM?'	ORB03060
	PRINT*, 'ENTER "Y" OR "N" :'	ORB03070
	READ*, LOOP	ORB03080
	PRINT*, LOOP	ORB03090
	GOTO 10	ORB03100
	ENDIF	ORB03110
☆	DISSPLA SUBROUTINE TO TELL GRAPHICS TERMINAL PLOTTING	ORB03120
☆	SESSION IS DONE	ORB03130
	CALL DONEPL	ORB03140
	STOP	ORB03150
	END	ORB03160
	*****	ORB03170
		ORB03180
		ORB03190
		ORB03200
		ORB03210
		ORB03220
		ORB03230
	SUBROUTINE INTRO	ORB03240
☆	THIS SUBROUTINE WILL GIVE THE USER A Brief INTRODUCTION OF THE	ORB03250
☆	USES OF THE PROGRAM	ORB03260
		ORB03270
	PRINT*, 'THIS PROGRAM IS A GRAPHICS DISPLAY OF Satellite ORBITS.'	ORB03280
	PRINT*, 'YOU WILL BE ASKED TO INPUT THE INITIAL VELOCITY AND'	ORB03290
	PRINT*, 'POSITION VECTORS OF THE Satellite. THE PROGRAM WILL'	ORB03300
	PRINT*, 'THEN CALCULATE THE ORBITAL PARAMETERS AND THE'	ORB03310

PRINT*, 'Unperturbed ORBIT. THE USER WILL THEN HAVE THE'	ORB03320
PRINT*, 'CHOICE OF DISPLAYS: '	ORB03330
PRINT*, ' -PERIFICAL (SHOWS RELATIVE SIZE OF ORBIT)'	ORB03340
PRINT*, ' -Equatorial (SHOWS ORBIT INCLINED, USER INPUT'	ORB03350
PRINT*, ' LONGITUDE TO VIEW AT)'	ORB03360
PRINT*, ' -GROUND TRACK'	ORB03370
PRINT*, ' '	ORB03380
PRINT*, 'THE USER IS THEN ASKED TO CHOOSE ONE OF THE FOLLOWING: '	ORB03390
PRINT*, ' -Unperturbed ORBITS'	ORB03400
PRINT*, ' -Perturbed ORBITS'	ORB03410
PRINT*, ' -VELOCITY CHANGES'	ORB03420
PRINT*, 'THE USER'S CHOICE WILL BE USED IN DEVELOPING THE'	ORB03430
PRINT*, 'GRAPHICAL OUTPUT.'	ORB03440
PRINT*, ' '	ORB03450
PRINT*, 'THE USER IS THEN GIVEN THE FOLLOWING CHOICES: '	ORB03460
PRINT*, ' -CLEAR ALL THE PREVIOUS ORBITS'	ORB03470
PRINT*, ' -CHANGE THE INITIAL PARAMETERS'	ORB03480
PRINT*, ' -CHANGE VELOCITY AT A SPECIFIC TRUE Anomaly'	ORB03490
PRINT*, ' -PLOT ANOTHER VIEW OF THE SAME ORBIT'	ORB03500
RETURN	ORB03510
END	ORB03520

SUBROUTINE OPTION(IOPT1, IOPT2, NUM, RIRAY, RJRAY, RKRAY, RARAY,	ORB03530
+ TARAY, AINRAY, APRAY, TIMRAY)	ORB03540
* THIS SUBROUTINE GIVES THE USER A CHOICE OF OPERATIONS THAT CAN BE	ORB03550
* PERFORMED ON THE PROGRAM AND RETURNS THE USERS CHOICE WITH	ORB03560
* VARIABLES IOPT1 AND IOPT2	ORB03570
DIMENSION RIRAY(500), RJRAY(500), RKRAY(500), RARAY(500), TARAY(500),	ORB03580
+ AINRAY(500), APRAY(500), TIMRAY(500)	ORB03590
CHARACTER*1, YORN	ORB03600
IOPT1 = 0	ORB03610
* PROMPT USER FOR OPTION	ORB03620
103 PRINT*, 'WHICH OF THE FOLLOWING OPTIONS WOULD YOU LIKE: '	ORB03630
PRINT*, ' 1. -CALCULATE THE NEXT ORBIT USING THE SAME'	ORB03640
PRINT*, ' PARAMETERS'	ORB03650
PRINT*, ' 2. -CHANGE THE INITIAL PARAMETERS OF THE ORBIT'	ORB03660
PRINT*, ' 3. -CHANGE THE VELOCITY AT A POINT IN THE ORBIT'	ORB03670
PRINT*, ' (THE UNPERTURBED ORBIT WILL BE USED)'	ORB03680
PRINT*, ' 4. -PLOT ANOTHER VIEW OF THE ORBIT(S)'	ORB03690
PRINT*, ' 5. -QUIT'	ORB03700
PRINT*, 'ENTER 1, 2, 3, 4, OR 5: '	ORB03710
READ*, IOPT2	ORB03720
PRINT*, IOPT2	ORB03730
CALL EXCNS('CLRSCRN')	ORB03740
IF (IOPT2 .GT. 5) THEN	ORB03750
GOTO 103	ORB03760
ENDIF	ORB03770
* Prompt USER FOR TYPE OF ORBIT DESIRED	ORB03780
105 IF (IOPT2 .EQ. 1) THEN	ORB03790
PRINT*, 'WHICH TYPE OF ORBIT WOULD YOU LIKE TO SEE, '	ORB03800
PRINT*, ' 1. -Unperturbed ORBITS'	ORB03810
	ORB03820
	ORB03830
	ORB03840
	ORB03850
	ORB03860
	ORB03870

```

PRINT*, ' 2.-Perturbed ORBITS'
PRINT*, ' ENTER 1 OR 2:'
READ*, IOPT1
PRINT*, IOPT1
CALL EXCMS('CLRSCRN')
IF ((IOPT1 .NE. 1) .AND. (IOPT1 .NE. 2)) THEN
    PRINT*, 'INVALID ENTRY!'
    GOTO 105
ENDIF
ENDIF
107 * PROMPT USER TO CLEAR PREVIOUS ORBITS
IF ((IOPT2 .EQ. 1) .OR. (IOPT2 .EQ. 3)) THEN
    PRINT*, 'DO YOU WANT TO CLEAR THE PREVIOUS ORBITS?'
    PRINT*, 'ENTER "Y" OR "N" : '
    READ*, YORN
    PRINT*, YORN
    CALL EXCMS('Clrscrn')
    IF (YORN .EQ. 'Y') THEN
        RIRAY(1) = RIRAY(NUM)
        RJRAY(1) = RJRAY(NUM)
        RKRAY(1) = RKRAY(NUM)
        RARAY(1) = RARAY(NUM)
        TARAY(1) = TARAY(NUM)
        AINRAY(1) = AINRAY(NUM)
        APRAY(1) = APRAY(NUM)
        TIMRAY(1) = TIMRAY(NUM)
        NUM = 1
    ELSEIF (YORN .NE. 'N') THEN
        PRINT*, 'INVALID ENTRY!!'
        PRINT*, 'ALL INPUTS MUST BE CAPITOL LETTERS'
        GOTO 107
    ENDIF
ENDIF
* CHECK FOR INVALID OPTION
IF ((IOPT2 .NE. 1) .AND. (IOPT2 .NE. 2) .AND. (IOPT2 .NE. 3) .AND.
+ (IOPT2 .NE. 4) .AND. (IOPT2 .NE. 5)) THEN
    PRINT*, 'INVALID ENTRY!'
    GOTO 103
ENDIF
RETURN
END

*****
* COORDINATE TRANSFORMATIONS
*****

SUBROUTINE PQWIJK(LAN,AP,INC,P,Q,W,I,J,K)
* THIS SUBROUTINE TRANSFORMS PQW COORDINATES TO IJK COORDINATES

DOUBLE PRECISION INC,P,Q,W,I,J,K,R11,R12,R13,R21,R22,R23,
+ R31,R32,R33,LAN,AP
R11 = DCOS(LAN)*DCOS(AP) - DSIN(LAN)*DSIN(AP)*DCOS(INC)
R12 = -DCOS(LAN)*DSIN(AP) - DSIN(LAN)*DCOS(AP)*DCOS(INC)
R13 = DSIN(LAN)*DSIN(INC)

```

```

ORB03880
ORB03890
ORB03900
ORB03910
ORB03920
ORB03930
ORB03940
ORB03950
ORB03960
ORB03970
ORB03980
ORB03990
ORB04000
ORB04010
ORB04020
ORB04030
ORB04040
ORB04050
ORB04060
ORB04070
ORB04080
ORB04090
ORB04100
ORB04110
ORB04120
ORB04130
ORB04140
ORB04150
ORB04160
ORB04170
ORB04180
ORB04190
ORB04200
ORB04210
ORB04220
ORB04230
ORB04240
ORB04250
ORB04260
ORB04270
ORB04280
ORB04290
ORB04300
ORB04310
ORB04320
ORB04330
ORB04340
ORB04350
ORB04360
ORB04370
ORB04380
ORB04390
ORB04400
ORB04410
ORB04420
ORB04430

```

R21 = DSIN(LAN)*DCOS(AP) + DCOS(LAN)*DSIN(AP)*DCOS(INC)	ORB04440
R22 = -DSIN(LAN)*DSIN(AP) + DCOS(LAN)*DCOS(AP)*DCOS(INC)	ORB04450
R23 = -DCOS(LAN)*DSIN(INC)	ORB04460
R31 = DSIN(AP)*DSIN(INC)	ORB04470
R32 = DCOS(AP)*DSIN(INC)	ORB04480
R33 = DCOS(INC)	ORB04490
I = R11*P + R12*Q + R13*W	ORB04500
J = R21*P + R22*Q + R23*W	ORB04510
K = R31*P + R32*Q + R33*W	ORB04520
RETURN	ORB04530
END	ORB04540
*****	ORB04550
*****	ORB04560
*****	ORB04570
SUBROUTINE IJKPQW(LAN,AP,INC,I,J,K,P,Q,W)	ORB04580
* THIS SUBROUTINE TRANSFORMS IJK COORDINATES TO PQW COORDINATES	ORB04590
DOUBLE PRECISION INC,I,J,K,P,Q,W,R11,R12,R13,R21,R22,R23,	ORB04600
+ R31,R32,R33,LAN,AP	ORB04610
R11 = DCOS(LAN)*DCOS(AP) - DSIN(LAN)*DSIN(AP)*DCOS(INC)	ORB04620
R21 = -DCOS(LAN)*DSIN(AP) - DSIN(LAN)*DCOS(AP)*DCOS(INC)	ORB04630
R31 = DSIN(LAN)*DSIN(INC)	ORB04640
R12 = DSIN(LAN)*DCOS(AP) + DCOS(LAN)*DSIN(AP)*DCOS(INC)	ORB04650
R22 = -DSIN(LAN)*DSIN(AP) + DCOS(LAN)*DCOS(AP)*DCOS(INC)	ORB04660
R32 = -DCOS(LAN)*DSIN(INC)	ORB04670
R13 = DSIN(AP)*DSIN(INC)	ORB04680
R23 = DCOS(AP)*DSIN(INC)	ORB04690
R33 = DCOS(INC)	ORB04700
P = R11*I + R12*J + R13*K	ORB04710
Q = R21*I + R22*J + R23*K	ORB04720
W = R31*I + R32*J + R33*K	ORB04730
RETURN	ORB04740
END	ORB04750
*****	ORB04760
*****	ORB04770
*****	ORB04780
SUBROUTINE IJKRSW(LAN,AL,INC,I,J,K,R,S,W)	ORB04790
* THIS SUBROUTINE CHANGES FROM IJK COORDINATES TO RSW COORDINATES	ORB04800
DOUBLE PRECISION INC,I,J,K,R,S,W,R11,R12,R13,R21,R22,R23,	ORB04810
+ R31,R32,R33,LAN,AL	ORB04820
R11 = DCOS(LAN)*DCOS(AL) - DSIN(LAN)*DCOS(INC)*DSIN(AL)	ORB04830
R12 = DSIN(LAN)*DCOS(AL) + DSIN(AL)*DCOS(LAN)*DCOS(INC)	ORB04840
R13 = DSIN(INC)*DSIN(AL)	ORB04850
R21 = -DCOS(LAN)*DSIN(AL) - DSIN(LAN)*DCOS(INC)*DCOS(AL)	ORB04860
R22 = -DSIN(LAN)*DSIN(AL) + DCOS(LAN)*DCOS(INC)*DCOS(AL)	ORB04870
R23 = DSIN(INC)*DCOS(AL)	ORB04880
R31 = DSIN(LAN)*DSIN(INC)	ORB04890
R32 = -DCOS(LAN)*DSIN(INC)	ORB04900
R33 = DCOS(INC)	ORB04910
R = R11*I + R12*J + R13*K	ORB04920
S = R21*I + R22*J + R23*K	ORB04930
W = R31*I + R32*J + R33*K	ORB04940
RETURN	ORB04950
END	ORB04960
	ORB04970
	ORB04980

*****		ORB04990
SUBROUTINE RSWIJK(LAN,AL,INC,R,S,W,I,J,K)		ORB05000
*	THIS SUBROUTINE CHANGES FROM RSW COORDINATES TO IJK COORDINATES	ORB05010
DOUBLE PRECISION INC,R,S,W,I,J,K,R11,R12,R13,R21,R22,R23,		ORB05020
+ R31,R32,R33,LAN,AL		ORB05030
R11 = DCOS(LAN)*DCOS(AL) - DSIN(LAN)*DCOS(INC)*DSIN(AL)		ORB05040
R21 = DSIN(LAN)*DCOS(AL) + DSIN(AL)*DCOS(LAN)*DCOS(INC)		ORB05050
R31 = DSIN(INC)*DSIN(AL)		ORB05060
R12 = -DCOS(LAN)*DSIN(AL)-DSIN(LAN)*DCOS(INC)*DCOS(AL)		ORB05070
R22 = -DSIN(LAN)*DSIN(AL) + DCOS(LAN)*DCOS(INC)*DCOS(AL)		ORB05080
R32 = DSIN(INC)*DCOS(AL)		ORB05090
R13 = DSIN(LAN)*DSIN(INC)		ORB05100
R23 = -DCOS(LAN)*DSIN(INC)		ORB05110
R33 = DCOS(INC)		ORB05120
I = R11*R + R12*S + R13*W		ORB05130
J = R21*R + R22*S + R23*W		ORB05140
K = R31*R + R32*S + R33*W		ORB05150
RETURN		ORB05160
END		ORB05170
*****		ORB05180
SUBROUTINE PQWRSW(TA,P,Q,W,R,S,WN)		ORB05190
*	THIS SUBROUTINE CHANGES FROM PQW COORDINATES TO RSW COORDINATES	ORB05200
DOUBLE PRECISION P,Q,W,R,S,WN,R11,R12,R13,R21,R22,R23,		ORB05210
+ R31,R32,R33,TA		ORB05220
R11 = DCOS(TA)		ORB05230
R12 = DSIN(TA)		ORB05240
R13 = 0.0		ORB05250
R21 = -DSIN(TA)		ORB05260
R22 = DCOS(TA)		ORB05270
R23 = 0.0		ORB05280
R31 = 0.0		ORB05290
R32 = 0.0		ORB05300
R33 = 1.0		ORB05310
R = R11*P + R12*Q + R13*W		ORB05320
S = R21*P + R22*Q + R23*W		ORB05330
WN = R31*P + R32*Q + R33*W		ORB05340
RETURN		ORB05350
END		ORB05360
*****		ORB05370
SUBROUTINE RSWPQW(TA,R,S,W,P,Q,WN)		ORB05380
*	THIS SUBROUTINE CHANGES FROM RSW COORDINATES TO PQW COORDINATES	ORB05390
DOUBLE PRECISION R,S,W,P,Q,WN,R11,R12,R13,R21,R22,R23,		ORB05400
+ R31,R32,R33,TA		ORB05410
R11 = DCOS(TA)		ORB05420
R21 = DSIN(TA)		ORB05430
R31 = 0.0		ORB05440
R12 = -DSIN(TA)		ORB05450

R22 = DCOS(TA)	ORB05550
R32 = 0.0	ORB05560
R13 = 0.0	ORB05570
R23 = 0.0	ORB05580
R33 = 1.0	ORB05590
P = R11*R + R12*S + R13*W	ORB05600
Q = R21*R + R22*S + R23*W	ORB05610
W = R31*R + R32*S + R33*W	ORB05620
RETURN	ORB05630
END	ORB05640
*****	ORB05650
* STORE ELEMENTS IN ARRAYS	ORB05660
*****	ORB05670
	ORB05680
	ORB05690
SUBROUTINE STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM,	ORB05700
+ I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB05710
* THIS SUBROUTINE STORES THE POSITION AND ELEMENTS IN ARRAYS IN	ORB05720
* SINGLE PRECISION FORM, FOR PLOTTING	ORB05730
	ORB05740
DOUBLE PRECISION RI,RJ,RK,R,TA,I,AP,TT	ORB05750
	ORB05760
DIMENSION RIRAY(500),RJRAY(500),RKRAY(500),RARAY(500),TARAY(500),	ORB05770
+ AINRAY(500),APRAY(500),TIMRAY(500)	ORB05780
	ORB05790
RIRAY(NUM) = SNGL(RI)	ORB05800
RJRAY(NUM) = SNGL(RJ)	ORB05810
RKRAY(NUM) = SNGL(RK)	ORB05820
RARAY(NUM) = SNGL(R)	ORB05830
TARAY(NUM) = SNGL(TA)	ORB05840
AINRAY(NUM) = SNGL(I)	ORB05850
APRAY(NUM) = SNGL(AP)	ORB05860
TIMRAY(NUM) = SNGL(TT)	ORB05870
RETURN	ORB05880
END	ORB05890
*****	ORB05900
* INITIAL POSITION, VELOCITY	ORB05910
*****	ORB05920
	ORB05930
	ORB05940
SUBROUTINE INPUTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,QUIT PI)	ORB05950
* THIS SUBROUTINE GIVES THE USER A CHOICE TO ENTER THE	ORB05960
* INITIAL POSITION AND VELOCITY VECTOR OR TO END THE PROGRAM	ORB05970
* CALCULATE THE INITIAL POSITION AND VELOCITY FROM USER PROMPTED	ORB05980
* INPUTS	ORB05990
	ORB06000
* SUBROUTINES CALLED FROM THIS SUBROUTINE:	ORB06010
* INELTS = Prompts USER FOR ORBITAL ELEMENTS	ORB06020
* IPOS = PROMPTS USER FOR INITIAL POSITION (IJK)	ORB06030
* IVEL = PROMPTS USER FOR INITIAL Velocity (IJK)	ORB06040
	ORB06050
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,MU,PI	ORB06060
CHARACTER*1,QUIT	ORB06070
	ORB06080
* PROMPT USER FOR METHOD TO ENTER INPUTS	ORB06090
195 PRINT*, 'IN WHICH MANNER WOULD YOU LIKE TO INPUT THE INITIAL'	ORB06100

PRINT*, 'POSITION AND VELOCITY OF THE SATELLITE?'	ORB06110
PRINT*, '1: BY Inputting THE INITIAL POSITION AND VELOCITY'	ORB06120
PRINT*, 'VECTORS IN THE PERIFOCAL COORDINATE SYSTEM (IJK)'	ORB06130
PRINT*, '2: BY LETTING THE SATELLITE BE PLACED ON THE "I"'	ORB06140
PRINT*, 'AXIS OF THE (IJK) SYSTEM AT A DESIRED RADIUS OF'	ORB06150
PRINT*, 'PERIGEE(RP) AND INPUTTING EITHER A DESIRED RADIUS'	ORB06160
PRINT*, 'OF APOGEE(RA), A DESIRED ECCENTRICITY(E), OR THE'	ORB06170
PRINT*, 'DESIRED VELOCITY AT THAT RADIUS, AND A DESIRED'	ORB06180
PRINT*, 'INCLINATION(I).'	ORB06190
PRINT*, '3: QUIT'	ORB06200
PRINT*, 'ENTER 1, 2 OR 3: '	ORB06210
READ*, ICHC	ORB06220
PRINT*, ICHC	ORB06230
CALL EXCMS('CLRSCRN')	ORB06240
* USER INPUTS POSITION AND VELOCITY VECTORS	ORB06250
IF (ICHC .EQ. 1) THEN	ORB06260
CALL IFOS(RI,RJ,RK,R)	ORB06270
CALL IVEL(VI,VJ,VK,V,R,MU)	ORB06280
* USER INPUTS ORBITAL ELEMENTS TO GET POSITION AND VELOCITY	ORB06290
ELSEIF (ICHC .EQ. 2) THEN	ORB06300
CALL INELTS(RI,RJ,RK,R,VI,VJ,VK,V,MU,PI)	ORB06310
* STOP PROGRAM	ORB06320
ELSEIF (ICHC .EQ. 3) THEN	ORB06330
QUIT = 'N'	ORB06340
ELSE	ORB06350
PRINT*, 'INVALID ENTRY! TRY AGAIN!'	ORB06360
GOTO 195	ORB06370
ENDIF	ORB06380
RETURN	ORB06390
END	ORB06400
*****	ORB06410
SUBROUTINE IFOS(RI,RJ,RK,R)	ORB06420
* THIS SUBROUTINE ASKS THE USER FOR THE INITIAL POSITION OF THE	ORB06430
* Satellite IN GEOCENTRIC-EQUATORIAL COORDINATE SYSTEM	ORB06440
DOUBLE PRECISION RI,RJ,RK,R	ORB06450
CHARACTER*1, CHOICE	ORB06460
LOGICAL CORREC	ORB06470
CORREC = .FALSE.	ORB06480
* PROMPT USER FOR VELOCITY VECTOR	ORB06490
180 IF(.NOT.CORREC) THEN	ORB06500
CALL EXCMS('CLRSCRN')	ORB06510
PRINT*, 'ENTER RADIUS VECTOR VALUES IN "KM"'	ORB06520
PRINT*, 'RADIUS OF THE EARTH = 6400 KM'	ORB06530
CORREC = .TRUE.	ORB06540
PRINT*, 'ENTER RI : '	ORB06550
READ*, RI	ORB06560
PRINT*, 'RI = ', RI, 'KM'	ORB06570
PRINT*, 'ENTER RJ : '	ORB06580
	ORB06590
	ORB06600
	ORB06610
	ORB06620
	ORB06630
	ORB06640
	ORB06650
	ORB06660

READ*,RJ	ORB06670
PRINT*, 'RJ = ',RJ,'KM'	ORB06680
PRINT*, 'ENTER RK : '	ORB06690
READ*,RK	ORB06700
PRINT*, 'RK = ',RK,'KM'	ORB06710
★ CALCULATE TOTAL R	ORB06720
R = DSQRT((RJ**2) + (RK**2))	ORB06730
PRINT*, 'R = ',R,'KM'	ORB06740
IF (R .LE. 6400.0) THEN	ORB06750
PRINT*, 'RADIUS TOO SMALL!! ENTER NEW VALUES!!'	ORB06760
GOTO 180	ORB06770
ENDIF	ORB06780
★ CHECK WITH USER THAT Values ARE CORRECT	ORB06790
PRINT*, 'ARE THESE VALUES CORRECT?'	ORB06800
PRINT*, 'ENTER "Y" OR "N" : '	ORB06810
READ*,CHOICE	ORB06820
CHOICE = 'Y'	ORB06830
PRINT*,CHOICE	ORB06840
IF (CHOICE.EQ. 'Y') THEN	ORB06850
CORREC = .TRUE.	ORB06860
ENDIF	ORB06870
GOTO 180	ORB06880
ENDIF	ORB06890
RETURN	ORB06900
END	ORB06910
*****	ORB06920
*****	ORB06930
*****	ORB06940
*****	ORB06950
*****	ORB06960
SUBROUTINE IVEL(VI,VJ,VK,V,R,MU)	ORB06970
★ THIS SUBROUTINE ASKS THE USER FOR THE INITIAL VELOCITY OF THE	ORB06980
★ Satellite	ORB06990
	ORB07000
DOUBLE PRECISION VI,VJ,VK,V,R,VCIR,VMAX,MU	ORB07010
	ORB07020
CHARACTER*1, CHOICE	ORB07030
LOGICAL CORREC	ORB07040
CORREC = .FALSE.	ORB07050
	ORB07060
★ CALCULATE ESCAPE VELOCITY AND CIRCULAR VELOCITY AND PROMPT USER	ORB07070
★ FOR VELOCITY VECTOR	ORB07080
190 IF(.NOT.CORREC) THEN	ORB07090
CALL EXCMS('CLRSCRN')	ORB07100
VCIR = DSQRT(MU/R)	ORB07110
VMAX = DSQRT((2.0*MU)/R)	ORB07120
PRINT*, 'CIRCULAR VELOCITY = ',VCIR,'KM/SEC'	ORB07130
PRINT*, 'MAXIMUM VELOCITY = ',VMAX,'KM/SEC'	ORB07140
CORREC = .TRUE.	ORB07150
PRINT*, 'ENTER VELOCITY VECTOR IN (KM/SEC)'	ORB07160
	ORB07170
PRINT*, 'ENTER VI : '	ORB07180
READ*,VI	ORB07190
PRINT*, 'VI = ',VI,'KM/SEC'	ORB07200
PRINT*, 'ENTER VJ : '	ORB07210
READ*,VJ	ORB07220

PRINT*, 'VJ = ', VJ, 'KM/SEC'	ORB07230
PRINT*, 'ENTER VK : '	ORB07240
READ*, VK	ORB07250
PRINT*, 'VK = ', VK, 'KM/SEC'	ORB07260
* CALCULATE TOTAL VELOCITY (V)	ORB07270
V = DSQRT((VI**2) + (VJ**2) + (VK**2))	ORB07280
PRINT*, 'V = ', V, 'KM/SEC'	ORB07290
	ORB07300
	ORB07310
* CHECK WITH USER THAT VALUES ARE CORRECTS	ORB07320
PRINT*, 'ARE THESE VALUES CORRECT?'	ORB07330
PRINT*, ' ENTER "Y" OR "N" : '	ORB07340
READ*, CHOICE	ORB07350
CHOICE = 'Y'	ORB07360
PRINT*, CHOICE	ORB07370
IF (CHOICE.EQ. 'Y') THEN	ORB07380
CORREC = .TRUE.	ORB07390
ENDIF	ORB07400
IF (V .GE. VMAX) THEN	ORB07410
PRINT*, 'VELOCITY IS GREATER THAN THE ESCAPE VELOCITY!!'	ORB07420
PRINT*, 'RE-ENTER VELOCITY!!!'	ORB07430
CORREC = .FALSE.	ORB07440
ENDIF	ORB07450
GOTO 190	ORB07460
ENDIF	ORB07470
RETURN	ORB07480
END	ORB07490
	ORB07500
*****	ORB07510
	ORB07520
SUBROUTINE INELTS(RI, RJ, RK, R, VI, VJ, VK, V, MU, PI)	ORB07530
* SATELLITE PLACED ON 'I' AXIS AND USER SUPPLY ORBITAL ELEMENTS TO	ORB07540
* GET INITIAL POSITION AND VELOCITY	ORB07550
	ORB07560
DOUBLE PRECISION RI, RJ, RK, R, VI, VJ, VK, V, MU, I, ENR, A, E, RP, RA, PI, VMAX	ORB07570
CHARACTER*1, CHOICE	ORB07580
	ORB07590
* PROMPT USER FOR PERIGEE RADIUS	ORB07600
198 PRINT*, 'ENTER RADIUS OF PERIGEE(RP) IN (KM), FOR EXAMPLE: '	ORB07610
PRINT*, 'LOW EARTH ORBIT (LEO), RP = 6600.0 KM'	ORB07620
PRINT*, 'GEOSYNCRONOUS ORBIT, RP = 42241.1 KM'	ORB07630
PRINT*, 'ENTER RP: '	ORB07640
PRINT*, '"RP" MUST BE > 6400KM'	ORB07650
READ*, RP	ORB07660
PRINT*, RP	ORB07670
	ORB07680
* CHECK FOR VALID RADIUS	ORB07690
IF (RP .LT. 6400.0) THEN	ORB07700
PRINT*, 'YOUR "RP" IS TOO SMALL!!'	ORB07710
GOTO 198	ORB07720
ENDIF	ORB07730
	ORB07740
* PROMPT USER FOR TYPE OF INPUT	ORB07750
PRINT*, 'DO YOU WANT TO ENTER THE ECCENTRICITY (E), '	ORB07760
PRINT*, 'RADIUS OF APOGEE (RA), OR VELOCITY (V)?'	ORB07770
PRINT*, 'ENTER "E", "R", OR "V": '	ORB07780

READ*,CHOICE	ORB07790
PRINT*,CHOICE	ORB07800
CALL EXCHS('CLRSCHN')	ORB07810
	ORB07820
★ USER ENTERS Eccentricity AND SEMI-MAJOR AXIS, ENERGY AND VELOCITY	ORB07830
★ IS CALCULATED IN THAT ORDER	ORB07840
IF (CHOICE .EQ. 'E') THEN	ORB07850
PRINT*, 'ENTER ECCENTRICITY (E): '	ORB07860
PRINT*, '0.0 <= E < 1.0'	ORB07870
READ*,E	ORB07880
PRINT*,E	ORB07890
	ORB07900
★ CHECK FOR VALID ECCENTRICITY	ORB07910
IF ((E .LT. 0.0) .OR. (E .GE. 1.0)) THEN	ORB07920
PRINT*, 'INVALID "E"'	ORB07930
GOTO 198	ORB07940
ENDIF	ORB07950
A = RP/(1-E)	ORB07960
ENR = -MU/(2.0*A)	ORB07970
V = DSQRT(2*(ENR+(MU/RP)))	ORB07980
	ORB07990
★ USER INPUTS RADIUS OF APOGEE AND ECCENTRICITY IS CALCULATED	ORB08000
★ THEN SEMI-MAJOR AXIS, ENERGY AND THEN VELOCITY.	ORB08010
ELSEIF (CHOICE .EQ. 'R') THEN	ORB08020
PRINT*, 'ENTER RADIUS OF APOGEE (RA) IN KM: '	ORB08030
PRINT*, '"RA" MUST BE >="RP", "RP" = ',RP	ORB08040
READ*,RA	ORB08050
PRINT*,RA	ORB08060
	ORB08070
★ CHECK FOR VALID RADIUS OF APOGEE	ORB08080
IF (RA .LT. RP) THEN	ORB08090
PRINT*, 'YOUR "RA" IS TOO SMALL!!'	ORB08100
GOTO 198	ORB08110
ENDIF	ORB08120
E = (RA-RP)/(RA+RP)	ORB08130
A = RP/(1-E)	ORB08140
ENR = -MU/(2.0*A)	ORB08150
V = DSQRT(2*(ENR+(MU/RP)))	ORB08160
	ORB08170
★ USER INPUTS MAGNITUDE OF VELOCITY, PROGRAM PROVIDES CIRCULAR	ORB08180
★ AND ESCAPE VELOCITY FOR COMPARISON AND TO CHECK FOR VALID	ORB08190
★ INPUTS	ORB08200
ELSEIF (CHOICE .EQ. 'V') THEN	ORB08210
PRINT*, 'ENTER VELOCITY IN KM/SEC: '	ORB08220
PRINT*, 'THE MINIMUM VELOCITY ALLOWED IS FOR A CIRCULAR ORBIT'	ORB08230
VCIRC = SQRT(SNGL(MU/RP))	ORB08240
PRINT*, 'ORBIT. V(Circular) = ',VCIRC, ' KM/s'	ORB08250
VMAX = DSQRT(2*(MU/RP))	ORB08260
PRINT*, 'THE MAXIMUM VELOCITY < ',VMAX, ' KM/s'	ORB08270
READ*,V	ORB08280
PRINT*,V	ORB08290
IF (V .LT. VCIRC) THEN	ORB08300
PRINT*, 'VELOCITY TOO SMALL!'	ORB08310
GOTO 198	ORB08320
ENDIF	ORB08330
IF (V .GE. VMAX) THEN	ORB08340

PRINT*, 'VELOCITY TO GREAT!!'	ORB08350
GOTO 198	ORB08360
ENDIF	ORB08370
ELSE	ORB08380
PRINT*, 'INVALID ENTRY! TRY AGAIN'	ORB08390
GOTO 198	ORB08400
ENDIF	ORB08410
* INCLINATION NEEDED TO GIVE Velocity A Direction	ORB08420
PRINT*, 'ENTER INCLINATION (I) IN DEGREES:'	ORB08430
READ*, I	ORB08440
PRINT*, I	ORB08450
I = (PI/180.0)*I	ORB08460
VK = V*DSIN(I)	ORB08470
VJ = V*DCOS(I)	ORB08480
VI = 0.0	ORB08490
	ORB08500
	ORB08510
* RADIUS VECTOR SET	ORB08520
RI = RP	ORB08530
RJ = 0.0	ORB08540
RK = 0.0	ORB08550
R = RP	ORB08560
RETURN	ORB08570
END	ORB08580
	ORB08590
*****	ORB08600
* CALCULATE THE ORBITAL ELEMENTS	ORB08610
*****	ORB08620
SUBROUTINE CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,	ORB08630
+ LP,TA,PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB08640
* THIS SUBROUTINE CALLS THE INDIVIDUAL SUBROUTINES TO CALCULATE THE	ORB08650
* ORBITAL ELEMENTS	ORB08660
	ORB08670
	ORB08680
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES(RETURNED VALUES)	ORB08690
* ENERGY = ENERGY PER MASS (ENR)	ORB08700
* ANGOM = ANGULAR MOMENTUM (H,HI,HJ,HK)	ORB08710
* NODE = NODE VECTOR (N,NI,NJ,NK)	ORB08720
* LATREC = SEMI-LATUS RECTUS (P)	ORB08730
* ECC = ECCENTRICITY (E,EI,EJ,EK)	ORB08740
* SMAXIS = SEMI-MAJOR AXIS (A)	ORB08750
* INCL = INCLINATION (I)	ORB08760
* ASNODE = LONGITUDE OF ASCENDING NODE (LAN)	ORB08770
* ARP = ARGUMENT OF PERIGEE (AP)	ORB08780
* IJKPQW = 'IJK' SYSTEM TO 'PQW' SYSTEM	ORB08790
* TANOM = TRUE ANOMALY (TA)	ORB08800
* ARLAT = ARGUMENT OF LATITUDE (AL)	ORB08810
* LONPER = LONGITUDE OF Perigee (LP)	ORB08820
* TLON = TRUE LONGITUDE (TL)	ORB08830
* PERIOD = PERIOD (PER)	ORB08840
* ECCAN = ECCENTRIC ANOMALY (EA)	ORB08850
* MEANMO = MEAN MOTION (MM)	ORB08860
* MEANAN = MEAN ANOMALY (MA)	ORB08870
* TFLGHT = TIME OF FLIGHT (TF)	ORB08880
	ORB08890
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,AL,	ORB08900

+	LF,TA,PER,EA,MA,AP,TF,HI,HJ,HK,H,NI,NJ,NK,N,P,PI,MU,MM,ENR,	ORB08910
+	TL,RP,RQ,RW,NP,NQ,NW	ORB08920
	CALL ENERGY(V,R,MU,ENR)	ORB08930
	CALL ANGOM(RI,RJ,RK,VI,VJ,VK,HI,HJ,HK,H)	ORB08940
	CALL NSDE(HI,HJ,NI,NJ,NK,N)	ORB08950
	CALL LATREC(H,P,MU)	ORB08960
	CALL ECC(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU)	ORB08970
	CALL SMAXIS(MU,ENR,A)	ORB08980
	CALL INCL(HK,H,I,PI)	ORB08990
		ORB09000
*	SPECIAL CASE IF INCLINATION = 0.0	ORB09010
	IF (I.NE. 0.0) THEN	ORB09020
	CALL ASNODE(NI,N,LAN,NJ,PI)	ORB09030
	CALL ARP(NI,NJ,N,EI,EJ,EK,E,AP,PI,NP,NQ,LAN)	ORB09040
	ELSE	ORB09050
	LAN = 0.0	ORB09060
	AP = 0.0	ORB09070
	ENDIF	ORB09080
		ORB09090
*	COORDINATE TRANSFORMATION OF 'R' AND 'V' VECTORS	ORB09100
	CALL IJKPQW(LAN,AP,I,RI,RJ,RK,RP,RQ,RW)	ORB09110
	CALL IJKPQW(LAN,AP,I,NI,NJ,NK,NP,NQ,NW)	ORB09120
	CALL TANON(EI,EJ,EK,E,RI,RJ,RK,RP,RQ,RW,R,VI,VJ,VK,TA,PI)	ORB09130
		ORB09140
		ORB09150
*	SPECIAL CASE FOR Inclination = 0.0	ORB09160
	IF (I.NE. 0.0) THEN	ORB09170
	CALL ARLAT(NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP)	ORB09180
	ELSE	ORB09190
	AL = TA	ORB09200
	ENDIF	ORB09210
	CALL LONPER(LAN,AP,LP)	ORB09220
	CALL TLON(LAN,AP,TA,TL)	ORB09230
	CALL PERIOD(A,PER,PI,MU)	ORB09240
	CALL ECCAN(E,TA,EA,PI)	ORB09250
	CALL MEANMO(A,MM,MU)	ORB09260
	CALL MEANAN(EA,E,MA)	ORB09270
	CALL TFLGHT(MM,MA,TF)	ORB09280
	RETURN	ORB09290
	END	ORB09300
		ORB09310
	*****	ORB09320
	SUBROUTINE ENERGY(V,R,MU,ENR)	ORB09330
*	THIS SUBROUTINE CALCULATES THE ENERGY OF THE ORBIT	ORB09340
		ORB09350
	DOUBLE PRECISION V,R,MU,ENR	ORB09360
		ORB09370
		ORB09380
	ENR = ((V**2)/2) - (MU/R)	ORB09390
	RETURN	ORB09400
	END	ORB09410
		ORB09420
	*****	ORB09430
	SUBROUTINE ANGOM(RI,RJ,RK,VI,VJ,VK,HI,HJ,HK,H)	ORB09440
*	THIS SUBROUTINE CALCULATES THE ANGULAR MOMENTUM	ORB09450
		ORB09460

DOUBLE PRECISION RI,RJ,RK,VI,VJ,VK,HI,HJ,HK,H	ORB09470
	ORB09480
	ORB09490
HI = (RJ * VK) - (RK * VJ)	ORB09500
HJ = (RK * VI) - (RI * VK)	ORB09510
HK = (RI * VJ) - (RJ * VI)	ORB09520
H = DSQRT((HI**2) + (HJ**2) + (HK**2))	ORB09530
RETURN	ORB09540
END	ORB09550
*****	ORB09560
	ORB09570
	ORB09580
SUBROUTINE NODE(HI,HJ,NI,NJ,NK,N)	ORB09590
* THIS SUBROUTINE CALCULATES THE NODE VECTOR	ORB09600
	ORB09610
DOUBLE PRECISION HI,HJ,NI,NJ,NK,N	ORB09620
	ORB09630
NI = -HJ	ORB09640
NJ = HI	ORB09650
NK = 0.0	ORB09660
N = DSQRT((NI**2) + (NJ**2))	ORB09670
RETURN	ORB09680
END	ORB09690
*****	ORB09700
	ORB09710
	ORB09720
SUBROUTINE LATREC(H,P,MU)	ORB09730
* THIS SUBROUTINE CALCULATES THE SEMI-LATUS RECTUM	ORB09740
	ORB09750
DOUBLE PRECISION H,P,MU	ORB09760
	ORB09770
P = (H**2)/MU	ORB09780
RETURN	ORB09790
END	ORB09800
*****	ORB09810
	ORB09820
	ORB09830
SUBROUTINE ECC(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU)	ORB09840
* THIS SUBROUTINE CALCULATES THE ECCENTRICITY	ORB09850
	ORB09860
DOUBLE PRECISION RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,MU,DOT	ORB09870
	ORB09880
* CALCULATE DOT PRODUCT OF 'R' AND 'V' VECTORS	ORB09890
DOT = (RI*VI) + (RJ*VJ) + (RK*VK)	ORB09900
EI = (1.0D+00/MU) * (((V**2) - (MU/R)) * RI - (DOT)*VI)	ORB09910
EJ = (1.0D+00/MU) * (((V**2) - (MU/R)) * RJ - (DOT)*VJ)	ORB09920
EK = (1.0D+00/MU) * (((V**2) - (MU/R)) * RK - (DOT)*VK)	ORB09930
E = DSQRT((EI**2) + (EJ**2) + (EK**2))	ORB09940
RETURN	ORB09950
END	ORB09960
*****	ORB09970
	ORB09980
	ORB09990
SUBROUTINE SMAXIS(MU,ENR,A)	ORB10000
* THIS SUBROUTINE Calculates THE SEMI-MAJOR AXIS	ORB10010
	ORB10020

DOUBLE PRECISION MU,ENR,A	ORB10030
A = -MU/(2*ENR)	ORB10040
RETURN	ORB10050
END	ORB10060
*****	ORB10070
	ORB10080
	ORB10090
	ORB10100
SUBROUTINE INCL(HK,H,I,PI)	ORB10110
* THIS SUBROUTINE CALCULATES THE INCLINATION	ORB10120
* 'I' ALWAYS LESS THAN 180 DEGREES	ORB10130
	ORB10140
DOUBLE PRECISION HK,H,I,PI	ORB10150
	ORB10160
I = DACOS(HK/H)	ORB10170
RETURN	ORB10180
END	ORB10190
*****	ORB10200
	ORB10210
	ORB10220
SUBROUTINE ASNODE(NI,N,LAN,NJ,PI)	ORB10230
* THIS SUBROUTINE CALCULATES THE LONGITUDE OF THE ASCENDING NODE	ORB10240
* IF 'NJ' > 0 THEN 'LAN' < 180 DEGREES	ORB10250
	ORB10260
DOUBLE PRECISION NI,N,LAN,NJ,PI	ORB10270
	ORB10280
LAN = DATAN2(NJ,NI)	ORB10290
IF (LAN .LT. 0.0) THEN	ORB10300
LAN = (2*PI) + LAN	ORB10310
ENDIF	ORB10320
RETURN	ORB10330
END	ORB10340
*****	ORB10350
	ORB10360
	ORB10370
SUBROUTINE ARP(NI,NJ,N,EI,EJ,EK,E,AP,PI,NP,NQ,LAN)	ORB10380
* THIS SUBROUTINE CALCULATES THE ARGUMENT OF Perigee	ORB10390
* IF 'EK' GREATER THAN 0 THEN 'AP' < 180	ORB10400
* VARIABLE TEMP USED AS A Temporary VALUE FOR ARCTAN	ORB10410
	ORB10420
DOUBLE PRECISION NI,NJ,N,EI,EJ,EK,E,AP,PI,NQ,NP,TEMP,LAN	ORB10430
	ORB10440
IF ((EI .EQ. 0.0) .AND. (EJ .EQ. 0.0)) THEN	ORB10450
AP = 0.0	ORB10460
ELSE	ORB10470
TEMP = DATAN2(EJ,EI)	ORB10480
IF (TEMP .GT. LAN) THEN	ORB10490
AP = TEMP - LAN	ORB10500
ELSE	ORB10510
AP = (2*PI) - (LAN - TEMP)	ORB10520
ENDIF	ORB10530
IF (AP .LT. 0.0) THEN	ORB10540
AP = (2*PI) + AP	ORB10550
ENDIF	ORB10560
IF (AP .GT. (2*PI)) THEN	ORB10570
AP = AP - (2*PI)	ORB10580

ENDIF	ORB10590
ENDIF	ORB10600
RETURN	ORB10610
END	ORB10620
*****	ORB10630
SUBROUTINE TANOM(EI,EJ,EK,E,RI,RJ,RK,RP,RQ,RW,R,VI,VJ,VK,	ORB10640
+ TA,PI)	ORB10650
* THIS SUBROUTINE CALCULATES THE TRUE Anomaly	ORB10660
* IF (R DOT V) > 0 THEN TA < 180 DEGREES	ORB10670
	ORB10680
	ORB10690
DOUBLE PRECISION DOT,EI,EJ,EK,E,RI,RJ,RK,R,VI,VJ,VK,TA,PI,	ORB10700
+ RP,RQ,RW	ORB10710
	ORB10720
TA = DATAN2(RQ,RP)	ORB10730
IF (TA .LT. 0.0) THEN	ORB10740
TA = (2 * PI) + TA	ORB10750
ENDIF	ORB10760
RETURN	ORB10770
END	ORB10780
*****	ORB10790
	ORB10800
SUBROUTINE ARLAT(NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP)	ORB10810
* THIS SUBROUTINE CALCULATES THE ARGUMENT OF LATITUDE	ORB10820
* IF (RK > 0) THEN AL < 180 DEGREES	ORB10830
	ORB10840
DOUBLE PRECISION NI,NJ,NK,N,RI,RJ,RK,R,AL,PI,TA,AP	ORB10850
	ORB10860
AL = TA + AP	ORB10870
RETURN	ORB10880
END	ORB10890
*****	ORB10900
	ORB10910
SUBROUTINE LONPER(LAN,AP,LP)	ORB10920
* THIS SUBROUTINE CALCULATES THE LONGITUDE OF PERIGEE	ORB10930
	ORB10940
DOUBLE PRECISION LAN,AP,LP	ORB10950
	ORB10960
LP = LAN + AP	ORB10970
RETURN	ORB10980
END	ORB10990
*****	ORB11000
	ORB11010
SUBROUTINE TLON(LAN,AP,TA,TL)	ORB11020
* THIS SUBROUTINE CALCULATES THE TRUE LONGITUDE AT EPOCH	ORB11030
	ORB11040
DOUBLE PRECISION LAN,AP,TA,TL	ORB11050
	ORB11060
TL = AP + LAN + TA	ORB11070
RETURN	ORB11080
END	ORB11090
	ORB11100
	ORB11110
	ORB11120
	ORB11130

* SUBROUTINE PERIOD(A,PER,PI,MU)
THIS SUBROUTINE CALCULATES THE PERIOD

DOUBLE PRECISION A,PER,PI,MU

PER = 2.0D+00*(PI)*DSQRT((A**3)/MU)
RETURN
END

* SUBROUTINE ECCAN(E,TA,EA,PI)
THIS SUBROUTINE CALCULATES THE ECCENTRIC Anomaly

DOUBLE PRECISION E,TA,EA,PI

EA = DACOS((E + DCOS(TA))/(1.0D+00 + E*DCOS(TA)))
IF (TA .GT. PI) THEN
EA = (2*PI) - EA
ENDIF
RETURN
END

* SUBROUTINE MEANMO(A,MM,MU)
THIS SUBROUTINE CALCULATES THE MEAN MOTION

DOUBLE PRECISION A,MM,MU

MM = DSQRT(MU/(A**3))
RETURN
END

* SUBROUTINE MEANAN(EA,E,MA)
THIS SUBROUTINE CALCULATES THE MEAN Anomaly

DOUBLE PRECISION EA,E,MA

MA = EA - E*DSIN(EA)
RETURN
END

* SUBROUTINE TFLGHT(MM,MA,TF)
THIS SUBROUTINE CALCULATES THE TIME OF FLIGHT

DOUBLE PRECISION MM,MA,TF

TF = (1/MM)*MA

ORB11140
ORB11150
ORB11160
ORB11170
ORB11180
ORB11190
ORB11200
ORB11210
ORB11220
ORB11230
ORB11240
ORB11250
ORB11260
ORB11270
ORB11280
ORB11290
ORB11300
ORB11310
ORB11320
ORB11330
ORB11340
ORB11350
ORB11360
ORB11370
ORB11380
ORB11390
ORB11400
ORB11410
ORB11420
ORB11430
ORB11440
ORB11450
ORB11460
ORB11470
ORB11480
ORB11490
ORB11500
ORB11510
ORB11520
ORB11530
ORB11540
ORB11550
ORB11560
ORB11570
ORB11580
ORB11590
ORB11600
ORB11610
ORB11620
ORB11630
ORB11640
ORB11650
ORB11660
ORB11670
ORB11680
ORB11690

RETURN	ORB11700
END	ORB11710
*****	ORB11720
* CALCULATE UNPERTURBED ORBIT	ORB11730
*****	ORB11740
	ORB11750
	ORB11760
SUBROUTINE UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,	ORB11770
+ VI,VJ,VK,V,MU,PI,H,A,E,N,TA,P,MM,MA,EA,	ORB11780
+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY,	ORB11790
+ TT)	ORB11800
* THIS SUBROUTINE CALCULATE THE UNPERTURBED ORBIT	ORB11810
	ORB11820
* THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES:	ORB11830
* NEWELT = CALCULATE NEW ELEMENTS AFTER TIME STEP	ORB11840
* NEWPOS = CALCULATE NEW POSITION AFTER TIME STEP	ORB11850
* NEWVEL = CALCULATE NEW VELOCITY AFTER TIME STEP	ORB11860
* STORE = STORES POSITION IN ARRAYS	ORB11870
	ORB11880
DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB11890
+ MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT	ORB11900
	ORB11910
DIMENSION RARAY(500),TARAY(500),RIRAY(500),RJRAY(500),	ORB11920
+ RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500)	ORB11930
	ORB11940
* SET TRUE ANOMALY TO NEGATIVE SO LOOP CAN BE EXECUTED	ORB11950
IF (TA .GT. 6.21) THEN	ORB11960
TA = TA - (2*PI)	ORB11970
ENDIF	ORB11980
	ORB11990
* CONTINUE AROUND ORBIT TILL CLOSE TO PERIGEE	ORB12000
230 IF ((TA .LE. 6.21) .AND. (T .LE. PER)) THEN	ORB12010
	ORB12020
* Increment TRUE TIME	ORB12030
TT = TT + DT	ORB12040
CALL NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER)	ORB12050
CALL NPOS(RI,RJ,RK,R,LAN,AP,I, TA,A,E)	ORB12060
CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU)	ORB12070
	ORB12080
* INCREMENT STEP COUNTER AND STORE VALUES	ORB12090
NUM = NUM + 1	ORB12100
CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,	ORB12110
+ RARAY,TARAY,NUM,I,AP,AINRAY,APRAY,	ORB12120
+ TT,TIMRAY)	ORB12130
	ORB12140
* INCREMENT TIME STEP COUNTER	ORB12150
T = T + DT	ORB12160
GOTO 230	ORB12170
ENDIF	ORB12180
RETURN	ORB12190
END	ORB12200
	ORB12210
*****	ORB12220
* CALCULATE THE UNPERTURBED NEW ELEMENTS	ORB12230
*****	ORB12240
	ORB12250

☆	SUBROUTINE NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER)	ORB12260
☆	THIS SUBROUTINE CALCULATES THE Unperturbed NEW ELEMENTS	ORB12270
☆		ORB12280
☆	THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES:	ORB12290
☆	NEA = NEW ECCENTRIC ANOMALY	ORB12300
☆	NTA = NEW TRUE ANOMALY	ORB12310
		ORB12320
	DOUBLE PRECISION MM,MA,E,EA,TA,TF,DT,PI,PER	ORB12330
		ORB12340
☆	Increment TIME OF FLIGHT AND CHECK IF TF GREATER THAN PERIOD	ORB12350
	TF = TF + DT	ORB12360
	IF (TF .GT. PER) THEN	ORB12370
	TF = TF - PER	ORB12380
	ENDIF	ORB12390
		ORB12400
☆	CALCULATE MEAN ANOMALY AND USE TO FIND ECCENTRIC Anomaly THEN NEW	ORB12410
☆	TRUE ANOMALY	ORB12420
	MA = MM*(TF)	ORB12430
	CALL NEA(MA,E,EA)	ORB12440
	CALL NTA(EA,E,TA,PI)	ORB12450
	RETURN	ORB12460
	END	ORB12470
		ORB12480
	*****	ORB12490
☆	CALCULATE PERTURBED ORBIT	ORB12500
	*****	ORB12510
		ORB12520
	SUBROUTINE PRETUR(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,	ORB12530
	+ VI,VJ,VK,V,FR,FS,FW,MU,PI,H,A,E,N,TA,P,MM,MA,EA,	ORB12540
	+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY,	ORB12550
	+ TT,TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDMM,TDMA,TDLAN,TDH,TDAP)	ORB12560
☆	THIS SUBROUTINE CALCULATES THE PERTURBED ORBIT.	ORB12570
		ORB12580
☆	THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES:	ORB12590
☆	TFORCE = CALCULATE THE TOTAL PERTURBING FORCE ON THE SATELLITE	ORB12600
☆	PNEWEL = CALCULATE THE Perturbed NEW ELEMENTS	ORB12610
☆	NPOS = NEW POSITION AFTER TIME STEP	ORB12620
☆	NVEL = NEW VELOCITY AFTER TIME STEP	ORB12630
☆	PERIOD = PERIOD OF PERTURBED ORBIT	ORB12640
☆	STORE = STORE POSITION AND ELEMENTS IN ARRAYS FOR PLOTTING	ORB12650
		ORB12660
	DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB12670
	+ FR,FS,FW,MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT,	ORB12680
	+ DI,DA,DE,DDM,DMA,DLAN,DH,DAP,EI,EJ,EK,HI,HJ,LP,M,	ORB12690
	+ DVR,DVS,DVW,DVI,DVJ,DVK	ORB12700
		ORB12710
	DIMENSION RARAY(500),TARAY(500),RIRAY(500),RJRAY(500),	ORB12720
	+ RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500)	ORB12730
		ORB12740
☆	SET MEAN RADIUS OF EARTH	ORB12750
	RE = 6400.0	ORB12760
		ORB12770
	DT = PER/50	ORB12780
	T = DT	ORB12790
	IF (TA .GT. 6.21) THEN	ORB12800
	TA = TA - (2*PI)	ORB12810

ENDIF	ORB12820
IF (TF .GE. PER) THEN	ORB12830
TF = TF - PER	CRB12840
ENDIF	ORB12850
* CONTINUE Around ORBIT FOR ONE PERIOD	CRB12860
240 IF ((TF .LT. PER) .AND. (T .LT. PER)) THEN	ORB12870
	ORB12880
	ORB12890
* INCREMENT TRUE TIME	ORB12900
TT = TT + DT	ORB12910
CALL TFORCE(AL, LAN, AP, I, RI, RJ, RK, R, VI, VJ, VK, V,	ORB12920
+ TT, FR, FS, FW, MU, PI,	ORB12930
+ FEA, FSU, FMO, FDRA, FOR,	ORB12940
+ EI, EJ, EK, E, A, T, LP, TA, PER, EA, MA, TF, P,	ORB12950
+ MM, N, H, HI, HJ, DT)	ORB12960
CALL PNEWEL(FR, FS, FW, H, R, A, E, P, TA, DT, I, LAN, AL,	ORB12970
+ AP, P, MM, MA, EA, TF, T, MU, PI,	ORB12980
+ DI, DA, DE, DMM, DMA, DLAN, DH, DAP)	ORB12990
CALL NPOS(RI, RJ, RK, R, LAN, AP, I, TA, A, E)	ORB13000
CALL NVEL(E, P, TA, LAN, AP, I, VI, VJ, VK, V, MU)	ORB13010
	ORB13020
* CALCULATE NEW PERIOD AND RESET TIME STEP AND TIME COUNTER	ORB13030
* IF NOT AT END OF ORBIT	ORB13040
IF (T .LT. (PER-DT)) THEN	ORB13050
CALL PERIOD(A, PER, PI, MU)	ORB13060
DT = PER/50	ORB13070
T = TF	ORB13080
ENDIF	CRB13090
	ORB13100
* INCREMENT STEP COUNTER	ORB13110
NUM = NUM + 1	ORB13120
241 CALL STORE(RI, RJ, RK, R, TA, RIRAY, RJRAY, RKRAY,	ORB13130
+ RARAY, TARAY, NUM, I, AP, AINRAY, APRAY,	ORB13140
+ TT, TIMRAY)	ORB13150
	CRB13160
* TOTAL ELEMENT CHANGES	ORB13170
TDI = TDI + SNGL(ABS(DI))	ORB13180
TDA = TDA + SNGL(ABS(DA))	ORB13190
TDE = TDE + SNGL(ABS(DE))	ORB13200
TDMM = TDMM + SNGL(ABS(DMM))	ORB13210
TDMA = TDMA + SNGL(ABS(DMA))	ORB13220
TDLAN = TDLAN + SNGL(ABS(DLAN))	ORB13230
TDH = TDH + SNGL(ABS(DH))	ORB13240
TDAP = TDAP + SNGL(ABS(DAP))	ORB13250
TFEA = TFEA + FEA	ORB13260
TFSU = TFSU + FSU	ORB13270
TFMO = TFMO + FMO	ORB13280
TFDRA = TFDRA + FDRA	ORB13290
	ORB13300
* CHECK FOR IMPACT	ORB13310
IF (R .LE. RE) THEN	ORB13320
PRINT*, 'SATELLITE WILL IMPACT THE EARTH!!'	ORB13330
T = PER	ORB13340
ENDIF	ORB13350
	ORB13360
* INCREMENT TIME COUNTER	CRB13370

T = T + DT
 GOTO 240
 ENDIF
 RETURN
 END

* CALCULATE THE PERTURBING FORCES

SUBROUTINE TFORCE(AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,TT,
 + FR,FS,FW,MU,PI,FEA,FSU,FMO,FDRA,FOR,
 + EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,TF,P,
 + MM,N,H,HI,HJ,DT)
 * THIS SUBROUTINE SUMS ALL THE PERTURBING FORCES FOR THE TOTAL
 * PERTURBING FORCE.

* THE FOLLOWING SUBROUTINES WERE CALLED:
 * OBERT = OBLATENESS OF THE EARTH
 * FSUN = GRAVITATIONAL Attraction OF THE SUN
 * FMOON = GRAVITATIONAL Attraction OF THE MOON
 * FDRAG = DRAG FORCES

DOUBLE PRECISION FER,FES,FEW,FSR,FSS,FSW,FMR,FMS,FMW,MU,PI,
 + FDR,FDS,FDW,FR,FS,FW,RI,RJ,RK,R,AL,I,TT,LAN,AP,VI,VJ,VK,V,
 + EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,TF,P,
 + MM,N,H,HI,HJ,DT

CALL OBEART(RI,RJ,RK,R,AL,I,FER,FES,FEW,MU)
 CALL FSUN(TT,RI,RJ,RK,R,FSR,FSS,FSW,PI)
 CALL FMOON(TT,RI,RJ,RK,R,FMR,FMS,FMW,PI)
 CALL FDRAG(RI,RJ,RK,R,VI,VJ,LAN,AP,I,FDR,FDS,FDW,
 + EI,EJ,EK,E,LP,TA,PER,EA,MA,AL,TF,P,PI,MU,
 + MM,N,H,HI,HJ,DT)

* SUM VECTOR FORCES
 FR = FER + FSR + FMR + FDR
 FS = FES + FSS + FMS + FDS
 FW = FEW + FSW + FMW + FDW

* CALCULATE TOTAL FORCE FROM EACH, AND TOTAL OF ALL
 FEA = SNGL(SQRT((FER**2)+(FES**2)+(FEW**2)))
 FSU = SNGL(SQRT((FSR**2)+(FSS**2)+(FSW**2)))
 FMO = SNGL(SQRT((FMR**2)+(FMS**2)+(FMW**2)))
 FDRA = SNGL(SQRT((FDR**2)+(FDS**2)+(FDW**2)))
 FOR = SNGL(SQRT((FR**2)+(FS**2)+(FW**2)))

RETURN
 END

SUBROUTINE OBEART(RI,RJ,RK,R,AL,I,FER,FES,FEW,MU)
 * THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO THE
 * OBLIQUENESS OF THE EARTH.

ORB13380
 ORB13390
 ORB13400
 ORB13410
 ORB13420
 ORB13430
 ORB13440
 ORB13450
 ORB13460
 ORB13470
 ORB13480
 ORB13490
 ORB13500
 ORB13510
 ORB13520
 ORB13530
 ORB13540
 ORB13550
 ORB13560
 ORB13570
 ORB13580
 ORB13590
 ORB13600
 ORB13610
 ORB13620
 ORB13630
 ORB13640
 ORB13650
 ORB13660
 ORB13670
 ORB13680
 ORB13690
 ORB13700
 ORB13710
 ORB13720
 ORB13730
 ORB13740
 ORB13750
 ORB13760
 ORB13770
 ORB13780
 ORB13790
 ORB13800
 ORB13810
 ORB13820
 ORB13830
 ORB13840
 ORB13850
 ORB13860
 ORB13870
 ORB13880
 ORB13890
 ORB13900
 ORB13910
 ORB13920
 ORB13930

DOUBLE PRECISION J2,RE,FER,FES,FEW,RI,RJ,RK,R,AL,I,MU,N

J2 = 1.082364D-03

RE = 6.3782E+03

FER = ((-3.0D+00* μ *J2*(RE**2))/(2.0D+00*(R**4)))
+ (1.0D+00 - (3.0D+00* ((DSIN(I))**2) * ((DSIN(AL))**2)))
FES = ((-3.0D+00* μ *J2*(RE**2))/(R**4))
+ (((DSIN(I))**2)* (DSIN(AL))* (DCOS(AL)))
FEW = ((-3.0D+00* μ *J2*(RE**2))/(R**4))
+ (DSIN(I)*DCOS(I)*DSIN(AL))
RETURN
END

SUBROUTINE FSUN(TT,RI,RJ,RK,R,FSR,FSS,FSW,PI)
* THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO THE SUN

* THE FOLLOWING SUBROUTINES ARE CALLED:
* SUNPOS = SUNS POSITION ORBITING AROUND EARTH
* HEVBOD = PERTURBING FORCE FROM A Heavenly BODY

DOUBLE PRECISION FSR,FSS,FSW,PI,
+ RSI,RSJ,RSK,SLAN,SI,SAL,SMU,TT,RI,RJ,RK,R,RS

* SUNS GRAVITATIONAL PARAMETER
SMC = 1.3271544D+11
CALL SUNPOS(TT,RSI,RSJ,RSK,RS,SLAN,SI,SAL,PI)
CALL HEVBOD(RI,RJ,RK,R,RSI,RSJ,RSK,RS,SLAN,SAL,SI,SMU,FSR,FSS,FSW)
RETURN
END

SUBROUTINE FMOON(TT,RI,RJ,RK,R,FMR,FMS,FMW,PI)
* THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO The MOON

* THE FOLLOWING SUBROUTINE ARE CALLED:
* MONPOS = MOONS POSITION ORBITING AROUND THE EARTH
* HEVBOD = PERTURBING FORCE FROM A HEAVENLY BODY

DOUBLE PRECISION FMR,FMS,FMW,RMI,RMJ,RMK,MLAN,MI,MAL,MMU,
+ TT,RI,RJ,RK,R,RM,PI

* MOONS GRAVITATIONAL PARAMETER
MMU = 4.90287D+03
CALL MONPOS(TT,RMI,RMJ,RMK,RM,MLAN,MI,MAL,PI)
CALL HEVBOD(RI,RJ,RK,R,RMI,RMJ,RMK,RM,MLAN,MAL,MI,MMU,FMR,FMS,FMW)
RETURN
END

SUBROUTINE HEVBOD(RI,RJ,RK,R,RPI,RPJ,RPK,RP,LAN,AL,INC,MUP,

ORB13940
ORB13950
ORB13960
ORB13970
ORB13980
ORB13990
ORB14000
ORB14010
ORB14020
ORB14030
ORB14040
ORB14050
ORB14060
ORB14070
ORB14080
ORB14090
ORB14100
ORB14110
ORB14120
ORB14130
ORB14140
ORB14150
ORB14160
ORB14170
ORB14180
ORB14190
ORB14200
ORB14210
ORB14220
ORB14230
ORB14240
ORB14250
ORB14260
ORB14270
ORB14280
ORB14290
ORB14300
ORB14310
ORB14320
ORB14330
ORB14340
ORB14350
ORB14360
ORB14370
ORB14380
ORB14390
ORB14400
ORB14410
ORB14420
ORB14430
ORB14440
ORB14450
ORB14460
ORB14470
ORB14480
ORB14490

+	FHR,FHS,FHW)	ORB14500
*	THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO A	ORB14510
*	HEAVENLY BODY.	ORB14520
		ORB14530
*	THE FOLLOWING SUBROUTINE WAS CALLED:	ORB14540
*	IJKRSW = 'IJK' SYSTEM TO THE 'RSW' SYSTEM	ORB14550
		ORB14560
	DOUBLE PRECISION DOT,FHI,FHJ,FHK,RI,RJ,RK,R,RPI,RPJ,RPK,RP,	ORB14570
+	LAN,AL,INC,MUP,I,J,K,IP,JP,KP,M,FHR,FHS,FHW	ORB14580
		ORB14590
*	CALCULATE UNIT VECTOR FOR SATELLITE AND PERTURBING BODIES POSITION	ORB14600
	I = RI/R	ORB14610
	J = RJ/R	ORB14620
	K = RK/R	ORB14630
	IP = RPI/RP	ORB14640
	JP = RPJ/RP	ORB14650
	KP = RPK/RP	ORB14660
		ORB14670
*	CALCULATE DOT PRODUCT OF UNIT VECTORS	ORB14680
	DOT = ((I*IP)+(J*JP)+(K*KP))	ORB14690
		ORB14700
*	CALCULATE FORCES IN THE 'IJK' SYSTEM	ORB14710
	FHI = (MUP/(RP**2))*(K/RP)*(3.0D+00*DOT*(IP)-(I))	ORB14720
	FHJ = (MUP/(RP**2))*(R/RP)*(3.0D+00*DOT*(JP)-(J))	ORB14730
	FHK = (MUP/(RP**2))*(R/RP)*(3.0D+00*DOT*(KP)-(K))	ORB14740
		ORB14750
*	Transform FORCES TO THE RSW SYSTEM	ORB14760
	CALL IJKRSW(LAN,AL,INC,FHI,FHJ,FHK,FHR,FHS,FHW)	ORB14770
	RETURN	ORB14780
	END	ORB14790
		ORB14800
	*****	ORB14810
		ORB14820
	SUBROUTINE SUNPOS(TT,RSI,RSJ,RSK,RS,SLAN,SI,SAL,PI)	ORB14830
*	THIS SUBROUTINE CALCULATES THE SUNS POSITION	ORB14840
		ORB14850
*	VARIABLES USED TO DESCRIBE THE SUNS ORBIT:	ORB14860
*	SI = SUNS INCLINATION	ORB14870
*	SLAN= SUNS Longitude OF ASCENDING NODE	ORB14880
*	SAP = SUNS ARGUMENT OF PERIGEE	ORB14890
*	RS = SUNS ORBITAL RADIUS	ORB14900
*	STA = SUNS TRUE ANOMALY	ORB14910
*	SAL = SUNS ARGUMENT OF LONGITUDE	ORB14920
		ORB14930
	DOUBLE PRECISION SLAN,SI,SAL,RS,STA,SAP,TT,RSI,RSK,	ORB14940
+	RSJ,RSP,RSQ,RSW,PI	ORB14950
		ORB14960
	SI = 4.09279709D-01	ORB14970
	SLAN = 0.0D+00	ORB14980
	SAP = 0.0D+00	ORB14990
	RS = 1.4959965D+08	ORB15000
	STA = ((2.0*PI)/(365.0 * 86400.0) * TT)	ORB15010
	SAL = STA + SAP	ORB15020
		ORB15030
*	CALCULATE SUNS POSITION IN 'PQW' SYSTEM	ORB15040
	RSP = RS*DCOS(STA) .	ORB15050

RSQ = RS*DSIN(STA)	ORB15060
RSW = 0.0D+00	ORB15070
★ TRANSFORM POSITION TO 'IJK' SYSTEM	ORB15080
CALL PQWIJK(SLAN,SAP,SI,RSP,RSQ,RSW,RSI,RSJ,RSK)	ORB15090
RETURN	ORB15100
END	ORB15110
*****	ORB15120
	ORB15130
	ORB15140
SUBROUTINE MONPOS(TT,RMI,RMJ,RMK,RM,MLAN,MI,MAL,PI)	ORB15150
★ THIS SUBROUTINE CALCULATES THE MOONS POSITION	ORB15160
	ORB15170
★ VARIABLES USED TO DESCRIBE THE SUNS ORBIT:	ORB15180
★ MI = MOONS INCLINATION	ORB15190
★ MLAN= MOONS Longitude OF ASCENDING NODE	ORB15200
★ MAP = MOONS ARGUMENT OF PERIGEE	ORB15210
★ RM = MOONS ORBITAL RADIUS	ORB15220
★ MTA = MOONS TRUE ANOMALY	ORB15230
★ MAL = MOONS ARGUMENT OF LONGITUDE	ORB15240
	ORB15250
DOUBLE PRECISION MLAN,MAL,RM,TM,MTA,RMP,RMQ,RMW,	ORB15260
+ RMI,RMJ,RMK,PI	ORB15270
	ORB15280
MI = 4.99164166D-01	ORB15290
RM = 3.844D+05	ORB15300
MLAN = 0.0	ORB15310
MTA = ((2.0*PI)/(27.3 * 3600) * TT)	ORB15320
MAP = 0.0D+00	ORB15330
MAL = MTA	ORB15340
	ORB15350
★ CALCULATE MOON POSITION IN 'PQW' SYSTEM	ORB15360
RMP = RM*DCOS(MTA)	ORB15370
RMQ = RM*DSIN(MTA)	ORB15380
RMW = 0	ORB15390
	ORB15400
★ TRANSFORM POSITION TO 'IJK' SYSTEM	ORB15410
CALL PQWIJK(MLAN,MAP,MI,RMP,RMQ,RMW,RMI,RMJ,RMK)	ORB15420
RETURN	ORB15430
END	ORB15440
*** *****	ORB15450
	ORB15460
	ORB15470
SUBROUTINE FDRAG(RI,RJ,RK,R,VI,VJ,VK,V,LAN,AP,I,FDR,FDS,FDW,	ORB15480
+ EI,EJ,EK,E,A,T,LP,TA,PER,EA,MA,AL,TF,P,PI,MU,	ORB15490
+ MM,N,H,HI,HJ,DT)	ORB15500
★ THIS SUBROUTINE CALCULATES THE PERTURBING FORCE DUE TO DRAG	ORB15510
	ORB15520
★ THE FOLLOWING VARIABLES ARE USED TO MODEL THE ATMOSPHERE:	ORB15530
★ RE = RADIUS OF EARTH	ORB15540
★ M = MASS OF SATELLITE	ORB15550
★ AR = FRONTAL SURFACE AREA OF SATELLITE	ORB15560
★ Z = ALTITUDE OF SATELLITE	ORB15570
★ K = EXPONENTIAL DECAY FACTOR	ORB15580
★ DENO = NORMAL DENSITY	ORB15590
★ CD = COEFFICIENT OF DRAG	ORB15600
	ORB15610

DOUBLE PRECISION	MAG,M,K,FDR,FDS,FDW,RE,AR,Z,DENO,CD,DEN,	ORB15620
+	FDJ,FDK,FDI,RI,RJ,RK,VI,VJ,VK,V,LAN,AP,I,R,	ORB15630
+	EI,EJ,EN,E,A,T,LP,TA,PER,EA,MA,AL,TF,P,PI,MU,	ORB15640
+	MM,N,H,HI,HJ,DT,DVR,DVS,DVW,DVI,DVJ,DVK	ORB15650
		ORB15660
		ORB15670
RE = 6.378145D+03		ORB15680
M = 1.0D+02		ORB15690
AR = 2.0D+01		ORB15700
Z = R - RE		ORB15710
		ORB15720
* DEPENDING ON ALTITUDE SET ATMOSPHERE VARIABLES		ORB15730
IF (Z.LE. 1.5D+02) THEN		ORB15740
K = 4.74D-02		ORB15750
DENO = 1.225D+00		ORB15760
CD = 1.0D+00		ORB15770
ELSEIF (Z.LE. 5.5D+02) THEN		ORB15780
K = 3.4614D-02		ORB15790
DENO = 1.79846D-01		ORB15800
CD = 2.0D+00		ORB15810
ELSE		ORB15820
K = 2.21698D-3		ORB15830
DENO = 1.015484D-07		ORB15840
CD = 2.0D+00		ORB15850
ENDIF		ORB15860
		ORB15870
* CALCULATE ATMOSPHERIC DENSITY		ORB15880
DEN = DENO * DEXP(-K*Z)		ORB15890
		ORB15900
* CALCULATE MAGNITUDE OF DRAG FORCE AND LIMIT IT TO 1.0E-20		ORB15910
MAG = -(0.5D+00)*CD*AR*DEN*V*(1.0D-03)/M		ORB15920
IF (ABS(MAG) .LT. 1.0D-20) THEN		ORB15930
MAG = -1.0D-20		ORB15940
ENDIF		ORB15950
		ORB15960
* GIVE DRAG FORCE A Direction OF MINUS THE VELOCITY		ORB15970
FDR = 0.0		ORB15980
FDS = MAG * V		ORB15990
FDW = 0.0		ORB16000
RETURN		ORB16010
END		ORB16020
		ORB16030
*****		ORB16040
* CALCULATE PERTURBED NEW ELEMENTS		ORB16050
*****		ORB16060
		ORB16070
SUBROUTINE PNEWEL(FR,FS,FW,H,R,A,E,N,TA,DT,I,LAN,AL,AP,P,		ORB16080
+ MM,MA,EA,TF,T,MU,PI,DI,DA,DE,DMN,DMA,DLAN,DH,DAP)		ORB16090
* THIS SUBROUTINE CALCULATES THE NEW ELEMENTS FROM THE PREVIOUS		ORB16100
* ELEMENTS ADDED TO THE RATES OF CHANGE FOR ONE STEP		ORB16110
		ORB16120
* THE FOLLOWING SUBROUTINES ARE CALLED:		ORB16130
* RATE = CALCULATES RATES OF CHANGE OF ORBITAL ELEMENTS		ORB16140
* NANGMO = NEW ANGULAR MOMENTUM (NEWH)		ORB16150
* NSMA = NEW SEMI-MAJOR AXIS (NEWA)		ORB16160
* NECC = NEW ECCENTRICITY (NEWE)		ORB16170

☆	NINCL = NEW INCLINATION (NEWI)	ORB16180
☆	NASNOD = NEW LONGITUDE OF ASCENDING NODE (NEWLAN)	ORB16190
☆	NARPER = NEW ARGUMENT OF PERIGEE (NEWAP)	ORB16200
☆	NINMO = NEW MEAN MOTION (NEWMM)	ORB16210
☆	MEANMO = MEAN MOTION (MM)	ORB16220
☆	MMAN = NEW MEAN ANOMALY (NEWMA)	ORB16230
☆	NEA = NEW ECCENTRIC ANOMALY (EA)	ORB16240
☆	NTA = NEW TRUE ANOMALY (TA)	ORB16250
☆	TFLGHT = TIME OF FLIGHT (TF)	ORB16260
		ORB16270
	DOUBLE PRECISION FR,FS,FW,DMM,H,R,A,E,N,TA,DT,I,LAN,AL,AP,P,	ORB16280
+	MM,MA,EA,TF,T,MU,PI,DA,DH,DE,DI,DLAN,DAP,DMA,	ORB16290
+	NEWH,NEWA,NEWE,NEWI,NEWLAN,NEWAP,NEWMM	ORB16300
☆	INCREMENT TIME OF FLIGHT BY ONE TIME STEP AND CALCULATE RATES	ORB16310
	TF = TF + DT	ORB16320
	CALL RATES(DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW,	ORB16330
+	TA,AL,H,P,T,MU,I)	ORB16340
		ORB16350
☆	CALCULATE NEW ELEMENTS	ORB16360
	CALL NANGMO(H,DT,DH,NEWH)	ORB16370
	CALL NSMA(A,DT,DA,NEWA)	ORB16380
	CALL NECC(E,DT,DE,NEWE)	ORB16390
	CALL NINCL(I,DT,DI,NEWI)	ORB16400
	CALL NASNOD(LAN,DT,DLAN,NEWLAN)	ORB16410
	CALL NARPER(AP,DT,DAP,NEWAP)	ORB16420
		ORB16430
☆	SET ELEMENTS TO NEW ELEMENTS	ORB16440
	A = NEWA	ORB16450
	E = NEWE	ORB16460
	I = NEWI	ORB16470
	LAN = NEWLAN	ORB16480
	AP = NEWAP	ORB16490
	P = A * (1 - E**2)	ORB16500
		ORB16510
☆	MOVE THE SATELLITE ONE TIME STEP	ORB16520
	CALL MEANMO(A,MM,MU)	ORB16530
	CALL MMAN(MA,MM,DT,TF,DMA,PI)	ORB16540
	CALL NEA(MA,E,EA)	ORB16550
	CALL NTA(EA,E,TA,PI)	ORB16560
	CALL TFLGHT(MM,MA,TF)	ORB16570
	AL = TA + AP	ORB16580
	RETURN	ORB16590
	END	ORB16600
		ORB16610
		ORB16620
	*****	ORB16630
☆	CALCULATE THE RATES OF CHANGE OF THE ORBITAL Elements	ORB16640
	*****	ORB16650
		ORB16660
	SUBROUTINE RATES(DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW,	ORB16670
+	TA,AL,H,P,T,MU,I)	ORB16680
☆	THIS SUBROUTINE CALLS THE FOLLOWING SUBROUTINES TO CALCULATE THE	ORB16690
☆	TIME RATE-OF- CHANGE OF THE ORBITAL ELEMENTS:	ORB16700
☆	RSMAX = RATE-OF-CHANGE OF THE SEMI-MAJOR AXIS (DA)	ORB16710
☆	RECC = RATE-OF-CHANGE OF THE ECCENTRICITY (DE)	ORB16720
☆	RINC = RATE-OF-CHANGE OF THE INCLINATION (DI)	ORB16730

★	RLAN	= RATE-OF-CHANGE OF THE Longitude OF THE ASCENDING NODE	ORB16740
★		(DLAN)	ORB16750
★	RAP	= RATE-OF-CHANGE OF THE ARGUMENT OF PERIGEE (DAP)	ORB16760
★	RMM	= RATE-OF-CHANGE OF THE MEAN MOTION (DMM)	ORB16770
★	RMA	= RATE-OF-CHANGE OF THE MEAN ANOMALY (DMA)	ORB16780
★	RANGMO	= RATE-OF-CHANGE OF THE ANGULAR MOMENTUM (DH)	ORB16790
			ORB16800
	DOUBLE PRECISION	DH,DA,DE,DI,DLAN,DAP,DMM,DMA,E,MM,R,A,FR,FS,FW,	ORB16810
	+	TA,AL,H,P,T,MU,I	ORB16820
	CALL	RSMAX(E,MM,R,A,FR,FS,DA,TA)	ORB16830
	CALL	RECC(E,MM,R,A,FR,FS,TA,DE)	ORB16840
	CALL	RINC(E,MM,R,A,FW,AL,DI)	ORB16850
	CALL	RLAN(E,MM,R,A,I,FW,AL,DLAN)	ORB16860
	CALL	RAP(E,MM,R,A,I,H,P,AL,TA,FR,FS,FW,DAP)	ORB16870
	CALL	RMM(MM,A,DMM,DA,MU)	ORB16880
	CALL	RMA(E,MM,R,A,TA,DMM,FR,FS,DMA,T)	ORB16890
	CALL	RANGMO(R,FS,FW,DH)	ORB16900
	RETURN		ORB16910
	END		ORB16920
			ORB16930
			ORB16940
	*****		ORB16950
			ORB16960
	SUBROUTINE	RANGMO(R,FS,FW,DH)	ORB16970
★	THIS	SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE	ORB16980
★	ANGULAR	MOMENTUM	ORB16990
	DOUBLE PRECISION	FS,FW,DHW,DHS,DH,R	ORB17000
			ORB17010
	DHW	= R * FS	ORB17020
	DHS	= R * FW	ORB17030
	DH	= DSQRT((DHW**2) + (DHS**2))	ORB17040
	RETURN		ORB17050
	END		ORB17060
			ORB17070
			ORB17080
	*****		ORB17090
			ORB17100
	SUBROUTINE	RSMAX(E,MM,R,A,FR,FS,DA,TA)	ORB17110
★	THIS	SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE SEMI-MAJOR	ORB17120
★	AXIS		ORB17130
	DOUBLE PRECISION	DA,FR,FS,E,MM,R,A,TA,ET	ORB17140
			ORB17150
★	TRAP (E)	SO DENOMINATOR DOES NOT GOTO ZERO	ORB17160
	IF (E.GT.0.9)	THEN	ORB17170
	ET	= 0.9	ORB17180
	ELSE		ORB17190
	ET	= E	ORB17200
	ENDIF		ORB17210
	DA	= ((2.0D+00*E *DSIN(TA))/(MM *DSQRT(1.0D+00-(ET**2))))*FR +	ORB17220
	+	((2.0D+00*A*DSQRT(1.0D+00-(E **2)))/(MM *R))*FS	ORB17230
	RETURN		ORB17240
	END		ORB17250
			ORB17260
			ORB17270
	*****		ORB17280
			ORB17290

☆	SUBROUTINE RECC(E,MM,R,A,FR,FS,TA,DE)	ORB17300
	THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE ECCENTRICITY	ORB17310
	DOUBLE PRECISION DE,FR,FS,E,MM,R,A,TA,ET	ORB17320
☆	TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO	ORB17330
	IF (E.LT.0.1) THEN	ORB17340
	ET = 0.1	ORB17350
	ELSE	ORB17360
	ET = E	ORB17370
	ENDIF	ORB17380
	DE = ((DSQRT(1.0D+00 - (E **2))*SIN(TA))/(MM *A))*FR +	ORB17390
	+ ((DSQRT(1.0D+00 - (E **2)))/(MM *ET*(A**2)))*	ORB17400
	+ ((A**2)*(1.0D+00 - (E **2))/(R) - (R))*FS	ORB17410
	RETURN	ORB17420
	END	ORB17430
	*****	ORB17440
		ORB17450
		ORB17460
		ORB17470
	SUBROUTINE RLAN(E,MM,R,A,I,FW,AL,DLAN)	ORB17480
☆	THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE LONGITUDE	ORB17490
☆	OF THE ASCENDING NODE	ORB17500
	DOUBLE PRECISION DLAN,FW,E,MM,R,A,I,AL,ET,IT	ORB17510
☆	TRAP (E) AND (I) SO DENOMINATOR DOES NOT GOTO ZERO	ORB17520
	IF (E.GT.0.9) THEN	ORB17530
	ET = 0.9	ORB17540
	ELSE	ORB17550
	ET = E	ORB17560
	ENDIF	ORB17570
	IF (I.LT.0.01745) THEN	ORB17580
	IT = 0.01745	ORB17590
	ELSE	ORB17600
	IT = I	ORB17610
	ENDIF	ORB17620
	DLAN = (R*FW*DSIN(AL))/(MM *(A**2)*DSQRT(1.0D+00 - (ET**2))*	ORB17630
	+ DSIN(IT))	ORB17640
	RETURN	ORB17650
	END	ORB17660
	*****	ORB17670
		ORB17680
		ORB17690
		ORB17700
		ORB17710
		ORB17720
	SUBROUTINE RAP(E,MM,R,A,I,H,P,AL,TA,FR,FS,FW,DAP)	ORB17730
☆	THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE ARGUMENT	ORB17740
☆	OF PERIGEE	ORB17750
	DOUBLE PRECISION DAPR,DAPS,DAPW,DAP,FR,FS,FW,E,MM,R,I,H,P,AL,TA,	ORB17760
	+ ET,A,IT	ORB17770
☆	TRAP (E) AND (I) SO DENOMINATOR DOES NOT GOTO ZERO	ORB17780
	IF (I.LT.0.01745) THEN	ORB17790
	IT = 0.01745	ORB17800
	ELSE	ORB17810
	IT = I	ORB17820
	ENDIF	ORB17830
		ORB17840
		ORB17850

IF (E.GT.0.9) THEN	ORB17860
ET = 0.9	ORB17870
ELSEIF (E.LT.0.1) THEN	ORB17880
ET = 0.1	ORB17890
ELSE	ORB17900
ET = E	ORB17910
ENDIF	ORB17920
DAPR = (-DSQRT(1.0+00 - (E **2))*DCOS(TA))/(MM *A*ET) * FR	ORB17930
DAPS = (P/(ET*M))*(DSIN(TA))*	ORB17940
+ (1.0D+00 + 1.0D+00/(1.0D+00 + ET*DCOS(TA))) *FS	ORB17950
DAPW = (-R*(1.0D+00/DTAN(IT))*DSIN(AL))/	ORB17960
+ (MM *(A**2)*DSQRT(1.0D+00 - (ET**2)))*FW	ORB17970
DAP = DAPR + DAPS + DAPW	ORB17980
RETURN	ORB17990
END	ORB18000
*****	ORB18010
SUBROUTINE RINC(E,MM,R,A,FW,AL,DI)	ORB18020
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE INCLINATION	ORB18030
DOUBLE PRECISION DI,FW,E,MM,R,A,AL,ET	ORB18040
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO	ORB18050
IF (E.GT.0.9) THEN	ORB18060
ET = 0.9	ORB18070
ELSE	ORB18080
ET = E	ORB18090
ENDIF	ORB18100
DI = (R*FW*DCOS(AL))/(MM *(A**2)*DSQRT(1.0D+00 - (ET**2)))	ORB18110
RETURN	ORB18120
END	ORB18130
*****	ORB18140
SUBROUTINE RMM(MM,A,DMM,DA,MU)	ORB18150
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE MEAN MOTION	ORB18160
DOUBLE PRECISION DMM,DA,MM,A,MU	ORB18170
DMM =((-3.0D+00*MU)/(2.0D+00*MM *(A**4)))* DA	ORB18180
RETURN	ORB18190
END	ORB18200
*****	ORB18210
SUBROUTINE RMA(E,MM,R,A,TA,DMM,FR,FS,DMA,T)	ORB18220
* THIS SUBROUTINE CALCULATES THE RATE OF CHANGE OF THE MEAN Anomaly	ORB18230
DOUBLE PRECISION DMAA,DMAB,DMAc,DMAc,DMM,FR,FS,DMA,E,MM,R,A,TA,	ORB18240
+ ET,T	ORB18250
* TRAP (E) SO DENOMINATOR DOES NOT GOTO ZERO	ORB18260
IF (E.GT.0.9) THEN	ORB18270
ET = 0.9	ORB18280
ELSEIF (E.LT.0.1) THEN	ORB18290
	ORB18300
	ORB18310
	ORB18320
	ORB18330
	ORB18340
	ORB18350
	ORB18360
	ORB18370
	ORB18380
	ORB18390
	ORB18400
	ORB18410

ET = 0.1	ORB18420
ELSE	ORB18430
ET = E	ORB18440
ENDIF	ORB18450
DMA = (-1.0D+00/(MM *A))*	ORB18460
+ (((2.0D+00*R)/A) - ((1 - (E **2))/ET)*DCOS(TA)) * FR -	ORB18470
+ (1-(E **2))/(MM *A*ET)*(1+ R/(A*(1-(E**2))))*(SIN(TA)*FS)-	ORB18480
+ (T * DMM)	ORB18490
RETURN	ORB18500
END	ORB18510
*****	ORB18520
* CALCULATE THE NEW ORBITAL ELEMENTS	ORB18530
*****	ORB18540
SUBROUTINE NSMA(A,DT,DA,NEWA)	ORB18550
* THIS SUBROUTINE CALCULATES THE NEW SEMI-MAJOR AXIS	ORB18560
DOUBLE PRECISION DA,DT,A,NEWA	ORB18570
NEWA = A + DA*DT	ORB18580
RETURN	ORB18590
END	ORB18600
*****	ORB18610
SUBROUTINE NECC(E,DT,DE,NEWE)	ORB18620
* THIS SUBROUTINE CALCULATES THE NEW ECCENTRICITY	ORB18630
DOUBLE PRECISION DE,DT,E,NEWE	ORB18640
NEWE = E + DE*DT	ORB18650
RETURN	ORB18660
END	ORB18670
*****	ORB18680
SUBROUTINE NINCL(I,DT,DI,NEWI)	ORB18690
* THIS SUBROUTINE CALCULATES THE NEW INCLINATION	ORB18700
DOUBLE PRECISION DI,DT,I,NEWI	ORB18710
NEWI = I + DI*DT	ORB18720
RETURN	ORB18730
END	ORB18740
*****	ORB18750
SUBROUTINE NASNOD(LAN,DT,DLAN,NEWLAN)	ORB18760
* THIS SUBROUTINE CALCULATES THE NEW LONGITUDE OF THE ASCENDING NODE	ORB18770
DOUBLE PRECISION DLAN,DT,LAN,NEWLAN	ORB18780
NEWLAN = LAN + DLAN*DT	ORB18790
RETURN	ORB18800
END	ORB18810
*****	ORB18820
SUBROUTINE NASNOD(LAN,DT,DLAN,NEWLAN)	ORB18830
* THIS SUBROUTINE CALCULATES THE NEW LONGITUDE OF THE ASCENDING NODE	ORB18840
DOUBLE PRECISION DLAN,DT,LAN,NEWLAN	ORB18850
NEWLAN = LAN + DLAN*DT	ORB18860
RETURN	ORB18870
END	ORB18880
*****	ORB18890
SUBROUTINE NASNOD(LAN,DT,DLAN,NEWLAN)	ORB18900
* THIS SUBROUTINE CALCULATES THE NEW LONGITUDE OF THE ASCENDING NODE	ORB18910
DOUBLE PRECISION DLAN,DT,LAN,NEWLAN	ORB18920
NEWLAN = LAN + DLAN*DT	ORB18930
RETURN	ORB18940
END	ORB18950
	ORB18960
	ORB18970

★ SUBROUTINE NARPER(AP,DT,DAP,NEWAP)
THIS SUBROUTINE CALCULATES THE NEW ARGUMENT OF PERIGEE

DOUBLE PRECISION DAP,DT,AP,NEWAP

NEWAP = AP + DAP*DT

RETURN

END

★ SUBROUTINE NNNAN(MA,MM,DT,TF,DMA,PI)
THIS SUBROUTINE CALCULATES THE NEW MEAN Anomaly

DOUBLE PRECISION DMM,FR,FS,DMA,DT,MA,E,R,A,TA,MM,TF,T,PI

MA = MM*(TF) + DMA*DT

IF (MA .GT. (2*PI)) THEN

MA = MA - (2*PI)

ENDIF

RETURN

END

★ SUBROUTINE NNNMO(MM,DMM,DT,NEWMM)
THIS SUBROUTINE CALCULATE THE NEW MEAN MOTION

DOUBLE PRECISION DMM,DT,MM,NEWMM

NEWMM = MM + DMM*DT

RETURN

END

★ SUBROUTINE NEA(MA,E,EA)
THIS SUBROUTINE CALCULATES THE NEW ECCENTRIC ANOMOLY BY USING
★ NEWTONS METHOD OF ROOT FINDING

DOUBLE PRECISION EAN,MAN,MA,E,EA,DIFF

★ LET (EA) EQUAL (MA) FOR INITIAL GUESS AT ROOT

EA = MA

EAN = EA + (MA - EA + E*DSIN(EA))/(1.0D+00 - E*DCOS(EA))

MAN = EAN - E*SIN(EAN)

★ CHECK DIFFERENCE (DIFF)

DIFF = ABS(MA -MAN)

EA = EAN

★ CONTINUE TO INTERATE UNTIL DIFFERENCE IS NEGLIGIBLE

200 IF(DIFF.GT.0.0000000001) THEN

ORB18980
ORB18990
ORB19000
ORB19010
ORB19020
ORB19030
ORB19040
ORB19050
ORB19060
ORB19070
ORB19080
ORB19090
ORB19100
ORB19110
ORB19120
ORB19130
ORB19140
ORB19150
ORB19160
ORB19170
ORB19180
ORB19190
ORB19200
ORB19210
ORB19220
ORB19230
ORB19240
ORB19250
ORB19260
ORB19270
ORB19280
ORB19290
ORB19300
ORB19310
ORB19320
ORB19330
ORB19340
ORB19350
ORB19360
ORB19370
ORB19380
ORB19390
ORB19400
ORB19410
ORB19420
ORB19430
ORB19440
ORB19450
ORB19460
ORB19470
ORB19480
ORB19490
ORB19500
ORB19510
ORB19520
ORB19530

EAN = EA + (MA - EA + E*DSIN(EA))/(1.0D+00 - E*DCOS(EA))	ORB19540
MAN = EAN - E*DSIN(EAN)	ORB19550
EA = EAN	ORB19560
DIFF = ABS(MA - MAN)	ORB19570
GOTO 200	ORB19580
ENDIF	ORB19590
EA = EAN	ORB19600
RETURN	ORB19610
END	ORB19620

SUBROUTINE NTA(EA,E,TA,PI)	ORB19630
* THIS SUBROUTINE CALCULATES THE NEW TRUE Anomaly	ORB19640
DOUBLE PRECISION EA,E,TA,PI	ORB19650
TA = DACOS((E - DCOS(EA))/(E*DCOS(EA) - 1.0D+00))	ORB19660
IF (EA.GT.PI) THEN	ORB19670
TA = (2*PI) - TA	ORB19680
ENDIF	ORB19690
RETURN	ORB19700
END	ORB19710

SUBROUTINE NANGMO(H,DT,DH,NEWH)	ORB19720
* THIS SUBROUTINE CALCULATES THE NEW ANGULAR MOMENTUM	ORB19730
DOUBLE PRECISION DH,DT,H,NEWH	ORB19740
NEWH = H + DH*DT	ORB19750
RETURN	ORB19760
END	ORB19770

SUBROUTINE INTSUM(TFEA,TFSU,TFMO,TFDRA,TDI,TDA,TDE,TDMM,TDMA,	ORB19780
TDLAN,TDH,TDAP)	ORB19790
* THIS SUBROUTINE INITIALIZES THE SUMS OF FORCES AND ELEMENT CHANGES	ORB19800
TFEA = 0.0	ORB19810
TFSU = 0.0	ORB19820
TFMO = 0.0	ORB19830
TFDRA = 0.0	ORB19840
TDI = 0.0	ORB19850
TDA = 0.0	ORB19860
TDE = 0.0	ORB19870
TDMM = 0.0	ORB19880
TDMA = 0.0	ORB19890
TDLAN = 0.0	ORB19900
TDH = 0.0	ORB19910
TDAP = 0.0	ORB19920
RETURN	ORB19930
END	ORB19940
	ORB19950
	ORB19960
	ORB19970
	ORB19980
	ORB19990
	ORB20000
	ORB20010
	ORB20020
	ORB20030
	ORB20040
	ORB20050
	ORB20060
	ORB20070
	ORB20080

*****	ORB20090
*****	ORB20100
* CALCULATE THE NEW POSITION AND VELOCITY VECTORS	ORB20110
*****	ORB20120
	ORB20130
SUBROUTINE NPOS(RI,RJ,RK,R,LAN,AP,INC, TA,A,E)	ORB20140
* THIS SUBROUTINE CALCULATES THE NEW POSITION VECTOR	ORB20150
	ORB20160
DOUBLE PRECISION XW,YW,ZW,INC,RI,RJ,RK,R,LAN,AP,TA,A,E	ORB20170
	ORB20180
* CALCULATE POSITION VECTOR IN 'PQW' SYSTEM	ORB20190
R = (A*(1 - (E**2)))/(1 + E*DCOS(TA))	ORB20200
XW = R*DCOS(TA)	ORB20210
YW = R*DSIN(TA)	ORB20220
ZW = 0	ORB20230
	ORB20240
* TRANSFORM POSITION TO 'IJK' SYSTEM	ORB20250
CALL PQW1JK(LAN,AP,INC,XW,YW,ZW,RI,RJ,RK)	ORB20260
R = DSQRT((RI**2) + (RJ**2) + (RK**2))	ORB20270
RETURN	ORB20280
END	ORB20290
*****	ORB20300
	ORB20310
SUBROUTINE NVEL(E,P,TA,LAN,AP,INC,VI,VJ,VK,V,MU)	ORB20320
* THIS SUBROUTINE CALCULATES THE NEW VELOCITY VECTOR	ORB20330
	ORB20340
DOUBLE PRECISION INC,VP,VQ,VW,MU,E,P,TA,LAN,AP,VI,VJ,VK,V	ORB20350
	ORB20360
* CALCULATE VELOCITY IN 'PQW' SYSTEM	ORB20370
VP = DSQRT(MU/P)*(-DSIN(TA))	ORB20380
VQ = DSQRT(MU/P)*(E + DCOS(TA))	ORB20390
VW = 0.0D+00	ORB20400
	ORB20410
	ORB20420
* TRANSFORM VELOCITY INTO 'IJK' SYSTEM	ORB20430
CALL PQW1JK(LAN,AP,INC,VP,VQ,VW,VI,VJ,VK)	ORB20440
V = DSQRT((VI**2) + (VJ**2) + (VK**2))	ORB20450
RETURN	ORB20460
END	ORB20470
	ORB20480
*****	ORB20490
* VELOCITY CHANGE	ORB20500
*****	ORB20510
	ORB20520
SUBROUTINE CHGVEL(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,	ORB20530
+ VI,VJ,VK,V,MU,PI,H,A,E,N,TA,P,MM,MA,EA,	ORB20540
+ TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,TARAY,AINRAY,APRAY,TIMRAY,	ORB20550
+ TT,EI,EJ,EK,LP,HI,HJ,IOPT1,TFEA,TFSU,TFMO,TFDRA,	ORB20560
+ TDI,TDA,TDE,TDMM,TDMA,TDLAN,TDH,TDAP)	ORB20570
* THIS SUBROUTINE CALCULATE VELOCITY CHANGES	ORB20580
	ORB20590
* THE FOLLOWING SUBROUTINES ARE CALLED:	ORB20600
* TACHG = RETURNS TRUE ANOMALY FOR VELOCITY CHANGE LOCATION (CHTA)	ORB20610
* AND AN INDICATOR OF LOCATION (ITA)	ORB20620
* CALCEL = CALCULATE Orbital ELEMENTS	ORB20630
* UNPREZ = CALCULATE UNPERTURBED ORBIT	ORB20640

```

* NPOS = CALCULATE NEW POSITION JRB20650
* NVEL = CALCULATE NEW VELOCITY ORB20660
* STORE = STORE POSITION AND ELEMENTS IN ARRAYS ORB20670
* ENERGY = ENERGY OF SATELLITE ORB20680
* ECC = ECCENTRICITY ORB20690
* SMAXIS = SEMI-MAJOR AXIS ORB20700
ORB20710
DOUBLE PRECISION T,DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V, ORB20720
+ MU,PI,H,A,E,N,TA,P,MM,MA,EA,TF,TT, ORB20730
+ NEWVI,NEWVJ,NEWVK,NEWV,VMAX,CHTA,EI,EJ,EK,LP,HI,HJ,VCIR, ORB20740
+ DI,DE,DA,DM,DMA,DLAN,DH,DAP,NEWEI,NEWJ,NEWK,NEWV,NEWNR, ORB20750
+ NEWA,NEWRP,RE ORB20760
ORB20770
DIMENSION RARRAY(500),TARRAY(500),RIRAY(500),RJRAY(500), ORB20780
+ RKRAY(500),AINRAY(500),APRAY(500),TIMRAY(500) ORB20790
ORB20800
CHARACTER*1,YORN,PYORN ORB20810
ORB20820
RE = 6.3782D+03 ORB20830
ORB20840
* PROMPT THE USER FOR THE VELOCITY Change LOCATION ORB20850
CALL TACNG(PI,CHTA,ITA) ORB20860
ORB20870
* SET TIME COUNTER TO ONE TIME STEP ORB20880
T = DT ORB20890
ORB20900
* ROTATE TO THE VELOCITY CHANGE LOCATION ORB20910
* THIS IS IDENTICAL TO THE Unperturbed ORBIT WITH THE EXCEPTION ORB20920
* THAT A COMPLETE ORBIT IS NOT CALCULATED ORB20930
PRINT*, 'ROTATE TO VELOCITY CHANGE LOCATION' ORB20940
IF ((ITA.EQ.2) .OR. (ITA.EQ.3)) THEN ORB20950
PRINT*, 'BEFORE TA =',TA ORB20960
IF (TA .GT. 6.21) THEN ORB20970
TA = TA - (2*PI) ORB20980
ENDIF ORB20990
250 IF((T.LE.PER).AND.(TA.LT.CHTA)) THEN ORB21000
* PRINT*, 'TA =',TA ORB21010
NUM = NUM + 1 ORB21020
TT = TT + DT ORB21030
CALL NEWELT(MM,MA,E,EA,TA,TF,DT,PI,PER) ORB21040
CALL NPOS(RI,RJ,RK,R,LAN,AP,I,TA,A,E) ORB21050
CALL NVEL(E,P,TA,LAN,AP,I,VI,VJ,VK,V,MU) ORB21060
CALL STORE(RI,RJ,RK,R,TA,RIRAY,RJRAY,RKRAY,RARRAY, ORB21070
+ TARRAY,NUM,I,AP,AINRAY,APRAY,TT,TIMRAY) ORB21080
ORB21090
T = T + DT ORB21100
GOTO 250 ORB21110
ENDIF ORB21110
IF (TF .GE. PER) THEN ORB21120
TF = TF - PER ORB21130
ENDIF ORB21140
ENDIF ORB21150
ORB21160
* PRINT ESCAPE VELOCITY AND CIRCULAR VELOCITY FOR Reference ORB21170
CALL EXCNS('CLRSCRN') ORB21180
PRINT*, 'AFTER TA =',TA ORB21190
PRINT*, 'THIS SHOULD BE THE DESIRED RADIUS RP OR RA' ORB21200

```

260	PRINT*, 'RADIUS =', R	ORB21210
	PRINT*, 'VELOCITY =', V	ORB21220
	VMAX = DSQRT(2.0*(MU / R))	ORB21230
	PRINT*, 'MAX VELOCITY AT THIS RADIUS IS:', VMAX	ORB21240
	VCIR = DSQRT(MU/R)	ORB21250
	PRINT*, 'CIRCULAR VELOCITY AT THIS RADIUS IS:', VCIR	ORB21260
*	PROMPT USER TO CHANGE VELOCITY IN ORBITAL PLANE	ORB21270
	PRINT*, 'DO YOU WANT TO CHANGE THE VELOCITY IN THE ORBITAL PLANE?'	ORB21280
	PRINT*, 'ENTER "Y" OR "N" :'	ORB21290
	READ*, PYORN	ORB21300
	PRINT*, PYORN	ORB21310
	IF (PYORN .EQ. 'Y') THEN	ORB21320
	PRINT*, 'GIVE THE TOTAL CHANGE IN VELOCITY, I.E. 5.0 KM.'	ORB21330
	PRINT*, 'THE PROGRAM WILL FIGURE OUT THE FINAL VELOCITY VECTOR'	ORB21340
	PRINT*, 'ENTER VELOCITY CHANGE: '	ORB21350
	READ*, CHGV	ORB21360
	PRINT*, CHGV	ORB21370
		ORB21380
		ORB21390
*	CALCULATE NEW VELOCITY FOR CHANGE IN THE ORBITAL PLANE	ORB21400
	NEWVI = VI + (CHGV * VI / V)	ORB21410
	NEWVJ = VJ + (CHGV * VJ / V)	ORB21420
	NEWVK = VK + (CHGV * VK / V)	ORB21430
		ORB21440
*	Velocity CHANGE OUT OF ORBITAL PLANE	ORB21450
	ELSEIF (PYORN .EQ. 'N') THEN	ORB21460
	PRINT*, 'ENTER THE NEW VELOCITY VECTOR: '	ORB21470
	PRINT*, 'ENTER THE NEW VI'	ORB21480
	READ*, NEWVI	ORB21490
	PRINT*, NEWVI	ORB21500
	PRINT*, 'ENTER THE NEW VJ'	ORB21510
	READ*, NEWVJ	ORB21520
	PRINT*, NEWVJ	ORB21530
	PRINT*, 'ENTER THE NEW VK'	ORB21540
	READ*, NEWVK	ORB21550
	PRINT*, NEWVK	ORB21560
	NUM = 1	ORB21570
	ITA = 3	ORB21580
	ELSE	ORB21590
	CALL EXCMS('GLRSCRN')	ORB21600
	GOTO 260	ORB21610
	ENDIF	ORB21620
		ORB21630
*	PRINT NEW VELOCITY FOR USER TO CHECK	ORB21640
	NEWV = DSQRT((NEWVI**2) + (NEWVJ**2) + (NEWVK**2))	ORB21650
	PRINT*, 'NEW VI =', NEWVI	ORB21660
	PRINT*, 'NEW VJ =', NEWVJ	ORB21670
	PRINT*, 'NEW VK =', NEWVK	ORB21680
	PRINT*, 'NEW V =', NEWV	ORB21690
	PRINT*, 'ARE THESE VALUES THE ONES YOU WANT?'	ORB21700
	PRINT*, 'ENTER "Y" OR "N" :'	ORB21710
	READ*, YORN	ORB21720
	PRINT*, YORN	ORB21730
	IF (YORN .EQ. 'N') THEN	ORB21740
	CALL EXCMS('CLRSCRN')	ORB21750
	GOTO 260	ORB21760

ENDIF	ORB21770
* CHECK FOR VALID VELOCITY	ORB21780
IF (NEWV .GT. VMAX) THEN	ORB21790
PRINT*, 'YOUR VELOCITY IS TO GREAT !!'	ORB21800
GOTO 260	ORB21810
ENDIF	ORB21820
	ORB21830
	ORB21840
* Calculate PERIGEE RADIUS TO SEE IF SATELLITE WILL IMPACT EARTH	ORB21850
CALL ENERGY(NEWV,R,MU,NEWENR)	ORB21860
CALL ECC(RI,RJ,RK,R,NEWVI,NEWVJ,NEWVK,NEWV,NEWEI,NEWVJ,NEWVK,	ORB21870
+ NEWV,MU)	ORB21880
CALL SMAXIS(MU,NEWENR,NEWA)	ORB21890
NEWRP = NEWA*(1.0 - NEWV)	ORB21900
IF (NEWRP .LE. RE) THEN	ORB21910
PRINT*, 'YOUR VELOCITY AT THIS POINT IS TO SMALL!!!'	ORB21920
PRINT*, 'THE SATELLITE WILL IMPACT THE EARTH!!!'	ORB21930
PRINT*, 'THE SATELLITES RADIUS OF PERIGEE WOULD BE ',NEWRP	ORB21940
PRINT*, 'A NEW VELOCITY WILL HAVE TO BE ENTERED!!'	ORB21950
GOTO 260	ORB21960
ENDIF	ORB21970
	ORB21980
* ACCEPT NEW VELOCITY	ORB21990
VI = NEWVI	ORB22000
VJ = NEWVJ	ORB22010
VK = NEWVK	ORB22020
V = NEWV	ORB22030
	ORB22040
* CALCULATE NEW ELEMENT WITH NEW VELOCITY AND SET TIME STEP	ORB22050
CALL CALCEL(RI,RJ,RK,R,VI,VJ,VK,V,EI,EJ,EK,E,A,I,LAN,LP,TA,	ORB22060
+ PER,EA,MA,AP,AL,TF,P,PI,MU,MM,N,H,HI,HJ)	ORB22070
DT = PER/50.0	ORB22080
T = DT	ORB22090
	ORB22100
* THE FOUR Different CASES OF VELOCITY CHANGES FOLLOWS:	ORB22110
	ORB22120
* VELOCITY CHANGE AT PERIGEE, AND NEWV > V Circular	ORB22130
IF((ITA.EQ.1).AND.(NEWV.GT.VCIR))THEN	ORB22140
CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22150
+ MU,PI,H,A,E,N,TA,P,MM,	ORB22160
+ MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22170
+ TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22180
	ORB22190
* Change VELOCITY AT PERIGEE, AND NEWV <= V CIRCULAR	ORB22200
* APOGEE AND PERIGEE SWAP	ORB22210
ELSEIF ((ITA.EQ.1).AND.(NEWV.LE.VCIR))THEN	ORB22220
	ORB22230
* CLEAR PREVIOUS PLOTS	ORB22240
NUM = 1	ORB22250
CALL STORE(RI,RJ,RK,R,TA,RIKAY,RJRAY,RKRAY,RARAY,TARAY,	ORB22260
+ NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB22270
T = PER/2	ORB22280
	ORB22290
* STEP SATELLITE TO NEW PERIGEE, ONLY A HALF ORBIT	ORB22300
CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22310
+ MU,PI,H,A,E,N,TA,P,MM,	ORB22320

+	MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22330
+	TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22340
☆	RESET TIME COUNTER TO ONE TIME STEP	ORB22350
	T = DT	ORB22360
☆	CALCULATE COMPLETE NEXT ORBIT	ORB22370
	CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22380
+	MU,PI,H,A,E,N,TA,P,MM,	ORB22390
+	MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22400
+	TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22410
☆	CHANGE VELOCITY AT APOGEE, AND NEW V < V CIRCULAR	ORB22420
	ELSEIF ((ITA.EQ.2) .AND. (NEWV .LT. VCIR)) THEN	ORB22430
	T = PER/2	ORB22440
☆	FINISH ORBIT	ORB22450
	CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22460
+	MU,PI,H,A,E,N,TA,P,MM,	ORB22470
+	MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22480
+	TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22490
☆	CHANGE VELOCITY AT Apogee, AND NEWV >= V CIRCULAR	ORB22500
☆	OR AT ANY OTHER TRUE Anomaly	ORB22510
	ELSEIF (((ITA.EQ.2).AND.(NEWV.GE.VCIR)) .OR. (ITA.EQ.3)) THEN	ORB22520
	IF (TA .GT. 6.21) THEN	ORB22530
	TA = TA - (2*PI)	ORB22540
	ENDIF	ORB22550
☆	CLEAR PREVIOUS ORBITS AND STEP SATELLITE TO NEW PERIGEE	ORB22560
	T = TF	ORB22570
	NUM = 1	ORB22580
	CALL STORE(RI,RJ,RK,R,TA,RIPAY,RJRAY,RKRAY,RARAY,TARAY,	ORB22590
+	NUM,I,AP,AINRAY,APRAY,TT,TIMRAY)	ORB22600
	CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22610
+	MU,PI,H,A,E,N,TA,P,MM,	ORB22620
+	MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22630
+	TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22640
	IF (TF .GE. PER) THEN	ORB22650
	TF = TF - PER	ORB22660
	ENDIF	ORB22670
☆	CALCULATE COMPLETE NEXT ORBIT	ORB22680
	T = DT	ORB22690
	CALL UNPRET(DT,PER,AL,LAN,AP,I,RI,RJ,RK,R,VI,VJ,VK,V,	ORB22700
+	MU,PI,H,A,E,N,TA,P,MM,	ORB22710
+	MA,EA,TF,T,NUM,RIRAY,RJRAY,RKRAY,RARAY,	ORB22720
+	TARAY,AINRAY,APRAY,TIMRAY,TT)	ORB22730
	ENDIF	ORB22740
	RETURN	ORB22750
	END	ORB22760
*****		ORB22770
		ORB22780
		ORB22790
		ORB22800
		ORB22810
		ORB22820
		ORB22830
		ORB22840
		ORB22850
		ORB22860
		ORB22870
		ORB22880

★	SUBROUTINE TACNG(PI,CHTA,ITA)	ORB22890
	THIS SUBROUTINE Asks THE USER FOR VELOCITY CHANGE LOCATION	ORB22900
	DOUBLE PRECISION CHTA,PI	ORB22910
	CALL EXCHS('CLRSCRN')	ORB22920
	PRINT*, 'WHERE DO YOU WANT TO CHANGE THE VELOCITY?'	ORB22930
	PRINT*, ' 1. AT CURRENT PERIGEE'	ORB22940
	PRINT*, ' 2. AT CURRENT Apogee'	ORB22950
	PRINT*, ' 3. AT A SPECIFIC TRUE Anomaly'	ORB22960
	PRINT*, 'ENTER "1", "2" OR "3"'	ORB22970
	READ*, ITA	ORB22980
	PRINT*, ITA	ORB22990
		ORB23000
		ORB23010
		ORB23020
★	SET TRUE ANOMALY CHANGE LOCATION (CHTA) TO DESIRED LOCATION	ORB23030
	IF (ITA .EQ. 1) THEN	ORB23040
	CHTA = 0.0	ORB23050
	ENDIF	ORB23060
	IF (ITA .EQ. 2) THEN	ORB23070
	CHTA = PI	ORB23080
	ENDIF	ORB23090
	IF (ITA .EQ. 3) THEN	ORB23100
	PRINT*, 'AT WHAT TRUE ANOMALY DO YOU WANT TO CHANGE THE'	ORB23110
	PRINT*, 'VELOCITY?'	ORB23120
	PRINT*, 'ENTER TRUE ANOMALY IN DEGREES'	ORB23130
	READ*, CHTA	ORB23140
	PRINT*, CHTA	ORB23150
	CHTA = CHTA * PI / 180	ORB23160
	ENDIF	ORB23170
	RETURN	ORB23180
	END	ORB23190
		ORB23200
	*****	ORB23210
★	OUTPUT PLOTS	ORB23220
	*****	ORB23230
	SUBROUTINE PLOTS(RIRAY,RJRAY,RKRAY,RARAY,TARAY,NUM,PI,INC,LP,A,	ORB23240
+	E,TF,AINRAY,APRAY,TIMRAY,TFEA,TFSU,TFMO,TFDRA,	ORB23250
+	PER,TDI,TDA,TDE,TDMM,TDMA,TDLAN,TDH,TDAP,	ORB23260
+	MM,MA,LAN,H,AP,R,V)	ORB23270
		ORB23280
		ORB23290
★	THIS SUBROUTINE ASKS THE USER FOR THE TYPE OF OUTPUT THAT IS	ORB23300
★	DESIRED PERIFOCAL, GROUND TRACK OR TO SKIP THE PLOT.	ORB23310
		ORB23320
★	THE FOLLOWING SUBROUTINES ARE CALLED:	ORB23330
★	PERIF = PLOT PERIFOCAL ORBIT	ORB23340
★	GRTRK = PLOT GROUND TRACK	ORB23350
★	DATE = DISPLAYS DATA ON PLOT	ORB23360
★	TEC618 = SET Disspla TO TEC 618 OUTPUT	ORB23370
★	ENDPL = END THIS DISSPLA PLOT	ORB23380
★	REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA	ORB23390
★	SUBROUTINES	ORB23400
		ORB23410
	DOUBLE PRECISION PI,A,E,INC,LP,TF,PER,MM,MA,LAN,H,AP,R,V	ORB23420
		ORB23430
	DIMENSION RIRAY(500),RJRAY(500),RKRAY(500),RARAY(500),TARAY(500),	ORB23440

+	AINRAY(500),APRAY(500),TIMRAY(500)	ORB23450
	CHARACTER*1,YORN	ORB23460
	CALL EXCHS('CLRSCRN')	ORB23470
		ORB23480
		ORB23490
*	CALCULATE SINGLE PRECISION VARIABLES	ORB23500
	SPI = SNGL(PI)	ORB23510
	SA = SNGL(A)	ORB23520
	SE = SNGL(E)	ORB23530
	SINC = SNGL(INC)	ORB23540
	SLP = SNGL(LP)	ORB23550
	STF = SNGL(TF)	ORB23560
	SPER = SNGL(PER)	ORB23570
	SMM = SNGL(MM)	ORB23580
	SMA = SNGL(MA)	ORB23590
	SLAN = SNGL(LAN)	ORB23600
	SH = SNGL(H)	ORB23610
	SAP = SNGL(AP)	ORB23620
	SV = SNGL(V)	ORB23630
	SR = SNGL(R)	ORB23640
		ORB23650
*	PROMPT USER FOR DISPLAY TYPE	ORB23660
340	PRINT*, 'WHAT TYPE OF Display IS DESIRED: '	ORB23670
	PRINT*, ' 1. PERIFOCAL'	ORB23680
	PRINT*, ' 2. GROUND TRACK'	ORB23690
	PRINT*, ' 3. SKIP PLOT'	ORB23700
	PRINT*, 'ENTER 1,2,3,4: '	ORB23710
	READ*, INPUT	ORB23720
	PRINT350, INPUT	ORB23730
350	FORMAT(I4)	ORB23740
		ORB23750
	CALL TEN618	ORB23760
		ORB23770
*	CALL APPROPRIATE PLOT	ORB23780
	IF (INPUT .EQ. 1) THEN	ORB23790
	CALL PERIF(RARAY,TARAY,NUM,SPI,SINC,SLP,SA,SE)	ORB23800
	ELSEIF (INPUT .EQ. 2) THEN	ORB23810
	CALL GRTRK(AINRAY,APRAY,TARAY,STF,NUM,TIMRAY)	ORB23820
	ELSEIF (INPUT .EQ. 3) THEN	ORB23830
	GOTO 360	ORB23840
	ELSE	ORB23850
	PRINT*, 'INVALID ENTRY!'	ORB23860
	GOTO 340	ORB23870
	ENDIF	ORB23880
		ORB23890
*	DISPLAY DATA	ORB23900
	CALL DATA(SINC,SA,SE,TFEA,TFSU,TFMO,TFDRA,SPER,SPI,TDI,TDA,TDE,	ORB23910
+	TDMM,TDMA,TDLAN,TDH,TDAP,SMM,SMA,SLAN,SH,SAP,SV,SR)	ORB23920
	CALL ENDPL(0)	ORB23930
		ORB23940
*	PROMPT USER IF ANOTHER DISPLAY TYPE IS DESIRED	ORB23950
	PRINT*, 'WOULD YOU LIKE ANOTHER PLOT USING THE SAME ORBITAL'	ORB23960
	PRINT*, 'PARAMETERS AND DATA: '	ORB23970
	PRINT*, 'ENTER "Y" OR "N" : '	ORB23980
	READ*, YORN	ORB23990
	PRINT*, YORN	ORB24000

IF (YORN .EQ. 'Y') THEN	ORB24010
GOTO 340	ORB24020
ENDIF	ORB24030
360 RETURN	ORB24040
END	ORB24050
*****	ORB24060
SUBROUTINE PERIF(RARAY,TARAY,NUM,PI,INC,LP,A,E)	ORB24070
THIS SUBROUTINE PLOTS OUT THE RESULTS OF THE PROGRAM USING THE	ORB24080
DISPLAY FEATURE ON THE MAIN FRAME.	ORB24090
REFER TO DISSPLA USERS GUIDE FOR EXPLANATION OF DISSPLA	ORB24100
SUBROUTINES.	ORB24110
REAL INC,LP	ORB24120
DIMENSION TARAY(500),RARAY(500),RIRAY(500),RJRAY(500),RKRAY(500)	ORB24130
I = 1	ORB24140
*	ORB24150
SET SCALE OF AXIS	ORB24160
RSTEP = (A*(1+E)) / 3	ORB24170
CALL TEN618	ORB24180
CALL RESET(3HALL)	ORB24190
CALL SCMPIN	ORB24200
CALL PHYSOR(1.25,4.)	ORB24210
CALL AREA2D(6.,6.)	ORB24220
CALL MESSAG('PERIFOCAL COORDINATE SYSTEM\$',100,1.0,6.5)	ORB24230
CALL XNAME('XW',2)	ORB24240
CALL YNAME('YW',2)	ORB24250
CALL XAXANG(90.0)	ORB24260
CALL YAXANG(0.0)	ORB24270
CALL INTAXS	ORB24280
CALL POLAR(1.,RSTEP,3.,3.)	ORB24290
CALL POLY3	ORB24300
CALL NOCHEK	ORB24310
CALL CURVE(TARAY,RARAY,NUM,1)	ORB24320
CALL COMPLX	ORB24330
CALL HEIGHT(.2)	ORB24340
CALL RESET('COMPLEX')	ORB24350
CALL RESET('HEIGHT')	ORB24360
CALL ENDGR(0)	ORB24370
*	ORB24380
Display EARTH PLOT	ORB24390
CALL EARTH1(A,E,INC,PI,RSTEP)	ORB24400
RETURN	ORB24410
END	ORB24420
*****	ORB24430
SUBROUTINE EARTH1(A,E,INC,PI,RSTEP)	ORB24440
THIS SUBROUTINE PLOTS A VIEW OF THE WORLD, LOOKING DOWN THE 'Z'	ORB24450
AXIS, PLACED ON THE ORIGIN. THE Latitude IS FIXED, BUT THE	ORB24460
LONGITUDE VARIES WITH THE INCLINATION.	ORB24470
REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA	ORB24480
SUBROUTINES	ORB24490
	ORB24500
	ORB24510
	ORB24520
	ORB24530
	ORB24540
	ORB24550
	ORB24560

REAL INC
COMMON IWORK(3800)
DATA IWDIM/3800/

RE = 6376.145

★ SCALE THE EARTH PLOT AND CENTER ON THE ORIGIN
SCFAC = RE/RSTEP
SCFAC2 = SCFAC * 2.0
XPHS = 1.25 + 3.0 * SCFAC
YPHS = 4.0 + 3.0 * SCFAC
YPOLE = 90 - (INC * 180 / PI)
IF(YPOLE .GT. 90) THEN
 YPOLE = YPOLE - 90
ENDIF
YORIG = YPOLE - 90
YMAX = YPOLE + 90
CALL RESET(3HALL)
CALL PHYSOR(XPHS,YPHS)
CALL PROJECT('LAMBERT EQ/AREA')
CALL MAPOLE(0.0,YPOLE)
CALL AREA2D(SCFAC2,SCFAC2)
CALL THKFRM(0.02)
CALL GRAF(-90.,30.,90.,YORIG,30.,YMAX)
CALL FRAME
CALL MAPFIL('MAPDTA')
CALL LBLANK('LAND',IWDIM)
CALL GRID(1,1)
CALL LBLANK('WATER',IWDIM)
CALL DASH
CALL GRID(1,1)
CALL RESET('DASH')
CALL ENDGR(0)
RETURN
END

SUBROUTINE GRTRK(AINRAY,APRAY,TARAY,TF,NUM,TIMRAY)

DIMENSION AINRAY(500),APRAY(500),TARAY(500),
+ ELARAY(500),ELORAY(500),TLONG(500),TLAT(500),TIMRAY(500)

RE = 6.3782E+03
EROT = 7.292115856E-05
STF = (TF)
I = 1

★ LOAD ARRAYS WITH LATITUDE AND LONGITUDE

410 IF (I .LE. NUM) THEN
 X = RE*COS(APRAY(I))*COS(TARAY(I))-RE*SIN(APRAY(I))*
 + SIN(TARAY(I))
 Y = RE*COS(AINRAY(I))*SIN(APRAY(I))*COS(TARAY(I)) +
 + RE*COS(AINRAY(I))*COS(APRAY(I))*SIN(TARAY(I))
 Z = RE*SIN(AINRAY(I))*SIN(APRAY(I))*COS(TARAY(I)) +
 + RE*SIN(AINRAY(I))*COS(APRAY(I))*SIN(TARAY(I))

ORB24570
ORB24580
ORB24590
ORB24600
ORB24610
ORB24620
ORB24630
ORB24640
ORB24650
ORB24660
ORB24670
ORB24680
ORB24690
ORB24700
ORB24710
ORB24720
ORB24730
ORB24740
ORB24750
ORB24760
ORB24770
ORB24780
ORB24790
ORB24800
ORB24810
ORB24820
ORB24830
ORB24840
ORB24850
ORB24860
ORB24870
ORB24880
ORB24890
ORB24900
ORB24910
ORB24920
ORB24930
ORB24940
ORB24950
ORB24960
ORB24970
ORB24980
ORB24990
ORB25000
ORB25010
ORB25020
ORB25030
ORB25040
ORB25050
ORB25060
ORB25070
ORB25080
ORB25090
ORB25100
ORB25110
ORB25120

*	CALCULATE LATITUDE	ORB25130
	ELARAY(I) = (ASIN(Z/RE)) * (180/3.14159)	ORB25140
*	TRAP 'X' AND 'Y' FOR ARCTAN IN CALCULATING LONGITUDE	ORB25150
	IF((Y .LE. 10) .AND. (Y .GE. 0.0)) THEN	ORB25160
	Y = 10.	ORB25170
	ELSEIF ((Y .GE. -10).AND.(Y .LE. 0.0)) THEN	ORB25180
	Y = -10.	ORB25190
	ENDIF	ORB25200
	IF((X .LE. 10) .AND. (X .GE. 0.0)) THEN	ORB25210
	X = 10.	ORB25220
	ELSEIF ((X .GE. -10).AND.(X .LE. 0.0)) THEN	ORB25230
	X = -10.	ORB25240
	ENDIF	ORB25250
*	CALCULATE LONGITUDE	ORB25260
	ELORAY(I) = (ATAN2(Y,X) - (EROT*TIMRAY(I))) * (180/3.14159)	ORB25270
*	MODIFY LONGITUDES TO (-180 TO 180)	ORB25280
420	IF (ELORAY(I) .LT. -180) THEN	ORB25290
	ELORAY(I) = ELORAY(I) + 360	ORB25300
	GOTO 420	ORB25310
	ENDIF	ORB25320
	I = I + 1	ORB25330
	GOTO 410	ORB25340
	ENDIF	ORB25350
*	SET DISSPLA	ORB25360
	CALL TEN61S	ORB25370
	CALL RESET(3HALL)	ORB25380
	CALL YANANG (0.)	ORB25390
	CALL PHYSOR(1.0,6.0)	ORB25400
	CALL XNAME(' ',1)	ORB25410
	CALL YNAME(' ',1)	ORB25420
	CALL AREA2D(7.5,3.75)	ORB25430
	CALL HEADIN ('GROUND TRACKS',100,1.5,1)	ORB25440
	CALL SCMPLN	ORB25450
	CALL MAPGR(-180.,90.,180.,-90.,30.,90.)	ORB25460
	CALL GRID (1,1)	ORB25470
	CALL MAPFIL ('MAPDTA')	ORB25480
	I = 1	ORB25490
		ORB25500
*	IGNORE Boundary POINTS	ORB25510
430	IF ((ELORAY(I) .LT. -175) .OR.	ORB25520
+	(ELORAY(I) .GT. 175) .OR.	ORB25530
+	(ELARAY(I) .LT. -85) .OR.	ORB25540
+	(ELARAY(I) .GT. 85)) THEN	ORB25550
	I = I + 1	ORB25560
	GOTO 430	ORB25570
	ENDIF	ORB25580
	ITEMP = 1	ORB25590
		ORB25600
*	LOAD FIRST POINT OF NEW PLOT SEGMENT	ORB25610
	IF (I .LE. NUM) THEN	ORB25620
		ORB25630
		ORB25640
		ORB25650
		ORB25660
		ORB25670
		ORB25680

TLONG(ITEMP) = ELORAY(I)	ORB25690
TLAT(ITEMP) = ELARAY(I)	ORB25700
I = I + 1	ORB25710
* IF (I .GE. NUM) THEN	ORB25720
* CALL POLY3	ORB25730
* CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB25740
* ENDIF	ORB25750
ENDIF	ORB25760
	ORB25770
* LOAD SECOND POINT IN LINE SEGMENT	ORB25780
IF (I .LE. NUM) THEN	ORB25790
ITEMP = ITEMP + 1	ORB25800
TLONG(ITEMP) = ELORAY(I)	ORB25810
TLAT(ITEMP) = ELARAY(I)	ORB25820
I = I + 1	ORB25830
IF (I .GE. NUM) THEN	ORB25840
CALL POLY3	ORB25850
CALL NOCHEK	ORB25860
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB25870
ENDIF	ORB25880
ENDIF	ORB25890
	ORB25900
* LOOP UNTIL SEGMENT REACHES EDGE OR NO MORE POINTS	ORB25910
440 IF (I .LE. NUM) THEN	ORB25920
	ORB25930
* BOTH LAT AND LONG INCREASING	ORB25940
IF((ELORAY(I - 2) .LE. ELORAY(I - 1)) .AND.	ORB25950
+ (ELARAY(I - 2) .LE. ELARAY(I - 1))) THEN	ORB25960
IF((ELORAY(I) .LT. -170) .OR.	ORB25970
+ (ELARAY(I) .LT. -80)) THEN	ORB25980
CALL POLY3	ORB25990
CALL NOCHEK	ORB26000
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26010
GOTO 430	ORB26020
ELSE	ORB26030
ITL = ITEMP + 1	ORB26040
TLONG(ITEMP) = ELORAY(I)	ORB26050
TLAT(ITEMP) = ELARAY(I)	ORB26060
ENDIF	ORB26070
	ORB26080
* BOTH LAT AND LONG DECREASING	ORB26090
ELSEIF((ELORAY(I - 2) .GT. ELORAY(I - 1)) .AND.	ORB26100
+ (ELARAY(I - 2) .GT. ELARAY(I - 1))) THEN	ORB26110
IF((ELORAY(I) .GT. 170) .OR.	ORB26120
+ (ELARAY(I) .GT. 80)) THEN	ORB26130
CALL POLY3	ORB26140
CALL NOCHEK	ORB26150
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26160
GOTO 430	ORB26170
ELSE	ORB26180
ITEMP = ITEMP + 1	ORB26190
TLONG(ITEMP) = ELORAY(I)	ORB26200
TLAT(ITEMP) = ELARAY(I)	ORB26210
ENDIF	ORB26220
	ORB26230
* LAT INCREASING, LONG. DECREASING	ORB26240

ELSEIF((ELORAY(I - 2) .GT. ELORAY(I - 1)) .AND.	ORB26250
+ (ELARAY(I - 2) .LE. ELARAY(I - 1))) THEN	ORB26260
IF((ELORAY(I) .GT. 170) .OR.	ORB26270
+ (ELARAY(I) .LT. -80)) THEN	ORB26280
CALL POLY3	ORB26290
CALL NOCHEK	ORB26300
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26310
GOTO 430	ORB26320
ELSE	ORB26330
ITEMP = ITEMP + 1	ORB26340
TLONG(ITEMP) = ELORAY(I)	ORB26350
TLAT(ITEMP) = ELARAY(I)	ORB26360
ENDIF	ORB26370
ORB26380	
★ LAT. DECREASING, LONG. INCREASING	ORB26390
ELSEIF((ELORAY(I - 2) .LE. ELORAY(I - 1)) .AND.	ORB26400
+ (ELARAY(I - 2) .GT. ELARAY(I - 1))) THEN	ORB26410
IF((ELORAY(I) .LT. -170) .OR.	ORB26420
+ (ELARAY(I) .GT. 80)) THEN	ORB26430
CALL POLY3	ORB26440
CALL NOCHEK	ORB26450
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26460
GOTO 430	ORB26470
ELSE	ORB26480
ITEMP = ITEMP + 1	ORB26490
TLONG(ITEMP) = ELORAY(I)	ORB26500
TLAT(ITEMP) = ELARAY(I)	ORB26510
ENDIF	ORB26520
ENDIF	ORB26530
IF(1 .EQ. NUM) THEN	ORB26540
CALL POLY3	ORB26550
CALL NOCHEK	ORB26560
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26570
ENDIF	ORB26580
I = I + 1	ORB26590
GOTO 440	ORB26600
ENDIF	ORB26610
	ORB26620
	ORB26630
CALL POLY3	ORB26640
CALL NOCHEK	ORB26650
CALL CURVE(TLONG,TLAT,ITEMP,1)	ORB26660
	ORB26670
	ORB26680
CALL COMPLX	ORB26690
CALL HEIGHT(.2)	ORB26700
CALL THKFRM (0.03)	ORB26710
CALL FRAME	ORB26720
CALL RESET('COMPLX')	ORB26730
CALL RESET('HEIGHT')	ORB26740
CALL ENDGR (0)	ORB26750
RETURN	ORB26760
END	ORB26770
	ORB26780
*****	ORB26790
	ORB26800

	SUBROUTINE DATA(I,A,E,TFEA,TFSU,TFMO,TFDRA,PER,PI,TDI,TDA,TDE,	ORB26810
	+ TDMH,TDMA,TDLAN,TDH,TDAP,MM,MA,LAN,H,AP,V,R)	ORB26820
★	THIS SUBROUTINE Displays THE ORBITAL DATA FOR BOTH THE PERIFOCAL	ORB26830
★	AND THE GROUND TRACK PLOTS.	ORB26840
★	REFER TO DISSPLA USER'S MANUAL FOR EXPLANATION OF DISSPLA	ORB26850
★	SUBROUTINES	ORB26860
		ORB26870
	REAL I,MM,MA,LAN	ORB26880
	MU = 3.986012E+05	ORB26890
		ORB26900
★	CALCULATE THE AVERAGE FORCES FROM THE TOTAL MAGNITUDE OF	ORB26910
★	FORCE CHANGES	ORB26920
	AVGFE = TFEA/50.0	ORB26930
	AVGFS = TFSU / 50.0	ORB26940
	AVGFM = TFMO / 50.0	ORB26950
	AVGFD = TFDRA / 50.0	ORB26960
		ORB26970
		ORB26980
★	CALCULATE ORBITAL ELEMENTS IN Usable UNITS	ORB26990
	PERH = PER/3600	ORB27000
		ORB27010
	DI = I * (180.0/PI)	ORB27020
	DLAN = LAN * (180.0/PI)	ORB27030
	DAP = AP * (180.0/PI)	ORB27040
		ORB27050
★	CALCULATE Average CHANGE IN ELEMENTS FOR ONE PERIOD	ORB27060
	AVGDI = TDI / 50.0	ORB27070
	AVGDA = TDA / 50.0	ORB27080
	AVGDE = TDE / 50.0	ORB27090
	AVGDMM = TDMH / 50.0	ORB27100
	AVGDMA = TDMA / 50.0	ORB27110
	AVGLAN = TDLAN / 50.0	ORB27120
	AVGDH = TDH / 50.0	ORB27130
	AVGDAP = TDAP / 50.0	ORB27140
		ORB27150
★	CALCULATE RADIUS'S AND VELOCITIES	ORB27160
	ENR = ((V**2)/2) - (MU/R)	ORB27170
	RP = A*(1 - E)	ORB27180
	RA = A*(1 + E)	ORB27190
	VP = SQRT(2*(ENR + (MU/RP)))	ORB27200
	VA = SQRT(2*(ENR + (MU/RA)))	ORB27210
		ORB27220
		ORB27230
★	SET DISSPLA	ORB27240
	CALL RESET(3HALL)	ORB27250
	CALL SCMPLN	ORB27260
	CALL PHYSOR(0.0,0.0)	ORB27270
	CALL AREA2D(8.5,4.0)	ORB27280
		ORB27290
★	PRINT DATA	ORB27300
	CALL MESSAG('I = \$',100,0.25,3.67)	ORB27310
	CALL REALNO(DI,3,'ABUT','ABUT')	ORB27320
	CALL MESSAG(' DEG. \$',100,'ABUT','ABUT')	ORB27330
	CALL MESSAG(' A = \$',100,'ABUT','ABUT')	ORB27340
	CALL REALNO(A,1,'ABUT','ABUT')	ORB27350
	CALL MESSAG(' KM\$ ',100,'ABUT','ABUT')	ORB27360

CALL MESSAG(' E = \$',100,'ABUT','ABUT')	ORB27370
CALL REALNO(E,3,'ABUT','ABUT')	ORB27380
CALL MESSAG(' PER = \$',100,'ABUT','ABUT')	ORB27390
CALL REALNO(PERH,2,'ABUT','ABUT')	ORB27400
CALL MESSAG(' HOURS\$',100,'ABUT','ABUT')	ORB27410
	ORB27420
CALL MESSAG('AVERAGE RATE OF CHANGE OF ELEMENTS PER SECOND \$',	ORB27430
+ 100,1.0,3.0)	ORB27440
	ORB27450
CALL MESSAG('DI/DT = \$',100,0.25,2.67)	ORB27460
CALL REALNO(AVGDI,-2,'ABUT','ABUT')	ORB27470
CALL MESSAG(' DA/DT = \$',100,'ABUT','ABUT')	ORB27480
CALL REALNO(AVGDA,-2,'ABUT','ABUT')	ORB27490
CALL MESSAG(' DE/DT = \$',100,'ABUT','ABUT')	ORB27500
CALL REALNO(AVGDE,-2,'ABUT','ABUT')	ORB27510
	ORB27520
CALL MESSAG('DMM/DT = \$',100,0.25,2.33)	ORB27530
CALL REALNO(AVGDM,-2,'ABUT','ABUT')	ORB27540
CALL MESSAG(' DMA/DT = \$',100,'ABUT','ABUT')	ORB27550
CALL REALNO(AVGDMA,-2,'ABUT','ABUT')	ORB27560
CALL MESSAG(' DLN/DT = \$',100,'ABUT','ABUT')	ORB27570
CALL REALNO(AVGLN,-2,'ABUT','ABUT')	ORB27580
	ORB27590
CALL MESSAG('DH/DT = \$',100,0.25,2.00)	ORB27600
CALL REALNO(AVGDMH,-2,'ABUT','ABUT')	ORB27610
CALL MESSAG(' DAP/DT = \$',100,'ABUT','ABUT')	ORB27620
CALL REALNO(AVGDMA,-2,'ABUT','ABUT')	ORB27630
	ORB27640
CALL MESSAG('AVERAGE MAGNITUDE OF FORCES PER UNIT MASS (KM/S**2)	ORB27650
+ \$',100,1.0,1.67)	ORB27660
	ORB27670
CALL MESSAG('EARTH = \$',100,0.10,1.33)	ORB27680
CALL REALNO(AVGFE,-1,'ABUT','ABUT')	ORB27690
CALL MESSAG(' MOON = \$',100,'ABUT','ABUT')	ORB27700
CALL REALNO(AVGFM,-1,'ABUT','ABUT')	ORB27710
CALL MESSAG(' SUN = \$',100,'ABUT','ABUT')	ORB27720
CALL REALNO(AVGFS,-1,'ABUT','ABUT')	ORB27730
CALL MESSAG(' DRAG = \$',100,'ABUT','ABUT')	ORB27740
CALL REALNO(AVGFD,-1,'ABUT','ABUT')	ORB27750
	ORB27760
CALL MESSAG('PERIGEE\$',100,2.75,1.0)	ORB27770
CALL MESSAG(' Apogee\$',100,'ABUT','ABUT')	ORB27780
	ORB27790
CALL MESSAG('RADIUS (KM)\$',100,0.25,0.67)	ORB27800
CALL MESSAG('RP = \$',100,2.75,0.67)	ORB27810
CALL REALNO(RP,1,'ABUT','ABUT')	ORB27820
CALL MESSAG(' \$',100,'ABUT','ABUT')	ORB27830
CALL MESSAG(' RA = \$',100,'ABUT','ABUT')	ORB27840
CALL REALNO(RA,1,'ABUT','ABUT')	ORB27850
	ORB27860
CALL MESSAG('VELOCITY (KM/SEC)\$',100,0.25,0.33)	ORB27870
CALL MESSAG('VP = \$',100,2.75,0.33)	ORB27880
CALL REALNO(VP,2,'ABUT','ABUT')	ORB27890
CALL MESSAG(' \$',100,'ABUT','ABUT')	ORB27900
CALL MESSAG(' VA = \$',100,'ABUT','ABUT')	ORB27910
CALL REALNO(VA,2,'ABUT','ABUT')	ORB27920

CALL RESET('COMPLX')
CALL ENDGR(0)
RETURN
END

ORB27930
ORB27940
ORB27950
ORB27960
ORB27970
ORB27980

APPENDIX B. COORDINATE SYSTEMS

A. 'IJK': GEOCENTRIC - EQUATORIAL

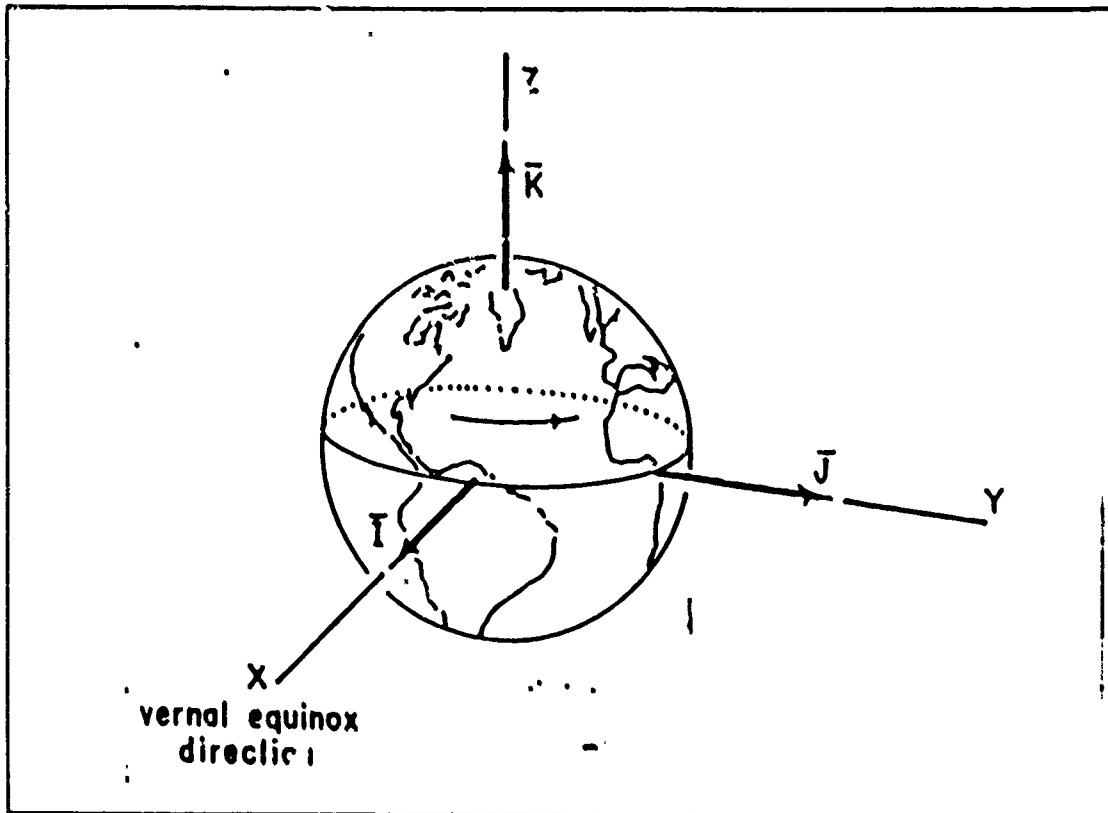


Figure 3. Geocentric-equatorial coordinate system

The geocentric-equatorial system as seen in Figure 3 has its origin at the earth's center. The fundamental plane is in the equator and the positive X-axis points in the vernal equinox direction. The Z-axis points in the direction of the north pole. This system is not fixed to the earth and turning with it; rather, the geocentric-equatorial frame is nonrotating with respect to the stars (except for precession of the equinoxes) and the earth turns relative to it. Unit vectors, $\bar{I}$, $\bar{J}$, and $\bar{K}$ shown in Figure 3, lie along the X, Y, and Z respectively. [Ref. 1: p.55]

B. 'PQW': PERIFOCAL

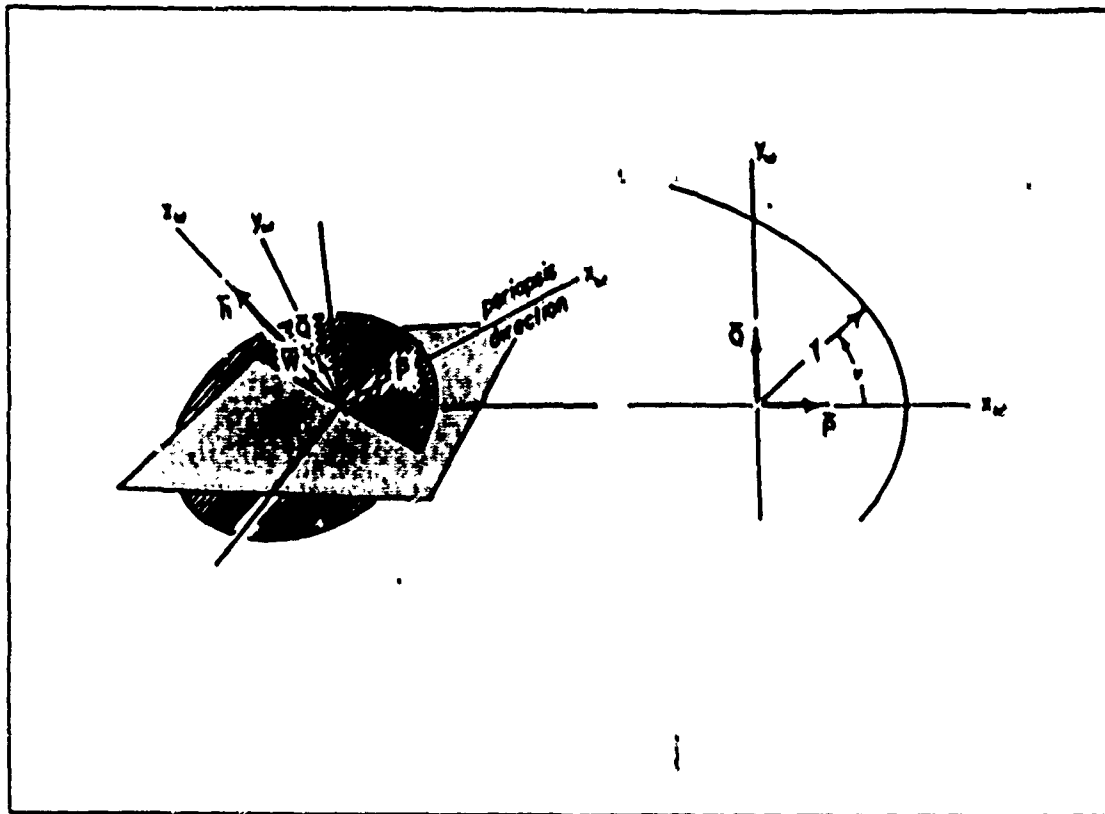


Figure 4. Perifocal coordinate system

The perifocal coordinate system has its fundamental plane in the plane of the satellite's orbit as seen in Figure 4. The coordinate axes are named, X_p , Y_p and Z_p . The X_p axis points toward the perigee; the Y_p axis is rotated 90 degrees in the direction of orbital motion and lies in the orbital plane; the Z_p axis along $\vec{h}$ completes the right-handed perifocal system. Unit vectors in the direction of X_p , Y_p and Z_p are called $\vec{P}$, $\vec{Q}$ and $\vec{W}$ respectively. [Ref. 1: p.57]

C. 'RSW': ORBITAL

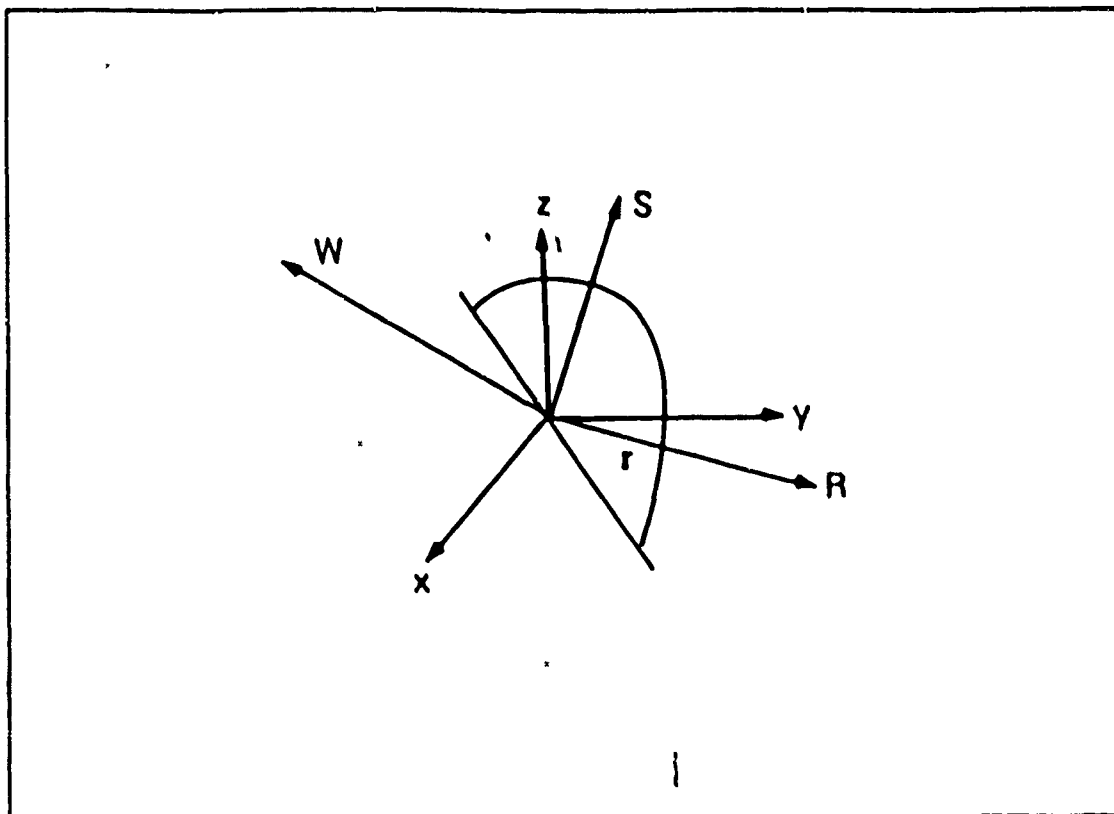


Figure 5. Orbital coordinate system

(Figure 9.4-1, Ref. 1)

The orbital coordinate system has its principle axis, R (unit vector r), along the instantaneous radius vector, r as seen in Figure 5. The axis S is rotated 90 degrees from R in the direction of increasing true anomaly. The third axis, W , is perpendicular to both R and S . Note that this coordinate system is simply rotated v_0 from the PQW perifocal system. [Ref. 1: p.398]

D. COORDINATE TRANSFORMATIONS

The coordinate transformations, for the previous coordinate systems, use angular rotations about the axis to evaluate the transformation matrix. The matrix elements r_{ij} are calculated, then applied to the old vector to get the vector in the new coordinate system. The following orbital elements are used:

Ω = longitude of ascending node

ω = argument of perigee

i = inclination

u_0 = argument of latitude

v_0 = true anomaly

The coordinate transformations follow [Ref. 1: p.74-83]

1. PQW to IJK

$$\begin{aligned} r_{11} &= \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i \\ r_{12} &= -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i \\ r_{13} &= \sin \Omega \cos \omega \\ r_{21} &= \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i \\ r_{22} &= -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i \\ r_{23} &= -\cos \Omega \sin i \\ r_{31} &= \sin \omega \sin i \\ r_{32} &= \cos \omega \sin i \\ r_{33} &= \cos i \\ \vec{I} &= r_{11}\vec{P} + r_{12}\vec{Q} + r_{13}\vec{W} \\ \vec{J} &= r_{21}\vec{P} + r_{22}\vec{Q} + r_{23}\vec{W} \\ \vec{K} &= r_{31}\vec{P} + r_{32}\vec{Q} + r_{33}\vec{W} \end{aligned}$$

2. IJK to PQW (inverse of #1)

$$\begin{aligned} \vec{P} &= r_{11}\vec{I} + r_{21}\vec{J} + r_{31}\vec{K} \\ \vec{Q} &= r_{12}\vec{I} + r_{22}\vec{J} + r_{32}\vec{K} \\ \vec{W} &= r_{13}\vec{I} + r_{23}\vec{J} + r_{33}\vec{K} \end{aligned}$$

3. IJK to RSW

$$\begin{aligned} r_{11} &= \cos \Omega \cos u_0 - \sin \Omega \sin u_0 \cos i \\ r_{12} &= \sin \Omega \cos u_0 + \sin u_0 \cos \Omega \cos i \\ r_{13} &= \sin i \sin u_0 \\ r_{21} &= -\cos \Omega \sin u_0 - \sin \Omega \cos u_0 \cos i \\ r_{22} &= -\sin \Omega \sin u_0 + \cos \Omega \cos u_0 \cos i \\ r_{23} &= \cos u_0 \sin i \\ r_{31} &= \sin \Omega \sin i \\ r_{32} &= -\cos \Omega \sin i \\ r_{33} &= \cos i \\ \vec{R} &= r_{11}\vec{I} + r_{12}\vec{J} + r_{13}\vec{K} \\ \vec{S} &= r_{21}\vec{I} + r_{22}\vec{J} + r_{23}\vec{K} \\ \vec{W} &= r_{31}\vec{I} + r_{32}\vec{J} + r_{33}\vec{K} \end{aligned}$$

4. RSW to IJK (inverse of #3)

$$\vec{I} = r_{11}\vec{R} + r_{21}\vec{S} + r_{31}\vec{H}$$

$$\vec{J} = r_{12}\vec{R} + r_{22}\vec{S} + r_{32}\vec{H}$$

$$\vec{K} = r_{13}\vec{R} + r_{23}\vec{S} + r_{33}\vec{H}$$

5. PQW to RSW

$$r_{11} = \cos v_0$$

$$r_{12} = \sin v_0$$

$$r_{13} = 0.0$$

$$r_{21} = -\sin v_0$$

$$r_{22} = \cos v_0$$

$$r_{23} = 0.0$$

$$r_{31} = 0.0$$

$$r_{32} = 0.0$$

$$r_{33} = 1.0$$

$$\vec{R} = r_{11}\vec{P} + r_{12}\vec{Q} + r_{13}\vec{H}$$

$$\vec{S} = r_{21}\vec{P} + r_{22}\vec{Q} + r_{23}\vec{H}$$

$$\vec{H} = r_{31}\vec{P} + r_{32}\vec{Q} + r_{33}\vec{H}$$

6. RSW to PQW (inverse of #5)

$$\vec{P} = r_{11}\vec{R} + r_{21}\vec{S} + r_{31}\vec{H}$$

$$\vec{Q} = r_{12}\vec{R} + r_{22}\vec{S} + r_{32}\vec{H}$$

$$\vec{H} = r_{13}\vec{R} + r_{23}\vec{S} + r_{33}\vec{H}$$

APPENDIX C. ORBITAL ELEMENTS

The user is assumed to be studying orbital mechanics and should understand the orbital elements and how to calculate them. A brief description of the elements and the equations used to calculate the elements follow. For a detailed explanation of the elements and the equations to calculate them refer to Chapters 1 and 2 of reference 1. Figure 6 on page 83 shows the orbital elements in the Geocentric-Equatorial and perifocal coordinate system.

1. Angular Momentum (h):

The specific angular momentum is a constant of the motion of the satellite, defined as $\vec{h} = \vec{r} \times \vec{v}$.

$$\vec{h} = \vec{r} \times \vec{v} = h_I \vec{I} + h_J \vec{J} + h_K \vec{K}$$

$$h_I = r_J v_K - r_K v_J$$

$$h_J = r_K v_I - r_I v_K$$

$$h_K = r_I v_J - r_J v_I$$

$$h = \sqrt{h_I^2 + h_J^2 + h_K^2}$$

2. Node Vector (n):

The node vector is a vector pointing along the line of nodes in the direction of the ascending node.

$$\vec{n} = \vec{K} \times \vec{h} = -h_J \vec{I} + h_I \vec{J}$$

$$n = \sqrt{h_J^2 + h_I^2}$$

3. Semi-latus rectum (p):

The semi-latus rectum is a geometric constant of the conic section.

$$p = \frac{h^2}{\mu}$$

4. Eccentricity (e):

The eccentricity is a constant defining the shape of the conic orbit.

$$\vec{e} = \frac{1}{\mu} \left[\left(v^2 - \frac{\mu}{r} \right) \vec{r} - (\vec{r} \cdot \vec{v}) \vec{v} \right]$$

$$e = |\vec{e}|$$

5. Semi-major axis (a):

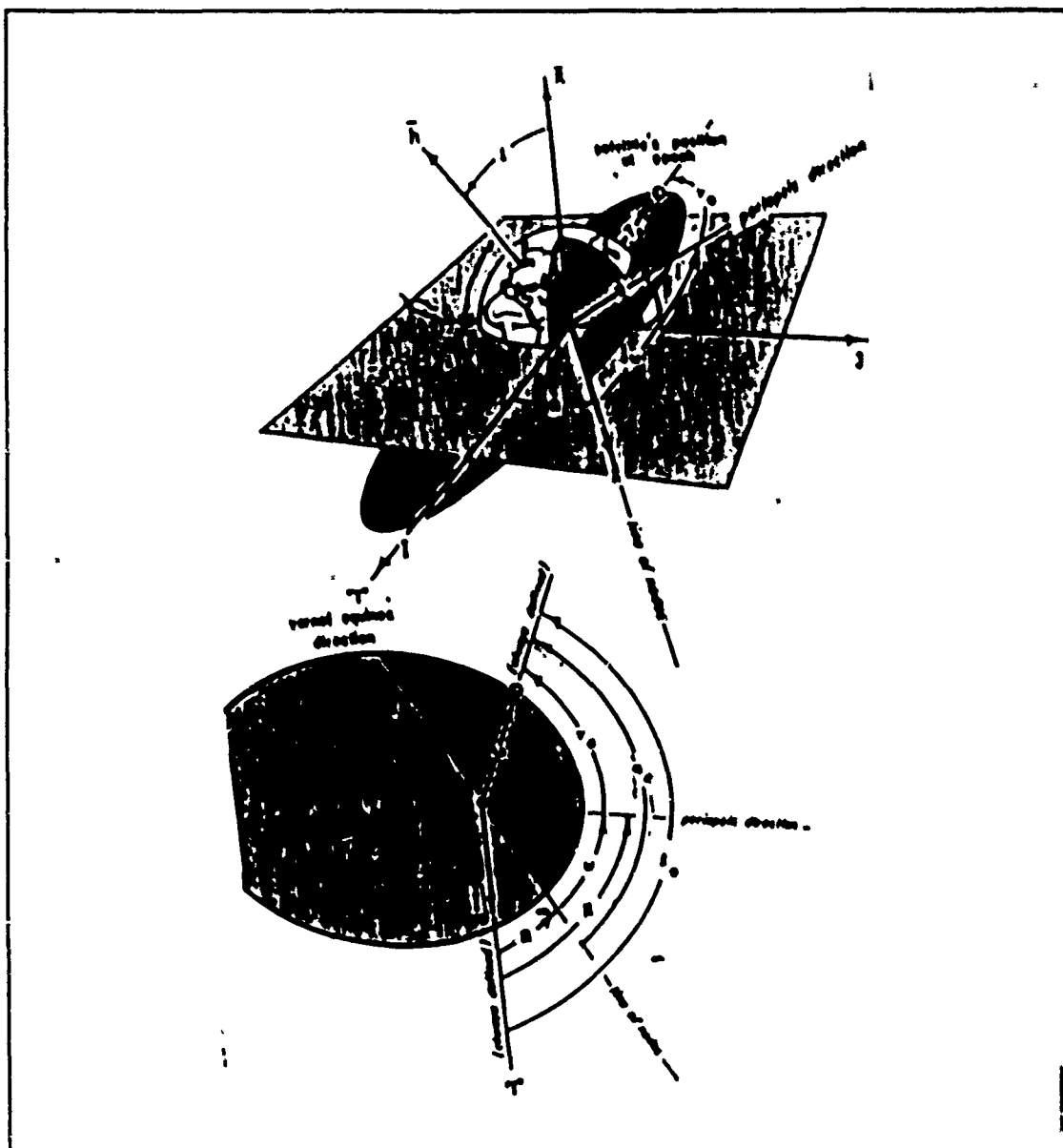


Figure 6. Orbital elements

The semi-major axis is a constant defining the size of the orbit.

$$a = \frac{(1 - e^2)}{p}$$

6. Inclination (i):

The inclination is the angle between the 'K' unit vector in the 'IJK' system and the angular momentum vector, 'h'.

$$i = \cos^{-1} \left(\frac{\vec{h} \cdot \vec{K}}{h} \right) = \cos^{-1} \left(\frac{h_z}{h} \right)$$

7. Longitude of ascending node (Ω):

The longitude of the ascending node is the angle in the fundamental plane, between the 'I' unit vector and the point where the satellite crosses through the fundamental plane in a northerly direction (ascending node) measured counter-clockwise when viewed from the north side of the fundamental plane.

$$\Omega = \cos^{-1} \left(\frac{n_z}{n} \right)$$

8. Argument of perigee (ω):

The argument of perigee is the angle in the plane of the satellite's orbit, between the ascending node and the perigee point, measured in the direction of the satellite's motion.

$$\omega = \cos^{-1} \left(\frac{\vec{n} \cdot \vec{e}}{ne} \right) = \cos^{-1} \frac{(n_1 e_1 + n_2 e_2)}{ne}$$

9. True anomaly at epoch (v_0):

The true anomaly at epoch is the angle in the plane of the satellite's orbit, between perigee and the position of the satellite at a particular time, t_0 , called the "epoch".

$$v_0 = \cos^{-1} \left(\frac{\vec{e} \cdot \vec{r}}{er} \right)$$

10. Argument of latitude (u):

The argument of latitude is the angle in the plane of the orbit, between the ascending node and the radius vector to the satellite at time t_0 .

$$u_0 = \cos^{-1} \left(\frac{\vec{n} \cdot \vec{r}}{nr} \right)$$

11. Longitude of perigee (Π):

The longitude of perigee is the angle from 'I' to perigee measured eastward to the ascending node and then in the orbital plane to perigee.

$$\Pi = \Omega + \omega$$

12. True longitude at epoch (l_0):

The true longitude at epoch is the angle between 'I' and r_0 (the radius vector to the satellite at t_0 measured eastward to the ascending node and then in the orbital plane to r_0).

$$l_0 = \omega + \Omega + v_0$$

13. Period (per):

The period is the time the for the satellite to complete one orbit.

$$Per = 2\pi \frac{a^3}{\mu}$$

14. Eccentric anomaly (EA):

The eccentric anomaly is the angle between the perigee and a position on an auxiliary circle circumscribed about the ellipse where a perpendicular line to the major axis has been extended from the epoch location of the satellite to the auxiliary circle.

$$EA = \cos^{-1} \frac{e + \cos(v)}{1 + e \cos(v)}$$

15. Mean motion (n'):

The mean motion is defined below:

$$n' = \sqrt{\frac{\mu}{a^3}}$$

16. Mean anomaly (MA):

The mean anomaly is defined below:

$$MA = n'(t - T) = EA - e \sin(EA)$$

17. Time of flight (TF):

The time of flight is the elapsed time from when the satellite was at perigee to the current epoch.

$$(t - T) = \sqrt{\frac{a^3}{\mu}} (EA - e \sin(EA))$$

APPENDIX D. SAMPLE ORBITS

To demonstrate the capabilities of the program, a variety of orbital plots will follow:

1. Low earth orbit (LEO).

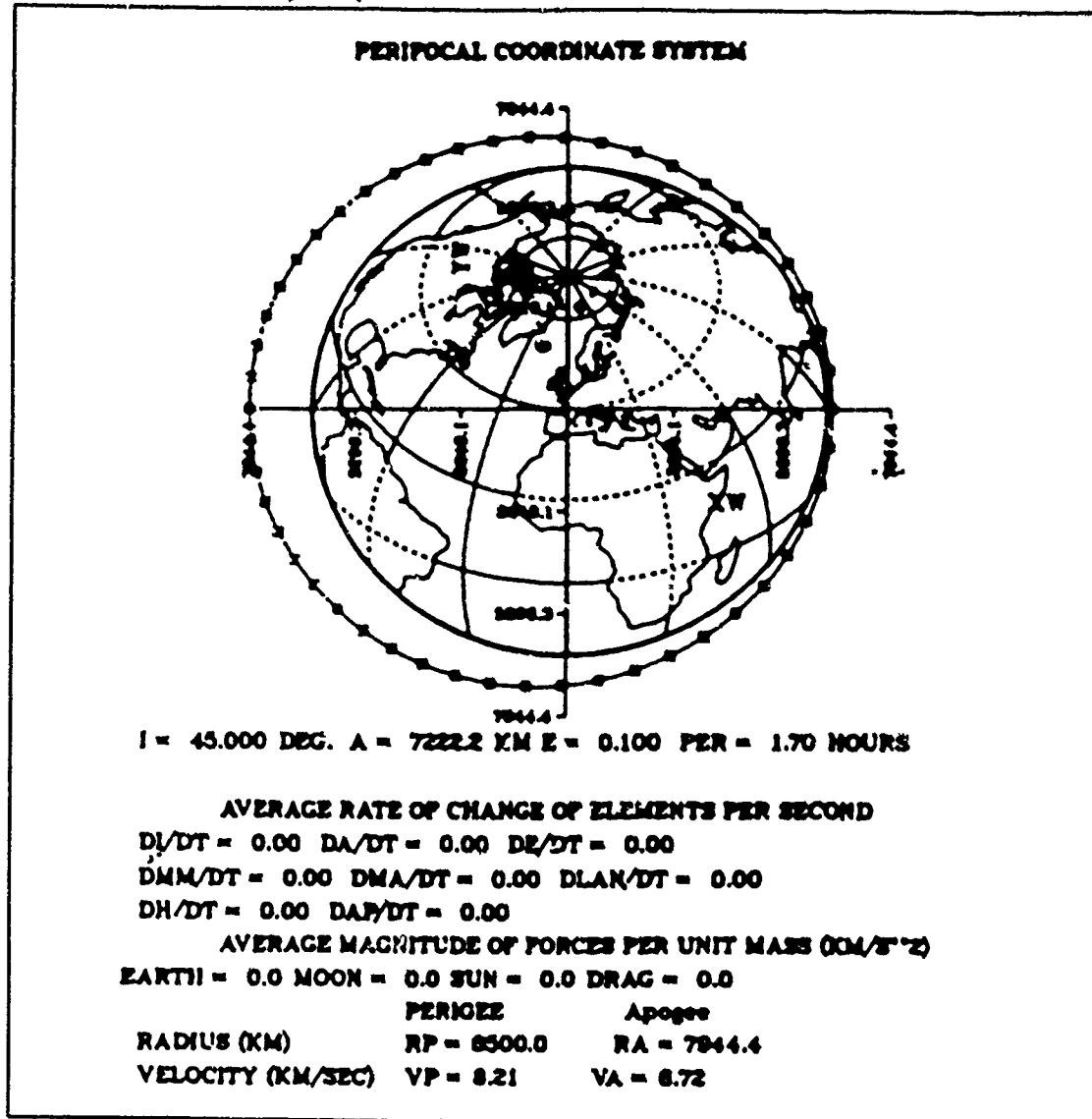


Figure 7. Unperturbed Low Earth Orbit (LEO)

Figure 7 shows the perifocal plot of a satellite in an unperturbed low earth orbit (LEO). The initial parameters of the orbit were entered as follows:
radius of perigee (RP) = 6500 km
eccentricity (e) = 0.1
inclination (i) = 45 degrees.

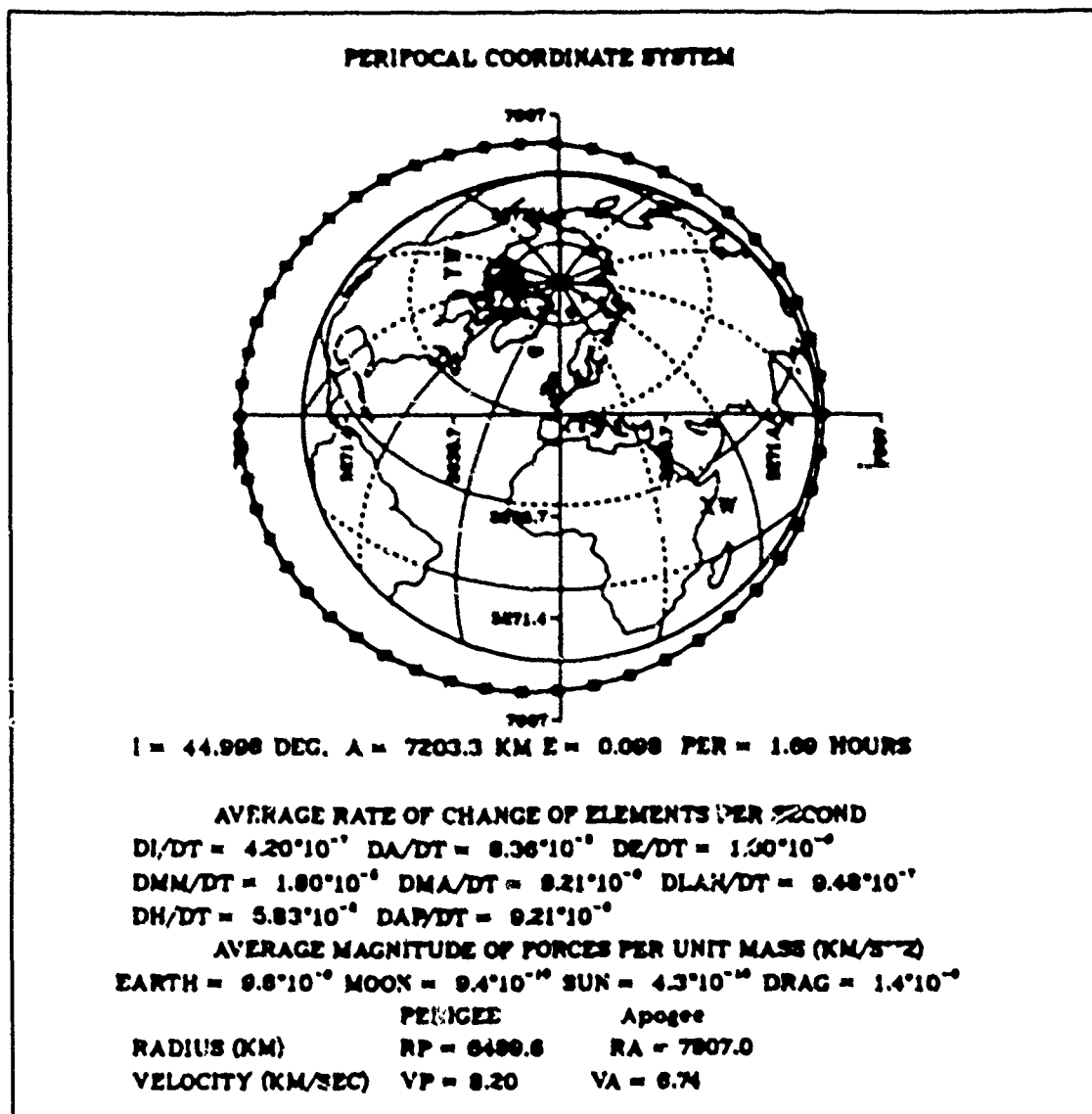


Figure 8. Perturbed Low Earth Orbit (LEO)

With perturbing forces applied to the previous LEO, the drag force will be the dominate perturbing force. The drag will act as a negative velocity change applied in the area of perigee, with the result of decreasing the semi-major axis length, this in effect will decrease the eccentricity of the orbit, as can be seen by comparing the orbital data of the unperturbed LEO in Figure 7 on page S6 with the orbital data of the perturbed LEO in Figure 8.

2. Circular orbit.

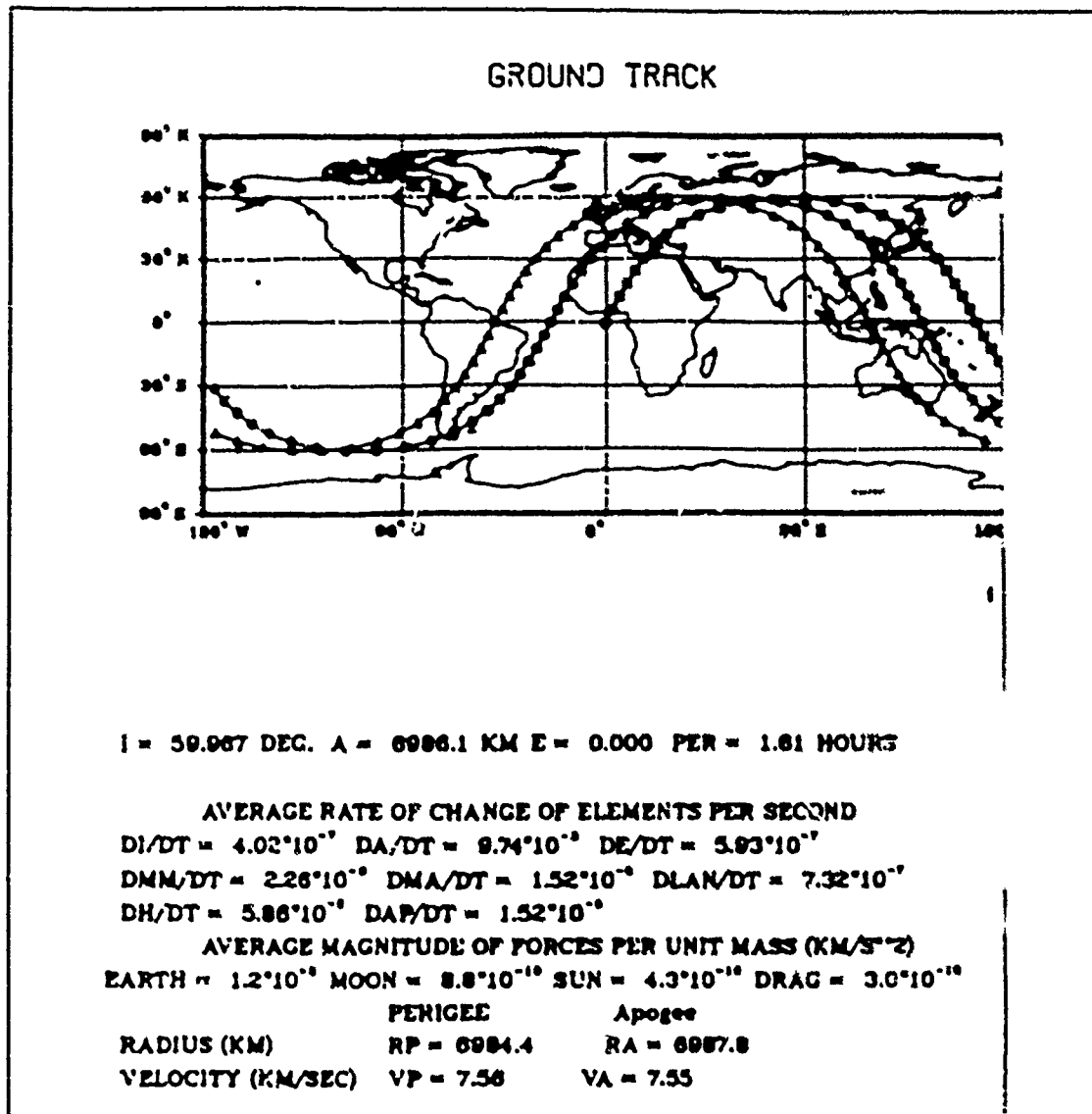


Figure 9. Circular Orbit

An example of the plot of the ground track of a sequence of three 60 degree inclined perturbed circular orbits with a radius of 7000 km is shown in Figure 9. The sequence of orbits displays the precession of the orbit around the earth.

3. Transfer orbit.

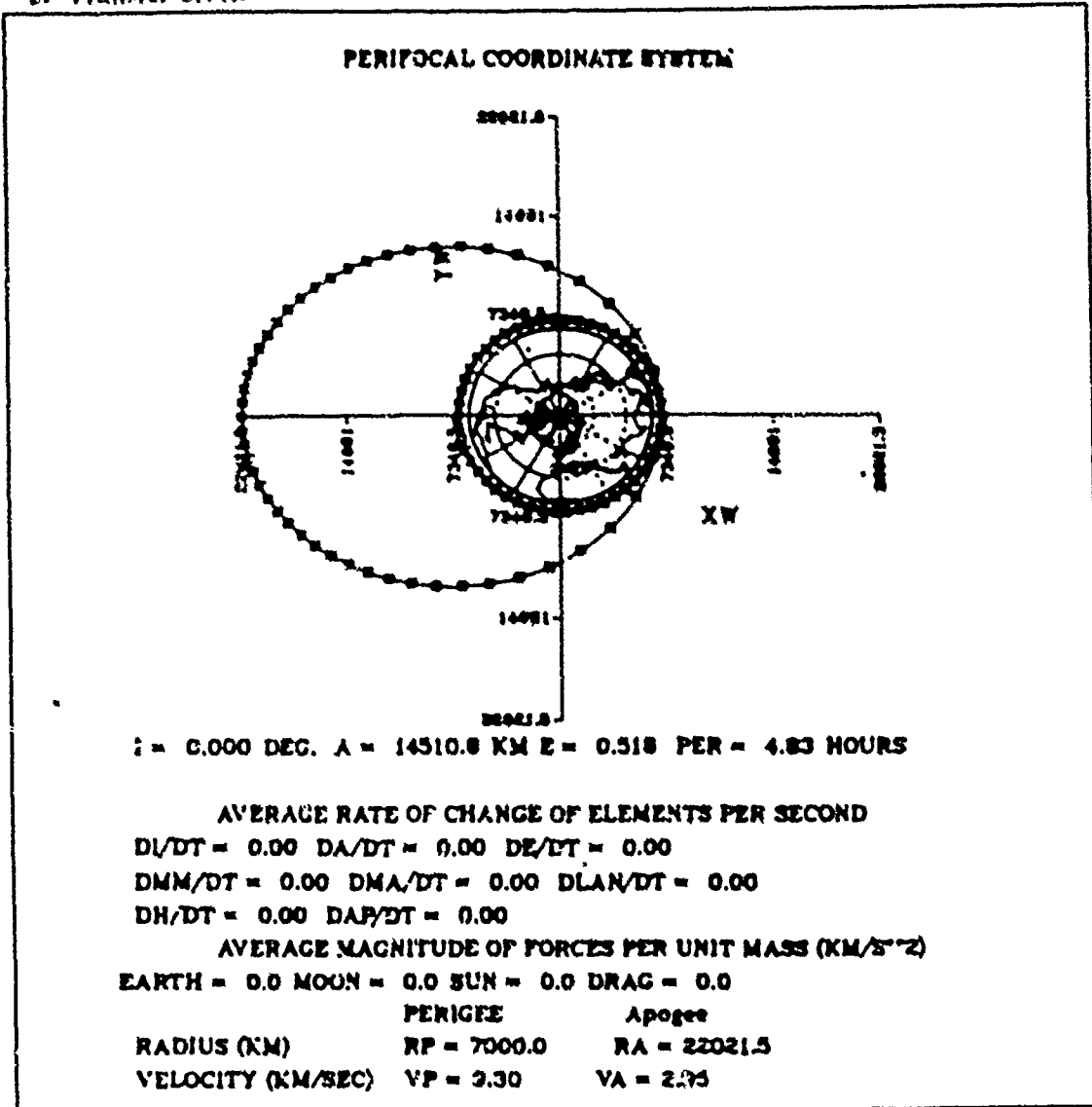


Figure 10. Transfer Orbit

The transfer orbit between a circular, equatorial LEO and a molniya orbit (high eccentric orbit) is shown in Figure 10. A velocity increase of 1.75 km/s was applied at the perigee to simulate a perigee kick to boost the satellite into the molniya orbit. A similar velocity change could then be applied at apogee to create a high altitude circular orbit, or a negative velocity change applied at perigee could be used to bring the satellite back to a LEO.

4. Geosynchronous orbit

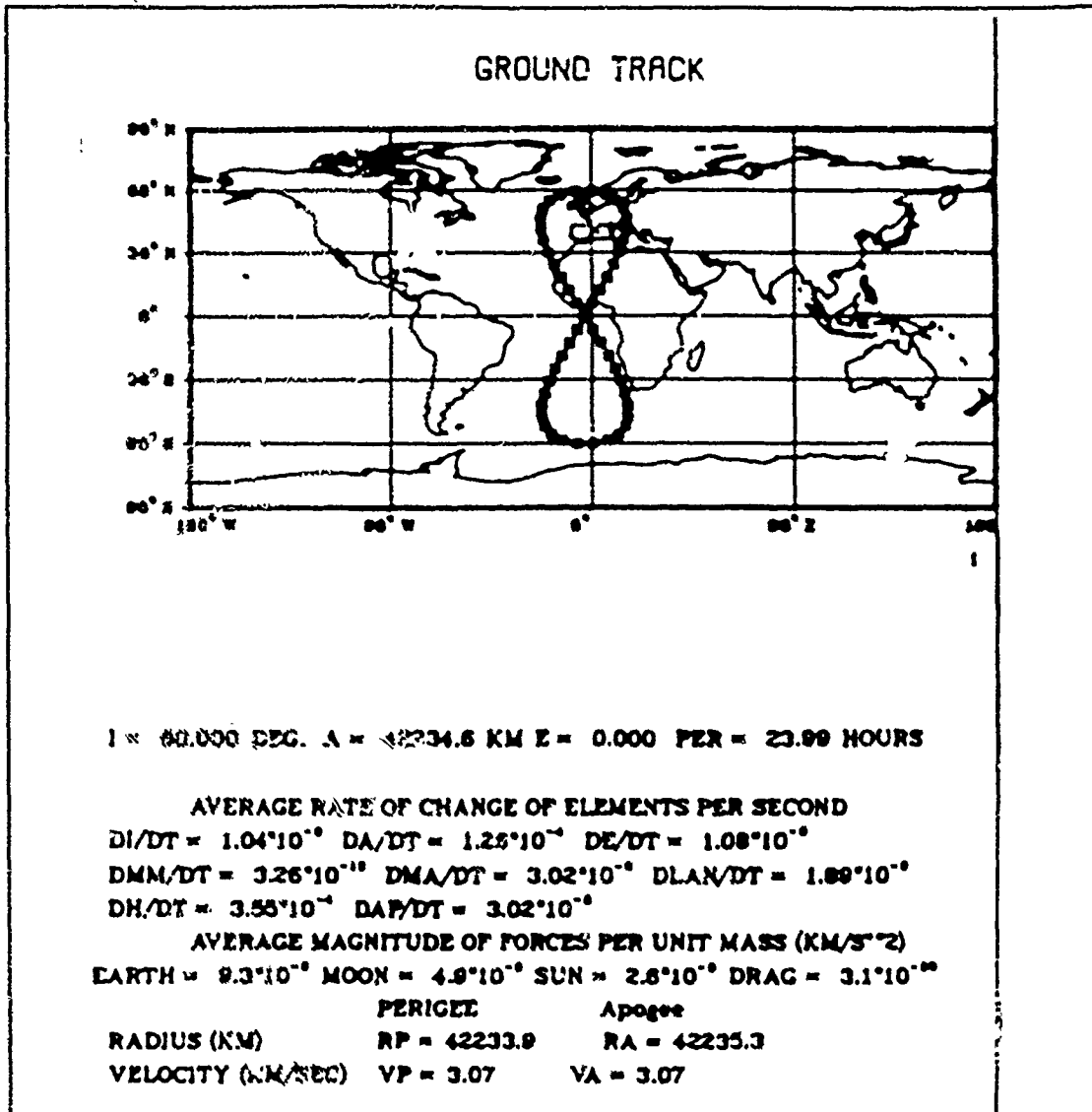


Figure 11. Geosynchronous Orbit

The ground track of a perturbed geosynchronous orbit inclined 60 degrees is shown in Figure 11. The orbit displays the figure eight typical with inclined geosynchronous orbits.

LIST OF REFERENCES

1. Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications, Inc., 1971.
2. Martin Marietta Corporation, Space Systems Division, *Orbital Flight Handbook*, volume 1, 1963.
3. Agrawal, B.N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., 1968.
4. Roy, A.E., *Orbital Motion*, Adam Hilger Ltd., 1982.
5. Computer Associates, *DISSPLA User's Manual*, version 10.0, 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Commander Naval Space Command Attn: Code N3 Dahlgren, VA 22448	1
4.	Commander United States Space Command Attn: Technical Library Peterson AFB, CO 80914	1
5.	Director Navy Space System Division Chief of Naval Operations (OP-943) Washington, D.C. 20305-2000	1
6.	Superintendent Naval Postgraduate School Space System Academic Group Attn: Prof. R. Panholzer (Code 62Pz) Monterey, CA 93943-5000	1
7.	Superintendent Naval Postgraduate School Attn: Prof. E. Milne (Code 61Mn) Monterey, CA 93943-5000	1
8.	Superintendent Naval Postgraduate School Attn: Prof. D. Danielson (Code 53Dd) Monterey, CA 93943-5000	1
9.	Commanding Officer Attn: LT Anthony Vraa Department Head School Class 111 Surface Warfare Officer School Newport, RI 02841-5012	2