AL-TR-90-006 AD:

AD-A224 201

Final Report
for the period
August 1986 to
December 1988

DEVELOPMENT OF THE ELECTRIC VEHICLE ANALYZER

June 1990

Authors:
M. R. Dickey Lt USAF
R. S. Klucz
K. A. Ennix
L. M. Matuszak Maj USAFR

Approved for Public Release

Astronautics Laboratory (AFSC)
Air Force Space Technology Center
Space Systems Division
Air Force Systems Command
Edwards AFB CA 93523-5000

NOTICE

When U.S. Government drawing, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any way licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may be related thereto.

FOREWARD

This special report was submitted on completion of this phase of JON: 286400N4 at the Astronautics Laboratory (AFSC), Edwards Air force Base CA. AL Project Manager was Lt Michael R. Dickey.

This report has been reviewed and is approved for release and distribution in accordance with the distribution statement on the cover and on the DD Form 1473.

MICHAEL R. DICKEY, 1Lt, USAF
Project Manager

RICHARD S. MATLOCK
Chief, Integration and Analysis Branch

FOR THE DIRECTOR

MICHAEL E. HAVEY, COLONEL, USAF
Deputy Director

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICITIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AL-TR-90-006	5. MONITORING ORGANIZATION REPORT NUMBERS(S)
6a. NAME OF PERFORMING ORGANIZATION Astronautics Laboratory (AFSC) 6b. OFFICE SYMBOL *(If applicable)* AL/VSAB	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)* AL/VSAB Edwards AFB, CA 93523-5000	7b. ADDRESS *(City, State, and ZIP Code)*
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.
62302F	2864	00	N4

11. TITLE *(Include Security Classification)*

Development of the Electric Vehicle Analyzer (U)

12. PERSONAL AUTHOR(S)

Dickey, Lt Michael R. ; Klucz, Raymond S. ; Ennix, Kimberly A. ; Matuszak, Maj Leo M. , USAFR

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Special	FROM 89. TO 89.10	9006	42

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
21	03	
22	03	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

Electric Orbital Transfer Vehicles (EOTVs); Electric Insertion Transfer Experiment (ELITE); Electric Vehicle Analyzer (EVA).

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The increasing technological maturity of high power (>20 kW) electric propulsion devices has led to renewed interest in their use as a means of efficiently transferring payloads between earth orbits. Several systems and architecture studies have identified the poten tial cost benefits of high perfor mance Electric Orbital Transfer Vehicles (EOTVs) , . These studies led to the initia tion of the Electric Insertion Transfer Experiment (ELITE) in 1988. Managed by the Astronautics Laboratory, ELITE is a flight experiment designed to sufficiently demonstrate key technologies and options to pave the way for the full-scale development of an operational EOTV.

An important consideration in the development of the ELITE program is the capability of available analytical tools to simulate the or bital mechanics of a low thrust, electric propulsion transfer vehicle. These tools are necessary not only for ELITE mission planning exercises but also for continued, effi cient, accurate evaluation of DoD space transportation architectures which include EOTVs. This paper presents such a tool: the Electric Vehicle Analyzer (EVA).

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Lt Michael R. Dickey	22b. TELEPHONE *(Include Area Code)* (805) 275-5524 22c. OFFICE SYMBOL AL/VSAB

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

A ≡ Vehicle cross sectional area
a ≡ Semi-major axis
B^* ≡ Eclipse entrance/exit angle
C_d ≡ Drag coefficient
D ≡ Drag force
E ≡ Specific mechanical energy
F ≡ Applied force on vehicle
f_e ≡ Fraction of orbit in eclipse
f_s ≡ Fraction of orbit in sun
g_c ≡ Pound mass/pound weight conversion factor
g_o ≡ Gravitational acceleration at the earth's surface
h ≡ Vehicle altitude
i ≡ Inclination
I_{sp} ≡ Specific impulse
Δi ≡ Inclination change imparted to the orbit
J_2 ≡ C_{20} gravitational coefficient (oblateness)
m ≡ Vehicle mass
$\dot{m}$ ≡ Propellant mass flow rate
M_F ≡ Dry vehicle mass
M_I ≡ Wet vehicle mass
M_t ≡ Thermal management system mass
P ≡ Available power
P_I ≡ Initial system power
P_T ≡ Period of transfer orbit
$\bar{q}$ ≡ Dynamic pressure
r ≡ Instantaneous radius of spacecraft orbit
$\dot{r}$ ≡ Rate of change of radius
r_1 ≡ Radius of initial orbit
r_2 ≡ Radius of final orbit
R_E ≡ Radius of Earth
T ≡ Total thrust
t ≡ Time since beginning of transfer
T_{eff} ≡ Effective thrust
T_N ≡ Normal thrust
t_o ≡ Time of vernal equinox (March 21)
T_p ≡ In-plane thrust
V ≡ Orbit velocity
v_1 ≡ Initial orbit velocity
v_2 ≡ Final orbit velocity
ΔV ≡ Velocity change imparted to the orbit
W_t ≡ Thermal managment system specific mass
α ≡ Out-of-plane steering angle
α_s ≡ Solar right ascension
δ ≡ Solar declination
ε ≡ Obliquity of ecliptic
γ ≡ Flight path angle
η_{ppu} ≡ Power processor efficiency
η_t ≡ Thruster efficiency
μ ≡ Earth's gravitational parameter
ρ ≡ Atmospheric density
$\dot{\theta}$ ≡ Angular rate of vehicle
ω_S ≡ Angular motion of sun
Ω ≡ Right ascension of ascending node
$\dot{\Omega}$ ≡ Rate of change of Ω

INTRODUCTION

The increasing technological maturity of high power (>20 kW) electric propulsion devices has led to renewed interest in their use as a means of efficiently transferring payloads between earth orbits. Several systems and architecture studies have identified the potential cost benefits of high performance Electric Orbital Transfer Vehicles (EOTVs)[1,2]. These studies led to the initiation of the Electric Insertion Transfer Experiment (ELITE) in 1988.[3] Managed by the Astronautics Laboratory, ELITE is a flight experiment designed to sufficiently demonstrate key technologies and options to pave the way for the full-scale development of an operational EOTV.

An important consideration in the development of the ELITE program is the capability of available analytical tools to simulate the orbital mechanics of a low thrust, electric propulsion transfer vehicle. These tools are necessary not only for ELITE mission planning exercises but also for continued, efficient, accurate evaluation of DoD space transportation architectures which include EOTVs. This paper presents such a tool: the Electric Vehicle Analyzer (EVA).

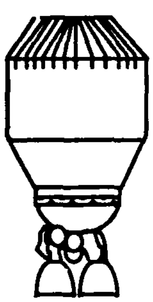
BACKGROUND

To properly understand the need for EVA and its modeling capabilities, it is necessary to look at some background information which has shaped its development.

EOTV Design and Operation

An Orbital Transfer Vehicle (OTV) is a vehicle used to transfer payload (e.g., a satellite) from a low earth orbit into a higher orbit. One of the most common destination orbits, as well as one of the most demanding on the OTV, is the geosynchronous orbit. A circular orbit in the plane of the equator at an altitude of 35000 km, with a period of exactly one day. This orbit is popular for communications satellites.

The main difference between conventional OTVs and EOTVs is the type of propulsion. Conventional OTVs use chemical propulsion, whereas EOTVs utilize externally supplied electrical power instead of a chemical reaction. There are different options possible to supply the electric power including photovoltaic, solar dynamic, and nuclear power systems. ELITE has baselined photovoltaic solar arrays to provide electrical power.



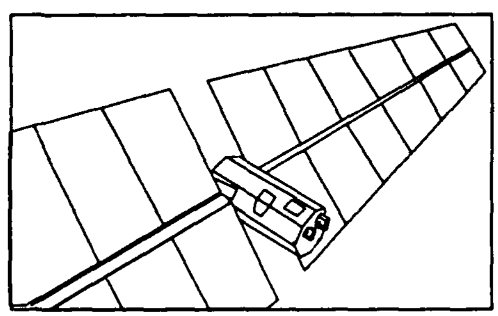


Figure 1
OTV Comparison

Although the theoretical I_{sp} limits on chemical propulsion (≈500 sec) do not apply to electric propulsion, allowing I_{sp}s up to 5,000 seconds, photovoltaic power limitations restrict thrust levels to less than 10N (≈2 lb). Short, impulsive burns which are characteristic of chemical propulsion are not possible at such low thrust levels, and continuous thrusting throughout the mission becomes the standard mode of operation. Hohmann transfers, which consist of two impulsive, high-thrust burns (Figure 2), give way to trajectories that slowly spiral upward to the destination orbit, circling the earth hundreds or even thousands of times along the way.

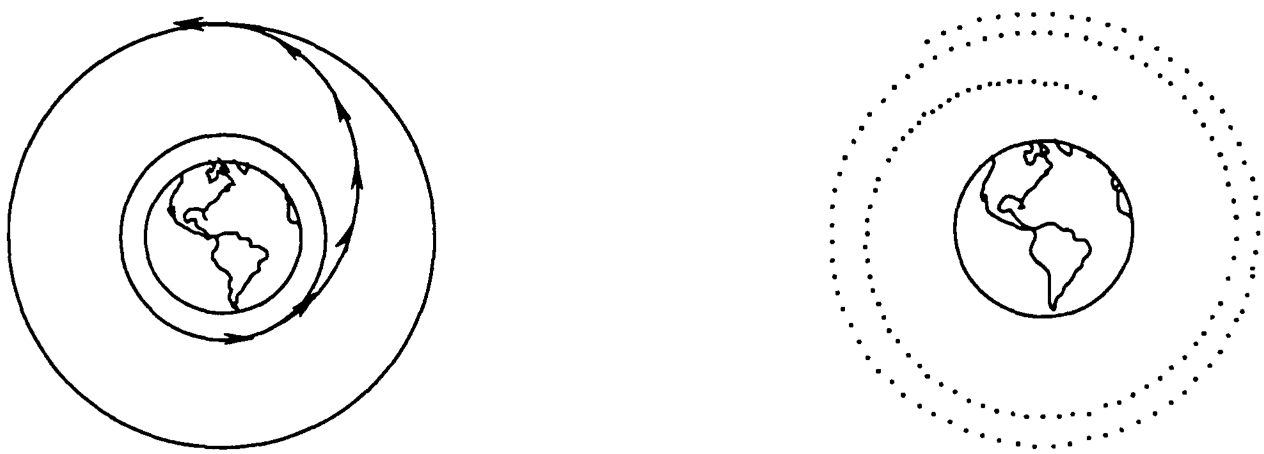


Figure 2
Hohmann vs Spiral Trajectory

Desired Analytical Capability

In order to do their job, EOTV mission planners will have to have knowledge of the effects of operating a vehicle in such a slow, low-thrust, spiral trajectory. Some of these effects include gravity losses, atmospheric drag, solar occultation, and radiation damage to the solar arrays.

Gravity Loss. One consequence of flying a low-thrust spiral trajectory rather than the conventional two impulse burn Hohmann transfer is gravity loss. In the ideal Hohmann scenario, an impulse burn takes place at periapsis and at apoapsis where the thrust vector is perpendicular to the gravity vector of the central body. All thrust applied can be used to add energy to the orbit. Longer burn times create the situation shown in Figure 3; in which the burn takes place through points on the orbit, at which the flight path angle is not zero. A portion of the thrust must therefore by used to overcome the local effect of gravity. This manifests itself as a loss, and for low earth to geosynchronous missions can increase total velocity (Δv) required to complete the mission by 30% .

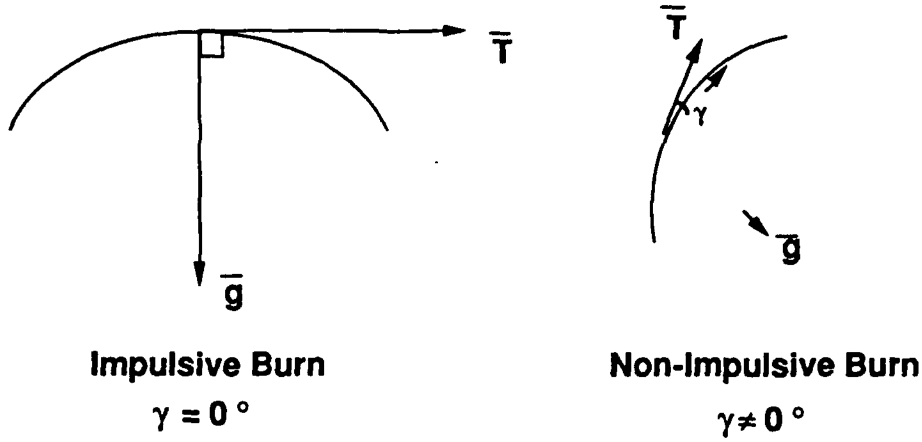


Figure 3
Impulsive vs Non-Impulsive Burn

Atmospheric Drag. With current state of the art solar array technology, array areas on the order of hundreds of square meters are necessary to provide enough electric power to produce reasonable thrust and trip times for EOTVs. These large areas, combined with the low thrust of the electric thrusters, make atmospheric drag at low earth orbits a significant factor in mission planning. These factors may even dictate minimum deployment altitudes below which the thrust to drag ratio will be unacceptable, causing decay of the vehicle's orbit and ultimately atmospheric reentry.

Occultation. Time spent in the Earth's shadow, or solar occultation, is another significant factor to consider in the mission planning of solar-powered EOTVs. Since such an EOTV derives its electrical power from the sun, it must coast through the occultation periods. This not only alters the shape of the spiral trajectory but also affects the trip time. To illustrate this effect consider an EOTV starting out in a circular orbit (Figure 4). It is thrusting when in sun; not thrusting when in shadow. This period of

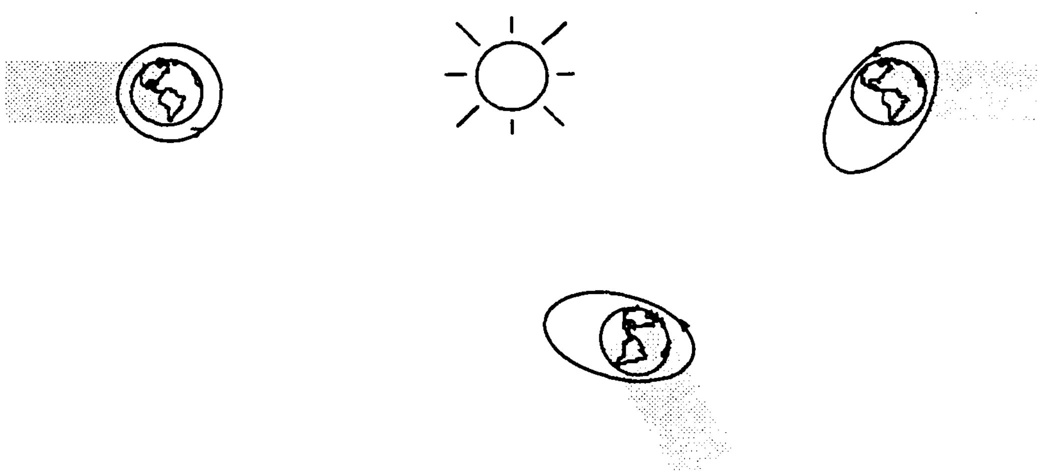


Figure 4
Effect of Occultation on Trajectory

coasting will cause the orbit to become slightly elliptical. Also, since the shadow cylinder rotates in inertial space as the Earth moves in its orbit about the sun, the perigee location of the now slightly elliptical orbit will also rotate in inertial space. Thus as the EOTV spirals upward toward its destination orbit, its trajectory will be a slightly elliptical spiral rather than a perfectly circular spiral.

A typical eclipse history is shown in Figure 5 for a transfer between low earth orbit and geosynchronous orbit. Over the course of the entire transfer, the vehicle will experience several hundred occultation cycles resulting in a total trip time penalty of up to several weeks.

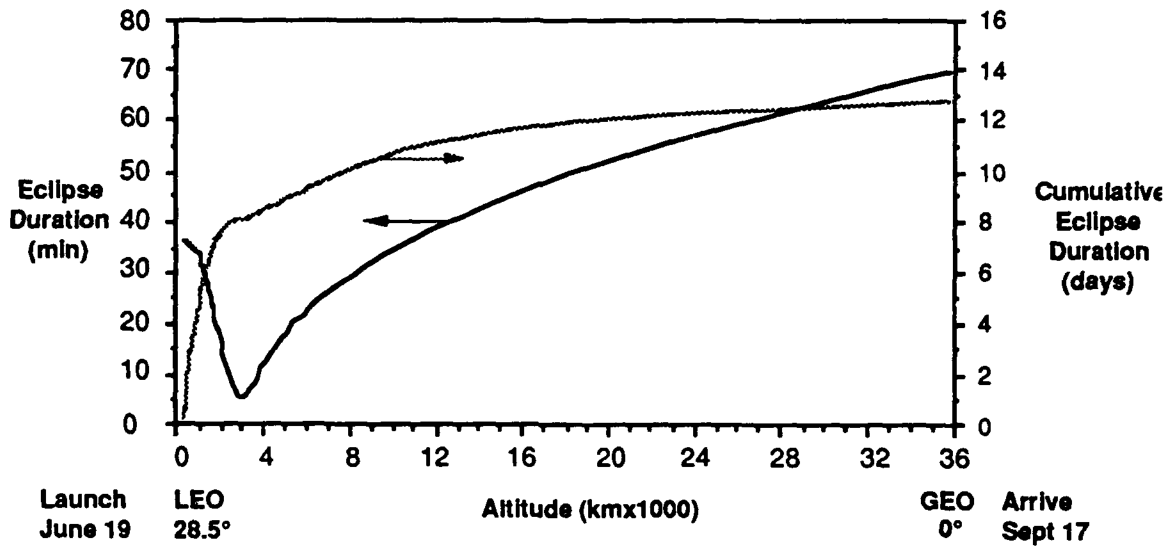


Figure 5
Typical Eclipse History for a LEO to GEO Transfer

One option available to overcome this phenomenon is to carry enough battery power to continue thrusting through the occultation periods. Conventional batteries could provide the power, however, their weight would be prohibitive.

Another factor to consider is the combination of atmospheric drag and occultation. The vehicle will undergo occultation even at low earth orbit where atmospheric drag is most severe. Depending on the size of the arrays, the amount of thrust generated by the thrusters, and the actual altitude of the low earth orbit, the vehicle's orbit could easily deteriorate to reentry because of drag during an occultation period.

Radiation. At mid altitudes (1,000 - 10,000 km) lie the Van Allen radiation belts. Consisting of trapped, high-energy electrons and protons, these regions can cause serious damage to photovoltaic solar arrays, and vehicle electronics. Because of the low thrust nature of EOTVs, the vehicle will spend many days in the Van Allen belts. The primary effect on the EOTV itself is the damage to solar arrays which manifests itself in the form of reduced power producing ability. As the power degrades, the thrusters will have to be throttled or shut down, which, in turn, will have an effect on the shape of the transfer trajectory and total transfer time. Shielding can be added to the solar arrays to minimize the radiation damage; however, a weight penalty is incurred.

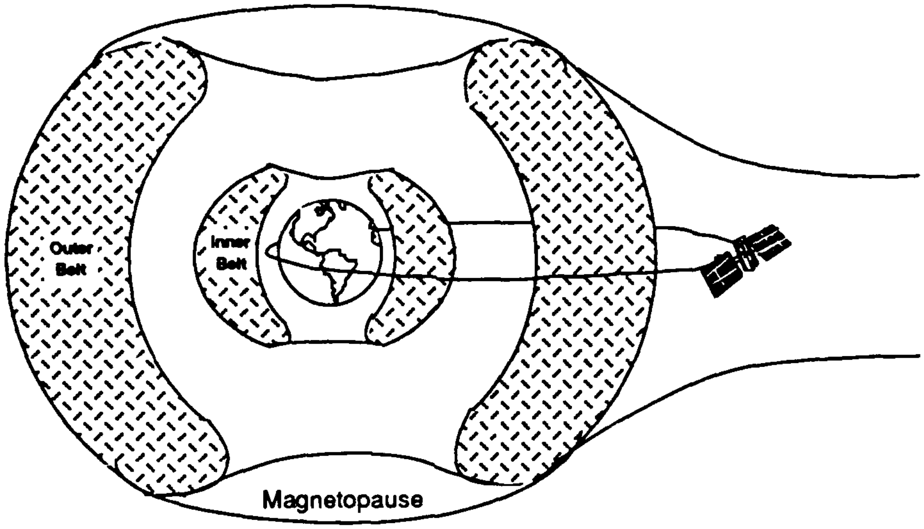


Figure 6
The Van Allen Radiation Belts

Each of these effects, while not necessarily important for conventional chemical OTVs, are very important to consider in the mission planning of electric OTVs. Any useful mission planning tool should address these factors to some degree.

Also of importance, especially for the ELITE mission planning tool, is the capability to model different electric propulsion systems and mission parameters. Preferably this would be accomplished by making quick and easy adjustments to an input data file. While very accurate evaluation of the various low-thrust effects presented above is desirable, a rigorous integration of complex orbit and attitude equations would slow the program down so much as to be undesirable for use in the PC environment. The goal was to have a code that would be as accurate as possible while maintaining run times of less than 30 seconds in a VAX environment and also be PC compatible.

Existing Tools

A survey of low thrust transfer codes currently in use within government agencies was conducted and two validated and well proven programs were identified: SPACEDRIVE and the Solar Electric Control Knob Setting Program for Optimal Trajectories (SECKSPOT).

SPACEDRIVE. An MS DOS-based program developed by Electric Propulsion Lab, Inc. of Lancaster, CA; is continuing to evolve as a versatile mission analysis and propulsion subsystem design tool.[4] In an effort funded by the Strategic Defense Initiative and managed by NASA/Lewis Research Center, the Electric Propulsion Lab developed this user-interactive software. The mission analysis routine in SPACEDRIVE was investigated for applicability to specific AL needs. Some simplifications employed by SPACEDRIVE such as the restriction to coplanar transfers, the absence of atmospheric drag, and the inability to model solar power degradation were seen as detractions to the tool's strong point of user friendliness. The particular iterative orbit transfer scheme used by SPACEDRIVE also made run times longer than was desired. For these reasons, SPACEDRIVE was not used for our orbital transfer mission planning program. However, we continue to use SPACEDRIVE for its extensive database and subsystem design model.

SECKSPOT. This program represents the other side of the analytical spectrum to SPACEDRIVE. Written in the early 1970's by MIT's Charles Stark Draper Laboratory, SECKSPOT optimizes steering control algorithms to minimize time of flight for low thrust propulsion systems.[5] SECKSPOT continuously updates the vehicle's orbital state and attitude in its analysis of the trajectory. Among other things, the program takes into account occultation, power system degradation, and earth oblateness. It represents one of the highest fidelity low thrust transfer programs available. The large number of differential and integral equations which must be solved, however, make it extremely unwieldy in a PC or even a VAX environment. SECKSPOT, for this reason, was determined to be inappropriate for the desired capability.

Analytical Gap. SPACEDRIVE and SECKSPOT represent the only two well documented and proven software tools available within government agencies. The mission planning capabilities of these programs represent two ends of the analytical spectrum. Unfortunately, the needs of the Astronautics Laboratory fall in the center of this "analytical gap". As such, it was determined that in-house propulsion, orbital mechanics, and software development expertise should be pooled to develop a tool for parametric mission studies which encompassed all of the capabilities discussed above. This program came to be known as the Electric Vehicle Analyzer.

ELECTRIC VEHICLE ANALYZER PROGRAM

Program Architecture

EVA was designed as a user-friendly, preliminary mission planning tool that would be able to model the effects of a low-thrust trajectory as accurately as possible while keeping run times on the order of seconds on a VAX and minutes on a PC. It was also designed to have the ability to model a variety of missions and types of electric propulsion systems. It was structured so that it could be easily upgraded to include more user-friendly features or more accurate mathematical models.

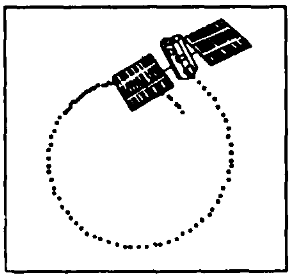
To accomplish the user-friendliness goal, the input data for EVA was separated and placed into a self-contained input file in namelist format. Any or all of the parameters could be easily changed in this file and then the program rerun. This method was chosen so as to maximize the number of input parameters that could be changed while at the same time eliminating the need to re-input the entire data set in a lengthy series of interactive questions at the beginning of each run. All of the mission parameters, electric propulsion system specifications, and program flags are contained in this input file. See Appendix A for a complete list of variables with definitions.

The main code is separated into several separate subroutines. The workhorse of the program is Subroutine Transfer which handles all of the orbital transfer calculations (Appendix A). The atmosphere model, the radiation tables, and the power degradation calculations are further separated into individual subroutines. The main reason for choosing this construction is to facilitate future upgrades in the individual mathematical models so they can be easily incorporated into the code.

Mathematical Model

The primary goal of the EVA orbital mechanics mathematical model was to describe the effects of low thrust trajectories as accurately as possible without taking a lot of CPU time. In order to accomplish this, some assumptions were made about the nature of the transfer process. These will be discussed along with the actual mathematical derivations in the following sections.

Altitude Raising. At the heart of EVA is the equation which determines the rate of change of the orbital radius. The effects of plane changing, atmospheric drag, power degradation, and other important perturbations hinge on this key equation. The assumption made here is that the vehicle is on a perfectly circular spiral trajectory. That is to say that at every altitude along the spiral, the vehicle trajectory is tangent to an equivalent circular orbit of that altitude. As mentioned earlier, this may not be completely accurate due to the occultation phenomenon, however, it has been estimated that the cumulative effect during a low earth to geosynchronous transfer would result in an elliptic geosynchronous orbit with an eccentricity of less than 0.2. Following the approach by Shepard[6], the radial rate equation can be derived as follows. The specific mechanical energy of the vehicle is defined as



$$E = -\frac{\mu}{2a}$$

where μ is the earth's gravitational parameter and a is the orbit semi-major axis. Based on the assumption that the spiral transfer orbit remains nearly circular at any given instant, the semi-major axis can be replaced by circular orbit radius, r, and the equation differentiated to give

$$\frac{dE}{dt} = \frac{\mu}{2a^2}\frac{da}{dt} \approx \frac{\mu}{2r^2}\frac{dr}{dt} \tag{2}$$

For tangentially applied forces, the rate of change of a vehicle's orbital energy is

$$m\frac{dE}{dt} = g_c FV \tag{3}$$

where m is the mass of the vehicle, F is the tangential force, V is the orbit velocity and g_c is a conversion factor. Combining Equations 2 and 3 gives,

$$\frac{dr}{dt} = 2g_c\left(\frac{r^2}{\mu}\right)\frac{F}{m}V \tag{4}$$

For a near circular orbit, the velocity can be approximated by

$$V \approx \sqrt{\frac{\mu}{r}} \tag{5}$$

Substituting Equation 5 into Equation 4 and rearranging gives the instantaneous rate of change of transfer orbit radius as a function of altitude and tangential acceleration. This is the key equation used for computing the spiral orbit altitude increase with time.

$$\frac{dr}{dt} = 2g_c\sqrt{\frac{r^3}{\mu}}\frac{F}{m} \tag{6}$$

Another useful piece of information regarding the transfer orbit is the instantaneous flight path angle. This can be directly derived from Equation 4.

$$\gamma = \frac{{}^{dr}\!/_{dt}}{V} = 2\, g_c \left(\frac{r^2}{\mu}\right)\frac{F}{m} \tag{7}$$

For constant accelerations, Equation 6 could be integrated directly to determine orbit radius as a function of time. Unfortunately, Equation 6 becomes non-linear when atmospheric drag, power system degradation, and thrust discontinuity due to eclipse are considered. Piecewise integration is used to solve Equation 6 over intervals in which the forces can be considered constant. The average altitude raising force over the course of the orbit is

$$F = T_P f_s - D \tag{8}$$

where f_S is the fraction of the orbit spent in sunlight with thrusters operating and T_P is the in-plane thrust component. F is comprised of thrust forces and (at low altitudes) drag forces—other perturbation forces are neglected. The calculation of f_S is discussed in the section on power system interactions. The computed value for dr/dt is then multiplied by the orbital period to find the change in altitude.

Orbital Period Adjustment. At low earth altitudes where the time rate of change of the spiral radius is small, the period is very close to the period of a circular orbit at that altitude. For larger values of dr/dt, however, the period will be somewhat longer, and a correction factor is introduced so that the transfer period can be more accurately determined. The derivation of the orbital period equation follows from Figure 7. Approximating to near circular orbits yields

$$\dot{\theta} \approx \frac{V}{r} \quad \text{and} \quad V \approx \sqrt{\frac{\mu}{r}} \tag{9}$$

The instantaneous radius during an orbit revolution can be approximated

$$r = r_1 + \dot{r}t \tag{10}$$

where $\dot{r}$ is the average calculated rate of change given by Equation 6 and r_1 is the radius at the start of the orbit. Combining Equations 9 and 10 yields

$$\dot{\theta} = \frac{\sqrt{\dfrac{\mu}{r_1 + \dot{r}\,t}}}{r_1 + \dot{r}\,t} = \frac{\sqrt{\mu}}{\left(r_1 + \dot{r}t\right)^{3/2}} \tag{11}$$

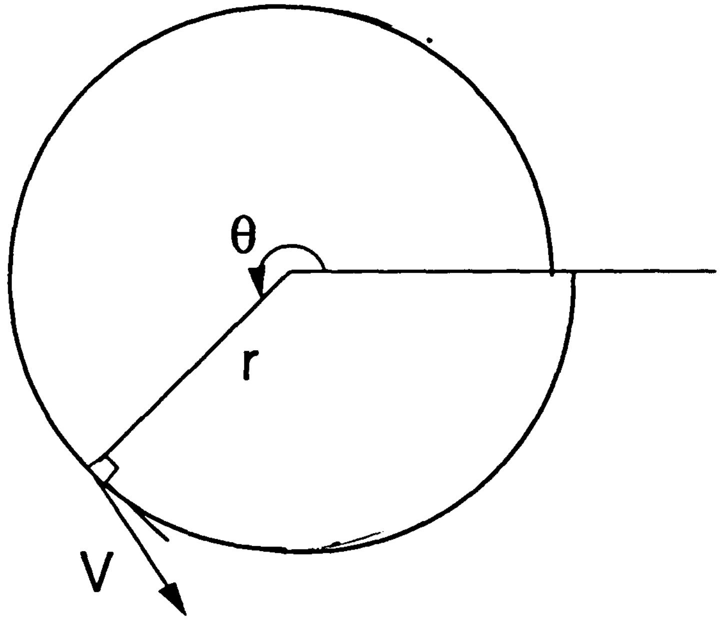


Figure 7
The Spiral Orbit Trajectory

Carrying out this integration over one orbit revolution

$$\int_{\theta=0}^{2\pi} d\theta = \sqrt{\mu}\int_{t=0}^{P_T} \frac{1}{\left(r_1 + \dot{r}\,t\right)^{3/2}}\,dt \tag{12}$$

yields

$$2\pi = \sqrt{\mu}\left[\frac{\left(r_1 + \dot{r}\,t\right)^{3/2}}{-\tfrac{1}{2}\,\dot{r}}\right]_0^{P_T} = \frac{-2\sqrt{\mu}}{\dot{r}}\left[\frac{1}{\sqrt{r_1 + \dot{r}\,TP}} - \frac{1}{\sqrt{r_1}}\right] \tag{13}$$

rearranging, gives the spiral transfer orbit period, P_T

$$P_T = \frac{r_1}{\dot{r}}\left\{\frac{\mu}{\left(\sqrt{\mu} - \pi\dot{r}\sqrt{r_1}\right)^2} - 1\right\} \tag{14}$$

with the constraint

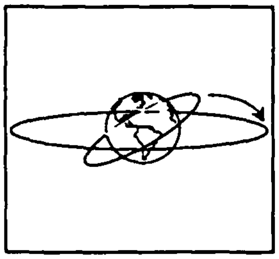
$$\dot{r} \neq 0 \tag{15}$$

The altitude change during an orbit revolution, from r_1 to r_2, can then be calculated by

$$r_2 = r_1 + \dot{r}P_T \tag{16}$$

from which V_2 can be found.

Inclination Changes. Since most satellites start out in a low earth orbit that is inclined to some degree, and since many orbital transfer missions have a destination of geosynchronous orbit at 0° inclination, any mission planning tool should have the ability to model inclination changes. This requirement introduces the need to use some thrust in an out-of-plane direction. This is readily handled by defining the desired thrust vector out-of-plane steering angle α. The total thrust available from the electric thruster devices can be computed knowing the available power, system efficiencies, and specific impulse.



$$\dot{m} = \frac{2\eta_t \eta_{ppu} P}{(g_o I_{sp})^2} \tag{17}$$

$$T = I_{sp}\, g_o\, \dot{m} \tag{18}$$

The normal thrust used for inclination changing, therefore, is given by

$$T_N = T \sin\alpha \tag{19}$$

where T is the total delivered thrust. While the available in-plane (altitude raising) thrust is simply

$$T_P = T \cos\alpha \tag{20}$$

Using the relationship developed by Edelbaum[7] for the low thrust delta-velocity requirement between two circular, non-coplanar orbits,

$$\Delta V = \sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos\left(\frac{\pi}{2}\Delta i\right)} \tag{21}$$

the incremental inclination change during an orbit revolution can be determined by solving for Δi.

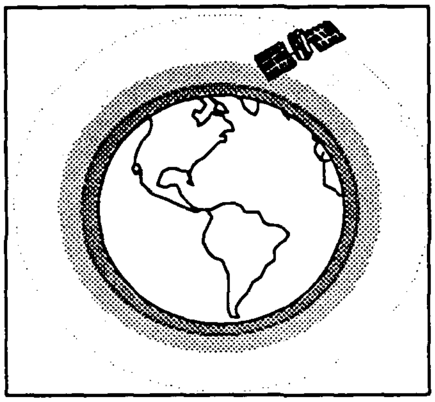
$$\Delta i = \frac{2}{\pi}\cos^{-1}\left[\frac{\left(V_1^2 + V_2^2 - \Delta V^2\right)}{2V_1V_2}\right] \tag{22}$$

The velocities at the start and end of the orbit revolution are determined from Equation 5 as discussed in the previous section. The ΔV can be calculated from the rocket equation

$$\Delta V = I_{sp}\, g_o \ln\left(\frac{M_I}{M_F}\right) \tag{23}$$

where the difference between M_I and M_F is the amount of propellant consumed during the last orbit which can be calculated knowing the characteristics of the propulsion system. It is important to note that once an inclination change is initiated, EVA maintains a constant α magnitude throughout the transfer. The modelling equations assume that the direction of α is reversed at the maximum and minimum orbital latitude positions (approximately 90° away from the nodes) to maintain inclination change in the same direction. If the desired inclination change is not achieved at the end of the transfer, the entire transfer is iterated using a secant method to converge on the appropriate value of α in order to meet the desired final inclination.

The code permits delaying out-of-plane thrusting until the vehicle reaches a specified altitude. This provides the option to "quickly" raise the orbital altitude beyond the majority of atmospheric interference and lengthy eclipses before diverting available thrust for an inclination change.



Atmospheric Drag. The drag force acting on the vehicle is computed using the familiar equation

$$D = C_d \bar{q} A \tag{24}$$

where Cd is the drag coefficient, A is the reference cross sectional area of the vehicle, and where $\bar{q}$ is the dynamic pressure given by

$$\bar{q} = \frac{1}{2} \rho V^2 \tag{25}$$

During an actual mission, the vehicle's solar arrays will be continuously tracking the sun and thus not be pointing at a constant angle with respect to the vehicle's velocity vector. This means that the vehicle's reference cross sectional area is constantly changing. As the arrays move from 90° to 0° with respect to the velocity vector, the reference area changes from 100% of the total cross sectional area to 0%. EVA calculates the worst case scenario where the reference area equals 100% of the total cross sectional area at all times. Circular orbital velocity at the time of the drag calculation is used to compute dynamic pressure. A self-contained subroutine, incorporating the 1976 U.S. Standard Atmosphere, is used to compute atmospheric density as a function of altitude.[8] This is a static atmosphere model. The drag coefficient is available as an input to the program. The drag force is then factored into Equation 6, effectively diminishing the available in-plane thrust.

The effects of atmospheric drag on the trajectory is assumed negligible above 1,000 kilometers altitude, and is not considered once the trajectory reaches that altitude. At the lower altitudes, the assumption is made that the vehicle's available in-plane thrust to drag ratio must exceed 1.1 in order to continue on the trajectory. That constraint is imposed by computing an effective thrust, T_{eff}, which is constrained to be greater than zero.

$$T_{eff} = 0.9 \cos\alpha \, F - D/f_s \tag{26}$$

where f_s represents the fraction of the orbit the vehicle is in sunlight and capable of providing thrust.

Power System Interaction

Occultation. To determine the amount of time during which the thrusters will not receive power from the power system, the fraction of the transfer vehicle's orbit spent in the earth's occultation zone must be calculated. The eclipse duration is a function of the date, which determines the earth-sun geometry, as well as the orbital parameters of the spacecraft. From spherical trigonometry and Figure 8,

$$\frac{\sin \delta}{\sin \varepsilon} = \frac{\sin[\omega_s (t - t_o)]}{\sin 90^\circ} \tag{27}$$

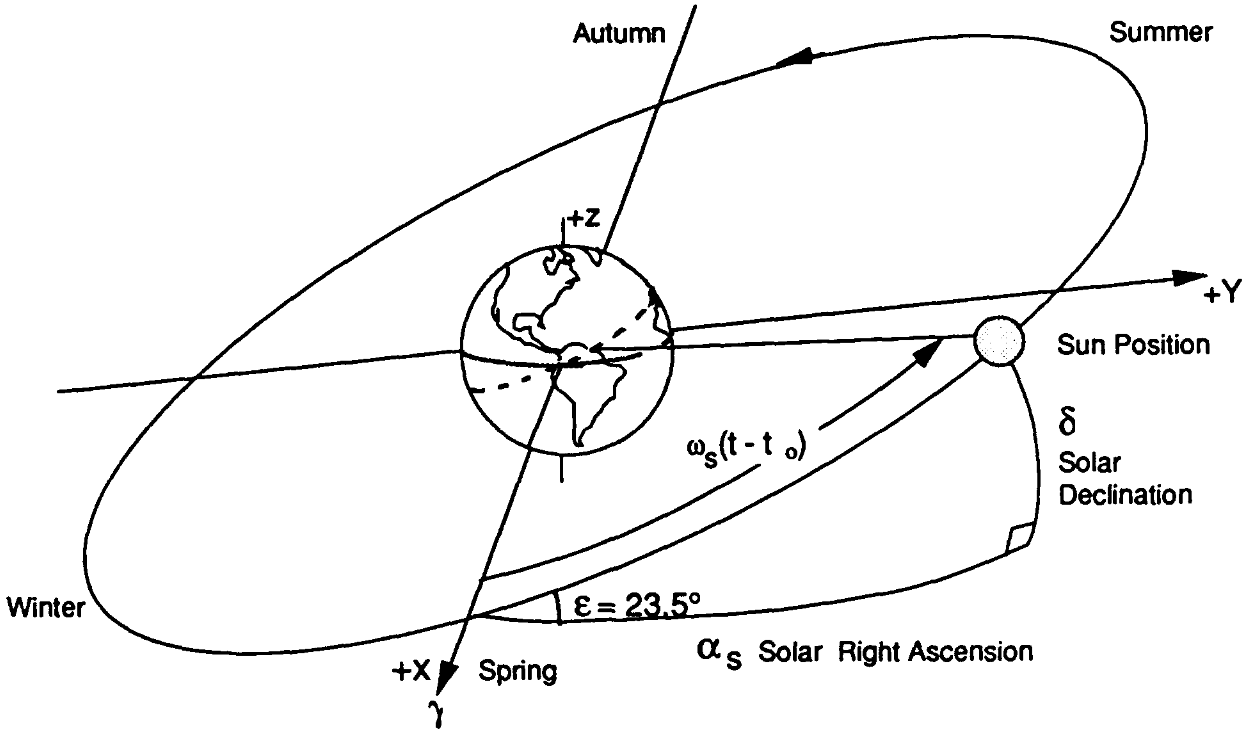


Figure 8
Right Ascension-Declination Coordinate System

$$\delta = \sin^{-1}\left\{\sin\varepsilon \sin[\omega_s(t - t_o)]\right\} \tag{28}$$

and

$$\cos\left[\omega_s(t - t_o)\right] = \cos\alpha \cos\delta + \sin\alpha \sin\delta \cos 90^\circ$$

$$\alpha = \cos^{-1}\left\{\frac{\cos[\omega_s(t - t_o)]}{\cos\delta}\right\} \tag{29}$$

Knowing α and δ and the spacecraft's orbital parameters, the angle between the sun and the orbital plane can be calculated.[9]

$$\beta = \sin^{-1}\left\{\cos\delta \sin i \sin(\Omega - \alpha) + \sin\delta \cos i\right\} \tag{30}$$

The spacecraft's right ascension of ascending node will slowly rotate during the transfer due to the earth's oblateness. This rotation will affect the beta angle; therefore, the eclipse duration. The regression of the node can be calculated using Equation 31.

$$\dot{\Omega} = J_2\left(\frac{R_E}{r}\right)^{3/2} \cos i \tag{31}$$

Using Figure 3 and defining the angle β^*, the fraction of the orbit spent in eclipse can be defined as in Equation 29. Note that the fraction of the orbit spent in sunlight (f_s) is simply 1- f_e. This fraction can then be incorporated into Equation 8 to calculate the orbit transfer possible with the given geometry.

$$f_e = \left\{ \begin{array}{ll} \frac{1}{180^\circ} \cos^{-1}\left[\frac{(h^2 + 2R_E h)^{\frac{1}{2}}}{(R_E + h)\cos\beta}\right] & |\beta| < \beta^* \\ 0 & |\beta| \geq \beta^* \end{array} \right\} \tag{32}$$

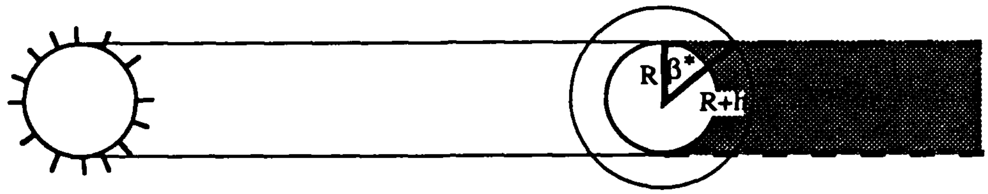


Figure 9
Sun-Earth Occultation Geometry

Solar Array Power Degradation. The slow spiral transfer through the Van Allen radiation belts can result in significantly degraded solar array power output and reduced vehicle performance. In conjunction with the orbit transfer calculations, the program automatically computes the cumulative fluence experienced by the vehicle and determines the resulting power degradation for the specified solar cell type. Logic within the program permits the degradation calculations to be bypassed as well as automatically terminating the run should power degrade to zero.

For each transfer orbit revolution, the annual equivalent 1 MeV electron fluences from trapped electrons and protons are determined from tabular fluence data stored in the program.[10] The data is tabulated for varying orbital altitudes, inclinations (up to 30°), and shield thicknesses (up to 60 mils). Linear interpolation is used to compute the fluence levels for specified shield thicknesses at the current transfer orbit altitude and inclination. The combined front and back side fluence experienced by the array cells is computed using Equation 33. Different shield thicknesses can be specified for front and back sides.

$$\phi_{combined} = \underbrace{[\phi_{electron} + \phi_{proton}]}_{\text{Front Side of Array}} + \underbrace{[\phi_{electron} + \phi_{proton})]}_{\text{Back Side of Array}} \tag{33}$$

The combined fluence (for the front and back sides of the solar array) is numerically integrated to yield the cumulative fluence which the vehicle arrays experience during the transfer. This cumulative fluence value is then translated into percent power degradation using tabulated data and logarithmic interpolation. The degraded power value is used in equations 17 and 18 to lower thrust and delivered I_{sp} as appropriate for the specific electric propulsion system design.

System Sizing. In addition to the orbital mechanics calculations, a mission planning tool needs calculations to determine weights of payload and necessary subsystems. The system sizing logic in EVA takes the actual EOTV dry mass and divides it into five major categories: arrays, structure, thermal management, propulsion system, and payload. The user is allowed to input specific mass (mass per unit power) relationships for the five categories in order to determine system weight. Also input are specific power (power per unit mass) and total system power level, which are used to calculate the necessary solar array surface area mandatory for drag calculations. Many different array types can be modelled by changing the specific power parameter.

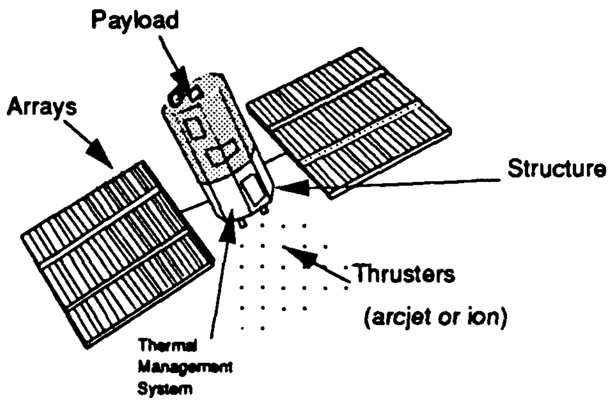


Figure 10
EOTV Subsystems

The propulsion system and power system masses are determined simply by multiplying their respective specific masses and the system power.

The thermal management system mass is dependent on the power processing unit (PPU) efficiency. The thermal management system must reject the power which the PPU cannot convert into usable thrust. Heat generated by thruster inefficiency is considered to be handled by direct radiation from the thruster to space and, hence, does not affect the mass of the thermal rejection hardware. The thermal management system mass (M_t) is determined by the following equation

$$M_t = \left(1 - \eta_{ppu}\right) * P_I * W_t \tag{34}$$

where P_I is the initial system power and W_t is the thermal management system specific mass. Structure mass is calculated using a fixed percent of the dry mass of the EOTV and its payload. This percentage is a user-defined entry in the input file—currently the default value is 12% which, after talking with several people in our in-house structure group, was decided upon as a good starting estimate.

The payload is everything not already accounted for in the above categories. It is determined by subtracting the sum of the above categories from the total EOTV dry mass.

Typical Problems

In its current form EVA can provide mission planners with useful, reasonably accurate answers to a wide variety of questions. Even without internal logic for paramaterization it is simple to generate a quick look of parametric effects on any given system. One important figure of merit, for example, is trip time for delivery of a satellite. EVA's flexibility allows the user to determine effects on trip time due to power levels, array protection, propulsion system efficiency, and drag, among others. It will also allow for the determination of optimal deployment altitudes–trading the increased throw weight by the launch vehicle to lower altitudes with the lower accelerations possible at these lower altitudes due to the more severe drag environment. The cumulative radiation fluence calculation also provides an opportunity to study the effects of radiation on the payload. The sizing logic allows the user to vary subsystem specific masses for subsystems such as arrays, power processors, thermal management, and structure and to understand the payload capability for any number of EOTV designs. This flexibility, coupled with the program's computational efficiency, make it an extremely useful preliminary mission analysis tool, capable of quickly studying a variety of missions. One such mission is shown in the following example problem.

A current, chemical OTV that is used on the Titan IV is the Inertial Upper Stage (IUS). It can deliver 2,268 kg to geosynchronous orbit using 12,429 kg of propellant and with a dry weight of 2,279 kg. How much of the propellant weight could be saved by using an EOTV to deliver the same 2,268 kg to the same geosynchronous orbit?

To use EVA to answer this, first go to the EVA input file. Check the initial and final orbit altitudes to make sure they reflect a LEO to GEO transfer. Also check to see that the inclination change and the altitude to start changing inclination are correct. The drag coefficient is defaulted to 2.0; if more accurate data is available, then this variable may be input directly. The day of the year desired to launch is also an input since it affects the amount of time spent in the earth's occultation zone. The default value is 80, corresponding to a vernal equinox launch, and need not be input unless a specific launch day is desired. Check the flags for power degradation and atmospheric drag. Enter the data for the specific type of EOTV you wish to run. For purposes of this example the values used are typical of an arcjet type electric thruster and are shown in Table 1. Appendix A contains the complete input variable list with corresponding definitions.

Table 1. Example Problem Inputs

Isp	1000	sec
system power	30	kW
vehicle cross sectional area (excluding solar arrays)	5	sq m
PPU efficiency	0.9	
thruster efficiency	0.38	
initial fueled vehicle mass	5,541	kg
specific mass of propulsion system	3.2	kg/kW
specific mass of thermal management system	18	kg/kW
structure fraction	0.14	

```
specific power..........................130   W/kg
power density...........................130   W/sq m
```

Now close the input file and run the program. After several seconds, the output file will be completed (see Appendix A for the full output listing).

Table 2. Partial Output Listing

```
            ELECTRIC PROPULSION MISSION SUMMARY

Structural mass:             431.20
Thermal management mass:      54.00
Propulsion system mass:       96.00
Power mass:                  230.77
                      811.97          *Total EOTV dry mass
  Payload:                  2268.03
                     3080.00         **Total Mass @ GEO
  Propellant:               2460.77
                     5540.77        ***Total Mass @ LEO
    MISSION:

    Initial Altitude (km):        300.
    Final altitude (km):        35775.
    Required Delta-V (m/s):   6104.76

    Time in sun    (days):     183.75
    Time in shadow (days):      36.94
    Total trip time(days):     220.69
    Total number of orbits:   1522.27
```

In answer to the problem, from the output we see that this EOTV would weigh approximately 812 kg, use 2,460 kg propellant, and carry 2,268 kg payload to geosynchronous orbit. The trip would take 221 days of which 37 days are spent in the earth's shadow (non-thrusting). This means that 9,970 kg of propellant could be saved by using an EOTV. This is important because the weight savings is enough to be able to launch the vehicle from an Atlas II rather than a Titan IV resulting in a potential savings of many millions of dollars.

A useful feature of the code is its ability to recognize the fact that atmospheric drag may be greater than the thrust of the vehicle. In such cases, the code will automatically raise the starting altitude up to such a height that thrust exceeds drag by at least 10%. Similarly, if during the vehicle's trek through the Van Allen radiation belts, the solar arrays become so damaged by radiation that the power produced falls to zero, the code will alert the user to that fact and terminate the run (Appendix A).

In The Future...

As previously mentioned, one of the objectives in creating EVA was to make it easy to update to include better models, and more user-friendly features. There are several upgrades on the horizon for future versions of the code. Two of these proposed features are automatic parametric runs and optimization.

Parametric studies are often essential in preliminary mission planning since many of the vehicle and mission variables are not defined. A useful feature of the EVA would be the capability to run a range of values of specific variables in a single run, and obtain a graph of the results (e.g. trip time vs system power). Of course, parametric runs are currently possible with EVA by simply editing the input file and changing the appropriate variable(s) and rerunning the code. This, however, is not as convenient as having the code do that automatically.

Optimization is another userful feature. Many of the variables directly impact other variables. For instance, more system power is desirable because it increases thrust, thus reducing trip time, but it also increases weight and drag which increases trip time. Within a certain range there is an optimum power level which minimizes trip time. An optimization routine would be useful in finding such optimal values.

SUMMARY/CONCLUSION

Recent activity in mission planning for EOTV systems and flight experiments such as ELITE, led to the need for a mission planning tool which incorporated several performance and system level parameters. Existing tools within government agencies did not adequately fulfill that need and, hence, the Electric Vehicle Analyzer was created. Although EVA does not model detailed attitude and flight control algorithms it will perform EOTV mission studies with enough accuracy to impact vehicle and mission design while running efficiently and quickly in a VAX or PC environment. The code allows the user a significant degree of flexibility in input parameters, allowing the capability to analyze many different missions performed by many different vehicles.

EVA has become a valuable part of the mission planning toolbox at the Astronautics Laboratory. It will continue to be updated and refined while always keeping an eye toward efficiency and user flexibility. The growing national interest in fielding EOTV systems ensure that the program's capabilities will be well exercised in the years to come. The development of the code focused on maintaining the flexibility to consider a number of types of missions and vehicles. The authors welcome any comments from the electric propulsion community as to things we may have missed or suggestions as to how the program can be made more responsive for specific applications.

REFERENCES

1. Penn, J.P. and Sponable, J.M., "Feasibility and Life Cycle Costing Assessment of an Electric Orbital Transfer Vehicle for Navstar Global Positioning System Satellite Delivery," Aerospace Report No. TOR-0086(6455-17)-1, June 1986.

2. Deininger, W.D., Vondra, R.J., "Electric Propulsion for Constellation Deployment and Spacecraft Maneuvering, AIAA paper 88-2833, presented at the AIAA/ASME/SAE/ASEE 24th Joint Propulsion Conference, Boston, Massachusetts, July 11-13, 1988.

3. Dickey, M.R., Matlock, R.S., and Feig, J.R., "ELITE: An On-orbit Demonstration of Electric Propulsion," AAS Paper No. 89-467, Presented to AAS/AIAA Astrodynamics Specialist Conference, Stowe, VT, August 10, 1989.

4. "SPACEDRIVE, Version 1.1, User's Guide., Electric Propulsion Laboratories, Inc., Lancaster CA.

5. Aston, M.B., Aston, G., and Brophy, J.R., "User Interactive Electric Propulsion Software Design," AIAA Paper No. 89-2376, Presented to the 25th Joint Propulsion Conference, Monterey, CA, July 12, 1989.

6. Shepard, Dennis G., Aerospace Propulsion, Elsevier Publishing Company, Inc., 1972.

7. Edelbaum, T.N., "Propulsion Requirements for Controllable Satellites," ARS Journal, August 1961. Presented at the ARS Semi-Annual Meeting, Los Angeles, CA, May 9-12, 1960.

8. Kwok, J., "Artificial Satellite Analysis Program," JPL External Memorandum 312/85-140, 1 April 1985.

9. Ginsberg, L. J., Orbit Planner's Handbook, August 18, 1976.

10. Anspaugh, B.E., Downing, R.G., Solar Cell Radiation Handbook, 3rd edition, Jet Propulsion Laboratory Publication 82-69, California Institute of Technology, Pasadena, California, November, 1982, Tables 6.6-6.17.

APPENDIX A

INPUT FORMAT

The input variables are entered into the program in the form of a namelist input file. This method allows the user to change any or all of these variables between runs in a timely fashion since changing the input file does not require any recompilation. Table 3 provides a definition of each input variable and its default value if any.

Table 3. Definition of Program Input Variables

ALPHA	Guess for out-of-plane burn angle used to change inclination (will be adjusted for INCLF), degrees
ALTI	Initial orbit altitude, km
ALTF	Final orbit altitude, km
ALTINCL	Altitude above which orbit inclination change is performed, km (ALTINCL must be set below ALTF if there is an inclination change.)
CD	Coefficient of Drag (Default value = 2.0)
DATE	Date during the year when the transfer is initiated in days from year start. (Default value = 80.0)
ENGEFF	Thruster efficiency
FAREA	Factor for adjusting array cross-sectional area in drag calculations to account for varying orientation with respect to the velocity vector. (Default = 1.00)
FDENS	Multiplication factor for atmospheric density term in drag computations. Used to approximate worst case drag conditions. (Default value = 1.00)
ICELL	Index specifying solar cell type for determining array power degradation (Input for IDEGRD = 1)
	1 - Si BBSFR Thin Cell, Gridded Back, 3.4 mil
	2 - ASEC OMCVD GaAs/Ge
IDRAG	Flag for including effect of atmospheric drag
	0 - No drag effects included
	1 - Drag effects included using 1976 Std. Atmosphere
IDEGRD	Flag ror including solar array power degradation effects
	0 - Neither cum fluence or array power degradation is computed during transfer
	1 - Cum fluence computed but array power not degraded; power output and thrust remain constant
	2 - Cum fluence computed and array power degraded
IPLOT	Flag for creating plot output file during run (Default = 0)
	0 - No plot output file created
	1 - Create plot file (Unit = 8) during run

IPRINT	Flag for type of output to be generated during run
	0 - Summary data only (Default)
	1 - Summary plus transfer trajectory
INCLI	Initial orbit inclination, degrees (Default = 28.5 deg)
INCLF	Final desired orbit inclination, degrees
ISP	Thruster specific impulse, sec
NPRINT	Orbit interval at which trajectory data will be output if IPRINT > 0. (Default value = 100)
NPLOT	Orbit interval at which trajectory data will be written to plot tape if IPLOT > 0. (Default value = 100)
OMEGAI	Right Ascension of Ascending Node at start of transfer, degrees. (Value will be adjusted during transfer to account for perturbation effects of equatorial bulge.)
PROPSYS	Specific mass of propulsion system, kg/kw
PPUEFF	Efficiency of the power processing unit
PWRDEN	Power density of solar arrays, W/m^2
PWRWGT	Specific power of solar arrays, W/kg
SCAREA	Spacecraft body cross-section excluding solar array area, m^2.
SCMASS	Initial fueled mass of spacecraft excluding solar arrays, kg
STRFRACT	Structural fraction of vehicle dry mass
SYSPWR	Initial system power provided by solar arrays, kW.(Remains constant if IDEGRD = 0) provided by solar arrays, kW
THMGMT	Specific mass of thermal management unit, kg/kW
THSTPWR	Power required per electric thruster, kW
TSHIELD(1)	Array front-side effective shield thickness for use in determining fluences, 0-60 mils (input for IDEGRD = 1)
TSHIELD(2)	Array back-side effective shield thickness for use in determining fluences, 0-60 mils (input for IDEGRD = 1)

OUTPUT FORMAT

The example problem output is shown below. The names of the variables output are printed at the top of the page as a header, and then the values of these variables are printed as a block at successive time intervals. At the end of this trajectory output, a summary of weights and propulsion requirements is printed.

Table 4. Example Problem Output

ELECTRIC VEHICLE ANALYZER (EVA) PROGRAM

TIME (DAYS)	ALT (KM) INCL (DEG) NUM ORBITS	VEL (M/SEC) RDOT (M/SEC) GAMMA (DEG)	TOT DV (M/S) PERIOD (HRS) BETA (DEG)	CUM FLUEN (MEV) FLUEN (MEV/YR) ECLIPSE FRACT	POWER DEGRAD POWER (KW) ATM DRAG (N)	TOT THRUST (N) ALPHA (DEG) PLN THRUST (N)	S/C MASS (KG) PROP MASS (KG) SPEC. IMP (SEC)
6.395	463.603	7632.83	133.53	0.1031D+11	1.00000	2.092555	5238.1854
	28.0919	0.3379	1.5643	0.1043D+13	30.0000	40.0362	71.8146
	100.00	0.00254	-19.1834	0.37509	0.1339D-01	1.602139	1000.00
13.060	671.077	7519.67	281.94	0.8070D+11	1.00000	2.092555	5159.5026
	27.6319	0.3790	1.6359	0.8493D+13	30.0000	40.0362	150.4974
	200.00	0.00289	-22.0584	0.34828	0.5978D-03	1.602139	1000.00
20.048	909.693	7395.54	444.16	0.3824D+12	1.00000	2.092555	5074.8578
	27.1210	0.4114	1.7196	0.2868D+14	30.0000	40.0362	235.1422
	300.00	0.00319	-6.6983	0.33825	0.7143D-04	1.602139	1000.00
27.414	1185.871	7259.27	622.21	0.14[illegible]3D+13	0.99718	2.086658	4983.5456
	26.5505	0.4594	1.818[illegible]	0.1002D+15	29.9155	40.0362	326.4544
	400.00	0.00363	15.4963	0.31184	0.00000	1.597791	1000.00
35.237	1521.700	7103.30	826.01	0.55[illegible]	0.98119	2.053197	4881.0393
	25.8842	0.5385	1.9405	0.3166D+15	29.4357	40.0362	428.9607
	500.00	0.00435	34.1077	0.24780	0.00000	1.572359	1000.00
43.643	1948.813	6918.73	1067.21	0.2024D+14	0.94982	1.987552	4762.4487
	25.0763	0.6305	2.0998	0.1104D+16	28.4946	40.0362	547.5513
	600.00	0.00523	40.5074	0.17969	0.00000	1.522378	1000.00
52.797	2465.812	6713.45	1335.47	0.7136D+14	0.90273	1.889020	4633.9303
	24.1520	0.6675	2.2983	0.3344D+16	27.0820	40.0362	676.0697
	700.00	0.00570	33.5985	0.18770	0.00000	1.447111	1000.00
62.851	3053.196	6501.04	1613.08	0.2255D+15	0.83926	1.756207	4504.5838
	23.1652	0.6846	2.5309	0.8290D+16	25.1779	40.0362	805.4162
	800.00	0.00604	21.5821	0.20917	0.00000	1.345664	1000.00
73.957	3724.050	6281.47	1900.05	0.5810D+15	0.76932	1.609848	4374.6742
	22.1106	0.7153	2.8056	0.1503D+17	23.0796	40.0362	935.3258
	900.00	0.00653	10.9755	0.21031	0.00000	1.233623	1000.00
86.329	4518.829	6048.06	2205.12	0.1199D+16	0.71050	1.486771	4240.6710
	20.9481	0.7754	3.1428	0.2045D+17	21.3151	40.0362	1069.3290
	1000.00	0.00735	4.7472	0.19769	0.00000	1.139169	1000.00
100.304	5513.106	5789.69	2542.86	0.2008D+16	0.66634	1.394358	4097.1034
	19.6075	0.8772	3.5822	0.2067D+17	19.9903	40.0362	1212.8966

	1100.00	0.00869	3.5645	0.17943	0.00000	1.068151	1000.00
116.452	6852.723	5488.77	2936.29	0.2796D+16	0.63803	1.335106	3935.9795
	17.9678	1.0536	4.2035	0.1447D+17	19.1408	40.0362	1374.0205
	1200.00	0.01102	6.1827	0.15691	0.00000	1.022549	1000.00
135.878	8881.845	5110.84	3430.57	0.3311D+16	0.62356	1.304829	3742.5053
	15.7744	1.3876	5.2047	0.5507D+16	18.7067	40.0362	1567.4947
	1300.00	0.01560	9.8067	0.12688	0.00000	0.999162	1000.00
161.259	12675.828	4573.79	4133.45	0.3466D+16	0.61966	1.296665	3483.6400
	12.3550	2.1385	7.2550	0.4148D+15	18.5897	40.0362	1826.3600
	1400.00	0.02691	9.8986	0.09456	0.00000	0.992790	1000.00
203.230	25005.406	3563.84	5458.91	0.3488D+16	0.61911	1.295525	3043.2086
	4.6179	5.2984	15.2626	0.1629D+15	18.5734	40.0362	2266.7914
	1500.00	0.08638	-4.0065	0.06193	0.00000	0.991914	1000.00
220.694	35775.000	3075.06	6104.76	0.3492D+16	0.61901	1.295311	2849.2350
	-0.0304	9.4586	6.4168	0.2532D+14	18.5703	40.0362	2460.7650
	1522.27	0.17761	-13.8908	0.00000	0.00000	0.991740	1000.00

ELECTRIC PROPULSION MISSION SUMMARY

THRUSTERS

Thrust/Thruster (N): 0.43
Quantity: 3
Total Thrust (N): 1.30
Specific Impulse (sec): 1000.

POWER:

Power Density (W/M**2): 130.00
Specific Power (W/kg): 130.00
Total Power (kW): 30.00
Excess Power (kW): 0.00
Array Area (M**2): 230.77
Array Mass (kg): 230.77
System Efficiency: 0.342

SPACECRAFT WEIGHT BREAKDOWN (kg):

Structural mass: 431.20
Thermal management mass: 54.00
Propulsion system mass: 96.00
Power mass: 230.77
811.97
*Total EOTV dry mass
Payload: 2268.03
3080.00
**Total Mass @ GEO
Propellant: 2460.77
5540.77
***Total Mass @ LEO

MISSION:

Initial Altitude (km): 300.
Final altitude (km): 35775.
Required Delta-V (m/s): 6104.76

Time in sun (days): 183.75
Time in shadow (days): 36.94
Total trip time(days): 220.69
Total number of orbits: 1522.27

A useful feature of the code is its ability to recognize the fact that atmospheric drag may be greater than the thrust of the vehicle. In such cases, the code will automatically raise the starting altitude To such a height that thrust exceeds drag by at least 10%.

Similarly, if during the vehicle's trek through the Van Allen radiation belts, the solar arrays become so damaged by radiation that the power produced falls to zero, the code will alert the user to that fact and terminate the run. Table 4 demonstrates this type of run.

Table 5. Low Altitude/High Power Degradation Case

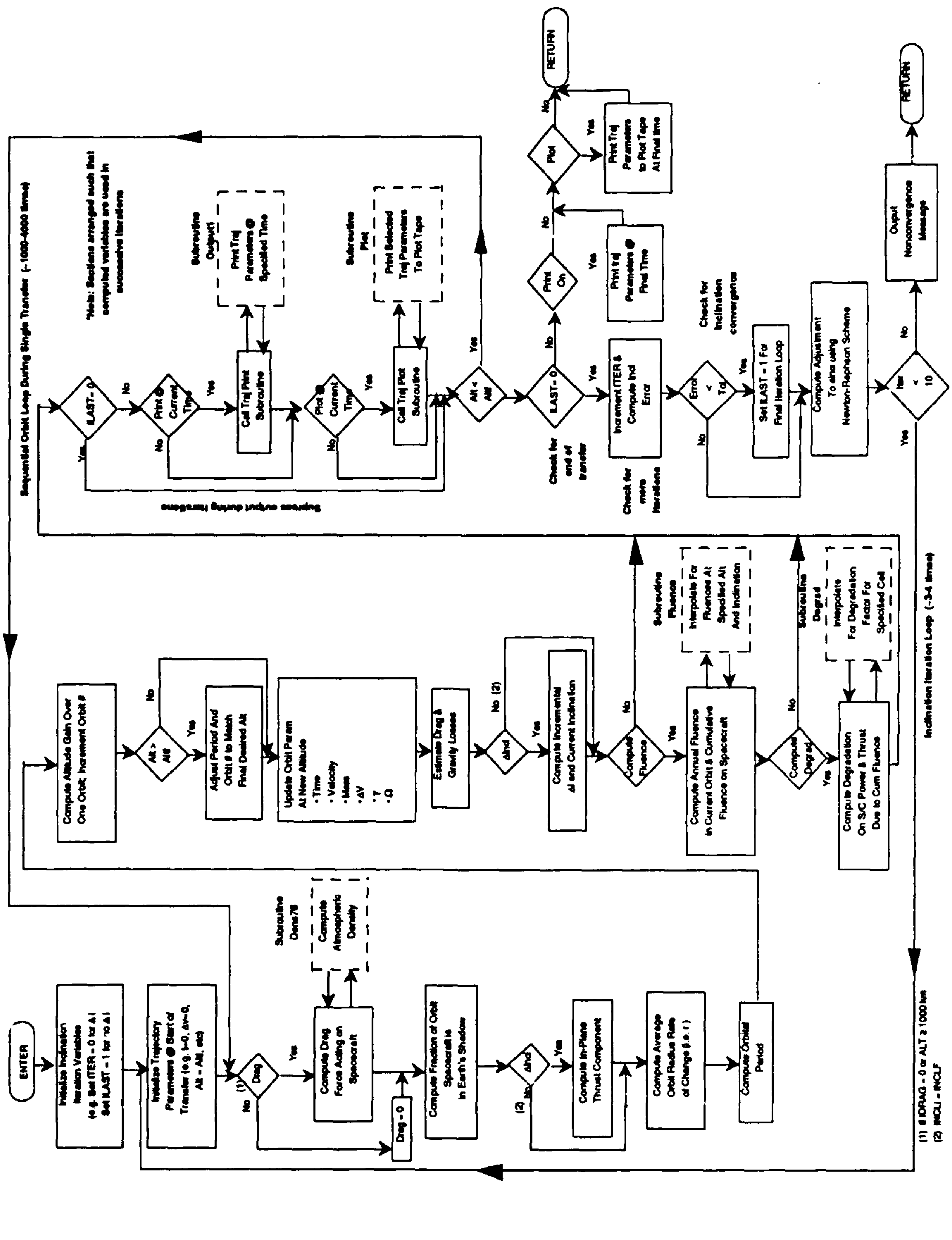
* DRAG EXCEEDED 90% OF IN-PLANE THRUST FOR INPUT PARAMETERS; STARTING ALTITUDE RAISED FROM 160.900 KM TO 260.900 KM

ELECTRIC VEHICLE ANALYSIS (EVA) PROGRAM

TIME (DAYS)	ALT (KM)	VEL (M/SEC)	TOT DV (M/S)	CUM FLUEN (MEV)	POWER DEGRAD	TOT THRUST (N)	S/C MASS (KG)
	INCL (DEG)	RDOT (M/SEC)	PERIOD (HRS)	FLUEN (MEV/YR)	POWER (KW)	ALPHA (DEG)	PROP MASS (KG)
	NUM ORBITS	GAMMA (DEG)	BETA (DEG)	ECLIPSE FRACT	ATM DRAG (N)	PLN THRUST (N)	SPEC IMP (SEC)
3.123	284.011	7735.03	24.44	-0.6639D+04	1.00000	2.007303	13084.4235
	0.0000	0.1042	1.5038	0.9641D+06	28.0000	45.0000	32.6565
	50.00	0.00077	1.1996	0.40679	0.3988D+00	2.007303	1000.00
6.267	316.646	7716.15	49.30	0.2133D+05	1.00000	2.007303	13051.2999
	0.0000	0.1333	1.5147	0.5628D+07	28.0000	45.0000	65.7801
	100.00	0.00099	2.4288	0.40173	0.1984D+00	2.007303	1000.00
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
888.759	11805.240	4682.00	3089.34	0.3353D+19	0.03145	0.063123	9572.3149
	0.0000	0.0460	6.7777	0.1758D+19	0.8805	45.0000	3544.7651
	5600.00	0.00056	9.4204	0.10184	0.0000D+00	0.063193	1000.00
902.912	11859.786	4675.00	3096.35	0.3420D+19	0.02974	0.059691	9565.4774
	0.0000	0.0433	6.8082	0.1735D+19	0.8326	45.0000	3551.6026
	5650.00	0.00053	4.0971	0.11152	0.0000D+00	0.059759	1000.00
914.013	11900.400	4669.80	3101.55	0.3473D+19	0.02843	0.057077	9560.4094
	0.0000	0.0414	0.4442	0.1717D+19	0.7962	45.0000	3556.6706
	5689.07	0.00051	-0.3410	0.11344	0.0000D+00	0.057081	1000.00

* ZERO POWER AVAILABLE FOR THRUSTERS AT 11900.446 KM DUE TO ARRAY DEGRADATION; RUN TERMINATED

SUBROUTINE TRANSFER FLOWCHART



APPENDIX B

PROGRAM EVA

```
      IMPLICIT DOUBLE PRECISION (A-M,O-Z)
      CHARACTER WEPTYP*(18),WPWRTYP*(50)
      INTEGER I,IS,IDRAG,IDEGRD,ICELL,IST(2),IPLOT,IPRINT
      INTEGER PWRTYP,EPTYP
      DIMENSION FLUEN(5),TSHIELD(2),TSHLDR(8),FST(2)
      COMMON/CONST/MU,SQMU,RE,PI,PI2,DTS,DTR,KTM,KTM3,G0,DYEAR,SEPSLN
      COMMON/FLAGS/IDRAG,IDEGRD,ICELL,IST,FST,IPRINT,IPLOT,NPRINT
      COMMON/TRAJ/T,DAYS,DATE,ALT,ALTI,ALTF,VEL,RDOT,INCL,INCLI,INCLF,
     *  ALTINCL,CALPH,OMEGAI,BETA,FE,CD,AREA,FDENS,GAMMA,ORBITS,PERIOD,
     *  FPWR,FLUEN,CFLUEN,THRUST,THRUSTI,THRUSTP,DRAG,ISP,ETA,DVTOT,
     *  DRGLOSS,GRVLOSS,MASSI,MASS,MPROP,SYSPWR,EPPWR,POWER,TECLPS,
     *  NUMTHST
      COMMON/MASS/STR,THMGMT,INEFF,PROPSYS,EPPWER,SASM,MEOTV,MPAY,AAREA,
     *  AMASS,XSPWR,PWRWGT,PWRDEN,SCMASS,PPUEFF,ENGEFF,STRFRACT
      COMMON/SUN/TSUN
C
C     Initialize Program Constants and Variables
C
      DATA MU,SQMU,RE/3.98600800D14, 1.9964989D7, 6.37813500D6/
      DATA PI,PI2,DTS/3.1415926535898D0, 6.2831853071796D0, 8.64D4/
      DATA DTR,KTM,KTM3/0.174532925199433D-1, 1000.D0, 1.D9/
      DATA G0,DYEAR,EPSILON/9.806194D0, 365.25D0, 23.4432D0/
      DATA IDRAG,IDEGRD,ICELL,IST,IPLOT,IPRINT/1,1,1,2*1,0,0/
      DATA DATE,FDENS,CD,INCLI,INCLF/80.D0,1.D0,2.0D0,2*28.5D0/
      DATA TSHLDR/0.D0,1.D0,3.D0,6.D0,12.D0,20.D0,30.D0,60.D0/
      DATA TSHIELD,FLUEN,FST,FAREA/2*0.D0,5*0.D0,2*0.D0,1.0D0/
      NAMELIST/EVAIN/ALTI,ALTF,INCLI,INCLF,ALTINCL,ALPHA,OMEGAI,DATE,
     *  IDRAG,SCAREA,CD,FDENS,FAREA,IDEGRD,TSHIELD,ICELL,SCMASS,SYSPWR,
     *  ISP,ENGEFF,PPUEFF,NTHSTR,THSTPWR,PWRDEN,PWRWGT,IPRINT,NPRINT,
     * IPLOT,NPLOT,THMGMT,PROPSYS,STRFRACT
      OPEN (UNIT=5, STATUS='OLD', FILE='INPUT')
      OPEN (UNIT=6, STATUS='NEW', FILE='EVAOUT')
      IF(IPLOT.EQ.1) OPEN (UNIT=8, STATUS='NEW', FILE='EVAPLOT')
C
C     Read Inputs and Perform Initial Computations
C
      READ(5,EVAIN)
      WRITE(6,EVAIN)
      INCLI = DTR*INCLI
      INCLF = DTR*INCLF
      CALPH = DCOS(DTR*ALPHA)
      SEPSLN = DSIN(DTR*EPSILON)
      ETA   = ENGEFF*PPUEFF
C
C     Determine No. of Thrusters and Total System Thrust at Mission Start
C
      NUMTHST = DINT(SYSPWR/THSTPWR)
      EPPWR = FLOAT(NUMTHST)*THSTPWR
      XSPWR = SYSPWR - EPPWR
      MDOT = (2.0D3*ETA*EPPWR)/(G0*ISP)**2.0D0
      THRUSTI = G0*ISP*MDOT
C
C     Determine S/C area and mass including arrays...
C
```

```
      ARRAY=1
      AMASS = 0.D0
      SASM=(1/PWRWGT)*1000.0D0
      MASSI = AMASS + SCMASS
      AAREA = SYSPWR*1000.D0/PWRDEN
      AREA  = AAREA + SCAREA
C
C     Determine Fluence Interpolation Factors for Specified Front and
C     Back-Side Array Shield Thicknesses (60 mils Maximum Thickness)
C
      DO 30 IS = 1,2
        DO 10 I = 2,8
          IF (TSHIELD(IS).LT.TSHLDR(I)) GOTO 20
   10   CONTINUE
        IST(IS) = 8
        FST(IS) = 0.D0
        GOTO 30
   20   IST(IS) = I - 1
        FST(IS) = (TSHIELD(IS) - TSHLDR(I-1))/(TSHLDR(I) - TSHLDR(I-1))
   30 CONTINUE
C
C     Check if In-Plane Thrust Exceeds Drag (over complete orbit) for
C     the Input Parameters and Raise Initial Orbit Altitude as Needed
C
      ALT = ALTI
      DO WHILE(TEFF.LE.0.D0)
        R = KTM*ALT + RE
        VEL2 = MU/R
        CALL DENS76(ALT,DENSITY)
        DENSITY = FDENS*DENSITY/KTM3
        DRAG = 0.5D0*CD*AREA*DENSITY*VEL2
C       Estimate Fraction Orbit is in Sunlight for Worst-Case Date
        FS = 0.5D0 + DACOS(RE/R)/PI
        TEFF = CALPH*THRUSTI - DRAG/FS
C       Compute Net Effective Thrust with 10% Minimum Margin over Drag
        TEFF = 0.9D0*CALPH*THRUSTI - DRAG/FS
        F(TEFF.LT.0.D0) ALT = ALT + 10.D0
      END DO
      IF(ALT.GT.ALTI) THEN
        WRITE(6,40) ALTI,ALT
   40   FORMAT(/,48H* DRAG EXCEEDED 90% OF IN-PLANE THRUST FOR INPUT,
     *   42H PARAMETERS; STARTING ALTITUDE RAISED FROM,F9.3,6H KM TO,
     *   F9.3,3H KM,/)
        ALTI = ALT
      END IF
C
C     Perform Spiral Orbit Transfer Analysis
C
      CALL TRANSFER
      CLOSE (UNIT=6)
      IF(IPLOT.EQ.1) CLOSE (UNIT=8)
      END
```

SUBROUTINE TRANSFER

```
C
C     Subroutine TRANSFER determines time and delta-V requirements for a
C     low-thrust spiral transfer from low earth orbit.  Includes effects
C     of atmospheric drag, earth shadow eclipses, solar array degradation,
C     and inclination change.  Tangential in-plane thrusting and constant
C     out-of-plane thrust angle are assumed during transfer. The transfer
C     orbit is also assumed to remain nearly circular at any instant.
C
      IMPLICIT DOUBLE PRECISION (A-M,O-Z)
      INTEGER IDRAG,IDEGRD,ICELL,IST(2),IPLOT,IPRINT,IFL,IAL,ICL
      INTEGER ITER,ILAST
      DIMENSION FLUEN(5),FST(2)
      COMMON/CONST/MU,SQMU,RE,PI,PI2,DTS,DTR,KTM,KTM3,G0,DYEAR,SEPSLN
      COMMON/FLAGS/IDRAG,IDEGRD,ICELL,IST,FST,IPRINT,IPLOT,NPRINT
      COMMON/TRAJ/T,DAYS,DATE,ALT,ALTI,ALTF,VEL,RDOT,INCL,INCLI,INCLF,
     *  ALTINCL,CALPH,OMEGAI,BETA,FE,CD,AREA,FDENS,GAMMA,ORBITS,PERIOD,
     *  FPWR,FLUEN,CFLUEN,THRUST,THRUSTI,THRUSTP,DRAG,ISP,ETA,DVTOT,
     *  DRGLOSS,GRVLOSS,MASSI,MASS,MPROP,SYSPWR,EPPWR,POWER,TECLPS,
     *  NUMTHST
      COMMON/MASS/STR,THMGMT,INEFF,PROPSYS,EPPWER,SASM,MEOTV,MPAY,AAREA,
     *  AMASS,XSPWR,PWRWGT,PWRDEN,SCMASS,PPUEFF,ENGEFF,STRFRACT
      COMMON/SUN/TSUN
      DATA ILAST,RE2/1,4.0680606D14/
C
C     Set Parameters for Inclination Iteration
C
      ITER = 0
      SALPH = DSQRT(1.D0 - CALPH*CALPH)
C     Set Flag to Inhibit Output Until Final Iteration Step
      IF(INCLF.NE.INCLI) ILAST = 0
C     Set Initial Point Values for Secant Iteration Scheme
      SALPH0 = 0.D0
      IDELT0 = INCLI - INCLF
C
C     Initialize Trajectory Variables at Start of Transfer
C
   10 ALT = ALTI
      R = KTM*ALT + RE
      T = 0.D0
      DAYS = 0.D0
      FPWR = 1.D0
      INCL = INCLI
      VEL2 = MU/R
      VEL  = DSQRT(VEL2)
      MASS = MASSI
      MPROP = 0.D0
      DVTOT = 0.D0
      OMEGA = OMEGAI
      POWER = SYSPWR
      THRUST = THRUSTI
      ORBITS = 0.D0
      CFLUEN = 0.D0
      TSUN  = 0.0D0
      TECLPS = 0.D0
      DRGLOSS = 0.D0
      GRVLOSS = 0.D0
```

```
C       Set Interpolation and Print Indices
        IFL = 2
        IAL = 2
        ICL = 3
        NPT = 1
        NPR = 0
        NPL = 0
C
C       Compute Drag Force Acting on Spacecraft
C
   20   IF((ALT.LT.1.D3).AND.(IDRAG.EQ.1)) THEN
          ALT = (R - RE)/KTM
          CALL DENS76(ALT,DENSITY)
          DENSITY = FDENS*DENSITY/KTM3
          DRAG = CD*AREA*0.5D0*DENSITY*VEL2
        ELSE
          DRAG = 0.0D0
        END IF
C
C       Compute Fraction of Orbit that Spacecraft is in Earth's Shadow
C
C       Compute Ascending Node Regression Rate (radians per day)
        OMEGDOT = -0.173903D0*(RE/R)**3.5D0*DCOS(INCL)
C       Compute Solar Declination and Right Ascension Angles for Current Date
        C1 = PI2*(DATE + DAYS - 80.D0)/DYEAR
        DELTA = DASIN(SEPSLN*DSIN(C1))
        RATS = DACOS(DCOS(C1)/DCOS(DELTA))
        IF(C1.GT.PI) RATS = -1.D0*RATS
C       Compute Beta Angle Between Orbital Plane and Sun Vector
        BETA = DASIN(DCOS(DELTA)*DSIN(INCL)*DSIN(OMEGA - RATS)
     *                                        + DSIN(DELTA)*DCOS(INCL))
        IF(DABS(BETA).LT.DASIN(RE/R)) THEN
          ALTM = ALT*1000.D0
          FE = DACOS(DSQRT(ALTM*ALTM + 2.D0*RE*ALTM)/(R*DCOS(BETA)))/PI
        ELSE
          FE = 0.0D0
        END IF
C
C       Compute In-Plane Thrust Component while Changing Inclination
C
        IF((INCLF.NE.INCLI).AND.(ALT.GE.ALTINCL)) THEN
          THRUSTP = CALPH*THRUST
          THRUSTN = SALPH*THRUST
        ELSE
          THRUSTP = THRUST
          THRUSTN = 0.D0
        END IF
C
C       Compute Average Radius Rate of Change
C
        R3MU = 2.0D0*DSQRT(R*R*R/MU)
        RDOT = R3MU*(THRUSTP*(1.0D0 - FE) - DRAG)/MASS
C
C       Compute Orbital Period with Spiral Orbit Adjustment if RDOT > 0.1 m/s
C
        IF(DABS(RDOT).GT.0.1D0) THEN
          PERIOD = (R/RDOT)*(MU/(SQMU - PI*RDOT*DSQRT(R))**2.D0 - 1.D0)
```

```
      ELSE
        PERIOD = 2.0D0*PI*R**1.5D0/SQMU
      END IF
C
C     Compute Altitude Increase over Orbit and Check Against Final Altitude
C
      DALT = RDOT*PERIOD/KTM
      ALT  = ALT + DALT
      IF(ALT.GT.ALTF) THEN
C       Adjust Period and Altitude to Match Desired Final Altitude
        C1 = KTM*(ALT - ALTF)/RDOT
        ORBITS = ORBITS + 1.D0 - C1/PERIOD
        PERIOD = PERIOD - C1
        DALT = DALT + ALTF - ALT
        ALT = ALTF
      ELSE
        ORBITS = ORBITS + 1.D0
      END IF
C
C     Update Orbital Parameters and Compute Total delta-V
C
      T   = T + PERIOD
      DAYS = DAYS + PERIOD/DTS
      R = KTM*ALT + RE
      VEL2OLD = VEL2
      VEL2 = MU/R
      VELOLD = VEL
      VEL = DSQRT(VEL2)
C     Compute Mean Flight Path Angle with respect to Local Horizontal
      GAMMA = 2.D0*(R*R/MU)*(THRUSTP*(1.0D0 - FE)  - DRAG)/MASS
C     Compute Propellant Mass Expelled during Previous Orbit
      DMASS = PERIOD*(1.0D0 - FE)*THRUST/(G0*ISP)
C     Compute delta-V for Previous Orbit and Cumulative delta-V
      DV = G0*ISP*DLOG(MASS/(MASS - DMASS))
      DVTOT = DVTOT + DV
C     Compute Cumulative Eclipse Time during Transfer
      TECLPS = TECLPS + FE*PERIOD
      TSUN = TSUN + (1.D0-FE)*PERIOD
C     Compute Total Propellant Expended and Current Spacecraft Mass
      MPROP = MPROP + DMASS
      MASS  = MASS  - DMASS
C     Compute Shift in Orbit's Ascending Node (radians)
      OMEGA = OMEGA + OMEGDOT*PERIOD/DTS
C
C     Estimate Cumulative delta-V Drag and Gravity Losses
C
C     Compute delta-V Needed for Same Orbit Change with No Drag
      IF(DRAG.GT.0.D0) THEN
        THRUSTD = THRUSTP - DRAG/(1.0D0 - FE)
        THRUST0 = DSQRT(THRUSTN*THRUSTN + THRUSTD*THRUSTD)
        DMASS0 = PERIOD*(1.0D0 - FE)*THRUST0/(G0*ISP)
        DV0 = G0*ISP*DLOG(MASS/(MASS - DMASS0))
      ELSE
        DV0 = DV
      END IF
      DRGLOSS = DRGLOSS + DV - DV0
      GRVLOSS = GRVLOSS + G0*RE2/(R*R)*GAMMA*PERIOD
```

```
C
C     Compute Inclination Change based on Relation in Edelbaum's Paper
C
      IF((INCLF.NE.INCLI).AND.(ALT.GE.ALTINCL)) THEN
        C1 = 2.D0/PI
        DINCL = C1*DACOS((VEL2OLD + VEL2 - DV0*DV0)/(2.D0*VEL*VELOLD))
        INCL = INCL - DINCL
      ELSE
        INCL = INCL
      END IF
C
C     Compute Cumulative Electron/Proton Fluence
C
      IF(IDEGRD.GE.1) THEN
        ALT1 = ALT - 0.5D0*DALT
        CALL FLUENCE(ALT1,INCL,FLUEN,IAL,ICL,IST,FST)
        CFLUEN = CFLUEN + FLUEN(5)*PERIOD/(DTS*DYEAR)
      END IF
C
C     Determine Array Power and Thrust Degradation due to Cum Fluence
C
      IF(IDEGRD.EQ.2) THEN
        CALL DEGRAD(CFLUEN,FPWR,ICELL,IFL)
        POWER = FPWR*SYSPWR
        F(POWER.GT.0.D0) THEN
C       (Future upgrade to include effect of Isp drop and logic for
C       possible step thrust decreases for multiple ion thrusters.)
        THRUST = FPWR*THRUSTI
      ELSE
        WRITE(6,30) ALT
   30   FORMAT(//,40H* ZERO POWER AVAILABLE FOR THRUSTERS AT ,F9.3,
     *  44H KM DUE TO ARRAY DEGRADATION; RUN TERMINATED,//)
        RETURN
      END IF
      END IF
C
C     Output Data at Specified Time Steps during Transfer
C
      IF(ILAST.EQ.0) GOTO 50
      IF(IPRINT.LE.0) GOTO 40
      NPR = NPR + 1
      IF(NPR.LT.NPRINT) GOTO 40
      NPR = 0
      CALL OUTPUT1(NPT)
C
   40 IF(IPLOT.LE.0) GOTO 50
      NPL = NPL + 1
      IF(NPL.LT.NPLOT) GOTO 50
      NPL = 0
      CALL PLOT(DAYS,NORBIT,ALT,INCL,DVTOT,CFLUEN,FPWR)
C
   50 IF(ALT.LT.ALTF) GOTO 20
      IF(ILAST.EQ.1) GOTO 70
C
C     Check Final Inclination Convergence and Adjust sin(ALPHA) Value
C
      ITER = ITER + 1
```

```
      IDELT = INCL - INCLF
C     Set Flag for Last Iteration Step if Inclin is within 0.5 deg of Goal
      IF(DABS(IDELT).LE.0.0087266463D0) ILAST = 1
C     Estimate Adjustment to sin(ALPHA) using Secant Method
      DALPH = IDELT*(SALPH - SALPH0)/(IDELT - IDELT0)
      SALPH0 = SALPH
      IDELT0 = IDELT
      SALPH = SALPH - DALPH
      CALPH = DSQRT(1.D0 - SALPH*SALPH)
      IF(ITER.LT.10) GOTO 10
      WRITE(6,60)
   60 FORMAT(/,51H*SECANT ITERATION FOR FINAL INCL FAILED TO CONVERGE)
   70 IF(IPRINT.EQ.1) CALL OUTPUT1(NPT)
      IF(IPLOT.EQ.1) CALL PLOT(DAYS,NORBIT,ALT,INCL,DVTOT,CFLUEN,FPWR)
C
      AMASS   = SYSPWR*1000.D0/PWRWGT
      STR     = STRFRACT * (SCMASS+AMASS-MPROP)
      INEFF   = (1.0D0-PPUEFF) * SYSPWR
      SMMEOTV = SASM + PROPSYS
      MEOTV   = ((SMMEOTV*EPPWR)+(THMGMT*INEFF))+STR
      MPAY    = SCMASS+AMASS-MPROP-MEOTV
C
C     Write Mission Data to Output File
C
      WRITE(6,350)
      WRITE(6,400) THRUST/FLOAT(NUMTHST),NUMTHST,THRUST,ISP
      WRITE(6,450) PWRDEN,PWRWGT,SYSPWR,XSPWR,AAREA,AMASS,ETA
      IF(IDECAY.GT.0) WRITE(6,500) ALTMIN
      WRITE(6,575) STR,THMGMT*INEFF,PROPSYS*EPPWR,SASM*EPPWR,MEOTV,
     +      MPAY,MEOTV+MPAY,MPROP,SCMASS+AMASS
      WRITE(6.550)ALTI,ALTF,DVTOT,TSUN/DTS,TECLPS/DTS,
     +      DAYS, ORBITS
C
C format statements...
C
  350 FORMAT(//,20x,'ELECTRIC PROPULSION MISSION SUMMARY')
  400 FORMAT(//,' THRUSTERS',
     +      //,' Thrust/Thruster (N):    ',2x,f17.2,
     +      /,' Quantity:               ',5x,i8,
     +      /,' Total Thrust (N):       ',2x,f17.2,
     +      /,' Specific Impulse (sec):',4X,F16.0)
  450 FORMAT(/,' POWER:',
     +      //,' Power Density (W/M**2):',2x,f7.2,
     +      /,' Specific Power (W/kg): ',2x,f7.2,
     +      /,' Total Power (kW):       ',2x,f7.2,
     +      /,' Excess Power (kW):      ',4x,f5.2,
     +      /,' Array Area (M**2):      ',1x,f8.2,
     +      /,' Array Mass (kg):        ',1x,f8.2,
     +      /,' System Efficiency:      ',5X,F5.3)
  500 FORMAT(/,' ORBIT TRANSFER WILL BEGIN AT',1x,f12.2,'(km)',/,
     +      IN ORDER TO OVERCOME THE DRAG FORCE')
  550 FORMAT(/,' MISSION:',
     +      //,' Initial Altitude (km): ',3x,f7.0,
     +      /,' Final altitude (km):    ',3x,f7.0,
     +      /,' Required Delta-V (m/s):',2x,f7.2,
     +      //,' Time in sun     (days): ',2x,f7.2,
     +      /,' Time in shadow (days): ',2x,f7.2,
```

```
     +        /,' Total trip time(days): ',2x,F7.2
     +        /,' Total number of orbits:',2x,F10.2)
 575  FORMAT(/,' SPACECRAFT WEIGHT BREAKDOWN (kg):',
     +        //,' Structural mass:          ',1x,f7.2,
     +        /,' Thermal management mass:',1x,f7.2,
     +        /,' Propulsion system mass: ',1x,f7.2,
     +        /,' Power mass:               ',1x,f7.2,
     +        /,19x,f8.2,'       *Total EOTV dry mass',
     +        /,'   Payload:            ',1x,f8.2,
     +        /,19x,f8.2,'      **Total Mass @ GEO',
     +        /,'   Propellant:         ',1x,f8.2,
     +        /,19x,f8.2,'     ***Total Mass @ LEO')
 600  FORMAT(/,' Time penalty due to drag:',1x,f7.2,' days',
     +        /,' Delta V penalty due to drag:',1X,F7.2,' m/sec')
C
      RETURN
      END
```

```
      SUBROUTINE DEGRAD(CFLUEN,FPWR,ICELL,IFL)
C
C     Subroutine DEGRAD computes solar array power degradation based on the cumulative electron and
C     proton fluence experienced during the orbit transfer. Normalized cell power data stored for
C     cumulative  1 MeV electron fluences of 1.E12, 3.E12, 1.E13, 3.E13, ... , 1.E16. Fluence values
C     stored as logarithms to facilitate interpolation.
C
      IMPLICIT DOUBLE PRECISION (A-M,O-Z)
      INTEGER I,IFL,ICELL
      DIMENSION LFLUEN(9), PWRNORM(9,8)
C     Reference Cumulative Fluences (Natural Logarithm Values) for which
C     Solar Cell Normalized Power Data is Input - 1E12 to 1E16 MeV
      DATA (LFLUEN(I), I = 1,9) /27.631021D0, 28.729633D0, 29.933606D0,
     * 31.032218D0, 32.236191D0, 33.348043D0, 34.538775D0, 35.637389D0,
     * 36.841361D0/
C     Normalized Max Power vs. Fluence for Si BBSFR Thin Cell, 3.4 mils;
C     "Solar Cell Radiation Handbook", Addendum 1, 15 Feb 89, Figure 18.
      DATA (PWRNORM(I,1), I = 1,9) /1.000D0, 0.992D0, 0.971D0, 0.938D0,
     *  0.889D0, 0.821D0, 0.726D0, 0.632D0, 0.530D0/
C     Normalized Max Power vs. Fluence for ASEC OMCVD GaAs/Ge Cell;
C     "Characterization of GaAs Solar Cells", JPL Pub. 88-39, Fig. 10.
      DATA (PWRNORM(I,2), I = 1,9) /1.000D0, 0.994D0, 0.976D0, 0.957D0,
     *  0.923D0, 0.874D0, 0.801D0, 0.691D0, 0.521D0/
      IF(CFLUEN.LE.1.D12) GOTO 30
C
C                             Interpolate for Normalized Power Degradation Factor
C
      LCFLUEN = DLOG(CFLUEN)
   10 IF((LCFLUEN.LE.LFLUEN(IFL)).OR.(IFL.EQ.8)) GOTO 20
      IFL = IFL + 1
      GOTO 10
   20 C1 = (LCFLUEN - LFLUEN(IFL-1))/(LFLUEN(IFL) - LFLUEN(IFL-1))
      FPWR = PWRNORM(IFL-1,ICELL) + C1*(PWRNORM(IFL,ICELL)
     *                                   - PWRNORM(IFL-1,ICELL))
      RETURN
   30 FPWR = 1.0D0
      RETURN
      END
```

```
      SUBROUTINE FLUENCE(ALT,INCL,FLUEN,IAL,ICL,IST,FST)
C
C   Subroutine FLUENCE computes annual equivalent 1 MeV Electron Fluence from Trapped
C     Electrons (Pmax and Voc) and Trapped Protons (Pmax and Voc) as a function of spacecraft orbital C
      altitude, inclination, and equivalent array front and back-side shield thicknesses. (Note: Fluence
C     data covers altitudes of 277 to 35,794 km, inclinations of 0 to 30 degrees, and 0 to 60 mils shield
C     thickness. Fluence data for higher inclination orbits maybe added by raising inclination index
C     beyond 4 and expanding tables.)
C
      IMPLICIT DOUBLE PRECISION (A-M,O-Z)
      INTEGER I,I1,I2,IS,IAL,ICL,IST(2)
      DIMENSION ALTR(34),INCLR(4),FLUEN(5),FST(2)
      DIMENSION EMAX(34,4,8),PMAX(34,4,8)
C   Reference Inclinations for Fluence Data, radians
      DATA (INCLR(I), I = 1,4)
     * / 0.0D0, 0.1745329252D0, 0.3490658504D0, 0.5235987756D0/
C   Reference Altitudes for Fluence Data, km
      DATA (ALTR(I), I = 1,34)
     * / 277D0, 463D0, 555D0, 833D0, 1111D0, 1481D0, 1852D0,
     *  2315D0, 2778D0, 3241D0, 3704D0, 4167D0, 4630D0, 5093D0,
     *  5556D0, 6482D0, 7408D0, 8334D0, 9260D0, 10186D0, 11112D0,
     * 12964D0, 14816D0, 16668D0, 18520D0, 20372D0, 22224D0, 24076D0,
     * 25928D0, 27780D0, 29632D0, 31484D0, 33336D0, 35794D0/
C   Annual Equivalent 1 MEV Electron Fluence from Trapped Electrons for
C   0 mils Shield Thickness and Orbit Inclinations of 0, 10, 20, 30 deg
      DATA ((EMAX(I1,I2,1), I1 = 1,34), I2 = 1,4)
     * /0.00D00, 1.14D07, 4.65D07, 8.03D09, 4.92D11, 7.25D12, 2.49D13,
     * 6.03D13, 9.80D13, 1.26D14, 1.46D14, 1.64D14, 1.73D14, 1.70D14, ...

(3 pages of numerical tables have been omitted here. See reference  for a complete listing of these tables)
C
C   Find Interpolation Coefficients for Increasing Alt/Decreasing Inclin
C
   10 IF((ALT.LE.ALTR(IAL)).OR.(IAL.EQ.34)) GOTO 20
      IAL = IAL + 1
      GOTO 10
   20 FALT = (ALT - ALTR(IAL-1))/(ALTR(IAL) - ALTR(IAL-1))
      IF((INCL.GE.INCLR(ICL)).OR.(ICL.EQ.1)) GOTO 30
      ICL = ICL - 1
      GOTO 20
   30 FINCL = (INCL - INCLR(ICL))/(INCLR(ICL+1) - INCLR(ICL))
C
C   Loop to Compute Fluences on Front (IS=1) and Back-Side (IS=2) of Array
C
      DO 50 IS = 1,2
C
C   Compute Trapped Electron Fluence (Pmax, Voc, Isc) as Function of
C   Orbital Inclination, Altitude, and Solar Array Shield Thickness
      I = IST(IS)
```

```
      CA1 = EMAX(IAL-1,ICL,I) + FINCL*(EMAX(IAL-1,ICL+1,I)
     *                                        - EMAX(IAL-1,ICL,I))
      CA2 = EMAX(IAL,ICL,I)   + FINCL*(EMAX(IAL,ICL+1,I)
     *                                        - EMAX(IAL,ICL,I))
      FLUEN(IS) = CA1 + FALT*(CA2 - CA1)
C     Bypass Interpolation Calculations if Ref. Shield Thickness Value Used
      IF(ABS(FST(IS)).LT.0.001) GOTO 40
      I = IST(IS) + 1
      CA1 = EMAX(IAL-1,ICL,I) + FINCL*(EMAX(IAL-1,ICL+1,I)
     *                                        - EMAX(IAL-1,ICL,I))
      CA2 = EMAX(IAL,ICL,I)   + FINCL*(EMAX(IAL-1,ICL+1,I)
     *                                        - EMAX(IAL,ICL,I))
      FLUEN(IS) = FLUEN(IS) + FST(IS)*((CA1 + FALT*(CA2 - CA1))
     *                                        - FLUEN(IS))
C
C     Compute Trapped Proton (Pmax and Voc) Fluence as a Function of
C     Orbital Inclination, Altitude, and Solar Array Shield Thickness
C
   40 I = IST(IS)
      I2 = IS + 2
      CA1 = PMAX(IAL-1,ICL,I) + FINCL*(PMAX(IAL-1,ICL+1,I)
     *                                        - PMAX(IAL-1,ICL,I))
      CA2 = PMAX(IAL,ICL,I)   + FINCL*(PMAX(IAL,ICL+1,I)
     *                                        - PMAX(IAL,ICL,I))
      FLUEN(I2) = CA1 + FALT*(CA2 - CA1)
C     Bypass Interpolation Calculations if Ref. Shield Thickness Value Used
      IF(ABS(FST(IS)).LT.0.001) GOTO 50
      I = IST(IS) + 1
      CA1 = EMAX(IAL-1,ICL,I) + FINCL*(EMAX(IAL-1,ICL+1,I)
     *                                        - EMAX(IAL-1,ICL,I))
      CA2 = EMAX(IAL,ICL,I)   + FINCL*(EMAX(IAL,ICL+1,I)
     *                                        - EMAX(IAL,ICL,I))
      FLUEN(I2) = FLUEN(I2) + FST(IS)*((CA1 + FALT*(CA2 - CA1))
     *                                        - FLUEN(I2))
   50 CONTINUE
C
C     Compute Combined Fluence on Front and Back Sides of Array
C
      FLUEN(5) = FLUEN(1) + FLUEN(2) + FLUEN(3) + FLUEN(4)
      RETURN
      END
```

```
      SUBROUTINE PLOT(DAYS,NORBIT,ALT,INCL,DVTOT,CFLUEN,FPWR)
      IMPLICIT DOUBLE PRECISION (A-M,O-Z)
      DATA DTR/0.17453292519943D0/
C
C     Write Trajectory Data to Plot File 8
C
      INCLD = INCL/DTR
      WRITE(8,100) DAYS,ORBITS,ALT,INCLD,DVTOT,FPWR,CFLUEN
  100 FORMAT(6F12.4,D12.4)
      RETURN
      END
```