N70-24892

NASA TECHNICAL MEMORANDUM

NASA TM X-2004

NASA TM X-2004

CASE FILE COPY

A COMPUTER PROGRAM FOR QUICKLY ANALYZING ELECTRIC PROPULSION MISSIONS

by Alfred C. Mascy

Office of Advanced Research and Technology
Mission Analysis Division
Moffett Field, Calif. 94035

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1970

1. Report No. NASA TM X-2004	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A COMPUTER PROGRAM FOR QUICKLY ANALYZING ELECTRIC PROPULSION MISSIONS		5. Report Date April 1970
		6. Performing Organization Code
7. Author(s) Alfred C. Mascy		8. Performing Organization Report No. A-3482
9. Performing Organization Name and Address NASA, Office of Advanced Research and Technology Mission Analysis Division Moffett Field, California 94035		10. Work Unit No. 130-06-04-07-15
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D. C., 20546		13. Type of Report and Period Covered Technical Memorandum
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program is described that is capable of determining the performance and system requirements of electrically propelled spacecraft in combination with specific launch vehicles and high-thrust upper stages. The formulation of the logic and optimization techniques are described as well as the functional relationships that define the characteristics of the high- and low-thrust systems. The several output formats, including a plot option, are illustrated, and complete descriptions of all input and output parameters and a program listing in Fortran IV are given. The input is simplified by use of colloquial variables. Example problems are provided which depict the usage of the various options available to the user. These options include:

Planet orbiters or flybys
Launch to parking orbit or direct to escape
Built in stable of launch vehicles or specified by input
High- or low-thrust Earth departure
High- or low-thrust planet arrival
Optimum or constrained power
Optimum or constrained thrust time
Optimum or constrained hyperbolic velocities
Optimum ($\alpha = f(P)$) or constrained propulsion system specific mass
All-ballistic high-thrust comparisons
Three output formats, including graphical

The program is quite accurate in simulating entire missions and can define their requirements very quickly due to the short execution times, which range from 0.1 second to 0.5 second, on an IBM 360-50 (0.05 to 0.25 on IBM 7094, 0.02 to 0.1 on IBM 360-75) depending on the option selected. Convergence is guaranteed.

17. Key Words Suggested by Author Electric propulsion Computer program Performance analysis	18. Distribution Statement Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	51	$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

TABLE OF CONTENTS

A COMPUTER PROGRAM FOR QUICKLY ANALYZING ELECTRIC PROPULSION MISSIONS

Alfred C. Mascy

Office of Advanced Research and Technology
Mission Analysis Division
Moffett Field, California 94035

SUMMARY

A computer program is described that is capable of determining the performance and system requirements of electrically propelled spacecraft in combination with specific launch vehicles and high-thrust upper stages. The formulation of the logic and optimization techniques are described as well as the functional relationships that define the characteristics of the high- and low-thrust systems. The several output formats, including a plot option, are illustrated, and complete descriptions of all input and output parameters and a program listing in Fortran IV are given. The input is simplified by use of colloquial variables. Example problems are provided which depict the usage of the various options available to the user. These options include:

Planet orbiters or flybys
Launch to parking orbit or direct to escape
Built-in stable of launch vehicles or specified by input
High- or low-thrust Earth departure
High- or low-thrust planet arrival
Optimum or constrained power
Optimum or constrained thrust time
Optimum or constrained hyperbolic velocities
Optimum ($\alpha = f(P)$) or constrained propulsion system specific mass
All-ballistic high-thrust comparisons
Three output formats, including graphical

The program is quite accurate in simulating entire missions and can define their requirements very quickly due to the short execution times, which range from 0.1 second to 0.5 second, on an IBM 360-50 (0.05 to 0.25 on IBM 7094, 0.02 to 0.1 on IBM 360-75) depending on the option selected. Convergence is guaranteed.

INTRODUCTION

As the national space program progresses, there is growing interest in performing missions that have greater propulsive energy requirements than those performed to date. One method of accomplishing such missions is through the use of electric propulsion. The analysis of the performance and system requirements for this type of advanced propulsion has in the past centered on detailed trajectory studies (ref. 1). The computation of optimized low-thrust trajectories is complicated by the requirement for integration of the equations of motion and the solution of the subsequent boundary-value problem with concomitant optimization of the system parameters. More than 50 attempts have been made over the years to develop low-thrust trajectory and mass-computation programs that ease this computer-time-consuming problem (ref. 2).

The slow execution speeds of most of these programs have excluded their use in investigating wide ranges of variables necessary to identify commonality in mission and system characteristics. Many programs are quite inflexible and do not allow study of interesting options such as constrained power level and contrained thrusting time, or various departure and arrival modes. The program described in this paper evolved from an effort to produce a useful low-thrust mission-analysis tool of acceptable accuracy and compute time that would be applicable to a range of problems.

The computer program defines the performance and system requirements of electrically propelled unmanned planet-orbiter and flyby missions using existing launch vehicles for the Earth launch phase, and high-thrust upper stages or low-thrust spiral maneuvers for Earth-departure and planet-arrival phases. The characteristics of the launch vehicles and high-thrust stages may be specified in lieu of the built-in values. The electric propulsion system may be completely optimized, or may be constrained in power level, thrusting time, propulsion system specific mass, or departure and arrival velocities. Rather than integrate the low-thrust trajectory, functional relationships for the energy requirements of precomputed optimum trajectories obtained from accurate computer programs are stored within the code (refs. 3, 4). Curve-fitting procedures have been used in defining the energy parameters as a function of time and hyperbolic excess velocity at Earth departure and planet arrival. A method of system optimization based on the near invariance of certain parameters with system variables was found to be quite accurate. Low-thrust and high-thrust planetocentric operations are expressed analytically, and their velocity is matched with the heliocentric phase. Correlation with exact trajectory data is excellent, and the computer times are less than a second per fully optimized case.

Most important are the fail-safe and user-convenience features of the code. Convergence is assured on any case that has a solution. On all other cases, the code repairs any damage to its logic and proceeds to the next input case. This facilitates the running of numerous cases with large ranges in parameters. Also, much effort has been expended in developing the program with the lay user in mind. The input has been simplified through the use of colloquial variables such as the proper names of launch vehicles and planets, and the straightforward spelling of parameters to indicate their function such as MODE = FLYBY, ARRIVE = HIGH, LAUNCH = ESCAPE. The Fortran IV program coding has been kept relatively simple so that the logic flow may be followed easily and changed to suit a user's particular needs. The program is being sent to the regional dissemination center, COSMIC, located at the University of Georgia, for general availability.

ANALYSIS

The definition of the performance and system requirements of an unmanned interplanetary space mission involves the apportionment of stage masses at each phase such that maximum payload may be delivered for a given launch weight and given constraints. The problem complexity increases when one of these stages is electrically propelled, for it is then necessary to properly mate both high- and low-thrust systems having markedly different characteristics. The optimization of the various stage and system parameters has generally required many iterations involving time-consuming low-thrust trajectory integration. To provide a computational tool for electric-propulsion mission analysis of sufficient speed to allow broad coverage of cases, the low-thrust trajectories have been precomputed and stored within the program described herein. The data, ready for instant recall, is stored in the form of functional relationships between the trajectory parameters $J = \int a^2 dt$, coast time, operating time, and initial and final velocities. Since J is a good indicator of energy requirements, the minimization of this parameter over the planetocentric and heliocentric phases will yield the optimum apportionment of the operating times within these phases. The energy parameter J is heavily time-dependent and is additive over the phases:

$$J_T = J_D + J_H + J_C \tag{1}$$

where

$$J_T = f(T_T)$$

$$J_D = f(T_D)$$

$$J_H = f(T_H)$$

$$J_C = f(T_C)$$

and subscripts:

T   total

D   departure

H   heliocentric

C   capture

The program thus minimizes the summation of J_T while seeking the best division of the total mission time among the various phases. The description of this problem solution will proceed in the order in which the code handles each phase.

Launch

Since most analyses of unmanned interplanetary missions begin on the launch pad, the characteristics of a stable of 11 presently conceived or operational launch vehicles have been built into the program. The characteristics of these vehicles, an example of which is shown in figure 1, are stored in 16 valued tables of PAYUP (payload mass of vehicles, kg) versus VC (characteristic velocity of vehicle, km/sec). The stored values of the selected launch vehicles may be found in subroutine DPART. If the analyst desires to use a vehicle that is not in storage, he need only input the tabular values of PAYUP and VC (a maximum of 16 values each). The convention adopted in this study is that all launch vehicles attain at least low circular orbit speed so that the initial values in the tables should correspond to low Earth orbit conditions.

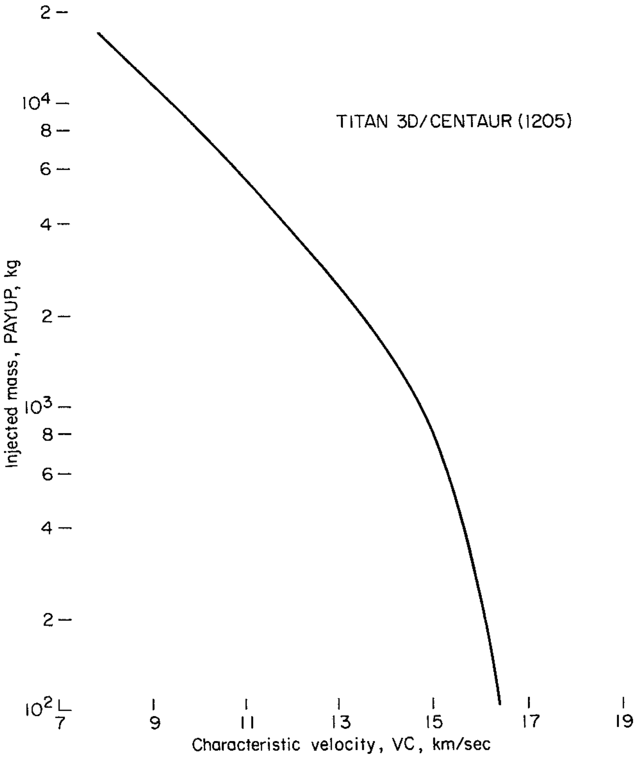


Figure 1.- Example of built-in launch vehicle characteristics.

If the launch vehicle is to place its payload into a parking orbit (LAUNCH = PARK), the input parameters of the parking orbit (RP1 and EPSD) are used to calculate the required characteristic velocity, VC. Since the performance of the vehicles are stored with minimum requirements starting in low Earth orbit, all launch velocities computed internally are additive to circular velocity, 7.75 km/sec, at 185-km (100-n.mi.) altitude. The velocity requirement to transfer from the 185-km circular orbit to the trajectory that will coast to the specified orbit perigee (RP1) is given by:

$$V1 = \sqrt{\frac{GM}{A_{12}}\left(\frac{1 + \varepsilon_{12}}{1 - \varepsilon_{12}}\right)} - 7.75 \qquad (2)$$

where

GM 39.86 (10^4 km^3/sec^2)

A_{12} semimajor axis of transfer orbit $= \frac{RG}{2}(RP1 + 1.029)$

ε_{12} eccentricity of transfer orbit $= \frac{RP1 - 1.029}{RP1 + 1.029}$

The velocity requirement to establish the desired parking orbit at the radius RP1 is given by:

$$V2 = \sqrt{\frac{GM}{A_D}\left(\frac{1 + EPSD}{1 - EPSD}\right)} - \sqrt{\frac{GM}{A_{12}}\left(\frac{1 - \varepsilon_{12}}{1 + \varepsilon_{12}}\right)} \tag{3}$$

where

EPSD eccentricity of desired orbit

A_D semimajor axis of desired orbit $= \frac{(RG)RP1}{1.0 - EPSD}$

RG radius of Earth

The total velocity required of the launch vehicle is:

$$VC = 7.75 + V1 + V2$$

The code then enters the tabular values of VC using a second-order interpolation to determine exact values of launch vehicle payload, BOOSTL.

If the launch vehicle is to place its payload unto an escape trajectory (LAUNCH = ESCAPE), the required velocity is simply:

$$VC = \sqrt{(VINF1)^2 + 2(7.75)^2} \tag{4}$$

where VINF1 is the departure velocity either constrained by input of VA or left for program optimization. The code then determines the payload (BOOSTL) from the appropriate launch vehicle tabular values.

Depart

When the launch vehicle is used to place its payload into a parking orbit (LAUNCH = PARK), the user should indicate his choice of departure stage thrust level by input. Departure from orbit via a high-thrust rocket (DEPART = HIGH) requires the calculation of the energy and performance based on the stage and orbital characteristics. The velocity increment required of the system is:

$$\Delta V = \sqrt{(VINF1)^2 + \frac{2(GM)}{RP1(RG)}} - \sqrt{\frac{GM}{A_D}\left(\frac{1 + EPSD}{1 - EPSD}\right)} \tag{5}$$

The payload ratio of the high-thrust system is given by:

$$DEPL = \frac{BOOSTL - WFUEL - WINERT}{BOOSTL} \tag{6}$$

where

WFUEL $\left\{1 - \exp\left[\frac{-\Delta V}{DISP(0.00981)}\right]\right\}$ BOOSTL

WINERT DINERT + DSIGMA[WFUEL]

DINERT input fixed stage weight

DSIGMA input tankage fraction

DISP input specific impulse

For internal accounting purposes, the high-thrust departure payload ratio is set equal to 1 whenever LAUNCH = ESCAPE, since the departure stage is part of the launch vehicle.

With DEPART = LOW, the code will simulate a low-thrust sprial escape of Earth from the designated parking orbit. The method of describing the spiral escape maneuvers uses expressions developed by Edelbaum (ref. 5) on the basis of the work of Breakwell and Rauch (ref. 6), and considers the asymptotic matching of the planetocentric and heliocentric trajectories that are under the influence of both the Sun and the Earth. The low-thrust characteristic velocity increment under optimal steering during planet escape is given by:

$$\Delta V = V - 1.84V \left[\frac{A_O A_D A_D}{(GM)\mu_1}\right]^{1/4} \tag{7}$$

where

A_O initial acceleration $= \dfrac{C(1 - \mu_1)}{T_D}$

V parking orbit velocity

μ_1 departure phase mass ratio

C exhaust velocity of system

T_D departure time

The low-thrust system is assumed to operate continuously during the spiral escape, therefore, T_D is the powered time. The final mass ratio for this maneuver is:

$$\mu_1 = \exp\left(\frac{-\Delta V}{C}\right) \tag{8}$$

and the energy parameter $J = \int a^2 dt$ for constant-thrust planet departure is given by:

$$J_D = \left(\frac{A_o^2}{\mu_1}\right) T_D \tag{9}$$

from which it follows that for a given orbit J_D is simply a function of C and T_D. Further, it can be shown that the influence of exhaust velocity on J_D is very slight and is herein calculated for a fixed value of C. Hence, the departure phase is described by:

$$J_D = f(T_D)$$

$$T_D = \text{departure time} = \text{powered time}$$

which will be used in the minimization of total J_T for the electric-propulsion system optimization. Again, for internal accounting purposes, the high-thrust departure payload ratio is set equal to 1 whenever DEPART = LOW.

Mode

The low-thrust heliocentric phase is the next stage of the analysis and may be either a flyby (MODE = FLYBY) or an orbiter (MODE = ORBIT). Under the flyby mode, the spacecraft is assumed to traverse an optimum heliocentric travel angle and to pass within the vicinity of the target planet with an unconstrained approach velocity. Orbiter spacecraft are assumed to traverse an optimum travel angle and to apply some braking propulsion such that a useful payload may be placed in a specified orbit about the target planet. To avoid the time-consuming problem of trajectory integration at each step of the optimization within this program, the low-thrust trajectories for a range of mission times and initial and final velocities have been precomputed using accurate programs (see fig. 2) and the data have been stored in the form of the following relationships:

$$\ln J_H = A + B \ln T_H + (\ln T_H)^2 \left[C_o + C_1 (V_1 + V_2)^{C_3}\right] \tag{10}$$

and powered time

$$T_{HP} = D T_H^{\ E} \tag{11}$$

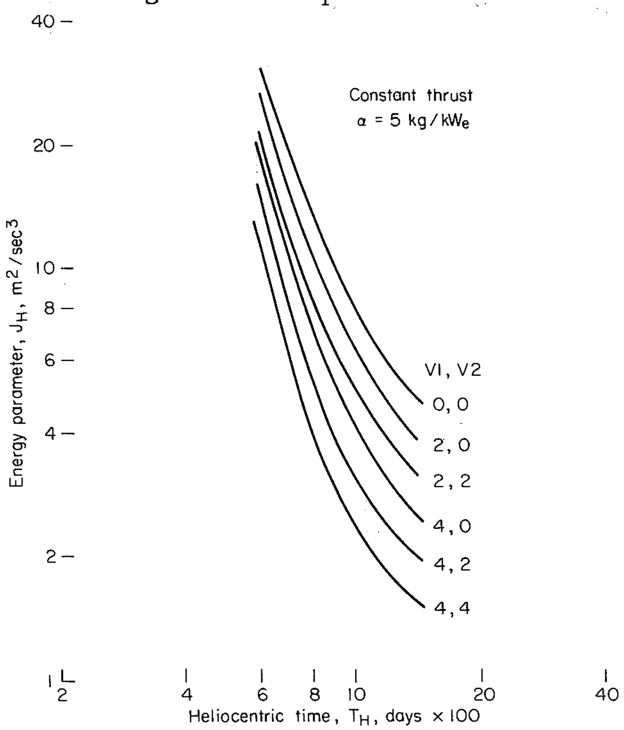


Figure 2.- Example of actual low-thrust performance stored in program for planet Jupiter.

where the constants A, B, C, D, and E are determined by the method of least squares to best represent the precomputed data. Hence, for given initial velocity V_1 and final velocity V_2 (in the case of orbiters), the heliocentric phase is described by:

$$J_H = f(T_H), \qquad T_{HP} = f(T_H)$$

For ease of convergence, the stored data have been generated using a powerplant specific mass α of 5 kg/kWe. The energy parameter J_H and the thrust time T_{HP} vary slightly with the system parameter α and are empirically described by:

$$J_H = J_H(1 + \text{vary}) \tag{12}$$

$$T_{HP} = T_{HP}(1 + \text{vary})^{1/3} \tag{13}$$

where

$$\text{vary} = 0.0001667 J_H(\alpha - 5)$$

Planetoheliocentric Matching

The heliocentric initial velocity V_1 differs from the planetocentric high-thrust departure velocity VINF1 by the amount gained in thrusting along the planetary escape trajectory immediately following high-thrust engine cutoff. This gain in velocity, due to applying even a small amount of finite thrust close to a gravitating body, is accounted for by either the method of asymptotic matching of the high-thrust hyperbolic departure trajectory with the low-thrust heliocentric trajectory (ref. 7) (MATCH = ASYMPT) or by the method of sphere of influence matching (MATCH = SPHERE built-in). Under sphere-of-influence matching, the low-thrust system initial velocity is related to the high-thrust system departure velocity by:

$$V1 = \sqrt{(VINF1)^2 + \frac{2GM}{145RG}} \tag{14}$$

$$V2 = \sqrt{(VINF2)^2 + \frac{2GM}{(RSPHERE)RGP}} \tag{15}$$

Under asymptotic matching, the low-thrust system initial velocity is related to the high-thrust system departure velocity by:

$$V_1 = G_{(X)} A_o{}^{1/4} \tag{16}$$

where

$$X = \frac{(VINF1)^2}{4\sqrt{(GM)A_o}}$$

$G_{(X)}$ contains complete elliptic integrals of the first and second kind, which have been accurately curve-fitted and stored within the program. Hence, it is noted that:

$$V_1 = f(VINF1, A_o)$$

In similar fashion for orbiters with high-thrust capture,

$$V_2 = G_{(X)} \left(\frac{A_o}{\mu_1}\right)^{1/4} \tag{17}$$

where

X $\dfrac{(VINF2)^2}{4\sqrt{GMP(A_o/\mu_1)}}$

GMP gravitational constant of target planet

VINF2 arrival velocity to be applied by high-thrust retrostage (constrained by input or left for optimization)

μ_1 final mass ratio of electric stage

Arrival

In the case of orbiters, a choice may be made on the thrust level for planet capture. ARRIVE = HIGH instructs the code to retrobrake into the desired orbit using a high-thrust stage of specified characteristics. The velocity increment is:

$$\Delta V = \sqrt{(VINF2)^2 + \frac{2(GMP)}{RP2(RGP)}} - \sqrt{\frac{GMP}{A_c}\left(\frac{1 + EPST}{1 - EPST}\right)} \tag{18}$$

where

RGP radius of target planet

RP2 periapsis of capture orbit

EPST eccentricity of capture orbit

A_c semimajor axis of capture orbit = RP2(RGP)/(1 - EPST)

The payload ratio of the high-thrust arrival system is given by:

$$ARRL = \frac{APROCH - WFUEL - WINERT}{APROCH} \tag{19}$$

where

WFUEL	$\left\{1 - \exp\left[\frac{-\Delta V}{\text{AISP}(0.00981)}\right]\right\}$ APROCH
WINERT	AINERT + ASIGMA(WFUEL)
AINERT	input fixed stage weight
ASIGMA	input tankage fraction
AISP	input specific impulse

With ARRIVE = LOW, the code will simulate a low-thrust spiral capture into the designated arrival orbit. The method of asymptotic matching similar to that described under DEPART = LOW yields the following:

$$J_c = f(T_c)$$

$$T_c = \text{capture time} = \text{powered time}$$

Optimization

The maximization of final payload requires the optimum allotment of mass during each phase. The overall payload is given by:

$$\text{PAYLOAD} = (\text{MLE})(\text{ARRL})(\text{DEPL})(\text{BOOSTL} - \text{WEJECT}) \tag{20}$$

where WEJECT represents any interstage mass, low-thrust start-up equipment, etc., which the analyst wishes to discard after launch vehicle injection. Thus, DEPL(BOOSTL - WEJECT) defines the initial gross mass of the low-thrust system. The definitions of BOOSTL, DEPL, and ARRL have been given above and require only iterations on the departure and arrival velocities to determine their values in the overall optimization scheme. The low-thrust payload mass fraction, MLE, can be determined as an integral part of minimizing J and apportioning the time spent in each phase. A method of system optimization (ref. 8), based on the near invariance of J with system parameters, has been found to be quite accurate, especially when the slight variation can be predicted and compensated. The underlying assumptions to this method are that the minimum value of J is invariant to μ_w, and the average thrust acceleration over a trajectory with a minimum J is also invariant to μ_w. The average thrust acceleration may be described by:

$$\bar{a} = (a_0 a_1)^{1/2} \tag{21}$$

and the initial acceleration by:

$$a_0 = \frac{2\eta\mu_w}{\alpha C} \tag{22}$$

where

C exhaust velocity

a_1 final acceleration = a_o/μ_1

An alternate expression for the average acceleration is:

$$\bar{a} = (J_T/T_P)^{1/2} \tag{23}$$

where J_T has previously been defined as:

$$J_T = \int_o^T a^2\,dt = J_D + J_H + J_C$$

and T_P is the total propulsion time along the entire low-thrust trajectory including all phases:

$$T_P = T_D + T_{HP} + T_C \tag{24}$$

The ratio of electric-propulsion payload mass (or net spacecraft mass as defined in this program) to its initial mass, DEPL(BOOSTL - WEJECT), is given by:

$$MLE = \mu_1 - \mu_W - \mu_T \tag{25}$$

and the final mass ratio is given by:

$$\mu_1 = \frac{\mu_W}{\mu_W + \frac{\gamma^2}{\eta}} \tag{26}$$

where

μ_W powerplant mass ratio

μ_T propellant tankage ratio = $k(1 - \mu_1)$

k tankage fraction (0.03 built-in)

η thrustor subsystem efficiency

γ^2 $\alpha J_T/2$

α powerplant specific mass

It is convenient to define the thrustor subsystem efficiency in the form of an analytical function whose derivative is continuous, thus:

$$\eta = \frac{B}{1 + (D/C)^2} \tag{27}$$

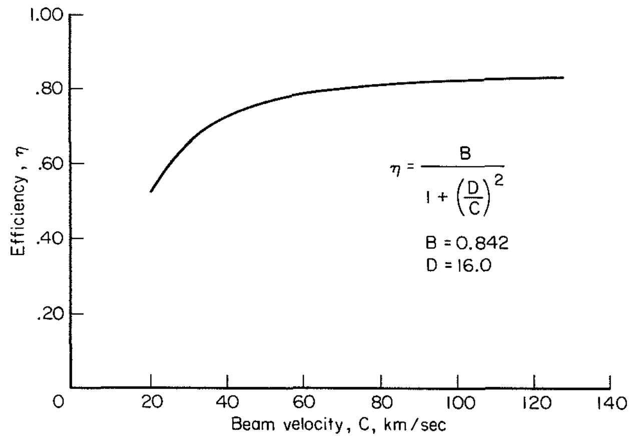


Figure 3.- Thrustor subsystem efficiency.

where

B constant (0.842 built-in)

D constant (16.0 built-in)

The best fit of B and D values to projections of references 9 and 10 is shown in figure 3, which includes a 90 percent power conditioning efficiency.

Optimum Power Level

To maximize payload, it is necessary to optimize the system parameters, exhaust velocity, and powerplant mass ratio. Setting the first variation of the payload mass ratio MLE equal to zero and using equations (23), (26), and (27) and the above relationships, the following expressions result for the optimum system parameters:

Exhaust velocity:

$$C = \left[\frac{2BT_P}{\alpha} + \frac{2BT_Pk}{\alpha} + D^2 - \frac{2T_P\gamma}{\alpha}\left(B + Bk + \frac{\alpha D^2}{2T_P}\right)^{1/2}\right]^{1/2} \tag{28}$$

Powerplant mass ratio:

$$\mu_W = \frac{\gamma + \gamma k + \frac{\gamma\alpha D^2}{T_P B}}{\left(B + Bk + \frac{\alpha D^2}{2T_P}\right)^{1/2}} - \frac{\gamma^2}{B} \tag{29}$$

The final mass ratio μ_1 and the payload fraction MLE may be found by direct substitution of μ_W and C into equations (26) and (27), giving:

$$\mu_1 = 1 - \frac{\gamma}{\left(B + Bk + \frac{\alpha D^2}{2T_P}\right)^{1/2}} \tag{30}$$

$$\text{MLE} = 1 + \frac{\gamma^2}{B} - \frac{2\gamma}{B}\left(B + Bk + \frac{\alpha D^2}{2T_P}\right)^{1/2} \tag{31}$$

The values of B, D, k, and α are presumably known; thus, the above optimum system equations require the total propulsion time T_P and the total energy parameter J_T which are functions of the heliocentric time T_H and the initial and final velocities V_1 and V_2. Hence, for maximum PAYLOAD, we need to determine the optimum V_1 and V_2 that maximize the payload of the combined heliocentric and planetocentric phases while seeking the best combination of T_D, T_H, and T_C. This is accomplished by incrementing the velocities in alternate fixed steps, DELV1, DELV2 (built-in values are both 0.5 km/sec) while iterating between the apportionment of time spent in each phase until the maximum overall payload is achieved. The optimum power level for the completely unconstrained case is given by:

$$\text{POWER} = \frac{\mu_W}{\alpha}\,\text{DEPL(BOOSTL - WEJECT)} \tag{32}$$

where DEPL(BOOSTL - WEJECT) represents the initial low-thrust system gross mass. The initial acceleration is:

$$\text{AZERO} = \frac{0.002\ \mu_W \eta}{\alpha C} \tag{33}$$

where μ_W and C are the optimum values found above.

Constrained Power Level

The preceding analysis has dealt with the case of fully optimized system parameters μ_W, T_P, C, and POWER. It often is necessary to determine the performance and requirements of a system that has a specified fixed power level. Through input of the desired powerplant characteristics ALPHA and POWER the mass and power level of the systems are constrained. The mass of the powerplant is given by:

$$\text{WPLANT} = \text{(ALPHA)POWER} \tag{34}$$

and the powerplant mass fraction is simply:

$$\mu_W = \frac{\text{WPLANT}}{\text{DEPL(BOOSTL - WEJECT)}} \tag{35}$$

From the previous definitions and equations (21), (22), (26), and (27), the following expression for exhaust velocity in terms of powerplant mass fraction results:

$$C = \left\{0.5\left(1 + \frac{\gamma^2}{B\mu_W}\right)\left(\frac{B^2 J T \mu_W^{\ 2}}{\gamma^4}\right)\left[1 + \sqrt{1 - \frac{1}{TJ}\left(\frac{2D\gamma^2}{\gamma^2 + B\mu_W}\right)^2}\,\right] - D^2\right\}^{1/2} \tag{36}$$

The required initial acceleration for this fixed power case is now:

$$A_{FP} = \frac{0.002\ \mu_W \eta}{\alpha C} \tag{37}$$

This change in initial acceleration, caused by a fixed μ_W and newly computed C, will affect the energy parameter J_T and the two assumptions underlying the previous method of system optimization. A technique of system optimization was therefore used which is based on the near invariance of trajectory characteristic length L with system parameters (ref. 11). In this technique, the characteristic length of a trajectory is assumed constant regardless of the type of propulsion system used to traverse its path. The form of this parameter, which is a measure of the energy requirements for the mission, is given by:

$$L = \frac{C^2}{A_o}\left[\left(1 - \sqrt{1 - \frac{A_o T_P}{C}}\right)^2 - \frac{A_o}{C}(T_T - T_P)R(ln)\left(1 - \frac{A_o T_P}{C}\right)\right] \tag{38}$$

where T_T is the total mission time. The constant R as derived (ref. 11) was 0.5. However, after inspection of numerous cases, the constant R used in this program was empirically set at 0.4, which causes L to more closely define both the optimum and constrained missions, and is identified as LPRIME (L'). Thus, L' is a function of C and T_P, since A_o depends on C. Ideally, one would hope to determine L' for the optimum power case and set it equal to the L' for the constrained case, thereby requiring only an iteration on T_P to determine the best C. Unfortunately, even L' varies with acceleration A_o, and, although slight, the variation is sufficient to cause unnecessary error in C and T_P and therefore low-thrust payload MLE. After some observation, the variation of characteristic length with acceleration, for the fixed-power case, was found to be simply:

$$L'_{FP} = L'_{OPT}\left(0.9 + 0.1\ \frac{A_{OPT}}{A_{FP}}\right) \tag{39}$$

where A_{OPT} and L'_{OPT} refer to the acceleration and length of the optimum power case, and A_{FP} and L'_{FP} refer to the fixed-power case. The method of solution is to first guess a T_P and solve equation (36) for C (with the known fixed μ_W). Next, determine the new A_{FP}, equation (37), and use equations (38) and (39) to determine a new value of T_P. Equation (36) is then solved for the new value of C. The final mass ratio is given by:

$$\mu_1 = 1 - \frac{A_{FP}T_P}{C} \tag{40}$$

and the payload mass ratio is given by:

$$MLE = \mu_1(1 + k) - \mu_W - k \tag{41}$$

Constrained Thrust Time With Fixed Power

In addition to the case of contrained power level, it is also realistic to specify a fixed upper bound on thrusting time. As for both optimum power and constrained power cases, the coast and thrust phases must still be optimally placed. The technique for system optimization of the fixed-thrust time with fixed-power case is similar to that of the fixed-power case. However, the trajectory characteristic length varies with both acceleration and thrust time in the following manner:

$$L'_{FTP} = L'_{FP}\left[0.85 + 0.15\left(\frac{A_{FP}}{A_{FTP}}\right)\left(\frac{TIMEON}{T_{FP}}\right)\right] \tag{42}$$

where A_{FTP} is the initial acceleration of the fixed-time, fixed-power case, TIMEON is the input constrained thrusting time upper limit, and T_{FP}, A_{FP}, and L'_{FP} are the thrust time, acceleration, and characteristic length found in the constrained-power, optimum-thrust-time case, which is solved prior to the constrained-power, constrained-time case. The procedure of solution is to guess a value of C, determine A_{FTP} and then compute L'_{FTP} by equation (42). Next, equation (38) is solved for the new value of C (knowing both T_P = TIMEON, which is input, and μ_W, which is computed from the constrained POWER input) and repeat the process until convergence. An excellent first guess for the exhaust velocity is:

$$C = C_{FP}\left(\frac{TIMEON}{T_{FP}}\right) \tag{43}$$

where the subscript FP refers to the previously solved fixed-power, optimum-time case. After determination of C, the low-thrust payload mass ratio computation proceeds in a manner similar to that of the previous case. The overall PAYLOAD optimization continues as the initial and final velocities are incremented and the phase times are apportioned with a subsequent iteration through the low-thrust system optimization, as described in the previous three subsections.

Powerplant Specific Mass as a Function of Power

An additional option may be exercised to investigate the effect of optimally sizing the powerplant to a launch vehicle - mission combination according to an assumed level of technology. Through the use of a functional relationship between power level and powerplant specific mass, the analyst can realistically determine the best compromise powerplant for a range of missions (ref. 12). Built into the code is the following empirical relationship, which

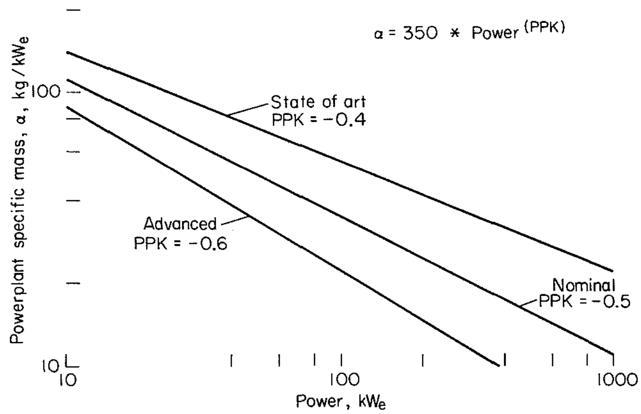


Figure 4.- Powerplant technology levels.

is shown in figure 4:

$$ALPHA = (WPLANT)POWER^{PPK} \qquad (44)$$

where

WPLANT constant describing the relationship (350.0 built-in)

PPK constant defining the technology level as:
-0.4 = state of art,
-0.5 = nominal,
-0.6 = advanced

It is seen that ALPHA will increase as POWER decreases. The code will automatically recycle itself until it finds the optimum power level, and hence ALPHA, which will yield the maximum payload for a given launch weight on a particular mission (ref. 13). When using this option, ALPHA or POWER are not input, as they will be computed internally. The only input required is PPK and WPLANT (if the user wishes to override the built-in value of 350.0).

DISCUSSION

General User Comments

The program described in this report has been used extensively for a wide range of missions. The experience gained in using the code has led to modifications and improvements to facilitate its use by others. No initial guesses are required on the part of the user since all initial values and starting solutions are provided internally. Convergence is completely assured on any case that has a solution. On all other cases, the code repairs any damage done to its logic and proceeds with the next input case. This expedites the running of numerous cases with large ranges in parameters. Built into the code are automatic input loops on ALPHA, TIME, POWER, and EPST, which allow the analyst the flexibility to cover large ranges in these variables. Upon input of a range of parameters, it is advantageous to run cases in the order of increasing difficulty (i.e., TIME = 1000.0, 800.0, 600.0; ALPHA = 30.0, 40.0, 50.0). Because of built-in cutoffs, time is saved by eliminating all missions more difficult than the last one to fail (negative payload). Reasonable values of most parameters have been built in (see appendix A) which, of course, the analyst can override with his own desired input values. The input itself has been simplified through the use of colloquial variables such as the proper names of launch vehicles and planets (BIRD = 'ATLAS/CENTAUR', 'PLANET = SATURN', etc.) and the straightforward spelling of parameters to indicate their function such as MODE = FLYBY, ARRIVE = HIGH, LAUNCH = ESCAPE, etc. All proper names in the input should be enclosed by quote marks.

The program has been coded in FORTRAN IV and has been kept relatively simple so that the logic flow may be followed easily, thereby facilitating the inevitable changes to suit a user's particular needs. The program has been exercised on the IBM 360-50 computer and the storage requirements are minimal. The execution times on this computer range from 0.1 second (flyby with optimum power) to 0.5 second (orbiter with constrained power and constrained thrust time and with initial and final velocities to be optimized). Execution times on other computers are estimated to be 0.05 to 0.25 on the IBM 7094, and 0.02 to 0.1 on the IBM 360-75. These times represent a two to three order-of-magnitude reduction in the compute times of other existing programs on the same subject. Nor has accuracy been sacrificed, as correlation with exact mission simulation data is excellent (refs. 1, 14).

The output format has been selected to allow a one-line listing per case so that as many as 40 missions may be concisely filed on one page. The output is to the line printer and is sized for inclusion in loose-leaf notebooks. Appendix B contains the output parameters and their definitions. In addition to the conventional listing, a graph of the data may be obtained by stating on input PRINT = GRAPH. This allows a plot of net spacecraft mass versus mission time for a family of powerplant specific mass. Built into the program is a diagnostic feature that on input of PRINT = HELP will output various key parameters internal to the program in case of unforeseen problems (see line 233 of main program).

In addition to the options covered thus far there is another feature that may be useful. To compute the all high-thrust propulsion cases for purposes of comparison with the low-thrust missions, simply input ALPHA = 0.0 and the VA and VB equal to the hyperbolic excess velocities associated with the all-ballistic mission. Accordingly, the code will compute only the launch, departure, and arrival conditions with a coast trajectory assumed for the entire heliocentric phase. All planetocentric maneuvers will be calculated using the characteristics of the high-thrust systems that are either built-in or overridden by input.

To illustrate the requirements for deck make-up of a job, appendix C displays the IBM cards used for example problem 3. It will be noticed that the fourth card includes the subroutines MPX04F1 through F5, which are basic to this program. The fifth card includes subroutines MOX01UP and MOX01IN. Subroutine MOX01UP is a graph or plot routine available to IBM 7094 users under the name UMPLOT. It has been modified to run on the IBM 360 series and requires the four program calls PLOT1 through 4. Subroutine MOX01IN is a data input routine that is similar to, and may be replaced by, IBM's NAMELIST input routine. The CALL INPUT within the program should be changed to be READ statements. The present input routine allows cases to be stacked by simply placing an asterisk card between each set of input. Only those input quantities which are to be changed need be added after the asterisk. All data may be punched on the cards in free form. To familiarize potential users with the code, a series of example problems are included in appendix D. A complete listing of the program FORTRAN statements is given in appendix E.

As with any program, there are certain limitations and restrictions inherent to the operation of the code described here. These are set forth in the following paragraphs along with some suggested extensions to its application.

Limitations

The simulation of a particular mission is limited by the amount of precomputed and stored data. The trajectory data, within the code, for Mercury and Venus do not include the variation with initial and final velocities, and there are no data stored for Pluto as yet. Jupiter swingby electric-propulsion data have not yet been included in the program and would represent a very useful extension of this research tool. Users of the program who find a need for data such as the above may conveniently store the information as outlined in the analysis section. Of course, the accuracy of any simulated mission is dependent on the accuracy of the stored data, which requires detailed trajectory computations and efficient curve fits. Inevitably, some missions will require extrapolations of the stored information, and the proper form of the representative curves will afford greater confidence in the result.

Extensions of the Program

If more refined simulation is desired, one may consider the coupling of this level 1 code with a more accurate level 2 trajectory program. The mission may then be recomputed using a detailed trajectory subroutine (ref. 15) with the level 1 parameters as an initial starting solution. This method allows the user various levels of analyses and gives him the flexibility of trading accuracy for time. The execute times for this scheme are greatly reduced since in the lower level of analyses the trajectory data has already been computed, and in the higher level excellent initial solutions are available for the optimization of system and trajectory parameters such as specific impulse, powerplant mass fraction, thrust acceleration, operating time, and departure and arrival velocities.

An integral part of overall mission simulation should include the definition of the propulsion system hardware. Although a simple relationship between power level and system mass is built-in, a more rigorous treatment of system analysis may be provided through the coupling of this present code with a systems and hardware definition program. Within such a code, the system may be detailed into subsystem modules, including thrustor and power conditioning mating; thermodynamic cycle calculations; radiator weight and area analysis; apportionment of accessory equipment, pumps, plumbing, etc.; reactor characteristics, shield weight breakdown; and geometric configuration design and weight summary. There would be, of course, feedback loops and optimizations between various subsystems, which may be subject to mission constraints such as distance from Sun, operating time, ambient temperature, power level, diameter of launch vehicle, etc. When fully developed, the mathematical modeling of the powerplant characteristics would allow the interplay necessary to an overall low-thrust mission simulation tool.

National Aeronautics and Space Administration
Moffett Field, Calif. 94035, Nov. 24, 1969

APPENDIX A

INPUT PARAMETERS

Variable name	Description
TIME	Total mission time, days
NT	Number of missions times input (NT = 1 built in)
ALPHA	Powerplant specific mass, kg/kWe
NA	Number of alphas input (NA = 1 built in)
LAUNCH	Type of launch trajectory desired = ESCAPE, will launch booster payload to escape = PARK, will launch booster payload to parking orbit
RP1	Radius of parking orbit if LAUNCH = PARK, Earth radii
EPSD	Eccentricity of parking orbit (0.0 built in)
BIRD	Proper name of launch vehicle selected (must be enclosed in quotes): = 'SATURNV' 'SATURNV/CENTAUR' 'SATURNI/CENTAUR' 'SIC/S4B' 'SIC/S4B/CENTAUR' 'TITAN3F', same as Titan 3X(1207) 'TITAN3F/CENTAUR' 'TITAN3D/CENTAUR', same as (1205) 'TITAN3D/AGENA' 'ATLAS/CENTAUR' 'ATLAS/AGENA' 'INPUT BOOST DATA', used if characteristics of launch vehicle are to be input
TARGET	Proper name of target or planet (must be enclosed in quotes): = 'MERCURY' 'VENUS' 'MARS' 'JUPITER' 'SATURN' 'URANUS' 'NEPTUNE' 'PLUTO' 'COMET HALEY', rendezvous 'EXTRA-ECLIPTIC', 1.0 AU rendezvous
ANGLE	Angle of inclination to the ecliptic if TARGET = 'EXTRA-ECLIPTIC', degrees
MODE	Type of mission (need not be input if TARGET = 'COMET HALEY' or TARGET = 'EXTRA-ECLIPTIC') = FLYBY - flyby = ORBIT - orbiter
POWER	Electrical power level, if constrained, kWe
NP	Number of powers input (NP = 1 built in)

RP2	Radius of capture orbit, if orbiter, planetary radii
EPST	Eccentricity of capture orbit (0.0 built in)
NET	Number of capture eccentricities input (NET = 1 built in)
DEPART	Thrust level of departure stage if launch = PARK = HIGH, ballistic-escape stage = LOW, electric stage, spiral escape
DISP	Specific impulse of departure stage, sec (DISP = 450.0 built in)
DSIGMA	Tank inert fraction of high-thrust departure stage (DSIGMA = 0.137 built in)
DINERT	Fixed inert mass of high-thrust departure stage, kg (DINERT = 0.0 built in)
ARRIVE	Thrust level of capture stage if MODE = ORBIT = HIGH, ballistic-capture stage = LOW, electric stage, spiral capture
AISP	Specific impulse of ARRIVAL stage, sec (AISP = 300.0 built in)
ASIGMA	Tank inert fraction of high-thrust ARRIVAL stage (ASIGMA = 0.10 built in)
AINERT	Fixed inert mass of high-thrust arrival stage, kg (AINERT = 0.0 built in)
D	Constant in thrustor-efficiency function, km/sec (D = 16.0 built in)
B	Constant in thrustor-efficiency function (B = 0.842 built in)
TANK	Low-thrust propellant tankage fraction (TANK = 0.03 built in)
DELV1	Increment size on departure hyperbolic velocity optimization, km/sec (DELV1 = 0.5 built in)
DELV2	Increment size on arrival hyperbolic velocity optimization, km/sec (DELV2 = 0.5 built in)
VA	Departure hyperbolic excess velocity if constrained
VB	Arrival hyperbolic excess velocity if constrained
TIMEON	Electric propulsion thrusting time upper limit if constrained, hours (TIMEON = 9999999.9 built in)
ENERGY	Source of electric power: = ATOMIC, nuclear electric (built in) = SOLAR, solar cell (no data built in yet)
WPLANT	Constant in alpha-power relationship, kg (WPLANT = 350.0 built in) (see eq. (44))
PPK	Constant in alpha-power relationship (PPK = 0.0 built in) (see eq. (44))
PAYUP	Tabular values of launch vehicle performance to be input if BIRD = 'INPUT BOOST DATA'; maximum of 16 values, kg (PAYUP = 16×0.0 built in)
VC	Tabular values of launch vehicle performance corresponding to input values of PAYUP; maximum of 16 values, km/sec (VC = 16×7.75 built in)
WEJECT	Interstage mass, low thrust start-up equipment, etc., discarded after launch vehicle injection, kg (WEJECT = 0.0 built in)

MATCH	Method of planetoheliocentric trajectory matching = SPHERE (sphere of influence, built in) = ASYMPT (asymptotic velocity matching)
PRINT	Output control = DATA built in yields standard output = GRAPH yields standard plus graph = HELP yields debug diagnostic

Under PRINT = GRAPH the following may be input:

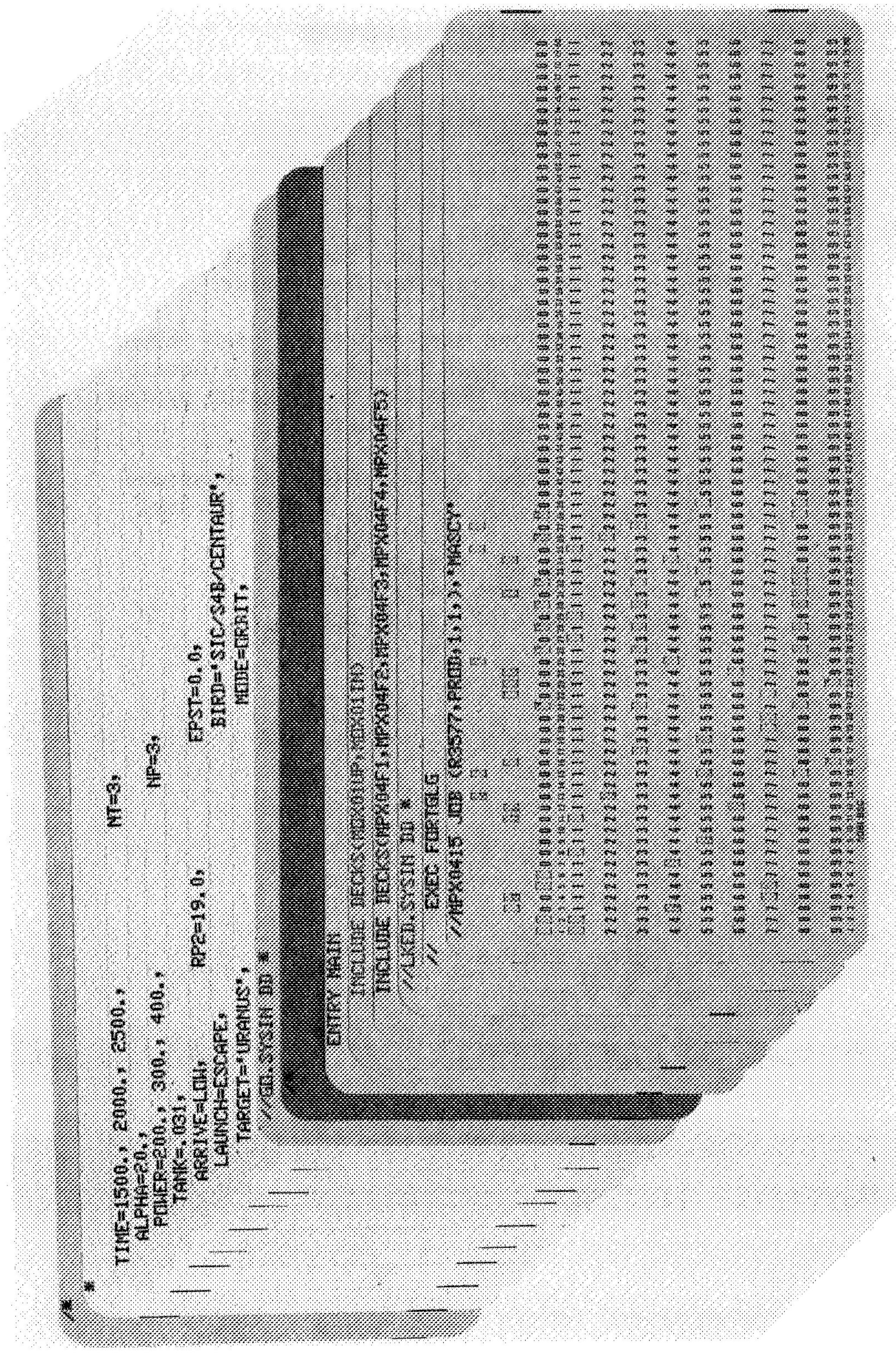
XMAX	Maximum trip time on abscissa, days (XMAX = 3200.0 built in)
XMIN	Minimum trip time on abscissa, days (XMIN = 0.0 built in)
YMAX	Maximum payload on ordinate, kg (YMAX = 70000.0 built in)
YMIN	Minimum payload on ordinate, kg (YMIN = 0.0 built in)
NHL	Number of horizontal grid lines ((NHL = 7 built in)
NVL	Number of vertical grid lines (NVL = 8 built in)
NSBH	Number of carriage spaces between horizontal lines (NSBH = 5 built in)
NSBV	Number of carriage spaces between vertical lines (NSBV = 10 built in)
NAME	Proper name of analyst (must be in quotes)

APPENDIX B

OUTPUT PARAMETERS

Variable name	Description	Units
TIME	Total mission time	days
ALPHA	Powerplant specific mass	kg/kWe
PAYLOAD	Net spacecraft mass	kg
MUP	Propellant mass fraction, low thrust	
MUW	Powerplant mass fraction, low thrust	
MLE	Payload mass fraction, low thrust	
DEPL	Departure payload mass fraction, high thrust.	
ARRL	Arrival payload mass fraction, high thrust	
POWER	Electrical power supplied to thrustor system	kWe
C	Exhaust velocity, low thrust	km/sec
T POW	Thrusting time, low thrust	hours
VINF 1	Hyperbolic excess velocity, Earth departure	km/sec
VINF 2	Hyperbolic excess velocity, Planet arrival	km/sec
BOOSTL	Launch vehicle injected payload	kg
TC	Capture time, low thrust	days
TH	Heliocentric time, low thrust	days
ETA	Thrustor subsystem efficiency p_j/p_e	

APPENDIX C - SAMPLE DECK MAKEUP



APPENDIX D

EXAMPLE PROBLEMS

Example 1 shows the input parameters and output variables of a simple Saturn flyby mission. The optimum power level is shown to decrease with increasing ALPHA and to decrease with increasing mission time. The optimized hyperbolic excess velocity increases with increased ALPHA. The BOOSTL is the mass injected on a hyperbolic escape trajectory by the ATLAS/AGENA launch vehicle.

Example 2, a Jupiter orbiter, shows the method of overriding the built-in characteristics of the high-thrust capture stage by specifying AISP and ASIGMA. Note that the capture orbit characteristics are those of an ellipse with periapsis, RP2 = 2.0 Jupiter radii and eccentricity, EPST = 0.9. The exhaust velocity C of the electric stage is shown to decrease with increasing ALPHA. For this mission, the hyperbolic velocities at both departure and arrival, VINF1 and VINF2, respectively, have been optimized to yield maximum payload.

Example 3 indicates the method of constraining the electric power level of the low-thrust propulsion system by input of the desired level of POWER. The constrained power levels are shown in the appropriate output column. The low-thrust propellant tankage fraction has also been input, TANK = 0.031. Note that when a single variable is input such as one ALPHA, there is no need to input the quantity of that variable such as NA = 1.

Example 4 shows that mode need not be input when TARGET = 'EXTRA-ECLIPTIC'. The printout in the center of the page indicates that it is a rendezvous type mission, that is, the probe arrives at the desired inclination to the ecliptic (ANGLE = 45.0) at 1.0 AU and remains in a circular orbit with those conditions.

Example 5 uses a desired mass in Earth orbit by specifying LAUNCH = PARK, and BIRD = 'INPUT BOOST DATA'. In this case, the fixed initial mass of 50,000 kilograms is input by PAYUP = 50000.0. Note that in all cases the output shows that BOOSTL = 50000.0. Departure from the parking orbit is by a high-thrust vehicle, DEPART = HIGH, whose input characteristics are those of an assumed nuclear stage, specific impulse DISP = 800.0 seconds, hydrogen tankage fraction DSIGMA = 0.20, and fixed inerts plus nuclear engine DINERT = 7000.0 kg.

Example 6 constrains the departure hyperbolic excess velocity VA = 4.0 kilometers per second. Note the output column VINF1. Also, in this example, the capture orbit is circular (RP2 = 16.0 planet radii); therefore, the eccentricity, EPST = 0.0, need not be input.

Example 7 utilizes the built-in power-alpha relationship, equation (44). The nominal technology selected for this example requires WPLANT = 350.0 and

PPK = -0.5. Note the POWER column outputs the power level best suited for this launch vehicle-departure mode that conforms to the above constraint. The ALPHA column shows the powerplant specific mass corresponding to this power relationship. Neither POWER nor ALPHA should be input under this option.

Example 8 illustrates the use of the program in nondimensional parameters. Through the input LAUNCH = PARK and BIRD = NO BOOST, the code will initiate all missions from Earth orbit and the payload will be normalized to Earth orbital mass. The PBAR column gives the ratio of POWER divided by PAYLOAD in units of kilowatts/kilograms. For example, a mission time of 2000 days and a powerplant specific mass of 20 kg/kWe yields a payload mass fraction of 0.0869 and a PBAR = 0.0580. If we desired our Earth orbital mass to be 10,000 kg, then our payload would be 869 kg and the primary electrical power required would be 50.3 kWe [(0.0580)(869)]. Similarly, for an orbital mass of 20,000 kg, payload would be 1738 kg and power would be 100.6 kWe.

Example 9 depicts a very useful output format. Through the input PRINT = GRAPH, the results of the mission analysis is graphically portrayed following the standard columnar printout. The plot shows mission time in days along the abscissa or x-axis and net spacecraft mass (payload) in kilograms along the ordinate or y-axis. The maximum and minimum values of these parameters are controlled by the inputs YMAX, YMIN, XMAX, XMIN. The number of horizontal and vertical grid spaces may also be changed by input as described in appendix A. The family of alphas is plotted with the symbol corresponding to ALPHA divided by 10, thus 3, 4, 5 refer to ALPHA = 30, 40, 50 kg/kWe. The conditions of launch, departure, and arrival are shown in the upper left corner as well as the power level. The word FIGURE is output for labeling convenience on the bottom left and the date of the computer run is automatically printed on the bottom right corner. The name of the analyst may be inscribed on the bottom right corner by the input NAME = 'user's name'.

```
TARGET='SATURN',                        MODE=FLYBY,
LAUNCH=ESCAPE,                          BIRD='ATLAS/AGENA',
ALPHA=10., 20., 30.,                    NA=3,
TIME=1000., 1200., 1400., 1600.,        NT=4,
*
```

EARTH TO SATURN FLYBY

ATLAS/AGENA LAUNCH TO ESCAPE

DEPART HIGH

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
1000.	10.0	356.	.231	.197	.565	1.000	1.000	12.4	77.6	12126.	0.0	0.0	630.	0.	1000.	.808
1000.	20.0	256.	.319	.261	.410	1.000	1.000	8.2	52.5	12126.	0.5	0.0	625.	0.	1000.	.770
1000.	30.0	187.	.368	.302	.319	1.000	1.000	5.9	41.9	12126.	1.5	0.0	587.	0.	1000.	.735
1200.	10.0	391.	.199	.175	.621	1.000	1.000	11.0	85.7	14257.	0.0	0.0	630.	0.	1200.	.814
1200.	20.0	302.	.277	.235	.480	1.000	1.000	7.4	58.3	14257.	0.0	0.0	630.	0.	1200.	.783
1200.	30.0	238.	.326	.275	.389	1.000	1.000	5.6	46.6	14257.	1.0	0.0	611.	0.	1200.	.753
1400.	10.0	413.	.179	.160	.656	1.000	1.000	10.1	92.7	16348.	0.0	0.0	630.	0.	1400.	.818
1400.	20.0	332.	.250	.216	.527	1.000	1.000	6.8	63.4	16348.	0.0	0.0	630.	0.	1400.	.792
1400.	30.0	272.	.300	.256	.435	1.000	1.000	5.3	50.6	16348.	0.5	0.0	625.	0.	1400.	.765
1600.	10.0	428.	.166	.150	.680	1.000	1.000	9.4	99.0	18406.	0.0	0.0	630.	0.	1600.	.821
1600.	20.0	352.	.232	.202	.558	1.000	1.000	6.4	67.9	18406.	0.0	0.0	630.	0.	1600.	.798
1600.	30.0	295.	.281	.242	.469	1.000	1.000	5.1	54.2	18406.	0.0	0.0	630.	0.	1600.	.774

Example 1.

```
TARGET='JUPITER',                       MODE=ORBIT,
LAUNCH=ESCAPE,                          BIRD='TITAN3F/CENTAUR',
ARRIVE=HIGH,        RP2=2.0,            EPST=.9,
AISP=310.0,                             ASIGMA=.11,
ALPHA=30.0,  40.0,  50.0,               NA=3,
TIME=1000.0, 1200.,  1300.,  1400.,     NT=4,
*
```

EARTH TO JUPITER ORBITER

TITAN3F/CENTAUR LAUNCH TO ESCAPE

DEPART HIGH ARRIVE HIGH

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
1000.	30.0	2540.	.211	.206	.576	1.000	0.585	51.7	50.0	14025.	1.5	5.5	7533.	0.	1000.	.764
1000.	40.0	2279.	.222	.225	.547	1.000	0.553	42.4	43.6	14025.	1.5	6.5	7533.	0.	1000.	.742
1000.	50.0	2077.	.225	.240	.529	1.000	0.536	35.1	39.4	14025.	2.0	7.0	7322.	0.	1000.	.723
1200.	30.0	2870.	.188	.183	.623	1.000	0.599	47.0	54.8	16513.	1.0	5.0	7688.	0.	1200.	.776
1200.	40.0	2624.	.198	.200	.595	1.000	0.585	37.7	47.7	16513.	1.5	5.5	7533.	0.	1200.	.757
1200.	50.0	2425.	.201	.211	.582	1.000	0.553	31.9	43.0	16513.	1.5	6.5	7533.	0.	1200.	.740
1300.	30.0	2997.	.181	.177	.637	1.000	0.612	45.2	56.9	17740.	1.0	4.5	7688.	0.	1300.	.780
1300.	40.0	2753.	.192	.193	.610	1.000	0.599	36.3	49.5	17740.	1.5	5.0	7533.	0.	1300.	.762
1300.	50.0	2561.	.194	.203	.597	1.000	0.570	30.6	44.6	17740.	1.5	6.0	7533.	0.	1300.	.746
1400.	30.0	3101.	.170	.166	.659	1.000	0.612	42.6	59.1	18958.	1.0	4.5	7688.	0.	1400.	.784
1400.	40.0	2864.	.186	.186	.622	1.000	0.599	35.8	51.1	18958.	1.0	5.0	7688.	0.	1400.	.767
1400.	50.0	2676.	.190	.197	.607	1.000	0.585	29.7	46.1	18958.	1.5	5.5	7533.	0.	1400.	.751

Example 2.

```
TARGET='URANUS',                       MODE=ORBIT,
LAUNCH=ESCAPE,                         BIRD='SIC/S4B/CENTAUR',
ARRIVE=LOW,          RP2=19.0,         EPST=0.0,
TANK=.031,
POWER=200., 300., 400.,                NP=3,
ALPHA=20.,
TIME=1500., 2000., 2500.,              NT=3,
*
```

EARTH TO URANUS ORBITER

SIC/S4B/CENTAUR LAUNCH TO ESCAPE

DEPART HIGH ARRIVE LOW

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
1500.	20.0	1499.	.583	.290	.109	1.000	1.000	200.0	55.1	21762.	6.0	0.0	13776.	10.	1490.	.776
1500.	20.0	1300.	.576	.333	.072	1.000	1.000	300.0	59.7	21754.	4.0	0.0	17992.	9.	1491.	.786
1500.	20.0	820.	.543	.399	.041	1.000	1.000	400.0	67.7	21751.	3.0	0.0	20048.	9.	1491.	.797
2000.	20.0	4733.	.525	.210	.248	1.000	1.000	200.0	56.2	28174.	3.5	0.0	19058.	19.	1981.	.779
2000.	20.0	5106.	.473	.277	.235	1.000	1.000	300.0	68.8	28162.	2.0	0.0	21697.	18.	1982.	.799
2000.	20.0	4832.	.412	.359	.217	1.000	1.000	400.0	84.7	28159.	1.5	0.0	22312.	17.	1983.	.813
2500.	20.0	7472.	.457	.184	.344	1.000	1.000	200.0	62.9	34446.	2.0	0.0	21697.	30.	2470.	.791
2500.	20.0	7904.	.378	.264	.347	1.000	1.000	300.0	83.9	34434.	1.0	0.0	22763.	29.	2471.	.812
2500.	20.0	7514.	.317	.347	.326	1.000	1.000	400.0	105.8	34430.	0.5	0.0	23038.	29.	2471.	.823

Example 3.

```
TARGET='EXTRA-ECLIPTIC',
ANGLE=45.0,
ALPHA=10., 20., 30.,                   NA=3,
TIME=400., 500., 600., 700.,           NT=4,
LAUNCH=ESCAPE,                         BIRD='TITAN3D/CENTAUR',
POWER=50.0,
*
```

EARTH TO EXTRA-ECLIPTIC RENDEZVU

TITAN3D/CENTAUR LAUNCH TO ESCAPE

DEPART HIGH

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
400.	10.0	1238.	.519	.134	.332	1.000	1.000	50.0	33.8	8928.	5.0	0.0	3734.	0.	400.	.688
400.	20.0	738.	.519	.268	.198	1.000	1.000	50.0	33.8	8928.	5.0	0.0	3734.	0.	400.	.688
400.	30.0	238.	.519	.402	.064	1.000	1.000	50.0	33.8	8928.	5.0	0.0	3734.	0.	400.	.688
500.	10.0	1654.	.487	.116	.383	1.000	1.000	50.0	36.7	11160.	4.0	0.0	4323.	0.	500.	.708
500.	20.0	1154.	.487	.231	.267	1.000	1.000	50.0	36.7	11160.	4.0	0.0	4323.	0.	500.	.708
500.	30.0	654.	.487	.347	.151	1.000	1.000	50.0	36.7	11160.	4.0	0.0	4323.	0.	500.	.708
600.	10.0	2052.	.459	.103	.424	1.000	1.000	50.0	39.7	13392.	3.0	0.0	4838.	0.	600.	.724
600.	20.0	1552.	.459	.207	.321	1.000	1.000	50.0	39.7	13392.	3.0	0.0	4838.	0.	600.	.724
600.	30.0	1052.	.459	.310	.217	1.000	1.000	50.0	39.7	13392.	3.0	0.0	4838.	0.	600.	.724
700.	10.0	2419.	.410	.099	.479	1.000	1.000	50.0	45.0	15624.	2.5	0.0	5055.	0.	700.	.748
700.	20.0	1919.	.410	.198	.380	1.000	1.000	50.0	45.0	15624.	2.5	0.0	5055.	0.	700.	.748
700.	30.0	1419.	.410	.297	.281	1.000	1.000	50.0	45.0	15624.	2.5	0.0	5055.	0.	700.	.748

Example 4.

```
TARGET='SATURN',                                MODE=ORBIT,
LAUNCH=PARK,                                    BIRD='INPUT BOOST DATA',
PAYUP=50000.0,
DEPART=HIGH,                                    RP1=1.05,
DISP=800.0,          DSIGMA=.20,                DINERT=7000.0,
ARRIVE=LOW,                                     RP2=20.0,
ALPHA=10.0, 20.0, 30.0,                         NA=3,
TIME=1000., 1500., 2000., 2500.,                NT=4,
*
```

EARTH TO SATURN ORBITER

INPUT BOOST DATA LAUNCH TO PARKING

DEPART HIGH ARRIVE LOW

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
1000.	10.0	6987.	.391	.263	.334	0.418	1.000	551.0	75.6	14624.	3.0	0.0	50000.	9.	991.	.806
1000.	20.0	3354.	.475	.299	.212	0.316	1.000	236.1	50.3	14644.	6.0	0.0	50000.	11.	989.	.765
1000.	30.0	1628.	.520	.324	.140	0.232	1.000	125.1	39.8	14661.	8.0	0.0	50000.	13.	987.	.725
1500.	10.0	11742.	.261	.208	.524	0.448	1.000	465.0	100.1	21283.	1.5	0.0	50000.	26.	1474.	.821
1500.	20.0	8219.	.342	.254	.393	0.418	1.000	266.1	67.4	21309.	3.0	0.0	50000.	28.	1472.	.797
1500.	30.0	5992.	.396	.284	.308	0.389	1.000	184.5	53.2	21329.	4.0	0.0	50000.	30.	1470.	.772
2000.	10.0	14101.	.204	.173	.617	0.457	1.000	395.6	118.5	27846.	0.5	0.0	50000.	50.	1950.	.827
2000.	20.0	11047.	.272	.218	.502	0.440	1.000	240.1	80.8	27887.	2.0	0.0	50000.	54.	1946.	.810
2000.	30.0	8975.	.323	.250	.417	0.430	1.000	179.2	64.1	27905.	2.5	0.0	50000.	55.	1945.	.793
2500.	10.0	15457.	.171	.150	.674	0.459	1.000	343.5	134.2	34378.	0.0	0.0	50000.	83.	2417.	.830
2500.	20.0	12750.	.231	.193	.569	0.448	1.000	216.3	92.0	34428.	1.5	0.0	50000.	87.	2413.	.817
2500.	30.0	10864.	.275	.223	.494	0.440	1.000	163.6	73.4	34457.	2.0	0.0	50000.	90.	2410.	.804

Example 5.

```
TARGET='NEPTUNE',                               MODE=ORBIT,
LAUNCH=ESCAPE,                                  BIRD='TITAN3D/AGENA',
ARRIVE=LOW,                                     RP2=16.0,
ALPHA=20., 30., 40.,                            NA=3,
TIME=3200., 3000., 2800.,    NT=3,
VA=4.0,
*
```

EARTH TO NEPTUNE ORBITER

TITAN3D/AGENA LAUNCH TO ESCAPE

DEPART HIGH ARRIVE LOW

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
3200.	20.0	734.	.396	.259	.333	1.000	1.000	28.5	90.2	42358.	4.0	0.0	2203.	52.	3148.	.816
3200.	30.0	493.	.483	.279	.224	1.000	1.000	20.5	68.5	42358.	4.0	0.0	2203.	52.	3148.	.798
3200.	40.0	314.	.555	.286	.142	1.000	1.000	15.8	55.3	42358.	4.0	0.0	2203.	52.	3148.	.777
3000.	20.0	657.	.424	.265	.298	1.000	1.000	29.2	86.0	40286.	4.0	0.0	2203.	45.	2955.	.814
3000.	30.0	412.	.516	.281	.187	1.000	1.000	20.7	64.7	40286.	4.0	0.0	2203.	45.	2955.	.793
3000.	40.0	232.	.593	.284	.106	1.000	1.000	15.6	51.6	40286.	4.0	0.0	2203.	45.	2955.	.768
2800.	20.0	570.	.457	.271	.259	1.000	1.000	29.9	81.4	38190.	4.0	0.0	2203.	38.	2762.	.811
2800.	30.0	322.	.556	.281	.146	1.000	1.000	20.6	60.3	38190.	4.0	0.0	2203.	38.	2762.	.787
2800.	40.0	145.	.639	.276	.066	1.000	1.000	15.2	47.4	38190.	4.0	0.0	2203.	38.	2762.	.756

Example 6.

```
TARGET='JUPITER',                              MODE=FLYBY,
LAUNCH=PARK,                                   BIRD='TITAN3F       ',
DEPART=LOW,
WPLANT=350.0,                                  PPK=-.5,
TIME=800., 1000., 1200., 1400., 1600., 1800., 2000.,       NT=7,
*
```

EARTH TO JUPITER FLYBY

TITAN3F LAUNCH TO PARKING

DEPART LOW

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
800.	24.4	5552.	.371	.293	.325	1.000	1.000	205.0	45.3	11760.	0.0	0.0	17088.	0.	653.	.749
1000.	26.5	6603.	.334	.270	.386	1.000	1.000	173.9	50.5	15158.	0.0	0.0	17088.	0.	785.	.765
1200.	28.2	7265.	.311	.254	.425	1.000	1.000	154.2	55.0	18657.	0.0	0.0	17088.	0.	909.	.776
1400.	29.5	7706.	.297	.243	.451	1.000	1.000	141.1	59.1	22267.	0.0	0.0	17088.	0.	1024.	.785
1600.	30.5	8010.	.287	.235	.469	1.000	1.000	132.0	63.1	26006.	0.0	0.0	17088.	0.	1127.	.791
1800.	31.2	8224.	.281	.229	.481	1.000	1.000	125.4	66.9	29889.	0.0	0.0	17088.	0.	1216.	.796
2000.	31.9	8376.	.277	.225	.490	1.000	1.000	120.7	70.7	33933.	0.0	0.0	17088.	0.	1291.	.801

Example 7.

```
TARGET='URANUS',                    MODE=ORBIT,
LAUNCH=PARK,                         BIRD='NO BOOST',
DEPART=HIGH,                        RP1=1.10,
DISP=420.,                          DSIGMA=.12,
ARRIVE=HIGH,                        RP2=2.0,           EPST=.9,
ALPHA=10., 20., 30., 40., 50.,      NA=5,
TIME=1600., 1800., 2000.,           NT=3,
*
```

EARTH TO URANUS ORBITER

NO BOOST LAUNCH TO PARKING

DEPART HIGH ARRIVE HIGH

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEP L	ARR L	PBAR	C	T POW	VINF1	VINF2	TC	TH	ETA
1600.	10.0	.1031	.407	.2604	0.3208	0.392	0.820	0.0990	93.0	22931.	1.5	2.0	0.0	1600.	.818
1600.	20.0	.0478	.510	.2854	0.1896	0.355	0.710	0.1059	60.3	22931.	3.0	4.0	0.0	1600.	.787
1600.	30.0	.0226	.550	.2990	0.1345	0.301	0.560	0.1324	47.6	22931.	4.5	6.0	0.0	1600.	.756
1600.	40.0	.0107	.568	.3127	0.1026	0.238	0.439	0.1734	40.7	22931.	6.0	7.5	0.0	1600.	.729
1600.	50.0	.0048	.556	.3282	0.0988	0.217	0.225	0.2949	37.2	22931.	6.5	10.5	0.0	1600.	.711
1800.	10.0	.1274	.361	.2473	0.3811	0.399	0.838	0.0774	101.6	25460.	1.0	1.5	0.0	1800.	.822
1800.	20.0	.0702	.461	.2794	0.2460	0.369	0.773	0.0735	66.5	25460.	2.5	3.0	0.0	1800.	.796
1800.	30.0	.0400	.514	.2953	0.1751	0.338	0.675	0.0833	52.0	25460.	3.5	4.5	0.0	1800.	.769
1800.	40.0	.0229	.540	.3082	0.1359	0.301	0.560	0.1013	44.1	25460.	4.5	6.0	0.0	1800.	.744
1800.	50.0	.0131	.548	.3211	0.1145	0.238	0.479	0.1170	39.4	25460.	6.0	7.0	0.0	1800.	.723
2000.	10.0	.1475	.321	.2328	0.4363	0.403	0.838	0.0637	109.6	27958.	0.5	1.5	0.0	2000.	.824
2000.	20.0	.0907	.419	.2708	0.2975	0.382	0.799	0.0570	72.2	27958.	2.0	2.5	0.0	2000.	.803
2000.	30.0	.0579	.476	2898	0.2198	0.355	0.743	0.0591	56.4	27958.	3.0	3.5	0.0	2000.	.779
2000.	40.0	.0374	.509	.3033	0.1728	0.320	0.675	0.0650	47.6	27958.	4.0	4.5	0.0	2000.	.757
2000.	50.0	.0242	.525	.3[illegible]53	0.1436	0.301	0.560	0.0785	42.2	27958.	4.5	6.0	0.0	2000.	.736

Example 8.

```
TARGET='SATURN',                              MODE=ORBIT,
LAUNCH=PARK,                                  BIRD='TITAN3F      ',
DEPART=LOW ,                                  RP1=1.05,
ARRIVE=HIGH,                                  RP2=2.0,            EPST=.9,
TIME=1000., 1200., 1400., 1600., 2000., 2600.,        NT=6,
ALPHA=20., 30., 40.,                          NA=3,
TIMEON=10000.,                                POWER=200.,
PRINT=GRAPH,                                  NAME='MASCY',
YMAX=7000.,         YMIN=0.0,                 XMAX=2600.,         XMIN=1000.,
*
```

EARTH TO SATURN ORBITER

TITAN3F LAUNCH TO PARKING

DEPART LOW ARRIVE HIGH

TIME	ALPHA	PAYLOAD	MUP	MUW	MLE	DEPL	ARRL	POWER	C	T POW	VINF1	VINF2	BOOSTL	TC	TH	ETA
1000.	20.0	1284.	.582	.234	.167	1.000	0.451	200.0	40.3	15414.	0.0	8.5	17088.	0.	917.	.727
1000.	30.0	557.	.516	.351	.118	1.000	0.278	200.0	43.4	15536.	0.0	11.5	17088.	0.	906.	.741
1000.	40.0	159.	.450	.468	.068	1.000	0.136	200.0	47.1	15675.	0.0	14.5	17088.	0.	893.	.755
1200.	20.0	2391.	.516	.234	.234	1.000	0.597	200.0	47.6	18319.	0.0	6.0	17088.	0.	1092.	.756
1200.	30.0	1338.	.461	.351	.174	1.000	0.451	200.0	50.8	18450.	0.0	8.5	17088.	0.	1080.	.766
1200.	40.0	586.	.407	.468	.113	1.000	0.305	200.0	54.6	18598.	0.0	11.0	17088.	0.	1067.	.776
1400.	20.0	3417.	.455	.234	.297	1.000	0.674	200.0	55.2	21230.	0.0	4.5	17088.	0.	1264.	.777
1400.	30.0	2173.	.413	.351	.224	1.000	0.569	200.0	58.4	21360.	0.0	6.5	17088.	0.	1252.	.783
1400.	40.0	1153.	.371	.468	.150	1.000	0.451	200.0	62.1	21506.	0.0	8.5	17088.	0.	1239.	.790
1600.	20.0	4291.	.393	.234	.361	1.000	0.696	200.0	64.1	24183.	0.0	4.0	17088.	0.	1430.	.793
1600.	30.0	2936.	.373	.351	.264	1.000	0.650	200.0	66.0	24263.	0.0	5.0	17088.	0.	1423.	.795
1600.	40.0	1740.	.333	.468	.189	1.000	0.540	200.0	70.3	24438.	0.0	7.0	17088.	0.	1407.	.801
2000.	20.0	5610.	.319	.234	.438	1.000	0.750	200.0	80.2	30013.	0.0	2.5	17088.	0.	1762.	.810
2000.	30.0	4144.	.301	.351	.339	1.000	0.716	200.0	82.7	30119.	0.0	3.5	17088.	0.	1753.	.812
2000.	40.0	2761.	.275	.468	.249	1.000	0.650	200.0	87.0	30298.	0.0	5.0	17088.	0.	1737.	.814
2600.	20.0	6844.	.241	.234	.518	1.000	0.773	200.0	105.8	38817.	0.0	1.5	17088.	0.	2244.	.823
2600.	30.0	5315.	.227	.351	.415	1.000	0.750	200.0	109.1	38964.	0.0	2.5	17088.	0.	2232.	.824
2600.	40.0	3828.	.220	.468	.305	1.000	0.734	200.0	111.0	39047.	0.0	3.0	17088.	0.	2224.	.825

```
**********************************************SATURN            ORBITER ******************************
TITAN3F            LAUNCH TO  PARKING
 LOW   THRUST ESCAPE
 HIGH  THRUST CAPTURE   RP2=  2.0  ECC=.90
POWER= 200. KWE                                          THRUST TIME UPPER LIMIT=10000.HOURS
```

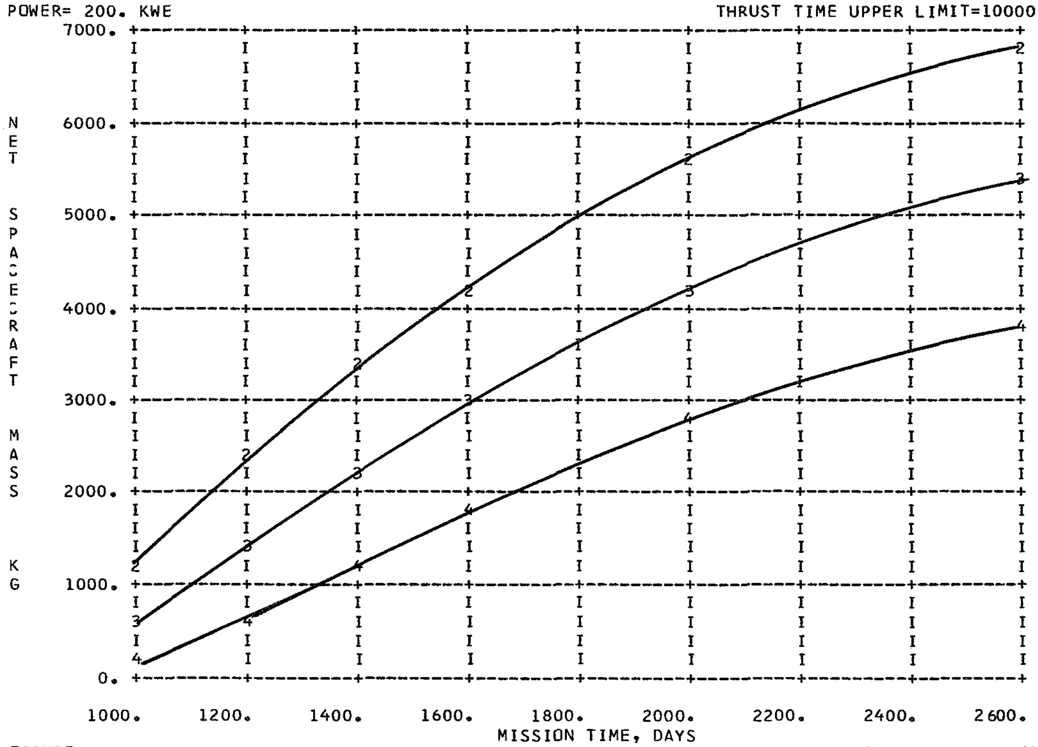


FIGURE MASCY 09/16/69

Example 9.

APPENDIX E - PROGRAM LISTINGS

```
C  MPX04F1   MAIN CONTROL PROGRAM FOR QUICKLY ANALYZING LOW THRUST MISSIONS
C
C
      LOGICAL MODE,FLYBY,ORBIT,ARRIVE,DEPART,HIGH,LOW
      LOGICAL LAUNCH,ESCAPE,PARK,MATCH, SPHERE, ASYMPT
      LOGICAL ENERGY,ATOMIC,SOLAR
      INTEGER DATA, GRAPH, HELP
      DIMENSION BIRD(4), VEHIK(3,13)
      DOUBLE PRECISION XLAUNC,ESCAP,PARKIN,RENDEZ
      DOUBLE PRECISION XMODE, ORBTR, FLBY, YLEVEL, YHIGH, XLEVEL
      DOUBLE PRECISION XHIGH,YLOW, XLOW
      DIMENSION T(20), ALPHA(20), THI(2), FTH(2), TTD(2), TTC(2)
      DIMENSION EPST(20) ,             POWERH(20),POWER(20)
      DIMENSION NSCALE(5),IMAGE(800) , DUMMY(1), CHAR(10)
      DIMENSION TIME(20)
      DIMENSION DAT(2),NAME(3)
      DIMENSION VC7(16), PAY7(16), VC(16), PAYUP(16)
      DIMENSION HOME(2),TARGET(4),PLAN(11)
      DIMENSION BDY(3,3,2)
      DATA  NAME/'MASCY        '/
      DATA PLAN/'MERC','VENU','EART', 'MARS','JUPI','SATU',
     1  'URAN', 'NEPT', 'PLUT','COME', 'EXTR'/
      DATA BLANK/'    '/
      DATA DUMMY/'0'/
      DATA NSCALE/1,0,0,0,0/
      DATA CHAR/'1','2','3','4','5','6','7','8','9' /
      DATA VEHIK/'NO B','OOST','    ',    'SATU','RNV ','    ',
     1  'SIC/','S4B/','CENT',    'SIC/','S4B ','    ',
     2     'TITA','N3F/','CENT',      'TITA','N3F ','    ',
     3     'INPU','T BO','OST ',          'SATU','RNV/','CENT',
     4      'SATU','RNI/','CENT',          'TITA','N3D/','AGEN',
     5       'ATLA','S/CE','NTAU',            'ATLA','S/AG','ENA ',
     6     'TITA','N3D/','CENT'/
      DATA ORBTR/'ORBITER '/
      DATA FLBY/ ' FLYBY  '  /
      DATA RENDEZ/'RENDEZVU'/
      DATA XHIGH/ ' HIGH   '/
      DATA XLOW/  '  LOW   ' /
      DATA  YHIGH/ '  HIGH  '  /
      DATA YLOW/  '  LOW   ' /
      DATA ESCAP/'ESCAPE'/
      DATA PARKIN/'PARKING'/
      DATA BIRD/'                '/
      DATA TARGET/'                '/
      DATA HOME/'EARTH   '/
      DATA BIRD/4HNO B,4HOOST/
      CALL DATE(DAT)
      DATA PAYUP/16*0.0/
      DATA  DATA,GRAPH,HELP/0,1,2/
      ORBIT=.TRUE.
      FLYBY=.FALSE.
      LOW=.TRUE.
      HIGH=.FALSE.
      ESCAPE=.TRUE.
      PARK=.FALSE.
      ATOMIC=.TRUE.
      SOLAR=.FALSE.
      ENERGY=ATOMIC
      SPHERE=.TRUE.
      ASYMPT=.FALSE.
      MATCH=SPHERE
      ARRIVE=HIGH
      NPO=0
      NGRAIN=0
      DEPART=HIGH
      LAUNCH=ESCAPE
      MODE=FLYBY
      NBIRD=1
      NPLAN2=0
      RP2=0.0
      GMEO=39.86E4
      GE=.00981
      DISP=450.
      DINERT=0.0
      DSIGMA=.137
      AISP=300.0
      AINERT=0.0
      ASIGMA=.10
      TC=8.64E4
      RAD=1.0/.01745329
      PY=180.0/RAD
      RGEO=6375.445
      EPSD=0.0
      WPLANT=350.0
      POWERH(1)=0.0
      POWER(1)=0.0
```

```
      ALPHA(1)=1.0
      PPK=0.0
      WEJECT=0.0
      TANK=.03
      L=1
      EPST(L)=0.0
      D=16.0
      B=.842
      NT=1
      NA=1
      NET=1
      NW=1
      NP=1
      NC=3
      NR=0
      JP=1
      AZERO=.5E-6
      AFINAL=1.E-6
      XMIN = 0.0
      YMIN=0.0
      XMAX=3200.0
      YMAX=70E3
      NHL=7
      NSBH=5
      NVL=8
      NSBV=10
      TIMEON=9999999.
      RT=.4
      VA=0.0
      VB=0.0
      SKIPA=1.0
      SKIPB=1.0
      DO 2 MB=1,16
2     VC(MB)=7.75 + MB*3
      DELV1=.5
      DELV2=.5
      RP1=1.05
      IPRINT=0
1     IF(ALPHA(1) .EQ. 0.0) GO TO 5
      WRITE(6,1002)
1002  FORMAT(1H1)
5     MPLAN2=NPLAN2
      MBIRD=NBIRD
6     CALL INPUT( 6HNT     ,NT     ,6HNA     ,NA     ,6HD      ,D      ,
     16HTIME   ,TIME   ,6HALPHA  ,ALPHA  ,6HRP1    ,RP1    ,6HRP2    ,RP2    ,
     26HHOME   ,HOME   ,6HLOW    ,LOW    ,6HMODE   ,MODE   ,6HFLYBY  ,FLYBY  ,
     36HORBIT  ,ORBIT  ,6HHIGH   ,HIGH   ,
     *6HARRIVE,ARRIVE,6HDEPART,DEPART,6HTARGET,TARGET,6HEPSD   ,EPSD   ,
     A6HEPST   ,EPST   ,6HPRINT  ,IPRINT,6HNET    ,NET    ,6HB      ,B      ,
     B6HDELV1  ,DELV1  ,6HDELV2  ,DELV2  ,6HDISP   ,DISP   ,6HAISP   ,AISP   ,
     D6HDSIGMA,DSIGMA,6HASIGMA,ASIGMA,6HBIRD   ,BIRD   ,6HLAUNCH,LAUNCH,
     E  6HESCAPE,ESCAPE,6HPARK   ,PARK   ,6HPPK    ,PPK    ,
     F6HVA     ,VA     ,6HVB     ,VB     ,6HVC     ,VC     ,6HPAYUP  ,PAYUP  ,
     G6HWPLANT,WPLANT,6HWEJECT,WEJECT,6HPOWER  ,POWERH,6HNW     ,NW     ,
     H 6HNP    ,NP     ,                6HYMAX   ,YMAX   ,6HANGLE  ,ANGLE  ,
     I6HXMAX   ,XMAX   ,6HNHL    ,NHL    ,6HNSBH   ,NSBH   ,6HNVL    ,NVL    ,
     J6HNSBV   ,NSBV   ,6HXMIN   ,XMIN   ,6HYMIN   ,YMIN   ,6HTANK   ,TANK   ,
     K6HSKIP1  ,SKIPA  ,6HSKIP2  ,SKIPB  ,6HENERGY,ENERGY,6HTIMEON,TIMEON,
     L6HDINERT,DINERT,6HAINERT,AINERT,6HNAME   ,NAME   ,6HDATA   ,DATA   ,
     M6HGRAPH  ,GRAPH  ,6HHELP   ,HELP   ,6HMATCH  ,MATCH  ,6HSPHERE,SPHERE,
     N6HASYMPT,ASYMPT,6HATOMIC,ATOMIC,6HSOLAR  ,SOLAR )
      IF(ALPHA(1) .EQ. 0.0) GO TO 7
      WRITE(6,1001)
1001  FORMAT(1H //)
7     DO 3  K=1,NT
3     T(K)=TIME(K)
      IF(PAYUP(2) .EQ. 0.0) GO TO 8
      DO 4 MB=1,16
      VC7(MB)=VC(MB)
4     PAY7(MB)=PAYUP(MB)
      GO TO 11
8     DO 9  MB=1,16
      VC7(MB)=VC(MB)
9     PAY7(MB)=PAYUP(1)
11    CONTINUE
      IF(VA .NE. 0.0) DELV1=0.0
      IF(VB .NE. 0.0) DELV2=0.0
      CALL PLOT1(NSCALE,NHL,NSBH,NVL,NSBV)
      CALL PLOT2(IMAGE,XMAX,XMIN,YMAX,YMIN,800)
      SKIP1=SKIPA
      SKIP2=SKIPB
      ARRML=1.0
      DEPML=1.0
      XMUL=1.0
      XMUW=0.0
      TIMON=TIMEON/24.
```

```
      IF(LAUNCH) DEPART=HIGH
      XLAUNC=PARKIN
      IF(LAUNCH) XLAUNC=ESCAP
      XMODE=FLBY
      IF(MODE)XMODE=ORBTR
      YLEVEL=YHIGH
      IF(DEPART) YLEVEL=YLOW
      XLEVEL=XHIGH
      IF(ARRIVE) XLEVEL=XLOW
      IF(DEPART) SKIP1=0.0
      IF(ARRIVE) SKIP2=0.0
      DO 10  NPLAN1=1,11
      IF (HOME(1).EQ.PLAN(NPLAN1)) GO TO 20
10    CONTINUE
15    WRITE(6,16)
16    FORMAT(2X,32HINPUT PLANET SPELLED INCORRECTLY)
      GO TO 1
20    DO 30 NPLAN2=1,11
      IF (TARGET(1).EQ.PLAN(NPLAN2)) GO TO 40
30    CONTINUE
      GO TO 15
40    CONTINUE
      IF(NPLAN2.EQ.10 .OR. NPLAN2 .EQ. 11) GO TO 41
      GO TO 42
41    MODE=FLYBY
      XMODE=RENDEZ
42    CONTINUE
      DO 433  NBIRD=1,13
      DO 438  JW=1,3
      IF(BIRD(JW).NE. VEHIK(JW,NBIRD)) GO TO 433
438   CONTINUE
      GO TO 435
433   CONTINUE
      WRITE(6,436)
436   FORMAT(2X,34HINPUT BOOST VEHICLE NOT IN STORAGE)
      GO TO 1
435   IF(MPLAN2.EQ.NPLAN2.AND.MBIRD.EQ.NBIRD.AND.ALPHA(1).EQ..0)GOTO2004
2005  WRITE(6,2000) HOME,TARGET,XMODE
2000  FORMAT(1H ,40X,2A4,2X,2HTO,4X,4A4,2X,A8/)
      WRITE(6,2040) BIRD,XLAUNC
2040  FORMAT(1H ,45X,4A4,2X,12HLAUNCH TO    ,A8/)
      IF (MODE) GO TO 2002
      WRITE(6,2001) YLEVEL
2001  FORMAT(1H ,50X,6HDEPART,3X,A8/)
      GO TO 2011
2002  WRITE(6,2003) YLEVEL,XLEVEL
2003  FORMAT(1H ,45X,6HDEPART,3X,A8, 8X,6HARRIVE,3X,A8/)
2011  IF(NBIRD.NE.1) GO TO 2030
      WRITE(6,2009)
2009  FORMAT(1H ,4HTIME,3X,5HALPHA,2X,7HPAYLOAD,2X,3HMUP,2X,3HMUW,
     13X,3HMLE,3X,5HDEP L,2X,5HARR L,
     Z3X,4HPBAR,5X,1HC,4X,5HT POW,2X,5HVINF1,2X,5HVINF2,
     2  3X,2HTC,4X,2HTH,4X,3HETA/)
      GO TO 2004
2030  WRITE(6,2031)
2031  FORMAT(1H ,4HTIME,2X,5HALPHA,2X,7HPAYLOAD,3X,3HMUP,2X,3HMUW,
     12X,3HMLE,2X,4HDEPL,2X,4HARRL,
     Z2X,5HPOWER,5X,1HC,4X,5HT POW,2X,5HVINF1,2X,5HVINF2,2X,6HBOOSTL,
     2  3X,2HTC,2X,2HTH,5X,3HETA/)
2004  CONTINUE
C   *******************************************************************ANALYSES
      K=1
2016  JP=1
2014  J=1
2013  L=1
2015  TOTAL1=0.0
      PAYSUM=10.0
      TLV1=0.0
      VINF1=VA
      VINF2=VB
      DMESH=2.0
2012  A2=RP2/(1.0-EPST(L))
      A1=RP1/(1.0-EPSD)
      IF(DEPART) SKIP1=0.0
      IF(ARRIVE) SKIP2=0.0
2044  NPASS=0
2045  IF(VINF1.LT.0.0) GO TO 726
      IF(VINF2.LT.0.0) GO TO 728
2050  IF(MATCH) NPASS=10
      CALL DPART          (DEPART,HIGH,LOW,DISP,GE,EPSD,RAD,DSIGMA,
     1RGEO,GMEO,RP1,DEPML,P,Q1,XJDBAR,DM,TDBAR,A1,VINF1,B,
     2   LAUNCH,NBIRD,BOOSTL,VC7,PAY7,DINERT)
      IF (MODE) GO TO 203
202   CALL FLYBUY            (NPLAN2,THBAR,THPBAR,PTK,
     1  HA,HB,HC,VINF1,DELV1,ANGLE,GMEO,AZERO,VASS1,SKIP1,NPASS,ENERGY)
      Q2=1.0
```

```
           Q=0.0
           CM=0.0
           TCBAR=0.0
           XJCBAR=0.0
           GO TO 114
  203      CALL ORBITR          (NPLAN2, THBAR, THPBAR, PTK, GM, RG,
          1  HA,HB,HC,VINF1,VINF2,DELV1,DELV2,AZERO,AFINAL,VASS1,VASS2,
          2SKIP1,SKIP2,NPASS,GMEO,ENERGY)
  3333     CALL ARRIV       (ARRIVE,NPLAN2,        P,Q,Q1,Q2,TCBAR,
          1 XJCBAR, CM, GM, EPST, TLTSI, AISP, GE,
          2  ASIGMA,RP2,A2,ARRML,VINF2 ,L,RG,B)
  114      AZUSED=AZERO
           AFUSED=AFINAL
           IF(IPRINT .EQ. 2) GO TO 113
           GO TO 112
  113      WRITE(6,9998) NPASS,VINF1,VINF2,TOTALL,BOOSTL,ALPHA(J),ARRML,
          1 XMUL,VASS1,VASS2,AZERO,AFINAL
  9998     FORMAT(2X,I2,2X,2F5.2,2X,2F10.4,2X,1F4.1,2X,2F7.5,2X,2F8.5,2X,
          12F15.10)
  112      IF(DEPML.LE.0.0) GO TO 720
           IF(ARRML .LE. 0.0) GO TO 722
           IF(BOOSTL .LE. 0.0    .AND.MODE .AND. .NOT. ARRIVE) GO TO 621
           IF(BOOSTL .LE. 0.0) GO TO 724
           IF(BOOSTL .LE. WEJECT .AND.MODE .AND. .NOT. ARRIVE) GO TO 621
           IF(BOOSTL.LE.WEJECT) GO TO 718
           IF(ALPHA(J).EQ.0.0) GO TO 690
C  *****************************************     MINIMUM  J  OPTIMIZATION
  115      THI(1)=0.45*T(K)
           PART2=Q1*DM*XJDBAR
           PART3=Q2*CM*XJCBAR
           PART6=1.0/(1.0-DM)
           PART7=1.0/(1.0-CM)
  116      DO 120  N=1,2
           IF(THI(N) .LE. 0.0) GO TO 716
           PART1=EXP(HA)*THI(N)**(HB+HC*ALOG(THI(N))-1.0)
           PART4=2.0*HC*ALOG(THI(N))
           PART5=1.0/((PART4+HB)*PART1)
           TTD(N)=P*ABS(PART2*PART5)**PART6
           TTC(N)=Q*ABS(PART3*PART5)**PART7
           FTH(N) = T(K) - TTD(N) - TTC(N)
           THI(2) = FTH(1)
  120      CONTINUE
           IF (P.EQ.0.0.AND.Q.EQ.0.0) GO TO 128
           IF (THI(2).EQ.THI(1)) GO TO 129
            EMM = (FTH(2) - FTH(1)) / (THI(2) - THI(1))
           TH = (FTH(1) - EMM*THI(1)) / (1. - EMM)
           TH=ABS(TH)
           IF(ABS(1.-(TH/THI(1)))-.0001) 124,124,122
  122      CONTINUE
           THI(1)=TH
           GO TO 116
  128      THI(2)=T(K)
  129      TH=THI(2)
C  *****************************************     COMPUTATION  TD, TC, TH, TP
  124      TD = TTD(2)
           TCAP=TTC(2)
           VARYJ=.0001667*(ALPHA(J) - 5.0)
           IF(SKIP .EQ. 0.0 ) VARYJ=0.0
           XJH=EXP(HA+HB*ALOG(TH)+HC*ALOG(TH)*ALOG(TH))
           XJH=XJH*(1.0 + XJH*VARYJ)
           THP=THPBAR*(TH/THBAR)**PTK
           THP=THP*(1.0 + XJH*VARYJ)**(1/3)
           TP= TD + THP + TCAP
           XJD = 0.
           IF(P.EQ.0.0) GO TO 130
  131      XJD=XJDBAR*((TD/TDBAR)**DM)
  130      XJC=0.
           IF(Q.EQ.0.0) GO TO 135
  134      IF(TCAP .LE. 0.0) GO TO 133
           XJC=XJCBAR*((TCAP/TCBAR)**CM)
           GO TO 135
  133      TCAP=0.0
           XJC=0.0
  135      XJMIN = XJD + XJH + XJC
C  *****************************************     SYSTEM  ANALYSIS
  136      GAMMA = SQRT(ALPHA(J)*XJMIN / 2000.)
           GAMMA2=GAMMA*GAMMA
           Z=B*(1.0 + TANK) + (GAMMA2*D*D)/(.0864*XJMIN*TP)
           IF((GAMMA/(Z**0.5)).GT.1.0) GO TO 648
           XMU1 = 1. - GAMMA / (Z**0.5)
           C=SQRT(0.0864*XJMIN*TP/GAMMA2*Z*XMU1)
           XMUW=2.0*GAMMA/B*Z**.5 - GAMMA*(1.0+TANK)/Z**.5 - GAMMA2/B
           ETA=B/(1.0 + (D/C)**2)
           AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
           AFINAL=AZERO/XMU1
           TERMA= 1.0-AZERO*TP*8.64E4/C
```

```
      IF(TERMA .LE. 0.0) GO TO 648
      TERMB=(1.0-SQRT(TERMA))**2
      TERMC=AZERO*(T(K)-TP)*(RT/C)*ALOG(TERMA)*8.64E4
      CL=(C*C/AZERO)*(TERMB-TERMC)
      AOPT=AZERO
      CLOPT=CL
      TOPT=TP
      IF(POWERH(1) .NE. 0.0 .AND. PPK .EQ. 0.0) GO TO 137
      IF(NPASS.LE.2) GO TO 2050
      XMUL = 1. - 2.*GAMMA*(Z**0.5)/B + (GAMMA2/B)
      GO TO 651
  648 IF(P .EQ. 1.0) GO TO 650
      TOTALL=.00001 + .00001*(VINF1 + VINF2)
      GO TO 630
C  *********************************************  SPECIFIED POWERPLANT
  137 WPLANT     =ALPHA(J)*POWERH(JP)
      XMUW=WPLANT     /((BOOSTL-WEJECT)*DEPML)
      IF(XMUW .GT. 1.0) GO TO 712
      GAMMA2=ALPHA(J)*XJMIN/2000.
      CCA=.5 + .5*GAMMA2/(B*XMUW)
      MT=0
  501 CCB=B*B*XJMIN*TP*.0864*XMUW*XMUW/(GAMMA2*GAMMA2)
      MT=MT + 1
      CCD=D*D/(CCA*CCA*CCB)
      IF(CCD.GT.1.0) GO TO 648
      CSQUA =CCA*CCB*(1.0 + SQRT(1.0-CCD))
      IF(CSQUA.LT.(D*D)) GO TO 648
      CSQUAR=CSQUA-(D*D)
      C=SQRT(CSQUAR)
      ETA=B/(1.0 + (D/C)**2)
      AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
      CL=CLOPT*(.9 + .1*(AOPT/AZERO))
  505 TERMA= 1.0-AZERO*TP*8.64E4/C
      IF(TERMA .LE. 0.0) GO TO 648
      IF(TP.EQ.T(K)) GO TO 516
      TERMB=(1.0-SQRT(TERMA))**2
      TERME=-C*(1.-SQRT(TERMA))*8.64E4/SQRT(TERMA)
      TERMG=ALOG(TERMA) + (T(K)-TP)*AZERO*8.64E4/(C*TERMA)
      TERMD=TERME-TERMG*RT*C*8.64E4
      TERMH=(T(K)-TP)*ALOG(TERMA)
      TERMF=CL-(C*C/AZERO)*TERMB + TERMH*RT*C*8.64E4
      TP1=TP - TERMF/TERMD
      IF(SKIP .EQ. 0.0) GO TO 650
      IF(ABS(1.-TP1/TP) .LE. .001) GO TO 510
      TP=TP1
      IF(TP .GT. T(K)) TP=T(K)
      GO TO 505
  510 IF(MT .GT. 1) GO TO 515
      TP=TP1
      GO TO 501
  515 ETA= B / (1. + (D/C)**2)
      AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
      AFP=AZERO
      TP=TP1
      TERMA= 1.0-AZERO*TP*8.64E4/C
      IF(TERMA .LE. 0.0) GO TO 648
      TERMB=(1.0-SQRT(TERMA))**2
      TERMC=AZERO*(T(K)-TP)*(RT/C)*ALOG(TERMA)*8.64E4
      AFINAL=AZERO/TERMA
      CL=(C*C/AZERO)*(TERMB-TERMC)
      CLFP=CL
      TPFP=TP
      IF(TIMON  .LE. TP) GO TO 520
  516 IF(NPASS.LE.2) GO TO 2050
      GO TO 650
C   *********************************************CONSTRAINED THRUSTING TIME
  520 MTT=0
      C=C*TIMON/TP
 521  ETA=B/(1. + (D/C)**2)
      AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
      CL=CLFP*(.85 + .15*(AFP/AZERO)*(TIMON/TPFP))
      TP=TIMON
  525 MTT=MTT + 1
      CC2=C
      MTTT=0
  530 MTTT=MTTT + 1
      ETA=B/(1. + (D/C)**2)
  535 CD=C*C + D*D
      ZA=2.E-3*XMUW*B/ALPHA(J)
      TERMM=1.0 - TP*ZA*8.64E4/CD
      IF(TERMM .LE. 0.0) GO TO 648
      TERMR=(1. - SQRT(TERMM))**2
      FOFC=CL - CD*C*TERMR/ZA + RT*8.64E4*C*(T(K)-TP)*ALOG(TERMM)
      TERMN=-(3.0*C*C + D*D)*TERMR/ZA
      TERMO=-2.*C*C*TP*8.64E4*(1.0-1.0/SQRT(TERMM))/CD
      TERMS=C*C*8.64E4*TP*2.0*ZA/(TERMM*CD*CD)
```

```
      TERMQ=ALOG(TERMM) + TERMS
      TERMP=RT*8.64E4*(T(K)-TP)*TERMQ
      FDOTC=TERMN + TERMO + TERMP
      CC1=C - FOFC/FDOTC
      IF(ABS(1. - CC1/C) .LE. .001) GO TO 540
      C=CC1
      IF(C .LT. D) GO TO 648
      GO TO 530
  540 C=CC1
      ETA=B/(1. + (D/C)**2)
      ABEFOR=AZERO
      AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
      CL=CLFP*(.85 + .15*(AFP/AZERO)*(TIMON/TPFP))
      IF(ABS(1.-AZERO/ABEFOR) .LE. .001) GO TO 555
      GO TO 525
  555 ETA=B/(1. + (D/C)**2)
      AZERO=2.E-3*XMUW*ETA/(ALPHA(J)*C)
      CL=CLFP*(.85 + .15*(AFP/AZERO)*(TIMON/TPFP))
      XMU1=1. - AZERO*TP*8.64E4/C
      IF(XMU1 .LE. XMUW) GO TO 648
      AFINAL=AZERO/XMU1
      IF(NPASS .LE. 2) GO TO 2050
      GO TO 650
C ********************************************* SYSTEM  PARAMETERS
  650 ETA =  B / (1. + (D/C)**2)
      XMU1=1. - AZERO*TP*8.64E4/C
      XMUL=XMU1*(1.0+ TANK) - XMUW - TANK
  651 ETA =  B / (1. + (D/C)**2)
      XMUP=1.0 - XMU1
      XMUT=TANK*XMUP
      TOTALL=DEPML*ARRML*(BOOSTL-WEJECT)*XMUL-AINERT
      IF(TOTALL.LE.0.0) TOTALL=.00001 + .00001*(VINF1 + VINF2)
      IF(XMUL.LT. 0.0) GO TO 630
      PBAR=XMUW*DEPML*(BOOSTL-WEJECT)/(ALPHA(J)*TOTALL)
      POWERR=PBAR*TOTALL
      IF(PPK.EQ.0.0) GO TO 630
  628 ALPHAC=WPLANT    *POWERR**PPK
      IF(ABS(1.0-ALPHA(J)/ALPHAC)-.000100)  630,630,629
  629 ALPHA(J)=ALPHAC
      GO TO 124
C ****************************************    HYPERBOLIC VELOCITY OPTIMIZATION
  630 IF(P.EQ.1.0.AND.Q.EQ.1.0)  GO TO 700
C    CARD 630  SCREENS  FOR ORBITER LOW-LOW
  631 IF(MODE) GO TO 610
C  CARD 631  SCREENS FOR  ORBITERS
  632 IF(P.EQ.1.0) GO TO 700
C    CARD  632 SCREENS  FOR FLYBY LOW DEPART
  605 IF(ABS(1.0-TOTAL1/TOTALL).LE..00002) GO TO 660
      IF(TOTALL-TOTAL1) 602,660,600
  600 TOTAL1=TOTALL
      VINF1=VINF1+DELV1*DMESH
      AZ=AZUSED
      AF=AFUSED
      ALPHAX=ALPHA(J)
      GO TO 2050
  602 VINF1=VINF1-DELV1*DMESH
      AZERO=AZ
      AFINAL=AF
      IF(PPK .NE. 0.0) ALPHA(J)=ALPHAX
      NPASS=2
      GO TO 2045
  610 IF(P.EQ.0.0.AND.Q.EQ.1.0) GO TO 605
C    CARD 610 SCREENS FOR ORBITER HIGH-LOW
  611 IF(P.EQ.0.0.AND.Q.EQ.0.0) GO TO 615
C    CARD 611 SCREENS FOR ORBITER  HIGH-HIGH
      IF(ABS(1.0-TOTAL1/TOTALL).LE..00002) GO TO 660
  612 IF(TOTALL-TOTAL1) 613,660,614
C    CARD 612 HANDLES ORBITER LOW-HIGH
  614 TOTAL1=TOTALL
      VINF2=VINF2+DELV2*DMESH
      AZ=AZUSED
      AF=AFUSED
      ALPHAX=ALPHA(J)
      GO TO 2050
  613 VINF2=VINF2-DELV2*DMESH
      AFINAL=AF
      AZERO=AZ
      IF(PPK .NE. 0.0) ALPHA(J)=ALPHAX
      NPASS=2
      GO TO 2045
  615 IF(XMUL .LT. 0.0) GO TO 621
      IF(ABS(1.0-TLV1/TOTALL  ).LE..00002)GO TO 619
      IF(TOTALL-TLV1) 618,619,620
C    CARD 615  IS  FOR  ORBITER HIGH-HIGH FOR FIXED VINF2
  620 VINF1=VINF1 + DELV1*DMESH
      TLV1=TOTALL
```

```
      AZ=AZUSED
      AF=AFUSED
      ALPHAX=ALPHA(J)
      GO TO 2050
  618 VINF1=VINF1-DELV1*DMESH
      AZERO=AZ
      AFINAL=AF
      IF(PPK .NE. 0.0) ALPHA(J)=ALPHAX
      NPASS=2
      GO TO 2045
  619 IF(ABS(1.0-TOTAL1/TOTALL).LE..00002)GO TO 660
      IF(TOTALL-TOTAL1) 622,660,621
C     CARD 619 IS FOR ORBITER HIGH-HIGH  WITH VARIATION ON VINF2
  621 TOTAL1=TOTALL
      VINF1A=VINF1
      VINF2A=VINF2
      AZ2=AZ
      AF2=AF
      ALPHAY=ALPHA(J)
      VINF1=VINF1A-2.0*DELV1*DMESH
      IF(VINF1.LE.VA) VINF1=VA
      VINF2=VINF2 + DELV2*DMESH
      AZERO=AZ
      AFINAL=AF
      TLV1=0.0
      GO TO 2044
  622 TLV1=TOTAL1
      VINF1=VINF1A
      VINF2=VINF2A
      AZERO=AZ2
      AFINAL=AF2
      IF(PPK .NE. 0.0) ALPHA(J)=ALPHAY
      NPASS=2
      GO TO 2045
  660 IF(DMESH.EQ.1.0) GO TO 700
      DMESH=1.0
      TLV1=0.0
      TOTAL1=0.0
      VINF1=VINF1-DELV1
      IF(VINF1.LE.VA) VINF1=VA
      VINF2=VINF2-DELV2
      IF(VINF2.LE.VB) VINF2=VB
      GO TO 2044
  670 POWERH(1)=POWERR
      POWER(1)=1.0
      GO TO 2015
  690 TOTALL=DEPML*ARRML*(BOOSTL-WEJECT)*XMUL-AINERT
C   BALLISTIC SYSTEM PRINTOUT
      WRITE(6,689) T(K),TOTALL,DEPML,ARRML,VINF1,VINF2,BOOSTL
  689 FORMAT(1H ,F5.0,7X,F10.3,15X,F5.3,1X,F5.3,25X,F4.1,3X,F4.1,2X,F7.0
     1)
      PAYSUM=10.0
      GO TO 805
  700 IF(XMUL.LT. 0.0) GO TO 730
      IF(VINF1.EQ.0.0)  VINF1=0.0
      IF(VINF2.EQ.0.0) VINF2=0.0
      IF(POWERH(1).EQ.0.0.AND.PPK.EQ.0.0.AND.TIMON.LE.TP) GO TO 670
      IF(POWER(1) .EQ. 1.0) POWERH(1)=0.0
      TP=TP*24.0
      TOTALL=DEPML*ARRML*(BOOSTL-WEJECT)*XMUL-AINERT
  691 IF(BOOSTL.NE.1.0) GO TO 693
C  ***************************************************************    PRINT OUT
      WRITE(6, 692)T(K),ALPHA(J),TOTALL,XMUP,XMUW,XMUL,DEPML,ARRML,
     1PBAR,C,TP,VINF1,VINF2,TCAP,TH,ETA
  692 FORMAT(1H ,F5.0,2X,F5.1,2X,F5.4,2X,F4.3,1X,F5.4,1X,F6.4,2X,F5.3,
     12X,F5.3,2X,F6.4,2X,F5.1,2X,F6.0,2X,F4.1,2X,F4.1,2X,
     2F5.1,1X,F5.0,2X,F4.3)
      GO TO 695
  693 WRITE(6, 694)T(K),ALPHA(J),TOTALL,XMUP,XMUW,XMUL,DEPML,ARRML,
     1POWERR,C,TP,VINF1,VINF2,BOOSTL,TCAP,TH,ETA
  694 FORMAT(1H ,F5.0,1X,F5.1,1X,F7.0,2X,F4.3,1X,F4.3,1X,F4.3,1X,F5.3,
     11X,F5.3,2X,F6.1,2X,F5.1,2X,F6.0,2X,F4.1,2X,F4.1,2X,F7.0,2X,
     2F4.0,1X,F5.0,2X,F4.3)
  695 IPC=ALPHA(J)/10.0+.05
      BCD=CHAR(IPC)
      NDATA=1
      TPLOT=T(K)
      TLPLOT=TOTALL
      CALL PLOT3(BCD,TPLOT,TLPLOT,NDATA)
  805 L=L + 1
      IF(L.GT.NET) GO TO 801
      GO TO 2015
  800 PAYSUM=-1.0
  801 L=1
  804 IF(PPK .NE. 0.0) GO TO 807
      J=J + 1
```

```
      IF(J.GT.NA) GO TO 802
      IF(ALPHA(J).GT.ALPHA(J-1).AND.PAYSUM.LT.0.0) GO TO 802
      GO TO 2013
  802 J=1
  806 JP=JP + 1
      IF(JP.GT.NP) GO TO 803
      IF(POWERH(JP) .GT. POWERH(JP-1) .AND. XMUW .GT.1.0) GO TO 803
      GO TO 2014
  803 JP=1
  807 K=K + 1
      IF(PPK .NE. 0.0) ALPHA(1)=1.0
      IF(K.GT.NT) GO TO 808
      IF(T(K).LT.T(K-1).AND.PAYSUM.LT.0.0) GO TO 808
      WRITE(6,809)
  809 FORMAT(1H )
      GO TO 2016
  808 K=1
      PAYSUM=10.0
      IF(IPRINT.EQ.1) GO TO 302
      GO TO 1
  302 WRITE(6,300)TARGET,XMODE
  300 FORMAT(1H1,45('*'),4A4,A8,30('*'))
  304 WRITE(6,305)BIRD,XLAUNC
  305 FORMAT(1X,4A4,2X,11HLAUNCH TO   ,A8)
      IF(LAUNCH) GO TO 318
      WRITE(6,306) YLEVEL
  306 FORMAT(1X,A8,13HTHRUST ESCAPE)
  307 IF(.NOT.MODE) GO TO 320
      WRITE(6,308) XLEVEL,RP2,EPST(L)
  308 FORMAT(1X,A8,14HTHRUST CAPTURE,3X,4HRP2=,F5.1,2X,4HECC=,F3.2)
  310 IF(PPK.NE.0.0) GO TO 311
      IF(POWERH(1).EQ.0.0) GO TO 313
      IF(TIMON .LE. TP) GO TO 324
      WRITE(6,323) POWERH(JP)
  323 FORMAT(1X,6HPOWER=,F5.0,1X,3HKWE,65X,19HOPTIMUM THRUST TIME)
      GO TO 317
  324 WRITE(6,315) POWERH(JP),TIMEON
  315 FORMAT(1X,6HPOWER=,F5.0,1X,3HKWE,49X,24HTHRUST TIME UPPER LIMIT=,
     1F6.0, 5HHOURS)
      GO TO 317
  318 WRITE(6,319)
  319 FORMAT()
      GO TO 307
  320 WRITE(6,319)
      GO TO 310
  311 PPKDIF = 1.0 + PPK
      WRITE(6,312) WPLANT,PPKDIF
  312 FORMAT(1X,11HPLANT MASS=,F5.0,8H*POWER**,F3.2)
      GO TO 317
  313 WRITE(6,314)
  314 FORMAT(1X,13HOPTIMUM POWER,67X,19HOPTIMUM THRUST TIME)
  317 CALL PLOT4(31,31H     NET  SPACECRAFT  MASS   KG)
      WRITE(6,301)
  301 FORMAT(45X,18HMISSION TIME, DAYS)
      WRITE(6,322) NAME,DAT
  322 FORMAT(1X,6HFIGURE,69X,3A4,2X,2A4)
  151 GO TO 1
  712 WRITE(6,713)
  713 FORMAT(2X,21HXMUW GREATER THAN 1.0)
      GO TO 806
  716 WRITE(6,717)
  717 FORMAT(2X,38HTHI(N) LESS THAN 0.0,CHECK TIME INPUTS)
      GO TO 805
  718 WRITE(6,719)
  719 FORMAT(2X,23HBOOSTL LESS THAN WEJECT)
      GO TO 800
  720 WRITE(6,721)
  721 FORMAT(2X,14HNEGATIVE DEPML)
      GO TO 800
  722 WRITE(6,723)
  723 FORMAT(2X,14HNEGATIVE ARRML)
      GO TO 800
  724 WRITE(6,725)
  725 FORMAT(2X,15HNEGATIVE BOOSTL)
      GO TO 800
  726 WRITE(6,727)
  727 FORMAT(2X,19HVINF1 LESS THAN 0.0)
      GO TO 805
  728 WRITE(6,729)
  729 FORMAT(2X,19HVINF2 LESS THAN 0.0)
      GO TO 805
  730 WRITE(6,731)
  731 FORMAT(2X,13HNEGATIVE XMUL)
      GO TO 800
      END
```

```
                 C        MPX04F5              DEPART   E-9
                 C
                 C
0001                   SUBROUTINE DPART    (DEPART,HIGH,LOW,DISP,GE,EPSD,RAD,DSIGMA,
                      1RGEO,GMEO,RP1,DEPML,P,Q1,XJDBAR,DM,TDBAR,A1,VINF1,B,
                      2   LAUNCH,NBIRD,BOOSTL,VC7,PAY7,DINERT)
0002                   DIMENSION SIS4V(16), T3FCP(16), T3FCV(16), T3FV(16), T3FP(16)
0003                   DIMENSION SVP(16), SVV(16), SIS4CP(16), SIS4CV(16), SIS4P(16)
0004                   DIMENSION PAY7(16), VC7(16)
0005                   DIMENSION SVCV(16), SVCP(16), SICV(16), SICP(16), T3DAV(16)
0006                   DIMENSION T3DAP(16), ACV(16), ACP(16), AAV(16), AAP(16)
0007                   DIMENSION T3DCV(16),  T3DCP(16)
0008                   DATA SVV/7.0, 7.75, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 15.5,
                      1  16.0, 16.2, 16.3, 16.4, 16.8, 30.0  /
0009                   DATA SVP /152000., 120000., 82000., 60000., 43000., 30500.,
                      1 21000., 12800., 6900., 4000., 1300., 400., 130.,30., 0., 0. /
0010                   DATA SIS4CV/7.,7.75,9., 10., 11., 12., 13., 14., 15., 16., 17.,
                      1  18., 19., 20., 21., 30. /
0011                   DATA SIS4CP/89000., 67000., 46000., 32000., 22800., 16000.,
                      1 11700., 8200., 5800., 4000., 2700., 1700., 890., 370.,100.,0. /
0012                   DATA SIS4V/7.,7.75, 9., 10., 11., 11.5, 12., 12.5, 12.8, 13.,
                      1 13.2, 13.4, 13.5, 13.6, 13.8, 30. /
0013                   DATA SIS4P/80000., 62000., 41000., 27500., 16600., 12300.,
                      1  8400., 5400., 3800., 2800., 1970., 1140., 680., 250., 30.,0. /
0014                   DATA T3FCV/7.,7.75, 9., 10., 11., 12., 13., 14., 15., 15.5,
                      1  16., 16.5, 16.8, 17., 17.5, 30. /
0015                   DATA T3FCP/30000., 23000., 15000., 10800., 7700., 5300., 3600.,
                      1  2300., 1380., 990., 680., 375., 200., 90., 0., 0. /
0016                   DATA T3FV/7., 7.75, 9., 10., 11., 11.5, 12., 12.2, 12.4, 12.6,
                      1  12.8, 13., 13.1, 13.2, 13.4, 30.6  /
0017                   DATA T3FP/ 25000., 18000., 10200., 6300., 3350., 2300., 1500.,
                      1  1200., 900., 630., 370., 150., 70., 20., 0.0, 0. /
0018                   DATA  SVCV / 7.0, 7.75, 9.0, 10., 12., 14., 16., 18., 20., 21.,
                      1 22., 22.5, 23., 23.25, 24., 30.0/
0019                   DATA SVCP / 152000.0, 120000.0, 82000.0, 62000.0, 34000.0,
                      1 18000., 9300., 4500., 2100., 1300., 600., 340., 125., 50.,0.,0./
0020                   DATA  SICV / 7.0, 7.75, 9.0, 10., 11., 12., 13., 13.5, 14., 14.5,
                      1  15., 15.8, 16.4, 17., 17.2, 30.0 /
0021                   DATA  SICP / 32200., 24000., 15000., 10440., 7080., 4760.0,
                      1  3130., 2450., 1910., 1408., 1044., 454., 136., 10., 0.0, 0.0/
0022                   DATA  T3DAV /  7.0, 7.75, 9., 10., 11., 12., 12.5, 13., 13.5, 14.,
                      1  14.4, 14.8, 15.2, 15.6, 16., 30.  /
0023                   DATA  T3DAP / 15880., 12100., 7530., 4950., 3040., 1838., 1360.,
                      1  1042., 741., 499., 317.,  186., 95.4, 45.4, 0.0, 0.0/
0024                   DATA  ACV /7.0, 7.75, 9.0, 10., 10.5, 11., 11.4, 11.8, 12.2,
                      1  12.4, 12.6, 12.8, 13., 13.3, 14., 30.0/
0025                   DATA   ACP / 7260., 5360., 3250., 2130., 1680., 1295., 1000.,
                      1  749.,  450., 330., 220., 118., 45., 10., 0.0, 0.0/
0026                   DATA   AAV / 7.0, 7.75, 9., 10., 10.4, 10.8, 11.2, 11.6,
                      1  12., 12.2, 12.4, 12.6, 12.8, 13., 14., 30.0/
0027                   DATA   AAP  / 5000., 3600., 2000., 1160., 910., 700., 530., 380.,
                      1   250., 192., 140., 80., 36., 17., 0.0,0.0/
0028                   DATA T3DCV / 7.0, 7.75, 9.0, 10.0, 11., 12.0, 13.0, 14.,
                      1 14.5, 15.0, 15.5, 16.0, 16.42, 17.0, 18.0, 30.0/
0029                   DATA  T3DCP  /  21600.0, 17000.0, 11000.,  7800.,  5500., 3800.,
                      1  2500., 1520., 1140., 790., 460., 240., 100., 50., 0.0, 0.0 /
0030                   LOGICAL DEPART,HIGH,LOW
0031                   LOGICAL LAUNCH, ESCAPE, PARK
0032                   BOOSTL=1.0
0033                   D1=0.0
0034                   IF(LAUNCH) GO TO 820
0035                   GO TO 821
0036              820  VBOOST=SQRT(VINF1*VINF1 + 120.1)
0037                   GO TO 800
0038              821  E100=(RP1*6375. - 6560.)/(RP1 * 6375. + 6560.)
0039                   A100 =(RP1*6375. + 6560.)/2.0
0040                   BIRDV1=SQRT(GMEO*(1.+E100)/((1.-E100)*A100))-7.75
0041                   BIRDV2=SQRT(GMEO*(1.+EPSD)/((1.-EPSD)*A1*RGEO))
                      1        -SQRT(GMEO*(1.-E100)/((1.+E100)*A100))
0042                   VBOOST=7.75 + BIRDV1 + BIRDV2
0043                   GO TO 800
0044              800  GO TO (801,802,803,804,805,806,807,808,809,810,811,812,813),NBIRD
0045              801  BOOSTL=1.0
0046                   VBOOST=7.75
0047                   GO TO 841
0048              802  CALL TAINT(SVV    ,SVP    ,VBOOST,BOOSTL,16,2,NERR1,D1)
0049                   GO TO 841
0050              803  CALL TAINT(SIS4CV,SIS4CP,VBOOST,BOOSTL,16,2,NERR1,D1)
0051                   GO TO 841
0052              804  CALL TAINT(SIS4V ,SIS4P ,VBOOST,BOOSTL,16,2,NERR1,D1)
0053                   GO TO 841
0054              805  CALL TAINT(T3FCV ,T3FCP ,VBOOST,BOOSTL,16,2,NERR1,D1)
0055                   GO TO 841
0056              806  CALL TAINT(T3FV  ,T3FP   ,VBOOST,BOOSTL,16,2,NERR1,D1)
0057                   GO TO 841
0058              807  CALL TAINT(VC7,PAY7,VBOOST,BOOSTL,16,2,NERR1,D1)
```

```
      GO TO 841
  808 CALL TAINT( SVCV , SVCP ,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  809 CALL TAINT(SICV , SICP  ,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  810 CALL TAINT( T3DAV,T3DAP ,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  811 CALL TAINT(   ACV,    ACP,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  812 CALL TAINT(   AAV,    AAP,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  813 CALL TAINT(T3DCV,T3DCP  ,VBOOST,BOOSTL,16,2,NERR1,D1)
      GO TO 841
  841 IF(LAUNCH) GO TO 505
      GO TO 504
  504 IF(DEPART) GO TO 503
      GO TO 501
  501 CONTINUE
      TLTSI=0.0
      VJET=DISP*GE
      EPS=EPSD
      TLTSR=TLTSI/RAD
      VINF=VINF1
      SIGMA=DSIGMA
      RD=RGEO
      GP=GMEO
      AS=RP1/(1.-EPS)
      PS=AS*RD*(1.-EPS*EPS)
      RLTS=PS/(1.+EPS*COS(TLTSR))
      VSU=SQRT((GP/(AS*RD))*(1.+2.*EPS*COS(TLTSR)+EPS*EPS)/(1.-EPS*EPS))
      DVI=SQRT((VINF**2)+(2.*GP/(RLTS)))-VSU
      BMFST=EXP((-DVI)/VJET)
      BPROP=(1.+SIGMA)*BMFST-SIGMA -DINERT/BOOSTL
      DEPML=BPROP
  505 P=0.0
      Q1=1.0
      XJDBAR=0.0
      DM=0.0
      TDBAR=0.0
      GO TO 596
  503 C=100.0
      D=20.0
      TC=8.64E4
      TDBAR=30.
      TDBAR=TDBAR*TC
      T=200.0*TC
      GP=GMEO
      EPS=EPSD
      RD=RGEO
      A1=RP1/(1.-EPS)
      AD=A1*RD
      VC=SQRT(GP/(A1*RD))
      U1=.9
  430 IF(U1.GT.1.0) U1=.999999
      P1=1.84*VC*((AD*AD*C/(GP*TDBAR))**.25)/C
      P2=(1.0/U1-1.0)**.25
      P3=(1.0/U1-1.0)**.75
      FU=ALOG(U1)+VC/C-(P1)*P2
      FUDOT=1.0/U1+P1/(P3*U1*U1*4.0)
      U2=U1-FU/FUDOT
      ETA=1./(1.+(D/C)**2)
      UALPHA=(1.0-U2)*1000.*C*C/(2.*ETA*TDBAR)
      IF(ABS(1.-(U1/U2))-.0001) 431,431,432
  432 U1=U2
      GO TO 430
  431 XJDBAR=2.0*ETA*UALPHA*1000.*(1.-U2)/U2
      U1=.9
  230 IF(U1.GT.1.0) U1=.999999
      P1=1.84*VC*((AD*AD*C/(GP*T))**.25)/C
      P2=(1.0/U1-1.0)**.25
      P3=(1.0/U1-1.0)**.75
      FU=ALOG(U1)+VC/C-(P1)*P2
      FUDOT=1.0/U1+P1/(P3*U1*U1*4.0)
      U2=U1-FU/FUDOT
      ETA=1./(1.+(D/C)**2)
      UALPHA=(1.0-U2)*1000.*C*C/(2.*ETA*T)
      IF(ABS(1.-(U1/U2))-.0001) 231,231,232
  232 U1=U2
      GO TO 230
  231 XJ=2.0*ETA*UALPHA*1000.*(1.-U2)/U2
      DM=(ALOG(XJDBAR/XJ))/(ALOG(TDBAR/T))
      IF(DM  .LT.0.0) GO TO 240
      DM=0.0
      XJDBAR=0.0
  240 TDBAR=TDBAR/TC
      P=1.0
      Q1=TDBAR**(-DM)
    1 CONTINUE
  596 RETURN
      END
```

```
C       MPX04F4                    FLYBY
C
C
        SUBROUTINE FLYBUY      (NPLAN2,THBAR,THPBAR,PTK,
       1  HA,HB,HC,VINF1,DELV1,ANGLE,GMEO,AZERO,VASS1,SKIP1,NPASS,ENERGY)
        LOGICAL ENERGY,ATOMIC,SOLAR
        D1=0.0
        C1=0.0
        C3=0.0
        VASS1=VINF1
        NPASS=NPASS + 1
        IF(SKIP1.EQ.0.0) GO TO 202
        IF(NPASS .GE. 10) GO TO 195
C     ********************************************   ASYMPTOTIC MATCHING
        XASS1=VINF1*VINF1*.25/SQRT(GMEO*AZERO)
        GOFX1=2.0*(XASS1+.651630)*(XASS1+4.113609)*(XASS1+1.214342)/(
       1  (XASS1+4.169068)*(XASS1+1.303312)*(SQRT(XASS1+1.0)))
  201   VASS1=GOFX1*(GMEO*AZERO)**.25
  202   GO TO (301,302,303,304,305,306,307,308,309,310,311),NPLAN2
  195   VASS1=SQRT(VINF1*VINF1 + 2.0*GMEO/(145.0000*6375.0))
C  **********************************  SPHERE OF INFLUENCE MATCHING
        GO TO 202
  301   THBAR=70.0
C  MERCURY
        HA=56.365662
        HB=-20.123871
        HC=1.848905
        DELV1=0.0
        THPBAR=43.796
        PTK=1.291965
        GO TO 500
  302   THBAR=50.0
C  VENUS
        HA=42.662186
        HB=-15.007878
        HC=1.328273
        DELV1=0.0
        THPBAR=27.5
        PTK=1.0
        GO TO 500
C  EARTH
  303   GO TO 1
C  MARS
  304   GO TO 1
  305   THBAR=300.0
C  JUPITER
        HA=56.788559
        HB=-15.133818
        HC=1.030477
        C1=-.002046
        C3=1.505424
        THPBAR=150.89
        PTK=1.056036
        GO TO 500
  306   THBAR=600.0
C  SATURN
        HA=54.259232
        HB=-13.236733
        HC=.827999
        C1=-.001736
        C3=1.368783
        THPBAR=321.034
        PTK=.887848
        GO TO 500
  307   THBAR=600.0
C  URANUS
        HA=40.394104
        HB=-8.366853
        HC=.438166
        C1=-.001075
        C3=1.439463
        THPBAR=329.991
        PTK=.907105
        GO TO 500
  308   THBAR=1500.
C  NEPTUNE
        HA=44.884598
        HB=-8.931028
        HC=.445309
        C1=-.000705
        C3=1.512190
        PTK=1.274
        THPBAR=786.
        GO TO 500
C  PLUTO
  309   GO TO 1
```

```
  310   THBAR=500.0
C  HALLEY'S  COMET RENDEZVOUS
        HA=-51.464508
        HB=18.605637
        HC=-1.553296
        C1=-.001243
        C3=.993871
        PTK=1.0
        THPBAR=450.0
        GO TO 500
  311   DIHI=2.0*57.29578*ARSIN(VASS1/59.54)
C  EXTRA-ECLIPTIC  RENDEZVOUS
        IF(DIHI .GT. ANGLE) DIHI=ANGLE
        DILO=ANGLE-DIHI
        VINC=DILO/10.0
        IF(.NOT. ENERGY) GO TO 411
        THBAR=400.0
        THPBAR=372.
        PTK=1.
        HA=.338472
        HB=2.247731
        HC=-.41
        C1=.0713
        C3=.335
        FACTOR=C1*VINC**C3
        GO TO 501
  411   HA=5.0
        HB=-1.004368
        HC=-.000119
        C1=2.315532
        C3=.443572
        THPBAR=180.+DILO
        THBAR=300.0
        PTK=1.0
        HA=HA+C1*VINC**C3
        GO TO 502
  1     WRITE(6,100)
  100   FORMAT(1H ,22HPLANET DATA NOT STORED)
  500   FACTOR=C1*VASS1**C3
  501   HC=HC+FACTOR
  502   RETURN
        END
```

```
C         MPX04F2                ORBITER
C
C
        SUBROUTINE ORBITR       (NPLAN2, THBAR, THPBAR, PTK, GM, RG,
      1  HA,HB,HC,VINF1,VINF2,DELV1,DELV2,AZERO,AFINAL,VASS1,VASS2,
      2SKIP1,SKIP2,NPASS,GMEO,ENERGY)
        LOGICAL ENERGY,ATOMIC,SOLAR
        D1=0.0
        C1=0.0
        C2=0.0
        C3=0.0
        C4=0.0
  203   GO TO (401,402,403,404,405,406,407,408,409),NPLAN2
  401   THBAR=80.
C  MERCURY
        HA=37.226028
        HB=-11.546978
        HC=.971943
        DELV1=0.0
        DELV2=0.0
        THPBAR=56.161
        PTK=1.156017
        GM=2.18E4
        RSPHER=46.
        RG=2420.
        GO TO 500
  402   THBAR=100.0
C  VENUS
        HA=42.995316
        HB=-14.002761
        HC=1.163274
        DELV1=0.0
        DELV2=0.0
        THPBAR=60.
        PTK=1.0
        GM=3.2485E5
        RSPHER=101.
        RG=6.05E3
        GO TO 500
C  EARTH
  403   GO TO 1
C  MARS
  404   GO TO 1
```

```
  405 THBAR=400.0
C  JUPITER
      HA=56.546051
      HB=-13.704782
      HC=.843276
      C1=-.0017318
      C3=1.258404
      THPBAR=257.175
      PTK=.895741
      RG=7.14E4
      RSPHER=674.
      GM=1.2671E8
      GO TO 500
  406 THBAR=700.
C  SATURN
      HA=46.593872
      HB=-9.870175
      HC=.525386
      C1=-.001485
      C3=1.145093
      THPBAR=437.583
      PTK=.908804
      RG=6.04E4
      RSPHER=905.
      GM=3.792E7
      GO TO 500
  407 THBAR=1200.
C  URANUS
      THPBAR=740.
      PTK=.88825
      HA=52.340347
      HB=-10.660658
      HC=.549718
      C1=-.001056
      C3=1.056167
      RG=2.35E4
      RSPHER=2210.
      GM=5.788E6
      GO TO 500
  408 THBAR=2000.
C  NEPTUNE
      HA=58.582687
      HB=-11.742033
      HC=.598062
      C1=-.000748
      C3=1.123747
      THPBAR=1220.
      PTK=.748
      RG=2.23E4
      RSPHER=3900.
      GM=6.8E6
      GO TO 500
C  PLUTO
  409 GO TO 1
  1   WRITE(6,100)
  100 FORMAT(2H , 22HPLANET DATA NOT STORED)
  500 VASS1=VINF1
      VASS2=VINF2
      NPASS=NPASS + 1
      IF(SKIP1.EQ.0.0) GO TO 499
      IF(NPASS .GE. 10) GO TO 195
C   *******************************************   ASYMPTOTIC MATCHING
      XASS1=VINF1*VINF1*.25/SQRT(GMEO*AZERO)
      GOFX1=2.0*(XASS1+.651 630)*(XASS1+4.113 609)*(XASS1+1.214342)/(
     1  (XASS1+4.169068)*(XASS1+1.303312)*(SQRT(XASS1+1.0)))
  201 VASS1=GOFX1*(GMEO*AZERO)**.25
  499 IF(SKIP2.EQ.0.0) GO TO 501
      IF(NPASS .GE. 10) GO TO 196
      XASS2=VINF2*VINF2*.25/SQRT(GM*AFINAL)
      GOFX2=2.0*(XASS2+.651 630)*(XASS2+4.113 609)*(XASS2+1.214342)/(
     1    (XASS2+4.169068)*(XASS2+1.303312)*(SQRT(XASS2+1.0)))
      GO TO 198
C  ************************************  SPHERE OF INFLUENCE MATCHING
  195 VASS1=SQRT(VINF1*VINF1 + 2.0*GMEO/(145.0000*6375.0))
      GO TO 499
  196 VASS2=SQRT(VINF2*VINF2 + 2.0*GM/(RSPHER *RG))
      GO TO 501
  198 VASS2=GOFX2*(GM*AFINAL)**.25
  501 FACTOR=C1*(VASS1 + VASS2)**C3
      HC=HC+FACTOR
      RETURN
      END
```

```
C       MPX04F3                    ARRIVE
C
C
      SUBROUTINE ARRIV (ARRIVE,NPLAN2,        P,Q,Q1,Q2,TCBAR,
     1 XJCBAR, CM, GM, EPST, TLTSI, AISP, GE,
     2  ASIGMA,RP2,A2,ARRML,VINF2 ,L,RG,B)
      LOGICAL MODE,FLYBY,ORBIT,ARRIVE,DEPART,HIGH,LOW
      DIMENSION EPST(20)
      RAD=1.0/.01745329
      IF (ARRIVE) GO TO 100
      GO TO 200
  100 C=100.0
      D=20.0
      TC=8.64E4
      TCBAR=30.0
      TCBAR=TCBAR*TC
      T=200.0*TC
      EPS=EPST(L)
      GP=GM
      A2=RP2/(1.-EPS)
      AD=A2*RG
      VC=SQRT(GP/(A2*RG))
      U1=.9
  430 IF(U1.GT.1.0) U1=.999999
      P1=1.84*VC*((AD*AD*C/(GP*TCBAR))**.25)/C
      P2=(1.0-U1)**.25
      P3=(1.0-U1)**.75
      FU=ALOG(U1)+VC/C-(P1)*P2
      FUDOT=1.0/U1+P1/(P3*U1*U1*4.0)
      U2=U1-FU/FUDOT
      ETA=1./(1.+(D/C)**2)
      UALPHA=(1.0-U2)*1000.*C*C/(2.*ETA*TCBAR)
      IF(ABS(1.-(U1/U2))-.0001) 431,431,432
  432 U1=U2
      GO TO 430
  431 IF(U2 .GE. 1.0) U2=.999998
      XJCBAR=2.0*ETA*UALPHA*1000.*(1.-U2)/U2
      U1=.9
  230 IF(U1.GT.1.0) U1=.999999
      P1=1.84*VC*((AD*AD*C/(GP*T))**.25)/C
      P2=(1.0-U1)**.25
      P3=(1.0-U1)**.75
      FU=ALOG(U1)+VC/C-(P1)*P2
      FUDOT=1.0/U1+P1/(P3*U1*U1*4.0)
      U2=U1-FU/FUDOT
      ETA=1./(1.+(D/C)**2)
      UALPHA=(1.0-U2)*1000.*C*C/(2.*ETA*T)
      IF(ABS(1.-(U1/U2))-.0001) 231,231,232
  232 U1=U2
      GO TO 230
  231 IF(U2 .GE. 1.0) U2=.999999
      XJ=2.0*ETA*UALPHA*1000.*(1.-U2)/U2
      CM=(ALOG(XJCBAR/XJ))/(ALOG(TCBAR/T))
      IF(CM  .LT.0.0) GO TO 240
      CM=0.0
      XJCBAR=1.0
  240 TCBAR=TCBAR/TC
  500 Q=1.0
      Q2=TCBAR**(-CM)
      GO TO 114
  200 Q2=1.0
      Q=0.
      CM=0.0
      TCBAR=0.0
      XJCBAR=0.0
      GP=GM
      EPS=EPST(L)
      TLTSI=0.0
      VJET=AISP*GE
      TLTSR=TLTSI/RAD
      VINF=VINF2
      SIGMA=ASIGMA
      AS=RP2/(1.-EPS)
      PS=AS*RG*(1.-EPS*EPS)
      RLTS=PS/(1.+EPS*COS(TLTSR))
      VSU=SQRT((GP/(AS*RG))*(1.+2.*EPS*COS(TLTSR)+EPS*EPS)/(1.-EPS*EPS))
      DVI=SQRT((VINF**2)+(2.*GP/(RLTS)))-VSU
      BMFST=EXP((-DVI)/VJET)
      BPROP=(1.+SIGMA)*BMFST-SIGMA
      ARRML=BPROP
      GO TO 114
   19 WRITE(6,221)
  221 FORMAT(1H ,22HPLANET DATA NOT STORED)
  114 RETURN
      END
```

REFERENCES

1. Mascy, A. C.; Dugan, D. W.; and Pitts, S. W.: Applications of Combined Electric, High-Thrust Propulsion Systems. J. Spacecraft and Rockets, vol. 5, no. 7, July 1968, pp. 785-791.

2. Mascy, A. C.: A Compilation of Current Computer Programs for Low-Thrust Trajectory Analysis. NASA TM X-1824, 1969.

3. United Aircraft Corp. Research Laboratories: Study of Trajectories and Upper Stage Propulsion Requirements for Exploration of the Solar System. NASA Contract NAS2-2928, Final Report, July 15, 1966, NASA CR-80297, 1966.

4. Ragsac, Robert V.: Handbook for the Study of Electric Propulsion Systems. Rep. G-110058-28, United Aircraft Corporation, Nov. 22, 1968.

5. Fimple, W. R.; and Edelbaum, T. N.: Applications of SNAP-50 Class Powerplants to Selected Unmanned Electric Propulsion Missions. AIAA paper 64-494, 1964.

6. Breakwell, J. V.; and Rauch, H. E.: Asymptotic Matching in Power-Limited Interplanetary Transfers. AAS Paper 66-114, 1966.

7. Melbourne, W. G.; and Sauer, C. G., Jr.: Performance Computations With Pieced Solutions of Planetocentric and Heliocentric Trajectories for Low-Thrust Missions. In JPL Space Summary No. 37-36, vol. IV, Dec. 1965, pp. 14-23.

8. Melbourne, W. G.; and Sauer, C. G., Jr.: Payload Optimization for Power-Limited Vehicles. JPL-TR-32-250, Jet Propulsion Laboratory, California Institute of Technology, April 9, 1962.

9. Masek, T. D.: Thrust System Technology for Solar Electric Propulsion. AIAA Paper 68-541, 1968.

10. Richley, E. A.: Bombardment Thruster Investigations at the Lewis Research Center. AIAA Paper 68-542, 1968.

11. Zola, Charles L.: A Method of Approximating Propellant Requirements of Low-Thrust Trajectories. NASA TN D-3400, 1966.

12. Mascy, Alfred C.: Low-Thrust Mission Simulation - Feedback to Hardware Definition. AIAA 7th Electric Propulsion Conference, Williamsburg, Virginia, March 3-5, 1969.

13. Casal, F. G.: A Method for Obtaining Performance Boundaries of Complete Space Propulsion Systems Containing Nuclear-Electric Stages. AIAA 7th Annual Electric Propulsion Conference, Williamsburg, Virginia, March 3-5, 1969.

14. Mascy, A. C.: Extraecliptic 1.0 AU Constant Power Electric Propulsion Missions. AIAA J. Spacecraft and Rockets, vol. 6, no. 12, Dec. 1969.

15. Hahn, D. W.; Johnson, F. T.; and Itzen, B. E.: Chebychev Trajectory Optimization Program. NASA Contract NAS2-5185, Final Report, July 1969, NASA CR-73359, 1969.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS A
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 15
Postal Manual) Do Not Retu

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546