NASA TECHNICAL NOTE

NASA TN D-5985

NASA TN D-5985

AN EARTH-MARS MISSION-ANALYSIS PROGRAM

by James F. Kibler
Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1971

1. Report No. NASA TN D-5985	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN EARTH-MARS MISSION-ANALYSIS PROGRAM		5. Report Date January 1971
		6. Performing Organization Code
7. Author(s) James F. Kibler		8. Performing Organization Report No. L-7213
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		10. Work Unit No. 815-10-01-20
		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 Note
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A rapid, flexible, preliminary Earth-Mars mission-analysis computer program has been developed. The program computes a conic interplanetary trajectory approximation, a noncoplanar impulsive deboost maneuver into a closed orbit about the target planet, and many mission-dependent and mission-independent parameters to allow examination of the entire flight profile. The capabilities of the program are discussed along with the requirements for computing a general planet-to-planet mission. Examples of program input and output and sample data analyses are presented for an Earth-Mars mission during the 1973 launch opportunity. A flow diagram for the main program, the input and output description, and a complete program listing are presented in the appendixes.

17. Key Words (Suggested by Author(s)) Mission analysis Trajectory programs Earth-Mars trajectories	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 73	22. Price* $3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AN EARTH-MARS MISSION-ANALYSIS PROGRAM

By James F. Kibler
Langley Research Center

SUMMARY

A rapid, flexible, preliminary Earth-Mars mission-analysis computer program has been developed. The program computes a conic interplanetary trajectory approximation, a noncoplanar impulsive deboost maneuver into a closed orbit about the target planet, and many mission-dependent and mission-independent parameters to allow examination of the entire flight profile. The capabilities of the program are discussed along with the requirements for computing a general planet-to-planet mission. Examples of program input and output and sample data analyses are presented for an Earth-Mars mission during the 1973 launch opportunity. A flow diagram for the main program, the input and output description, and a complete program listing are presented in the appendixes.

INTRODUCTION

Interplanetary flights from Earth to the planets represent a significant part of the space effort. Detailed study is required for each of these flights. A necessary part of the study is a preliminary mission analysis which consists of choosing a suitable mission profile from an infinite set of possible candidates. Many variables and trade-offs are available to the flight planner. For example, once a rocket booster is chosen, the maximum allowable spacecraft weight is specified for a given launch date and arrival date. The planner must then trade weight for required launch-arrival periods until a feasible combination is obtained. Once the spacecraft weight is determined, an allocation must be made for fuel to perform midcourse corrections, a deboost maneuver at the planet, and a deorbit maneuver to the surface of the planet. Each of these maneuvers will depend upon other mission constraints such as the desired orbit at the planet and the desired landing point on the surface. In addition, the mission planner must consider the effect of scientific requirements on the mission profile. For example, the landing point must be located in a scientifically interesting area and must have proper lighting for any onboard optical equipment. The orbit about the planet must satisfy constraints such as communication requirements with the Earth and the necessity for solar cells to be exposed to sunlight for the greater part of each orbit. The many different problems involved in preliminary mission analyses present a real task for the flight planner.

At Langley Research Center, computer programs have been developed to solve several individual parts of the mission-analysis problem. However, experience with the Viking project has shown the difficulty of data interchange between the programs and the necessity for an integrated approach to a preliminary mission analysis. The program described herein is an attempt to combine the many facets of preliminary mission design into one rapid and flexible program. In addition, capability not previously available in program form, such as a noncoplanar impulsive-burn deboost maneuver, has been included in this program.

The present version of the mission-analysis program is concerned only with Earth-Mars missions of the Viking type. However, modifications described herein would allow study of interplanetary missions to other planets. The accuracy of the program is limited by the use of Keplerian mechanics and impulsive-burn maneuvers rather than finite-burn integrating schemes. However, it is felt that for preliminary mission design, the order-of-magnitude accuracy involved in the approximations, as compared with an integrating program, is far outweighed by the several orders of magnitude gained in computational speed and program flexibility. Results from the various program elements agree with results obtained from other conic programs that were previously designed individually to study specific parts of the total mission.

Information required for operation of the program is contained in the appendixes. Appendix A includes a brief flow chart and a description of the primary subroutines. An explanation of the required input is given in appendix B. Appendix C describes the output parameters, and a complete FORTRAN listing is given in appendix D. The program was developed for use on a Control Data 6600 series computer and requires a field length of approximately 65000_8.

SYMBOLS

$\vec{B}$	a vector from center of target planet and perpendicular to approach asymptote of incoming hyperbola
C_3	twice total geocentric injection energy per unit mass, km^2/sec^2
D_{LA}	declination of launch asymptote as measured from Earth's equator, deg
f	true anomaly, deg
$f_{deorbit}$	true anomaly of deorbit, deg
G	Sun lighting angle, deg
h_a	apoapsis altitude of specified elliptical orbit, km

h_p	periapsis altitude of specified elliptical orbit, km
i	inclination of elliptical orbit, deg
J_2	oblateness coefficient for Mars
f_l	true anomaly of landing point (periapsis to landing-point angle on ellipse), deg
r_a	apoapsis radius of ellipse, km
r_{entry}	radius at entry to Martian atmosphere, km
r_p	periapsis radius of ellipse, km
$r_{\text{♂}}$	radius of Mars, km
$\vec{R}$	$\vec{R} = \vec{S} \times \vec{T}$ with $\vec{R}$ a unit vector completing the RST triad
$\vec{S}$	unit vector, parallel to approach asymptote at Mars and passing through center of planet
$\vec{T}$	unit vector perpendicular to $\vec{S}$ and parallel to ecliptic plane
V_e	velocity on ellipse, km/sec
V_h	velocity on hyperbola, km/sec
V_∞	hyperbolic excess velocity of spacecraft relative to Mars, km/sec
ΔV	deboost velocity-change requirement, km/sec
α	flight-path angle at entry into Martian atmosphere, deg
λ_l	declination of landing point, deg
λ_s	declination of subsolar point, deg
μ	gravitational constant for Mars, km^3/sec^2
ϕ_l	right ascension of landing point, deg

ϕ_s	right ascension of subsolar point, deg
Ω	right ascension of ascending node, deg
ω	argument of periapsis, deg

Subscripts:

1,2	denotes specific points on the ellipse
max	maximum
min	minimum

Symbols without arrows denote magnitudes.

MISSION-ANALYSIS CAPABILITIES

The computer program has been developed to fulfill a requirement for preliminary mission analysis and design. The program is intended to be a rapid engineering tool which may be easily modified to perform additional tasks as the need arises. In this light, the following paragraphs describe the basic assumptions and approximations, the method of calculation, and the program capabilities for each part of the mission-analysis problem treated by the program.

Heliocentric Trajectory

The Earth-Mars-Sun geometry used for calculating the heliocentric trajectory elements is shown in figure 1. Point masses and Keplerian mechanics are assumed throughout the analysis. The heliocentric orbits of Earth and Mars are represented by time-varying mean orbital elements. If the position vector to the Earth at a launch date and the position vector to Mars at an arrival date (fig. 1) are known, a number of methods can be used to generate a unique set of trajectory elements which connect these two points in the desired trip time. A true anomaly iteration method (ref. 1) is used here. Once the elements of the heliocentric transfer trajectory are known, many additional parameters of interest are computed. For example, C_3 (twice the injection energy of the spacecraft relative to the Earth) and D_{LA} (the declination of the launch asymptote relative to the Earth's equator) are calculated. These two parameters are of interest to the mission planner because they define launch-vehicle energy requirements per unit mass (C_3) and whether the injection into the interplanetary trajectory violates range-safety requirements

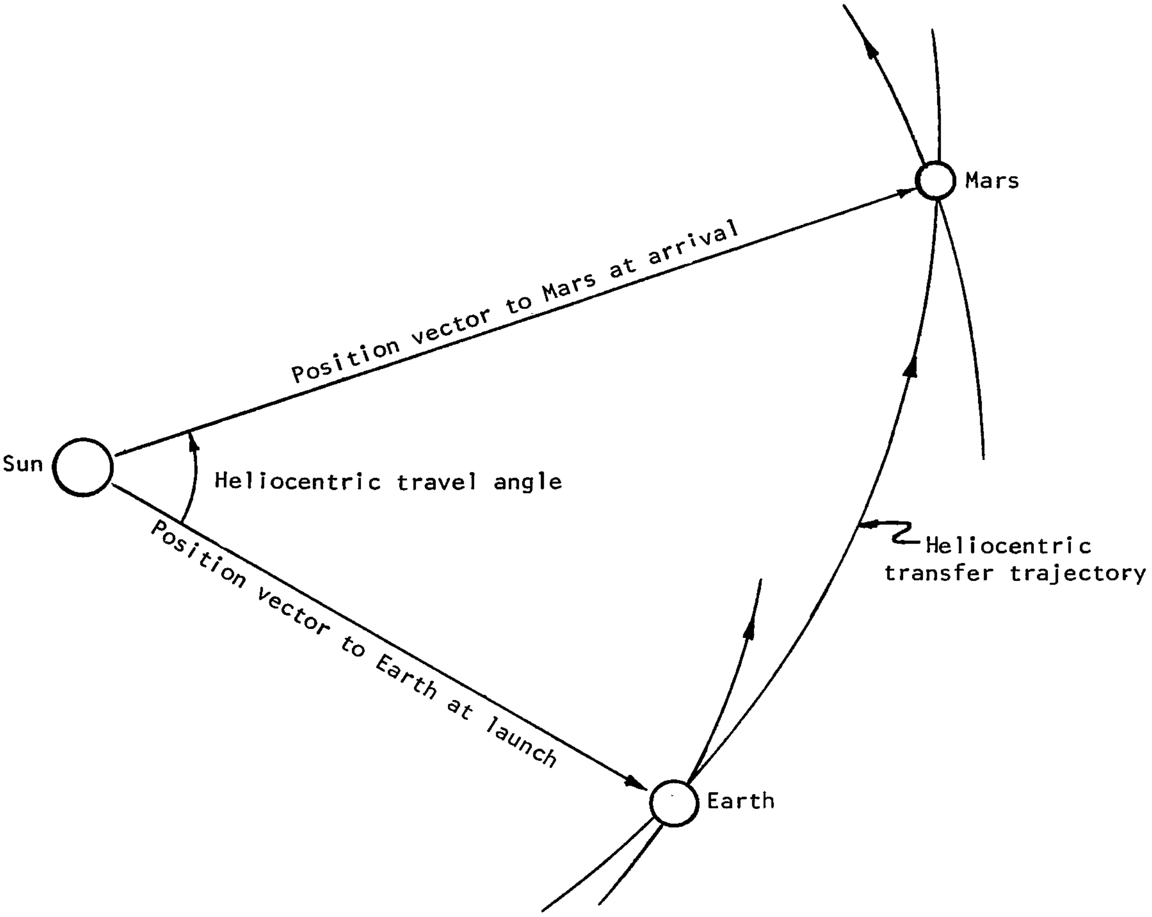


Figure 1.- Earth to Mars geometry.

(overflight restrictions on D_{LA}). Also, the hyperbolic excess velocity of the spacecraft relative to Mars V_∞ is computed here. Constraints on the maximum values of C_3, D_{LA}, and V_∞ are applied by the program. The trajectory for a given launch and arrival date is rejected if it violates any one of the constraints, and a new launch-arrival date pair is tried. Thus, the mission planner is spared the necessity of sifting through a number of impractical trajectories. The other parameters computed here are described in appendix C. This trajectory computation is very rapid and may be easily modified to compute additional quantities of interest to the mission planner.

Elliptical Orbit at Mars

The orbit trace and inertial landing-point geometry at Mars is shown in figure 2. The mission requirements of Sun lighting angle G, landing latitude λ_l, orbital inclination i, and the argument of the landing point f_l are specified. The declination and right

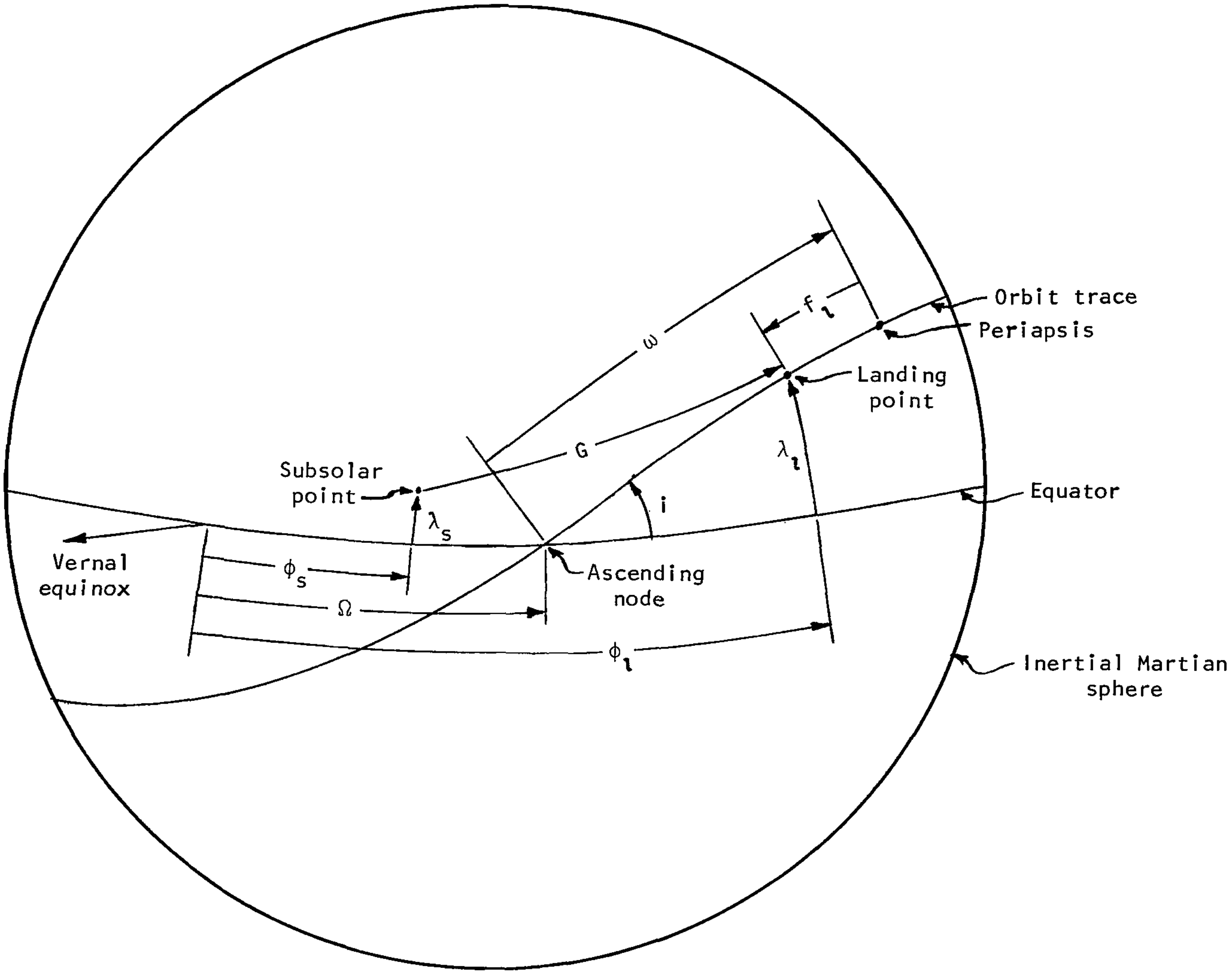


Figure 2.- Landing-point geometry. Arrows indicate positive sense.

ascension of the subsolar point, λ_s and ϕ_s, are calculated from the position of the Sun with respect to Mars. Since these quantities are known, the argument of periapsis ω and the right ascension of the ascending node Ω can be found. (See fig. 2.) The resulting equations depend on the location of the landing point with respect to the ascending or descending node of the orbit and with respect to the morning or evening terminator (that is, lighting conditions). The various combinations of landing-point conditions are chosen on option from the main program. Finally, the apoapsis and periapsis radii, r_a and r_p, are specified from experiment considerations. Thus, the orbital elements (r_a, r_p, i, Ω, and ω) of an ellipse which passes over the inertial landing point are known for the date of deorbit.

For photographic coverage, the spacecraft will be in orbit about Mars for a number of revolutions prior to landing. Because of the oblateness of Mars, Ω and ω will change as functions of time. Therefore, Ω and ω are regressed an amount dependent upon the required stay time in orbit. (See ref. 1.) Thus, the elements of the initial ellipse on the date of the deboost maneuver are determined.

The mission planner is interested in Sun and Earth occultations as seen by the spacecraft while in orbit about Mars. Therefore, such parameters as the first orbit on which occultation occurs, duration of occultation, and the time and true anomaly from periapsis of entrance to and exit from occultation are computed for both the Sun and Earth. These parameters are necessary to define quantities such as battery requirements (solar cells occulted from sunlight) and data-storage capability (direct telemetry occulted from tracking bases). The computed quantities are described in appendix C. It would be possible to compute occultation parameters for other celestial bodies (for example, Canopus) by a suitable modification to the program.

Deboost Maneuver

The deboost maneuver geometry is shown in figure 3. A minimum ΔV impulsive burn maneuver is computed. The maneuver is not constrained to be coplanar or to be a periapsis-to-periapsis transfer. The values of hyperbolic excess velocity V_∞ and a unit vector parallel to the approach asymptote and passing through the center of the planet $\vec{S}$ have been computed in the heliocentric trajectory part of the program. The quantities V_∞ and $\vec{S}$ define a family of approach hyperbolas. The orbital elements of the required ellipse at Mars have been determined in the elliptical orbit computation section. The deboost maneuver is designed to specify the family of approach hyperbolas that results in the minimum ΔV requirement for deboost.

The procedure is described with reference to figure 3. The approach hyperbola is rotated about $\vec{S}$ and its periapsis altitude is adjusted until it intersects the specified elliptical orbit at a particular true anomaly f_1. Since the radius of the hyperbola is constrained to be the same as the radius of the ellipse at that true anomaly, the orbital elements of the hyperbola are computed. The velocities on the ellipse $V_{e,1}$ and hyperbola $V_{h,1}$ at the intersection point are computed and their vector difference ΔV_1 represents an impulsive-burn transfer between the conics at their intersection. Next, another true anomaly f_2 is chosen, and the velocity difference ΔV_2 is computed at this new intersection of the hyperbola and ellipse. This process is repeated at true anomaly intervals around the ellipse. The minimum ΔV is calculated by a parabolic interpolation through the three smallest computed values of ΔV. The associated hyperbola is then defined to be the required conic. A maximum acceptable ΔV is defined by the user and any profiles which violate this constraint are rejected.

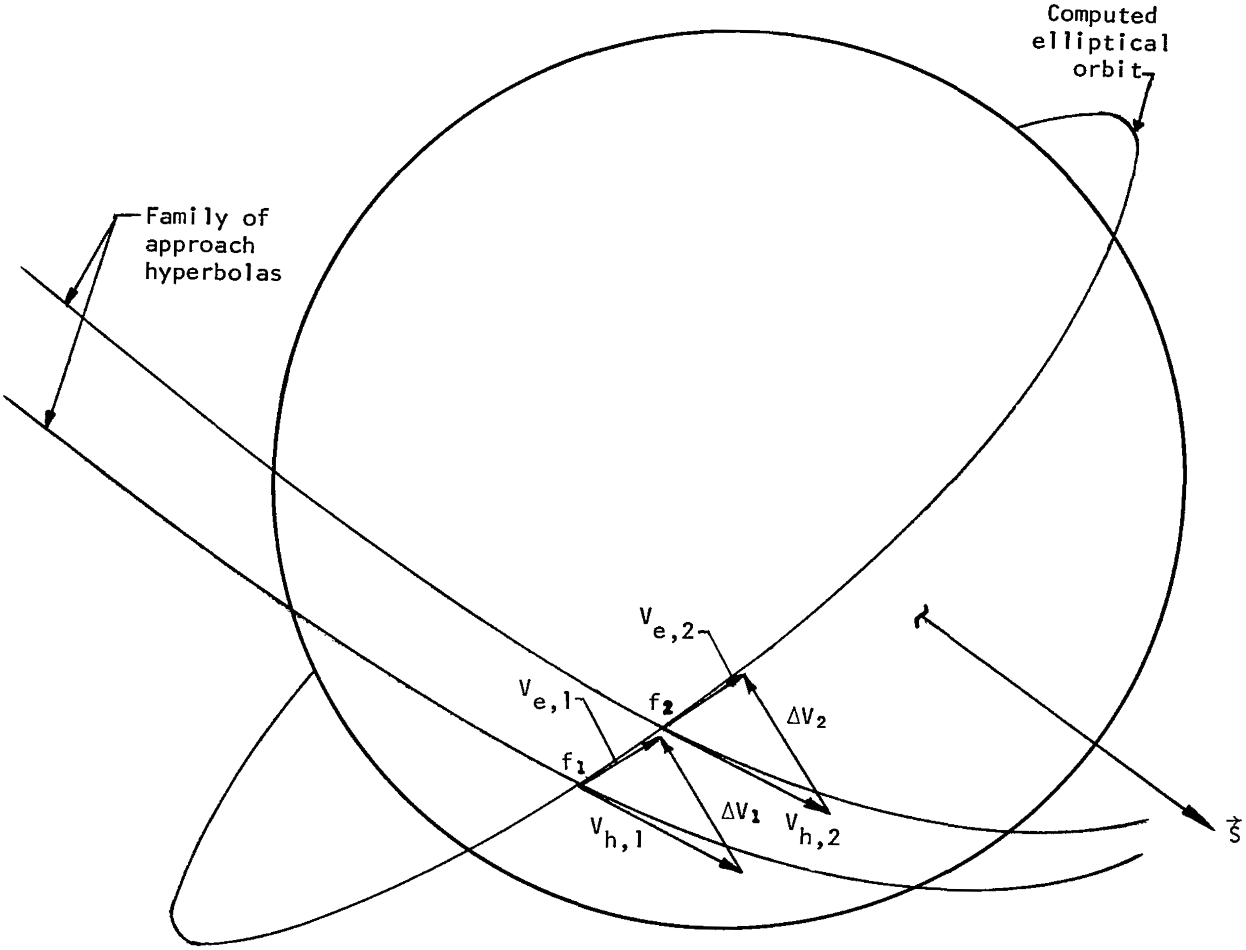


Figure 3.- Deboost velocity computation geometry.

Additional parameters of interest to the mission planner are computed in this part of the program. For example, the areocentric components of ΔV, the plane change involved in the maneuver, and the radius, time and true anomaly of the deboost maneuver are computed. Also computed are the components $B \cdot T$, $B \cdot R$ of a "miss parameter" $\vec{B}$ which is the perpendicular from the center of the planet to the approach asymptote. The components $B \cdot T$ and $B \cdot R$ lie in the plane formed by $\vec{T}$ (a unit vector perpendicular to $\vec{S}$ and parallel to the ecliptic plane) and $\vec{R}$ $(\vec{R} = \vec{S} \times \vec{T}$, with $\vec{R}$ a unit vector completing the triad). The RST areocentric coordinate system is a convenient targeting system for the mission planner. Other computed parameters associated with the deboost maneuver are described in appendix C.

Operational Modes

There are several modes of operation and program options available to the mission planner. There are three output modes which control computational flow as shown in the flow chart in appendix A. A sample input and a sample output for each of the computational modes are illustrated. An initial launch date of August 9, 1973, and an initial arrival date of March 16, 1974, are specified for each example. In each mode, the program automatically scans a grid of launch and arrival dates as determined by the user. Maximum values for the C_3, D_{LA}, V_∞, and ΔV constraints are selected. A set of landing-point parameters $(f_l, \lambda_l, G,$ i, and stay time in orbit$)$ are chosen which relate to the particular mission. Physical constants associated with the planet are specified. For each case, a set of program control integers is required. Each of the input and output parameters is described in appendixes B and C.

An example of the input and output for the minimum output mode is presented. This operational mode performs only the calculations required to define the basic mission profile. Output is restricted to a single line to facilitate scanning a wide range of possible launch and arrival date combinations. Only the profiles which satisfy constraints on C_3, D_{LA}, V_∞, and ΔV are printed out. This option requires less than 1 second of computer time per launch-arrival date pair.

A sample input and output described in appendixes B and C for the minimum output mode follows:

```
$INPT
E        =  0.73E+02,  0.8E+01,  0.9E+01,  0.0,  0.0,  0.0,
XM       =  0.74E+02,  0.3E+01,  0.16E+02,  0.0,  0.0,  0.0,
ILD      =  25,
IAD      =  25,
IJD      =  5,
JJD      =  5,
C3MAX    =  0.21E+02,
DLAMAX   =  0.4E+02,
VHPMAX   =  0.35E+01,
DELVMAX  =  0.2E+01,
PER      =  0.12E+02,
ELP      =  0.3E+02,
SPECLON  =  0.90866469013461E+02,
GEE      =  0.65E+02,
XI       =  0.4E+02,
DAYS     =  0.1E+02,
HA       =  0.3267E+05,
HP       =  0.14E+04,
RS       =  0.33934E+04,
RPMIN    =  0.47969E+04,
TADFORB  =  0.0,
FPA      = -0.16E+02,
RENTRY   =  0.363724E+04,
XJ2      =  0.197E-02,
U        =  0.428284E+05,
INC      =  1,
KEY1     =  0,
KEY3     =  1,
KEY4     =  0,
KEY5     =  0,
ISEARCH  =  0,
TDEST    =  0.0,
$END
```

```
 LAUNCH  ARRIVAL   C3    DLA   DELTAV E DBST E DBST DBST TIM SO EO  TSUNIN    DURSUN    TASIN  TASOUT    TEARIN   DUREAR    TAEIN  TAEOUT
  DATE     DATE                       ELIPSE HYPERB FROM PER


73   8  9 74   3 16 16.54 33.66 1.510 -63.7 -44.6  -1630.69  0  1      0.00      0.00     0.00    0.00      8.84     26.11    24.52   75.07
73   8  9 74   3 21 17.04 35.96 1.689 -74.4 -51.1  -2065.39  0  1      0.00      0.00     0.00    0.00      8.92     27.21    24.71   76.58
73   8  9 74   3 26 17.71 38.51 1.877 -84.1 -55.5  -2556.29  0  1      0.00      0.00     0.00    0.00      9.02     28.29    24.98   78.05

73   8 14 74   3 16 17.41 28.85 1.349 -60.8 -41.6  -1529.65  0  1      0.00      0.00     0.00    0.00      8.84     26.11    24.52   75.07
73   8 14 74   3 21 17.67 30.67 1.501 -71.5 -48.4  -1939.69  0  1      0.00      0.00     0.00    0.00      8.92     27.21    24.71   76.59
73   8 14 74   3 26 18.03 32.68 1.658 -81.3 -53.7  -2404.05  0  1      0.00      0.00     0.00    0.00      9.02     28.29    24.98   78.05
73   8 14 74   3 31 18.52 34.90 1.820 -90.2 -57.0  -2925.13  0  1      0.00      0.00     0.00    0.00      9.15     29.35    25.32   79.49
73   8 14 74   4  5 19.18 37.35 1.990 -98.0 -58.4  -3502.91  0  1      0.00      0.00     0.00    0.00      9.30     30.39    25.72   80.88

73   8 19 74   3 16 19.15 24.76 1.237 -59.4 -39.1  -1481.56  0  1      0.00      0.00     0.00    0.00      8.84     26.11    24.52   75.07
73   8 19 74   3 21 19.25 26.22 1.366 -69.8 -45.9  -1870.06  0  1      0.00      0.00     0.00    0.00      8.92     27.21    24.71   76.58
73   8 19 74   3 26 19.41 27.81 1.502 -79.5 -51.4  -2309.39  0  1      0.00      0.00     0.00    0.00      9.02     28.29    24.98   78.05
73   8 19 74   3 31 19.64 29.56 1.640 -88.2 -55.3  -2801.39  0  1      0.00      0.00     0.00    0.00      9.15     29.35    25.32   79.49
73   8 19 74   4  5 19.97 31.48 1.781 -96.1 -57.5  -3347.84  0  1      0.00      0.00     0.00    0.00      9.30     30.39    25.72   80.88
```

An example of the input and output for the extended printout mode is presented. This operational mode calculates numerous additional parameters (described in appendix C) associated with a particular launch and arrival date. In this mode, the "time-correction" option can be selected. This option allows the user to pick a particular longitude of landing (that is, "tying down" the inertial landing point to the rotating planet). The chosen longitude must rotate beneath the computed elliptic orbit with the given lighting conditions at a particular time. This constraint determines the landing time. With the time of landing known, event times of deorbit and deboost are computed. The deorbit-to-landing time increment is computed by use of an entry trajectory with no atmosphere to allow rapid computation. On option, a more accurate time increment can be put into the program. The deboost-to-deorbit time increment is computed along the specified elliptical orbit. One page of output and approximately 2 seconds of computer time per launch-arrival date pair are required in the extended output mode. Other quantities of interest to the mission planner could be computed and inserted into this output.

A sample input and output for the extended output mode (described in appendixes B and C) follow:

```
$INPT
E        =  0.73E+02,  0.8E+01,  0.9F+01,  0.0,  0.0,  0.0,
XM       =  0.74E+02,  0.3E+01,  0.16E+02,  0.0,  0.0,  0.0,
ILD      =  1,
IAD      =  1,
IJD      =  1,
JJD      =  1,
C3MAX    =  0.21E+02,
DLAMAX   =  0.4E+02,
VHPMAX   =  0.35E+01,
DELVMAX  =  0.2E+01,
PER      =  0.12E+02,
ELP      =  0.2E+02,
SPECLON  =  0.32E+03,
GEE      =  0.6F+02,
XI       =  0.3E+02,
DAYS     =  0.9E+01,
HA       =  0.3267E+05,
HP       =  0.14E+04,
RS       =  0.33934E+04,
RPMIN    =  0.47969E+04,
TADEORB  =  0.21475E+03,
FPA      = -0.16E+02,
RFNTRY   =  0.363724E+04,
XJ2      =  0.197E-02,
U        =  0.428284E+05,
INC      =  1,
KEY1     =  1,
KFY3     =  1,
KEY4     =  1,
KFY5     =  0,
ISEARCH  =  0,
TDFST    =  0.0,
$END
```

```
                  LAUNCH DATE              ARRIVAL DATE              DEBOOST TIME              DEORBIT TIME              LANDING TIME
CALENDAR        8  9 73  0  0  0         3 16 74  0  0  0         3 16 74 19 49  4         3 25 74 17  2 43         3 25 74 20 18 48
JULIAN             2441903.50               2442122.50               2442123.33               2442132.21               2442132.35

INTERPLANETARY FLIGHT PARAMETERS

    DLA =  3.36573132E+01      RAL  =  1.36567255E+01      C3  =  1.65407750E+01      TRIP TIME =  2.19000000E+02
    AREO. DEC. S-VECTOR = -1.08404062E+01      AREO. R.A. S-VECTOR =  1.21786355E+02      HYPER. EXCESS VEL. =  2.52683027E+00
    GEO. DEC. S-VECTOR = -1.74848575E+01      GEO. R.A. S-VECTOR =  4.17907729E+01      COMUNICATION DIST. =  2.32400607E+08
    ZAP =  1.03687925E+02      ETS =  1.73558382E+02      ZAE =  1.32057526E+02      ETE =  2.00221246E+02
    ZAC =  5.46630549E+01      ETC =  2.69475543E+02
    PROBE PERIHELION =  1.51318194E+08          PROBE APHELION =  2.45737365E+08          PROBE INCLINATION =  3.05159122E+00
    LAUNCH TRUE ANOM. =  8.84531180E+00      ARRIVAL TRUE ANOM. =  1.58446951E+02      HELIO. ANGLE TRAVEL =  1.49601639E+02
    B-VECTOR MAGNITUDE =  9.83210863E+03                B DOT T =  9.58532051E+03                       B DOT R = -2.18316179E+02

ELEMENTS AND DEBOOST PARAMETERS - HYPERBOLA

    A = -6.65501483E+03      E =  1.78401407E+00      I =  3.16568236E+01      CAP.OMEGA = -7.62074631E+01      OMEGA =  1.45091723E+02
    DBST TRUE ANOM. = -3.73713246E+01         DBST TIME = -7.83871231E+02
    PER. RADIUS =  5.21762525E+03      PER. DEC. =  1.74781230E+01      PER. R.A. =  7.29825332E+01
    V AT DBST =  4.54892991E+00      VX AT DBST = -3.31193319E+00      VY AT DBST =  2.68086655E+00      VZ AT DBST = -1.59272863E+00

ELEMENTS AND DEBOOST PARAMETERS - ELLIPSE

    A =  2.04284000E+04      E =  7.65356073E-01      I =  3.00000000E+01      CAP.OMEGA =  3.05385171E+02      OMEGA =  1.46776084E+02
    DBST TRUE ANOM. = -5.77435330E+01      DBST TIME = -1.42498943E+03      DORB.T.A. =  2.14750000E+02      VDOER =  2.20634813E-01
    PER. RADIUS =  4.79340000E+03      PER. DEC. =  1.58998217E+01      PER. R.A. =  9.58220685E+01      PERIOD =  1.02581424E+00
    V AT DBST =  3.48722587E+00      VX AT DBST = -2.83902172E+00      VY AT DBST =  1.89974324E+00      VZ AT DBST = -7.01195568E-01

DEBOOST MANEUVER PARAMETERS

    DELTA V =  1.27618493E+00      EXCESS DELTA V =  7.23815067E-01      RADIUS =  6.00794389E+03      PER =  1.20000000E+01
    VX =  4.72911478E-01      VY = -7.81123210E-01                   VZ =  8.91543065E-01      PLANE CHANGE =  1.11874388E+01

OCCULTATION PARAMETERS

    1ST ORBIT, SUN=  0        TIME, SUN =  0.                           DURATION, SUN =  0.
    TRUE ANOM., SUN IN =  0.                      TRUE ANOM., SUN OUT =  0.
    1ST ORBIT, EARTH =  1     TIME, EARTH =  1.25758721E+01      DURATION, EARTH =  2.44956743E+01
    TRUE ANOM., EARTH IN =  3.39979033E+01      TRUE ANOM., EARTH OUT =  7.77646115E+01

LANDING POINT PARAMETERS

    LONGITUDE =  3.20000000E+02      LATITUDE =  2.00000000E+01      SUN LIGHTING ANGLE  6.00000000E+01      DAYS =  9
```

The third output option is the "parametric-analysis" printout mode. An example of the input and output of this mode follows. This mode computes the heliocentric transfer trajectory for a specified launch and arrival date. Then the input parameters, f_l, λ_l, G, i, and stay time in orbit, are varied through specified ranges to determine their effect on the ΔV required for the deboost maneuver. One line of output and approximately 1 second of computer time are required for each combination of input parameters.

```
$INPT
E        =  0.73E+02,  0.8E+01,  0.9E+01,  0.0,  0.0,  0.0,
XM       =  0.74E+02,  0.3E+01,  0.16E+02,  0.0,  0.0,  0.0,
ILD      =  1,
IAD      =  1,
IJD      =  1,
JJD      =  1,
C3MAX    =  0.21E+02,
DLAMAX   =  0.4E+02,
VHPMAX   =  0.35E+01,
DELVMAX  =  0.2E+01,
PER      =  0.12E+02,
EIP      =  0.2E+02,
SPECLON  =  0.32E+03,
GEE      =  0.6E+02,
XI       =  0.3E+02,
DAYS     =  0.9E+01,
HA       =  0.3267E+05,
HP       =  0.14E+04,
RS       =  0.33934E+04,
RPMIN    =  0.47969E+04,
TADEORB  =  0.21475E+03,
FPA      = -0.16E+02,
RENTRY   =  0.363724E+04,
XJ2      =  0.197E-02,
```

```
U        =  0.428284E+05,
INC      =  1,
KEY1     =  2,
KEY3     =  1,
KEY4     =  1,
KEY5     =  0,
ISEARCH  =  1,
TDEST    =  0.0,
$END

 $PARAM
PER1     =  0.1E+02,
PER2     =  0.12E+02,
KPER     =  2,
ELP1     =  0.25E+02,
ELP2     =  0.3E+02,
KELP     =  5,
GEE1     =  0.6E+02,
GEE2     =  0.65E+02,
KGEE     =  5,
XI1      =  0.35E+02,
XI2      =  0.4E+02,
KXI      =  5,
DAY1     =  0.5E+01,
DAY2     =  0.1E+02,
KDAY     =  5,
$END
```

```
                     LAUNCH DATE           ARRIVAL DATE
CALENDAR          8  9 73  0  0  0        3 16 74  0  0  0
JULIAN                2441903.50             2442122.50

PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  5.0000E+00   DELTA V =  1.2627E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  1.0000E+01   DELTA V =  1.3353E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  4.0000E+01   DAYS =  5.0000E+00   DELTA V =  1.0735E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  4.0000E+01   DAYS =  1.0000E+01   DELTA V =  1.1385E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.5000E+01   XI =  3.5000E+01   DAYS =  5.0000E+00   DELTA V =  1.3832E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.5000E+01   XI =  3.5000E+01   DAYS =  1.0000E+01   DELTA V =  1.4547E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.5000E+01   XI =  4.0000E+01   DAYS =  5.0000E+00   DELTA V =  1.1932E+00
PER =  1.0000E+01   ELP =  2.5000E+01   GEE =  6.5000E+01   XI =  4.0000E+01   DAYS =  1.0000E+01   DELTA V =  1.2675E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  5.0000E+00   DELTA V =  1.6100E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  1.0000E+01   DELTA V =  1.6812E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.0000E+01   XI =  4.0000E+01   DAYS =  5.0000E+00   DELTA V =  1.2901E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.0000E+01   XI =  4.0000E+01   DAYS =  1.0000E+01   DELTA V =  1.3765E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.5000E+01   XI =  3.5000E+01   DAYS =  5.0000E+00   DELTA V =  1.7168E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.5000E+01   XI =  3.5000E+01   DAYS =  1.0000E+01   DELTA V =  1.7815E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.5000E+01   XI =  4.0000E+01   DAYS =  5.0000E+00   DELTA V =  1.4342E+00
PER =  1.0000E+01   ELP =  3.0000E+01   GEE =  6.5000E+01   XI =  4.0000E+01   DAYS =  1.0000E+01   DELTA V =  1.5179E+00
PER =  1.2000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  5.0000E+00   DELTA V =  1.2611E+00
PER =  1.2000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  3.5000E+01   DAYS =  1.0000E+01   DELTA V =  1.3326E+00
PER =  1.2000E+01   ELP =  2.5000E+01   GEE =  6.0000E+01   XI =  4.0000E+01   DAYS =  5.0000E+00   DELTA V =  1.0812E+00
```

There are several program options available within each of the output modes. The program user must select values for each of the control integers described in appendix B. Either a posigrade or retrograde hyperbola may be chosen for the approach to Mars. One of four combinations of ascending or descending node and morning or evening lighting are specified for the landing point on the surface of Mars. The user may also put in broken-plane trajectory parameters (that is, a combination of two intersecting conic trajectories with different orbital elements). If this input option is selected, no heliocentric trajectory computation is made within the program. (See the flow chart in appendix A.)

A brief description of the primary subroutines is given in appendix A. The other general purpose subroutines are given in the FORTRAN listing in appendix D. These subroutines are generally self-explanatory and may be used in many types of programs. The mathematics associated with several of the general purpose subroutines is discussed in reference 2.

Application to Other Planets

The modular construction of the mission-analysis program permits a relatively easy extension to the evaluation of a planet-to-planet mission. A planetary ephemeris subroutine must be substituted for the launch and arrival planets, and a coordinate transformation subroutine is required to rotate a vector between the mean planet equator-equinox coordinate systems. Changes in the planetary constants and minor FORTRAN modifications must also be made. The straightforward computation and flexibility of the mission-analysis program should permit the addition of any calculations required by the mission planner.

RESULTS AND DISCUSSION OF SAMPLE CASE

Many phases of preliminary mission analysis can be studied by use of the data computed by the program. Since this paper is intended as an explanation of the capabilities of the mission-analysis program, only two examples of data analysis are presented here. First, consider the problem of choosing feasible ranges of launch and arrival dates. This analysis is performed by the minimal output mode of the program. The choice of a launch and arrival date pair is constrained immediately by the available launch vehicle energy per unit mass C_3, range safety considerations D_{LA}, and the deboost velocity-change requirement ΔV. Program output is used to plot constant contours of these three quantities as functions of launch and arrival dates. (See fig. 4.) By establishing an upper value of each quantity, many combinations of launch-arrival dates which exceed one of the constraints can be eliminated immediately from further consideration. The shaded region in figure 4 indicates the launch and arrival dates which simultaneously satisfy the

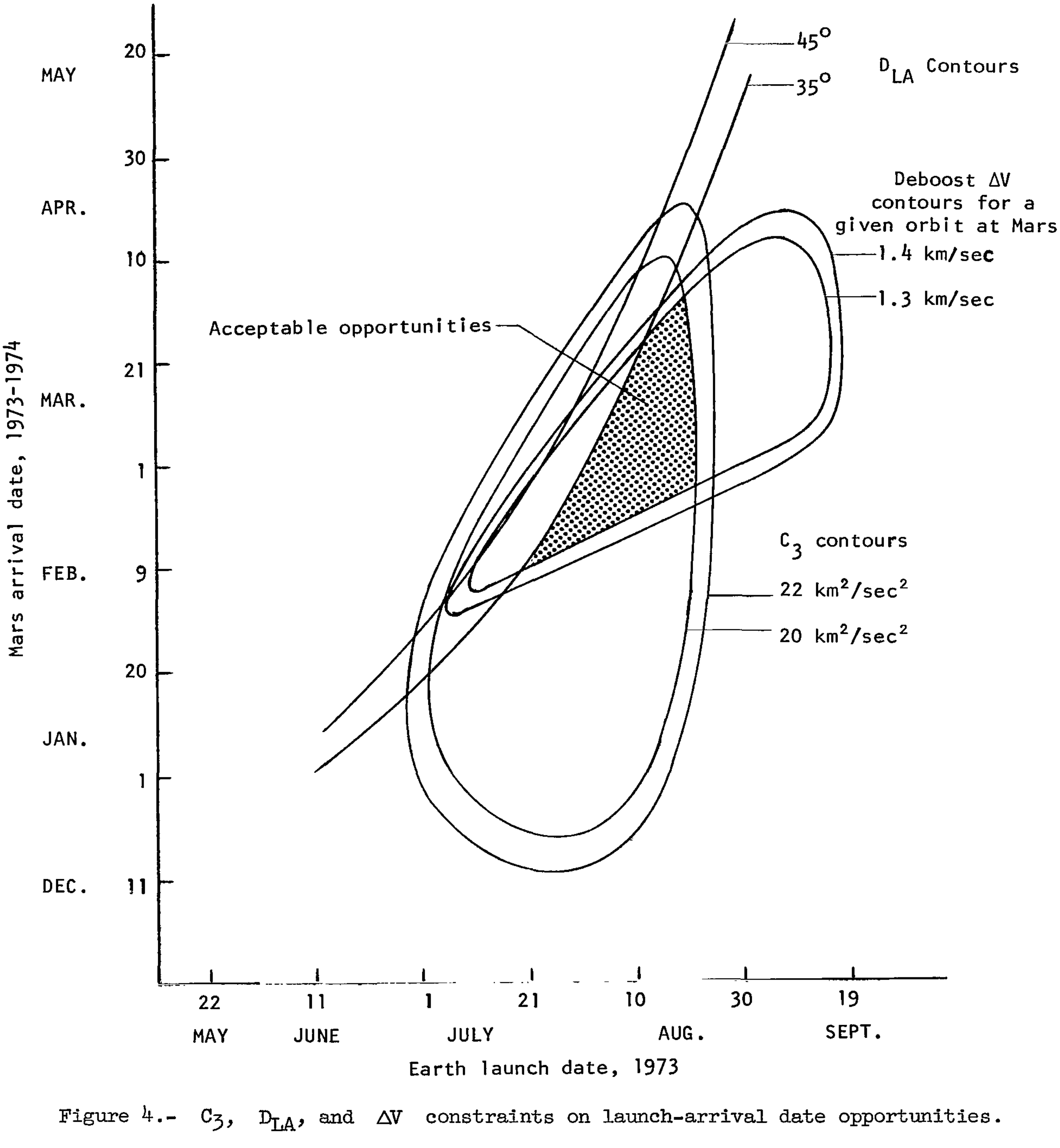


Figure 4.- C_3, D_{LA}, and ΔV constraints on launch-arrival date opportunities.

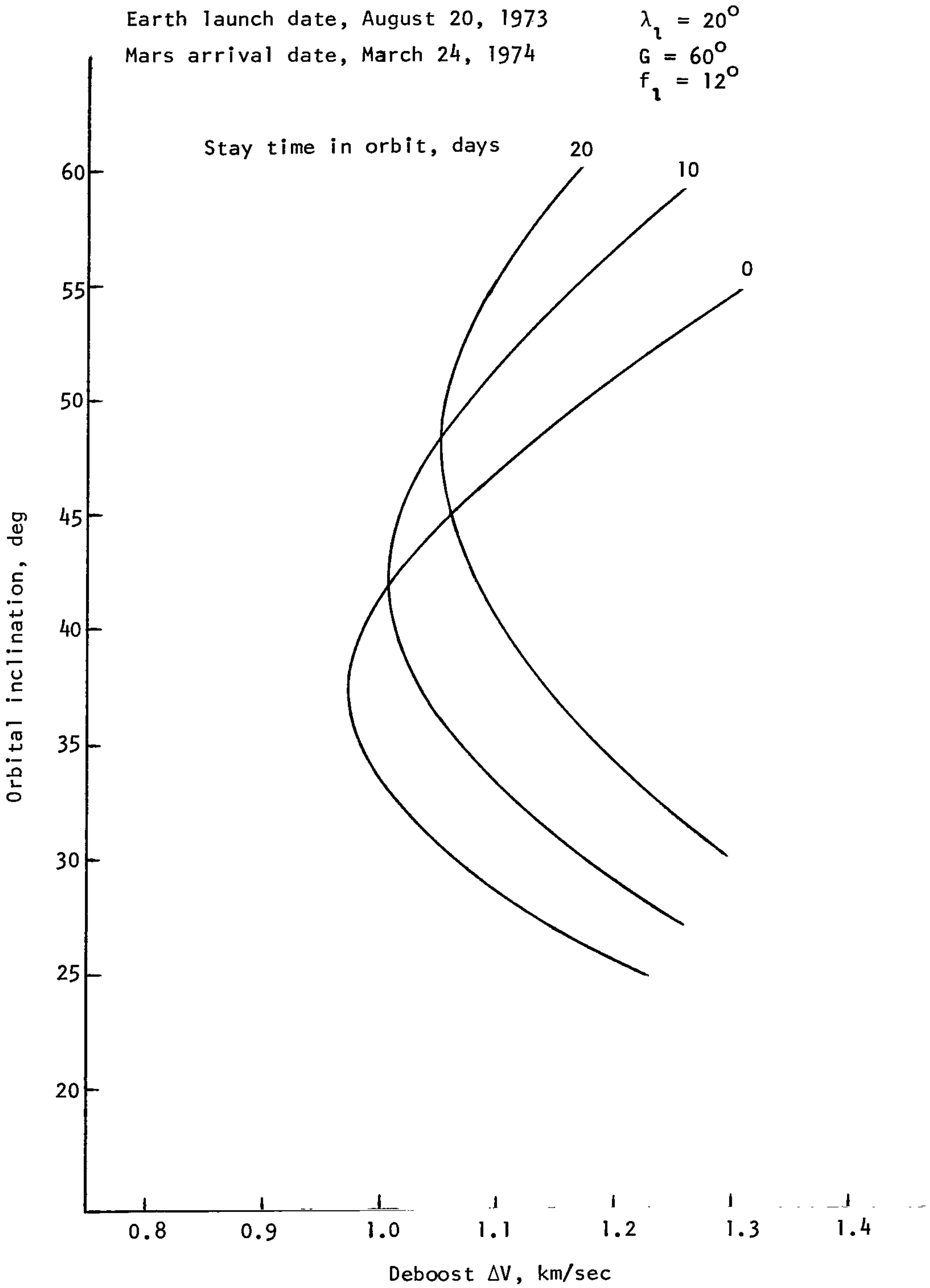
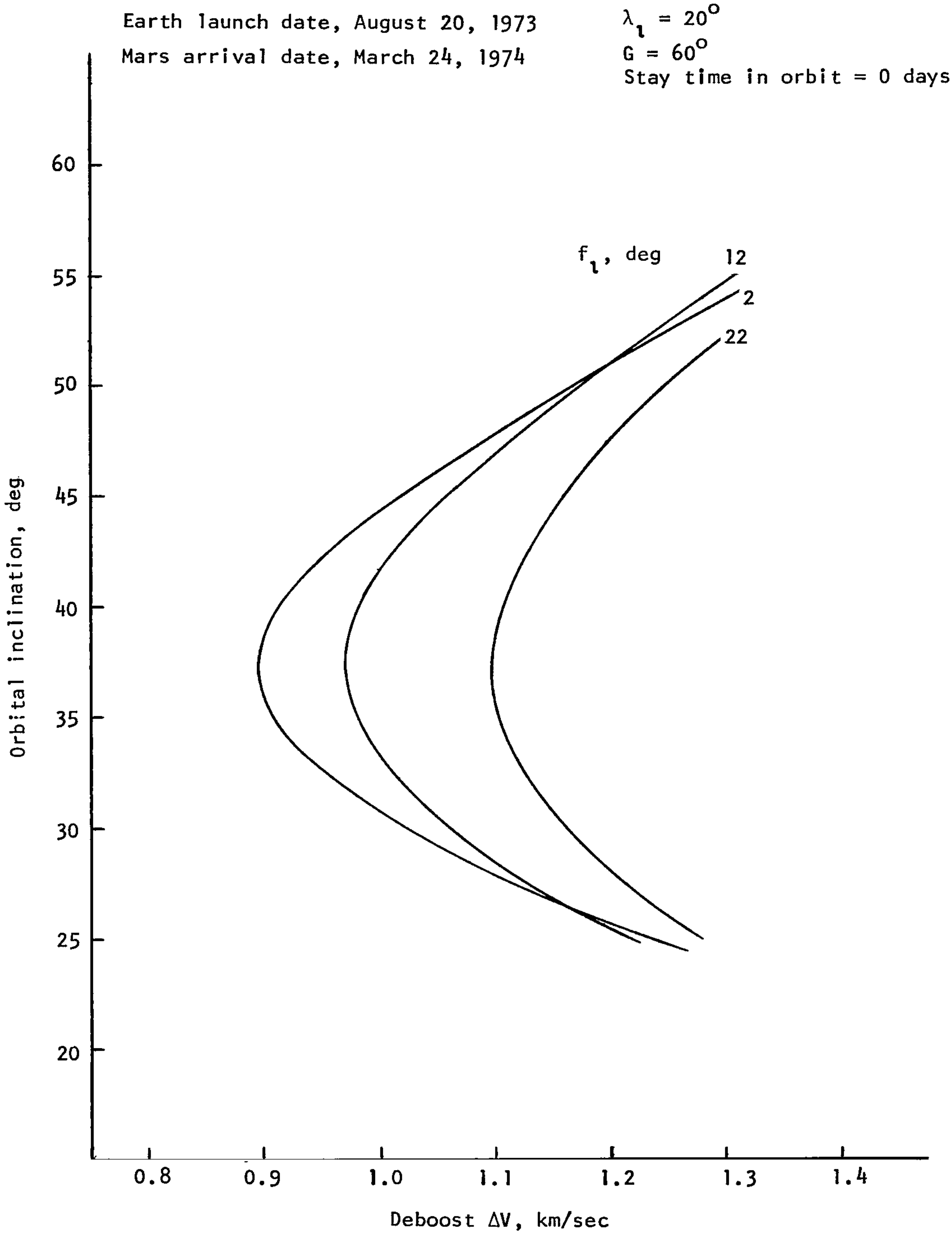


Figure 5.- Deboost ΔV variation with landing-point parameters.


Earth launch date, August 20, 1973

Mars arrival date, March 24, 1974

$\lambda_l = 20^\circ$

$G = 60^\circ$

Stay time in orbit = 0 days

Figure 5.- Continued.

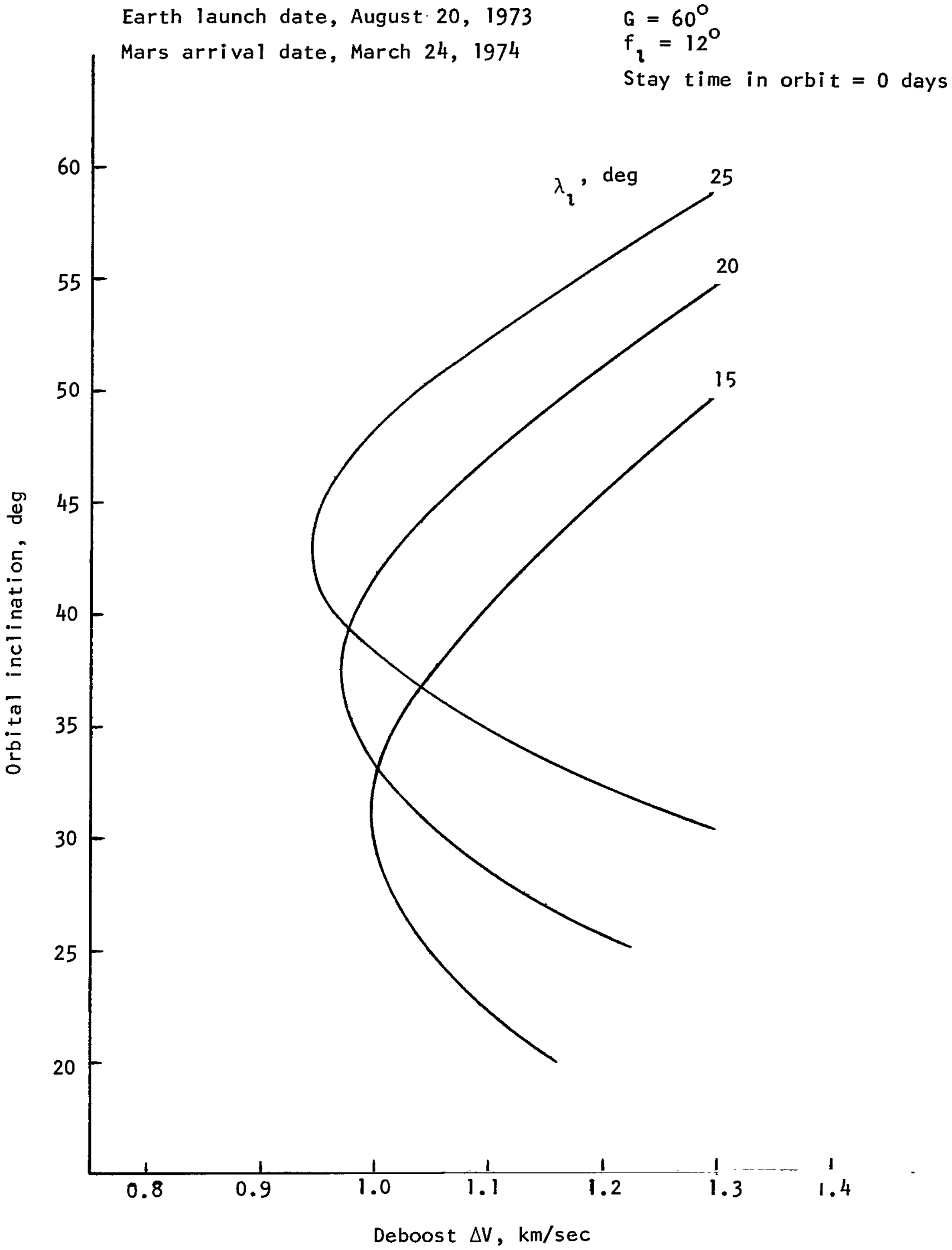


Figure 5.- Continued.

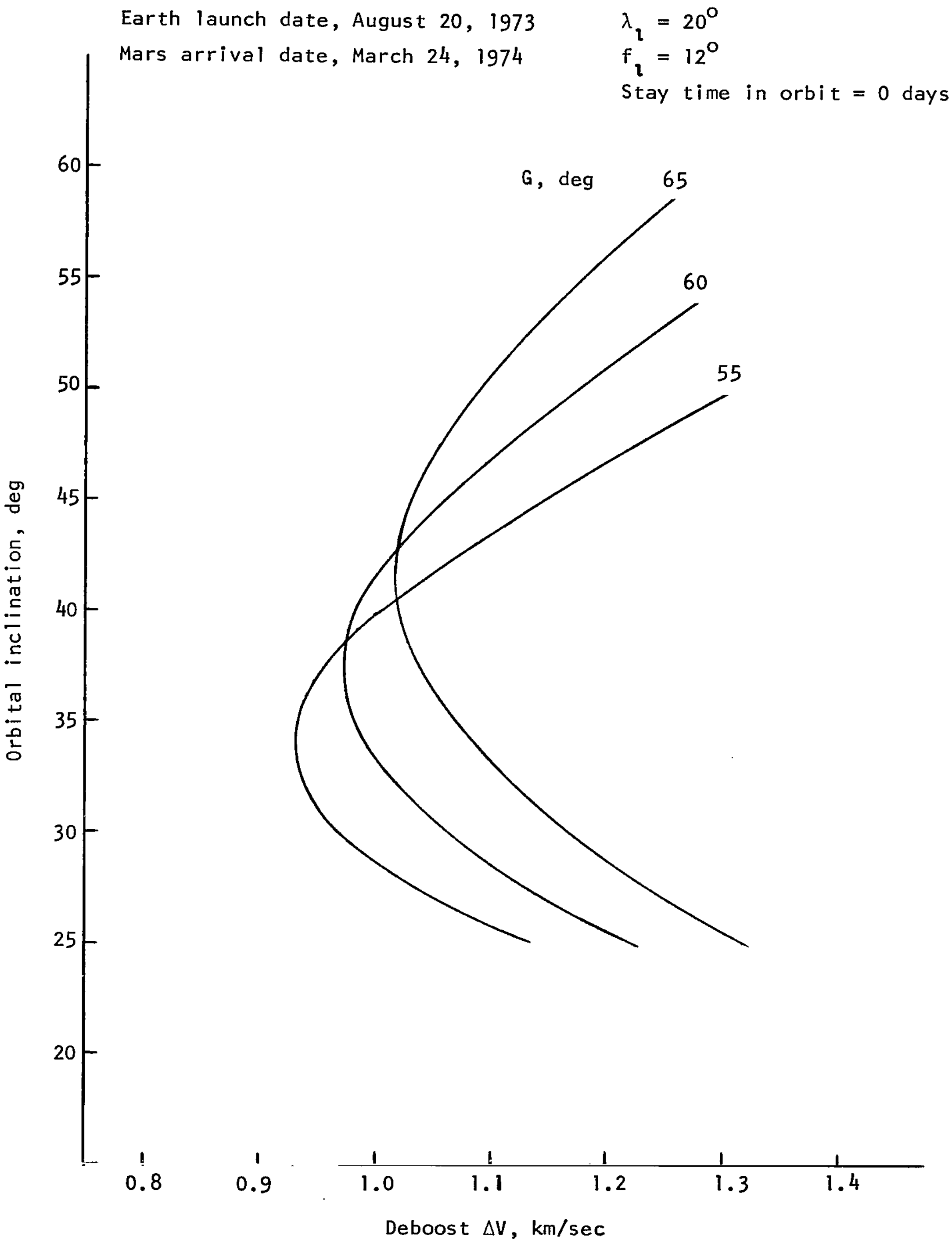


Figure 5.- Concluded.

constraints of $C_3 = 20\ \text{km}^2/\text{sec}^2$, $D_{LA} = 35^{\circ}$, and $\Delta V = 1.3$ km/sec. It should be noted that the ΔV contours are dependent upon a particular set of landing-point conditions which determine the elliptical orbit at Mars. Any variation in the elliptical orbit will shift the ΔV contours.

Next, it is logical to choose a launch-arrival opportunity from the shaded region of figure 4, and to determine the effect of the landing-point conditions on ΔV. This problem is easily handled by the parametric analysis output mode. Figure 5 shows the variation in inclination as a function of deboost ΔV for a launch date of August 20, 1973, and an arrival date of March 24, 1974. The plots show the variation in ΔV as dependent upon the Sun lighting angle, f_l, landing latitude, and stay time in orbit. Since these parameters are not truly independent, the two-dimensional plot will not tell the complete story. However, this type of analysis does indicate trends for additional study. The data presented in figures 4 and 5 can be generated in two runs of the program.

CONCLUDING REMARKS

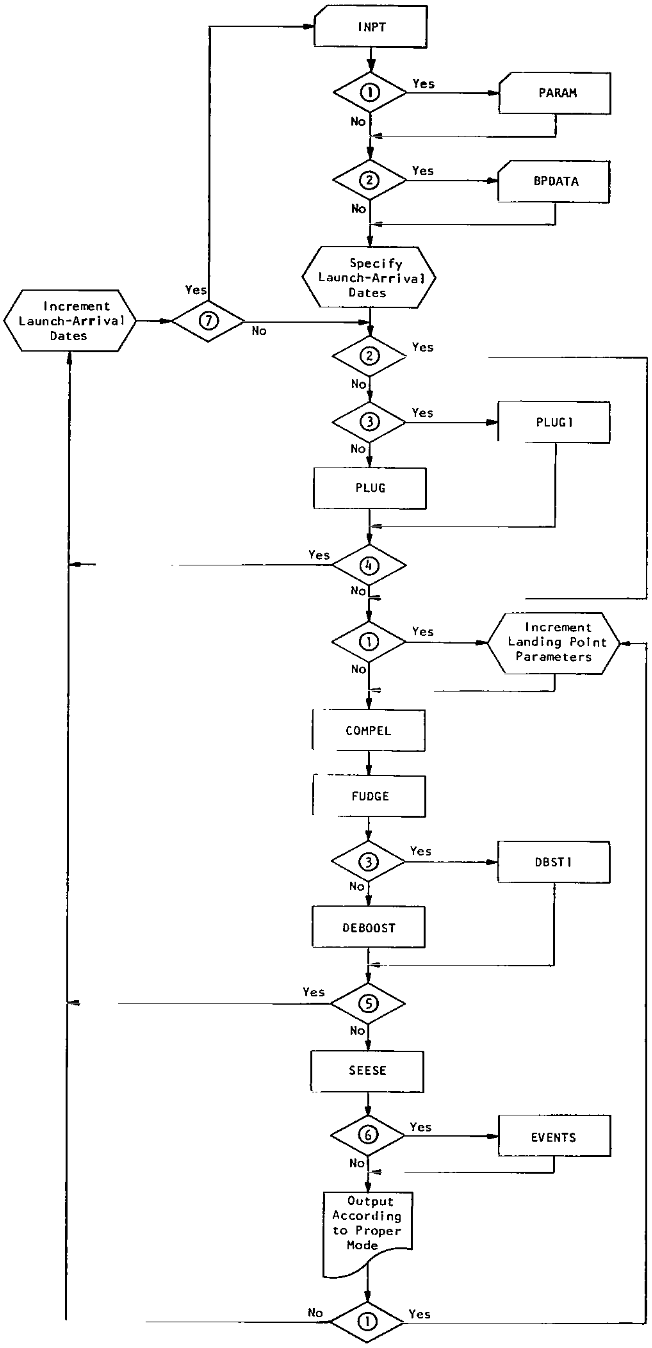
Rapid computing time, modular construction, and the combination of many phases of mission design into one package are assets of the mission-analysis program. The conic trajectory computations decrease the accuracy of the program as compared with an integrated trajectory, but this inaccuracy is felt to be of minor concern for preliminary mission planning purposes. The ease with which modifications can be made, and computational flexibility make the program a useful engineering tool.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., August 7, 1970.

APPENDIX A

MAIN PROGRAM

Flow Diagram



This flow diagram is a simple description of the computational flow of the main program. The three sets of input are described in appendix B. Each of the subroutines shown here is described briefly after the flow diagram. These subroutines, in turn, call the other subroutines given in appendix D. The numbered decisions are as follows:

① Is the program in the parametric analysis output mode?
② Is broken-plane input data required?
③ Is the program in the extended printout mode?
④ Are the C_3, D_{LA}, or V_∞ constraints exceeded?
⑤ Is the ΔV constraint exceeded?
⑥ Has the "time correction" option been selected?
⑦ Is the required range of launch-arrival dates completed?

Description of Primary Subroutines

The following is a brief description of the primary subroutines shown in the flow diagram.

PLUG and PLUG 1 – PLUG is called in the minimum output mode; it computes C_3, D_{LA}, and V_∞, and the declination and right ascension of $\vec{S}$. PLUG 1 is called in the extended output mode; it computes many additional trajectory parameters as described in appendix C. These subroutines compute the elements of a heliocentric trajectory between Earth and Mars for a given launch and arrival date. A unique conic trajectory is determined from the two radius vectors and the trip time. A true anomaly iteration method is used to establish the conic trajectory.

COMPEL – If the landing-point parameters of i, f_l, G, λ_l, stay time in orbit, and the date of landing are known, this subroutine computes the elements of an ellipse at Mars which passes over the landing point on the date of deorbit.

FUDGE – With the oblateness coefficient for Mars and the stay time in orbit known, this subroutine modifies Ω and ω as computed in COMPEL to account for oblateness effects accumulated during the specified stay time in orbit. The resultant elements determine the initial orbit at Mars.

DEBOOST AND DBST 1 – DEBOOST is called in the minimum output mode; it computes the minimum ΔV and the elements of the hyperbola associated with it, and the impact plane parameters. DBST 1 is called in the extended output mode; it computes the additional deboost parameters described in appendix C. These subroutines compute the minimum impulsive velocity change required to transfer from the approach hyperbola to the initial orbit at Mars.

SEESE – This subroutine determines whether occultations of the Sun or Earth as seen by the spacecraft occur during the specified stay time in orbit. If so, it computes the occultation parameters described in appendix C.

EVENTS – This subroutine is called in the "time correction" option. The user specifies a longitude of landing which determines the landing time. Then, event times of deorbit and deboost are computed. The user may specify the deorbit-to-landing time increment. If this time increment is not specified, an impulsive deorbit maneuver is performed and the deorbit quantities described in appendix C are computed.

APPENDIX B

DESCRIPTION OF PROGRAM INPUT

The NAMELIST feature of the Control Data 6600 computer is used to facilitate data input. Three namelists are used. INPT is the standard set of data and control variables and is always required. PARAM is a set of quantities used to vary the landing point data parametrically and is put in only in the parametric-analysis output mode. BPDATA is a set of broken-plane trajectory parameters. If this namelist is put in, no trajectory calculation is made within the program.

The following quantities are put in by the INPT namelist.

Program symbol	Program dimension	Mathematical symbol	Units	Definition
E	6		yr, mo, day, hr, min, sec	Calendar launch date (for example, 73, 10, 23, 0, 0, 0)
XM	6		yr, mo, day, hr, min, sec	Calendar arrival date
ILD	1		days	Scanning period size for launch
IAD	1		days	Scanning period size for arrival
IJD	1		days	Launch-date increment
JJD	1		days	Arrival-date increment
C3MAX	1	$C_{3,\max}$	km^2/sec^2	Constraint on vis-viva injection energy C_3
	1			
DLAMAX	1	$D_{LA,\max}$	deg	Constraint on declination of the launch asymptote D_{LA}
VHPMAX	1	$V_{\infty,\max}$	km/sec	Constraint on hyperbolic excess velocity V_∞
DELVMAX	1	$\Delta V_{\max}$	km/sec	Constraint on deboost ΔV
PER	1	f_l	deg	Periapsis to landing point angle
ELP	1	λ_l	deg	Landing-point latitude
SPECLON	1	ϕ_p	deg	Specified landing-point longitude (required only in time-correction mode)
GEE	1	G	deg	Sun lighting angle

APPENDIX B – Continued

Program symbol	Program dimension	Mathematical symbol	Units	Definition
XI	1	i	deg	Orbit inclination at Mars
DAYS	1		days	Stay time in orbit
HA	1	h_a	km	Height of apoapsis above Mars surface
HP	1	h_p	km	Height of periapsis above Mars surface
RS	1	$r_{♂}$	km	Radius of Mars
RPMIN	1	$r_{p,min}$	km	Minimum periapsis radius of approach hyperbola
TADEORB	1	$f_{deorbit}$	deg	True anomaly of deorbit (required only in time-correction mode)
FPA	1	α	deg	Flight-path angle at entry
RENTRY	1	r_{entry}	km	Radius at entry (FPA and RENTRY are required only in time-correction mode and when no estimated time from deorbit to landing has been specified)
XJ2	1	J_2	none	Oblateness coefficient for Mars
U	1	μ	km^3/sec^2	Gravitational constant for Mars
INC	1		none	INC is 1 for posigrade hyperbola and 2 for retrograde hyperbola
KEY1	1		none	Output control integer KEY1 is 0 for minimum-output mode, 1 for extended-output mode, and 2 for parametric-analysis output mode
KEY3	1		none	Landing-point control integer KEY3 is 1 for descending node, p.m. lighting; 2 for ascending node, p.m. lighting; 3 for descending node, a.m. lighting; 4 for ascending node, a.m. lighting
KEY4	1		none	Time-correction mode control integer KEY4 is 1 for time correction, and 0 for standard run

Program symbol	Program dimension	Mathematical symbol	Units	Definition
KEY5	1		none	Broken-plane input control integer KEY5 is 1 for broken-plane input (BPDATA namelist must be added) and 0 for standard input
ISEARCH	1		none	Parametric-analysis control integer ISEARCH is 1 for parametric analysis (PARAM namelist must be added) and 0 for standard run
TDEST	1		days	Estimated time from deorbit to landing TDEST is 0 yields computed time from deorbit to landing

The following quantities are input by the PARAM namelist when required.

Program symbol	Units	Definition
PER1	deg	Initial value of f_l
PER2	deg	Final value of f_l
KPER	deg	Incremental value of f_l
ELP1,ELP2,KELP	deg	Initial, final, and incremental values of λ_l
GEE1,GEE2,KGEE	deg	Initial, final, and incremental values of G
XI1,XI2,KXI	deg	Initial, final, and incremental values of i
DAY1,DAY2,KDAY	days	Initial, final, and incremental values of stay time in orbit

The following quantities are input by the BPDATA namelist when required. All are associated with a particular broken-plane trajectory.

Program symbol	Units	Definition
C3	km^2/sec^2	C_3
DLA	deg	D_{LA}
VHP	km/sec	V_∞
DPA	deg	Declination of the approach asymptote
RAP	deg	Right ascension of the approach asymptote
XM	yr, mo, day, hr, min, sec	Arrival date for broken-plane trajectory
E	yr, mo, day, hr, min, sec	Launch date for broken-plane trajectory

APPENDIX C

DESCRIPTION OF PROGRAM OUTPUT

Three output options are available in the program. The first output option is a minimum print mode. Several important parameters associated with a particular launch-arrival date pair are printed on a single line. This mode facilitates scanning a wide range of launch-arrival date combinations to select suitable mission profiles. Only profiles which satisfy the C_3, D_{LA}, V_∞, and ΔV deboost constraints are printed out. The program automatically scans a grid of launch and arrival dates as determined by the first six quantities in the INPT namelist. The following quantities are output in the minimum print mode.

Output	Units	Definition
LAUNCH DATE	yr, mo, day	Launch date at Earth
ARRIVAL DATE	yr, mo, day	Arrival date at Mars
C3	km^2/sec^2	Vis-viva injection energy for Earth-Mars trajectory
DLA	deg	Declination of launch asymptote
DELTAV	km/sec	Deboost velocity change requirement
F DBST ELLIPSE	deg	True anomaly of deboost on the elliptical orbit at Mars
F DBST HYPERB	deg	True anomaly of deboost on the approach hyperbola
DBST TIM FROM PER	sec	Time of deboost from the periapsis of the ellipse
SO	none	The first orbit on which occultations of the Sun take place; SO = 0 indicates no occultations of the Sun during the specified stay time in orbit
EO	none	The first orbit of Earth occultations; EO = 0 indicates no occultations during the specified stay time in orbit
TSUNIN	min	Time from elliptical periapsis of entrance to Sun occultation
DURSUN	min	Duration of Sun occultation
TASIN	deg	True anomaly of entrance to Sun occultation
TASOUT	deg	True anomaly of exit from Sun occultation
TEARIN,DUREAR, TAEIN,TAEOUT	min deg	Parameters associated with Earth occultations

The second output option is an extended printout mode. This option performs the same tasks as the minimum print mode, but with many additional parameters computed. The program must be in this mode in order to select the time-correction option. This operational mode is useful for examining a candidate mission profile in detail. The following quantities are output in the extended printout mode.

Output	Units	Definition
LAUNCH DATE		Calendar (mo, day, yr, hr, min, sec) and Julian (days) launch date from Earth
ARRIVAL DATE		Calendar and Julian arrival date at Mars
DEBOOST TIME		Calendar and Julian deboost date; output only in time-correction option
DEORBIT TIME		Calendar and Julian deorbit date; output only in time-correction option
LANDING TIME		Calendar and Julian landing date; output only in time-correction option
DLA	deg	Geocentric declination of the launch asymptote
RAL	deg	Geocentric right ascension of the launch asymptote
C3	km^2/sec^2	Vis-viva injection energy
TRIP TIME	days	Trip time
AREO.DEC.S-VECTOR	deg	Areocentric declination of $\vec{S}$
AREO.R.A.S-VECTOR	deg	Areocentric right ascension of $\vec{S}$
HYPER.EXCESS VEL.	km/sec	Hyperbolic excess velocity
GEO.DEC.S-VECTOR	deg	Geocentric declination of $\vec{S}$
GEO.R.A.S-VECTOR	deg	Geocentric right ascension of $\vec{S}$
COMMUNICATION DIST.	km	Line-of-sight distance from Mars center to Earth center at arrival date
ZAP	deg	Angle between $\vec{S}$ and Mars-to-Sun vector
ETS	deg	Angle measured clockwise from $\vec{T}$-axis to negative of projection of Mars-to-Sun vector on the $\vec{R}\vec{T}$ plane (measured in areocentric, equatorial, arrival date coordinates)
ZAE	deg	Same as ZAP with Mars-to-Earth vector

Output	Units	Definition
ETE	deg	Same as ETS with Mars-to-Earth vector
ZAC	deg	Same as ZAP with Mars-to-Canopus vector
ETC	deg	Same as ETS with Mars-to-Canopus vector
PROBE PERIHELION	km	Periapsis of heliocentric transfer trajectory
PROBE APHELION	km	Apoapsis of heliocentric transfer trajectory
PROBE INCLINATION	deg	Inclination to the ecliptic of heliocentric transfer trajectory
LAUNCH TRUE ANOM.	deg	True anomaly of launch point on transfer trajectory
ARRIVAL TRUE ANOM.	deg	True anomaly of arrival point on transfer trajectory
HELIO. ANGLE TRAVEL	deg	Heliocentric angle between launch and arrival points
B-VECTOR MAGNITUDE	km	Magnitude of $\vec{B}$ ("miss distance" from center of planet perpendicular to the approach asymptote)
B DOT T	km	Component of $\vec{B}$ along the $\vec{T}$-axis; areocentric ecliptic of date coordinate system
B DOT R	km	Component of $\vec{B}$ along the $\vec{R}$-axis; areocentric ecliptic of date coordinate system

The following parameters are output for both the approach hyperbola and the elliptical orbit about Mars.

Output	Units	Definition
A	km	Semimajor axis of conic
E	none	Eccentricity of conic
I	deg	Inclination to Martian equator of conic
CAP.OMEGA	deg	Right ascension of the ascending node of conic
OMEGA	deg	Argument of periapsis of conic
DBST TRUE ANOM.	deg	True anomaly of deboost point on conic
DBST TIME	sec	Deboost time from periapsis on conic

APPENDIX C – Continued

Output	Units	Definition
PER. RADIUS	km	Periapsis radius of conic
PER. DEC.	deg	Areocentric declination of periapsis of conic
PER. R.A.	deg	Areocentric right ascension of periapsis of conic
V AT DBST	km/sec	Magnitude of velocity on conic at deboost
VX,VY,VZ AT DBST	km/sec	Areocentric components of velocity on conic at deboost

In addition to these parameters, the following quantities are output for the ellipse.

Output	Units	Definition
DORB. T.A.	deg	True anomaly of deorbit point on ellipse; output only in time-correction mode
VDORB	km/sec	Velocity change required for deorbit; output only in time-correction mode
PERIOD	days	Period of the ellipse

The following parameters are output under the headings of "DEBOOST MANEUVER," "OCCULTATION," and "LANDING POINT."

Output	Units	Definition
DELTA V	km/sec	Magnitude of velocity increment required for deboost
EXCESS DELTA V	km/sec	$\Delta V_{max} - \Delta V_{deboost}$
RADIUS	km	Radius on conics at deboost point
PER	deg	True anomaly of landing point beneath the ellipse – (positive if landing before periapsis, negative if landing after periapsis)
VX,VY,VZ	km/sec	Areocentric components of DELTAV
PLANE CHANGE	deg	Angle between the planes of the approach hyperbola and the elliptical orbit
1ST ORBIT, SUN	none	The first orbit on which occultations of the Sun occur

Output	Units	Definition
TIME, SUN	min	Time of entrance to Sun occultation from periapsis
DURATION, SUN	min	Duration of Sun occultation
TRUE ANOM., SUN IN	deg	True anomaly of entrance to Sun occultation
TRUE ANOM., SUN OUT	deg	True anomaly of exit from Sun occultation
1ST ORBIT, EARTH	none	The first orbit on which occultations of the Earth occur
TIME, EARTH	min	Parameters associated with Earth occultations
DURATION, EARTH	min	
TRUE ANOM., EARTH IN	deg	
TRUE ANOM., EARTH OUT	deg	
LONGITUDE	deg	Areocentric right ascension of the landing point (in time-correction mode, the specified longitude is substituted)
LATITUDE	deg	Specified latitude of the landing point
SUN LIGHTING ANGLE	deg	Specified lighting angle at the landing point
DAYS	days	Stay time in orbit prior to deorbit

The third output option is a parametric-analysis printout mode. This mode lists the launch and arrival date for a particular trajectory. Then, on a single line, PER, ELP, GEE, XI, DAYS, and DELTA V are listed. Each landing-point parameter can be varied in turn to determine its effect of ΔV required for the deboost maneuver.

APPENDIX D

PROGRAM LISTING

The following is a FORTRAN listing of the mission-analysis program and associated subroutines:

```
      PROGRAM MISHAP  (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION E(6),XM(6), E1(6),XM1(6)
      NAMELIST/INPT/E,XM,ILD,IAD,IJD,JJD,C3MAX,DLAMAX,VHPMAX,DELVMAX,
     1PER,ELP,SPECLON,GEE,XI,DAYS,HA,HP,RS,RPMIN,TADEORB,FPA,RENTRY,XJ2,
     2U,INC,KEY1,KEY3,KEY4,KEY5,ISEARCH,TDEST
     4        /PARAM/PER1,PER2,KPER,ELP1,ELP2,KELP,GEE1,GEE2,KGEE,XI1,X
     5I2,KXI,DAY1,DAY2,KDAY
     2/BPDATA/C3,DLA,VHP,DPA,RAP,XM,E
      DIMENSION DBST(6),DEOR(6),XLAND(6)
    1 READ(5,INPT)
      WRITE(6,INPT)
      IF(ISEARCH.NE.1)GO TO 11
      READ(5,PARAM)
      WRITE(6,PARAM)
   11 CONTINUE
C
C
C     INPUT FOR MAIN
C
C     E(1-6)=FIRST CALENDAR DATE OF LAUNCH PERIOD.
C     E(1)=CALENDAR YEAR(2 DIGITS), E(2)=CAL. MONTH, E(3)=CAL DAY, E(4)=HOURS
C     E(5)=MINUTES, E(6)=SECONDS.
C     XM(1-6)=FIRST CALENDAR DATE OF ARRIVAL PERIOD.
C     XM(1,2,3,4,5,6)-SAME AS FOR LAUNCH.
C     ILD=LAUNCH PERIOD SIZE, IAD=ARRIVAL PERIOD SIZE.
C     IJD=LAUNCH DATE INCREMENT, JJD=ARRIVAL DATE INCREMENT,
C     PER=IMPACT ANGLE, ELP=DECLINATION OF IMPACT, SPECLON=SPECIFIED
C     LONGITUDE OF IMPACT, GEE=SUN LIGHTING ANGLE, XI=INCLINATION OF
C     ELLIPTICAL ORBIT, HA=HEIGHT OF APOAPSIS, HP=HEIGHT OF PERIAPSIS,
C     DAYS=STAY TIME PRIOR TO DEORBIT.
C     RS=RADIUS OF MARS, RPMIN=MINIMUM RADIUS OF PERIAPSIS OF HYPERBOLA,
C     TADEORB=TRUE ANOMALY OF DEORBIT, FPA=FLIGHT PATH ANGLE AT ENTRY,
C     RENTRY=RADIUS AT ENTRY,
C     XJ2=OBLATENESS COEFFICIENT, U=MU FOR MARS
C     KEY1=0 FOR MINIMAL OUTPUT, =1 FOR EXTENDED OUTPUT, =2 FOR PARAMETRIC
C     ANALYSIS OUTPUT.
C     KEY3= 1,DESCENDING NODE(PM) - 2,ASCENDING NODE(PM) -
C           3,DESCENDING NODE(AM) - 4,ASCENDING NODE(AM) -
C     KEY4=1 FOR TIME CORRECTION LOOP, =0 FOR NO CORRECTION,
C     KEY5=1 FOR BROKEN PLANE INPUT, =0 FOR STANDARD INPUT.
C     ISEARCH=1 FOR PARAMETER RUN ON PARTICULAR LAUNCH-ARRIVAL DATE PAIR,0=NO
C     PER1,ELP1,GEE1,XI1,DAY1 = FIRST VALUES TO BE INPUT WHEN SEARCHING
C     ON A PARTICULAR LAUNCH-ARRIVAL DATE PAIR.
C     PER2,ELP2,GEE2,XI2,DAY2 = LAST VALUES.
C     KPER,KELP,KGEE,KXI,KDAY = INCREMENTAL VALUES(INTEGER ONLY)
C     TDEST=ESTIMATED TIME FROM DEORBIT TO LANDING, =0 YIELDS COMPUTED TIME.
C
C     ALL LENGTHS IN KILOMETERS, ALL ANGLES IN DEGREES.
```

```
C
      WRITE(6,650)
      IF(KEY1.NE.0)GO TO 2
      WRITE(6,875)
      WRITE(6,876)
    2 CONTINUE
      IF(KEY5.NE.1)GO TO 21
   24 READ(5,BPDATA)
   21 CONTINUE
C
      CALL CALJUL(EWJD,EFJD,WND,FD,E)
      CALL CALJUL(AWJD,AFJD,WND,FD,XM)
      DO 700 I=1,ILD,IJD
      WRITE(6,500)
      DO 700 J=1,IAD,JJD
      DATEE=FLOAT(I-1)+EWJD+EFJD
      DATEM=FLOAT(J-1)+AWJD+AFJD
C
      EWDI=IFIX(DATEE)
      EFDI=DATEE-EWDI +.0000001
      AWDI=IFIX(DATEM)
      AFDI=DATEM-AWDI    +.0000001
      CALL JULCAL(E1,EWDI,EFDI,0)
      CALL JULCAL(XM1,AWDI,AFDI,0)
C
      IF(KEY5.EQ.1)GO TO 22
      IF(KEY1.NE.0)GO TO 4
      CALL PLUG(DATEE,DATEM,C3,DLA,VHP,DPA,RAP,KK)
      GO TO 5
    4 CALL PLUG1(DATEE,DATEM,C3,DLA,VHP,DPA,RAP,TT,RAL,DCOM,PROBPER,PROB
     1AP,PROBINC,TAL,TAA,HELANG,GDA,GRA,ZAP,ETS,ZAE,ETE,ZAC,ETC,KK)
    5 CONTINUE
C
      IF(KK.EQ.0)GO TO 700
C
      IF(C3.GT.C3MAX)GO TO 700
      IF(DLA.GT.DLAMAX)GO TO 700
      IF(VHP.GT.VHPMAX)GO TO 700
      GO TO 23
   22 CONTINUE
   23 CONTINUE
C
C
      IF(ISEARCH.NE.0)GO TO 12
      JPER=1.
      JELP=1.
      JGEE=1.
      JXI =1.
      JDAY=1.
   12 CONTINUE
      IF(ISEARCH.NE.1)GO TO 13
      JPER=ABS(PER1-PER2)+1
      JELP=ABS(ELP1-ELP2)+1
```

```
      JGEE=ABS(GEE1-GEE2)+1
      JXI =ABS( XI1- XI2)+1
      JDAY=ABS(DAY1-DAY2)+1
   13 CONTINUE
      IF(KEY1.NE.2)GO TO 15
      WRITE(6,900)E1(2),E1(3),E1(1),E1(4),E1(5),E1(6),XM1(2),XM1(3),XM1(
     11),XM1(4),XM1(5),XM1(6),DATEE,DATEM
   15 CONTINUE
      DO 700 I1=1,JPER,KPER
      DO 700 I2=1,JELP,KELP
      DO 700 I3=1,JGEE,KGEE
      DO 700 I4=1,JXI ,KXI
      DO 700 I5=1,JDAY,KDAY
C
      IF(ISEARCH.NE.1)GO TO 14
      PER=PER1+I1-1.
      ELP=ELP1+I2-1.
      GEE=GEE1+I3-1.
      XI = XI1+I4-1.
      DAYS=DAY1+I5-1.
   14 CONTINUE
C
      KEY2=1
      CALL COMPEL(PER,ELP,GEE,XI,CAPW,XITW,DATEM,HA,HP,AE,EE,DAYS,RS,KEY
     12,KEY3,ELONP)
      EPRAD=HP + RS
      IF(KEY2.EQ.0)GO TO 700
C
      CALL FUDGE(XJ2,XI,AE,EE,DELCOM,DELSOM,PD,XITW,CAPW,OE,WE,U,DAYS,RS
     1)
C
      IF(KEY1.NE.0)GO TO 6
      M=10
      CALL DEBOOST(AE,EE,XI,WE,OE,FE,U,DPA,RAP,VHP,RPMIN,INC,AZ,EH,ZI, W
     1Z,OZ,FH,BT,BR,DELTV,TPERE,M)
      GO TO 7
    6 M=1
      CALL DBST1(AE,EE,XI,WE,OE,FE,U,DPA ,RAP ,VHP ,RPMIN,INC,AZ,EH,ZI,W
     1Z,OZ,FH,BT,BR,DELTV,TPERE,TPERH,B,HPRAD,DBRAD,VXH,VYH,VZH,VXE,VYE,
     2VZE,VXD,VYD,VZD,DECHP,RAHP,DECEP,RAEP,VDBH,VDBE,PLANE,M)
    7 CONTINUE
C
C  ******TRANSFORMATION OF BT,BR FROM EQUATORIAL TO ECLIPTIC.
C
      DR=.017453292519943
      XS=COS(DPA*DR)*COS(RAP*DR)
      YS=COS(DPA*DR)*SIN(RAP*DR)
      ZS=SIN(DPA*DR)
      CALL RECEQ(DATEM,0.,0.,1.,XEQ,YEQ,ZEQ)
      CALL REQMEQ(DATEM,XEQ,YEQ,ZEQ,XMEQ,YMEQ,ZMEQ,DM,RM)
      CALL CROSS(XS,YS,ZS,XMEQ,YMEQ,ZMEQ,TX,TY,TZ,PRODT)
      CALL CROSS(XS,YS,ZS,TX,TY,TZ,RX,RY,RZ,PRODR)
```

```
      CALL CROSS(XS,YS,ZS,0.,0.,1.,TEX,TEY,TEZ,PRODET)
      CALL CROSS(XS,YS,ZS,TEX,TEY,TEZ,REX,REY,REZ,PRODER)
      CALL DOT(TX,TY,TZ,TEX,TEY,TEZ,ANG1)
      CALL DOT(TX,TY,TZ,REX,REY,REZ,ANG2)
      CALL DOT(RX,RY,RZ,TEX,TEY,TEZ,ANG3)
      CALL DOT(RX,RY,RZ,REX,REY,REZ,ANG4)
      BDT=BT
      BDR=BR
      BT=BDT*COS(ANG1*DR)+BDR*COS(ANG2*DR)
      BR=BDT*COS(ANG3*DR)+BDR*COS(ANG4*DR)
C  ****** END OF TRANSFORMATION.
C
      VEXCESS=DELVMAX-DELTV
C
      IF(DELTV.GT.DELVMAX)GO TO 700
C
      CALL SEESE(DATEM,PD,AE,EE,XI,WE,OE,U,RS,EX,EY,EZ,IS,IE,TSUNIN,DURS
     1UN,TASIN,TASOUT,TEARIN,DUREAR,TAEIN,TAEOUT,DELSOM,DELCOM,DAYS)
C
      IF((KEY1.NE.1).OR.(KEY4.NE.1))GO TO 17
      CALL EVENTS(ELONP,SPECLON,AE,EE,FE,DAYS,PD,DATEM,PER,FPA,RENTRY,RS
     1,TDEST,U,TADEORB,TIMLAND,TIMDEOR,TIMDBST,VDEORB,KK)
      IF(KK.EQ.0)GO TO 700
      DBW=IFIX(TIMDBST)
      DOW=IFIX(TIMDEOR)
      XLW=IFIX(TIMLAND)
      DBF=TIMDBST-DBW
      DOF=TIMDEOR-DOW
      XLF=TIMLAND-XLW
      CALL JULCAL(DBST,DBW,DBF,0)
      CALL JULCAL(DEOR,DOW,DOF,0)
      CALL JULCAL(XLAND,XLW,XLF,0)
      GO TO 20
   17 CONTINUE
      SPECLON=ELONP
      TADEORB=0.
      VDEORB=0.
   20 CONTINUE
C
      IF(KEY1.EQ.2)GO TO 16
      IF(KEY1.EQ.1)GO TO 8
      WRITE(6,800)E1(1),E1(2),E1(3),XM1(1),XM1(2),XM1(3),C3,DLA,DELTV,FE
     1,FH,TPERE,IS,IE,TSUNIN,DURSUN,TASIN,TASOUT,TEARIN,DUREAR,TAEIN,TAE
     2OUT
      GO TO 9
    8 CONTINUE
      IF(KEY4.NE.1)GO TO 18
      WRITE(6,899)E1(2),E1(3),E1(1),E1(4),E1(5),E1(6),XM1(2),XM1(3),XM1(
     11),XM1(4),XM1(5),XM1(6),DBST(2),DBST(3),DBST(1),DBST(4),DBST(5),DB
     2ST(6),DEOR(2),DEOR(3),DEOR(1),DEOR(4),DEOR(5),DEOR(6),XLAND(2),XLA
     3ND(3),XLAND(1),XLAND(4),XLAND(5),XLAND(6),DATEE,DATEM,TIMDBST,TIMD
     4EOR,TIMLAND
```

```
      GO TO 19
   18 CONTINUE
      WRITE(6,900)E1(2),E1(3),E1(1),E1(4),E1(5),E1(6),XM1(2),XM1(3),XM1(
     11),XM1(4),XM1(5),XM1(6),DATEE,DATEM
   19 CONTINUE
      WRITE(6,901)DLA,RAL,C3,TT,DPA,RAP,VHP,GDA,GRA,DCOM,ZAP,ETS,ZAE,ETE
     1,ZAC,ETC,PROBPER,PROBAP,PROBINC,TAL,TAA,HELANG,B,BT,BR
      WRITE(6,902)AZ,EH,ZI,OZ,WZ,FH,TPERH,HPRAD,DECHP,RAHP,VDBH,VXH,VYH,
     1VZH
      WRITE(6,903)AE,EE,XI,OE,WE,FE,TPERE,TADEORB,VDEORB,EPRAD,DECEP,RAE
     1P,PD,VDBE,VXE,VYE,VZE
      WRITE(6,904)DELTV,VEXCESS,DBRAD,PER,VXD,VYD,VZD,PLANE
      WRITE(6,905)IS,TSUNIN,DURSUN,TASIN,TASOUT,IE,TEARIN,DUREAR,TAEIN,T
     1AEOUT
      WRITE(6,906)SPECLON,ELP,GEE,DAYS
      GO TO 9
   16 CONTINUE
      WRITE(6,910)PER,ELP,GEE,XI,DAYS,DELTV
    9 CONTINUE
C
  600 CONTINUE
  700 CONTINUE
      IF(KEY5.EQ.1)GO TO 24
      GO TO 1
C
C
  500 FORMAT(*0*)
  650 FORMAT(*1*)
  800 FORMAT(1X,6F3.0,2F6.2,1X,F5.3,2F6.1,1X,F9.2,2I3,2F9.2,2F8.2,2F9.2,
     12F8.2)
  875 FORMAT(*   LAUNCH  ARRIVAL   C3   DLA  DELTAV F DBST F DBST DBST T
     1IM SO EO  TSUNIN   DURSUN   TASIN  TASOUT   TEARIN   DUREAR   TAEI
     1N  TAEOUT*)
  876 FORMAT(*    DATE     DATE                        ELIPSE HYPERB FROM P
     1ER*/)
  899 FORMAT(20X,*LAUNCH DATE*,12X,*ARRIVAL DATE*,12X,*DEBOOST TIME*,12X
     1,*DEORBIT TIME*,12X,*LANDING TIME*,/,* CALENDAR*,6X,6F3.0,6X,6F3.0
     2,6X,6F3.0,6X,6F3.0,6X,6F3.0,/,* JULIAN*,5X,F18.2,6X,F18.2,6X,F18.2
     3,6X,F18.2,6X,F18.2,/)
  900 FORMAT(21X,*LAUNCH DATE*,13X,*ARRIVAL DATE*,/,* CALENDAR*,7X,6F3.0
     1,7X,6F3.0,/,* JULIAN*,7X,F18.2,7X,F18.2,/)
  901 FORMAT(* INTERPLANETARY FLIGHT PARAMETERS*,// 5X,*DLA =*,E16.8, 5X
     1,*RAL =*,E16.8, 5X,*C3 =*,E16.8, 5X,*TRIP TIME =*,E16.8,/ 5X,*AREO
     2. DEC. S-VECTOR =*,E16.8, 5X,*AREO. R.A. S-VECTOR =*,E16.8, 5X,*HY
     3PER. EXCESS VEL. =*,E16.8,/ 5X,*GEO. DEC. S-VECTOR =*,E16.8, 7X,*G
     4EO. R.A. S-VECTOR =*,E16.8, 5X,*COMUNICATION DIST. =*,E16.8,/ 5X,*
     5ZAP =*,E16.8, 5X,*ETS =*,E16.8, 5X,*ZAE =*,E16.8, 5X,*ETE =*,E16.8
     6,/5X,*ZAC =*,E16.8, 5X,*ETC =*,E16.8,/5X,*PROBE PERIHELION =*,E16.
     78,10X,*PROBE APHELION =*,E16.8, 7X,*PROBE INCLINATION =*,E16.8,/5X
     8,*LAUNCH TRUE ANOM. =*,E16.8, 5X,*ARRIVAL TRUE ANOM. =*,E16.8, 5X,
     9*HELIO. ANGLE TRAVEL =*,E16.8,/5X,*B-VECTOR MAGNITUDE =*,E16.8,15X
```

```
     $,*B DOT T =*,E16.8,17X,*B DOT R =*,E16.8,/)
 902 FORMAT(* ELEMENTS AND DEBOOST PARAMETERS - HYPERBOLA*,// 5X,*A =*,
     1E16.8, 5X,*E =*,E16.8, 5X,*I =*,E16.8, 5X,*CAP.OMEGA =*,E16.8, 5X,
     2*OMEGA =*,E16.8,/ 5X,*DBST TRUE ANOM. =*,E16.8, 7X,*DBST TIME =*,E
     316.8,/5X,*PER. RADIUS =*,E16.8, 5X,*PER. DEC. =*,E16.8, 6X,*PER. R
     4.A. =*,E16.8,/5X,*V AT DBST =*,E16.8, 6X,*VX AT DBST =*,E16.8, 5X,
     5*VY AT DBST =*,E16.8, 5X,*VZ AT DBST =*,E16.8,/)
 903 FORMAT(* ELEMENTS AND DEBOOST PARAMETERS - ELLIPSE*,// 5X,*A =*,
     1E16.8, 5X,*E =*,E16.8, 5X,*I =*,E16.8, 5X,*CAP.OMEGA =*,E16.8, 5X,
     2*OMEGA =*,E16.8,/ 5X,*DBST TRUE ANOM. =*,E16.8, 7X,*DBST TIME =*,E
     316.8, 5X,*DORB.T.A. =*,E16.8,5X,*VDORB =*,E16.8,/5X,
     $         *PER. RADIUS =*,E16.8, 5X,*PER. DEC. =*,E16.8, 6X,*PER. R
     4.A. =*,E16.8, 9X,*PERIOD =*,E16.8,/5X,*V AT DBST =*,E16.8, 6X,*VX
     $AT DBST =*,E16.8, 5X,
     5*VY AT DBST =*,E16.8, 5X,*VZ AT DBST =*,E16.8,/)
 904 FORMAT(* DEBOOST MANEUVER PARAMETERS*,// 5X,*DELTA V =*,E16.8, 5X,
     1*EXCESS DELTA V =*,E16.8, 5X,*RADIUS =*,E16.8,5X,*PER =*,E16.8,/ 5
     2X,*VX =*,E16.8,10X,*VY =*,E16.8,17X,*VZ =*,E16.8, 9X,*PLANE CHANGE
     3 =*,E16.8,/)
 905 FORMAT(* OCCULTATION PARAMETERS*,// 5X,*1ST ORBIT, SUN=*,I3,8X,*TI
     1ME, SUN =*,E16.8, 7X,*DURATION, SUN =*,E16.8,/ 5X,*TRUE ANOM., SUN
     2 IN =*,E16.8, 7X,*TRUE ANOM., SUN OUT =*,E16.8,/ 5X,*1ST ORBIT, EA
     3RTH =*,I3    , 5X,*TIME, EARTH =*,E16.8, 5X,*DURATION, EARTH =*,E16
     4.8,/ 5X,*TRUE ANOM., EARTH IN =*,E16.8, 5X,*TRUE ANOM., EARTH OUT
     5=*E16.8,/)
 906 FORMAT(* LANDING POINT PARAMETERS*,//5X,*LONGITUDE =*,E16.8, 5X,*L
     1ATITUDE =*,E16.8, 5X,*SUN LIGHTING ANGLE*,E16.8, 5X,*DAYS =*,F3.0,
     2/*1*)
 910 FORMAT(/,1X,*PER =*,E12.4,3X,*ELP =*,E12.4,3X,*GEE =*,E12.4,3X,*XI
     1 =*,E12.4,3X,*DAYS =*,E12.4,3X,*DELTA V =*,E12.4)
      END
```

```
      SUBROUTINE PLUG(DATEE,DATEM,C3,DLA,VHP,DPA,RAP,KK)
      USUN=1.3271411E+11
      CALL EEARTH(DATEE,XE,YE,ZE,DXE,DYE,DZE)
      CALL EMARS(DATEM,XM,YM,ZM,DXM,DYM,DZM)
      TT=DATEM-DATEE
      CALL LAMBRT(XE,YE,ZE,XM,YM,ZM,TT*24.*3600.,A,E,XI,W,O,TA1,TA2,USUN
     1,KK)
      IF(KK.EQ.0)RETURN
      CALL CONCAR(A,E,XI,W,O,TA1,X1,Y1,Z1,DX1,DY1,DZ1,USUN)
      DX1E=DX1-DXE
      DY1E=DY1-DYE
      DZ1E=DZ1-DZE
      C3=DX1E**2+DY1E**2+DZ1E**2
      CALL RECEQ(DATEE,DX1E,DY1E,DZ1E,XEQ,YEQ,ZEQ)
      CALL LATLNG(XEQ,YEQ,ZEQ,DLA,RAL)
      CALL CONCAR(A,E,XI,W,O,TA2,X2,Y2,Z2,DX2,DY2,DZ2,USUN)
      DX2M=DX2-DXM
      DY2M=DY2-DYM
      DZ2M=DZ2-DZM
      VHP=SQRT(DX2M**2+DY2M**2+DZ2M**2)
      CALL RECEQ(DATEM,DX2M,DY2M,DZ2M,XEQ,YEQ,ZEQ)
      CALL REQMEQ(DATEM,XEQ,YEQ,ZEQ,XMEQ,YMEQ,ZMEQ,DPA,RAP)
      RETURN
      END




      SUBROUTINE PLUG1(DATEE,DATEM,C3,DLA,VHP,DPA,RAP,TT,RAL,DCOM,PROBPE
     1R,PROBAP,PROBINC,TAL,TAA,HELANG,GDA,GRA,ZAP,ETS,ZAE,ETE,ZAC,ETC,KK
     2)
      USUN=1.3271411E+11
      CALL EEARTH(DATEE,XE,YE,ZE,DXE,DYE,DZE)
      CALL EMARS(DATEM,XM,YM,ZM,DXM,DYM,DZM)
      CALL EEARTH(DATEM,XEA,YEA,ZEA,DXEA,DYEA,DZEA)
      XME=XEA-XM
      YME=YEA-YM
      ZME=ZEA-ZM
      DCOM=SQRT(XME*XME+YME*YME+ZME*ZME)
      TT=DATEM-DATEE
      CALL LAMBRT(XE,YE,ZE,XM,YM,ZM,TT*24.*3600.,A,E,XI,W,O,TA1,TA2,USUN
     1,KK)
      IF(KK.EQ.0)RETURN
      PROBPER=A-A*E
      PROBAP=A+A*E
      PROBINC=XI
      TAL=TA1
      TAA=TA2
      HELANG=TAA-TAL
      CALL CONCAR(A,E,XI,W,O,TA1,X1,Y1,Z1,DX1,DY1,DZ1,USUN)
```

```
  DX1E=DX1-DXE
  DY1E=DY1-DYE
  DZ1E=DZ1-DZE
  C3=DX1E**2+DY1E**2+DZ1E**2
  CALL RECEQ(DATEE,DX1E,DY1E,DZ1E,XEQ,YEQ,ZEQ)
  CALL LATLNG(XEQ,YEQ,ZEQ,DLA,RAL)
  CALL CONCAR(A,E,XI,W,O,TA2,X2,Y2,Z2,DX2,DY2,DZ2,USUN)
  DX2M=DX2-DXM
  DY2M=DY2-DYM
  DZ2M=DZ2-DZM
  CALL DOT(-XM,-YM,-ZM,DX2M,DY2M,DZ2M,ZAP1)
  CALL DOT(XME,YME,ZME,DX2M,DY2M,DZ2M,ZAE1)
  VHP=SQRT(DX2M**2+DY2M**2+DZ2M**2)
  CALL RECEQ(DATEM,DX2M,DY2M,DZ2M,XEQ,YEQ,ZEQ)
  CALL LATLNG(XEQ,YEQ,ZEQ,GDA,GRA)
  CALL REQMEQ(DATEM,XEQ,YEQ,ZEQ,XMEQ,YMEQ,ZMEQ,DPA,RAP)
  SX=XMEQ/VHP
  SY=YMEQ/VHP
  SZ=ZMEQ/VHP
  SRT=SQRT(SX*SX+SY*SY)
  TX=SY /SRT
  TY=-SX/SRT
  TZ=0.

  SX1=DX2M/VHP
  SY1=DY2M/VHP
  SZ1=DZ2M/VHP
  SRT1=SQRT(SX1*SX1+SY1*SY1)
  TX1=SY1 /SRT1
  TY1=-SX1/SRT1
  TZ1=0.
  CALL RECEQ(DATEM,TX1,TY1,TZ1,TXEQ,TYEQ,TZEQ)
  CALL REQMEQ(DATEM,TXEQ,TYEQ,TZEQ,TXMEQ,TYMEQ,TZMEQ,DEC,RA)
  CALL DOT(TX,TY,TZ,TXMEQ,TYMEQ,TZMEQ,ERROR)

  CALL CROSS(SX,SY,SZ,TX,TY,TZ,RX,RY,RZ,RMAG)
  CALL VECTOR(DATEM,X1,X2,X3,X,X,X,SUNX,SUNY,SUNZ,EX,EY,EZ,CX,CY,CZ,
 14)
  SUNS=SX*SUNX+SY*SUNY+SZ*SUNZ
  SUNT=TX*SUNX+TY*SUNY+TZ*SUNZ
  SUNR=RX*SUNX+RY*SUNY+RZ*SUNZ
  EAS =SX*EX  +SY*EY  +SZ*EZ
  EAT =TX*EX  +TY*EY  +TZ*EZ
  EAR =RX*EX  +RY*EY  +RZ*EZ
  CAS =SX*CX  +SY*CY  +SZ*CZ
  CAT =TX*CX  +TY*CY  +TZ*CZ
  CAR =RX*CX  +RY*CY  +RZ*CZ
  CALL LATLNG(SUNT,SUNR,SUNS,SDEC,SRA)
  CALL LATLNG(EAT,EAR,EAS,EDEC,ERA)
  CALL LATLNG(CAT,CAR,CAS,CDEC,CRA)
```

```
      ETS=SRA+180.
      ETE=ERA+180.
      ETC=CRA+180.
      ZAP =90,-SDEC
      ZAE=90.-EDEC
      ZAC=90.-CDEC
      IF(ABS(ZAP1-ZAP).GT.1.)WRITE(6,100)ZAP1,ZAP
      IF(ABS(ZAE1-ZAE).GT.1.)WRITE(6,200)ZAE1,ZAE
  100 FORMAT(2E16.8,* ERROR IN ZAP*)
  200 FORMAT(2E16.8,* ERROR IN ZAE*)
      RETURN
      END



      SUBROUTINE COMPEL(PER,ELP,GEE,XI,CAPW,XITW,DATEM,HA,HP,AE,EE,DAYS,
     1RS,KEY2,KEY3,ELONP)
C     ALL ANGLES INPUT IN DEGREES AND OUTPUT IN DEGREES.
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      XSIN(X)=SIN(DR*X)
      XCOS(X)=COS(DR*X)
      IF(ABS(XI).LT.ABS(ELP))GO TO 50
      DR=.017453292519943
      DJUL=DATEM+DAYS
      ICODE=4
      CALL VECTOR(DJUL,ELS,ELONS,DECE,RAE,DECC,RAC,SX,SY,SZ,EX,EY,EZ,CX,
     1CY,CZ,ICODE)
      IF(ABS(SIGN(GEE,ELP)+ELS).LE.ABS(ELP))GO TO 50
      RD=57.295779513082321
      ARG=(XCOS(GEE)-XSIN(ELS)*XSIN(ELP))/(XCOS(ELS)*XCOS(ELP))
      IF(ABS(ARG).GT.1.)GO TO 50
      ARC1=ASIN(XSIN(ELP)/XSIN(XI))*RD
      ARC2=ASIN((XSIN(ELP)/XCOS(ELP))*(XCOS(XI)/XSIN(XI)))*RD
      GO TO (10,20,30,40)KEY3
   10 CONTINUE
      ELONP=(ACOS(ARG))*RD+ELONS
      XITW=PER+180.-ARC1
      CAPW=ELONP-180.+ARC2
      GO TO 100
   20 CONTINUE
      ELONP=(ACOS(ARG))*RD+ELONS
      XITW=PER+ARC1
      CAPW=ELONP-ARC2
      GO TO 100
   30 CONTINUE
      ELONP=ELONS-(ACOS(ARG))*RD
      XITW=PER+180.-ARC1
      CAPW=ELONP-180.+ARC2
      GO TO 100
   40 CONTINUE
```

```
      ELONP=ELONS-(ACOS(ARG))*RD
      XITW=PER+ARC1
      CAPW=ELONP-ARC2
      GO TO 100
  100 CONTINUE
      XITW=ANGLE(XITW)
      CAPW=ANGLE(CAPW)
      RA=RS    +HA
      RP=RS    +HP
      AE= (RA+RP)/2.
      EE=(RA-RP)/(2.*AE)
      RETURN
   50 KEY2=0
      RETURN
      END



      SUBROUTINE FUDGE(XJ2,XI,AE,EE,DELCOM,DELSOM,PD,XITW,CAPW,OE,WE,U,D
     1AYS,RS)
      XN=SQRT(U/AE**3)
      PI=3.14159265358979З
      DR=PI/180.
      C1=(RS /(AE*(1.-EE*EE)))**2
      DELSOM=6.*PI*XJ2*C1*(1.-1.25*SIN(XI*DR)**2)
      DELCOM=-3.*PI*XJ2*C1*COS(XI*DR)
      ENBAR=XN*(1.+1.5*C1*XJ2*SQRT(1.-EE*EE)*(1.-1.5*SIN(XI*DR)**2))
      PD=2.*PI/ENBAR/86400.
      B=DAYS/PD
      OE=CAPW-(B*DELCOM)*180./PI
      WE=XITW-(B*DELSOM)*180./PI
      RETURN
      END



      SUBROUTINE SEESE(DATEM,PD,AE,EE,XI,WE,OE,U,RS,EX,EY,EZ,IS,IE,TSUNI
     1N,DURSUN,       TASIN,TASOUT,TEARIN,DUREAR,TAEIN,TAEOUT,DELSOM,DELC
     2OM,DAYS)
      IS=0
      IE=0
      TSUNIN=0.
      DURSUN=0.
      TASIN=0.
      TASOUT=0.
      TEARIN=0.
      DUREAR=0.
      TAEIN=0.
      TAEOUT=0.
      ISTOP=DAYS/PD+1
```

```
      DO 20 J=1,ISTOP
      TIME=DATEM+PD*FLOAT(J-1)
      CALL VECTOR(TIME,DECS,RAS,DECE,RAE,DECC,RAC,SX,SY,SZ,EX,EY,EZ,CX,C
     1Y,CZ,4)
      CALL OCCULT(AE,EE,XI,WE+PD*FLOAT(J-1)*DELSOM,OE+PD*FLOAT(J-1)*DELC
     1OM,U,RS,SX,SY,SZ,DURSU,TSUNI,ALT1,TASI,DEC1,RA1,T2,ALT2,TASOU,DEC2
     1,RA2,KS)
      CALL OCCULT(AE,EE,XI,WE+PD*FLOAT(J-1)*DELSOM,OE+PD*FLOAT(J-1)*DELC
     1OM,U,RS,EX,EY,EZ,DURE,TEARI,ALT1,TAEI,DEC1,RA1,T2,ALT2,TAEOU,DEC2,
     1RA2,KE)
      IF(IS.NE.0)GO TO 10
      IF(KS.EQ.1)GO TO 8
      GO TO 10
    8 IS=J
      TSUNIN=TSUNI
      DURSUN=DURSU
      TASIN=TASI
      TASOUT=TASOU
   10 IF(IE.NE.0)GO TO 15
      IF(KE.EQ.1)GO TO 11
      GO TO 20
   11 IE=J
      TEARIN=TEARI
      DUREAR=DURE
      TAEIN=TAEI
      TAEOUT=TAEOU
   15 IF(IE.NE.0 .AND. IS.NE.0)RETURN
   20 CONTINUE
      RETURN
      END


      SUBROUTINE EVENTS(ELONP,SPECLON,AE,EE,FE,DAYS,PD,DATEM,PER,FPA,REN
     1TRY,RS,TDEST,U,TADEORB,TIMLAND,TIMDEOR,TIMDBST,VDEORB,KK)
C
C     COMPUTES EVENT TIMES FOR LANDING,DEORBIT AND DEBOOST, GIVEN LANDING
C     POINT AND ELEMENTS OF ELLIPSE AND TRUE ANOMALY OF DEORBIT.
C
C     CORRECT FOR TIME OF DAY.
C
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      PMDOT=350.891962
      HA=145.845+350.891962*(DATEM+DAYS-2418322.)
      DELTLON=ELONP-SPECLON-HA
      DELTLON=ANGLE(DELTLON)
      DELTJD=DELTLON/PMDOT
      TIMLAND=DATEM+DAYS+DELTJD
C
C
      IF(TDEST.NE.0.)GO TO 1
      PERO=-PER
      CALL CONFPA(AE,EE,TADEORB,PERO,RS,RENTRY,FPA,U,AL,EL,FLO,FLD,VDEOR
     1B,THETA,KK)
```

```
      IF(KK.EQ.0)RETURN
      CALL TCONIC(U,EL,AL,FLD,TDEOR)
      CALL TCONIC(U,EL,AL,FLO,TLAND)
      TDEOR=TDEOR/86400.
      TLAND=TLAND/86400.
      PDT=2.*3.1415926536*SQRT(AL*AL*AL)/SQRT(U)
      PDT=PDT/86400.
      IF(TDEOR.GT.0.)DELDEOR=TLAND+PDT-TDEOR
      IF(TDEOR.LT.0.)DELDEOR=TLAND-TDEOR
      GO TO 2
    1 CONTINUE
      DELDEOR=TDLST
    2 CONTINUE
      TIMDEOR=TIMLAND-DELDEOR
C
C     CALCULATE DEBOOST TIME.
C
      BACKUP=DAYS/PD
      DELTIM=IFIX(BACKUP)
      CALL TCONIC(U,EL,AL,FL,TPERE)
      CALL TCONIC(U,EE,AE,TADEORB,TDEORB)
      TPERE=TPERE/86400.
      TDEORB=TDEORB/86400.
      IF(TDEORB.GT.0.)DELODB=TDEORB-TPERE
      IF(TDEORB.LT.0.)DELODB=TDEORB-TPERE+PD
      TIMDBST =TIMDEOR-DELTIM-DELODB
      RETURN
      END
```

```
      SUBROUTINE DEBOOST(A1,E1,I1,W1,O1,F1,U,LATS.LONS,VINF,RPMIN,INC,AZ
     $,E2,I2,W2,O2,F2,BT,BR,DELTV,TPERE,M)
      REAL I1,I2,LATS,LONS,NX,NY,NZ,N,IZP,IZM
      DATA DR,RD,PI/.17453292519943E-1,57.295779513082321,3.141592653589
     $793/
      ANGLE(X)=AMOD(X,2.*PI)+PI-SIGN(PI,X)
      DIMENSION DV(360),TA(360),HYP(360,6)
      DIMENSION PX(3),PY(3,6)
      CLAT=COS(DR*LATS)
      SLAT=SIN(DR*LATS)
      CLON=COS(DR*LONS)
      SLON=SIN(DR*LONS)
      SX=CLAT*CLON
      SY=CLAT*SLON
      SZ=SLAT
      DO 1 I=1,360,M
      TA(I)=FLOAT(I)-180.
      F=DR*TA(I)
      CWF=COS(DR*W1+F)
      SWF=SIN(DR*W1+F)
      CI=COS(DR*I1)
      SI=SIN(DR*I1)
```

```
      CO=COS(DR*O1)
      SO=SIN(DR*O1)
      RX=CWF*CO-SWF*SO*CI
      RY=CWF*SO+SWF*CO*CI
      RZ=SWF*SI
      AZ=-U/VINF**2
      RO=A1*(1.-E1*E1)/(1.+E1*COS(F))
      RS=RX*SX+RY*SY+RZ*SZ
      A=AZ**2
      B=RO**2*RS**2+2.*RO*AZ*RS-RO**2-2.*AZ**2           +2.*AZ*RO
      C=2.*RO**2*RS-2.*RO*AZ*RS+2.*RO**2+AZ**2-2.*AZ*RO
      DV(I)=1.E20
      TEST=B*B-4.*A*C
      IF(TEST.LT.0.)GO TO 1
      DISC=SQRT(TEST)
      EZP=SQRT((-B+DISC)/(2.*A))
      EZM=SQRT((-B-DISC)/(2.*A))
      IF(EZM.LE.1.)EZM=EZP
      PHIP=ACOS(1./EZP)
      PHIM=ACOS(1./EZM)
      FZP=ACOS((AZ*(1.-EZP**2)-RO)/(EZP*RO))
      FZM=ACOS((AZ*(1.-EZM**2)-RO)/(EZM*RO))
      IF(ABS(COS(ANGLE(PHIP-FZP))-RS).GT.1.E-7)FZP=-FZP
      IF(ABS(COS(ANGLE(PHIM-FZM))-RS).GT.1.E-7)FZM=-FZM
      NX=RY*SZ-RZ*SY
      NY=RZ*SX-RX*SZ
      NZ=RX*SY-RY*SX
      N=SQRT(NX**2+NY**2+NZ**2)
      IZP=ACOS(NZ/N)
      IF(ANGLE(PHIP-FZP).GT.PI)IZP=ACOS(-NZ/N)
      IZM=ACOS(NZ/N)
      IF(ANGLE(PHIM-FZM).GT.PI)IZM=ACOS(-NZ/N)
      IF((IZP.LE.PI/2..AND.INC .EQ.1).OR.(IZP.GT.PI/2..AND.INC .EQ.2))2,
     $3
    2 EZ=EZP
      IZ=RD*IZP
      FZ=RD*FZP
      PHI=RD*PHIP
      GO TO 4
    3 EZ=EZM
      IZ=RD*IZM
      FZ=RD*FZM
      PHI=RD*PHIM
    4 RPZ=AZ*(1.-EZ)
      DV(I)=1.E20
      IF(RPZ.LT.RPMIN)GO TO 1
      WS=ASIN(SZ/SIN(DR*IZ))
      WZ=RD*WS-PHI
      IF(ABS(RZ-SIN(DR*(WZ+FZ))*SIN(DR*IZ)).GT.1.E-7)WZ=180.-RD*WS-PHI
      DET=COS(DR*(WZ+PHI))**2+COS(DR*IZ)**2*SIN(DR*(WZ+PHI))**2
      CO=(COS(DR*(WZ+PHI))*SX+COS(DR*IZ)*SIN(DR*(WZ+PHI))*SY)/DET
      SO=(-COS(DR*IZ)*SIN(DR*(WZ+PHI))*SX+COS(DR*(WZ+PHI))*SY)/DET
      OZ=RD*ATAN2(SO,CO)
```

```
      HYP(I,1)=AZ
      HYP(I,2)=EZ
      HYP(I,3)=IZ
      HYP(I,4)=WZ
      HYP(I,5)=OZ
      F1=RD*F
      HYP(I,6)=FZ
      CALL CONCAR(AZ,EZ,IZ,WZ,OZ,FZ,X,Y,Z,DX,DY,DZ,U)
      CALL CONCAR(A1,E1,I1,W1,O1,F1,X,Y,Z,VX,VY,VZ,U)
      DV(I)=SQRT((DX-VX)**2+(DY-VY)**2+(DZ-VZ)**2)
 1    CONTINUE
      IMIN=0
      DELTV=1.E20
      DO 8 I=1,360,M
      IF(DV(I).GT.DELTV)GO TO 8
      IMIN=I
      DELTV=DV(I)
 8    CONTINUE
      IMINM=IMIN-M
      IMINP=IMIN+M
      IF(IMINM.LE.0.OR.IMINP.GE.361)GO TO 6
      IF(DV(IMIN-M).EQ.1.E20.OR.DV(IMIN+M).EQ.1.E20)GO TO 6
      PX(1)=TA(IMIN-M)
      PX(2)=TA(IMIN)
      PX(3)=TA(IMIN+M)
      PY(1,1)=DV(IMIN-M)
      PY(2,1)=DV(IMIN)
      PY(3,1)=DV(IMIN+M)
      DO 5 I=2,6
      PY(1,I)=HYP(IMIN-M,I)
      PY(2,I)=HYP(IMIN,I)
 5    PY(3,I)=HYP(IMIN+M,I)
      CALL PARIN(F1,DELTV,PX,PY(1,1),0)
      CALL PARIN(F1,EZ,PX,PY(1,2),1)
      CALL PARIN(F1,IZ,PX,PY(1,3),1)
      CALL PARIN(F1,WZ,PX,PY(1,4),1)
      CALL PARIN(F1,OZ,PX,PY(1,5),1)
      CALL PARIN(F1,FZ,PX,PY(1,6),1)
      GO TO 9
 6    F1=TA(IMIN)
      AZ=HYP(IMIN,1)
      EZ=HYP(IMIN,2)
      IZ=HYP(IMIN,3)
      WZ=HYP(IMIN,4)
      OZ=HYP(IMIN,5)
      FZ=HYP(IMIN,6)
 9    B=-AZ*SQRT(EZ*EZ-1.)
      BT=B*COS(DR*IZ)/CLAT
      BR=B*SIN(DR*IZ)*COS(DR*(LONS-OZ))
      CALL TCONIC(U,E1,A1,F1,TPERE)
      RETURN
      END
```

```
 SUBROUTINE CBST1(A1,E1,I1,W1,O1,F1,U,LATS,LONS,VINF,RPMIN,INC,AZ,E
1Z,IZ,WZ,OZ,FZ,BT,BR,DELTV,TPERE,TPERH,B,HPRAD,DBRAD,VXH,VYH,VZH,VX
2E,VYE,VZE,VXD,VYD,VZD,DECHP,RAHP,DECEP,RAEP,VDBH,VDBE,PLANE,M)
 REAL I1,IZ,LATS,LONS,NX,NY,NZ,N,IZP,IZM
 DATA DR,RD,PI/.17453292519943E-1,57.295779513082321,3.141592653589
$793/
 ANGLE(X)=AMOD(X,2.*PI)+PI-SIGN(PI,X)
 DIMENSION DV(360),TA(360),HYP(360,6)
 DIMENSION PX(3),PY(3,6)
 CLAT=COS(DR*LATS)
 SLAT=SIN(DR*LATS)
 CLON=COS(DR*LONS)
 SLON=SIN(DR*LONS)
 SX=CLAT*CLON
 SY=CLAT*SLON
 SZ=SLAT
 DO 1 I=1,360,M
 TA(I)=FLOAT(I)-180.
 F=DR*TA(I)
 CWF=COS(DR*W1+F)
 SWF=SIN(DR*W1+F)
 CI=COS(DR*I1)
 SI=SIN(DR*I1)
 CO=COS(DR*O1)
 SO=SIN(DR*O1)
 RX=CWF*CO-SWF*SO*CI
 RY=CWF*SO+SWF*CO*CI
 RZ=SWF*SI
 AZ=-U/VINF**2
 RO=A1*(1.-E1*E1)/(1.+E1*COS(F))
 RS=RX*SX+RY*SY+RZ*SZ
 A=AZ**2
 B=RO**2*RS**2+2.*RO*AZ*RS-RO**2-2.*AZ**2          +2.*AZ*RO
 C=2.*RO**2*RS-2.*RO*AZ*RS+2.*RO**2+AZ**2-2.*AZ*RO
 DV(I)=1.E20
 TEST=B*B-4.*A*C
 IF(TEST.LT.0.)GO TO 1
 DISC=SQRT(TEST)
 EZP=SQRT((-B+DISC)/(2.*A))
 EZM=SQRT((-B-DISC)/(2.*A))
 IF(EZM.LE.1.)EZM=EZP
 PHIP=ACOS(1./EZP)
 PHIM=ACOS(1./EZM)
 FZP=ACOS((AZ*(1.-EZP**2)-RO)/(EZP*RO))
 FZM=ACOS((AZ*(1.-EZM**2)-RO)/(EZM*RO))
 IF(ABS(COS(ANGLE(PHIP-FZP))-RS).GT.1.E-7)FZP=-FZP
 IF(ABS(COS(ANGLE(PHIM-FZM))-RS).GT.1.E-7)FZM=-FZM
 NX=RY*SZ-RZ*SY
 NY=RZ*SX-RX*SZ
 NZ=RX*SY-RY*SX
 N=SQRT(NX**2+NY**2+NZ**2)
 IZP=ACOS(NZ/N)
 IF(ANGLE(PHIP-FZP).GT.PI)IZP=ACOS(-NZ/N)
 IZM=ACOS(NZ/N)
 IF(ANGLE(PHIM-FZM).GT.PI)IZM=ACOS(-NZ/N)
```

```
      IF((IZP.LE.PI/2..AND.INC .EQ.1).OR.(IZP.GT.PI/2..AND.INC .EQ.2))2,
     $3
 2    EZ=EZP
      IZ=RD*IZP
      FZ=RD*FZP
      PHI=RD*PHIP
      GO TO 4
 3    EZ=EZM
      IZ=RD*IZM
      FZ=RD*FZM
      PHI=RD*PHIM
 4    RPZ=AZ*(1.-EZ)
      DV(I)=1.E20
      IF(RPZ.LT.RPMIN)GO TO 1
      WS=ASIN(SZ/SIN(DR*IZ))
      WZ=RD*WS-PHI
      IF(ABS(RZ-SIN(DR*(WZ+FZ))*SIN(DR*IZ)).GT.1.E-7)WZ=180.-RD*WS-PHI
      DET=COS(DR*(WZ+PHI))**2+COS(DR*IZ)**2*SIN(DR*(WZ+PHI))**2
      CO=(COS(DR*(WZ+PHI))*SX+COS(DR*IZ)*SIN(DR*(WZ+PHI))*SY)/DET
      SC=(-COS(DR*IZ)*SIN(DR*(WZ+PHI))*SX+COS(DR*(WZ+PHI))*SY)/DET
      OZ=RD*ATAN2(SC,CC)
      HYP(I,1)=AZ
      HYP(I,2)=EZ
      HYP(I,3)=IZ
      HYP(I,4)=WZ
      HYP(I,5)=OZ
      F1=RD*F
      HYP(I,6)=FZ
      CALL CONCAR(AZ,EZ,IZ,WZ,OZ,FZ,X,Y,Z,DX,DY,DZ,U)
      CALL CONCAR(A1,E1,I1,W1,O1,F1,X,Y,Z,VX,VY,VZ,U)
      DV(I)=SQRT((DX-VX)**2+(DY-VY)**2+(DZ-VZ)**2)
 1    CONTINUE
      IMIN=0
      DELTV=1.E20
      DO 8 I=1,360,M
      IF(DV(I).GT.DELTV)GO TO 8
      IMIN=I
      DELTV=DV(I)
 8    CONTINUE
      IMINM=IMIN-M
      IMINP=IMIN+M
      IF(IMINM.LE.0.OR.IMINP.GE.361)GO TO 6
      IF(DV(IMIN-M).EQ.1.E20.OR.DV(IMIN+M).EQ.1.E20)GO TO 6
      PX(1)=TA(IMIN-M)
      PX(2)=TA(IMIN)
      PX(3)=TA(IMIN+M)
      PY(1,1)=DV(IMIN-M)
      PY(2,1)=DV(IMIN)
      PY(3,1)=DV(IMIN+M)
      DO 5 I=2,6
      PY(1,I)=HYP(IMIN-M,I)
      PY(2,I)=HYP(IMIN,I)
 5    PY(3,I)=HYP(IMIN+M,I)
      CALL PARIN(F1,DELTV,PX,PY(1,1),0)
      CALL PARIN(F1,EZ,PX,PY(1,2),1)
```

```
      CALL PARIN(F1,IZ,PX,PY(1,3),1)
      CALL PARIN(F1,WZ,PX,PY(1,4),1)
      CALL PARIN(F1,OZ,PX,PY(1,5),1)
      CALL PARIN(F1,FZ,PX,PY(1,6),1)
      GO TO 9
    6 F1=TA(IMIN)
      AZ=HYP(IMIN,1)
      EZ=HYP(IMIN,2)
      IZ=HYP(IMIN,3)
      WZ=HYP(IMIN,4)
      OZ=HYP(IMIN,5)
      FZ=HYP(IMIN,6)
    9 B=-AZ*SQRT(EZ*EZ-1.)
      BT=B*COS(DR*IZ)/CLAT
      BR=B*SIN(DR*IZ)*COS(DR*(LONS-OZ))
      CALL TCONIC(U,E1,A1,F1,TPERE)
      CALL TCONIC(U,EZ,AZ,FZ,TPERH)
      HPRAD =AZ-AZ*EZ
      DBRAD=(A1-A1*E1*E1)/(1.+E1*COS(DR*F1))
      CALL CONCAR(AZ,EZ,IZ,WZ,OZ,FZ,X,Y,Z,VXH,VYH,VZH,U)
      CALL CONCAR(A1,E1,I1,W1,O1,F1,X,Y,Z,VXE,VYE,VZE,U)
      VDBH=SQRT(VXH*VXH+VYH*VYH+VZH*VZH)
      VDBE=SQRT(VXE*VXE+VYE*VYE+VZE*VZE)
      VXD=VXE-VXH
      VYD=VYE-VYH
      VZD=VZE-VZH
      CALL CONCAR(AZ,EZ,IZ,WZ,OZ,0.,XPH,YPH,ZPH,DX,DY,DZ,U)
      CALL CONCAR(A1,E1,I1,W1,O1,0.,XPE,YPE,ZPE,DX,DY,DZ,U)
      CALL LATLNG(XPH,YPH,ZPH,DECHP,RAHP)
      CALL LATLNG(XPE,YPE,ZPE,DECEP,RAEP)
      WXE=SIN(DR*I1)*SIN(DR*O1)
      WYE=-COS(DR*O1)*SIN(DR*I1)
      WZE=COS(DR*I1)
      WXH=SIN(DR*IZ)*SIN(DR*OZ)
      WYH=-COS(DR*OZ)*SIN(DR*IZ)
      WZH=COS(DR*IZ)
      CALL DOT(WXE,WYE,WZE,WXH,WYH,WZH,PLANE)
      RETURN
      END


      SUBROUTINE CALJUL(WJD,FJD,WND,FD,X)
      DIMENSION X(6),A(12)
      D50=2433282.
      YD=X(1)-48.
      YL=YD/4.
      KYL=YL
      CK=KYL
      IF(YL-CK)1,1,3
    1 IF(X(2)-2.)4,4,3
    3 DS=CK
      GO TO 5
    4 DS=CK-1.
    5 DS=DS+365.*(YD-2.)
```

```
      DO 6 I=1,12
    6 A(I)=1.0
      K=X(2)
      DO 7 I=K,12
    7 A(I)=0.0
      DS=DS+31.*(A(1)+A(3)+A(5)+A(7)+A(8)+A(10)+A(12))
     1+30.*(A(4)+A(6)+A(9)+A(11))+28.*A(2)
      DS=DS+X(3)-1.
      WND=DS
      FD=X(4)/24.+X(5)/1440.+X(6)/86400.
      IF(FD-.4999999)9,8,8
    8 FJD=FD-.5
      WJD=1.
      GO TO 10
    9 FJD=FD+.5
      WJD=0.
   10 WJD=D50+WJD+WND
      RETURN
      END
```

```
      SUBROUTINE CONCAR(A,E,XI,W,O,F,X,Y,Z,DX,DY,DZ,U)
      DATA DR/.017453292519943/
      FR=DR*F
      WFR=DR*(W+F)
      OR=DR*O
      XIR=DR*XI
      DEN=1.+E*COS(FR)
      R=A*(1.-E*E)/DEN
      V=SQRT(U*(2./R-1./A))
      GAM=ATAN(E*SIN(FR)/DEN)
      WFGR=WFR-GAM
      CWF=COS(WFR)
      SWF=SIN(WFR)
      SO=SIN(OR)
      CO=COS(OR)
      SI=SIN(XIR)
      CI=COS(XIR)
      SWFG=SIN(WFGR)
      CWFG=COS(WFGR)
      X=R*(CWF*CO-SWF*SO*CI)
      Y=R*(CWF*SO+SWF*CO*CI)
      Z=R*SWF*SI
      DX=V*(-SWFG*CO-CWFG*SO*CI)
      DY=V*(-SWFG*SO+CWFG*CO*CI)
      DZ=V*CWFG*SI
      RETURN
      END
```

```
      SUBROUTINE CONFPA(AO,EO,FO,PERO,RS,RE,FPAE,U,AL,EL,FLD,FLO,DELV,TH
     1ETA,KK)
      DIMENSION P(2)
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      DELV=0.
      DR=.017453292519
      RD=57.2957795130
      KK=0
      ANG12=ANGLE(PERO-FO)
      S12=SIN(ANG12*DR)
      C12=COS(ANG12*DR)
      CFPA=COS(FPAE*DR)
      SFO=SIN(FO*DR)
      CFO=COS(FO*DR)
      R1=AO*(1.-EO*EO)/(1.+EO*CFO)
      R2=RS
      VO=SQRT(U*(2./R1-1./AO))
      FPAO=ASIN(EO*SFO/SQRT(1.+2.*EO*CFO+EO*EO))*RD
C
      A=-R2*R2-R1*R1+2.*R1*R2*C12+(R1*R2*S12/RE/CFPA)**2
      B=2.*(R1*R2*R2+R1*R1*R2-R1*R1*R2*C12-R1*R2*R2*C12-(R1*R2*S12)**2/R
     1E)
      C=R1*R1*R2*R2*(-2.+2.*C12+S12*S12)
C
      CALL QADRAT(A,B,C,P(1),P(2),KK)
      IF(KK.EQ.0) GO TO 800
C
      DO 1 I=1,KK
      IF(P(I).LE.0.) GO TO 1
      E2=1.-2.*P(I)/RE+(P(I)/RE/CFPA)**2
      IF(E2.LT.0.) GO TO 1
      EL=SQRT(E2)
      AL=P(I)/(1.-E2)
      CF2=(P(I)-R2)/EL/R2
      SF2=-SQRT(1.-CF2*CF2)
      CHECK=R2*(1.+EL*CF2)-R1*(1.+EL*(C12*CF2+S12*SF2))
      IF(ABS(CHECK).GT.1.) GO TO 1
      I=2
      F2=ANGLE(ATAN2(SF2,CF2)*RD)
      F1=F2-ANG12
      FPAL1=ASIN(EL*SIN(F1*DR)/SQRT(1.+2.*EL*COS(F1*DR)+EL*EL))*RD
      VL1=SQRT(U*(2./R1-1./AL))
      DELV=SQRT(VO*VO+VL1*VL1-2.*VO*VL1*COS((FPAO-FPAL1)*DR))
      STH=VL1/DELV*SIN((FPAO-FPAL1)*DR)
      CTH=(VO*VO+DELV*DELV-VL1*VL1)/2./DELV/VO
      THETA=ATAN2(STH,CTH)*RD
    1 CONTINUE
C
      IF(DELV.LT..001) GO TO 800
      FLO=F1
      FLD=F2
      KK=1
  800 RETURN
      END
```

```
      SUBROUTINE CROSS(X1,Y1,Z1,X2,Y2,Z2,PX,PY,PZ,PRODUCT)
C      CALCULATE VECTOR CROSS PRODUCT
      PX=Y1*Z2-Z1*Y2
      PY=Z1*X2-X1*Z2
      PZ=X1*Y2-Y1*X2
      PRODUCT=SQRT(PX*PX+PY*PY+PZ*PZ)
      RETURN
      END


      SUBROUTINE CUBIC(A,B,C,D,X1,X2,X3,KK)
C
C     THIS SUBROUTINE SOLVES THE EQUATION AX**3 +BX**2 +CX +D = 0 FOR
C     THE REAL ROOTS
C
C     A,B,C,D - COEFFICIENT OF THE DIFFERENT POWERS OF X
C     X1,X2,X3 - REAL ROOTS OF THE EQUATION
C     KK - NUMBER OF REAL ROOTS
C
      CBRT(X)=SIGN(ABS(X)**.333333333,X)
      KK=0
      PI=3.1415927
      IF(A.LT..1E-30.AND.A.GT.-.1E-30) GO TO 4
      P=B/A
      Q=C/A
      R=D/A
      SA=(3.*Q-P**2)/3.
      SB=(2.*P**3-9.*P*Q+27.*R)/27.
      DEL=(4.*Q**3-Q**2*P**2-18.*Q*P*R+27.*R**2+4.*P**3*R)/108.
      IF(DEL.LT..1E-30.AND.DEL.GT.-.1E-30) GO TO 3
      IF(DEL)1,3,2
    1 KK=3
      CPHI=-SB/2./SQRT(SA**3/(-27.))
      IF(ABS(CPHI).GT.1.)GO TO 10
      SPHI=SQRT(1.-CPHI**2)
      PHI=ATAN2(SPHI,CPHI)
      GO TO 11
   10 SPHI=SQRT((27.*DEL)/SA**3)
C     SINCE  FOR  SMALL  ANGLES  SPHI=PHI
      BETA=SPHI
      IF(-SB.GT.0.)PHI=BETA
      IF(-SB.LT.0.)PHI=3.141592653589793-BETA
   11 EO=2.*SQRT(-SA/3.)
      X1=EO*COS(PHI/3.)-P/3.
      X2=EO*COS(PHI/3.+2.*PI/3.)-P/3.
      X3=EO*COS(PHI/3.+4.*PI/3.)-P/3.
      GO TO 7
    2 KK=1
      X1=CBRT(-SB/2.+SQRT(DEL))+CBRT(-SB/2.-SQRT(DEL))-P/3.
      GO TO 7
    3 KK=3
      X1=2.*CBRT(-SB/2.)-P/3.
```

```
      X2=CBRT(SB/2.)-F/3.
      X3=X2
      GO TO 7
    4 CONTINUE
      DIS=C**2-4.*B*D
      IF(DIS)7,5,5
    5 X1=(-C+SQRT(DIS))/2./B
      X2=(-C-SQRT(DIS))/2./B
      KK=2
    7 CONTINUE
      RETURN
      END



      SUBROUTINE DETER(A,B)
      DOUBLE PRECISION AP,BP
      DIMENSION B(3,3),BP(3,3)
      DO 10 I=1,3
      DO 10 J=1,3
      BP(I,J)=0.D0
      BP(I,J)=DBLE(B(I,J))
   10 CONTINUE
      AP=BP(1,1)*BP(2,2)*BP(3,3)-BP(3,1)*BP(2,2)*BP(1,3)+
     1BP(1,2)*BP(2,3)*BP(3,1)-BP(1,2)*BP(2,1)*BP(3,3)+BP(1,3)*BP(2,1)
     2*BP(3,2)-BP(1,1)*BP(2,3)*BP(3,2)
      A=SNGL(AP)
      RETURN
      END


      SUBROUTINE DOT(X1,Y1,Z1,X2,Y2,Z2,ANGLE)
C
C     THIS SUBROUTINE COMPUTES THE ANGLE BETWEEN TWO VECTORS
C
C     X1,Y1,Z1 - COMPONENTS OF THE VECTOR R1
C     X2,Y2,Z2 - COMPONENTS OF THE VECTOR R2
C     ANGLE - ANGLE BETWEEN VECTORS R1 AND R2
C
      RD=57.2957795130823
      R1=SQRT(X1*X1+Y1*Y1+Z1*Z1)
      R2=SQRT(X2*X2+Y2*Y2+Z2*Z2)
C
      ANGLE=ACOS((X1*X2+Y1*Y2+Z1*Z2)/R1/R2)*RD
      RETURN
      END
```

```
      SUBROUTINE EEARTH(JD,XHE,YHE,ZHE,DXHE,DYHE,DZHE)
C
C     THIS SUBROUTINE COMPUTES THE HELIOCENTRIC POSITION AND VELOCITY OF
C     THE EARTH IN MEAN EQUINOX AND ECLIPTIC OF DATE COORDINATE SYSTEM.
C     THIS ROUTINE CALLS SUBROUTINES TINVS AND CONCAR.
C
C     JD - JULIAN DATE
C     XHE,YHE,ZHE - POSITION OF EARTH
C     DXHE,DYHE,DZHE - VELOCITY OF EARTH
C
      REAL JD
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      DR=.017453292519943
      RD=57.295779513 0823
      USUN=1.3271411E+11
      AU=149598845.
C
      D=JD-2415020.
      CD=D/10000.
      TE=D/36525.
C
      AE=1.00000023*AU
      EE=0.01675104-0.00004180*TE-0.000000126*TE**2
      XIE=0.0
      WE=101.220833+0.000047068*D+0.0000339*CD**2+0.00000007*CD**3
      OE=0.0
      XME=ANGLE(358.475845+0.985600267*D-0.0000112*CD**2-0.00000007*CD**
     13)
C
      CALL TINVS(XME*DR,EE,ECE,FE)
      CALL CONCAR(AE,EE,XIE,WE,OE,FE*RD,XHE,YHE,ZHE,DXHE,DYHE,DZHE,USUN)
C
      RETURN
      END


      SUBROUTINE EMARS(JD,XHM,YHM,ZHM,DXHM,DYHM,DZHM)
C
C     THIS SUBROUTINE COMPUTES THE MEAN HELIOCENTRIC POSITION AND
C     VELOCITY OF MARS IN THE MEAN EARTH EQUINOX AND ECLIPTIC OF DATE
C     COORDINATE SYSTEM. THIS ROUTINE CALLS SUBROUTINES TINVS AND CONCAR
C
C     JD - JULIAN DATE
C     XHM,YHM,ZHM - POSITION OF MARS
C     DXHM,DYHM,DZHM - VELOCITY OF MARS
C
      REAL JD
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      DR=.017453292519943
      RD=57.295779513 0823
      USUN=1.32715445E+11
      AU=149598845.
```

```
      D=JD-2415020.
      CD=D/10000.
      TE=D/36525.
C
      AM=1.5236915*AU
      EM=0.09331290+0.000092064*TE-0.000000077*TE**2
      XIM=1.850334-0.000675*TE+0.000012*TE**2
      OM=48.786442+0.770991*TE-0.0000015*TE**2-0.00000576*TE**3
      WM=334.218203+1.840759*TE+0.000130*TE**2-0.00000129*TE**3-OM
      XMM=ANGLE(319.529425+0.5240207666*D+0.000013553*CD**2+0.000000025*
     1CD**3)
C
      CALL TINVS(XMM*DR,EM,ECM,FM)
      CALL CONCAR(AM,EM,XIM,WM,OM,FM*RD,XHM,YHM,ZHM,DXHM,DYHM,DZHM,USUN)
C
      RETURN
      END




      SUBROUTINE EULER(X,Y,Z,XP,YP,ZP,PHI,PSI,THETA,DPHI,DPSI,DTHETA,WXP
     1,WYP,WZP,J,K)
      XPHI=PHI*.0174532925
      XPSI=PSI*.0174532925
      XTH=THETA*.0174532925
      IF(J)10,12,11
   10 X=(COS(XPSI)*COS(XPHI)-COS(XTH)*SIN(XPHI)*SIN(XPSI))*XP+(-SIN(XPSI
     1)*COS(XPHI)-COS(XTH)*SIN(XPHI)*COS(XPSI))*YP+(SIN(XTH)*SIN(XPHI))*
     2ZP
      Y=(COS(XPSI)*SIN(XPHI)+COS(XTH)*COS(XPHI)*SIN(XPSI))*XP+(-SIN(XPSI
     1)*SIN(XPHI)+COS(XTH)*COS(XPHI)*COS(XPSI))*YP+(-SIN(XTH)*COS(XPHI))
     2*ZP
      Z=(SIN(XTH)*SIN(XPSI))*XP+(SIN(XTH)*COS(XPSI))*YP+(COS(XTH))*ZP
      GO TO 12
   11 XP=(COS(XPSI)*COS(XPHI)-COS(XTH)*SIN(XPHI)*SIN(XPSI))*X+(COS(XPSI)
     1*SIN(XPHI)+COS(XTH)*COS(XPHI)*SIN(XPSI))*Y+(SIN(XTH)*SIN(XPSI))*Z
      YP=(-SIN(XPSI)*COS(XPHI)-COS(XTH)*SIN(XPHI)*COS(XPSI))*X+(-SIN(XPS
     1I)*SIN(XPHI)+COS(XTH)*COS(XPHI)*COS(XPSI))*Y+(COS(XPSI)*SIN(XTH))*
     2Z
      ZP=(SIN(XTH)*SIN(XPHI))*X+(-SIN(XTH)*COS(XPHI))*Y+COS(XTH)*Z
   12 IF(K)13,15,14
   13 DPHI=(WXP*SIN(XPSI)+WYP*COS(XPSI))/SIN(XTH)
      DPSI=WZP-(COS(XTH)*(WXP*SIN(XPSI)+WYP*COS(XPSI)))/SIN(XTH)
      DTHETA=WXP*COS(XPSI)-WYP*SIN(XPSI)
      GO TO 15
   14 WXP=DPHI*SIN(XTH)*SIN(XPSI)+DTHETA*COS(XPSI)
      WYP=DPHI*SIN(XTH)*COS(XPSI)-DTHETA*SIN(XPSI)
      WZP=DPSI+DPHI*COS(XTH)
   15 RETURN
      END
```

```
      SUBROUTINE JULCAL(X,WDI,FDI,IND)
C
C     THIS SUBROUTINE CONVERTS A GIVEN JULIAN DATE OR THE NUMBER OF
C     WHOLE AND FRACTIONAL DAYS SINCE JANUARY 1, 1950, 0 HRS., TO THE
C     CORRESPONDING CALENDAR DATE.
C
C     WDI - INTEGRAL PART OF JULIAN DATE OR WHOLE NUMBER OF DAYS SINCE
C           JANUARY 1, 1950, 0 HRS.
C     FDI - FRACTIONAL PART OF JULIAN DATE OR FRACTIONAL NUMBER OF DAYS
C           SINCE JANUARY 1, 1950, 0 HRS.
C     IND - CONTROL INTEGER. 0 IMPLIES JULIAN DATE,1 INPLIES DAYS
C     X(1-6) - CALENDAR DATE (YEAR,MONTH,DAY,HOUR,MINUTE,SECOND )
C
      DIMENSION X(6),A(12),W(12)
      WD=WDI
      FD=FDI
      IF(IND)1,1,5
    1 IF(FD-.5)2,2,3
    2 FD=FD+.5
      WD=WD-1.
      GO TO 4
    3 FD=FD-.5
    4 WD=WD-2433282.
    5 WD=WD+1.
      DY=365.
      Z=2.
      N=0
      Q=4.
    6 WD=WD-DY
      IF(WD)10,10,7
    7 N=N+1
      Z=Z+1.
      CK=Q-Z
      IF(CK)9,9,8
    8 DY=365.
      FC=28.
      GO TO 6
    9 DY=366.
      Q=Q+4.
      FC=29.
      GO TO 6
   10 WD=WD+DY
      DO 11 I=1,12
   11 A(I)=0.
      C1=31.
      C2=30.
      DO 13 I=1,12
      A(I)=1.
      CA=FC*A(2)+C1*(A(1)+A(3)+A(5)+A(7)+A(8)+A(10)+A(12))+C2*(A(4)+
     1   A(6)+A(9)+A(11))
      W(I)=WD-CA
      IF(W(I))12,12,13
   12 IF(I-1)15,15,16
```

```
   15 MON=1
      GO TO 14
   16 MON=I-1
      WD=W(MON)
      MON=MON+1
      GO TO 14
   13 CONTINUE
   14 N=N+50
      X(1)=N
      X(2)=MON
      X(3)=WD
      FH=FD*24.
      N=FH
      X(4)=N
      FM=(FH-X(4))*60.
      N=FM
      X(5)=N
      X(6)=(FM-X(5))*60.
      RETURN
      END


      SUBROUTINE LAMBRT(X1,Y1,Z1,X2,Y2,Z2,TIME12,A,E,XI,W,O,TA1,TA2,U,KK
     1)
C
C         IF X2,Y2,AND Z2 ARE ZERO, THEN X1 IS CONSIDERED THE RADIAL
C         DISTANCE TO POINT 1, Y1 THE DISTANCE TO POINT 2, AND Z1 THE
C         ANGLE FROM POINT 1 TO POINT 2.  MOTION IS ALWAYS CONSIDERED
C         FROM POINT 1 TO POINT 2.
C
      REAL M12,N
      ATANH(X)=.5*ALOG((1.+X)/(1.-X))
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      DATA DR,RD,PI/.01745329251994,57.2957795,3.1415926535/
      M=0
      KK=1
      TA2=0.
      KEY=1
      IORBIT=1
      IF(TIME12.LE.0.) GO TO 800
      IF((ABS(X2).LT..1).AND.(ABS(Y2).LT..1).AND.(ABS(Z2).LT..1))GO TO 1
      R1=SQRT(X1*X1+Y1*Y1+Z1*Z1)
      R2=SQRT(X2*X2+Y2*Y2+Z2*Z2)
      CPSI=(X1*X2+Y1*Y2+Z1*Z2)/R1/R2
      SPSI=(X1*Y2-X2*Y1)/ABS(X1*Y2-X2*Y1)*SQRT(1.-CPSI*CPSI)
      PSI=ANGLE(ATAN2(SPSI,CPSI)*RD)
      GO TO 2
    1 R1=X1
      R2=Y1
      PSI=ANGLE(Z1)
      XI=0.
      O=0.
      W=0.
    2 C=SQRT(R1*R1+R2*R2-2.*R1*R2*COS(PSI*DR))
```

```
      IF(PSI.LT.0.01) GO TO 800
      AM=(R1+R2+C)/4.
      S=2.*AM
      TP=SQRT(2./U)*(S**1.5-(S-C)**1.5)/3.
      TTP=SQRT(2./U)*(S**1.5+(S-C)**1.5)/3.
      IF((PSI.LE.180.).AND.(TIME12.LT.TP)) IORBIT=2
      IF((PSI.GE.180.).AND.(TIME12.LT.TTP)) IORBIT=2
    3 CTA2=COS(TA2*DR)
      CTA1=COS((TA2-PSI)*DR)
      Q=R2*CTA2-R1*CTA1
      IF(ABS(Q).GT.1.) GO TO 5
    4 IF(KEY.GT.1)GO TO 25
      TA2=TA2+5.
      GO TO 3
    5 E=(R1-R2)/Q
      IF(E.LT.0.) GO TO 4
      A=R2*(1.+E*CTA2)/(1.-E*E)
      GO TO (6,7),IORBIT
    6 IF(E.GT.1..OR.A.LT.0.) GO TO 4
      TEMP=SQRT((1.-E)/(1.+E))
      EC1=ANGLE(2.*ATAN(TEMP*TAN((TA2-PSI)*DR/2.))*RD)*DR
      EC2=ANGLE(2.*ATAN(TEMP*TAN(TA2*DR/2.))*RD)*DR
      DELEC=EC2-EC1
      IF(DELEC.LT.0.) DELEC=2.*PI+DELEC
      M12=DELEC-E*(SIN(EC2)-SIN(EC1))
      GO TO 8
    7 IF(E.LT.1..OR.A.GT.0.) GO TO 4
      TEMP=SQRT((E-1.)/(E+1.))
      EC1=2.*ATANH(TEMP*TAN((TA2-PSI)*DR/2.))
      EC2=2.*ATANH(TEMP*TAN((TA2*DR/2.)))
      M12=E*(SINH(EC2)-SINH(EC1))-EC2+EC1
    8 N=SQRT(U/ABS(A)**3)
      F=TIME12-M12/N
      GO TO (9,10,11),KEY
    9 KEY=2
      TALAST=TA2
      TA2=TA2+1.
      GO TO 13
   10 KEY=3
      GO TO 12
   11 M=M+1
      IF(M.GT.60)GO TO 800
      IF(ABS(F).LE.ABS(FLAST)) GO TO 12
   25 DFDTA2=DFDTA2*2.
      M=M+1
      IF(M.GT.60)GO TO 800
      TA2=TALAST-FLAST/DFDTA2
      GO TO 3
   12 ERROR=F/TIME12
      IF(ABS(ERROR).LT..00001) GO TO 14
      DFDTA2=(F-FLAST)/(TA2-TALAST)
      TALAST=TA2
      TA2=TA2-F/DFDTA2
```

```
   13 FLAST=F
      GO TO 3
C
   14 TA1=TA2-PSI
      IF((ABS(X2).LT..1).AND.(ABS(Y2).LT..1).A.(ABS(Z2).LT..1))GO TO 900
      D1=Y1*Z2-Z1*Y2
      D2=Z1*X2-X1*Z2
      D3=X1*Y2-Y1*X2
      HH=SQRT(D1*D1+D2*D2+D3*D3)
      IF(D3.GT.0.) GO TO 15
      D1=-D1
      D2=-D2
      D3=-D3
   15 COSXI=D3/HH
      XI=ATAN2(SQRT(1.-COSXI*COSXI),COSXI)*RD
      SO=D1/(HH*SIN(XI*DR))
      CO=-D2/(HH*SIN(XI*DR))
      IF(SO.EQ.0..AND.CO.EQ.0.) CO=1.
      O=ANGLE(ATAN2(SO,CO)*RD)
      W=ANGLE(ATAN2((-X1*SO+Y1*CO)*COSXI+Z1*SIN(XI*DR),X1*CO+Y1*SO)*RD-T
     1A1)
      GO TO 900
  800 KK=0
  900 RETURN
      END


      SUBROUTINE LATLNG(X,Y,Z,XLAT,XLONG)
C
C     THIS SUBROUTINE COMPUTES THE LATITUDE AND LONGITUDE OF A GIVEN
C     POSITION VECTOR
C
C     X,Y,Z - COMPONENTS OF THE POSITION VECTOR
C     XLAT,XLONG - LATITUDE AND LONGITUDE
C
      ARCOS(X)=ACOS(X)
      ARSIN(X)=ASIN(X)
      RD=57.2957795
      R=SQRT(X**2+Y**2+Z**2)
      XLONG=ATAN2(Y,X)*RD
      XLAT=ARSIN(Z/R)*RD
      RETURN
      END
```

```
      SUBROUTINE  OCCULT(A,E,XI,W,O,U,RS,EX,EY,EZ,TOCC,T1,ALT1,F1,DEC1,R
     1A1,T2,ALT2,F2,DEC2,RA2,KK)
C
C     THIS SUBROUTINE COMPUTES THE ENTRANCE AND EXIT TRUE ANOMALIES OF
C     OCCULTATION. THIS ROUTINE CALLS SUBROUTINES RXYZPQW, QARTIC,
C     CROSS, DOT, TOCNIC, RPQWXYZ, AND LATLNG.
C
C     A,E,XI - SEMIMAJOR AXIS, ECCENTRICITY, INCLINATION
C     W,O - ARGUMENT OF PERIAPSIS, LONGITUDE OF ASCENDING NODE
C     U,RS - GRAVITATIONAL CONSTANT AND RADIUS OF THE PLANET
C     EX,EY,EZ - COMPONENTS OF UNIT VECTOR TOWARD THE BODY OCCULTED
C     TOCC - LENGTH OF TIME IN SHADOW
C     T1,ALT1,F1,DEC1,RA1 - CONDITIONS AT ENTRY INTO THE SHADOW, TIME
C             FROM PERIAPSIS, ALTITUDE, TRUE ANOMALY, DECLINATION, AND
C             RIGHT ASCENSION
C     T2,ALT2,F2,DEC2,RA2 - CONDITIONS AT EXIT FROM THE SHADOW
C     KK - CONTROL INTEGER. 0 IMPLIES NO OCCULTATION, 1 IMPLIES OCCULT.
C
      DIMENSION RPQW(3,3),CF(4),RXYZ(3,3)
      ANGLE(X)=AMOD(X,360.)+180.-SIGN(180.,X)
      DR=.017453292519943
      RD=57.2957795130823
      F1=2000.
      F2=2000.
      KK=0
      P=A*(1-E*E)
      CALL RXYZPQW(EX,EY,EZ,XI,W,O,RPQW,BETA,XXI,ZBODY)
      C1=(RS/P)**4*E**4-2.*(RS/P)**2*(XXI**2-BETA**2)*E**2+(BETA**2+XXI*
     1*2)**2
      C2=4.*(RS/P)**4*E**3-4.*(RS/P)**2*(XXI**2-BETA**2)*E
      C3=6.*(RS/P)**4*E**2-2.*(RS/P)**2*(XXI**2-BETA**2)-2.*(RS/P)**2*(1
     1.-XXI**2)*E**2+2.*(XXI**2-BETA**2)*(1.-XXI**2)-4.*BETA**2*XXI**2
      C4=4.*(RS/P)**4*E-4.*(RS/P)**2*(1.-XXI**2)*E
      C5=(RS/P)**4-2.*(RS/P)**2*(1.-XXI**2)+(1.-XXI**2)**2
      CALL QARTIC(C1,C2,C3,C4,C5,CF(1),CF(2),CF(3),CF(4),JJ)
      IF(JJ.EQ.0) GO TO 800
C
      DO 3 I=1,JJ
      IF(ABS(CF(I)).LT.1.0001.AND.ABS(CF(I)).GT.1.) CF(I)=0.999999
      IF(ABS(CF(I)).GT.1.) GO TO 3
      SF=SQRT(1.-CF(I)**2)
      R=P/(1.+E*CF(I))
      CALL CROSS(R*CF(I),R*SF,0.,BETA,XXI,ZBODY,PX,PY,PZ,PRODUCT)
      CALL DOT(CF(I),SF,0.,BETA,XXI,ZBODY,ANG)
      IF(ABS(PRODUCT-RS).LT..01.AND.ANG.GT.90.) GO TO 1
      SF=-SF
      CALL CROSS(R*CF(I),R*SF,0.,BETA,XXI,ZBODY,PX,PY,PZ,PRODUCT)
      CALL DOT(CF(I),SF,0.,BETA,XXI,ZBODY,ANG)
      IF(ABS(PRODUCT-RS).LT..01.AND.ANG.GT.90.) GO TO 1
      GO TO 3
    1 IF(F1.LT.1000.)GO TO 2
      F1=ANGLE(ATAN2(SF,CF(I))*RD)
      GO TO 3
    2 F2=ANGLE(ATAN2(SF,CF(I))*RD)
```

```
      GO TO 4
    3 CONTINUE
C
    4 IF(F2.GT.1000.)GO TO 800
      CF1=COS(F1*DR)
      SF1=SIN(F1*DR)
      DSDF=2.*RS*RS*(1.+E*CF1)*(-E*SF1)+2.*P*P*(BETA*CF1+XXI*SF1)*(-BETA
     1*SF1+XXI*CF1)
      IF(DSDF.GT.0.)GO TO 5
C
      TEMP=F1
      F1=F2
      F2=TEMP
C
    5 CALL TCONIC(U,E,A,F1,T1)
      CALL TCONIC(U,E,A,F2,T2)
      TOCC=T2-T1
      IF(TOCC.LT.0.)TOCC=6.2831853*SQRT(A**3/U)+TOCC
      T1=T1/60.
      T2=T2/60.
      TOCC=TOCC/60.
      ALT1=P/(1.+E*COS(F1*DR))-RS
      ALT2=P/(1.+E*COS(F2*DR))-RS
      CALL RPQWXYZ(COS(F1*DR),SIN(F1*DR),0.,XI,W,O,RXYZ,RX,RY,RZ)
      CALL LATLNG(RX,RY,RZ,DEC1,RA1)
      CALL RPQWXYZ(COS(F2*DR),SIN(F2*DR),0.,XI,W,O,RXYZ,RX,RY,RZ)
      CALL LATLNG(RX,RY,RZ,DEC2,RA2)
      KK=1
      GO TO 900
C
  800 CONTINUE
      TOCC=0.
      T1=0.
      ALT1=0.
      F1=0.
      DEC1=0.
      RA1=0.
      T2=0.
      ALT2=0.
      F2=0.
      DEC2=0.
      RA2=0.
  900 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PARIN(XP,YP,X,Y,II)
      DIMENSION X(3),Y(3),DET(3,3)
      DO 10 I=1,3
      DET(I,1)=X(I)**2
      DET(I,2)=X(I)
      DET(I,3)=1.0
   10 CONTINUE
      CALL DETER(D,DET)
      IF(ABS(D)-1.0E-16)100,100,20
   20 DO 30 I=1,3
      DET(I,1)=Y(I)
   30 CONTINUE
      CALL DETER(A,DET)
      A=A/D
      DO 40 I=1,3
      DET(I,1)=X(I)**2
      DET(I,2)=Y(I)
   40 CONTINUE
      CALL DETER(B,DET)
      B=B/D
      DO 50 I=1,3
      DET(I,2)=X(I)
      DET(I,3)=Y(I)
   50 CONTINUE
      CALL DETER(C,DET)
      C=C/D
      IF(II)200,150,200
  150 XP=-B/(2.0*A)
      YP=C-B**2/(4.0*A)
      GO TO 300
  200 YP=(A*XP+B)*XP+C
      GO TO 300
  100 XP=0.0
      YP=0.0
  300 RETURN
      END
```

```
      SUBROUTINE PRECES(JD1,XE1,YE1,ZE1,JD2,XE2,YE2,ZE2)
C
C     THIS SUBROUTINE TRANSFORMS GEOCENTRIC EARTH EQUATORIAL COORDINATES
C     FROM EPOCH JD1 TO EPOCH JD2. THIS ROUTINE CALLS SUBROUTINE EULER.
C
C     JD1,JD2 - JULIAN DATES OF INITIAL AND FINAL EPOCH
C     XE1,YE1,ZE1 - COMPONENTS OF VECTOR IN JD1 COORDINATE SYSTEM
C     XE2,YE2,ZE2 - COMPONENTS OF VECTOR IN JD2 COORDINATE SYSTEM
C
      REAL JD1,JD2
      T=ABS((JD2-JD1)/36524.219879)
      TO=(JD2-2415020.)/36524.219879
      ZETAO=(0.64006944+0.38777778E-3*TO)*T+0.83888889E-4*T**2+0.5E-5*T*
     1*3
```

```
      CZETAO=ZETAO+0.21972222E-3*T**2
      THETAO=(0.55685611-0.2369444E-3*T0)*T-0.11833333E-3*T**2-0.1166666
     17E-4*T**3
C
      IF(JD2-JD1.GT.0.) GO TO 1
      TEMP=ZETAO
      ZETAO=-CZETAO
      CZETAO=-TEMP
      THETAO=-THETAO
C
    1 CALL EULER(XE1,YE1,ZE1,XE2,YE2,ZE2,90.-ZETAO,-(90.+CZETAO),THETAO,
     1DPHI,DPSI,DPSI,WXP,WYP,WZP,1,0)
C
      RETURN
      END



      SUBROUTINE QADRAT(A,B,C,X1,X2,KK)
C              SOLVES EQUATION A*X**2+B*X+C=0
C              KK = NUMBER OF REAL ROOTS
      KK=0
      DIS=B*B-4.*A*C
      IF(DIS.LT.0.) GO TO 800
      X1=(-B+SQRT(DIS))/2./A
      X2=(-B-SQRT(DIS))/2./A
      KK=2
  800 RETURN
      END



      SUBROUTINE QARTIC(A,B,C,D,E,X1,X2,X3,X4,KK)
C
C     THIS SUBROUTINE SOLVES THE EQUATION AX**4 +BX**3 +CX**2 +DX +E = 0
C     FOR THE REAL ROOTS. THIS ROUTINE CALLS SUBROUTINES QADRAT AND CONIC
C
C     A,B,C,D,E - COEFFICIENTS OF THE DIFFERENT POWERS OF X
C     X1,X2,X3,X4 - REAL ROOTS OF THE EQUATION
C     KK - NUMBER OF REAL ROOTS
C
      KK=0
      BP=B/A
      CP=C/A
      DP=D/A
      EP=E/A
C
      H=-BP/4.
      H2=H**2
      H3=H2*H
      H4=H3*H
      P=6.*H2+3.*BP*H+CP
      Q=4.*H3+3.*BP*H2+2.*CP*H+DP
      R=H4+BP*H3+CP*H2+DP*H+EP
```

```
C
      CALL CUBIC(1.,2.*P,P*P-4.*R,-Q*Q,T1,T2,T3,NROOT)
C
      GO TO (1,2,3),NROOT
    1 RP=T1
      GO TO 4
    2 RP=AMAX1(T1,T2)
      GO TO 4
    3 RP=AMAX1(T1,T2,T3)
C
    4 CONTINUE
      SQRP=SQRT(RP)
      XI=(P+RP-Q/SQRP)/2.
      BETA=(P+RP+Q/SQRP)/2.
C
      CALL QADRAT(1.,SQRP,XI,Y1,Y2,IROOT)
      CALL QADRAT(1.,-SQRP,BETA,Y3,Y4,JROOT)
      IF(IROOT+JROOT.EQ.0) GO TO 800
      IF(IROOT+JROOT.EQ.4) GO TO 6
      IF(IROOT.EQ.0) GO TO 5
      X1=Y1+H
      X2=Y2+H
      KK=2
      GO TO 800
    5 X1=Y3+H
      X2=Y4+H
      KK=2
      GO TO 800
    6 X1=Y1+H
      X2=Y2+H
      X3=Y3+H
      X4=Y4+H
      KK=4
  800 CONTINUE
      RETURN
      END


      SUBROUTINE RECEQ(JD,XEC,YEC,ZEC,XEQ,YEQ,ZEQ)
C
C     THIS SUBROUTINE ROTATES A VECTOR FROM GEOCENTRIC, ECLIPTIC, TO
C     THE GEOCENTRIC, EARTH EQUATORIAL COORDINATE SYSTEM
C
C     JD - JULIAN DATE
C     XEC,YEC,ZEC - COMPONENTS OF THE VECTOR IN THE GEOCENTRIC, ECLIPTIC
C                   COORDINATE SYSTEM
C     XEQ,YEQ,ZEQ - COMPONENTS OF THE VECTOR IN THE GEOCENTRIC, EARTH
C                   EQUATORIAL, COORDINATE SYSTEM
C
      REAL JD
      DR=.017453292519943
      TE=(JD-2415020.)/36525.
      XIE=23.452294-0.0130125*TE-0.00000164*TE**2+0.000000503*TE**3
      C=COS(XIE*DR)
      S=SIN(XIE*DR)
```

```
      XEQ=XEC
      YEQ=YEC*C-ZEC*S
      ZEQ=YEC*S+ZEC*C
      RETURN
      END



      SUBROUTINE REQMEQ(JD,XEQ,YEQ,ZEQ,XMEQ,YMEQ,ZMEQ,DECMEQ,RAMEQ)
C
C     THIS SUBROUTINE ROTATES A VECTOR FROM THE MEAN EARTH EQUATOR-
C     EQUINOX TO THE MEAN MARS EQUATOR-EQUINOX COORDINATE SYSTEM. THIS
C     ROUTINE CALLS SUBROUTINES REQPEQ AND LATLNG.
C
C     JD - JULIAN DATE AT TIME OF INTEREST
C     XEQ,YEQ,ZEQ - VECTOR IN THE EARTH EQUATORIAL SYSTEM
C     XMEQ,YMEQ,ZMEQ - VECTOR IN THE MARS EQUATORIAL SYSTEM
C     DECMEQ,RAMEQ - DECLINATION AND RIGHT ASCENSION OF THE VECTOR IN
C                    THE MARS EQUATORIAL SYSTEM
C
      REAL JD
C
      TE=(JD-2415020.)/36525.
      TAU=AMOD(TE*100.,1.)
      TP=TE*100.-TAU-50.
C
      ALPHAO=316.55+45.*.006751+.006751*TP-.001013*TAU
      GAMMAO=52.85+45*.00348+.00348*TP-.000631*TAU
      OMEGA=48.78644167+0.77099167*TE-0.1388889E-5*TE**2
      XI=1.850333333-0.675E-3*TE+0.1261111E-4*TE**2
C
      CALL REQPEQ(JD,ALPHAO,GAMMAO,OMEGA,XI,XEQ,YEQ,ZEQ,XMEQ,YMEQ,ZMEQ)
      CALL LATLNG(XMEQ,YMEQ,ZMEQ,DECMEQ,RAMEQ)
C
      RETURN
      END
```

```
      SUBROUTINE RECPEQ(JD,ALPHAO,GAMMAO,OMEGA,XI,XEQ,YEQ,ZEQ,XPEQ,YPEQ,
     1ZPEQ)
C
C     THIS SUBROUTINE ROTATES A VECTOR FROM MEAN EARTH EQUATOR-EQUINOX
C     TO PLANET EQUATOR-EQUINOX COORDINATE SYSTEM. THIS ROUTINE CALLS
C     SUBROUTINE EULER.
C
C     JD - JULIAN DATE AT TIME OF INTEREST
C     ALPHAO,GAMMAO - RIGHT ASCENSION AND DECLINATION OF THE PLANETS
C              AXIS OF ROTATION EXPRESSED IN THE EARTH EQUATORIAL
C              COORDINATE SYSTEM
C     OMEGA,XI - LONGITUDE OF THE ASCENDING NODE AND INCLINATION OF THE
C                PLANETS ORBITAL PLANE REFERENCED TO THE ECLIPTIC AND
C                VERNAL EQUINOX
C     XEQ,YEQ,ZEQ - COMPONENTS OF THE VECTOR IN THE EARTH EQUATORIAL
C                   COORDINATE SYSTEM
C     XPEQ,YPEQ,ZPEQ - COMPONENTS OF THE VECTOR IN THE PLANET EQUATORIAL
C                      COORDINATE SYSTEM
      REAL JD
      DIMENSION RPQW(3,3)
      DR=0.017453292519943
      RD=57.2957795130823
      TE=(JD-2415020.)/36525.
      E=23.45229444-0.130125E-1*TE-0.16388889E-5*TE**2+0.50277778E-6*TE*
     1*3
C
      CE=COS(E*DR)
      SE=SIN(E*DR)
      CAL=COS(ALPHAO*DR)
      SAL=SIN(ALPHAO*DR)
      CGM=COS(GAMMAO*DR)
      SGM=SIN(GAMMAO*DR)
      COM=COS(OMEGA*DR)
      SOM=SIN(OMEGA*DR)
C
      CZP=CE*SOM*CAL-COM*SAL
      SZP=SQRT(1.-CZP*CZP)
      ZP=ATAN2(SZP,CZP)*RD
      SXP=SE*CAL/SZP
      CXP=(-CE*COM*CAL-SOM*SAL)/SZP
      XP=ATAN2(SXP,CXP)*RD
      SYP=SE*SOM/SZP
      CYP=(CE*SOM*SAL+COM*CAL)/SZP
      YP=ATAN2(SYP,CYP)*RD
      CI=COS((XP-XI)*DR)*SIN((YP-GAMMAO)*DR)+SIN((XP-XI)*DR)*COS((YP-GAM
     1MAO)*DR)*CZP
      SI=SQRT(1.-CI*CI)
      SWP=SZP*SIN((XP-XI)*DR)/SI
      CWP=(-COS((XP-XI)*DR)*COS((YP-GAMMAO)*DR)+SIN((XP-XI)*DR)*SIN((YP-
     1GAMMAO)*DR)*CZP)/SI
      WP=ATAN2(SWP,CWP)*RD
      CALL EULER(XEQ,YEQ,ZEQ,XPEQ,YPEQ,ZPEQ,90.+ALPHAO,WP+180.,90.-GAMMA
     1O,0.,0.,0.,0.,0.,0.,1,0)
      RETURN
      END
```

```
      SUBROUTINE RPQWXYZ(VP,VQ,VW,XI,W,O,RXYZ,VX,VY,VZ)
C
C     THIS SUBROUTINE ROTATES A VECTOR FROM THE PQW TO THE XYZ
C     COORDINATE SYSTEM
C
C     VP,VQ,VW - COMPONENTS OF THE VECTOR IN THE PQW SYSTEM
C     XI,W,O - INCLINATION, ARGUMENT OF PERIAPSIS, LONGITUDE OF ASCENDING
C            NODE
C     RXYZ - ROTATIONAL MATRIX FROM THE PQW TO THE XYZ COORDINATE SYSTEM
C     VX,VY,VZ - COMPONENTS OF THE VECTOR IN THE XYZ SYSTEM
C
      DIMENSION RXYZ(3,3)
      DR=.017453292519943
C
      CW=COS(W*DR)
      SW=SIN(W*DR)
      CO=COS(O*DR)
      SO=SIN(O*DR)
      CXI=COS(XI*DR)
      SXI=SIN(XI*DR)
C
      RXYZ(1,1)=CW*CO-SW*SO*CXI
      RXYZ(1,2)=-SW*CO-CW*SO*CXI
      RXYZ(1,3)=SO*SXI
      RXYZ(2,1)=CW*SO+SW*CO*CXI
      RXYZ(2,2)=-SW*SO+CW*CO*CXI
      RXYZ(2,3)=-CO*SXI
      RXYZ(3,1)=SW*SXI
      RXYZ(3,2)=CW*SXI
      RXYZ(3,3)=CXI
C
      VX=RXYZ(1,1)*VP+RXYZ(1,2)*VQ+RXYZ(1,3)*VW
      VY=RXYZ(2,1)*VP+RXYZ(2,2)*VQ+RXYZ(2,3)*VW
      VZ=RXYZ(3,1)*VP+RXYZ(3,2)*VQ+RXYZ(3,3)*VW
C
      RETURN
      END


      SUBROUTINE RXYZPQW(VX,VY,VZ,XI,W,O,RPQW,VP,VQ,VW)
C
C     ROTATES A VECTOR FROM THE  XYZ  TO THE PQW COORDINATE SYSTEM
C
      DIMENSION RPQW(3,3)
      DR=.017453292519943
C
      CW=COS(W*DR)
      SW=SIN(W*DR)
      CO=COS(O*DR)
      SO=SIN(O*DR)
      CXI=COS(XI*DR)
      SXI=SIN(XI*DR)
C
```

```
      RPQW(1,1)=CW*CO-SW*SO*CXI
      RPQW(1,2)=CW*SO+SW*CO*CXI
      RPQW(1,3)=SW*SXI
      RPQW(2,1)=-SW*CO-CW*SO*CXI
      RPQW(2,2)=-SW*SO+CW*CO*CXI
      RPQW(2,3)=CW*SXI
      RPQW(3,1)=SO*SXI
      RPQW(3,2)=-CO*SXI
      RPQW(3,3)=CXI
C
      VP=RPQW(1,1)*VX+RPQW(1,2)*VY+RPQW(1,3)*VZ
      VQ=RPQW(2,1)*VX+RPQW(2,2)*VY+RPQW(2,3)*VZ
      VW=RPQW(3,1)*VX+RPQW(3,2)*VY+RPQW(3,3)*VZ
C
      RETURN
      END


      SUBROUTINE TCONIC(U,EC,A,TA,T)
      DATA DR/.17453292519943E-1/
      TA2=TA*DR
      SLR=A*(1.-EC*EC)
      AB=ABS(A)
      FAC=AB*SQRT(AB/U)
      ECA=(1.-EC)/(1.+EC)
      ABE=SQRT(ABS(ECA))
      THE=TAN(.5*TA2)
      IF(ABE-.5E-10)11,11,12
   12 CONTINUE
      ECA=2.*ATAN(ABE*THE)
      IF(A)14,11,13
   13 T=FAC*(ECA-EC*SIN(ECA))
      GO TO 16
   14 ANG=ABE*THE
      ANG=1.+2.*ANG/(1.-ANG)
      T=FAC*(EC*TAN(ECA)-ALOG(ANG))
      GO TO 16
   11 FAC=SQRT(SLR**3/U)*2./((1.+EC)**2)
      EC1=ECA*THE**2
      T=FAC*(THE+THE**3*((1.-2.*ECA)/3.-(2.-3.*ECA)*EC1/5.+(3.-4.*ECA)*E
     1C1**2/7.-(4.-5.*ECA)*EC1**3/9.))
   16 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  TINVS(M,E,EC,F)
      REAL M,MO
      DATA PI/3.141592653589793/
      ASINH(X)=SIGN(ALOG(ABS(X)+SQRT(X**2+1.)),X)
      IF(E.GE.1.)GO TO 100
      EC=M
   10 MO=EC-E*SIN(EC)
      DM=M-MO
      DE=DM/(1.-E*COS(EC))
      EC=EC+DE
      IF(ABS(DE).GT.1.E-12  )GO TO 10
      HEC= EC/2.
      HF=ATAN(SQRT((1.+E)/(1.-E))*SIN(HEC)/COS(HEC))
      IF(HF.LT.0.)HF=HF+PI
      F=2.*HF
      GO TO 800
  100 CONTINUE
      EC=ASINH(M/E)
  101 MO=E*SINH(EC)-EC
      DM=M-MO
      DE=DM/(E*COSH(EC)-1.)
      EC=EC+DE
      IF(ABS(DE).GT.1.E-12  )GO TO 101
      F=2.*ATAN(SQRT((E+1.0)/(E-1.0))*TANH(EC/2.0))
  800 RETURN
      END
```

```
      SUBROUTINE VECTOR(JD,DECS,RAS,DECE,RAE,DECC,RAC,SX,SY,SZ,EX,EY,EZ,
     1CX,CY,CZ,IBODY)
C
C     THIS SUBROUTINE COMPUTES THE POSITION OF THE SUN, EARTH, AND
C     CANOPUS IN PLANET EQUATOR, MEAN PLANET EQUINOX OF DATE, AND WRITES
C     DATA. THIS ROUTINE CALLS SUBROUTINES EEARTH, EMARS, EVENUS, PRECES,
C     LATLNG, DOT, RECEQ, REQVEQ, AND REQMEQ.
C
C     JD - JULIAN DATE AT TIME OF INTEREST
C     IBODY - CONTROL INTEGER. 2 IMPLIES VENUS, 4 IMPLIES MARS.
C     DECS,RAS - DECLINATION AND RIGHT ASCENSION OF THE SUN.
C     DECE,RAE - DECLINATION AND RIGHT ASCENSION OF THE EARTH.
C     DECC,RAC - DECLINATION AND RIGHT ASCENSION OF CANOPUS.
C     SX,SY,SZ - UNIT VECTOR FROM THE PLANET TO THE SUN.
C     EX,EY,EZ - UNIT VECTOR FROM THE PLANET TO THE EARTH.
C     CX,CY,CZ - UNIT VECTOR FROM THE PLANET TO CANOPUS.
C
      REAL JD
      RD=57.2957795130823
C
      CALL EEARTH(JD,XHE,YHE,ZHE,DXHE,DYHE,DZHE)
      CALL PRECES(2433282.,-.060340592,.60342839,-.79513092,JD,CXE,CYE,C
     1ZE)
```

```
C
      IF(IBODY.EQ.4) GO TC 2
    2 CALL EMARS(JD,XHP,YHP,ZHP,DXHP,DYHP,DZHP)
C
    3 XHPE=XHE-XHP
      YHPE=YHE-YHP
      ZHPE=ZHE-ZHP
      RSE=SQRT(XHE**2+YHE**2+ZHE**2)
      RSP=SQRT(XHP**2+YHP**2+ZHP**2)
      RPE=SQRT(XHPE**2+YHPE**2+ZHPE**2)
      SEX=XHE/RSE
      SEY=YHE/RSE
      SEZ=ZHE/RSE
      SPX=XHP/RSP
      SPY=YHP/RSP
      SPZ=ZHP/RSP
      PEX=XHPE/RPE
      PEY=YHPE/RPE
      PEZ=ZHPE/RPE
      CALL LATLNG(SEX,SEY,SEZ,EHLAT,EHLONG)
      CALL LATLNG(SPX,SPY,SPZ,PHLAT,PHLONG)
      CALL DCT(SEX,SEY,SEZ,SPX,SPY,SPZ,ESP)
      CALL DCT(SEX,SEY,SEZ,PEX,PEY,PEZ,SEP)
      CALL DCT(SPX,SPY,SPZ,-PEX,-PEY,-PEZ,SPE)
      CALL RECEQ(JD,-SPX,-SPY,-SPZ,SXE,SYE,SZE)
      CALL RECEQ(JD,PEX,PEY,PEZ,EXE,EYE,EZE)
C
      IF(IBODY.EQ.4) GO TC 5
    5 CALL RECMEQ(JD,SXE,SYE,SZE,SX,SY,SZ,DECS,RAS)
      CALL RECMEQ(JD,EXE,EYE,EZE,EX,EY,EZ,DECE,RAE)
      CALL RECMEQ(JD,CXE,CYE,CZE,CX,CY,CZ,DECC,RAC)
C
    6 XPS=SX*RSP
      YPS=SY*RSP
      ZPS=SZ*RSP
      XPE=EX*RPE
      YPE=EY*RPE
      ZPE=EZ*RPE
C
  800 RETURN
      END
```

REFERENCES

1. Escobal, Pedro Ramon: Methods of Orbit Determination. John Wiley & Sons, Inc., 1965.
2. Green, Richard N.: Investigation of Occultation and Imagery Problems for Orbital Missions to Venus and Mars. NASA TN D-5104, 1969.

NASA-Langley, 1971 —— 30 L-7213

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

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AN
SPACE ADMINISTRATION

03U 001 55 51 3DS 71012 00903
AIR FORCE WEAPONS LABORATORY /WLOL/
KIRTLAND AFB, NEW MEXICO 87117

ATT E. LOU BOWMAN, CHIEF,TECH. LIBRARY

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

---

"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 Technology Surveys.

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

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

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