NASA CONTRACTOR REPORT

NASA CR-1430

RAPID OPTIMIZATION OF MULTIPLE-BURN ROCKET FLIGHTS

by K. R. Brown, E. F. Harrold, and G. W. Johnson

Prepared by
INTERNATIONAL BUSINESS MACHINES CORPORATION
Cambridge, Mass.
for George C. Marshall Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1969

1. Report No. NASA CR-1430	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle RAPID OPTIMIZATION OF MULTIPLE-BURN ROCKET FLIGHTS		5. Report Date September 1969
		6. Performing Organization Code
7. Author(s) K. R. Brown, E. F. Harrold, and G. W. Johnson		8. Performing Organization Report No. M150
9. Performing Organization Name and Address IBM - Federal Systems Division Cambridge Advanced Space Systems Cambridge, Massachusetts 02138		10. Work Unit No. 125-17-05-33-62
		11. Contract or Grant No. NAS8-21315
12. Sponsoring Agency Name and Address NASA-Marshall Space Flight Center Aero-Astrodynamics Laboratory Marshall Space Flight Center, Ala. 35812		13. Type of Report and Period Covered March 8, 1968 to March 8, 1969 ~~Contractor~~ Report
		14. Sponsoring Agency Code

15. Supplementary Notes

Additional work is being done under a renewal of this contract.

16. Abstract

An indirect (shooting) method has been developed for optimization of multiple-burn rocket flights. The method is fast enough to warrant consideration as a real-time guidance scheme, yet it is free of special-purpose approximations or artificial mission restrictions. This report describes the algorithm and the analysis on which it was based, and gives test results indicative of the power of the method.

17. Key Words Trajectory optimization, optimal guidance numerical analysis	18. Distribution Statement Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price *
Unclassified	Unclassified	53	$3.00

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

TABLE OF CONTENTS

1. Introduction:

Optimal trajectory problems usually involve the determination of a vehicle acceleration history that will accomplish a required change in vehicle state with a minimum expenditure of fuel. Functional optimization problems of this type can be reduced to boundary value problems in ordinary differential equations by application of the well-known necessary conditions of optimality in the form of the classical Calculus of Variations, or one of its more recent counterparts, dynamic programming(1) and the maximum principle(2). However, except in rare instances, the principal parts of the solutions to these boundary value problems are not available in closed form. State-of-the-art guidance schemes(3),(4),(5) circumvent this difficulty by considering approximate formulations of the original problem that allow analytic construction of major parts of the solution, so that only a simple iterative process is required. Since these approaches avoid some of the time-consuming numerical integration procedures that would be required to compute a general solution to the fundamental problem, the speed needed for real-time application has been the primary motivation for semi-explicit methods of this type. As a result of the approximations, the accuracy and flexibility of current flight schemes are limited, primarily in that they are nearly optimal only for short arcs of powered flight and for specialized mission (boundary-value) conditions in a restricted coordinate system. This limitation can be relaxed somewhat in practice by use of special purpose adjustments, but only at the expense of additional preflight analysis and simulation.

---

(1) DREYFUS, S.E. Dynamic Programming and the Calculus of Variations, Academic Press, Inc., New York and London, 1965.

(2) PONTRIAGIN, L.S. et al. The Mathematical Theory of Optimal Processes, Interscience Publishers, John Wiley and Sons, Inc., New York and London, 1962.

(3) SMITH, I.E. "A Three Dimensional Ascending Iterative Guidance Mode," NASA-MSFC Report MTP-AERO-63-49, June, 1963.

(4) MacPHERSON, D. "An Explicit Solution to the Powered Flight Dynamics of a Rocket Vehicle," Aerospace Corporation, Report TDR-169(3126)TN-2, October, 1962.

(5) CHERRY, G. W. "A General Explicit Optimizing Guidance Law for Rocket-Propelled Spaceflight," AIAA Paper 64-638, August, 1964.

General iterative procedures for solving two-point boundary value problems may be classified under two main headings: "direct" methods and "indirect" methods. Roughly speaking, direct methods search over the space of functions satisfying the boundary value requirements for a function satisfying the differential equations, whereas indirect methods search over the space of functions satisfying the differential equations for a function satisfying the boundary-value requirements.

Prior to 1965, general (flexible) numerical procedures for computing precise optimal trajectories were far too unreliable in convergence and costly in computation requirements to be considered for real-time guidance. However, an indirect method for computing optimal trajectories, OPGUID, was developed in 1965(6),(7) incorporating improved techniques to obtain a substantial gain in speed, convergence, and flexibility. The principal improvements were the use of an efficient integration algorithm that was tailored to special features of the initial-value problem, and the use of closed form representations of the final-value transversality conditions for general orbital injection missions of interest.

Specifically, the OPGUID algorithm developed in 1965 required less than one-half second per iteration of the boundary-value problem (on an IBM 7094) as compared to a minute or more per iteration required by other existing algorithms that were applicable to launch vehicle trajectory problems at that time. A simple scaling rule for the amount of the Newtonian correction that was permitted per iteration resulted in an extremely large region of convergence that was surprisingly insensitive to accurate initialization (e.g. a complete 180^o reversal in the initial thrust direction could be corrected for in ten to twenty iterations for typical Saturn and Apollo launch trajectories).

The indirect approach is particularly well suited for real-time use, where the guidance scheme must continually adjust to perturbations in initial conditions

(6) BROWN, K.R. and JOHNSON, G.W. "Optimal Guidance for Orbital Transfer," IBM Report #65-221-0003H, Huntsville, Alabama, 30 August 1965.

(7) BROWN, K.R. and JOHNSON, G.W. "Rapid Computation of Optimal Trajectories," IBM Journal of Research and Development, Volume 11, Number 4, July 1967, pp. 373-382.

which is readily accomplished by a single Newtonian iteration on the boundary-value problem. Feasibility of the use of OPGUID as a real-time guidance scheme for optimizing single-burn-arc (i.e. a sequence of thrusting stages separated by relatively short, fixed, staging intervals) orbital injection missions was demonstrated in 1966(8).

However, many orbit transfer problems require the use of several burn arcs separated by relatively long optimal coast arcs. Several authors(9),(10),(11) have reported on the need to use the more complex "direct" methods, quasi-linearization or generalized Newton-Raphson, in order to obtain convergence for this problem for the restricted case of motion in a plane and fixed boundary conditions. However, a multi-burn-arc version of OPGUID, developed in 1967(12), demonstrated that the attractive fundamental approach of OPGUID could successfully converge a general formulation of this problem, with variable boundary conditions. A sophisticated version of the multi-burn program (SWITCH) has been developed under Contract NAS 8-21315, that has successfully converged a variety of orbital transfer problems with an efficiency and reliability comparable to that of the original OPGUID. Currently a maximum of 0.25 seconds computation time is required per iteration on a CDC 6600 and normally only three to six iterations are needed for typical problems.

As a result, the indirect method of SWITCH is not only feasible but considerably superior to existing implementations of quasilinearization in convergence as well as efficiency. A principal feature of SWITCH is the use of classical two-body theory to render the computations for coast arcs explicit. Since high-thrust multi-burn orbit transfers usually involve coast arcs many times longer in duration than burn arcs, this results in a substantial saving in computation

---

(8) BROWN, K.R. and JOHNSON, G.W. "Real-Time Optimal Guidance," IEEE Transactions on Automatic Control, Vol. AC-12, No. 5, October, 1967.

(9) KENNETH, P. and McGILL, R. "Two-Point Boundary-Value Problem Techniques," Advances in Control Systems, Vol. 3, C. T. Leendes (ed.), 1966.

(10) McCUE, G.A. and BENDER, D.F. "Satellite Orbit Transfer Studies," NASA Report #N66-36050, 1966.

(11) O'MAHONY, M.S., ESKRIDGE, C.D. and HANAFY, L.M. "The Optimal Solution of Trajectory Problems Consisting of Several Extremal Subarcs by the Generalized Newton-Raphson Algorithm" American Astronautical Society, Paper AAS 67-348, 1967.

(12) JOHNSON, G.W. and SHULL, N.W. "Optimal Guidance with Controllable Propellant Mass Flow Rate," IBM Report CESD #009 December, 1967.

per iteration. A universal variable formulation of the two-body problem(13) with closed-form expressions for the state transition matrix is used. This formulation was adapted in a novel way to avoid the cumbersome computation of the three dimensional tensor of second partial derivatives of final state with respect to initial state that is required when computing the partial derivative of final costate with respect to initial state.

Since all the necessary partial derivatives are available at each iteration, as was the case for the original OPGUID, the SWITCH algorithm is appropriate for computing real-time corrections to in-flight perturbations.

Unlike the OPGUID algorithm, the SWITCH algorithm does require reasonable initialization. That is, it is not possible with SWITCH as it was with OPGUID to misalign the thrust direction by 90 or 180 degrees and retain convergence. However, rough estimates of impulsive solutions have proved more than adequate as initialization in every trial case. Moreover, deformation of the desired mission characteristics by very large amounts - for example a change in the eccentricity of the destination orbit from 0.0 to 0.5 - has been successful even when initialization values were unchanged. This is evidence that the SWITCH scheme would have a large safety margin in its ability to reconverge guidance solutions in response to worst-case real-time perturbations. However, it remains to be shown by vehicle flight simulation tests that the SWITCH algorithm possesses the speed and convergence properties that are necessary for real-time guidance. Such a demonstration would represent a significant advance since present real-time guidance schemes are not capable of revising an entire multiburn trajectory in response to in-flight perturbations, but can only modify one burn arc at a time.

---

(13) GOODYEAR, W.H. "Completely General Closed-Form Solution for Coordinates and Partial Derivatives of the Two-Body Problem," The Astronomical Journal, Vol. 70, No. 3, April, 1965, pp. 189-192.

2. Formulation of the Multi-burn Optimization Problem

In this section, different formulations of the fuel optimization problem for multiple burn trajectories are considered. It is shown that certain usual idealizing assumptions lead to an ill-posed optimization problem for which no solution exists, and several ways are discussed for avoiding such difficulties by more realistic problem statements.

An Idealized Problem Statement

The equations of space vehicle motion in a central gravitational field may be expressed as

2.1
$$\dot{r} = v$$
$$\dot{v} = -\frac{\mu r}{|r|^3} - \frac{c\dot{m}}{m}\frac{u}{|u|}$$

where r is position, v is velocity, the direction of the unit vector $u/|u|$ is control, c is the exhaust velocity of the rocket engine and $\dot{m}$ is the rate of change of vehicle mass due to propellant expenditure, which also is part of control and can be chosen within the limits $\alpha \leq \dot{m} \leq 0$.

The instantaneous rate of cost, L, is to be $-\dot{m}$; i.e. we wish to minimize mass loss (or maximize final mass), so, letting state be $x = (r,v,m)$ and costate be $p = \partial J^0/\partial x = (q,s,w)$, the Hamiltonian is

2.2 $$H = L + p^T\dot{x} = -\dot{m} + q^T v + s^T\left(\frac{-\mu r}{|r|^3} - \frac{c\dot{m}}{m}\frac{u}{|u|}\right) + w\dot{m}$$

which is minimized over thrust direction if and only if

2.3 $$\frac{u}{|u|} = -\frac{s}{|s|}$$

2.4 $$\min_{\substack{u \\ |u|}} H = \left(-1 + w + \frac{|s|c}{m}\right)\dot{m} + q^T v - \frac{\mu}{|r|^3}(s^T r)$$

Letting S (the switching function) be $(1 - w - c|s|/m)$, we have that 2.4 is minimized over $\dot{m}$ if $\dot{m} = \alpha$ for $S \leq 0$, and $\dot{m} = 0$ for $S > 0$. So

2.5 $$H^0 = S\alpha U(-S) + q^T v - \frac{\mu}{|r|^3}(s^T r)$$

where U is a unit step function, $U(x) = 0$ for $x < 0$, and $U(x) = 1$ for $x \geq 0$. The costate equations are easily seen to be

$$\dot{q} = r(-3\mu|r|^{-5} r^T s) + s(\mu|r|^{-3})$$

2.6 $$\dot{s} = -q$$

$$\dot{w} = -|s|\frac{c\dot{m}}{m^2}$$

It immediately follows that the time derivative of the switching function is independent of $\dot{m}$, or

2.7 $$\dot{S} = -\frac{c}{m}\frac{d}{dt}|s|$$

We have already incorporated two idealizing assumptions that are conventional:

(i) Apart from thrust acceleration, motion is Keplerian; hence usually periodic.

(ii) Thrust is truly proportional to mass rate; hence, mass loss is zero when thrust acceleration is zero.

We now add two more assumptions:

(iii) No terminal constraint is time-dependent.

(iv) There is no limit on (or penalty for) the number of separate coasts or burns.

Taken together, assumptions (i) - (iv) preclude the possibility of globally optimal solution trajectories for most missions. We can see this by arguing from the necessary conditions of optimality. First, observe that assumption (iii) implies that the Hamiltonian $H \equiv 0$. Second, observe that if we consider the vector $p' = (q,s,w-1)$, then altering the costate vector by multiplying p' by any positive scalar k does not affect the resulting trajectory or any of the necessary conditions. Therefore, of the seven degrees of freedom available in choosing the costate vector p, only 5 degrees of freedom are usable for selecting different optimal trajectories. Clearly, given an initial state x_0, all these 5 degrees of freedom plus the freedom of choice of terminal time, t_f, are needed in order for the problem of reaching a prescribed position and velocity r,v to be even locally well-posed. We shall see, however, using our assumptions (i) - (iv), that true optimality would impose still further requirements on the costate vector p so that fewer than 5 degrees of freedom are actually available in choice of costate. Suppose there is a fuel optimal trajectory for transfer from an original orbit to a given destination orbit. Let r,v,m be any state on this optimal trajectory at which the switching function S is negative. We suppose, by assumption (i), that r,v define an elliptical (or circular) orbit. Let r',v' be any other position and velocity on that elliptical orbit. Starting from r',v',m, an optimal maneuver for reaching the given destination orbit must be simply to coast around to r,v and then follow the original optimal maneuvers from that point on. For, if there were a cheaper maneuver, it could have been combined with the first part of the original optimal trajectory to r,v,m plus a coast to r',v',m, thus obtaining a better solution to the original problem. Now this maneuver of coasting from r', v', m to r,v,m and following the old trajectory thereafter, being optimal, must itself satisfy the necessary conditions 2.1 - 2.6. It can be easily verified, however, that any such trajectory agreeing with the original trajectory during a burn and at least one switching point (point at which $S = 0$), must agree exactly even in costate with the original trajectory (apart from a positive scalar multiplication of the vector (q,s,w-1)). This is a contradiction, because on the new trajectory we must have $S = 0$ at r,v,m whereas we supposed that S was negative on the original trajectory at that point. The only way out of the contradiction is to suppose either that the original trajectory was not, in fact, optimal after all or that S is never negative along an optimal trajectory. In the latter case, S must either be always positive (no burns at all), or by 2.7, we must have $\dot{S} = 0$ whenever $S = 0$. But this latter requirement, again by 2.7, clearly removes at least one degree of freedom from the available choices of costate.

A more direct way of seeing the typical nonexistence of mass optimal trajectories under assumptions (i) - (iv) is available if we accept two lemmas. Lemma 1: given an optimal impulsive solution to an orbital transfer problem, there is no finite burn maneuver with as low a cost. Lemma 2: sufficiently small single-impulse maneuvers are approached arbitrarily closely in cost by the best single finite burn, single-coast maneuvers for achieving the same orbit change. Both of these lemmas are easily verified from Robbins(14). We can easily see, since any impulse can be broken up into a large number of very small impulses, that lemma 2 implies that any one-impulse transfer between elliptical orbits can be matched arbitrarily closely in cost by a sufficiently large number of finite burns and coasts. This, in turn, together with lemma 1 show that the cost of an optimal impulsive maneuver must be a greatest lower bound of the set of all achievable finite burn costs, but not itself a member of that set.

Realistic Problem Statements

The main result of assumptions (i) - (iv) that leads to nonexistence of optimal trajectories is that, at any point in a trajectory, one can insert a coast of arbitrary duration without increasing the cost of that

(14) H.M. ROBBINS. "An Analytical Study of the Impulsive Approximation," AIAA Journal, Vol. 4, No. 8, pp. 1417-1423, August, 1966.

trajectory at all. If there were a fixed terminal time by which the maneuver had to be completed, or a penalty for restarting the rocket engine, or a nonzero rate of cost during coasts, then there would be a limit to how far the expedient of inserting coasts could be pushed. In practical orbital transfer missions, of course, there are either time limits or engine restart penalties or coast duration penalties. For example, existing rocket engines actually lose a small amount of mass per second, even during coasts, through venting of vaporized propellant. Also, astronauts and even on-board equipment such as flight computers and telemetry electronics, can function in space only for limited times. There is also a penalty for restarting and stopping the rocket engine. In fact, even assumption (i) is not precisely satisfied because perturbations due to earth oblateness, third body effects, solar wind, and other flight perturbations cause the motion to be not precisely periodic, although it is very nearly so. It would, however, be extremely difficult and also unsatisfactory to try to obtain a unique solution only on the strength of these slight departures from periodicity.

There are three attractive alternatives for changing the problem definition so that a unique optimum will exist: a) to impose a terminal time constraint, b) to impose a limit on the number of engine restarts, or c) to incorporate a non-zero (and non-negligible) rate of cost during coasts. Any one of these alternatives would be a sufficient device and may be genuinely appropriate depending on the mission being analyzed. It is important to observe that the solutions given by alternative a) and the solutions given by alternative c) form the same class of trajectories. For consider a solution given by alternative a). The necessary conditions for optimality are as before except that the Hamiltonian H is not zero, although it is, of course, a constant along the trajectory. It can be argued that in this case the constant H must be negative, so let L_0 (a positive number) be -H. If we now consider L_0 to be a rate of cost that is independent of thrust, and remove the time constraint , we get the same solution but to a different problem: the problem of optimizing over free terminal time in the presence of an added constant rate of cost equal to L_0. So, insofar as the mathematical nature of the solution is concerned, it makes no difference if we use alternative a) or alternative c). Alternative b), however, gives solutions that are not also solutions to problems involving alternatives a) or c). In fact, we can see that solutions given by alternative b) must satisfy all of our original necessary conditions 2.1 - 2.6 plus the condition that $H \equiv 0$. For these necessary conditions can be deduced using only the assumption that each switching time (that is, time at which the engine is started or stopped) is optimized relative to the rest of the trajectory and that the trajectory thrust direction history is optimized relative to the switching times. However, alternative b) can be made to approach alternative c) while holding the time penalty L_0 fixed by increasing thrust levels. Thus for high thrust missions, alternatives a),b), and c) may turn out to be practically equivalent.

Numerical Considerations

It should be recognized that even when we resolve the basic problem of non-existence of global optima by changing the problem definition according to alternative a), b), or c), some results of the original ill-conditioning are still important. We know now that since every trajectory that is fuel-optimal with respect to the choice of each of a fixed number of switching times satisfies the necessary conditions 2.1 - 2.6, there must be an infinite progression of solutions to the boundary value problem all satisfying 2.1 - 2.6, and each one cheaper than the last. Let $\rho_1, \rho_2, \ldots$ be a corresponding progression of normalized initial costate vectors (ρ_i = the result of dividing p_i by its own magnitude). Since the space of normalized costate is closed and bounded, it follows that this space contains a limit point ρ_∞. Now consider the function $x(t) = g(t,\rho)$ defined as the result of propagating initial state forward in time to time t from initial state x_0 at t_0 using initial normalized costate = ρ.

It is clear that in the vicinity of p_∞ some functions of the trajectory (e.g. its duration) become arbitrarily sensitive to small variations in ρ. Such unbounded sensitivities would be a serious hazard to any iterative scheme for obtaining numerical solutions, whether to the problem definition given by alternative a), b) or c). It is desirable to choose an approach under which the procedure searches over a set of independent variables such that either the dependent variables are never extremely sensitive to the independent variables and conversely or the nature of the search confines the independent variables to a "safe" region within which extreme sensitivities in either direction are avoided. It is evident, however, that no choice of the independent, and dependent, variables for the search can avoid extreme sensitivities unless the range of values of the variables is confined to a region which excludes trajectories for which ρ is near ρ_∞.

Alternative b) has the advantage that a simple and intuitively meaningful constraint of limiting the number of burns to two, or, in some cases, three, can be applied to each individual iterate in the search so that the range of values of the independent and dependent variables automatically excludes by a large margin the vicinity of trajectories corresponding to ρ near ρ_∞. There is also a distinct advantage for the logical structure of an implemented algorithm for numerical solution of problem b) in that the number of switching points is fixed, instead of varying from iterate to iterate as it would under alternatives a) and c). Since both these advantages carry real weight from a practical point of view, we have adopted alternative b) as the preferred approach. However, if for any reason an implementation of alternative a) or c) is desired, most of the current algorithm would be transferable without modification to the revised problem.

Thus the problem to be solved consists of finding a sequence of burns and coasts that achieves the desired orbit with minimum fuel subject to the constraint that the number of separate burns is limited to k, where k is usually two, sometimes three. In some cases, it is desired to permit an initial coast prior to the first burn as well as one coast after each burn. In certain other cases (such as circular to circular coplanar missions) the initial coast is not permitted since its length would be indeterminate under the necessary conditions .

3. Iterative Solution of the Boundary Value Problem

In section 2, both dynamical necessary conditions and necessary boundary conditions were discussed for several possible multi-burn optimization problems. In this section, a method is described in detail for the numerical solution of one of these boundary value problems. The particular problem chosen, called alternative b) in section 2, requires that trajectory which achieves the desired orbit (or orbital characteristics) with minimum fuel expenditure subject to a limit on the total number of separate burn arcs.

The dynamical necessary conditions for this problem are equations 2.1, 2.3, and 2.6. The boundary conditions are given at the left end by the initial position, velocity, and mass of the vehicle and at the right end by six conditions defining the desired characteristics of the destination orbit, which are discussed in section 5 for several important mission types. The intermediate point constraints are that the switching function S be zero at each interior switching time. In addition we require that the switching function be positive on coasting arcs and negative on burn arcs. This last requirement can be used to detect the desirability of adding further burn arcs.

The sequence of burns and coasts of the desired trajectory is partially prescribed in advance as part of the problem statement. The number of separate burn arcs may be chosen, and the initial arc may be designated as a burn or a coast. In a fundamental sense, a genuine orbit transfer problem which begins at a prescribed position and velocity, requires an initial coast arc in order to be fuel-optimal. But two important cases exist in which the initial coast must be suppressed. First, in the case of a circular to circular coplanar mission, the length of an initial coast would be completely arbitrary in the sense that all initial coast durations are equally compatible with fuel optimality. Second, there are cases in which mission geometry constraints preclude an optimal initial coast because it would result in a fall into the lower atmosphere or even beneath the earth's surface.

For purposes of specifying precisely the independent and dependent variables of the iterative scheme for solution of the boundary value problem, it is necessary to treat separately the case where an initial optimal coast is permitted and the case where the trajectory is required to begin with a burn arc. Let t_0 be the initial epoch for which the initial position, velocity, and mass are given: $r(t_0) = r_0$, $v(t_0) = v_0$, $m(t_0) = m_0$. Let n be the number of separate burn arcs (usually two or three). Then, for a trajectory beginning with a coast arc, the switching times are t_{2j-1} = start of j^{th} burn and t_{2j} = end of j^{th} burn for $j = 1,\ldots,n$. Combining the constancy of H with the condition that $S = 0$, we have, at each switching point, by equation 2.5, the condition that $q^T v - \mu s^T r/|r|^3 = H$. Recalling that this same condition is the principal transversality condition in the single-burn case, we will call $q^T v - \mu s^T r/|r|^3$ the transversality variable Tv and so write the conditions as $Tv(t_i) = H$, $i = 1,\ldots,2n$. Now this gives only about half of the actual necessary conditions at switching times, for it is easily seen from 2.1 and 2.6 that Tv(t) is identically constant along any coast arc, so if $Tv(t_{2j}) = H$, it is automatically true that $Tv(t_{2j+1}) = H$ also. However, additional necessary conditions can be deduced from the fact that the switching function S must be zero at the end of a coast as well as at the beginning; thus $S(t_{2j+1}) - S(t_{2j}) = 0$. Since $S = 1 - w - c|s|/m$ and since $\dot{m}$ and hence $\dot{w}$ are zero during coasts, this implies that $|s(t_{2j+1})| - |s(t_{2j})| = 0$. Thus we obtain the following set of $2n - 1$ necessary conditions at intermediate switching points:

(a) $\quad Tv(t_{2j-1}) = Tv(t_{2j}), \quad j = 1, \ldots, n$

(b) $\quad |s(t_{2j+1})| = |s(t_{2j})|, \; j = 1,\ldots,n - 1$

The advantage of this set of conditions compared to the more straightforward requirement that $H(t_i) = 0$ and $S(t_i) = 0$, $i = 1,\ldots,2n - 1$ is that it becomes possible to completely drop the costate variable w from all equations. For we can regard w_i as defined by the requirement that $S(t_i) = 0$ and then all necessary conditions are implied by (a) and (b).

For missions in which the trajectory is required to begin with a burn arc, the switching conditions are altered in that the t_{2j} and t_{2j+1} now require conditions appropriate to the beginning and ending of burn arcs respectively and in that there is no necessary condition applied at the beginning of the first burn. Thus we have 2n - 2 conditions:

(a') $\quad Tv(t_{2j}) = Tv(t_{2j+1}), \quad j = 1, \ldots, n - 1$

(b') $\quad |s(t_{2j})| - |s(t_{2j-1})| = 0, \; j = 1,\ldots,n - 1$

Observe that we have one more switching condition on an n-burn mission beginning with a coast than on an n-burn mission beginning with a burn. This corresponds to the fact that when an initial coast is permitted, its duration is an added variable for optimization.

In summary, we can view the boundary value problems as follows:

For a mission beginning with a coast arc: Find values for the six components of initial costate and the 2n switching times $t_1,\ldots,t_{2n}$ such that the result of integrating 2.1 and 2.6 forward from t_o to t_{2n} satisfies six right end mission conditions at t_{2n}, the 2n - 1 intermediate switching conditions given by (a) and (b), and the condition $|u_o| = 1$. For a mission beginning with a burn arc: Find values for the six components of initial costate and the 2n - 1 switching times $t_1,\ldots,t_{2n-1}$ such that the results of integration 2.1 and 2.6 forward from t_o to t_{2n-1} satisfies six right end mission condition, the 2n - 2 intermediate switching point condition (a'), (b') and the condition $|u_o| = 1$.

To solve numerically either of these problems, we apply the multi-dimensional Newton method. Let the six components of initial costate and the 2n switching times be combined into a vector y of dimension 2n + 6, and let the six constrained right end variables, the 2n - 1 intermediate switching variables of (a) and (b), and the variable $|u_o|$ be combined into a vector z of dimension 2n + 6. Then if z* is a vector of desired values for the components of z, the boundary value problem for missions with initial coasts becomes the problem of finding a vector zero of z* - z(y). A similar definition applies to the problem when an initial burn is required but with z and y having dimension 2n + 5 instead of 2n + 6.

The Newtonian iterative solution scheme is familiar; starting from an initial guess y_o, we obtain successive estimates

$$3.1 \qquad y_{i+1} = y_i + \left[\frac{\partial z(y)}{\partial y}\right]^{-1}_{y_i} \left[z^* - z(y_i)\right] K$$

where K is a scalar between zero and one. When K is one, the scheme is a pure Newton's method; when K is less than one, the adjustment $y_{i+1} - y_i$ represents an attempt to obtain $z_{i+1} = z_i + K(z^* - z_i)$. This is often useful when $z^* - z_i$ would be too large a change in z to correspond approximately linearly to a change in y.

In order to be able to implement 3.1, it is necessary to be able to compute the vector z(y) and the matrix $\partial z/\partial y$ as functions of y. In the present case this means the ability to compute a trajectory and its first variations with respect to initial costate and the switching times t_i. The trajectory is computed as a sequence of burn arcs and coast arcs, burn arcs being computed by numerical integration of 2.1, 2.3, and 2.6, and coast arcs being computed explicitly by a method discussed in section 4. First variations of burn and coast arcs are computed concurrently with the arcs themselves.

The method of integration used for the burn arcs is a fourth-order Runge-Kutta scheme suited for equations such as 2.1 and 2.6 (and the associated equations of first variation) which can be expressed as second order differential equations with first derivatives absent. In order to estimate and control truncation error by altering step sizes, a Richardson extrapolation method is used with a special combination of Runge-Kutta steps consisting of three steps of size h of integration of 2.1 and 2.6 together with one overlapping step of size 3h of integration of 2.1, 2.6. and the associated equations of first variation. The resulting scheme has proved very efficient and reliable and is described in full detail in reference 7.

A special consideration that deserves mention in connection with the computation of $\partial z/\partial y$ is the correct generation of those columns of the matrix that represent partial derivatives with respect to switching times. It is well known that if we have a vector system of differential equations

3.2 $$\dot{\alpha} = f(\alpha)$$

the partial derivative of the solution $\alpha(t)$ with respect to any variable β on which the initial value α_o depends, satisfy the equations

3.3 $$\frac{d}{dt}\frac{\partial \alpha(t)}{\partial \beta} = \frac{\partial f(\alpha)}{\partial \alpha}\frac{\partial \alpha(t)}{\partial \beta}$$

and the initial condition, when t_o does not itself depend on β, is

3.4 $$\left.\frac{\partial \alpha(t)}{\partial \beta}\right|_{t=t_0} = \frac{\partial \alpha_o}{\partial \beta}$$

The problem of computing $\partial z/\partial y$ for purposes of the iteration 3.1 consists mainly of determining the partial derivatives of such a vector α consisting of both state and costate, with respect to the components of y, and this is accomplished by solving 3.3 explicitly on coast arcs and numerically on burn arcs. But in the case of those components of y that are switching times t_i, the initial condition is not 3.4 as usual. Instead the condition is as follows:

3.5 $$\left.\frac{\partial \alpha(t)}{\partial t_i}\right|_{t=t_i} = \dot{\alpha}(t_i-) - \dot{\alpha}(t_i+)$$

To appreciate this, it is sufficient to observe that since what is desired is the initial value of the (continuous) solution to a differential equation (3.3), we really want

3.6 $$\left.\frac{\partial \alpha(t)}{\partial t_i}\right|_{t=t_i+} \overset{\Delta}{=} \lim_{t \to t_i+} \left[\frac{\partial \alpha(t)}{\partial t_i}\right]$$

which, under the assumption that $\dot{\alpha} = f_1(\alpha)$ before t_i and $\dot{\alpha} = f_2(\alpha)$ after t_i, gives the following relation between α at time t shortly after t_i and α at time T shortly before t_i

3.7 $$\alpha(t) = \alpha(T) + f_1(\alpha(\xi_1))(t_i - T) + f_2(\alpha(\xi_2))(t - t_i)$$

where $\xi_1 \varepsilon (T, t_i)$ and $\xi_2 \varepsilon (t_i, t)$. Differentiation of 3.7 with respect to t_i yields

3.8 $$\frac{\partial \alpha(t)}{\partial t_i} = f_1(\alpha(\xi_1)) - f_2(\alpha(\xi_2)) + O(t_i - T) + O(t - t_i)$$

which in the limit as $T \to t_i-$ and $t \to t_i+$ gives 3.5.

4. Efficient Coast Arc Computations

The computations required for burn arcs of the multi-burn SWITCH algorithm are essentially the same as those of the OPGUID algorithm, for single burn arc missions. The chief remaining computational section of the algorithm is that which updates state, costate, and their partial derivatives along a coast arc. Considerable effort was expended in planning these computations in order to take full advantage of the computational economies offered by the known efficient (semi-) explicit schemes for computing state and its partial derivatives along coast arcs. After extended analysis, it was found that costate and its partial derivatives with respect to initial costate and state could be computed easily using the corresponding computations for state and its partials.

For each coast, we need values of r, v, u, and $\dot{u}$ (where $u \triangleq -s$) at the end of a coast given their values at the beginning and given the duration in time of the coast. We require in addition the partial derivatives of the final values with respect to the initial values since these partial derivatives form a necessary link in the chain of computations leading ultimately to the partial derivatives of all the dependent variables of the boundary value search, described in section 3 , with respect to the independent variables. For purposes of the coast arc computations we can ignore m and w because their final values are simply equal to their initial values and neither m nor w appears in the differential equations for r,v,u and $\dot{u}$ which apply during coasts:

4.1 $$\ddot{r} = \frac{-\mu r}{|r|^3}$$

4.2 $$\ddot{u} = r(3\mu|r|^{-5}r^T u) + u(-\mu|r|^{-3})$$

It turns out to be convenient in what follows to define state x as $(r,\dot{r})$ and (modified) costate γ as $(u,\dot{u})$, even though the costate that falls out directly from the calculus of variations or the maximum principle is $p = (\dot{u},-u)$. The advantage of γ over p is that it bears significantly simpler relations to state x.

Completely general closed-form solutions are well-known for the propagation of state x according to 4.1 and for the corresponding computation of the state transition matrix $\partial x(t)/\partial x(t_o)$. A good formulation is Goodyear's[13] which bases all computations on a given initial value for state x_o and a time interval $t - t_o$. What we need beyond what Goodyear's or other usual formulations give are these additional six by six matrices: $\partial \gamma(t)/\partial \gamma(t_o)$, $\partial \gamma(t)/\partial x(t_o)$, and $\partial x(t)/\partial \gamma(t_o)$. Now, since $\dot{x}$ is independent of γ along a coast, we can see immediately that the last of these matrices vanishes.

4.3 $$\partial x(t)/\partial \gamma(t_o) \equiv 0$$

First we observe that if we write 4.2 as a differential equation for the whole (modified) costate vector γ, we obtain

4.4 $$\frac{d}{dt}\gamma(t) = \begin{bmatrix} 0_3 & I_3 \\ \left.\frac{\partial \mathbf{r}}{\partial r}\right|_{(t)} & 0_3 \end{bmatrix} \gamma(t) = \frac{\partial \dot{x}(t)}{\partial x(t)} \cdot \gamma(t)$$

Hence the differential equations for the matrices $\partial \gamma(t)/\partial \gamma(t_o)$ and $\partial x(t)/\partial x(t_o)$ are the same

4.5 $$\frac{d}{dt}\frac{\partial \gamma(t)}{\partial \gamma(t_o)} = \frac{\partial \dot{x}(t)}{\partial x(t)}\frac{\partial \gamma(t)}{\partial \gamma(t_o)}$$
$$\frac{d}{dt}\frac{\partial x(t)}{\partial x(t_o)} = \frac{\partial \dot{x}(t)}{\partial x(t)}\frac{\partial x(t)}{\partial x(t_o)}$$

Since both state and costate transition matrices must be initially equal to the 6 x 6 identity matrix I_6, we obtain

4.6 $$\frac{\partial \gamma(t)}{\partial \gamma(t_o)} \equiv \frac{\partial x(t)}{\partial x(t_o)} \triangleq \phi(t,t_o)$$

A useful property of the matrix $\phi(t,t_o)$ is the symplectic property which may be written:

4.7 $$\phi(t,t_o)^T J \phi(t,t_o) = J$$

where

4.8 $$J = \begin{bmatrix} 0_3 & I_3 \\ -I_3 & 0_3 \end{bmatrix}$$

This property may be derived by observing that 4.7 holds trivially at t_o since $\phi(t_o,t_o) = I_6$ and by writing out

$$\frac{d}{dt} \phi(t,t_o)^T J\phi(t,t_o)$$

and noticing that it vanishes. The symplectic property is principally useful in obtaining $\phi(t,t_o)^{-1} = \phi(t_o,t)$ for, by 4.7 and 4.

4.9 $$\phi(t,t_o)^{-1} = J^T\phi(t,t_o)^T J$$

which is a simple rearrangement (with some changes of sign) of the elements of $\phi(t,t_o)$ itself.

Now since $\dot{\gamma}$ is linear homogeneous in γ, $\gamma(t) = \partial\gamma(t)/\partial\gamma(t_o)\ \gamma(t_o)$, i.e.

4.10 $$\gamma(t) = \phi(t,t_o)\gamma(t_o)$$

The remaining matrix to be computed, $\partial\gamma(t)/\partial x(t_o)$ is most directly expressable from 4.10; thus

4.11 $$\left[\frac{\partial\gamma(t)}{\partial x(t_o)}\right]_{ij} = \sum_k \frac{\partial\phi(t,t_0)_{ik}}{\partial x_j(t_o)}\ \gamma(t_o)_k$$

This equation would be very cumbersome to evaluate directly because of the 6 x 6 x 6 tensor of second partial derivatives

4.12 $$\frac{\partial\phi(t,t_o)_{ik}}{\partial x_j(t_o)} = \frac{\partial}{\partial x_j(t_o)}\frac{\partial x_i(t)}{\partial x_k(t_o)} = \frac{\partial}{\partial x_k(t_o)}\frac{\partial x_i(t)}{\partial x_j(t_o)}$$

But, by equation 4.12, 4.11 can be rewritten thus:

4.13 $$\left[\frac{\partial \gamma(t)}{\partial x(t_o)}\right]_{ij} = \sum_k \frac{\partial}{\partial x_k(t_o)} \frac{\partial x_i(t)}{\partial x_j(t_o)} \gamma_k(t_o)$$

which can be interpreted as the <u>linearized</u> change d $\phi(t,t_o)_{ij}$ in $\phi(t,t_o)_{ij}$ due to a change in x_o, $dx_o \doteq \gamma(t_o)$. This linearized change $d\phi$ can be computed with very slight added labor when ϕ itself is computed.

This computation scheme has been implemented in a computer program which is much simpler than and about one half the size of the only known earlier program for explicit computation of state, costate, and their transition matrices along a coast arc(15).

5. Right End Conditions for Various Orbital Missions

Regardless of the kind of orbit transfer mission, the left end boundary conditions are given by the position, velocity, and mass of the vehicle at the initial epoch t_o, and the intermediate switching point conditions are as given in section 3. These conditions leave, in each case, six degrees of freedom remaining at the right end, which must be eliminated by necessary conditions that define the particular kind of orbit transfer desired. When fewer than six conditions define the desired character of the destination orbit, supplementary transversality conditions, requiring that the unconstrained orbit parameters be chosen optimally, are added to make a total of six independent conditions.

The most basic mission is defined by desired values for a complete set of five orbital constants; that is, five independent functions of position

(15) PRESTON, E.L. "Computer Program for State Transition Matrices," Fall, 1967 Co-op Period, Mechanical Engineering Department, Purdue University.

and velocity that are constant in the absence of thrust. There cannot, of course, be any set of six independent orbital constants in this sense, for that would imply that position and velocity are themselves constant during unpowered flight. Thus the five orbital constant mission is the most fully defined orbit transfer mission that does not involve the time origin of the destination orbit. Five orbital constants completely determine the locus of the orbit, but not the absolute time at which any particular position is reached.

The most basic six constraint mission is rendezvous, which can be thought of as a mission in which five orbital constants plus the time origin of the orbit are specified. Thus not only the complete locus of the orbit but also the locus as a function of absolute time is prescribed.

Also of interest are several missions in which fewer than five orbital constants are specified. While a full set of five orbital constants defines completely the plane of the orbit and the size, shape, and orientation of the conic within that plane, it is possible with fewer than five orbital constants to prescribe only part of the geometry of the orbit and optimize over the remainder.

From the calculus of variations or the maximum principle it is known that if the mission requirements are prescribed values for only k functions of final state

5.1 $$g_i(x_f) = 0 \quad i = 1,\dots,k$$

where the state vector x has $n > k$ components, then optimality with respect to the $n - k$ remaining degrees of freedom in final state requires that the final costate vector p_f lie the space spanned by the gradients of the constrained functions. This means that if $n - k$ vectors $a_i(x)$, $i = k + 1,\dots,n$, can be found such that the a_i span the space orthogonal to the space spanned by the gradients $\frac{\partial g_i(x)}{\partial x}$ of the g_i, then p_f must be orthogonal to $a_i(x)$, $i = k + 1,\dots,n$. In equations, we have

5.2 $$p_f^T a_i(x) = 0 \quad i = k + 1,\dots,n$$

5.3 $$\frac{\partial g_i(x)}{\partial x} a_j(x) = 0; \; i = 1,\ldots,k; \; j = k+1,\ldots,n$$

In the case of the fundamental five orbital constant mission, $k = 5$ and $n = 6$, so only one vector $a_6(x)$ is needed and it must be orthogonal to the gradient of every orbital constant. By definition, a function $g_i(x)$ is an orbital constant just in case

5.4 $$0 = \frac{d}{dt}[g_i(x)] = \frac{\partial g_i(x)}{\partial x}\dot{x}$$

where $\dot{x}$ is computed for unpowered flight,

5.5 $$\dot{r} = v; \; \dot{v} = -\frac{\mu r}{|r|^3}$$

Therefore it is immediate that one vector $a_6(x)$ that will satisfy 5.3 when the g_i are all orbital constants is

5.6 $$a_6(x) = \left(v, -\frac{\mu r}{|r|^3}\right)^T$$

Thus the transversality condition 5.2 for this mission can be written as

5.7 $$q_f^T v_f - \frac{\mu}{|r|^3} s_f^T r_f = 0$$

which will be recognized as the same as the switching condition which was applied at the end of an intermediate burn during the trajectory, now applied also to the end of the last burn. That is, 5.7 is equivalent to

5.8 $$Tv(t_f) = 0$$

The five constrained functions $g_i(x)$ can be expressed in various equivalent ways, but probably the simplest formulation is this: Let $h = r x v$ be the angular velocity vector and let $e = -\left[\frac{r}{|r|} + \frac{h x v}{\mu}\right]$ be the "eccentricity vector" whose magnitude is eccentricity and whose direction is the direction of pericenter. Then define $g_i(x)$, for $i = 1,\ldots,5$ as follows:

5.9
$$g_1(x) = h_1 - h_1^*$$
$$g_2(x) = h_2 - h_2^*$$
$$g_3(x) = h_3 - h_3^*$$
$$g_4(x) = e_1 - e_1^*$$
$$g_5(x) = e_2 - e_2^*$$

where "*" means "desired value of."

It is easy to verify that these five functions $g_i(x)$ are independent and uniquely define the values of all orbital constants except in the isolated case where $h_3(x) = 0$, which may be accommodated by reordering the coordinate axes. Accordingly, the six right end conditions for the basic five orbital constant missions have been implemented as 5.7 and 5.1 for $g_i(x)$ defined in 5.9.

Right end conditions for the rendezvous mission are easier to formulate but slightly harder to implement. We require that at the end of the last burn, at t_f, the position and velocity of the vehicle must match the position and velocity of a "target body." The position and velocity of the target, R(T) and V(t), must be given as functions of absolute time t, but the target body as such may be fictitious; as, for example, in the case of a synchronous orbit injection mission in which no actual body yet occupies the desired orbit. The six right end conditions are

5.10
$$\left.\begin{aligned} g_i(x_f,t_f) &= r_i(t_f) - R_i(t_f) = 0 \\ g_{i+3}(x_f,t_f) &= v_i(t_f) - V_i(t_f) = 0 \end{aligned}\right\} i = 1,2,3$$

To implement 5.10 requires an algorithm for computing $R(t_f)$ and $V(t_f)$ for variable t_f. Since R(t), V(t) represent a free (unpowered) trajectory, this can be carried out using parts of the computations described in section 4, provided $R(t_o)$ and $V(t_o)$ are supplied for some initial epoch t_o.

Useful four, three, and two-orbital constant missions are defined and transversality conditions answering to 5.3 derived for them on pp. 502 and 503 of reference 8.

6. Test Results

The convergence properties of the multi-burn orbital transfer routine SWITCH were evaluated using basic Apollo type missions. The missions involved transfer from low earth orbits of about 100 nautical mile altitudes to orbits with altitudes ranging from 200 nautical miles to 19,300 nautical miles. Both high and low thrust vehicles were used in the evaluation of the program.

The first mission consisted of transfer from a near-circular orbit with an altitude of approximately 100 nautical miles to a circular orbit at a synchronous altitude of 19,300 nautical miles. The vehicle was assumed to have an initial mass of 1.26×10^7 kgm, an exhaust velocity of 4.15 km/sec and a mass rate of 2.24×10^4 kgm/sec. In order to use the SWITCH program one must estimate the independent variables of the boundary value search (i.e. the initial costate vector and the lengths of the burn and coast arcs) and fix the number of burn and coast arcs. The number of separate burn arcs was set at four (coast-burn-coast-burn) and the initial costate vector and the lengths of arcs were chosen to correspond to values used in evaluation of an earlier program. The SWITCH algorithm converged to the solution in one iteration.

Since the initialization for this first mission was essentially equal to the converged solution (presented in Table 1), the final orbit was deformed into an ellipse to test the program more fully. The deformation was accomplished by choosing the eccentricity vector, e (the vector whose magnitude is equal to the eccentricity and which points in the direction of the pericenter), to be parallel to the injection vector obtained in the first case and with the magnitude of the vector allowed to vary from .05 to .5. The initial orbit, vehicle parameters, and initial estimates of costate and arc times were fixed. The results are presented in Table 1 and as expected the number of iterations for convergence increased with increase in the eccentricity of the final orbit.

Of special interest is the case where the eccentricity was set at .5. The converged lengths of the second coast and burn arcs were changed by approximately 35 percent and a significant change was also required in the initial costate vector. The algorithm was able to overcome these difficulties and converged in five iterations.

The right end boundary constraints were changed to allow solution of rendezvous missions and the same test cases were used to evaluate the altered algorithm. The solutions obtained for the five constraint missions were used to specify the radius vector, velocity vector and time necessary to describe the target vehicle. As indicated by Table 1 the rendezvous missions generally required a larger number of iterations to converge. The sensitivity of the algorithm to changes in the time frame of the target vehicle was also investigated. The final orbit eccentricity was fixed at .05 and the target vehicle was allowed to be both late and early in achieving the fixed position and velocity. The plus and minus 200 second change in orbital epoch time required, in each case, about the same number of iterations for convergence as the nominal epoch case. The direction of the time change did not seem to affect the convergence properties.

The second test mission consisted of transfer from a 100 n.m. circular orbit to a 19,300 n.m. circular orbit which was rotated 44 degrees out of the plane. The vehicle characteristics were chosen to be the same as used in the first cases and the algorithm was able to converge to the solution in five iterations. The five constraint solution was again used to define the position of the target vehicle as a function of time and the rendezvous form of the algorithm converged in eight iterations. The results are presented in Table 2, and, as shown, large changes were required in the initial costate vector.

The third mission consisted of a burn-coast-burn transfer from a 96 n.m. circular orbit to a 196 n.m. circular orbit with a relatively low thrust vehicle. The vehicle was assumed to have a mass of 9248 kgm, a mass rate of 2.31 kgm/sec and an exhaust velocity of 2.16 km/sec. The initial costate vector was chosen such that u_o was in the direction of the vehicle velocity vector and that $\dot{u}_o$ was in the direction of the acceleration vector of the coasting vehicle. The entire initial costate vector was scaled such that the magnitude of the u_o vector was one. The length of the first burn arc was determined from the amount of velocity change required to deform the 96 n.m. orbit into a 96 n.m. x 196 n.m. orbit. The coast arc was chosen to be one half of the period of the elliptical orbit and the length of the second burn arc was chosen to gain the velocity necessary to circularize the orbit at apogee. The problem converged in two iterations and required five percent changes in the lengths of the burn arcs as well as a .7 change in magnitude of the final u vector.

The last mission tested required a transfer from 81 n.m. x 120 n.m. orbit to a coplanar 210 n.m. circular orbit. The vehicle characteristics consisted of an initial vehicle mass of 14,486 kgm, mass rate of 29.36 kgm/sec, and exhaust velocity of 3.07 km/sec. The mission was chosen to consist of four spearate arcs and the initialization of the costate vector and the switching times was accomplished by a rough pencil and paper estimation of the impulsive solution similar to that for the third mission. In this case the length of the initial coast arc was required and it was chosen to insure that the initial burn arc was centered around perigee. The converged solution was within ten percent of the estimated arc lengths and required a .07 change in the magnitude of the final u vector. This test mission required four iterations for convergence.

All cases in which even a crude impulsive solution was used for initialization resulted in essentially immediate convergence. Only when an initialization appropriate to one mission was carried over unchanged to a significantly altered mission did the program require an appreciable number of iterations.

	Destination Orbit Specifications		Duration of Burn and Coast Arcs				Magnitude of Changes in u		Number of Iterations to Converge
Type of Mission	Eccentricity	Minimum Altitude (n. m.)	1st Coast Arc (sec)	1st Burn Arc (sec)	2nd Coast Arc (sec)	2nd Burn Arc (sec)	$\vert\Delta u_o\vert$	$\vert\Delta u_F\vert$	
Five Orbital Constant	0.	19,300	399.83	255.82	18,729.54	129.11	.24E-3	.29E-3	1
	.05	18,300	401.16	253.47	17,572.73	134.87	.80E-2	.26E-2	3
	.1	17,200	402.47	251.14	16,541.93	140.57	.16E-1	.55E-2	3
	.5	11,700	412.63	232.95	11,155.42	184.04	.78E-1	.27E-1	5
RENDEZVOUS NOMINAL	0.	19,300	399.83	255.82	18,729.51	129.11	.24E-3	.29E-3	1
	.05	18,300	401.15	253.47	17,572.72	134.87	.81E-2	.26E-2	3
	.1	17,200	402.48	251.14	16,541.81	140.57	.16E-1	.55E-2	5
	.5	11,700	412.61	232.95	11,155.54	184.04	.78E-1	.27E-1	28
Target 200 sec LATE	.05	18,300	379.53	253.80	17,827.69	134.62	.17E-0	.29E-1	3
Target 200 sec EARLY	.05	18,300	423.07	253.80	17,316.21	134,63	.16E-0	.19E-1	4
FIRST ITERATION SPECIFICATIONS			400.	255.82	18,727.46	129.11	-	-	-

Table 1

Low Altitude to Coplanar Synchronous Missions

Type of Mission	Duration of Burn and Coast Arcs: 1st Coast Arc (sec)	1st Burn Arc (sec)	2nd Coast Arc (sec)	2nd Burn Arc (sec)	Magnitude of Changes in u: $\lvert\Delta u_o\rvert$	$\lvert\Delta u_F\rvert$	Number of Iterations to Converge
Five Orbital Constant	1211.55	255.41	18,728.65	124.99	.58E-1	.78E-0	5
RENDEZVOUS	1211.56	255.41	18,728.56	124.99	.59E-1	.78E-0	8
FIRST ITERATION SPECIFICATIONS	1113.69	255.18	18,729.19	118.80	-	-	-

Table 2

Low Altitude to Synchronous with 44^o

Plane Change

7 Conclusions

The development of the OPGUID algorithm in 1965 provided an efficient and reliable means for optimizing single-burn-arc transfer missions. The basic approach of the OPGUID scheme was to use an ordinary shooting method but with carefully chosen coordinate systems and numerical methods aimed at speed and range of convergence. The simulated use of OPGUID as a real-time guidance scheme demonstrated superior performance to the existing IGM semi-explicit scheme, especially in those cases involving large perturbations from the planned normal mission characteristics. The ability to recover smoothly in real-time from large perturbations is due to the well-behaved nature of the OPGUID formulation and its avoidance of artificial approximations that limit the flexibility and accuracy of current guidance schemes.

The OPGUID scheme handled single-burn missions well, including relatively short coast arcs of fixed duration - as in the case of a multiple stage boost flight. General multiple burn problems, however, require long coast arcs (often about one half of an orbital period) and permit optimal choice of the lengths of both burns and coasts. A different algorithm was needed to deal efficiently with the very long coast arcs and to choose optimally the associated switching times.

A sophisticated multi-burn optimization program, SWITCH, has been developed under Contract NAS 8-21315, and has converged a variety of orbit transfer problems with an efficiency and reliability approaching that of OPGUID. Computing time per iteration is under one quarter second on a CDC 6600, and most missions require fewer than half a dozen iterations. However, it remains to be shown that the SWITCH algorithm possesses the speed and convergence properties required for real-time guidance. Such a demonstration would be a significant advance in the art since present guidance schemes cannot revise an entire multiburn trajectory in response to in-flight perturbations, but can only modify each single-burn arc separately. In addition to adapting the multiburn optimization program for real-time use, it is desirable to extend the boundary value formulations beyond that of the five constraint orbital constant and rendezvous missions that have already been solved.

Appendix: SWITCH Program

I. Users Guide

The MAIN routine reads the data necessary to specify the mission. The data is read as follows:

Card No.	Variable Names	FORMAT
1	NCASE	I2
2	UK, AMASS, BMAX, CEXV, HMAX	5F10.9
3	XOE	6F10.9
4	QOE	6F10.9
5	C	7F10.9
6	LEGMAX, IMAX	2I2
7	ATP(1,1), ATP(1,2)	2F10.9
8	ATP(2,1), ATP(2,2)	2F10.9
.	. .	.
.	. .	.
.	. .	.
8+LEGMAX	ATP(LEGMAX+1,1) ATP(LEGMAX+1,2)	2F10.9

NCASE appearing on card 1 tells the program how many separate data cases are to follow. Cards 2 – 8+LEGMAX completely define a case and must be repeated for each separate mission. Card 2 contains a description of the vehicle and the gravitation attraction of the earth. The variables are defined as follows: UK – gravitational constant (km^3/sec^2), AMASS – initial vehicle mass (kgm), BMAX – mass flow rate (kgm/sec), CEXV – exhaust velocity of the engines (km/sec) and HMAX – maximum allowable integration step size (sec). XOE is a six vector containing the initial position of the vehicle in its first three components and the initial velocity in its last three components. These vectors are specified in km and km/sec in an inertial geocentric cartesian coordinate system. QOE is a six vector containing the initial costate vector $(u_o, \dot{u}_o)^T$ with the magnitude of u_o assumed to be unity. The C vector for the basic five orbital constant mission contains the desired angular velocity vector $h = r \times v$ of the final orbit in its first three components and e_1 e_2 (first two components of the eccentricity vector of the final orbit $e = -\left[\frac{r}{|r|} + \frac{h x v}{\mu}\right]$ as its last two components. LEGMAX is an integer specifying the number of burn and coast arcs and IMAX is the maximum allowable number of iterations for one mission. The ATP array defines the duration of the coast and burn arcs. ATP(1,1) contains the starting time of the mission and ATP(1,2) is positive if the first arc is a coast and negative if the first arc is a burn. ATP(2,1) defines the time at which the first arc is completed and the second arc started. The last arc is always assumed to be a burn and therefore ATP(LEGMAX,2) should always be negative.

In order for the program to perform properly, the mass, mass rate and exhaust velocity should be such that the vehicle is able to carry out the required mission. If the mass is very large in comparison to the thrusting ability of the vehicle it will take very long duration burns to move the vehicle from its present orbit. If the mass rate is large and the usable mass small, the amount of possible burn time would be very small. A reasonableness check should be made before submitting any test case to the computer, for experience has shown that a very large percentage of unsatisfactory computer runs are due to gross errors in the input data which tender the missions essentially impossible.

II. Program Organization

The multi-burn optimal guidance package consists of nine subroutines SWITCH, COAST, AMULT, RKGO31, RKSTEP, YDDRHS, BVEVAL, ADJUST, and SOLVE. The entire package was written in FORTRAN IV and was tested on a CDC 6600 data processing system. The algorithm, not including the two routines MAIN and OUT used for inputting and outputting data, required approximately 7300 core locations and required at most .25 seconds for each iteration.

The SWITCH subroutine is the executive routine for the program. It accepts the input data from MAIN and performs the necessary initializations. SWITCH computes an entire trial trajectory for each iteration including computation of the necessary partial derivatives by means of one call on COAST or RKGO31 for each coasting or burning arc of the trajectory. At the end of each arc control is returned to SWITCH which then calculates the additional partial derivatives associated with the end of the arc as well as the difference between the desired condition at the end of the arc and the actual condition. When all arcs have been completed it calls the BVEVAL and ADJUST subroutines which calculate the allowable changes in the independent variables. At the end of the iteration SWITCH checks to see if a solution has been reached or if the total number of iterations equals the maximum that is allowable. If either of these conditions is met SWITCH returns control to MAIN and the mission is assumed completed.

The COAST routine propagates the trajectory along coast arcs. It accepts as inputs the present state and costate of the vehicle as well as the length of the proposed coasting arc. It calls AMULT to do the necessary matrix multiplications and outputs the state and costate at the end of the coasting arc as well as the partials of final state and costate with respect to initial state.

The burn subarcs are propagated by subroutines RKGO31, RKSTEP and YDDRHS. RKGO31 acts as the executive routine for burn subarcs. It initializes the arrays needed for the other routines, controls the integration step size and determines when the burn arc has been completed. When the final burn arc has been completed, the routine also determines the partials of state and costate with respect to the final time, T_F.

The RKSTEP routine performs a single Runge-Kutta fourth-order numerical integration step on the system of second-order matrix differential equations which describe the motion of the thrusting vehicle. The integration scheme requires three separation evaluations of the right hand side of the differential equation and these are performed by the YDDRHS routine.

When the entire trial trajectory has been calculated, the BVEVAL routine is called. This subroutine computes the difference DC between the desired end conditions C and their actual value as determined by XF and QF. It also computes the matrix G of partial derivatives of C with respect to XF and QF and the matrix E of partial derivatives of C with respect to QO and the switching times. The DC vector is then added as an additional column of the E matrix and the ADJUST routine is called. This routine immediately calls the SOLVE routine which performs a Gauss-Jordan reduction of the E matrix and determines the adjustment necessary, on a linearized basis, to null out the DC vector. In order to insure that the assumption of linearity is not violated, a quantity CK is calculated which limits the adjustments to be performed on the independent variables of the boundary value search. The ADJUST routine then calculates the new QO and switching times and returns control to SWITCH.

III. Variable Definitions

Program Name	Mathematical Symbol	Description
XOE(I)	x_o	Initial state vector
QOE(I)	q_o	Estimated initial costate vector
C(I)	C	Vector of desired end conditions
ATP(I, 1)	-	Switching times
ATP(I, 2)	-	Type of ARC (+ Coast, -Burn)

Program Name	Mathematical Symbol	Description
AM	M	Mass at end of each leg
AMASS	M	Initial mass
BMAX	$\dot{M}$	Mass flow rate
CEXV	V_{EX}	Exhaust velocity of the engines
HMAX	h	Maximum allowable integration step size.
LEGMAX	-	Number of burn and coast arcs
IMAX	-	Maximum number of iterations allowable
QO1 (I)	q_o	Initial costate vector, updated each iteration
UK	μ	Gravitation constant
QO(I)	q_o	Costate at start of arc
QF(I)	q_F	Costate at end of arc
XO(I)	x_o	State at start of arc
XF(I)	x_F	State at end of arc
E(I, J)	$\frac{\partial \quad C}{\partial q_o t_1 \cdots t_F}$	Partials of end conditions with respect to the independent variables
DC(I) (Also last column of E matrix)		Vector difference between desired end conditions and attained end conditions.
Z(I, J)	$\frac{\partial \quad x, q}{\partial q_o, t_1 \cdots t_f}$	Partial of present state and costate with respect to the independent variables
PHI(I, J)	$\frac{\partial x_F}{\partial x_o} = \frac{\partial q_f}{\partial q_o}$	Partial of state at end of coast with respect to state at start of coast

Program Name	Mathematical Symbol	Description
DPHI(I, J)	$\frac{\partial q_F}{\partial x_o}$	Partial of costate at end of coast with respect to state at start of coast
DXF(I)	-	Projected change in final state
DRF	-	Magnitude of projected change in position
DVF	-	Magnitude of projected change in velocity
CK	-	Fraction of the current Newton-Raphson adjustment accepted ($0 < CK \leq 1$)

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
     COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
      DIMENSION C(12),Q0E(6),X0E(6)
1     FORMAT(7F10.9)
     READ(5,3) NCASE
      NC=0
5     NC=NC+1
     WRITE(6,6) NC
6     FORMAT(1H1,13H CASE NUMBER ,I2,//)
      READ(5,1)UK,AMASS,BMAX,CEXV,HMAX
      READ(5,1)X0E
      READ(5,1)Q0E
      READ(5,1)(C(I),I=1,7)
     READ(5,3) LEGMAX,IMAX
3     FORMAT(2I2)
4     FORMAT(2F10.9)
      L1=LEGMAX+1
      READ(5,4) (ATP(I,1),ATP(I,2),I=1,L1)
      CALL SWITCH(AMASS,LEGMAX,IMAX,C,Q0E,X0E,HMAX)
      IF(NCASE.GT.NC) GO TO 5
      STOP
      END


     SUBROUTINE OUT(X,Q,XF,QF,LEGMAX,NPATH)
     COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
      COMMON/WADJ/A(8),DXF(6)
      COMMON/WCOAST/PSY,ALPHA,FT
      COMMON/WSWIT/E(12,13),DC(12),DRF,DVF,CK,EVT,KCOUNT,BURNT
      DIMENSION X(6),H(3),E1(3),TIME(50,7),DC1(50,12),DRF1(50),DVF1(50)
      DIMENSION KC1(50),Q(6),XF(6),QF(6),CK1(50)
      IF(NPATH)1,2,3
1     LEG1=LEG-1
      IF(LEG.LT.3) WRITE(6,40)KCOUNT
40    FORMAT(1H0,17HITERATION NUMBER ,I3,/)
      IF(LEG.EQ.1) GO TO 2
      WRITE(6,10)LEG1,X,Q
10    FORMAT(20X,9HCOAST ARC,/25X,4HLEG=,
    1I2,1X12HSTATE AT END,1X,6E14.6,/30X,14HCOSTATE AT END,1X,6E14.6)
      WRITE(6,22)PSY,ALPHA,FT
22   FORMAT(30X,4HPSY=,E14.6,5X,6HALPHA=,E14.6,5X,22HCALCULATED COAST T
    1IME=,F14.6)
2     H(1)=X(2)*X(6)-X(3)*X(5)
      H(2)=X(3)*X(4)-X(1)*X(6)
      H(3)=X(1)*X(5)-X(2)*X(4)
      HM=SQRT(H(1)**2+H(2)**2+H(3)**2)
      RM=SQRT(X(1)**2+X(2)**2+X(3)**2)
      E1(1)=-(X(1)/RM+(H(2)*X(6)-H(3)*X(5))/UK)
      E1(2)=-(X(2)/RM+(H(3)*X(4)-H(1)*X(6))/UK)
      E1(3)=-(X(3)/RM+(H(1)*X(5)-H(2)*X(4))/UK)
      EM=SQRT(E1(1)**2+E1(2)**2+E1(3)**2)
      ENERGY=-UK/RM+.5*(X(4)**2+X(5)**2+X(6)**2)
      AAXIS=-UK/(2.0*ENERGY).
      RMIN=AAXIS*(1.0-EM)
      RMAX=AAXIS*(1.0+EM)
      PERIOD=(6.2831853)*SQRT(ABS(AAXIS**3/UK))
      WRITE(6,11)AAXIS,RMIN,RMAX,ENERGY,PERIOD,HM,H,EM,E1,RM
11   FORMAT(25X,15HSEMIMAJOR AXIS=,E14.6,1X5HRMIN=,E14.6,1X5HRMAX=,E14
    1.6,1X7HENERGY=,E14.6,/25X,7HPERIOD=,E14.6,1X5HHMAG=,E14.6,1X8HH VE
    2CTOR,3E14.6,/25X5HEMAG=,E14.6,1X8HE VECTOR,3E14.6,1X5HRMAG=,E14.6)
```

```
      IF(NPATH)5,6,6
5      WRITE(6,12)LEG,XF,QF,AM
12     FORMAT(20X,8HBURN ARC,/25X,4HLEG=,I2,1X12HSTATE AT END,1X6E14.6,/
     130X,14HCOSTATE AT END,1X6E14.6,/30X,19HMASS AT END OF LEG=,E14.6)
       IF(LEG.LT.LEGMAX) RETURN
       KC=MOD(KCOUNT-1,50)
       KC=KC+1
       DO 7 I=1,LFG
7      TIME(KC,I)=ATP(I+1,1)-ATP(I,1)
       RETURN
 6     L6=6+LEG
       DETE=1.0
       DO 4 I=1,L6
4      DETE=DETE*E(I,I)
      TIME(KC,7)=BURNT
       WRITE(6,13)TIME(KC,7),(TIME(KC,I),I=1,LEG)
13    FORMAT(20X,16HTOTAL BURN TIME=,E14.6,1X9HARC TIMES,1X4E14.6,/61X
     12F14.6)
       WRITE(6,14)DC,DETE,(F(I,I),I=1,L6)
14    FORMAT(25X,2HDC,1X6E14.6,/28X,6F14.6,/25X,17HDETERMINANT OF E=,
     1E14.6,1X13HDIAGONAL OF E,4E14.6,/25X,8E14.6)
       L1=LEG+1
       L2=L1+1
       WRITE(6,15)DXF,DRF,DVF,CK,EVT,(A(I),I=1,L2)
15     FORMAT(25X,4HDXF ,6E14.6,/25X,4HDRF=,E14.6,5H DVF=,E14.6,4H CK=,
     1E14.6,5H EVT=,E14.6,/25X,10HCK=MIN OF ,7E13.5)
       L7=LEG+7
       WRITE(6,30)(E(I,L7),I=1,L6)
30    FORMAT(/25X66HCHANGE REQUISTED IN INITIAL COSTATE,SWITCHING TIMES
     1AND FINAL TIME,/25X,6F14.6,/35X,6E14.6)
      WRITE(6,16)KCOUNT,QF,(ATP(I,1),I=2,L1)
16    FORMAT(1X,24HEND OF ITERATION NUMBER ,I3,/15X,7HNEW Q0 ,6E14.6,/
     115X,16HNEW SWITCH TIMES,1X6E14.6,//)
       DO 8 I=1,12
8     DC1(KC,I)=DC(I)
       CK1(KC)=CK
       DRF1(KC)=DRF
       DVF1(KC)=DVF
       KC1(KC)=KCOUNT
       RETURN
3      IF(KCOUNT.GT.50) KCOUNT=50
       WRITF(6,17)
17     FORMAT(1H1,50X,14HSUMMARY TABLES,//,1X9HITERATION,1X15HTOTAL BURN
     1 TIME,21X,29HLENGTH OF BURN AND COAST ARCS,/2X6HNUMBER,//)
       DO 9 I=1,KCOUNT
9     WRITE(6,18) KC1(I),TIME(I,7),(TIME(I,J),J=1,LEG)
18    FORMAT(3X,I3,5X,7E15.7)
       WRITE(6,19)
19    FORMAT(1H0,//1X9HITERATION,39X,36HERROR IN BOUNDARY CONDITIONS-DC(
     11-8),/2X,6HNUMBER,//)
      WRITE(6,20)(KC1(I),(DC1(I,J),J=1,8),I=1,KCOUNT)
20     FORMAT(3XI3,5X,8E14.6)
       WRITE(6,21)
21    FORMAT(1H0,//1X9HITERATION,26X,8HDC(9-12),32X,3HDRF,12X,3HDVF,12X2
     1HCK,/2X6HNUMBER,//)
      WRITE(6,18)(KC1(I),(DC1(I,J),J=9,12),DRF1(I),DVF1(I),CK1(I),I=1,
     1KCOUNT)
       RETURN
       END
      SUBROUTINE SWITCH(AMASS,LEGMAX,IMAX,C,Q0E,X0E,HMAX)
      COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
       COMMON/WSWIT/E(12,13),DC(12),DRF,DVF,CK,EVT,KCOUNT,BURNT
      DIMENSION C(12),X0(6),Q0(6),XF(6),QF(6),Z(12,12),X0E(6),Q01(6)
      DIMENSION Q0E(6),DUMMY(13),PHI(6,6),DPHI(6,6),DZ(12,12)
      KCOUNT=0
       NO=0
       L7=7+LEGMAX
        L6=6+LEGMAX
       EVT=1.E-8
       KMAX=IMAX
       L1=LEGMAX+1
       DO 2 I=1,6
2      Q01(I)=Q0E(I)
       WRITE(6,100)UK,AMASS,BMAX,CEXV,HMAX
100   FORMAT(1H0,1X,3HUK=,F10.2,1X,13HINITIAL MASS=,E16.8,1X,10HMASS RAT
     1E=,E16.8,1X,17HEXHAUST VELOCITY=,E16.8,6H HMAX=F10.3,//)
       WRITE(6,121) IMAX,LEGMAX
121   FORMAT(6H0IMAX=,I3,5X,7HLEGMAX=,I3,/)
       WRITE(6,101)(ATP(I,1),ATP(I,2),I=1,L1)
101   FORMAT(1H0,20X,4HTIME,20X,27HTYPE OF ARC(+ COAST,- BURN)/(20X,
     1E16.8,15X,E16.8))
       WRITE(6,102)X0E,Q01,(C(I),I=1,7)
102   FORMAT(17H0INITIAL STATE X0,6F17.7/16H    ESTIMATED Q0,6F17.7/
     116H0DESIRED FINAL C,7E16.8/)
1      LEG=1
       AM=AMASS
       KCOUNT=KCOUNT+1
       BURNT=0.0
       DO 3 I=1,6
       Q0(I)=Q01(I)
3      X0(I)=X0E(I)
       DO 4 I=1,12
       E(I,13)=0.0
       DO 4 J=1,12
       E(I,J)=0.0
4      Z(I,J)=0.0
       DO 5 I=1,6
5      Z(I+6,I)=1.0
       IF(ATP(1,2))6,7,7
7      CALL COAST(X0,Q0,XF,QF,PHI,DPHI,UK,LEG,ATP,NO)
       LEG6=6+LEG
       LEG5=LEG+5
       DO 30 I=1,6
       DO 30 J=1,LEG5
       DZ(I,J)=0.0
       DZ(I+6,J)=0.0
       DO 30 K=1,6
      DZ(I,J)=DZ(I,J)+PHI(I,K)*Z(K,J)
30    DZ(I+6,J)=DZ(I+6,J)+PHI(I,K)*Z(K+6,J)+DPHI(I,K)*Z(K,J)
       DO 31 I=1,12
       DO 31 J=1,LEG5
31     Z(I,J)=DZ(I,J)
       UM=SQRT(QF(1)**2+QF(2)**2+QF(3)**2)
       CBMU=(CEXV*BMAX)/(AM*UM)
       DO 8 I=1,3
 8     Z(I+3,6+LEG)=-CBMU*QF(I)
       IF(LEG.LE.1) GO TO 12
       E(6+LEG,7+LEGMAX)=UMP-UM
      DO 9 J=1,LEG6
      E(LEG6,J)=-E(LEG6,J)
      DO 9 K=1,3
9     E(LEG6,J)=E(LEG6,J)+(QF(K)/UM)*Z(K+6,J)
      DO 11 I=1,6
      Q0(I)=QF(I)
 11   X0(I)=XF(I)
      LEG=LEG+1
6     CALL RKG031(X0,Q0,XF,QF,Z,EVT,HMAX,LEGMAX,NO)
      BURNT=BURNT+ATP(LEG+1,1)-ATP(LEG,1)
      CALL OUT(X0,Q0,XF,QF,LEGMAX,-1)
      LEG6=6+LEG
      IF(LEG.GE.LEGMAX) GO TO 18
13    UM=SQRT(QF(1)**2+QF(2)**2+QF(3)**2)
      UMP=UM
      CBMU=(CEXV*BMAX)/(AM*UM)
      DO 14 I=1,3
 14   Z(I+3,6+LEG)=CBMU*QF(I)
      DO 15 J=1,LEG6
      DO 15 K=1,3
 15   E(7+LEG,J)=E(7+LEG,J)+(QF(K)/UM)*Z(K+6,J)
 12   R2=XF(1)*XF(1)+XF(2)*XF(2)+XF(3)*XF(3)
      RS=XF(1)*QF(1)+XF(2)*QF(2)+XF(3)*QF(3)
      C3=-UK/(R2*SQRT(R2))
      C4=-3.0*C3*RS/R2
     E(6+LEG,7+LEGMAX)=XF(4)*QF(4)+XF(5)*QF(5)+XF(6)*QF(6)-C3*RS
      DO 16 I=1,3
      DUMMY(I)=-C3*QF(I)-C4*XF(I)
      DUMMY(I+3)=QF(I+3)
      DUMMY(I+6)=-C3*XF(I)
 16   DUMMY(I+9)=XF(I+3)
      DO 17 J=1,LEG6
      DO 17 K=1,12
17    E(6+LEG,J)=-DUMMY(K)*Z(K,J)+E(6+LEG,J)
      DO 19 I=1,6
      Q0(I)=QF(I)
 19    X0(I)=XF(I)
      LEG=LEG+1
      IF(ATP(LEG,2).GT.0) GO TO 7
      F(7,7)=0.0
      GO TO 6
 18   CALL BVEVAL(XF,QF,Z,C,F,DC)
      DO 20 I=1,6
      I6=I+6
      E(I,7+LEGMAX)=DC(I)
      DC(I6)=E(I6,LEGMAX+7)
20    E(6+LEGMAX,I)=Q01(I)
      CALL ADJUST(Q01,E,Z,DRF,DVF,CK,LEGMAX,ATP)
      EVT=AMAX1(1.E-14,1.E-8*DVF**2)
      EVT=AMIN1(EVT,1.E-8)
      CALL OUT(XF,QF,X0,Q01,LEGMAX,0)
      IF(KCOUNT.GE.KMAX) GO TO 79
      IF(DRF.GE..1.OR.DVF.GE..005) GO TO 1
79    CALL OUT(X0,Q0E,XF,QF,LEGMAX,1)
      RETURN
      END


      SUBROUTINE COAST(X0,Q0,XF,QF,PHI,DPHI,UK,LEG,ATP,NO)
      COMMON/WCOAST/PSY,ALPHA,FT
      COMMON/CCOAST/ANN(2,2),BNN(3,3),XX0(6,6),R0(3),V0(3),R(3),V(3)
     1,DR0(3),DV0(3),DR(3),DV(3),DAN(2,2),DBNN(3,3),DXX0(6,6)
      COMPLEX CALPH,CAPSY
      DIMENSION X0(6),Q0(6),XF(6),QF(6),PHI(6,6),DPHI(6,6),ATP(7,2)
      DIMENSION H0(3)
      LEG5=5+LEG
C       TIME OF COAST
      T=ATP(LEG+1,1)-ATP(LEG,1)
C       Q0   (Q AT START OF COAST)
      DO 40 I=1,3
      I3=I+3
      R0(I)=X0(I)
      V0(I)=X0(I+3)
      DR0(I)=Q0(I)
 40   DV0(I)=Q0(I3)
C       R0,V0  STATE AT START OF COAST
      JUMP=0
      RM0=SQRT(R0(1)*R0(1)+R0(2)* R0(2)+R0(3)*R0(3))
      DRM0=(R0(1)*DR0(1)+R0(2)*DR0(2)+R0(3)*DR0(3))/RM0
      SIG0=R0(1)*V0(1)+R0(2)*V0(2)+R0(3)*V0(3)
      DSIG0=(V0(1)*DR0(1)+V0(2)*DR0(2)+V0(3)*DR0(3)+R0(1)*DV0(1)
     1+R0(2)*DV0(2)+R0(3)*DV0(3))
      ALPHA=V0(1)*V0(1)+V0(2)*V0(2)+V0(3)*V0(3)-2.*UK/RM0
      H0(1)=R0(2)*V0(3)-R0(3)*V0(2)
      H0(2)=R0(3)*V0(1)-R0(1)*V0(3)
      H0(3)=R0(1)*V0(2)-R0(2)*V0(1)
      P0=(H0(1)*H0(1)+H0(2)*H0(2)+H0(3)*H0(3))/UK
 80   PSY=T/P0
      IF(ALPHA)81,82,82
 81   CALPH=CMPLX(SQRT(-ALPHA),0.)
      GO TO 83
 82    CALPH=CMPLX(0.,SQRT(ALPHA))
 83   CAPSY=PSY*CALPH
      S0=REAL(CCOS(CAPSY))
      S1=REAL(CSIN(CAPSY)/CALPH)
      S2=(S0-1.0)/ALPHA
      S3=(S1-PSY)/ALPHA
      FT=RM0*S1+SIG0*S2+UK*S3
      RM=RM0*S0+SIG0*S1+UK*S2
      IF(JUMP.EQ.1) GO TO 84
      PSY=PSY+(T-FT)/RM
      IF(ABS(T-FT).GE..0001) GO TO 83
      JUMP=1
      GO TO 83
 84   FM1=-UK*S2/RM0
      F=1.0+FM1
      FD=-UK*S1/(RM*RM0)
      G=FT-UK*S3
      GDM1=-UK*S2/RM
      GD=1.0+GDM1
      UKR3=UK/(RM*RM*RM)
      UKR03=UK/(RM0*RM0*RM0)
      DALPH=2.0*(V0(1)*DV0(1)+V0(2)*DV0(2)+V0(3)*DV0(3)+UKR03*(
     1R0(1)*DR0(1)+R0(2)*DR0(2)+R0(3)*DR0(3)))
      DAPA=DALPH/ALPHA
      DAPA2=DAPA/ALPHA
      DPSY=-(DRM0*S1+DSIG0*S2+RM0*(PSY*S0-S1)*DAPA*.5+SIG0*(PSY*S1*.5-
     1S2)*DAPA+UK*(PSY-1.5*S1+PSY*S0*.5)*DAPA2)/RM
      DS0=(ALPHA*DPSY+.5*PSY*DALPH)*S1
      DS1=S0*DPSY+(PSY*S0-S1)*DAPA*.5
```

```
       DS2=S1*DPSY+(.5*PSY*S1-S2)*DAPA
      DS3=S2*DPSY+(PSY-1.5*S1+.5*PSY*S0)*DAPA2
      S4=(S2-PSY*PSY*.5)/ALPHA
      DS4=S3*DPSY+(PSY*PSY*.5-2.0*S2+.5*PSY*S1)*DAPA2
      S5=(S3-PSY*PSY*PSY/6.0)/ALPHA
      DS5=S4*DPSY+(PSY*PSY*PSY/6.0+(2.0*PSY-2.5*S1+.5*PSY*S0)/ALPHA)
     1*DAPA2
      U=S2*FT+UK*(PSY*S4-3.0*S5)
       DU=DS2*FT+UK*(DPSY*S4+PSY*DS4-3.0*DS5)
      DRM=S0*DRM0+DS0*RM0+S1*DSIG0+DS1*SIG0+UK*DS2
       DF=(-UK*DS2-FM1*DRM0)/RM0
       DG=-UK*DS3
       DGD=(-UK*DS2-GDM1*DRM)/RM
        R01=RM0*RM
      DR01=RM*DRM0+DRM*RM0
      DFD=(-UK*DS1-FD*DR01)/R01
       DO 4 I=1,3
       DR(I)=R0(I)*DF+V0(I)*DG+DR0(I)*F+DV0(I)*G
       QF(I)=DR(I)
       R(I)=R0(I)*F+V0(I)*G
      DV(I)=R0(I)*DFD+V0(I)*DGD+DR0(I)*FD+DV0(I)*GD
       QF(I+3)=DV(I)
 4     V(I)=R0(I)*FD+V0(I)*GD
C        R,V  STATE AT END OF COAST
       IF(NO.EQ.1) GO TO 90
       DUKR3=-3.0*UKR3*DRM/RM
       DUR03=-3.0*UKR03*DRM0/RM0
      S1R0=S1/RM0
      DS1R0=(DS1-S1R0*DRM0)/RM0
        S1R=S1/RM
       DS1R=(DS1-S1R*DRM)/RM
       R02=1.0/(RM0*RM0)
       R2=1.0/(RM*RM)
       UUK03=-U*UKR03
       DUUK3=-DU*UKR03-U*DUR03
      ANN(1,1)=-FD*S1R0-FM1*R02
       ANN(1,2)=-FD*S2
      ANN(2,1)=FM1*S1R0+UUK03
       ANN(2,2)=FM1*S2
      DUMM1=ANN(1,1)
       DAN(1,1)=-FD*DS1R0-DFD*S1R0+UKR03*DS2+DUR03*S2
       DUMMY=DAN(1,1)
       DAN(1,2)=-DFD*S2-FD*DS2
       DAN(2,1)=FM1*DS1R0+DF*S1R0+DUUK3
       DAN(2,2)=FM1*DS2+DF*S2
       CALL AMULT
      DO 5 I=1,3
       DO 5 J=1,3
       DXX0(I,J)=DBNN(I,J)
 5    XX0(I,J)=BNN(I,J)
       DO 6 I=1,3
       DXX0(I,I)=DXX0(I,I)+DF
 6     XX0(I,I)=XX0(I,I)+F
       ANN(1,1)=ANN(1,2)
       ANN(2,1)=ANN(2,2)
       ANN(1,2)=-GDM1*S2
       ANN(2,2)=G*S2-U
       DAN(1,1)=DAN(1,2)
       DAN(2,1)=DAN(2,2)
       DAN(1,2)=-GDM1*DS2-DGD*S2
      DAN(2,2)=-DU+DG*S2+G*DS2
      CALL AMULT
      DO 7 I=1,3
      DO 7 J=1,3
      J3=J+3
      DXXO(I,J3)=DBNN(I,J)
 7    XXO(I,J3)=BNN(I,J)
      DO 8 I=1,3
      I3=I+3
      DXXO(I,I3)=DXXO(I,I3)+DG
 8    XXO(I,I3)=XXO(I,I3)+G
      ANN(2,1)=-ANN(1,1)
      ANN(2,2)=-ANN(1,2)
      ANN(1,1)=-FD*S1R-GDM1*R2
      ANN(1,2)=U*UKR3-GDM1*S1R
      DAN(2,1)=-DAN(1,1)
      DAN(2,2)=-DAN(1,2)
      DAN(1,1)=-DFD*S1R-FD*DS1R+UKR3*DS2+DUKR3*S2
      DAN(1,2)=-GDM1*DS1R-DGD*S1R+DU*UKR3+U*DUKR3
      CALL AMULT
      DO 9 I=1,3
      DO 9 J=1,3
      I3=I+3
      J3=J+3
      DXXO(I3,J3)=DBNN(I,J)
 9    XXO(I3,J3)=BNN(I,J)
      DO 10 I=4,6
      DXXO(I,I)=DXXO(I,I)+DGD
 10  XXO(I,I)=XXO(I,I)+GD
      ANN(1,2)=ANN(1,1)
      ANN(2,2)=ANN(2,1)
      ANN(2,1)=-DUMM1
     ANN(1,1)=-FD*(SO/R01+R2+R02)-UUK03*UKR3
      DAN(1,2)=DAN(1,1)
      DAN(2,2)=DAN(2,1)
      DAN(2,1)=-DUMMY
     DAN(1,1)=-UUK03*DUKR3-DUUK3*UKR3-DFD*(SO/R01+R2+R02)-FD*((DSO-SO*
    1DR01/R01)/R01-2.0*(R2*DRM/RM+R02*DRM0/RM0))
      CALL AMULT
      DO 11 I=1,3
      DO 11 J=1,3
       I3=I+3
      DXXO(I3,J)=DBNN(I,J)
 11   XXO(I3,J)=BNN(I,J)
      DO 12 I=1,3
      I3=I+3
     DXXO(I3,I)=DXXO(I3,I)+DFD
 12    XXO(I3,I)=XXO(I3,I)+FD
C       XXO PARTIAL OF XF WITH RESPECT TO XO
C      DXXO PARTIAL OF QF WITH RESPECT TO XO
      DO 50 I=1,6
      DO 50 J=1,6
      PHI(I,J)=XXO(I,J)
 50   DPHI(I,J)=DXXO(I,J)
 90   DO 52 I=1,3
      XF(I)=R(I)
  52  XF(I+3)=V(I)
      RETURN
      END
      SUBROUTINE AMULT
      COMMON/CCOAST/ANN(2,2),BNN(3,3),XXO(6,6),RO(3),VO(3),R(3),V(3)
     1,DRO(3),DVO(3),DR(3),DV(3),DAN(2,2),DBNN(3,3),DXXO(6,6)
      DIMENSION DA(3,2)
     DIMENSION A(3,2)
      DO 1 I=1,3
      DO 1 J=1,2
     DA(I,J)=DR(I)*ANN(1,J)+DV(I)*ANN(2,J)+R(I)*DAN(1,J)+V(I)*DAN(2,J)
1    A(I,J)=ANN(1,J)*R(I)+ANN(2,J)*V(I)
     DO 2 I=1,3
      DO 2 J=1,3
     DBNN(I,J)=A(I,1)*DRO(J)+A(I,2)*DVO(J)+DA(I,1)*RO(J)+DA(I,2)*VO(J)
2    BNN(I,J)=A(I,1)*RO(J)+A(I,2)*VO(J)
      RETURN
      END


     SUBROUTINE RKG031(X0,Q0,XF,QF,Z,EVT,HMAX,LEGMAX,NO)
     COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
     DIMENSION X0(6),Q0(6),XF(6),QF(6),Z(12,12),YN(6,13),YDN(6,13),
    1Y3H(6,13),YD3H(6,13),YM(6,13),YDM(6,13),EY(6),EYD(6)
      LEG6=LEG+6
     DO 1 I=1,3
      YN(I,1)=X0(I)
      YN(I+3,1)=Q0(I)
      YDN(I,1)=X0(I+3)
     YDN(I+3,1)=Q0(I+3)
     DO 1 J=2,LEG6
      YN(I,J)=Z(I,J-1)
      YN(I+3,J)=Z(I+6,J-1)
      YDN(I,J)=Z(I+3,J-1)
1     YDN(I+3,J)=Z(I+9,J-1)
      TF=ATP(LEG+1,1)
      TO=ATP(LEG,1)
     H=SIGN(HMAX,TF-TO)
     TN=TO
     GO TO 7
2     CALL RKSTEP(YN,YDN,TN,Y3H,YD3H,H,1,LEG)
      CALL RKSTEP(YN,YDN,TN,YM,YDM,H/3.,0,LEG)
      CALL RKSTEP(YM,YDM,TN+(H/3.),YM,YDM,H/3.,0,LEG)
      CALL RKSTEP(YM,YDM,TN+(H/3.)*2.,YM,YDM,H/3.,0,LEG)
     DO 3 I=1,6
     EY(I)=.125E-1*(YM(I,1)-Y3H(I,1))
   3 EYD(I)=.125E-1*(YDM(I,1)-YD3H(I,1))
     EV2MIN=1.E-13*(YDM(1,1)**2+YDM(2,1)**2+YDM(3,1)**2)
      EVL2=AMAX1(EVT,EV2MIN)
      R=(EYD(1)**2+EYD(2)**2+EYD(3)**2)/EVL2
   4 DO 5 I=1,6
     YN(I,1)=YM(I,1)+EY(I)
   5 YDN(I,1)=YDM(I,1)+EYD(I)
     DO 6 I=1,6
      DO 6 J=2,LEG6
     YN(I,J)=Y3H(I,J)
   6 YDN(I,J)=YD3H(I,J)
     IF( ABS(H) .GE. ABS(TF-TN) ) GO TO 8
     TN=TN+H
     IF(R.LT. 0.04) R=.04
     H=H/R**.125
   7 IF ( ABS(H) .GT. ABS(TF-TN)  ) H=TF-TN
```

```
      IF( ABS(H) .GT. HMAX) H=SIGN(HMAX,H)
      GO TO 2
 8     IF(LEG.LT.LEGMAX) GO TO 10
      CALL YDDRHS(YN,YD3H,1.,ATP(LEG+1,1),0)
       DO 11 I=1,3
       Z(I,6+LEGMAX)=YDN(I,1)
       Z(I+3,6+LEGMAX)=YD3H(I,1)
       Z(I+6,6+LEGMAX)=YDN(I+3,1)
 11    Z(I+9,6+LEGMAX)=YD3H(I+3,1)
 10    DO 12 I=1,3
       XF(I)=YN(I,1)
       XF(I+3)=YDN(I,1)
       QF(I)=YN(I+3,1)
       QF(I+3)=YDN(I+3,1)
       DO 12 J=2,LEG6
       Z(I,J-1)=YN(I,J)
       Z(I+3,J-1)=YDN(I,J)
       Z(I+6,J-1)=YN(I+3,J)
 12    Z(I+9,J-1)=YDN(I+3,J)
       AM=AM-BMAX*(ATP(LEG+1,1)-ATP(LEG,1))
      RETURN
      END


      SUBROUTINE RKSTEP(YN,YDN,TN,YN1,YDN1,H,N,LEG)
C  THIS PROGRAM ADVANCES YN AND YDN BY A STEP OF SIZE H TO YN1 AND YDN1
C  USING A FOURTH-ORDER RUNGE-KUTTA NUMERICAL INTEGRATION SCHEME. IF N
C  IS POSITIVE, ALL ELEMENTS OF THE MATRICES YN AND YDN ARE ADVANCED.
C  OTHERWISE ONLY THE FIRST COLUMN OF EACH MATRIX IS UPDATED.
      DIMENSION YN(6,13),YDN(6,13),D1(6,13),D2(6,13),D3(6,13),Y(6,13)
     1,YN1(6,13),YDN1(6,13)
      JMAX=1
       IF(N.GT.0) JMAX=6+LEG
      H2=.5*H
      CALL YDDRHS(YN,D1,H,TN,N)
      DO 1 J=1,JMAX
      DO 1 I=1,6
    1 Y(I,J)=YN(I,J)+H2*(YDN(I,J)+.25*D1(I,J))
      CALL YDDRHS(Y,D2,H,TN+H2,N)
      DO 2 J=1,JMAX
      DO 2 I=1,6
    2 Y(I,J)=YN(I,J)+H*(YDN(I,J)+.5*D2(I,J))
      CALL YDDRHS(Y,D3,H,TN+H,N)
      DO 3 J=1,JMAX
      DO 3 I=1,6
      YN1(I,J)=YN(I,J)+H*(YDN(I,J)+.16666667*(D1(I,J)+2.*D2(I,J)))
    3 YDN1(I,J)=YDN(I,J)+.16666667*(D1(I,J)+4.*D2(I,J)+D3(I,J))
      RETURN
      END


      SUBROUTINE YDDRHS(Y,YDD,H,T,N)
      COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
      DIMENSION Y(6,13),YDD(6,13),R(3),U(3),B(6,6)
C  COMPUTE BASIC QUANTITIES COMMON TO MANY COMPONENTS OF YDD.
       LEG6=6+LEG
      DO 1 I=1,3
      R(I)=Y(I,1)
    1 U(I)=Y(I+3,1)
      AM1=AM-(T-ATP(LEG,1))*BMAX
      R2=1./(R(1)*R(1)+R(2)*R(2)+R(3)*R(3))
      U2=1./(U(1)**2+U(2)**2+U(3)**2)
      RU=R(1)*U(1)+R(2)*U(2)+R(3)*U(3)
      ALPHA=-H*UK*R2*SQRT(R2)
       BETA=H*SQRT(U2)*CEXV*BMAX/AM1
      GAMMA=-3.*ALPHA*R2*RU
C   COMPUTE RDD AND UDD.
      DO 2 I=1,3
      YDD(I,1)=R(I)*ALPHA+U(I)*BETA
    2 YDD(I+3,1)=R(I)*GAMMA+U(I)*ALPHA
C   DECIDE WHETHER WDD IS REQUIRED AT THIS TIME.
   21 IF(N.LE.0) RETURN
C   COMPUTE ADDITIONAL QUANTITIES COMMON TO MANY COMPONENTS OF WDD.
      DELTA=-3.*ALPHA*R2
      EPSIL=-BETA*U2
      ZETA=-5.*GAMMA*R2
C   COMPUTE THE MATRIX B NEEDED IN THE MATRIX EQUATION  WDD=B*W.
      DO 3 J=1,3
      RRJ=DELTA*R(J)
      RUJ=EPSIL*U(J)
      URJ=ZETA*R(J)+DELTA*U(J)
      UUJ=RRJ
      DO 3 I=1,3
      Q=0.
      IF (I.EQ. J) Q=1.
      B(I,J)=R(I)*RRJ+Q*ALPHA
      B(I,J+3)=U(I)*RUJ+Q*BETA
      B(I+3,J)=R(I)*URJ+U(I)*UUJ+Q*GAMMA
    3 B(I+3,J+3)=B(I,J)
C   PERFORM THE MATRIX MULTIPLICATION B*W TO GET WDD.
      DO 5 I=1,6
       DO 5 J=2,LEG6
      SUM=0.
      DO 4 K=1,6
    4 SUM=SUM+B(I,K)*Y(K,J)
    5 YDD(I,J)=SUM
      IF(LEG.LE.1) RETURN
       RDDMB=-BETA*BMAX/AM1
       DO 6 J=8,LEG6
       DO 6 I=1,3
       J1=J-7
 6     YDD(I,J)=YDD(I,J)-SIGN(RDDMB,ATP(J1,2))*U(I)
      RETURN
      END


       SUBROUTINE BVEVAL(XF,QF,Z,C,E,DC)
C
C   THIS IS VERSION A OF SUBPROGRAM BVEVAL, DECK NAME 'OPDK3A'.
C   THIS VERSION DEALS WITH A FIVE-CONSTRAINT RIGHT-END BOUNDARY-VALUE
C   PROBLEM WHERE THE FIVE CONSTRAINED FUNCTIONS ARE THE THREE COMPON-
C   ENTS OF THE ORBITAL ANGULAR VELOCITY VECTOR (R CROSS V) AND THE FIRST
C   TWO COMPONENTS OF THE VECTOR WHOSE DIRECTION IS THE DIRECTION OF
C   PERICENTER AND WHOSE MAGNITUDE IS THE ORBITAL ECCENTRICITY.  THUS, IN
C   EFFECT, ALL OF THE SIX CLASSICAL ORBITAL ELEMENTS ARE CONSTRAINED
C   EXCEPT THE MEAN ANOMALY, WHICH IS FREE.
C
      COMMON/CCPINJ/UK,LEG,ATP(7,2),AM,BMAX,CEXV
       DIMENSION XF(6),C(12),DC(12),G(7,6),Z(12,12),E(12,13),R(3),V(3)
     1,QF(6),DUMMY(12)
```

```
      LEG6=6+LEG
      DO 1 I=1,3
      R(I)=XF(I)
    1 V(I)=XF(I+3)
      R2=R(1)**2+R(2)**2+R(3)**2
      G(1,2)=V(3)
      G(1,3)=-V(2)
      G(2,3)=V(1)
      G(1,5)=-R(3)
      G(1,6)=R(2)
      G(2,6)=-R(1)
      RM=SQRT(R2)
      R3=RM*R2
       C1=-1.0/RM+(V(1)**2+V(2)**2+V(3)**2)/UK
       C2=-(R(1)*V(1)+R(2)*V(2)+R(3)*V(3))/UK
      DO 4 I=1,3
      DO 3 J=1,3
      IF(I.LE.J) GO TO 2
      G(I,J)=-G(J,I)
      G(I,J+3)=-G(J,I+3)
    2 G(I+3,J)=R(I)*R(J)/R3-V(I)*V(J)/UK
    3 G(I+3,J+3)=(R(I)*V(J)*2.-V(I)*R(J))/UK
      G(I,I)=0.
      G(I,I+3)=0.
      G(I+3,I)=G(I+3,I)+C1
    4 G(I+3,I+3)=G(I+3,I+3)+C2
      DO 5 I=1,3
      DC(I)=0.
      DO 5 J=1,3
    5 DC(I)=DC(I)+G(I,J)*R(J)
       DO 8 I=1,2
      SUM=0.
      DO 7 J=1,3
    7 SUM=SUM+G(I,J)*DC(J)
    8 DC(I+3)=C(I+3)+R(I)/RM+SUM/UK
      DO 9 I=1,3
9      DC(I)=C(I)-DC(I)
       DO 10 I=1,5
       DO 10 J=1,LEG6
      E(I,J)=0.
      DO 10 K=1,6
   10 E(I,J)=E(I,J)+G(I,K)*Z(K,J)
       RS=XF(1)*QF(1)+XF(2)*QF(2)+XF(3)*QF(3)
       C3=-UK/R3
       C4=-3.0*C3*RS/R2
       DC(6)=           XF(4)*QF(4)+XF(5)*QF(5)+XF(6)*QF(6)-C3*RS
       DO 16 I=1,3
       DUMMY(I)=-C3*QF(I)-C4*XF(I)
       DUMMY(I+3)=QF(I+3)
       DUMMY(I+6)=-C3*XF(I)
 16    DUMMY(I+9)=XF(I+3)
       DO 17 J=1,LEG6
       DO 17 K=1,12
 17   E(6,J)=-DUMMY(K)*Z(K,J)+E(6,J)
      RETURN
      END


       SUBROUTINE ADJUST(Q0,E,Z,DRF,DVF,CK,LEGMAX,ATP)
       COMMON/WADJ/A(8),DXF(6)
       DIMENSION Q0(6),E(12,13),Z(12,12),ATP(7,2)
       CALL SOLVE(E,LEGMAX)
       L6=6+LEGMAX
       L7=LEGMAX+7
      DO 5 I=1,6
      DXF(I)=0.
       DO 5 K=1,L6
5      DXF(I)=DXF(I)+Z(I,K)*E(K,7+LEGMAX)
      DRF=SQRT(DXF(1)**2+DXF(2)**2+DXF(3)**2)
      DVF=SQRT(DXF(4)**2+DXF(5)**2+DXF(6)**2)
       DU2=SQRT(E(1,7+LEGMAX)**2+E(2,7+LEGMAX)**2+E(3,7+LEGMAX)**2)
      DUD2=SQRT(E(4,7+LEGMAX)**2+E(5,7+LEGMAX)**2+E(6,7+LEGMAX)**2)
       A(1)=.2/DU2
       A(2)=.0003/DUD2
       A(3)=1.0
      IF(ATP(1,2).LT.0) A(3)=(.5*(ATP(2,1)-ATP(1,1))/ABS(E(7,L7)))
       CK=AMIN1(1.0,A(1),A(2),A(3))
       DO 8 I=2,LEGMAX
       I2=I+2
       A(I2)=(.5*(ATP(I+1,1)-ATP(I,1))/ABS(E(I+6,L7)-E(I+5,L7)))
 8    CK=AMIN1(CK,A(I2))
       DO 6 I=1,6
       ATP(I+1,1)=ATP(I+1,1)+CK*E(I+6,7+LEGMAX)
 6     Q0(I)=Q0(I)+CK*E(I,7+LEGMAX)
      UM=SQRT(Q0(1)**2+Q0(2)**2+Q0(3)**2)
       DO 7 I=1,6
7      Q0(I)=Q0(I)/UM
      RETURN
      END


       SUBROUTINE SOLVE(A,LEGMAX)
       DIMENSION A(12,13)
       L6=6+LEGMAX
       L7=7+LEGMAX
       DO 6 N=1,L6
      IBIG=N
       DO 1 I=N,L6
      IF( ABS( A(I,N) ).GT. ABS(A(IBIG,N)))  IBIG=I
    1 CONTINUE
      IF (IBIG.EQ.N) GO TO 3
       DO 2 J=N,L7
      Q=A(N,J)
      A(N,J)=A(IBIG,J)
    2 A(IBIG,J)=Q
   3   DO 5 I=1,L6
      IF(I.EQ.N) GO TO 5
      Q=A(I,N)/A(N,N)
      M=N+1
       DO 4 K=M,L7
    4 A(I,K)=A(I,K)-Q*A(N,K)
    5 CONTINUE
    6 CONTINUE
       DO 7 I=1,L6
   7   A(I,L7)=A(I,L7)/A(I,I)
      RETURN
      END


    DATA FOR CASE 1
 1
398601.56 12644651. 22384.40794.1540729 100.
5087.493  4132.6342 114.70552 4.900796  -6.0577962.0669144
.4668414  -.7801086 -.1528746 -.5666 E-3-.6626 E-3-.1548 E-3
 1674.843496091.504 -87016.17 0.0       0.0
 450
1034.      1.0
1434.      -1.0
1689.8176 1.0
20417.2732-1.0
20546.38321.0
```

```
CASE NUMBER  1


UK= 398601.56 INITIAL MASS=  1.26446510E+07 MASS RATE=  2.23844079E+04 EXHAUST VELOCITY=  4.15407290E+00 HMAX=    100.000


IMAX= 50     LEGMAX=  4


                     TIME                      TYPE OF ARC(+ COAST,- BURN)
                      1.03400000E+03                   1.00000000E+00
                      1.43400000E+03                  -1.00000000E+00
                      1.68981760E+03                   1.00000000E+00
                      2.04172732E+04                  -1.00000000E+00
                      2.05463832E+04                   1.00000000E+00

INITIAL STATE X0      5087.4930000      4132.6342000      114.7055200        4.9007960         -6.0577962          .0669144
  ESTIMATED Q0          .4668414         -.7801086         -.1528746         -.0005666         -.0006624         -.0001848

DESIRED FINAL C -1.67484340E+03  9.60915040E+04 -8.70161700E+04  0.              0.             -0.             -0.


ITERATION NUMBER    1

                     COAST ARC
                         LEG= 1 STATE AT END    6.408976E+03  1.340597E+03  1.277190E+02  1.579651E+00 -7.638430E+00 -3.115995E-03
                            COSTATE AT END    1.678600E-01 -9.661125E-01 -1.961118E-01 -8.810358E-04 -2.277036E-04 -5.677916E-05
                            PSY=  6.104706E-02      ALPHA= -6.088936E+01      CALCULATED COAST TIME=  4.000000E+02
                         SEMIMAJOR AXIS=  6.546326E+03 RMIN=  6.531170E+03 RMAX=  6.561482E+03 ENERGY= -3.044468E+01
                         PERIOD=  5.271167E+03 HMAG=  5.108191E+04 H VECTOR  9.713954E+02  2.217218E+02 -5.107219E+04
                         EMAG=  2.315202E-03 E VECTOR  7.232001E-05 -2.314056E-03 -8.670576E-06 RMAG=  6.548930E+03
                     BURN ARC
                         LEG= 2 STATE AT END    6.535795E+03 -9.242956E+02  6.411060E+01 -6.737298E-01 -1.018129E+01 -5.422821E-01
                            COSTATE AT END  -6.291197E-02 -9.793821E-01 -2.014162E-01 -8.896451E-04  1.235212E-04  1.567432E-05
                            MASS AT END OF LEG=  6.918325E+06
                     COAST ARC
                         LEG= 3 STATE AT END  -4.213018E+04 -1.570124E+03 -8.056517E+02 -7.446084E-02  1.591464E+00  8.167224E-02
                            COSTATE AT END  -2.683806E-02  2.094087E-01  9.794587E-01  2.541943E-05  1.117771E-06  7.909943E-07
                            PSY=  7.576405E-01      ALPHA= -1.636083E+01      CALCULATED COAST TIME=  1.872746E+04
                         SEMIMAJOR AXIS=  2.436317E+04 RMIN=  6.558651E+03 RMAX=  4.216769E+04 ENERGY= -8.180413E+00
                         PERIOD=  3.784522E+04 HMAG=  6.726664E+04 H VECTOR  1.153930E+03  3.500856E+03 -6.716557E+04
                         EMAG=  7.307965E-01 E VECTOR  7.302401E-01  2.492529E-02  1.384498E-02 RMAG=  4.216713E+04
                     BURN ARC
                         LEG= 4 STATE AT END  -4.214131E+04 -1.336984E+03 -6.660099E+02 -1.017078E-01  2.062185E+00  2.278494E+00
                            COSTATE AT END  -2.355503E-02  2.095438E-01  9.795174E-01  2.543611E-05  9.739700E-07  1.186818E-07
                            MASS AT END OF LEG=  4.028275E+06
                         SEMIMAJOR AXIS=  4.217548E+04 RMIN=  4.216691E+04 RMAX=  4.218406E+04 ENERGY= -4.725513E+00
                         PERIOD=  8.619865E+04 HMAG=  1.296581E+05 H VECTOR -1.672874E+03  9.608644E+04 -8.703913E+04
                         EMAG=  2.032591E-04 E VECTOR -1.798668E-04 -6.525732E-05 -6.858351E-05 RMAG=  4.216777E+04
                     TOTAL BURN TIME=  3.849276E+02 ARC TIMES   4.000000E+02  2.558176E+02  1.872746E+04  1.291100E+02

                         DC  -1.968911E+00  5.063665E+00  2.296253E+01  1.798668E-04  6.525732E-05  1.145031E-08
                              9.004042E-07  1.874982E-08 -9.780427E-05  0.             0.             0.
                         DETERMINANT OF E= -4.953781E+06 DIAGONAL OF E  2.363277E+06  7.716025E+04  2.779820E+04  5.059689E+03
                          -1.282283E+03  1.344838E+01  1.453115E-02 -9.844767E-06  5.183974E-03 -1.510309E-08
                         DXF  -1.395360E-01 -2.832318E-01  3.898002E-01  2.955619E-04 -5.430284E-04  1.182267E-04
                         DRF=  5.016321E-01 DVF=  6.294555E-04 CK=  1.000000E+00 EVT=  1.000000E-14
                         CK=MIN OF    9.25126E+02  1.46230E+02  1.00000E+00  7.52355E+05  4.49457E+03  2.23327E+05

                         CHANGE REQUISTED IN INITIAL COSTATE,SWITCHING TIMES AND FINAL TIME
```

```
                    NEW Q0   5.061979E-01 -8.463290E-01 -1.658037E-01 -6.144147E-04 -7.187790E-04 -1.680213E-04
                    NEW SWITCH TIMES   1.433832E+03  1.689650E+03  2.041919E+04  2.054830E+04
          ITERATION NUMBER   2

                        COAST ARC
                            LEG= 1 STATE AT END   6.408711E+03  1.341881E+03  1.277195E+02  1.581180E+00 -7.638110E+00 -3.085522E-03
                                COSTATE AT END   1.820784E-01 -1.048089E+00 -2.127385E-01 -9.556442E-04 -2.473559E-04 -6.175324E-05
                                PSY=  6.102138E-02     ALPHA= -6.088936E+01     CALCULATED COAST TIME=  3.998319E+02
                            SEMIMAJOR AXIS=  6.546326E+03 RMIN=  6.531170E+03 RMAX=  6.561482E+03 ENERGY= -3.044468E+01
                            PERIOD=  5.271167E+03 HMAG=  5.108191E+04 H VECTOR  9.713954E+02  2.217218E+02 -5.107219E+04
                            EMAG=  2.315202E-03 E VECTOR  7.232001E-05 -2.314056E-03 -8.670576E-06 RMAG=  6.548933E+03
                        BURN ARC
                            LEG= 2 STATE AT END   6.535921E+03 -9.229923E+02  6.411993E+01 -6.722314E-01 -1.018145E+01 -5.422219E-01
                                COSTATE AT END  -6.826809E-02 -1.062571E+00 -2.185335E-01 -9.652236E-04  1.336812E-04  1.684696E-05
                                MASS AT END OF LEG=  6.918322E+06
                        COAST ARC
                            LEG= 3 STATE AT END  -4.213035E+04 -1.570370E+03 -8.052712E+02 -7.415466E-02  1.591469E+00  8.167749E-02
                                COSTATE AT END  -2.911961E-02  2.269079E-01  1.062597E+00  2.757984E-05  1.191153E-06  8.667272E-07
                                PSY=  7.576990E-01     ALPHA= -1.636078E+01     CALCULATED COAST TIME=  1.872954E+04
                            SEMIMAJOR AXIS=  2.436324E+04 RMIN=  6.558651E+03 RMAX=  4.216783E+04 ENERGY= -8.180388E+00
                            PERIOD=  3.784539E+04 HMAG=  6.726666E+04 H VECTOR  1.153300E+03  3.500816E+03 -6.716560E+04
                            EMAG=  7.307973E-01 E VECTOR  7.302390E-01  2.498246E-02  1.384107E-02 RMAG=  4.216730E+04
                        BURN ARC
                            LEG= 4 STATE AT END  -4.214144E+04 -1.337261E+03 -6.656211E+02 -1.014117E-01  2.061643E+00  2.278624E+00
                                COSTATE AT END  -2.555755E-02  2.270517E-01  1.062662E+00  2.759790E-05  1.035336E-06  1.373526E-07
                                MASS AT END OF LEG=  4.028264E+06
                            SEMIMAJOR AXIS=  4.216817E+04 RMIN=  4.216790E+04 RMAX=  4.216844E+04 ENERGY= -4.726332E+00
                            PERIOD=  8.617623E+04 HMAG=  1.296468E+05 H VECTOR -1.674842E+03  9.609199E+04 -8.701623E+04
                            EMAG=  6.319555E-06 E VECTOR -6.318798E-06 -9.661769E-08  1.492601E-08 RMAG=  4.216790E+04
                        TOTAL BURN TIME=  3.849281E+02 ARC TIMES   3.998319E+02  2.558178E+02  1.872954E+04  1.291103E+02

                            DC  -1.123247E-03 -4.825637E-01  5.628740E-02  6.318798E-06  9.661769E-08 -2.406201E-11
                                -6.245005E-17  4.207506E-11  1.198258E-05  0.            0.            0.
                            DETERMINANT OF E= -4.952307E+06 DIAGONAL OF E  2.178922E+06  7.119091E+04  2.557792E+04  4.666866E+03
                             -1.390846E+03  1.243308E+01  1.454159E-02 -1.167385E-05  5.562259E-03 -1.638006E-08
                            DXF  -8.877080E-03 -8.556068E-03 -1.748166E-03 -7.214129E-07 -1.813445E-06 -1.194664E-05
                            DRF=  1.245251E-02 DVF=  1.210501E-05 CK=  1.000000E+00 EVT=  1.000000E-14
                            CK=MIN OF    1.06815E+05  1.80497E+04  1.00000E+00  1.80719E+06  2.52071E+07  1.10879E+05

                            CHANGE REQUISTED IN INITIAL COSTATE,SWITCHING TIMES AND FINAL TIME
                              1.550672E-07  4.476341E-07 -1.811483E-06 -4.608971E-10  1.774056E-10 -1.587006E-09
                                       -2.636723E-05  4.441062E-05  4.159234E-04 -1.662874E-04
          END OF ITERATION NUMBER   2
                    NEW Q0   5.061981E-01 -8.463286E-01 -1.658055E-01 -6.144152E-04 -7.187788E-04 -1.680229E-04
                    NEW SWITCH TIMES   1.433832E+03  1.689650E+03  2.041919E+04  2.054830E+04
```

```
                                             SUMMARY TABLES
ITERATION TOTAL BURN TIME                         LENGTH OF BURN AND COAST ARCS
 NUMBER

    1       3.8492760E+02  4.0000000E+02  2.5581760E+02  1.8727456E+04  1.2911000E+02
    2       3.8492806E+02  3.9983188E+02  2.5581777E+02  1.8729539E+04  1.2911029E+02


ITERATION                                           ERROR IN BOUNDARY CONDITIONS-DC(1-8)
 NUMBER

    1      -1.968911E+00  5.063665E+00  2.296253E+01  1.798668E-04  6.525732E-05  1.145031E-08  9.004042E-07  1.874982E-08
    2      -1.123247E-03 -4.825637E-01  5.628740E-02  6.318798E-06  9.661769E-08 -2.406201E-11 -6.245005E-17  4.207506E-11


ITERATION                         DC(9-12)                                        DRF            DVF            CK
 NUMBER

    1      -9.7804266E-05  0.            0.            0.              5.0163209E-01  6.2945552E-04  1.0000000E+00
    2       1.1982583E-05  0.            0.            0.              1.2452507E-02  1.2105011E-08  1.0000000E+00
```