Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming

Paul J. Enright* and Bruce A. Conway†
University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

A new method is described for the determination of optimal spacecraft trajectories in an inverse-square field using finite, fixed thrust. The method employs a recently developed direct optimization technique that uses a piecewise polynomial representation for the state and controls and collocation, thus converting the optimal control problem into a nonlinear programming problem, which is solved numerically. This technique has been modified to provide efficient handling of those portions of the trajectory that can be determined analytically, i.e., the coast arcs. Among the problems that have been solved using this method are optimal rendezvous and transfer (including multirevolution cases) and optimal multiburn orbit insertion from hyperbolic approach.

Introduction

HISTORICALLY, research into the optimization of finite-thrust spacecraft trajectories has fallen into two categories. Indirect methods require the derivation of the necessary conditions for optimality using calculus of variations techniques and result in a two-point boundary value problem (TPBVP), which typically needs to be solved numerically. Direct methods, on the other hand, approximate the continuous problem with a parameter optimization problem, usually requiring mathematical programming for solution. This reduction can involve parameterization of the control variables only, in which case the state trajectory is explicitly integrated (direct shooting), or the states and controls can both be parameterized, requiring some sort of implicit integration scheme to satisfy the equations of motion.

The application of the standard variational optimal control techniques (the indirect method) to the finite-thrust spacecraft trajectory problem results in the primer vector formulation of Lawden.[1] The resulting TPBVP can be difficult to solve because of sensitivity problems. Kelley[2] solved the TPBVP using his method of gradients. Melbourne and Sauer,[3] Handelsman,[4] Kern and Greenwood,[5] Hazelrigg and Lion,[6] and, more recently, Redding and Breakwell[7] used shooting methods to solve the TPBVP. McCue[8] used quasilinearization. All of these methods require fairly accurate initial estimates of the solution for convergence. These are derived by extrapolation from known impulsive solutions. A different approach to solving the indirect formulation TPBVP was taken by Dickmanns and Well,[9] who used a collocation method employing piecewise Hermite cubics. This method avoids the inherent sensitivity of shooting methods and demands less accurate initial guesses for successful solution.

Many different direct optimization techniques have been proposed and attempted. Direct shooting uses a finite-parameter representation of the control history and explicitly integrates the state trajectory. In general, mathematical programming is required to optimize the control parameters to minimize the cost and simultaneously satisfy terminal boundary conditions. Sensitivity problems can arise, and many trajectory integrations may be required during the optimization process, which can be costly. A unique approach was taken by Zondervan et al.,[10] where a hybrid (direct/indirect) formulation[11] using nonlinear programming for solution was used to solve some transfer problems. A purely direct approach was described by Johnson,[12] and later by Hargraves et al.,[13] who represented the control and state histories using Chebyshev polynomials and used integral penalty functions to enforce the equations of motion, thus converting the optimal control problem to a problem in unconstrained minimization, which is solved numerically. A different direct approach for solving optimal control problems uses Dickmanns's collocation method with Hermite cubics[9] to convert the optimal control problem into a nonlinear programming problem. This is the method used in the current study and is referred to as direct collocation with nonlinear programming (DCNLP). Hargraves and Paris[14] used this method to solve several atmospheric trajectory optimization problems and installed it in their trajectory optimization routine OTIS. This approach was also described by Kraft,[15] but the handling of the controls was different from that of Hargraves and Paris,[14] and no computational experience was related.

Direct Collocation with Nonlinear Programming

In the DCNLP approach, the trajectory is approximated by piecewise polynomials, which are represented by state and control values at a number of discrete points (nodes). For a given state variable, the state trajectory over a given segment between two nodes is taken to be the unique cubic that goes through the endpoints of the segment with the appropriate derivatives that are dictated by the evaluation of the differential equations of motion at the endpoints. (This is the Hermite cubic because it is determined by the states and their first derivatives.) Following Hargraves and Paris,[14] the control history is taken to be linear across the segment. A collocation point is taken at the center of the segment, where the derivative obtained from the Hermite cubic is compared to the derivative obtained from the evaluation of the equations of motion. The difference is termed the defect[9] and is a measure of nonsatisfaction of the equations of motion over the given segment, i.e., if the defect is zero, then the differential equation is satisfied at the center collocation point as well as at the endpoints.

Let the system equations of motion be given by $\dot{x} = f(x,u)$, where x is the state vector, u the control vector, and $(\dot{\ })$ denotes differentiation with respect to time. Let the length (in

Presented as Paper 89-350 at the AAS/AIAA Astrodynamics Specialist Conference, Stowe, VT, Aug. 7–10, 1989; received April 9, 1990; revision received July 6, 1990; accepted for publication Aug. 2, 1990.

*Graduate Research Assistant, Department of Aeronautics and Astronautics Engineering; currently Member of the Technical Staff, Jet Propulsion Laboratory, Pasadena, CA 91109. Member AIAA.

†Associate Professor, Department of Aeronautics and Astronautics Engineering, College of Engineering, 101 Transportation Building, 104 South Mathews Avenue, Associate Fellow AIAA.

time) of a particular segment be T. Then, the Hermite-interpolated state vector at the center point is[14]

$$x_c = (1/2)(x_l + x_r) + (T/8)[f(x_l) - f(x_r)] \tag{1}$$

where x_l and x_r are the state vectors at the left and right nodes (Fig. 1). The linearly interpolated control vector is simply

$$u_c = (1/2)(u_l + u_r) \tag{2}$$

The derivative of the interpolating Hermite cubic at the center point is

$$\dot{x}_c = -[3/(2T)](x_l - x_r) - (1/4)[f(x_l) + f(x_r)] \tag{3}$$

The defect vector is then calculated as

$$d = f(x_c,u_c) - \dot{x}_c \tag{4}$$

If x_l, u_l, x_r, and u_r are chosen such that the elements of d are sufficiently small, the interpolating polynomials become an accurate approximation to the solution of the differential eqtions of motion (implicit integration).

Now that the satisfaction of the differential equations of motion has been parameterized, we can proceed to convert the optimal control problem into a nonlinear programming problem. From the states and controls at the nodes, we can calculate the objective function, the defects, and any problem constraints (e.g., constraints on the initial and final states and/or path constraints). If the final time is allowed to vary, then the segment times are calculated by some fixed relationship to the final time. The nonlinear programming routine is used to minimize the objective, subject to the constraints that all of the defects are acceptably small and all other problem constraints are satisfied.

Application to Finite-Thrust Spacecraft Trajectories

For the current study of finite-thrust spacecraft trajectories, the equations of motion, in two-dimensional polar form (Fig. 2), are

$$\dot{x}_1 = x_3 \tag{5a}$$

$$\dot{x}_2 = x_4/x_1 \tag{5b}$$

$$\dot{x}_3 = x_4^2/x_1 - 1/x_1^2 + x_5 \sin u_1 \tag{5c}$$

$$\dot{x}_4 = -x_3 x_4/x_1 + x_5 \cos u_1 \tag{5d}$$

$$\dot{x}_5 = x_5^2/c \tag{5e}$$

where x_1 is the radius from center of attraction, x_2 the angle from some reference direction, x_3 the velocity component along the radial direction, x_4 the velocity component along the horizontal direction, x_5 the thrust acceleration, u_1 the thrust vector angle, and c the effective exhaust velocity. Units are chosen such that the gravitational parameter μ is 1.

The trajectory is first divided into a sequence of thrust arcs (constant thrust) and coast arcs (no thrust). The thrust arcs are further subdivided (evenly) into segments (Fig. 3). At each segment center, collocation defects are calculated from the states and controls at the nodes and the duration of the thrust arc. The coast arcs are not divided into segments because the solution over a coast arc can be determined analytically. Instead, integrals of the motion (e.g., orbital elements) are calculated from the state vectors at each end of the coast arc. If the endpoints are compatible, these integrals are unchanged. Thus, a generalized defect is taken to be the difference between the integrals at the left and right nodes of the coast arc:

$$q_i = Q_i(x_l) - Q_i(x_r)$$

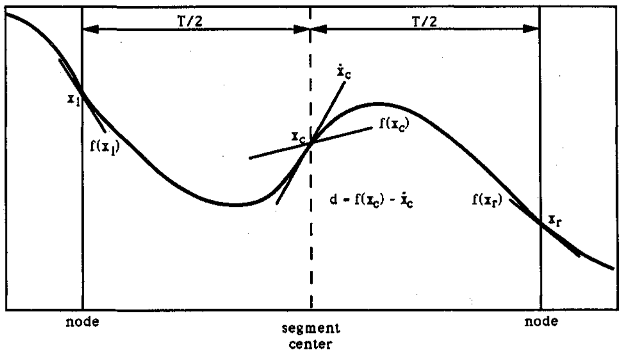

Fig. 1 Collocation using Hermite polynomials.

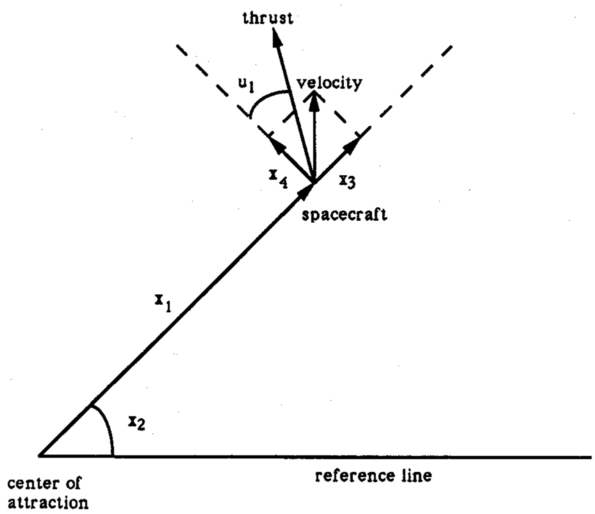

Fig. 2 State and control variables.

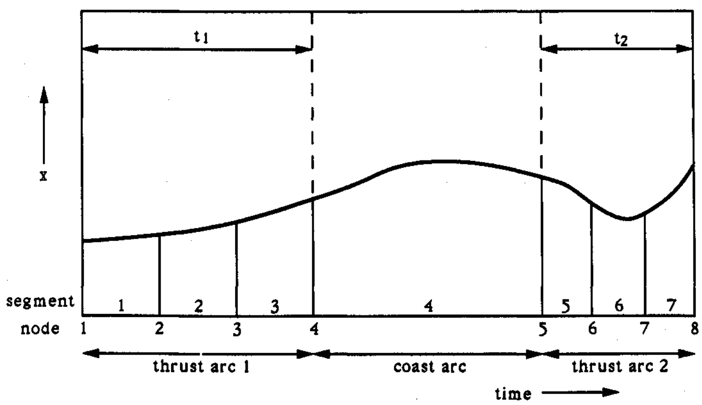

Fig. 3 Problem structure.

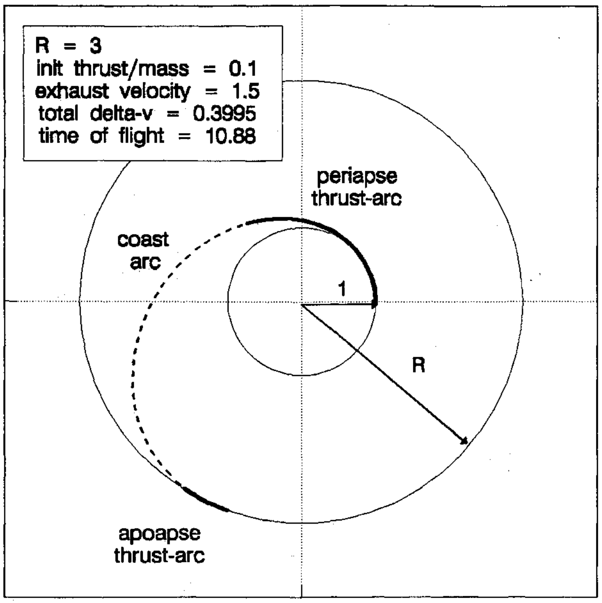

Fig. 4 Trajectory for optimal two-burn transfer.

where q_i is the ith generalized defect over the given coast arc, Q_i the ith integral of the motion, x_l the state vector at the left node of the arc, and x_r the state vector at the right node of the coast arc. If x_l and x_r are chosen such that the q_i are acceptably small, then they are endpoints of the same Keplerian coast arc.

For the two-dimensional thrust-free orbit equations [Eqs. (5) with $x_5 = 0$], there are three such constants of the motion. Angular momentum, energy, and longitude of periapse were usually chosen. A fourth integral is provided by the simple requirement that the thrust acceleration at the end of the thrust arc, which precedes the coast arc, equals the thrust acceleration at the start of the thrust arc, which follows (because no mass is expelled during a coast arc). Using this scheme, the four integrals become

$$Q_1(x) = x_1 x_4 \tag{6a}$$

$$Q_2(x) = (x_3^2 + x_4^2)/2 - 1/x_1 \tag{6b}$$

$$Q_3(x) = x_2 + \tan^{-1}[-x_3 x_4/(x_4^2 - 1/x_1)] \tag{6c}$$

$$Q_4(x) = x_5 \tag{6d}$$

The introduction of the generalized defects may greatly reduce the number of segments required, thus greatly reducing the dimensions of the nonlinear programming problem.

Nonlinear programming routines typically require the evaluation of the derivatives of the objective function and the constraints with respect to the variables. Hargraves and Paris[14] used finite-difference approximations to these derivatives; however, analytical expressions are easily obtained for the current problem and are used exclusively. Derivatives of the collocation defects involve chain-rule differentiation of the interpolation and equation of motion relationships, Eqs. (1-5). For example, the derivative of a defect vector with respect to the state vector at the node on the segment's left is given by

$$F(x_c,u_c)[I_5/2 + (T/8)F(x_l,u_l)] + [3/(2T)]I_5 + (1/4)F(x_l,u_l) \tag{7}$$

where $F(x,u)$ denotes the 5×5 matrix, which results from differentiation of the right side of the equations of motion $f(x,u)$ with respect to the state vector x, and I_5 is the 5×5 identity matrix. Defect derivatives with respect to the right state vector, the left and right control vectors, and the thrust arc duration are calculated similarly. Derivatives of the generalized defects are obtained by simply differentiating the integrals of the thrust-free motion, Eqs. (6), with respect to the state vector. The objective function is usually taken to be the sum of the thrust arc durations, and so its derivative is trivial. Finally, any other problem constraints (except constraints expressed as simple bounds) must be differentiated. These are usually constraints on initial or terminal orbital elements (path constraints can also be handled), and their derivatives are found in a manner similar to that of the derivatives of the generalized defects.

Structure and Solution of the Nonlinear Programming Problem

The variables for the nonlinear programming (NLP) problem are the collected state vectors and control vectors at the nodes and the thrust arc durations. These quantities are assembled into the NLP state vector X:

$$X^T = [x_1^T, u_1^T, x_2^T, u_2^T, \ldots, x_n^T, u_n^T, t_1, t_2, \ldots, t_k] \tag{8}$$

where n is the total number of nodes and k is the number of thrust arcs on the trajectory. The defects, generalized defects, and any other problem constraints are collected into the NLP constraint vector C:

$$C^T = [g_1^T, g_2^T, \ldots, g_{n-1}^T, w^T] \tag{9}$$

where g_i is either the defect vector for the corresponding thrust arc segment or the vector of generalized defects for the corresponding coast arc, and w is a vector of additional problem constraints.

To solve the NLP problem, a sequential quadratic programming (SQP) method was employed.[16] This is a Lagrangian method that involves the solution of a sequence of quadratic programming problems that approximate the full nonlinear problem. The particular routine used is part of the Numerical Algorithms Group (NAG) library[17] and is essentially identical to the NPSOL routine used by Hargraves and Paris.[14] The NAG routine is a FORTRAN subroutine, which must be called by the user's program. The routine must be provided with subroutines that evaluate the objective, the gradient of the objective with respect to the states, the constraints, and the Jacobian of the constraints with respect to the states. In addition, a user's main program must set up algorithm parameters, problem dimensions, any linear constraints or simple bounds, and initial conditions.

These routines were coded in Cray FORTRAN 77 on the Cray X-MP/48 at the National Center for Supercomputing Applications (NCSA) at the University of Illinois.

Solved Problems

Among the problems solved using this method were optimal two-burn transfer, optimal three-burn transfer (multirevolution), and optimal two-burn insertion (multirevolution), all using finite, fixed thrust. Canonical units are used exclusively. For the transfer problems, the initial orbit radius is one distance unit, and the initial orbit period is 2π ($\mu = 1$). For the insertion problem, units are similarly normalized to the final orbit.

The two-burn transfer problem is the finite-thrust analogy to the impulsive Hohmann transfer. Figure 4 depicts the optimal trajectory for this circle-to-circle time-open transfer. Figure 5 shows the optimal thrust pointing angle (the angle u_1 of Fig. 2) over each of the burns. The trajectory was assumed to consist of two thrust arcs separated by a coast arc. Each thrust arc was divided into 12 segments, resulting in a total of 26 nodes. The NLP state vector contained 158 elements [26 nodes $\times$ (5 states + 1 control) + 2 thrust arc durations]. The only constraints were the defects (5×24) and the generalized defects (4×1), resulting in a constraint vector dimension 124. The circular terminal orbit conditions can be specified as sim-

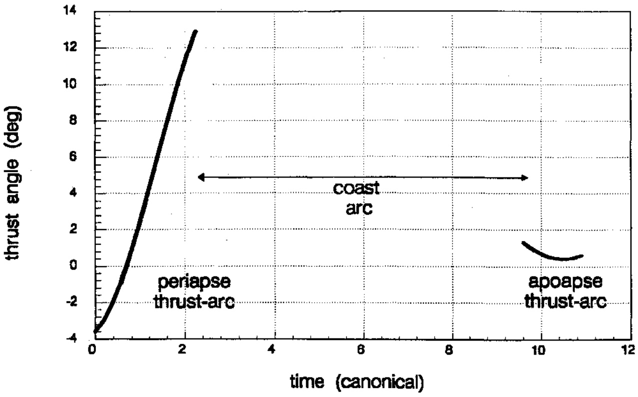

Fig. 5 Thrust angle for optimal two-burn transfer.

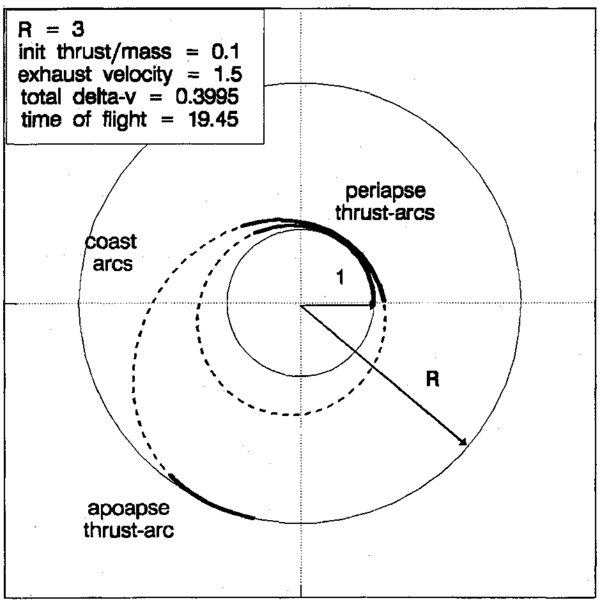

Fig. 6 Trajectory for optimal three-burn transfer.

ple bounds on the state variables and are not considered constraints, per se, by the NLP routine.

The three-burn transfer problem is similar to the two-burn problem, except that two burns must take place at periapse, with a complete-revolution coast arc between them. Figure 6 depicts the optimal trajectory for this solution. The trajectory was assumed to consist of three thrust arcs separated by two coast arcs. Each thrust arc was represented by 10 segments, resulting in a NLP problem size of 201 states and 158 constraints. The optimal cost is less than that of the two-burn solution, as expected; however, both are local time-open optima.

The two-burn insertion problem calls for optimal insertion from a hyperbolic approach trajectory, with specified excess velocity, into a circular orbit, using one burn for initial capture and a subsequent burn for circularization. Figure 7 depicts the optimal solution. The assumed arc sequence was thrust-coast-thrust. Each thrust arc was represented by 10 segments, resulting in a NLP problem size of 134 states and 106 constraints. The circular-orbit terminal constraints are treated as simple bounds on the final state vector. The initial hyperbolic trajectory energy was constrained to match the required excess velocity, but the initial angular momentum (and thus the aim point) was free. Thus, the approach targeting, the elements of the intermediate coast ellipse, the thrust arc durations, and the thrust direction over the burns were all simultaneously optimized.

This problem was intended to be representative of the insertion of the Mars Observer spacecraft. In fact, the problem was inspired by an optimization strategy described by Parvez and

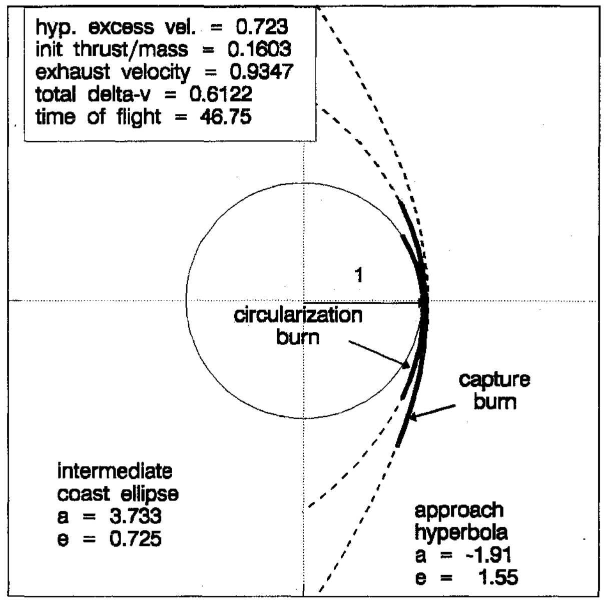

Fig. 7 Trajectory for optimal two-burn insertion.

Holman,[18] in which the optimal thrust direction was assumed to be opposite the velocity vector, and the terminal boundary conditions were met by iterative adjustment of the initial conditions.

Many other problems have been solved, including circle-to-circle transfers with more than three burns, circle-to-circle rendezvous problems, fixed acceleration problems, and variants of the Mars Observer problem with different constraints on the intermediate coast ellipse. An interesting feature of the rendezvous problems is that the coast arc times must be calculated from the forward Kepler equation and appropriately constrained.

Performance

During development, the method was tested against known solutions where possible. Finite-thrust, circle-to-circle transfer results were compared to results from a program[19] based on the Redding-Breakwell[7] approach. For the circle-to-circle cases presented earlier, the optimal costs (Δv) matched to six decimal places.

The best indication of solution accuracy was obtained by exploiting the relationship between the costate variables of the Euler-Lagrange equations for the continuous optimal control problem and the discrete Lagrange multipliers at the solution of the nonlinear programming problem. The authors show elsewhere[20] that the Lagrange multipliers corresponding to the defects provide a discrete approximation to the continuous problem costates. The costate information was used to verify that the Euler-Lagrange equations were indeed satisfied by the discrete-approximated solution. This process is described in detail in Ref. 20.

The DCNLP method is remarkably robust to initial guesses, a feature pointed out by Hargraves and Paris.[14] (Dickmanns and Well[9] also reported such robustness for their collocation method applied to the indirect TPBVP.) We have used very poor and sometimes physically unreasonable initial guesses of the solution, yet the optimizer converges to the optimal trajectory.

Perhaps the method's strongest feature is its adaptability. Constraints of almost any type can be added in a very straightforward fashion. An example was the addition of a constraint on the semimajor axis of the intermediate coast ellipse for the Mars Observer problem. This was accomplished simply by the addition of a few lines of code.

Execution times on the Cray X-MP/48 were very brief: 2.5 s for the two-burn transfer, 10.7 s for the three-burn transfer, and 3.6 s for the Mars Observer two-burn insertion. Execution times on a Convex C220 were longer by a factor of 5.

Current and Future Work

Current research involves application of the DCNLP method to low-thrust trajectories in Earth-Moon space. These problems involve many revolutions, and no integrable coast arcs, requiring many segments and resulting in fairly large nonlinear programming problems. We would like to improve the efficiency of the solution of these larger problems. An area of particular interest is the banded structure of the constraint Jacobian, which is due to the local nature of the collocation method. (Variations of the states at a given node affect only the defects of the adjacent segments.) Hargraves and Paris[14] recognized this sparsity and avoided the (unnecessary) calculation of the null entries by their finite-difference derivative routine. However, the aspect more critical to us is the possible exploitation of this structure by the NLP routine. (Kraft[15] also mentions the need for such a routine.) SQP routines that are tailored to handle sparse problems are being investigated.[21,22] Other NLP routines that are designed to handle large, sparse problems[23,24] are also being considered.

Conclusions

A new direct optimization method has been adapted for the optimization of finite-thrust spacecraft trajectories. The method has been improved by the introduction of generalized defects that allow known solutions for pieces of the optimal trajectory, e.g., coast arcs, to be used directly rather than solved for numerically, reducing the size of some problems significantly. The method was used to solve several problems, including multiple-finite-burn transfer, rendezvous, and insertion. The method appears to be considerably more powerful and robust than other optimization methods.

Acknowledgments

This work was supported by NASA Grant NAG 3-805, administered by NASA Lewis Research Center, and utilized the Cray X-MP/48 system at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL.

References

[1]Lawden, D. F., *Optimal Trajectories for Space Navigation,* Butterworths, London, 1963.

[2]Kelley, H. J., "Gradient Theory of Optimal Flight Paths," *American Rocket Society Journal,* Vol. 30, No. 10, 1960, pp. 947–954.

[3]Melbourne, W. G., and Sauer, C. G., "Optimal Thrust Programs for Power-Limited Propulsion Systems," *Astronautica Acta,* Vol. 8, 1961, pp. 205–227.

[4]Handelsman, M., "Optimal Free-Space Fixed-Thrust Trajectories Using Impulsive Trajectories as Starting Iteratives," *AIAA Journal,* Vol. 4, No. 6, 1966, pp. 1077–1082.

[5]Kern, E. A., and Greenwood, D. T., "Minimum-Fuel Thrust-Limited Transfer Trajectories Between Coplanar Elliptic Orbits," AIAA Paper 69-914, Aug. 1969.

[6]Hazelrigg, G. A., and Lion, P. M., "Analytical Determination of the Adjoint Vector For Optimum Space Trajectories," AIAA Paper 69-916, Aug. 1969.

[7]Redding, D. C., and Breakwell, J. V., "Optimal Low-Thrust Transfers to Synchronous Orbit," *Journal of Guidance, Control, and Dynamics,* Vol. 7, No. 2, 1984, pp. 148–155.

[8]McCue, G. A., "Quasilinearization Determination of Optimum Finite-Thrust Orbital Transfers," *AIAA Journal,* Vol. 5, No. 4, 1967, pp. 755–763.

[9]Dickmanns, E. D., and Well, H., "Approximate Solution of Optimal Control Problems Using Third-Order Hermite Polynomial Functions," *Proceedings of the 6th Technical Conference on Optimization Techniques,* Springer-Verlag, New York, 1975.

[10]Zondervan, K. P., Wood, L. J., Caughey, T. K., "Optimal Low-Thrust, Three-Burn Orbit Transfers with Large Plane Changes," *Journal of the Astronautical Sciences,* Vol. 32, No. 3, July–Sept. 1984, pp. 407–427.

[11]Bartholomew-Biggs, M. C., "The Optimisation of Spacecraft Orbital Maneuvers," *Numerical Optimisation of Dynamic Systems,* edited by L. C. W. Dixon and G. P. Szego, North-Holland, Amsterdam, 1980.

[12]Johnson, F. T., "Approximate Finite-Thrust Trajectory Optimization," *AIAA Journal,* Vol. 7, No. 6, 1969, pp. 993–997.

[13]Hargraves, C. R., Johnson, F., Paris, S., and Rettie, I., "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance and Control,* Vol. 4, No. 4, 1981, pp. 406–414.

[14]Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics,* Vol. 10, No. 4, 1987, pp. 338–342.

[15]Kraft, D., "On Converting Optimal Control Problems Into Nonlinear Programming Problems," *Computational Mathematical Programming,* Vol. F15, NATO ASI Series, Springer-Verlag, Berlin, 1985.

[16]Fletcher, R., *Practical Methods of Optimization,* Wiley, New York, 1987.

[17]Numerical Algorithms Group, E04VCF—NAG FORTRAN Routine Library Document.

[18]Parvez, S. A., and Holman, J. M. L., "Low-Thrust Insertion into Orbit Around Mars," *Journal of Guidance, Control, and Dynamics,* Vol. 11, No. 5, 1988, pp. 475–477.

[19]Marinos, G. M., and Sentman, L. H., "Low Thrust, Optimal, Time-Free Transfer Between Coaxial, Aligned, Noncoplanar Elliptical Orbits," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 85-1918, Snowmass, CO, 1985.

[20]Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics* (to be published).

[21]Nickel, R. H., and Tolle, J. W., "A Sequential Quadratic Programming Algorithm For Solving Large, Sparse Nonlinear Programs," Center for Naval Analyses, Alexandria, VA, May 1984.

[22]Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "A Schur-Complement Method For Sparse Quadratic Programming," Systems Optimization Laboratory, Stanford Univ., Stanford, CA, TR-SOL-87-12, Oct. 1987.

[23]Murtaugh, B. A., and Sanders, M. A., "A Projected Lagrangian Algorithm and Its Implementation for Sparse Nonlinear Constraints," *Mathematical Programming Study 16,* North-Holland, Amsterdam, 1982, pp. 84–117.

[24]Betts, J. T., and Huffman, W. P., "The Application of Sparse Nonlinear Programming to Trajectory Optimization," Boeing Computer Services, Seattle, WA, Jan. 1990.