Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

A Parallel Quasi-Linearization Algorithm for Air Vehicle Trajectory Optimization

P.K.A. Menon* and L.L. Lehman*
Integrated Systems, Inc., Palo Alto, California

Introduction

PARALLEL computing architectures are currently being explored to improve speed in "compute bound"[1,2] situations. Flight vehicle trajectory optimization is an area that could significantly benefit from these emerging technologies. Trajectory optimization problems continue to be computationally challenging, because the application of Pontriagin's minimum principle[3] produces in general nonlinear two point boundary-value problems (TPBVP) with attendant numerical difficulties. Several algorithms have been proposed for their solution, such as multiple shooting,[4] quasi-linearization,[3] and continuation-in-shooting.[4] Alternate approaches, classed under direct methods via nonlinear programming, have also been used to solve these problems (see Ref. 5, for example). Previous research on parallel nonlinear programming alogithms have been discussed by Straeter[6] and Chazan and Miranker.[7] Travassos and Kaufman[8] have presented some parallel algorithms for TPBVP solution including a parallel shooting algorithm.

In this note, a parallel quasi-linearization approach will be developed using the integrating matrix[9] technique. Quasilinearization techniques have been used to solve TPBVP arising in optimal control.[3] These techniques are based on the fact that linear TPBVP are often computationally simpler to solve than the nonlinear TPBVP. In one version of quasilinearization, the nominal state/costate trajectories that satisfy as many boundary conditions as possible are chosen. The state/costate system is then linearized about these trajectories to form a nonhomogeneous, time-varying, linear TPBVP. Solution to this linear TPBVP is then used to update the nominal solution. This process is repeated until the solution satisfies the state/costate system and the boundary conditions to the desired accuracy. An important feature of quasilinearization methods is that they converge rapidly if the initial guess is near the optimal solution. The next section will provide an outline of this algorithm.

The computational speed of a quasi-linearization algorithm is largely controlled by the speed with which the linear TPBVP can be solved. Several approaches are currently available for this purpose, including the sweep method, transition matrix method and simple shooting. The sweep technique and the transition matrix methods, given in Ref. 3, are inherently serial. An alternative linear TPBVP solution method, the integrating matrix approach, has recently emerged from the structural dynamics area.[9] In general terms, this approach uses a discretization matrix operator to convert a given linear TPBVP into a linear matrix equation. Boundary conditions are similarly transformed. The solution is then obtained by solving this system of linear equations using well-known algorithms. Depending on the expected nature of the solution to a given problem, one of several possible integrating matrices may be used. The resulting quasilinearization algorithm displays a high degree of parallelism and can be realized using parallel computer architectures, such as the systolic arrays of Kung.[1]

Received Dec. 12, 1984; revision received July 2, 1985.

*Research Scientist. Member AIAA.

Nonlinear TPBVP Solution by Quasi-Linearization

A brief outline of the quasi-linearization algorithms will be given in this section to motivate subsequent development. Application of Pontriagin's minimum principle to an optimal control problem yields the Euler-Lagrange equations and transversality conditions. These together with the state equations and boundary conditions form the TPBVP to be solved to generate optimal control histories. For a fixed-time problem, if $x \epsilon R^n$ and $u \epsilon R^m$, the TPBVP, in general, has the form

$$\dot{x}=f(x,u,t), \quad \dot{\lambda}=g(x,\lambda u,t) \qquad 2n \text{ nonlinear ordinary differential equations} \tag{1}$$

$$h(x,u,\lambda,t)=0 \qquad m \text{ nonlinear algebraic equations} \tag{2}$$

$$\theta[x(t_0),\lambda(t_0)]=0,$$

$$\psi[x(t_f),\lambda(t_f)]=0, \qquad 2n \text{ boundaty conditions} \tag{3}$$

The algorithm begins with the initial guess of nominal state/costate trajectories $x(t)$ and $\lambda(t)$ which satisfy approximately the boundary conditions of Eq. (3) and the differential-algebraic system of Eqs. (1) and (2). Note that if the nonlinear algebraic equations [Eq. (2)] cannot be solved explicitly for the control variables u, then, in addition to $x(t)$ and $\lambda(t)$. The nominal control histories would have to be assumed also. A nonhomogeneous, time-varying linear TPBVP is then set up by imbedding a convergence parameter ϵ, which is increased from a small value to unity in several iterations. Reference 3 has further details on the quasilinearization algorithm, including its extension for the free-time case. In the next section, the integrating matrix technique for solving the resulting linear TPBVP will be discussed. It will be seen that the integrating matrix approach yields an inherently parallel structure to the solution.

Integrating Matrix for Linear TPBVP Solution

Integrating matrices provide a convenient and efficient method for solving a nonhomogeneous linear TPBVP. Although numerical solution by integrating matrix is not entirely new, this approach has received little attention compared to other numerical tools. A survey of the early applications of the integrating matrix technique is given by Lehman.[9] The integrating matrix development is based on continuous polynomials that approximate the solution behavior of the state variables. Integrating matrices can be developed for piecewise continuous functions also, enabling their use in problems requiring jumps in states and costates. Additionally, since the integrating matrix method easily accommodates very high order approximating functions, a considerable reduction in problem size is often achieved for a required solution accuracy. For notational convenience, the nonhomogeneous, time-varying, linear TPBVP from the quasi-linearization is written as

$$\dot{y}=A(t)y+C(t), \quad (\alpha y+\beta)_{t=t_0}=0, \quad (\gamma y+\delta)_{t=t_f}=0 \tag{4}$$

A block diagonal global integrating matrix is next defined as

$$\tilde{L}=\begin{bmatrix} L & & \\ & \ddots & \\ & & L \end{bmatrix} \tag{5}$$

with each block being an $(N+1)\cdot(N+1)$ integrating matrix. N is the number of intervals in the solution of the linear TPBVP. In a given problem, this is a user selected entity. The number of diagonal blocks in the global integrating matrix is equal to the number of states and costates. The integrating matrix L is the product of two matrices: a specifically formulated numerical quadrature weighting matrix and a lower triangular summing matrix, i.e.,

$$L=SW \tag{6}$$

Several other types of integrating matrices are available, see Ref. 11 for example. In concise terms, the integrating matrix can be viewed as a linear matrix operator that performs integrations via matrix multiplication. Selection of an appropriate integrating matrix for a given problem should be based on the expected nature of the solution and the required accuracy. Further, error reduction concepts such as Richardson's extrapolation in ordinary quadrature may be extended for integrating matrices also.

The global integrating matrix [Eq. (5)] is now applied as a matrix operator on both sides of Eq. (4) yielding

$$\tilde{y}=\tilde{L}\tilde{A}\tilde{y}+\tilde{L}\tilde{C}+\tilde{k} \tag{7}$$

Here $\tilde{y}$ is a column vector with dimensions $2n(N+1)$ and $\tilde{k}$ is a constant vector of integration to be determined from the boundary conditions. Since $\tilde{L}$ is banded and $\tilde{A}$ is sparse, the matrix products $\tilde{L}\tilde{A}$ and $\tilde{L}\tilde{C}$ can be computed very efficiently. To solve for the constant vector of integration, it is useful to introduce the boundary condition matrices B_0 and B_n. These can be written as

$$B_0=1b_0^T \quad \text{and} \quad B_1=1b_n^T$$

where

$$1=[1...1]^T, \; b_0^T=[1,0...0] \text{ and } b_n^T=[0,...0,1]$$

The dimensions of B_0 and B_n are $(N+1)\cdot(N+1)$, the same as a normal integrating matrix. Also, since the first row of the integrating matrix contains only zeros, $B_0L=0$ is obtained.

Corresponding to the global block diagonal integrating matrix $\tilde{L}$, there is a similar block diagonal boundary condition matrix $\tilde{B}$. For a two-point boundary value problem, each matrix block on the diagonal of $\tilde{B}$ is specified by applying either B_0 or B_n (or, equivalently, b_0^T or b_n^T) to the corresponding state variable and solving for its constant vector of integration. Since b_0^T and b_n^T contain mostly zero elements, along with a strategically located unit term, their operational effect on a discrete state vector is to select the "degree of freedom" at which a boundary condition is to be applied.

In addition to the homogeneous boundary condition matrix $\tilde{B}$, $\tilde{B}_{nh}$ can be defined to account for nonhomogeneous boundary conditions that can be written in terms of the state variables. The specific form of $\tilde{B}_{nh}$ has to be determined for each particular problem. When $\tilde{B}_{nh}$ does need to be determined, it is defined in such a way that when it premultiplies the global state vector, it produces the required nonhomogeneous boundary terms. Like $\tilde{B}$, $\tilde{B}_{nh}$ will consist mainly of zeros but will have a few strategically located nonzero terms. The nonzero terms in $\tilde{B}_{nh}$, however, are not usually unit terms as was required for $\tilde{B}$.

With the foregoing, to obtain a general expression for $\tilde{k}$, Eq. (7) is first multiplied through by $\tilde{B}$. Since $\tilde{B}$ has been defined for homogeneous boundary conditions, we have that $\tilde{B}\tilde{y}=0$. Furthermore, the form of $\tilde{k}$ is specified such that $\tilde{B}\tilde{k}=\tilde{k}$. With these two identities and with the aid of the nonhomogeneous boundary term $\tilde{B}_{nh}\tilde{y}$, the general result is obtained from Eq. 7

$$\tilde{k}=-\tilde{B}\tilde{L}(\tilde{A}\tilde{y}+\tilde{C})-\tilde{B}_{nh}\tilde{y} \tag{8}$$

Equation (8) is next substituted in Eq. (7) to obtain the following result

$$[(I+\tilde{B}_{nh})-(I-\tilde{B})\tilde{L}\tilde{A}]\tilde{y}=(I-\tilde{B})\tilde{L}\tilde{C} \tag{9}$$

I is an identity matrix with compatible dimensions. Equation (9) is a system of linear equations which can be solved using well-conditioned numerical algorithms. Moreover, the solution of these equations on parallel computing architectures has been addressed by several investigators.[1,2,10] In a given problem, the matrix $(I-\tilde{B})\tilde{L}$ can be precomputed and stored. During iterations, $\tilde{A}$ and $\tilde{C}$ and the elements of $\tilde{B}_{nh}$ have to be substituted in Eq. (9). Since the integrating matrix is selected before the solution process begins, the elements of $\tilde{A},\tilde{C},\alpha,\beta,\gamma$ and δ can be computed in parallel. Therefore, in a given iteration, there are at least $2n(N+1)$ parallel operations feasible. Note, however, that this would vary from architecture to architecture based on constraints on the computing hardware.

The global integrating matrix $\tilde{L}$ and the global boundary condition matrix $\tilde{B}$ are block diagonal. Further the integrating matrix $\tilde{L}$ contains approximately 40% zero entries. Recent algorithms for sparse matrices[11] may be used to realize substantial improvements in solution speed.

Boost-Glide Missile Trajectory Optimization

As an application example, the quasi-linearization technique with integrating matrix algorithms is applied to the trajectory optimation for a phase of boost-glide missile flight. With an objective of maximum range, the boost-glide air-launched missile starts to coast after the burnout of the rocket. Throughout the trajectory, the dynamic pressure should be high enough to maintain maneuverability. At the end of the

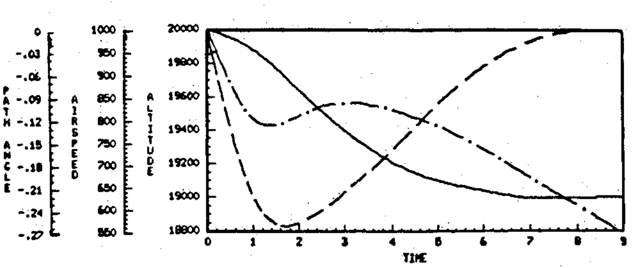

Fig. 1 Optimal trajectory for boost-glide missile:—, altitude; — · — · —, airspeed; --- flight-path angle.

coast phase, the missile must satisfy certain altitude and flight path angle conditions dictated by the terminal guidance requirements. The case of flight in a vertical plane will be considered here. The optimal control problem considered is

$$\min_n \left[-\frac{1}{2} W_S S^2(t_f) - \int_{t_0}^{t_f} W_q q \mathrm{d}t \right] \tag{10}$$

subject to the differential constraints

$$\dot{S} = V\cos\gamma, \dot{h} = V\sin\gamma, \dot{V} = -D/m - g\sin\gamma, \dot{\gamma} = g/V(n-\cos\gamma) \tag{11}$$

Final time t_f is given. Here S is down range distance, h altitude, V airspeed, γ flight-path angle, D aerodynamic drag, g acceleration due to gravity, m missile mass, and q dynamic pressure n is the load factor, the control variable in the present problem. W_S and W_q are weights on final range and dynamic pressure. The aerodynamic drag is modeled as

$$D = qsC_{D_0}(M) + K(M)m^2gn^{22}/qs$$

In the above expression,, s is the reference area, C_{D_0} the zero-lift drag coefficient, K the induced drag coefficient and M is the Mach number. Load factor, the control variable, is constrained as

$$|n| \leq n\max$$

The initial conditions on all the state variables are given, while only the final altitude and flight path angle are specified. The quasi-linearization scheme is next set up by linearizing the state/costate system along with the transversality condition about nominal trajectories.[3]

A computer code was written on a VAX11/750 serial machine to evaluate the quasi-linearization-integrating matrix algorithm. Figure 1 gives the results of this numerical study for the boost-glide missile trajectory problem for the following boundary conditions.

Initial conditions: $h(0) = 20000\text{m}, V(0) = 1000\text{m/s}$, $\gamma(0) = 0$ deg

Final conditions: $h(9) = 19000\text{m}$, $\gamma(9) = 0$ deg

In this particular example, a step size of one second was chosen and a third-degree Newton integrating matrix was employed. The initial guess trajectories for the states were chosen as constants. The initial guesses on costates were generated using an adjoint-control transformation similar to that given in Ref. 12. The quasi-linearization convergence factor ϵ was gradually increased from 0.001 in the first iteration to unity in the 14th iteration. The solutions given in Fig. 1 have been compared with those obtained from a nonlinear programming approach and have been found consistent.

Conclusions

The quasi-linearization alogirithm using the integrating matrix approach described in this paper reduces the nonlinear two-point boundary-value problem to a series of linear algebraic systems and is highly suited for implementation on parallel computing machines. The efficacy of this algorithm was illustrated for a boost-glide missile trajectory optimization on a serial computing machine. This approach can also be used for solving multipoint boundary-value problems and systems with state-costate discontinuities. Mechanization of this algortihm on state-of-the-art parallel processors will be of future interest.

Acknowledgment

The research for this study was supported in part by Eglin Air Force Base under SBIR Contract F8635-83-C-0480.

References

[1]Kung, H.T., "Why Systolic Architectures?," *IEEE Computer Journal*, Jan. 1982, pp. 37-46.

[2]Mead, C., and Conway, L., *Introduction to VLSI Systems*, Addison-Wesley, Reading, MA, 1980.

[3]Bryson, A.E., and Ho., Y.C., *Applied Optimal Control*, Blaisdell, Waltham, MA, 1969.

[4]Keller, H.B., "Numerical Methods of Two-Point Boundary-Value Problems," SIAM Monograph, Philadelphia, PA, 1976.

[5]Hargraves, C., Johnson, F., Parris, S., and Rettie, I., "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance and Control*, Vol. 4, July-Aug. 1981, pp. 406-414.

[6]Straeter, T., "A Parallel Variable Metric Optimization Algorithm," NASA TND-7329, 1973.

[7]Chazan, D., and Miranker, W.L., "A Nongradient and Parallel Algorithm for Unconstrained Minimization," *SIAM Journal of Control*, Vol. 8, 1970, pp. 207-217.

[8]Travassos, R., and Kaufman, H., "Parallel Algorithms for Solving Nonlinear Two-Point Boundary-Value Problems Which Arise in Optimal Control," *Journal of Optimization Theory and Applications*, Vol. 30, Jan. 1980, pp. 53-71.

[9]Lehman, L.L., "Hybrid State Vector Methods for Structural Dynamic and Aeroelastic Boundary Value Problems," NASA CR-3591, Aug. 1982.

[10]Heller, D., "A Survey of Parallel Algorithms in Numerical Linear Algebra," *SIAM Review*, Vol. 20, Oct. 1978, pp. 740-777.

[11]Duff I.S. (Editor), *Sparse Matrices and Their Uses*, Academic Press, New York, 1981.

[12]Dixon, L.C.W., and Bartholomew-Biggs, M.C., "Adjoint-Control Transformations for Solving Practical Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 2, 1981, pp. 365-381.

The Principal Minor Test for Semidefinite Matrices

John E. Prussing*
University of Illinois at Urbana-Champaign
Urbana, Illinois

POSITIVE definite and positive semidefinite (also called nonnegative definite) real quadratic and Hermitian forms play important roles in many control and dynamics applications. A quadratic form $q = x^T Ax$ and its associated real symmetric $n \times n$ matrix A are termed positive definite if $q > 0$ for all $x \neq 0$, and positive semidefinite if $q \geq 0$ for all x. The eigenvalues of a positive definite (semidefinite) matrix are all positive (nonnegative).

Received June 14, 1985.

*Professor, Department of Aeronautical and Astronautical Engineering. Associate Fellow AIAA.