APPROXIMATE SOLUTION OF OPTIMAL CONTROL PROBLEMS USING THIRD ORDER HERMITE POLYNOMIAL FUNCTIONS

by

ERNST D. DICKMANNS* and KLAUS H. WELL†

Abstract

An algorithm for the approximate solution of two point boundary value problems of Class C^2 is given. A simple version having one check point at the center of each polynomial segment results in an algorithm which is easy to program and very efficient. Computer test runs with a Newton-Raphson iterator and numerical differentiation to generate the partial derivatives required show a fast convergence compared to extremal field methods and gradient methods in function space.

Introduction

Optimal control problems with smooth continuous solutions will be treated. They are transformed into mathematical programming problems in two steps. First applying the calculus of variations or the maximum principle a two-point boundary value problem results. This is then solved approximately by parameterization using piecewise polynomial approximations.

Assuming that the frequency content of the solution can be estimated the range of the independent variable is subdivided into sections within which the solution may be well approximated by third order polynomials. For each segment and each variable the four coefficients of the polynomial are determined from the function values at each end - which are the unknown parameters that have to be estimated initially - and from the derivative obtained by evaluating the right hand side of the differential equations with these values. By this the approximating function is continuous and has continuous first derivatives. At one or more check points within each segment the interpolated function values are computed. The derivatives evaluated with these values from the right hand side of the differential equations are then compared to the slope of the interpolating polynomial at this point. The sum of the squares of all these errors plus the errors in the prescribed boundary conditions is chosen as the payoff quantity to be minimized.

---

*Head, †Scientist, Trajectory Section, Institut für Dynamik der Flugsysteme, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., 8031 Oberpfaffenhofen, FRG.

In this paper an algorithm taking one checkpoint in the middle of each segment is developed using a modified Newton-Raphson scheme for iterative parameter adjustment. In connection with the third order polynomial which can be determined from the function values at adjacent gridpoints only (parameters) this leads to especially simple relations. Higher order approximations over more than one segment and more than one check point are of course feasable but not investigated here.

Statement of the Problem

The extremal value of the functional

$$J = \phi(x_f, t_f) \tag{1}$$

under the differential equation constraints

$$\dot{x} = t_f \cdot f(x,u) \qquad \text{(n-vector)} \tag{2}$$

with the control vector u having m components, the initial values

$$x(o) = x_0 \qquad \text{(n-vector)} \tag{3}$$

and the final constraints

$$\psi(x_f, t_f) = 0 \qquad \text{(q-vector)} \tag{4}$$

has to be found.

The solution is assumed to be continuous with continuous first derivatives; t_f is a final time parameter allowing to treat open final time problems in a formulation with the independent variable normalized to the range $0 \le t \le 1$.

Reduction to a Boundary Value Problem

Applying the calculus of variations or the maximum principle [1] the determination of the optimal control is transformed into solving a two-point boundary value problem. The differential equation constraints (2) lead to an additional set of time varying multipliers which are given by

$$\dot{\lambda} = -t_f \cdot \left(\frac{\partial f}{\partial x}\right)^T \lambda \qquad \text{(n-vector)} . \tag{5}$$

The final constraints (4) invoke constant multipliers μ (q-vector) and transversality conditions have to be satisfied

$$T = [\lambda + \phi_x + \mu^T \psi_x]_{t=1} = 0 . \qquad \text{(n-vector)} \tag{6}$$

Here the subscript x means partial differentiation with respect to x. For open final value of the independent variable the Hamiltonian function

$$H = t_f \cdot \lambda^T f \tag{7}$$

has to satisfy the condition

$$R = \left[- \frac{\partial H}{\partial t_f} + \frac{\partial \phi}{\partial t_f} + \mu^T \frac{\partial \psi}{\partial t_f} \right]_{t=1} = 0 \qquad \text{(scalar)} \quad . \tag{8}$$

Eqs. (2) and (5) may be written in the form

$$\dot{z} = g(z)$$

where $z^T = (x^T, \lambda^T)$ is a (2n)-vector which has to satisfy the boundary conditions (3), (4), (6) and (8). This is a nonlinear boundary value problem. The functions z(t), the multipliers μ and the parameter t_f have to be determined.

Parameterization and Iteration Scheme

In figure 1 the basic idea of the algorithm is displayed. Three segments have been chosen (NS = 3) resulting in a total of 4 gridpoints per variable (full dots and empty or full squares). The slope evaluated by introducing the estimated function values into the right hand side of the differential equations (9) is given by solid straight lines at the gridpoints j. The resulting interpolating cubic polynomials are shown as the wavelike solid curves. The function values at the check points are marked by empty circles and the resulting slopes from the differential equations by dashed straight lines. Both the initial estimate and the converged curves are given.

Changing one function value at a gridpoint (z_{22} in fig. 1, empty triangle) affects only the two bordering check points (full triangles), however, for all variable z. For each segment a new time variable $0 \le t' \le (T_{j+1} - T) = \tau$ is introduced which yields as interpolated function value at the (central) check point

$$C_j = \frac{1}{2} \left[z_j + z_{j+1} + t_f \frac{\tau}{4} (f(z_j) - f(z_{j+1})) \right] \tag{10}$$

and

$$\dot{C}_j = \frac{3}{2\tau} (z_{j+1} - z_j) - \frac{t_f}{4} (f(z_j) + f(z_{j+1})) . \tag{11}$$

The difference in the slopes at the check point then is

$$\Delta_j = \frac{3}{2\tau}(z_{j+1} - z_j) - \frac{t_f}{4}(f(z_j) + f(z_{j+1})) - t_f \cdot f(c_j) \qquad (12)$$

(2n-vector).

With this the contribution of the segment j to the convergence measure is

$$S_j = \frac{1}{2}\Delta_j^T \Delta_j \quad . \qquad (13)$$

As total convergence measure the sum

$$M = \sum_{j=1}^{NS} S_j + \frac{1}{2}[\psi^T\psi + T^T T + R^2] \qquad (14)$$

is chosen, where ψ, T and R are given by eqs. (4), (6), and (8). Convergence is considered to be achieved for $M \leq \varepsilon$, where ε is a predetermined small number.

Starting from estimated values z_{ji} for all gridpoints j and variables $i=1...2n$, for the multipliers μ and for the parameter t_f improved values of the total parameter vector

$$p^T = [\lambda_1^T, z_2^t \; ... \; z_{NGP}^T, \mu^T, t_f] \qquad (15)$$

have to be found to drive M towards 0. Using a modified Newton-Raphson iteration scheme the linearized iteration equations may be written

$$\begin{aligned} &\frac{\partial M}{\partial p}\delta p = -\alpha \cdot M \quad , \\ &\left(\frac{\partial \psi}{\partial z}\right)_{NGP} \delta z_{NGP} + \frac{\partial \psi}{\partial t_f}\delta t_f = -\alpha\psi \quad , \\ &\left(\frac{\partial T}{\partial z}\right)_{NGP} \Delta z_{NGP} + \frac{\partial T}{\partial \mu}\delta\mu + \frac{\partial T}{\partial t_f}\delta t_f = -\alpha T \quad , \\ &\left(\frac{\partial R}{\partial z}\right)_{NGP} \delta z_{NGP} + \frac{\partial R}{\partial \mu}\delta\mu + \frac{\partial R}{\partial t_f}\delta t_f = -\alpha R \quad , \end{aligned} \qquad (16)$$

where α is a factor to improve convergence. Taking advantage of the fact that each element S_j of M depends only on the values z_j and z_{j+1} adjacent to it and on the parameter t_f, the total variation of S is

$$dS = \Delta_j^T\left(\frac{\partial \Delta_j}{\partial z_j}\delta z_j + \frac{\partial \Delta_j}{\partial z_{j+1}}\delta z_{j+1} + \frac{\partial \Delta_j}{\partial t_f}\delta t_f\right) \quad . \qquad (17)$$

Introducing this into (16) yields for each segment

$$\frac{\partial \Delta_j}{\partial z_j}\,\delta z_j + \frac{\partial \Delta_j}{\partial z_{j+1}}\,\delta z_{j+1} + \frac{\partial \Delta_j}{\partial t_f}\,\delta t_f = -\alpha \cdot \Delta_j \tag{18}$$

and as set of iteration equations there follows (with $\frac{\partial \Delta_j}{\partial z_k} = A_{j,k}$, and $\frac{\partial \Delta_j}{\partial t_f} = N_j$; for indices see fig. 1).

$$\begin{array}{cc}
\underline{\text{Dim.}} & \begin{array}{cccccccc} n & 2n & \cdots & \cdots & \cdots & 2n & q & 1 \end{array} \\
\begin{array}{c} 2n \\ \vdots \\ \vdots \\ \vdots \\ 2n \\ q \\ n \\ 1 \end{array} &
\begin{bmatrix}
A_{1,1} & A_{1,2} & 0 & 0 & & & & N_1 \\
0 & A_{2,2} & A_{2,3} & 0 & & \mathbf{0} & & N_2 \\
0 & 0 & A_{3,3} & A_{3,4} & & & & \vdots \\
 & & & \ddots & \ddots & & & \vdots \\
 & & & & A_{NS,NS} & A_{NS,NGP} & 0 & N_{NS} \\
 & & & & & \psi_z & 0 & \psi_{t_f} \\
 & & \mathbf{0} & & & T_z & T_\mu & T_{t_f} \\
 & & & & & R_z & R_\mu & R_{t_f}
\end{bmatrix}
\end{array}
\begin{bmatrix} \\ \\ \\ \\ \delta p \\ \\ \\ \\ \end{bmatrix}
= -\alpha \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \vdots \\ \Delta_{NS} \\ \psi \\ T \\ R \end{bmatrix} \tag{19}$$

The submatrices have the following dimensions:

A_{11}	$2n \times n$	$T_\mu = \frac{\partial T}{\partial \mu}$	$n \times q$
$A_{i,j}$	$2n \times 2n$	$T_{t_f} = \frac{\partial T}{\partial t_f}$	$n \times 1$
$N_j = \frac{\partial N_j}{\partial t_f}$	$2n \times 1$	$R_z = \frac{\partial R}{\partial z}$	$1 \times 2n$
$\psi_z = \frac{\partial \psi}{\partial z}$	$q \times 2n$	$R_\mu = \frac{\partial R}{\partial \mu}$	$1 \times q$
$\psi_{t_f} = \frac{\partial \psi}{\partial t_f}$	$q \times 1$	$R_{t_f} = \frac{\partial R}{\partial t_f}$	1×1
$T_z = \frac{\partial T}{\partial z}$	$n \times 2n$		

The partial derivative matrices $A_{j,k}$ are computed by numerical differentiation. The dimension of the linear system of equations (19) is $2n{\times}NS + n + q + 1$. Because of the bidiagonal form in the upper left part it is conveniently reduced for solution to a $(3n + q + 1)$-system independent of the number of segments NS chosen.

Numerical Examples

The algorithm has been tested on a variety of problems such as time minimal accelerated turns of a Hovercraft (n = 2, one control, figure 1), optimal landing approach trajectory of an aircraft (n = 4, one control), maximum lateral range of gliding entry vehicles (n = 5, two controls) and threedimensional skips with prescribed heading change and minimum energy loss at the exit of the atmosphere for the same class of vehicles.

The last problem will be given here. The differential equations are [2, 3]

$$\dot{x}_1 = [-a\cdot b\cdot \exp(-\beta x_4)\cdot(C_{D0}+ku_1^n)x_1^2-G/(R+x_4)^2\cdot\sin x_3]\cdot t_f$$

$$\dot{x}_2 = [a\cdot b\cdot x_1\cdot\exp(-\beta x_4)u_1\sin u_2/\cos x_3-x_1/(R+x_4)\cdot\tan x_5\cdot\cos x_2\cdot\cos x_3]\cdot t_f$$

$$\dot{x}_3 = [a\cdot b\cdot x_1\exp(-\beta x_4)u_1\cos u_2-(G/((R+x_4)^2x_1)-x_1/(R+x_4))\cdot\cos x_3]\cdot t_f \quad (20)$$

$$\dot{x}_4 = [x_1/(R+x_4)\cdot\cos x_3\cdot\sin x_2]\cdot t_f$$

$$\dot{x}_5 = [x_1\sin x_3]\cdot t_f$$

where a = 1/550 m^2/kg, C_{D0} = .04, k = 1., n = 1.86 are vehicle parameters and b = 1.54 kg/m^3, G = $3.9865\cdot 10^5$ km^3/s^2, R = 6371 km, β = .0145 km^{-1} are parameters of the planet and its atmosphere. For the

initial values			and	the final conditions		
$x_1(o)$	= 8.18	km/s		ψ_1	=	$x_2(1)$ - 2.5 = 0
$x_2(o)$	= 0	deg		ψ_2	=	$x_3(1)$ - 1.25 = 0
$x_3(o)$	= -1.25	deg		ψ_3	=	$x_4(1)$ -80.0 = 0
$x_4(o)$	= 80	km				$x_5(1)$ open
$x_5(o)$	= 0	deg				$x_1(1)$ to be maximized

the control time history of the lift coefficient (u_1) and the aerodynamic bank angle (u_2) are to be found which yield the final value of x_1 to be maximal.

Initial estimates were found by linear interpolation between given boundary values or by physical reasoning in the other cases. The convergence behaviour is shown in fig. 2. Table 1 gives the intentionally bad initial estimates and the converged values of the parameters. The estimated initial controls are seen to be very poor. The result achieved with NS = 5 segments is in very good agreement with results obtained by multiple shooting [4] and a refined gradient algorithm based on [5, 6]. Computer time needed was only a fraction (about 1/5 or less) of that of the other methods. Systematic investigations of the radius of convergence are being performed.

Conclusion

An algorithm for the approximate solution of two point boundary value problems of class C^2 has been given. It is based on third order Hermite polynomial approximation. With one check point in the center of each segment it results in an algorithm which is simple to program and very efficient. Numerical test runs with a Newton-Raphson iterator and numerical differentiation to generate the partial derivatives required showed fast convergence compared to extremal field methods and gradient methods in function space.

Literature

[1] Pontryagin,L.S., Boltyansky,V.G., Gamkrelidse,R.V., Mishchenko, E.F.: Mathematische Theorie optimaler Prozesse. Oldenbourg Verlag (1964).

[2] Dickmanns,E.D.: Gesteuerte Drehung von Satellitenbahnen durch Eintauchen in die dichtere Atmosphäre. Dissertation, R.W. T.H. Aachen (1969) also as DFVLR-Sonderdruck No. 42.

[3] Dickmanns,E.D.: Maximum Range Threedimensional Lifting Planetary Entry. NASA TR-382 (1972).

[4] Stoer,J., Bulirsch,R.: Einführung in die Numerische Mathematik II. Springer Verlag, Heidelberger Taschenbücher, Band 114 (1973), pp. 170-191.

[5] Bryson,A.E., Denham,W.F.: A Steepest Ascent Method for Solving Optimum Programming Problems. J. Applied Mech., Vol. 84, No. 2 (1962).

[6] Gottlieb,R.G.: Rapid Convergence to Optimum Solutions Using a Min-H Strategy. AIAA J., Vol. 5 (Feb. 1967), No. 2, pp. 322-329.

[7] Ciarlet,P.G., Schultz,M.H., Varga,R.S.: Numerical Methods of High Order Accuracy for Nonlinear Boundary Value Problems. Numerische Mathematik 9, pp. 394-430 (1967).

→ variables $Z_{j,K}$

j	t/t_f		1	2	3	4	5	6	7	8	9	10	controls	
			$X_1=V$	$X_2=\chi$	$X_3=\gamma$	$X_4=\Lambda$	$X_5=h$	$Z_{j,6}=\lambda_V$	$Z_{j,7}=\lambda_\chi$	$Z_{j,8}=\lambda_\gamma$	$Z_{j,9}=\lambda_\Lambda$	$Z_{j,10}=\lambda_h$	$U_1=C_L$	$U_2=\mu'/\mathrm{deg}$
1		e	8.18	0.	-.021817	0.	80.	1.	7.2578	-4.1313	0	0	.5	120
		c	8.18	0.	-.021817	0.	80.	1.12447	3.1684	1.4157	-1.2005	1.955'-3	.156	65.9
		a	8.18	0.	-.021817	0.	80.	1.12689	3.1635	1.4401	-1.2015	1.965'-3	.156	65.5
2	0.2	e	8.144	.008727	-.013090	.002	80.	1.	7.2270	-4.1729	0	0	.5	120
		c	8.1752	.003715	-.014742	.000107	71.322	1.10666	3.2476	1.1900	-.96595	1.535'-3	.158	69.9
		a	8.1753	.003695	-.014785	.000171	71.301	1.10863	3.2429	1.2125	-.96679	1.542'-3	.158	69.5
3	0.4	e	8.108	.017453	-.004363	.004	80	1.	7.1956	-4.1544	0	0	.5	120
		c	8.1413	.013712	-.005840	.000700	66.388	1.09288	3.3094	1.2778	-.72649	6.153'-4	.167	68.9
		a	8.1414	.013680	-.005879	.000698	66.340	1.09445	3.3049	1.2996	-.72713	6.154'-4	.166	68.5
4	0.6	e	8.072	.026180	.004363	.006	80	1.	7.1637	-4.1360	0	0	.5	120
		c	8.0812	.028585	.005309	.002229	66.200	1.07557	3.3533	1.7994	-.48456	-1.500'-4	.186	61.8
		a	8.0810	.028580	.005333	.002229	66.142	1.07671	3.3490	1.8258	-.48498	-1.507'-4	.186	61.4
5	0.8	e	8.036	.034906	.013090	.008	80.	1.	7.1312	-4.1175	0	0	.5	120
		c	8.0271	.039594	.015655	.004728	71.194	1.04480	3.3795	2.4409	-.24208	4.008'-5	.216	54.1
		a	8.0266	.039609	.015710	.004734	71.168	1.04538	3.3753	2.4677	-.24229	5.327'-5	.216	53.8
6	1.0	e	8.0	.043633	.021817	.01	80	1.	7.0981	-4.0991	0	0	.5	120
		c	7.9958	.043633	.021815	.007738	80.000	1.00000	3.3881	2.8848	0.0000	4.525'-4	.246	49.6
		a	7.9952	.043633	.021817	.007751	80.000	1.00000	3.3839	2.9043	0.0000	4.736'-4	.247	49.4

t_f = 300.0 (e), 288.38 (c), 288.69 sec (a); M = $1.02 \cdot 10^4$ (e), $7.69 \cdot 10^{-8}$ (c)

Table 1: Threedimensional atmospheric skip, numerical example: initial estimate (e); converged solution with presented method (c) and accurate numerical solution by multiple shooting (a)

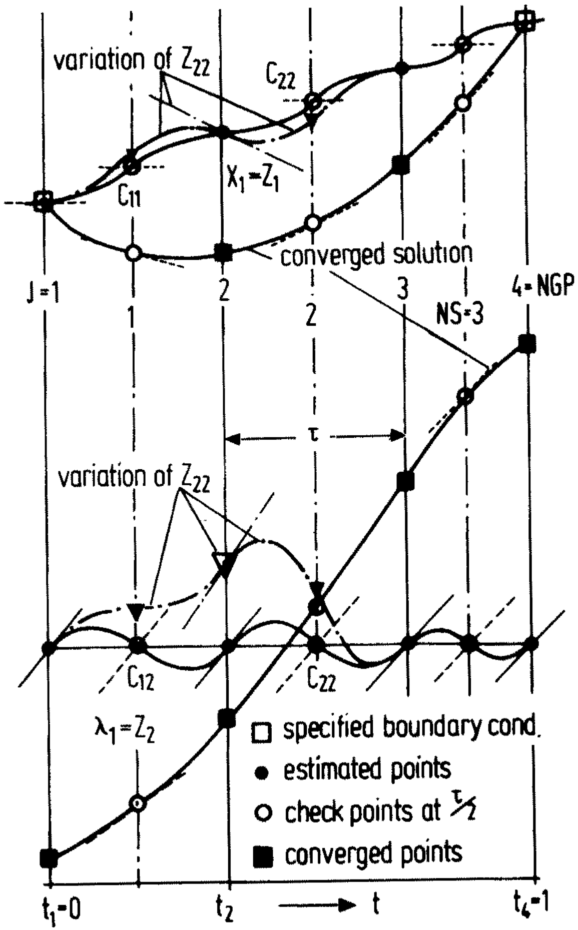

Figure 1: Basic Idea

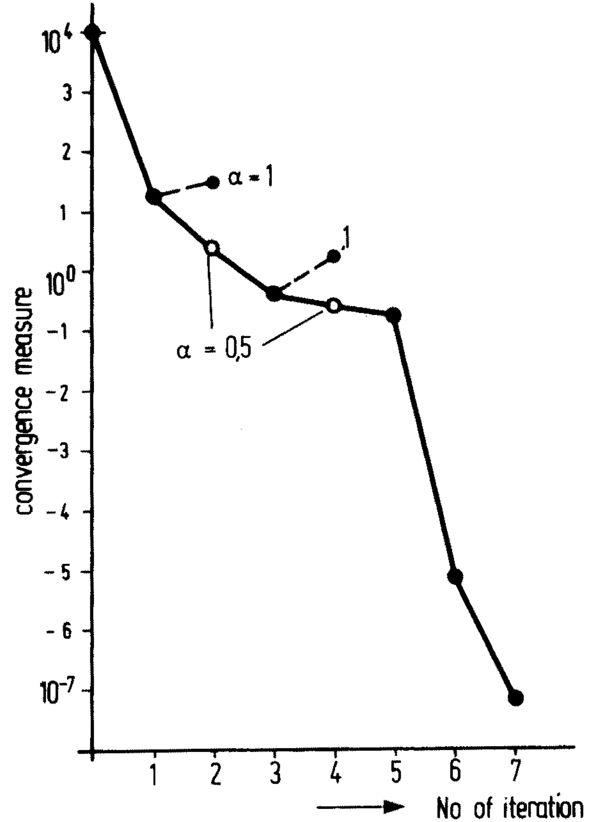

Figure 2: convergence behaviour for three-dimensional atmospheric skip problem