International Journal of Computer and Information Sciences, Vol. 8, No. 5, 1979

Computer Algebra for the Calculus of Variations, the Maximum Principle, and Automatic Control[1]

David R. Stoutemyer[2]

Received June 1977; revised January 1978

This paper describes how a computer-algebra system can solve variational optimization problems analytically. For a calculus-of-variations problem, users provide functional integrands and constraints. A program derives corresponding Euler–Lagrange equations, together perhaps with first integrals. Other programs attempt analytic solution of these equations. For an optimal control problem, users provide analytic expressions for the differential constraints on the state variables. A program determines the corresponding Hamiltonian and differential equations for the auxiliary variables, together with solutions to any trivial auxiliary equations. Other programs attempt analytic solution of the remaining equations while maximizing the Hamiltonian.

KEY WORDS: Optimization; calculus of variations; maximum principle; optimal control; computer symbolic mathematics; MACSYMA.

1. INTRODUCTION

This paper describes how computer algebra can be used to derive the differential equations necessary for stationary values of a constrained or unconstrained function of one or more independent and one or more dependent variables. For simple enough cases, computer algebra is also used to derive a closed-form analytical solution to these equations.

[1] This material is based upon work supported by the National Science Foundation under Grant No. MCS75-22983 A01.

[2] Electrical Engineering Department, University of Hawaii, Honolulu, Hawaii.

0091-7036/79/1000-0419$03.00/0

Section 2 outlines the implemented analytic techniques, and Sec. 3 contains a brief demonstration of the program listed in the Appendix. Section 4 summarizes results for some more comprehensive test cases, with conclusions in Sec. 5.

2. MATHEMATICAL TECHNIQUES

Let $y = (y_1, y_2, \ldots, y_n)^T$ be a function of t, continuous with a continuous first derivative almost everywhere on $a \leqslant t \leqslant b$, and let y satisfy the following boundary conditions, isoperimetric constraints, and differential constraints:

$$y(a) = \tilde{y}(a) \tag{1}$$

$$y(b) = \tilde{y}(b) \tag{2}$$

$$J_j = \int_a^b f_j(t, y, y')\, dt, \qquad (j = 1, 2, \ldots, p) \tag{3}$$

$$f_j(t, y, y') = 0, \qquad (j = p + 1, p + 2, \ldots, q) \tag{4}$$

We seek among these admissable y, the particular ones, denoted $\tilde{y}$, that make the following functional stationary:

$$\int_a^b f_0(t, y, y')\, dt \tag{5}$$

Here a, b, n, p, q, and J_1 through J_p are given constants.

An inequality constraint of the form $\phi_\mu(t, y, y') \leqslant 0$, with $\mu > q$, can be converted to the form of Eq. (4) by adding the square of a new dependent "slack" variable y_j to ϕ_μ: $f_\mu = \phi_\mu + y_j^2 = 0$. Note also that Eq. (4) includes algebraic constraints, where y' is absent.

Denote $(f_0, f_1, \ldots, f_q)^T$ by f and $(\lambda_0, \lambda_1, \ldots, \lambda_q)^T$ by λ, and let the augmented integrand be the inner product

$$F(t, y, y', \lambda) = \lambda^T f \tag{6}$$

where λ_0 through λ_p are unknown constants, and λ_{p+1} through λ_q are unknown functions of t. Also let

$$F_y = \left(\frac{\partial F}{\partial y_1}, \frac{\partial F}{\partial y_2}, \ldots, \frac{\partial F}{\partial y_n}\right)^T$$

$$F_{y'} = \left(\frac{\partial F}{\partial y_1'}, \frac{\partial F}{\partial y_2'}, \ldots, \frac{\partial F}{\partial y_n'}\right)^T$$

where the derivatives are taken as if t, y, and y' were independent. Then $\lambda \neq 0$ and λ is continuous, except possibly where $\tilde{y}'$ is discontinuous. Also, λ and $\tilde{y}$ satisfy the DuBois–Reymond integrodifferential equations

$$F_{y'}(t, \tilde{y}, \tilde{y}', \lambda) = \int_a^t F_y(\tau, \tilde{y}, \tilde{y}', \lambda)\, d\tau + k \tag{7}$$

where k is a vector of integration constants. Wherever $\tilde{y}'$ does not exist, it can be taken consistently as either the left or right derivative, where both are presumed to exist uniquely.

If $\tilde{y}$ and λ satisfy Eq. (7), then so do $\tilde{y}$ and any multiple of λ; so without loss of generality, we may normalize λ to 1, using any vector norm. Alternatively, when there are no differential or algebraic constraints, we may set λ_0 to any arbitrary nonzero constant, such as 1.

Wherever $\tilde{y}'$ exists, Eq. (7) can be differentiated to give the Euler–Lagrange equations:

$$\frac{d}{dt} F_{y'}(t, \tilde{y}, \tilde{y}', \lambda) = F_y(t, \tilde{y}, \tilde{y}', \lambda) \tag{8}$$

When F does not depend explicitly on t, it can be shown that

$$\tilde{y}'^T F_{y'}(t, \tilde{y}, \tilde{y}', \lambda) = F(t, \tilde{y}, \tilde{y}', \lambda) + k_0 \tag{9}$$

where k_0 is a constant. Also, when F does not depend explicitly on some y_i, Eq. (7) gives

$$F_{y_i'}(t, \tilde{y}, \tilde{y}', \lambda) = k_i\,. \tag{10}$$

The program also treats problems with derivatives up to arbitary order m in F, in which case Eq. (8) generalizes to

$$\sum_{j=1}^{m} (-1)^j \frac{d^j}{dt^j} F_{y^{(j)}} = F_j \tag{11}$$

where

$$F_{y^{(j)}} = \left[\frac{\partial F}{\partial \left(\dfrac{d^j y_1}{dt^j}\right)}, \frac{\partial F}{\partial \left(\dfrac{d^j y_2}{dt^j}\right)}, \ldots, \frac{\partial F}{\partial \left(\dfrac{d^j y_n}{dt^j}\right)} \right]^T$$

When integral (5) is a multiple integral over an arbitrary number of independent variables, $t_1, t_2, \ldots, t_r$, the derivative on the left side of Eq. (11) generalizes to a sum over all partial derivatives of total order j. For simplicity, the program treats only the case with no mixed partial derivatives, for which Eq. (11) generalizes to

$$\sum_{j=1}^{m} (-1)^j \sum_{i=1}^{r} \frac{d^j}{dt^j} F_{y_{(i)}^{(j)}} = F_y \tag{12}$$

where

$$F_{y_{(i)}^{(j)}} = \left[\frac{\partial F}{\partial \left(\dfrac{d^j y_1}{dt_i{}^j} \right)}, \frac{\partial F}{\partial \left(\dfrac{d^j y_2}{dt_i{}^j} \right)}, \ldots, \frac{\partial F}{\partial \left(\dfrac{d^j y_n}{dt_i{}^j} \right)} \right]^T$$

Equations (7)–(12) are necessary conditions for stationary values of a functional, rather than sufficient conditions for an optimum. Thus not all solutions derived from these necessary conditions are necessarily optima.

The maximum principle is an approximately equivalent alternative to the calculus of variations. To convert a problem similar to formulas (1)–(5) to a form suitable for the maximum principle:

1. Substitute an additional dependent variable y_α, for every y_i' that occurs in f_0 in (5), and include corresponding constraints, $y_i' = y_\alpha$.

2. Whenever t occurs explictly in formula (3), (4), or (5), substitute another dependent variable, y_β, and include the additional constraint $y_\beta' = 1$, together with the boundary condition $y_\beta(a) = a$.

3. For each instance of Eq. (3), introduce a unique dependent variable y_γ and include the constraint $y_\gamma' = f_j$, together with the boundary conditions $y_\gamma(a) = 0$, $y_\gamma(b) = J_j$.

4. If a and b are considered fixed, introduce an additional dependent variable y_ξ, together with the constraint $y_\xi' = 1$, and the boundary conditions $y_\xi(a) = a$, $y_\xi(b) = b$.

5. If expression (5) includes a term $g_0[a, y(a), b, y(b)]$ outside the integral, introduce another unique dependent variable y_η, replace f_0 with $f_0 + y_\eta$, and include the constraint $y_\eta' = 0$ together with the boundary condition $y_\eta(a) = g_0/(b - a)$.

6. Except for time-optimal problems, introduce a new variable y_0, together with the constraint $y_0' = f_0$.

7. Differentiate any of Eq. (4) that are algebraic rather than differential constraints, and include the algebraic form evaluated at a or at b as a boundary condition.

8. Solve Eq. (4) simultaneously, and combine with any new constraints introduced by the above substitutions, to get all of the differential constraints in the form $y' = \tilde{f}(y)$.

The dependent variables that appear in the derivatives are called *state variables*, and the other dependent variables are called *control* or *decision variables*. Without loss of generality, let u denote the vector of control variables and let x denote the vector of state variables. We can then write the differential constraints or state equations as

$$x' = \tilde{f}(x, u) \tag{13}$$

Let ψ denote a vector of unknown time-dependent auxiliary variables with the same number of components as x; the Hamiltonian is then defined by

$$H[\psi(t), x(t), u(t)] = \psi^T \tilde{f}(x, u) \tag{14}$$

The auxiliary equations are defined by

$$\psi' = H_x \tag{15}$$

where $H_x = (\partial H/\partial x_1, \partial H/\partial x_2, ...)^T$. Then the optimal control maximizes H with respect to u while satisfying the state and auxiliary differential equations, together with the boundary conditions. Moreover, the maximum value of H is nonnegative.

At maxima where $(\partial H/\partial u_1, \partial H/\partial u_2, ...)$ exists uniquely, it is zero. This necessary condition can be used to help determine an analytical solution. Through Lagrange multipliers, elimination, changes of variable, and/or a combinatorial technique, this approach can be extended to the situation where u is subject to equality and/or inequality constraints.[2]

MACSYMA has built-in analytical facilities for differentiation, integration, the solution of nonlinear simultaneous algebraic equations, and the solution of differential equations. These facilities are crucial for the derivation and solution of the equations in this section.

The Appendix lists the MACSYMA program that incorporates the variational mathematical techniques outlined in this section and Stoutemyer[3] contains a thorough discussion of that program.

3. DEMONSTRATION

The interactive MACSYMA computer-algebra language[1] has been used to write a function named *EL* that derives specific instances of the Euler–Lagrange equations from the calculus of variations. MACSYMA has various built-in capabilities that are relevant to this application, including the symbolic operations of differentiation, integration, Laplace transforms, inverse Laplace transforms, solution of algebraic equations, and solution of differential equations. MACSYMA is interactive, prompting the user for each interaction cycle with a unique label beginning with the letter C. The user then types an expression terminated by a semicolon or a dollar sign. MACSYMA generates a simplified version of the expression with a corresponding label beginning with the letter D. This D-labeled expression is printed in a natural two-dimensional format when a semicolon terminator is used, and there may also be intermediate output, either unlabeled or labeled, beginning with the letter E. Any result may be used in subsequent expressions by using its label.

As an example, let $Y(T)$ be any continuous function with a continuous first derivative almost everywhere on $A \leqslant T \leqslant B$, and let $Y(A) = \tilde{Y}(A)$, $Y(B) = \tilde{Y}(B)$. Suppose that we seek among these admissable Y the particular ones that make the following functional stationary:

$$\int_A^B Q(Y)\left[1 + \left(\frac{dY}{dT}\right)^2\right]^{1/2} dT$$

where $Q(Y)$ is a given function. After loading in the function EL and being prompted by the label $C2$, to derive the corresponding Euler–Lagrange equation that these particular Y must obey, we type

$$EL(Q(Y)*SQRT(1 + 'D(Y, T)**2), Y, T)\ \$$$

The automatic response is

$$Q(Y)\,SQRT\left(\left(\frac{dY}{dT}\right)^2 + 1\right) - \frac{Q(Y)\left(\frac{dY}{dT}\right)^2}{SQRT\left(\left(\frac{dY}{dT}\right)^2 + 1\right)} = K_0 \qquad \text{(E2)}$$

$$\frac{d}{dT}\left(\frac{Q(Y)\frac{dY}{dT}}{SQRT\left(\left(\frac{dY}{dT}\right)^2 + 1\right)}\right) = SQRT\left(\left(\frac{dY}{dT}\right)^2 + 1\right)\left(\frac{d}{dY}Q(Y)\right) \qquad \text{(E3)}$$

TIME = 131 msec

Equation E3 is the Euler–Lagrange equation. Equation E2 is a first integral of equation E3, with K_0 being the constant of integration. The computing time, in milliseconds on a DEC KL-10, is also printed. To solve Eq. E2 for dY/dT, we apply the built-in MACSYMA algebraic equation solver as follows:

$$\text{SOLVE}(\text{E2},\ 'D(Y, T));$$

The automatic response is

SOLUTION

$$\frac{dY}{dT} = -\frac{SQRT(Q^2(Y) - K_0^2)}{K_0} \tag{E4}$$

$$\frac{dY}{dT} = \frac{SQRT(Q^2(Y) - K_0^2)}{K_0} \tag{E5}$$

$$\text{TIME} = 246 \text{ msec} \qquad [\text{E4, E5}] \tag{D5}$$

For completeness, we must solve both of these alternatives. Beginning with the second alternative, we apply the built-in MACSYMA differential equation solver as follows:

$$ODE2(\text{E5}, Y, T);$$

The automatic response is

$$\text{TIME} = 921 \text{ msec}$$

$$-\frac{T - K_0 \int \frac{1}{SQRT(Q^2(Y) - K_0^2)}\, dY}{K_0} = C \tag{D6}$$

Here C is another constant of integration. Out of its menu of techniques, $ODE2$ used separation of variables to derive Y as an implicit function of T.

$Q(Y) = 1$ is associated with the Euclidean shortest path (a straight line), $Q(Y) = \sqrt{Y}$ is associated with the stationary Jacobian-action path of a projectile (a parabola), $Q(Y) = Y$ is associated with a minimum-surface body of revolution (a hyperbolic cosine), and $Q(Y) = 1/\sqrt{Y}$ is associated with the minimum-time path of a falling body, starting at $Y = 0$, measured down (a cycloid). For example, to specialize the integrand to the latter case, we use the built-in substitution function as follows:

$$SUBST(Q(Y) = 1/SQRT(Y), \quad D6) \quad \$$$

$$\text{TIME} = 15 \text{ MSEC.}$$

To force a reattempt at integration, we type

D7, INTEGRATE;

The built-in integrator asks a question:

Is K_0 zero or nonzero?

Our answer is

$$\text{NONZERO;}$$

We are then given the answer:

$$\text{TIME} = 319 \text{ msec}$$

$$\frac{T - K_0\left(-\dfrac{SQRT\left(\dfrac{1}{Y} - K_0^2\right)}{K_0^2\left(\dfrac{1}{Y} - K_0^2\right) + K_0^4} - \dfrac{ATAN\left(\dfrac{SQRT\left(\dfrac{1}{Y} - K_0^2\right)}{K_0}\right)}{K_0^3}\right)}{K_0} = C \qquad \text{(D8)}$$

It turns out that differential equation E4 yields the same implicit algebraic equation for $Y(T)$.

Another function named *HAM* has been written that derives the Hamiltonian and auxiliary equations for solution of optimal control problems by the maximum principle. As a classic example, suppose there is a unit mass at position $X = 1$ with arbitrary velocity $V = 0$ at time $T = 0$, and we wish to vary the force F, subject to the constraint $-1 \leqslant F \leqslant 1$, such that the mass arrives at position $X = 0$ with velocity $V = 0$ in minimum time. The force equals the rate of change of momentum, so the motion is governed by the pair of first-order differential equations: $dV/dt = F$, $dX/dt = V$. To derive the associated Hamiltonian and auxiliary equations we type

$$HAM(['D(V, T) = F, \quad 'D(X, T) = V]) \quad \$ \qquad \text{(C9)}$$

The automatic response is

$$AUX_2\, V + AUX_1\, F \qquad \text{(E9)}$$

$$\frac{d}{dT} AUX_1 = -AUX_2 \qquad \text{(E10)}$$

$$\frac{d}{dT} AUX_2 = 0 \qquad \text{(E11)}$$

$$AUX_2 = C_2 \qquad \text{(E12)}$$

$$\text{TIME} = 61 \text{ msec}$$

Equation E9 is the Hamiltonian which must be maximized while satisfying differential Eqs. E10, E11, and those in C9. Equation E12 is the solution of the trivial equation E11, with C_2 being the constant of integration.

To substitute Eq. E12 into Eq. E10 and then solve, we type

$$ODE2(SUBST(\text{E12, E10}), \quad AUX[1], \quad T);$$

The response is

TIME = 60 msec

$$AUX_1 = C - C_2T \tag{D13}$$

To substitute the values of the auxiliary variables into the Hamiltonian, we type

$$SUBST([\text{D13, E12}], \text{E9});$$

with the corresponding response being

TIME = 7 msec

$$C_2V + F\,(C - C_2T) \tag{D14}$$

F should be varied within its constraints so as to maximize this expression for all values of T in the time interval of interest. In this example, clearly F should be $SIGN(C - C_2T)$. Thus, every optimal control is $F = 1$ and/or $F = -1$, with at most one switch between them, when $T = C/C_2$. Since F is piecewise constant, we may combine the equations of motion, solving them by typing

$$ODE2('D(X, T, 2) = F, X, T);$$

to get the response

TIME = 99 msec

$$X = \frac{FT^2}{2} + K2\,T + K1 \tag{D15}$$

Now we can choose the integration constants $K1$ and $K2$ to satisfy the given boundary conditions. Assuming we begin with $F = -1$, we use the built-in MACSYMA initial condition function as follows:

$$IC(SUBST(F = -1, \text{D15}), T = 0, X = 1, 'D(X, T) = 0);$$

The corresponding response is

TIME = 80 msec

$$\left[X = 1 - \frac{T^2}{2}\right] \tag{D17}$$

Assuming we terminate with $F = +1$, we type

$IC(SUBST(F = 1, \mathrm{D15}), T = TFINAL, X = 0, 'D(X, T) = 0);$

for a response of

TIME = 125 msec

$$\left[X = \frac{TFINAL^2}{2} - T\, TFINAL + \frac{T^2}{2}\right] \tag{D19}$$

To determine *TFINAL* and the time *T* at which *F* switches sign, we use substitution to impose the conditions that the two solutions must yield the same position and velocity at that time:

$SUBST(\mathrm{D19}, \mathrm{D17}[1]);$

This yields the equation

TIME = 2 msec

$$\frac{TFINAL^2}{2} - T\, TFINAL + \frac{T^2}{2} = 1 - \frac{T^2}{2} \tag{D20}$$

To solve this equation together with its derivative, we type

$SOLVE([\mathrm{D20}, D(\mathrm{D20}, T)]);$

giving the solutions

TIME = 126 msec

$$[[TFINAL = -2, T = -1], [TFINAL = 2, T = 1]] \tag{D21}$$

For the second of these two solutions, $0 < T < TFINAL$, justifying the assumption that F switches from -1 to 1. F is now completely determined as a function of time. To determine C and C_2 in the representation $F = SIGN(C - C_2T)$, we substitute the second solution into the equation $C - C_2T = 0$:

$SUBST(\mathrm{D21}[2], C - C[2]*T = 0);$

This gives an equation relating C and C_2:

TIME = 8 msec

$$C - C_2 = 0 \tag{D22}$$

As is always the case, one of the auxiliary integration constants is redundant. Consequently, we can set C_2 and C to the same arbitrary negative constant, such as -1.

4. TEST RESULTS

Regarding computer analytic variational optimization, the questions of interest are: Within the available memory space and a reasonable amount of computer time:

1. What are the most complicated problems for which the programs *EL* or *HAM* can derive the Euler–Lagrange or auxiliary differential equations?
2. Of these, what are the most complicated problems for which the analytic differential equation solver can solve these equations?

The following three cases were the most complicated ones found in a modest literature search.

The first case, by Stuiver,[4] is concerned with transient, one-dimensional, compressible gas flow. A, B, and C are known constants; Y is the dependent variable; T is the time; and X is the position along the flow axis. After

$$EL((A - 'D(Y, T) - 'D(Y, X)**2/2)**B*(X + T + C)**2, Y, [T, X])\$$$

is typed, the program output is

$$\frac{D}{DX}\left(-B(X+T+C)^2\frac{DY}{DX}\left(-\frac{\left(\frac{DY}{DX}\right)^2}{2}-\frac{DY}{DT}+A\right)^{B-1}\right)$$
$$+\frac{D}{DT}\left(-B(X+T+C)^2\left(-\frac{\left(\frac{DY}{DX}\right)^2}{2}+A\right)^{B-1}\right)=0 \qquad \text{(E13)}$$

TIME = 1697 msec

Expansion of the derivatives required another 914 msec. Stuiver reports that it required considerably longer than (1697 + 914) msec to derive and check this result by hand. No attempt was made to derive an analytic solution to this partial differential equation.

The second case, by Payne[5] is an optimal spacecraft reentry problem. Too lengthy to reproduce here, the functional and differential constraints occupy about one page of MACSYMA output.

The output of *HAM* occupied about three pages, computed in about 6 sec. A complete closed-form solution is hopeless, so none was attempted.

The third case, by Miele,[6] is concerned with optimal nonsteady flight over a spherical earth in a great-circle plane. Too lengthy to reproduce here, the problem statement occupies about one page of MACSYMA output. The output of *EL* occupies about five pages, computed in about 28 sec. Here too, a complete closed-form solution is hopeless, so no such solution was attempted. However, the computer did reveal two typesetting errors in the published results: the dot time-derivative operator was missing on the left side of two equations.

None of the above three cases taxed the memory capacity or required an undue amount of computation time.

Regarding the second question, here are four examples that typify the complexity limit beyond which closed-form analytic solutions cannot be constructed by MACSYMA:

It required a total of 13 sec to construct the following solution to the hanging-string problem: $\tilde{y} = k_0 \cosh(t/k_0 + c) - \lambda_1$, with $k_0 \sinh(b/k_0 + c) - k_0 \sinh(a/k_0 + c) = L$, where $\tilde{y}(t)$ is the y that minimizes the functional $\int_a^b y[1 + (y')^2]^{1/2}\, dt$, subject to the constraint $L = \int_a^b [1 + (y')^2]^{1/2}\, dt$.

A total of 11 sec were required to construct the solution to the non-uniform beam bending problem: minimize $\int_1^2 [x(y'')^2 + y]\, dx$, with $y(1) = y'(1) = y(2) = 0$, and $y'(2) = 1$. The solution is

$$y = \frac{-\dfrac{49x(24 - 24\log x)}{72\log 2 - 48} + x^3 - \dfrac{6(110\log 2 - 57)\,x^2}{18\log 2 - 12}}{24} + \frac{\left(1 - \dfrac{4(110\log 2 - 57)}{18\log 2 - 12}\right)x}{8} + \frac{-\dfrac{588}{72\log 2 - 48} + \dfrac{3(110\log 2 - 57)}{18\log 2 - 12} - 1}{12}.$$

It required a total of 12 sec to construct the solution to the following problem of a shearing beam on an elastic foundation:

Minimize $\int_0^1 [(y'')^2 + 2(y')^2 + y^2 + y]\,dx$ and $y(0) = y'(0) = y(1) = y'(1) = 0$

The solution is

$$y = \frac{\left(\dfrac{2(e^2 - 2e - 1)}{2e^2 + 4e - 2} + \dfrac{2(e^2 - 2e + 1)}{e^2 + 2e - 1} + 1\right) xe^{-x}}{8} + \frac{\left(\dfrac{e^2 - 2e + 1}{e^2 + 2e - 1} + 1\right) e^{-x}}{4} + \frac{\left(-\dfrac{2e^2 - 2e - 1}{2e^2 + 4e - 2} + \dfrac{2(e^2 - 2e + 1)}{e^2 + 2e - 1} - 1\right) xe^{x}}{8} - \frac{\left(\dfrac{e^2 - 2e + 1}{e^2 + 2e - 1} - 1\right) e^{x}}{4} - \frac{1}{2}$$

It required a total of 15 sec to construct the solution to the following loudspeaker dynamics problem derived from an example[7]:

$$\text{minimize} \int_0^T [(y')^2 - y^2/2 + (q')^2/2 + yq' + q \sin t]dt$$

with

$$y(0) = y'(0) = q'(0) = 0.$$

Let $I(t)$ denote $q'(t)$; the solution is

$$y(t) = e^{-t/2} \left[\frac{\sin\left(\frac{\sqrt{3}\,t}{2}\right)}{\sqrt{3}} - \frac{\cos\left(\frac{\sqrt{3}\,t}{2}\right)}{3} \right] - \frac{2\sin(t)}{5} - \frac{\cos(t)}{5} + \frac{8e^{-t/2}}{15}$$

$$I(t) = e^{-t/2} \left[-\frac{\sin\left(\frac{\sqrt{3}\,t}{2}\right)}{\sqrt{3}} - \frac{\cos\left(\frac{\sqrt{3}\,t}{2}\right)}{3} \right] + \frac{3\sin(t)}{5} - \frac{\cos(t)}{5} + \frac{8e^{-t/2}}{15}$$

For the loudspeaker and beam examples, the success of the differential equation solver was quite sensitive to the initial or boundary values and the numerical coefficients in the functional. It required some experimentation to find values that permitted closed-form solutions.

The difficulty with numerical coefficients is primarily attributable to the fact that without recourse to the cubic and quartic formulas, linear third- and fourth-order systems of differential equations cannot generally be solved in closed form unless some variables or derivatives are missing or unless the equations have constant coefficients that permit exact factoring of the characteristic equation. The difficulty with initial or boundary values is attributable to the fact that even for linear differential equations, imposition of the initial or boundary conditions often involves nonlinear algebraic equations.

5. CONCLUSIONS

Computer algebra is clearly helpful for avoiding the tedium and blunders associated with hand derivation of the necessary differential equations for large variational problems. It is also somewhat helpful for deriving closed-form analytical solutions when they exist.

Stoutemyer[2] describes how a similar program for optimization of a multivariate function proved to be useful as an instructional tool for an optimization course. Analytical treatment of nontrivial problems increased the students' understanding of the theoretical foundations, while nicely complementing numerical experiments. Perhaps the same will prove true for the variational optimization programs.

ACKNOWLEDGMENTS

For their help and encouragement, I thank Richard Bogen, Richard Fateman, Jeffrey Golden, Martin Griss, Ken Harrenstein, Frank Kuo, Joel Moses, Willem Stuiver, and Paul Wang.

APPENDIX

(C3) /* This file contains functions and option settings for variational optimization using the calculus of variations and the maximum principle. For a description of its usage see the text file OPTVAR USAGE. */

/* Set options to automatically print cpu time in milliseconds, force attempted equation solution even when there are more variables than unknowns, when an equation involves logs or exponentials, or when a coefficient matrix is singular: */

```
TIME : GRINDSWITCH : SOLVERADCAN : SINGSOLVE : TRUE$
TIME = 13 MSEC.
```

(C4) /* establish D as an alias for the differentiation function: */

```
ALIAS(D,DIFF)$
TIME = 5 MSEC.
```

```
(C5) HAM(ODES) := BLOCK(
/* This function computes the Hamiltonian & the auxiliary equations */
  [T, NSV, STATEVARS, AUXVARS, ANSW, ELIST, AUXDE],
  /* declare local vars */
  IF NOT LISTP(ODES) THEN ODES: [ODES], /* ensure list argument */
  T: PART(ODES, 1, 1, 2), /* get independent var from derivative */
  NSV: LENGTH(ODES), /* determine number of state variables */
  /* Form list of state and auxiliary variables: */
  STATEVARS: AUXVARS: ELIST: [],
  FOR I THRU NSV DO (STATEVARS: ENDCONS(PART(ODES, 1, 1,
    1), STATEVARS), AUXVARS: ENDCONS(AUX[I], AUXVARS)),
  ANSW: [SUM(RHS(ODES[I])*AUX[I], I, 1, NSV)], /* form Hamiltonian
    */
  /* Form list of auxiliary equations and any trivial solutions: */
  FOR I THRU NSV DO (
    AUXDE: 'DIFF (AUX[I], T) = -DIFF (ANSW[1], STATEVARS[I]),
    ANSW: ENDCONS (AUXDE, ANSW),
    IF RHS (AUXDE) = 0 THEN ANSW: ENDCONS(AUX[I] = C[I],
    ABSW)),
```

```
/* Form list of E-labels while displaying computed results: */
FOR ITEM IN ANSW DO ELIST: ENDCONS(FIRST(DISP(ITEM)),
ELIST), ELIST)$
```

TIME = 102 MSEC.

```
(C6) EL(F, YLIST, TLIST): = BLOCK ( /* This function computes the
Euler–Lagrange equations and any trivial first integrals: */
  [LY, LT, FSUB, ENERGYCON, ANSW, ELIST], /* declare local
    variables */
  IF NOT LISTP(TLIST) THEN TLIST: [TLIST],
  IF NOT LISTP(YLIST) THEN YLIST: [YLIST],
  /* compute number of independent & independent variables: */
  LY: LENGTH(YLIST), LT: LENGTH(TLIST), FSUB: F,
  /* no conservation of energy if more than 1 independent var: */
  ENERGYCON: EV(LT = 1, PRED),
  FOR I THRU LY DO /* substitute for derivatives: */
    FOR J THRU LT DO (DD[I, J]: DERIVDEGREE(FSUB, YLIST[I],
      TLIST[J]),
      IF DD[I, J] > 1 THEN ENERGYCON: FALSE,
      FOR K THRU DD[I, J] DO
        FSUB: SUBST('DIFF(YLIST[I], TLIST[J], K) = DYDT[I, J, K],
        FSUB)),
  /* no conservation of energy if independent var. in integrand: */
  IF NOT FREEOF(TLIST[1], FSUB) THEN ENERGYCON: FALSE,
  ANSW:IFENERGYCON THEN(FSUB)ELSE[],/*form list of results:*/
  FOR I THRU LY DO (FY: DIFF(FSUB, YLIST[I]),
    ANSW: ENDCONS(
      SUM(SUM((-1)**(K - 1)*'DIFF(DIFF(FSUB, DYDT[I, J, K]),
      TLIST[J], K),
      K, 1, DD[I, J]), J, 1, LT) = FY, ANSW),
    IF ENERGYCON THEN ANSW[1]: ANSW[1] -
      DIFF(FSUB, DYDT[I, 1, 1])*'DIFF(YLIST[T], TLIST[1]),
    IF FY = 0 AND LT = 1 AND DD[I, 1] = 1 THEN /* momentum
      integral */
      ANSW: ENDCONS(DIFF(FSUB, DYDT[I, 1, 1]) = K[I], ANSW)),
  IF ENERGYCON THEN ANSW[1]: ANSW[1] = K[0],
  FOR I THRU LY DO /* resubstitute original derivatives: */
    FOR J THRU LT DO
      FOR K THRU DD[I, J] DO
        ANSW: SUBST(DYDT[I, J, K] = 'DIFF(YLIST[I], TLIST[J], K),
        ANSW),
```

```
  ELIST: [], /* form list of E-labels while displaying results: */
  FOR EQN IN ANSW DO ELIST: ENDCONS(FIRST(DISP(EQN)),
  ELIST),
  ELIST) $
TIME = 194 MSEC.

(C7) CONVERT(ODES, YLIST, T) : = BLOCK([ANSW],
/* This function converts output of EL or HAM to form required by
DESOLVE from the file DESOLN. */
  IF NOT LISTP(YLIST) THEN YLIST: [YLIST],
  ANSW: EV(ODES, EVAL),  /* if E-labels, replace with values */
  FOR YY IN YLIST DO ANSW: SUBST(YY = YY(T), ANSW),
  RETURN(ANSW)) $
TIME = 29 MSEC.
```

REFERENCES

1. The Mathlab Group, "*MACSYMA* Reference Manual," Project MAC, Massachusetts Institute of Technology, Cambridge, Massachusetts (1974).
2. D. R. Stoutemyer, "Analytical optimization using computer algebraic manipulation," *ACM Trans. Math. Software* **1**:147–164 (June 1975).
3. D. R. Stoutemyer, "Computer Algebraic Manipulation for the Calculus of Variations, the Maximum Principle, and Automatic Control," ALOHA Project Technical Report A74-5, University of Hawaii (November 1974).
4. W. Stuiver, "On a Variational Approach to the Problems of Propagation of Plane and Spherical Waves," Ph.D. thesis, Division of Engineering Mechanics, Stanford University (December 1959).
5. J. A. Payne, "Computational Methods in Optimal Control Problems, in *Advances in Control Systems*, Vol. 7 (Academic Press, New York, 1969), pp. 73–164.
6. A. Miele, "A Survey of the Problem of Optimizing Flight Paths of Aircraft and Missiles," in R. E. Bellman, ed., *Mathematical Optimization Techniques* (University of California Press, Berkeley, 1963), Chapter 1.
7. S. H. Crandal, D. C. Karnop, E. R. Kurtz, Jr., and D. C. Pridmore-Brown, *Dynamics of Mechanical and Electromechanical Systems* (McGraw-Hill, New York, 1968), pp. 308–311.

Printed in Belgium