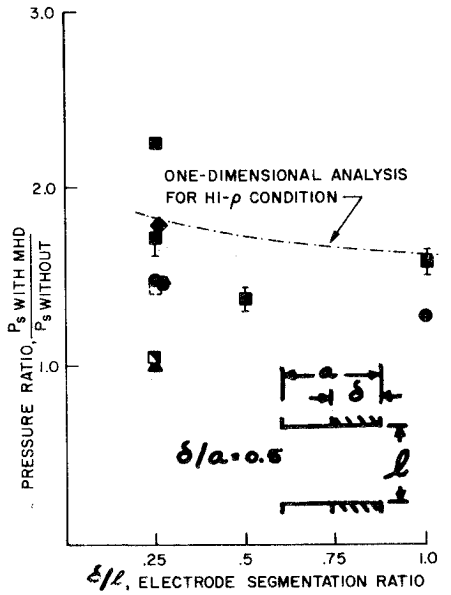

Fig. 6 MHD augmentation based on impact pressure increase, □-Hi-ρ; ○-MEDI-ρ; △-low enthalpy, unseeded, $B = 5.2$; ◇-low enthalpy, seeded, $B = 4.0$; ⬭-low enthalpy, seeded $B = 5.2$; filled symbols—fully powered MHD; dashed square—MHD Aug. @ 50% electric currents; half filled symbols—60% E-Field, no B-Field.

Operation of the facility under so-called "fully powered" conditions (as compared to "underpowered" conditions) eliminated the boundary-layer loss and resulted in an operational efficiency of 75% (somewhat below the 85%–90% value reported in Ref. 2).

Velocity increase data were obtained primarily from the impact pressure measurements. Fig. 5 shows nozzle impact pressure responses, and Fig. 6 is a summary of the wide range of impact data obtained compared to the one-dimensional analysis as a function of variation in MHD channel electrode segmentation ratio. The increase in impact pressure is quite pronounced and the analytically predicted increases in flow velocity were readily achieved experimentally. A tabulation of the various operational and flow conditions shown in Fig. 6 is given in Ref. 9 (Chart II). These conditions cover high and medium flow density (ρ), high and low reservoir enthalpy, and include a very limited study with seeded air using a novel "basket seeder."[9] Under the most optimum conditions used in this experiment, the velocity in the test section increased from 19,600 fps to 24,000 fps due to upstream MHD augmentation.

References

[1] Warren, W. R., et al., "Feasibility Study of High Density Shock Tunnel Augmentation by an MHD Accelerator," AEDC TR-62-225, Oct. 1965, Arnold Engineering Development Center, Tullahoma, Tenn.

[2] Pate, S. R., et al., "Development of an MHD-Augmented, High Enthalpy, Shock Tunnel Facility," *AIAA Journal*, Vol. 12, No. 3, March 1974, pp. 298–297.

[3] Warren, W. R. and Harris, C. J., "A Critique of High Performance Shock Tube Driving Techniques," *Proceedings of the 7th International Shock Tube Symposium*, Toronto, Ontario, Canada, 1969.

[4] Harris, C. J., "Comment on 'Nonequilibrium Effects on High-Enthalpy Expansion of Air'," *AIAA Journal*, Vol. 4, No. 6, June 1966, pp. 1148–49.

[5] Marston, C. H., "MHD Accelerator Performance for Specified Interaction Parameter," *AIAA Journal*, Vol. 4, No. 11, Nov. 1966, pp. 2078–2079.

[6] Harris, C. J., "Experimental Identification of Shock Tunnel Flow Regimes," *AIAA Journal*, Vol. 7, No. 2, Feb. 1969, pp. 365–369.

[7] Harris, C. J., et al., "Measuring Techniques Applied in Determining the Performance of an MHD Augmented Shock Tunnel," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-3, No. 3, May 1967, pp. 433–440.

[8] Harris, C. J., et al., "MHD Generator and Accelerator Experiments in Seeded and Unseeded Air Flows," *Transactions on International Electricity from MHD Symposium*, Vol. 1, International Atomic Energy Agency, Salzburg, 1966.

[9] Harris, C. J., "MHD Augmented Shock Tunnel Study," AEDC TR-68-161, Aug. 1968, Arnold Engineering Development Center, Tullahoma, Tenn.

Interpolation in Numerical Optimization

KENNETH R. HALL* AND DAVID G. HULL†
The University of Texas at Austin, Austin, Texas

A NUMBER of numerical optimization methods, for example, the gradient method, the generalized perturbation method,[1] and quasilinearization, iterate on variable histories. In order to facilitate the updating process from one iteration to the next, a fixed-step, multistep integrator, such as the Adams-Moulton, Adams-Bashforth integrator, is normally used. With a low-order integrator, say fourth order, it may be necessary to use a large number of integration steps to get the accuracy needed to achieve convergence, and this is costly with respect to both computer time and computer storage. At the same time, a large number of integration steps can lead to round-off error in the numerical integration process. The problems previously mentioned can be circumvented by using a variable-step integrator with stepsize control based on local relative truncation error. However, this procedure requires the use of an interpolation method because the pattern of integration steps will differ from one iteration to the next. In addition, the orders of the interpolator and the integrator must be similar to achieve the desired accuracy. One arrangement which has been used successfully[1,2] is the Runge-Kutta-Fehlberg 3(4) integrator[3] and the cubic-spline interpolator. The purpose of this Note is to discuss the cubic-spline interpolator in terms of the requirements imposed by numerical optimization.

In general, the requirements imposed on the cubic-spline interpolator by numerical optimization are that the spline be generated accurately and with the minimum amount of storage. The standard procedure for generating a spline involves the solution of a linear system in which the coefficient matrix is tridiagonal. The linear system can be solved by an elimination method (for example, Gauss elimination) or by an iterative method (for example, successive over-relaxation). With regard to accuracy, both methods yield essentially the same results. However, Gauss elimination requires the storage of three more N-vectors than successive over-relaxation, where N is the number of nodal points. If N is large and if several curves are being interpolated, storage minimization makes iterative methods more attractive than elimination methods. It should be mentioned, however, that iterative methods usually require more computer time.

An algorithm and a code for generating the cubic spline with successive over-relaxation is presented in Ref. 4. However, there are two aspects of the procedure used in Ref. 4 which

Received April 17, 1974; revision received September 10, 1974. This research was supported by NASA/MSC Contract NAS 9-11964.

Index category: Navigation, Control, and Guidance Theory.

* Assistant Professor of Aerospace Engineering and Engineering Mechanics.

† Associate Professor of Aerospace Engineering and Engineering Mechanics. Member AIAA.

can be improved. First, the code is written in such a way that eight N-vectors must be stored in addition to the two N-vectors which define the table being interpolated. By a simple rewriting of the code, the eight N-vectors can be reduced to one. Second, the algorithm of Ref. 4 sets up the linear system in terms of the second derivatives at the nodal points. Then, both the second and third derivatives are used in the interpolation computations. If the first derivatives at the nodal points are needed, additional computations would have to be made. An algorithm can be formulated such that the solution of the linear system yields the first derivatives at the nodal points and that this is the only N-vector which needs to be stored along with the two defining the table. The remainder of this Note presents the equatons for such an algorithm.

Consider the cubic polynomial for the kth interval where the k subscripted quantities are evaluated at the beginning of the interval and the $k+1$ subscripted quantities are evaluated at the end of the interval. Let t denote the value of the independent variable within the kth interval for which the interpolated value of the dependent variable y is desired. Let A, B, C, and D represent the constant coefficients of the cubic polynomial in this interval, and write the following expressions:

$$\begin{aligned} y &= A+B(t-t_k)+(C/2)(t-t_k)^2+(D/6)(t-t_k)^3 \\ y' &= B+C(t-t_k)+(D/2)(t-t_k)^2 \\ y'' &= C+D(t-t_k) \end{aligned} \tag{1}$$

In terms of the notation $h = t-t_k$, $H_k = t_{k+1}-t_k$, and $R = h/H_k$, the expressions which relate the constant coefficients of the cubic to the values of the dependent variable and its first derivative at the ends of the interval are given by

$$\begin{aligned} A &= y_k \\ B &= y'_k \\ C &= (6/H_k^2)(y_{k+1}-y_k)-(2/H_k)(y'_{k+1}+2y'_k) \\ D &= (-12/H_k^3)(y_{k+1}-y_k)+(6/H_k^2)(y'_{k+1}+y'_k) \end{aligned} \tag{2}$$

These values may now be substituted into the expressions for the dependent variable and its derivatives to produce the following relations:

$$\begin{aligned} y &= y_k+(3R^2-2R^3)(y_{k+1}-y_k)+ \\ &\qquad H_k(R-2R^2+R^3)y'_k+H_k(-R^2+R^3)y'_{k+1} \\ y' &= (6/H_k)(R-R^2)(y_{k+1}-y_k)+ \\ &\qquad (1-4R+3R^2)y'_k+(-2R+3R^2)y'_{k+1} \\ y'' &= (6/H_k^2)(1-2R)(y_{k+1}-y_k)+ \\ &\qquad (1/H_k)(-4+6R)y'_k+(1/H_k)(-2+6R)y'_{k+1} \end{aligned} \tag{3}$$

The equations written in this form assure continuity of the dependent variable and its first derivative at the ends of the interval as may be demonstrated by using $R=0$ and $R=1$ in the above equations to obtain values at the beginning of the kth interval and at the end of kth interval. In order to assure continuity of the second derivative between intervals, the value of y'' at the end of the kth interval is equated to the value of y'' at the beginning of the $k+1$st interval. This yields the following recursive relationship for y'_k:

$$\begin{aligned} &H_k y'_{k-1}+2(H_{k-1}+H_k)y'_k+H_{k-1}y'_{k+1} = \\ &\qquad 3(H_{k-1}/H_k)(y_{k+1}-y_k)+3(H_k/H_{k-1})(y_k-y_{k-1}) \end{aligned} \tag{4}$$

The output of the Runge-Kutta-Fehlberg numerical integration process is the table of values of y_k and t_k. For N stepping points (including the first) used by the numerical integrator, N values of y'_k must be determined to construct the cubic spline. The recursive relations of Eq. (4) provide N-2 equations in y'_k. The input of y'_1 and y'_N, the values at the beginning and end of the integration interval, provides a solvable set of equations in y'_k. If conditions are imposed on y'_1 and y'_N, the corresponding conditions on the first derivatives can be obtained from the third of Eqs. (3).

The successive over-relaxation iterative method for solving the linear system is defined by the relation

$$y'_k = y'_k+1.0717968\Delta_k \tag{5}$$

where

$$\Delta_k = \frac{1}{2(H_{k-1}+H_k)}\left[3\frac{H_{k-1}}{H_k}(y_{k+1}-y_k)+3\frac{H_k}{H_{k-1}}(y_k-y_{k-1})- H_k y'_{k-1}-H_{k-1}y'_{k+1}\right]-y'_k \tag{6}$$

To start the iterative process, the values of y'_k corresponding to a quadratic spline are used; that is,

$$y'_k = \frac{H_{k-1}}{H_k+H_{k-1}}\frac{1}{H_k}(y_{k+1}-y_k)+\frac{H_k}{H_k+H_{k-1}}\frac{1}{H_{k-1}}(y_k-y_{k-1}) \tag{7}$$

The iterative process is continued until the change in the values of y'_k at each nodal point between iterations satisfies a prescribed tolerance which should be correlated with the tolerance prescribed for the integrator. Convergence of the iterative method is assured since the coefficient matrix of the linear system is tridiagonal and irreducibly diagonally dominant. Finally, once the values of y'_k are known, interpolated values of y can be computed from the first of Eqs. (3).

From the previous relations, it is apparent that the only quantities which need to be stored during the computation process are t_k, y_k, and y'_k. The values of H_k, H_{k-1}, and Δk can be computed at each nodal point and do not need to be stored as vectors.

The use of variable-step integration with numerical optimization methods which iterate on variable histories requires interpolation. Furthermore, the interpolation method must be generated accurately and must use minimal storage. The procedure suggested here has a guaranteed accuracy and requires the minimum amount of storage.

References

[1] Gottlieb, R. G., Hull, D. G., and Fowler, W. T., "Generalized Perturbation Method for Solving Two-Point Boundary-Value Problems," *AIAA Journal*, Vol. 12, No. 5, May 1974, pp. 648–650.

[2] Hall, K. R., "Numerical Solutions to Interplanetary Trajectory Problems Involving Close Planetary Approaches," Applied Mechanics Research Laboratory Rept. 1060, 1974, University of Texas, Austin, Texas.

[3] Fehlberg, E., "Low-Order Classical Runge-Kutta Formulas with Stepsize Control and Their Application to Some Heat Transfer Problems," Technical Rept. R-315, 1969, NASA.

[4] Ralston, A. and H. S. Wilf, *Mathematical Methods for Digital Computers*, Wiley, New York, 1967, pp. 156–168.

Torsional Vibrations and Stability of Thin-Walled Beams on Continuous Elastic Foundation

C. KAMESWARA RAO* AND A. APPALA SATYAM†
Andhra University, Waltair, A.P., India

Nomenclature

A = area of cross section of the beam
A_i = constants ($i = 1 \ldots 4$)
a_i, B_n = constants ($i = 0 \ldots 4$; $n = 1, 2, \ldots, \infty$)
C_s = torsion constant

Received May 2, 1974; revision received July 16, 1974. The authors wish to thank Prof. P. K. Sarma and Prof. T. Venugopala Rao, Head of the Department of Mechanical Engineering, Andhra University, for their encouragement in the preparation of this paper.

Index categories: Structural Dynamic Analysis; Structural Stability Analysis.

* Associate Lecturer, Department of Mechanical Engineering.
† Post-Graduate Student, Department of Mechanical Engineering.