An Automated Method for Determining the Flutter Velocity and the Matched Point

Kumar G. Bhatia*
NASA Langley Research Center, Hampton, Va.

A new method is developed for the automated determination of flutter solutions and the matched-point flutter density. The method employs a Laguerre iteration technique and closed-form analytical expressions to compute the first and second derivatives of damping with respect to inverse of reduced frequency and of flutter velocity with respect to square root of inverse of air density. The mathematical development of the method is discussed, and the results of applications of the method to an all-movable control surface and transport-type wing are presented. The efficiency and accuracy are compared to the traditional damping-velocity graphical method. The results indicate that the new technique is accurate and fast and is well suited for use in automated flutter design. The use of the method in aeroelastic optimization and potential application to the *p-k* method are discussed.

Nomenclature

A	$= A(k) =$ aerodynamic matrix, see Eq. (1)
AF	$= \rho(AI)$
AI	$= A/k^2 =$ aerodynamic inertia matrix
c	= reference chord length
F	= lowest flutter velocity—airspeed at specified Mach number and standard atmosphere, see Eq. (14)
g	= structural damping
i	$= (-1)^{1/2}$
K	= structural stiffness matrix
k	= reduced frequency $c\omega/v$
M	= structural inertia matrix
NM	= number of modes
RFI	= inverse of reduced frequency
U	= right-hand eigenvector, see Eq. (1)
V	= left-hand eignevector, see Eq. (8)
v	= velocity
v_f	= flutter velocity
λ	$= \omega^2/1 + ig =$ eigenvalue, see Eq. (3)
μ	$= 1/\lambda$
ξ	$= (\rho_o/\rho)^{1/2}$
ρ	= air density
ρ_o	= air density at sea level, standard atmosphere
ω	= harmonic frequency

Superscripts

R and I = denote real and imaginary parts, respectively
T = denotes a matrix transpose

Subscripts

l and m = denote the mode number

Subscripts following a parenthesis denote derivatives.

Presented as Paper 73-195 at the AIAA 11th Aerospace Sciences Meeting, Washington, D.C., January 10–12, 1973; submitted February 9, 1973, revision received October 25, 1973. The author wishes to express his gratitude to R. V. Doggett, Jr., of NASA Langley Research Center, for many valuable discussions during the course of this work. Many thanks are also due to R. N. Desmarais and R. M. Bennett, also of NASA Langley Research Center, for their help, in particular with their unsteady aerodynamics computer program, and the computer plot program used for Figs. 4, 5, 8, and 9.

Index categories: Aeroelasticity and Hydroelasticity; Structural Design, Optimal; Aircraft Vibration.

*NRC-NASA Postdoctoral Resident Research Associate; presently Senior Engineer, Boeing Commercial Airplane Company, Renton, Wash. Member AIAA.

I. Introduction

THE motivation underlying this study was to develop an automated method for determination of flutter solutions for use in automated design and aeroelastic optimization. There is considerable interest and need for developing techniques for automating as much of the aircraft design as possible since an automated procedure can both reduce the cycle time for each design iteration and reduce the man-hour level required.[1] Furthermore, the increasingly complex and heavier new structures may impose substantial weight penalties if a careful attempt is not made to optimize their total weight and performance. In this context, the optimization of complex structures to satisfy flutter requirements throughout some defined flight envelope has received much attention in the recent past, and Refs. 2–6 are a partial listing of the published work in this area. The author's experience has been that one of the problems in aeroelastic optimization is that the computer time required to perform the flutter analyses is a significant (too high) percentage of the total time for the optimization, and the existing methods are not well suited for automated design. The traditional flutter solution procedures such as the k-method and the p-k method[7] require repetitive solutions of a complex eigenvalue problem or a complex determinant for large number of reduced frequencies or velocities. This means that in an automated design process where the flutter velocity will have to be determined many times, it may be necessary to solve the complex eigenvalue problem several hundred or thousand times.

The application of the usual root finding techniques to speed up the flutter solution procedure is made difficult by the very nature of the flutter equation. In the k-method where the variation of damping is considered with respect to the velocity, the root finding techniques will not always work because the damping is not necessarily a single-valued function of the velocity as illustrated in Fig. 1a (e.g., for mode I damping is a triple-valued function at velocity $= v_1$). The damping is, however, a single-valued function of k or $1/k$. Figure 1b shows an example of the g versus $1/k$ plot. The lowest flutter velocity will not necessarily correspond to the first crossing in the g versus $1/k$ plot. This can be easily seen from the definition $1/k = v/c\omega$. Further, it is not possible, in general, to definitely preserve the identity or the ordering of the flutter eigenvalues at different values of the reduced frequency unless

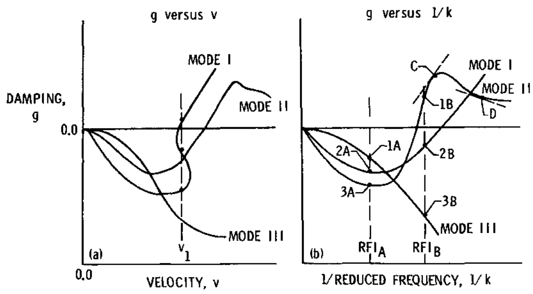

Fig. 1 Comparison of g-v and g-$1/k$ representation.

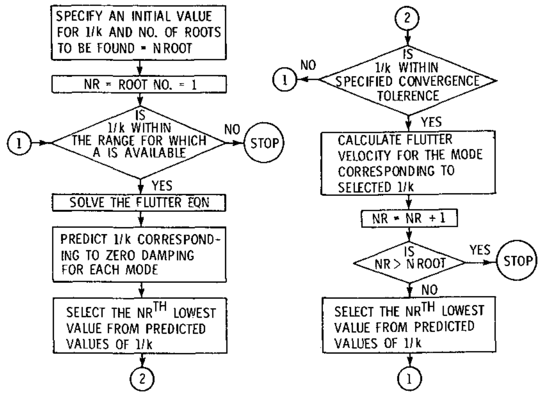

Fig. 2 Basic logic of the method.

the reduced frequency is varied in very small increments. This is illustrated in Fig. 1b which shows that in general the ordering of the damping by magnitude as obtained from the flutter solution (i.e., $1/k = RFI_A$ and $1/k = RFI_B$) does not coincide with the ordering of the associated modes. If an attempt is made to determine the value of $1/k$ at the first crossing by interpolating between RFI_A and RFI_B, several trials will be required. Thus, even when the variation of the damping is considered with respect to the inverse of reduced frequency so that the damping is a single-valued function, the usual root finding techniques such as marching techniques are not always successful.[3] In order to automate and speed up the flutter solution, it is, therefore, desirable to a) express the damping as a single-valued function, and b) to use a root finding technique which does not require preservation of the ordering of the flutter eigenvalues.

The method presented in this paper requires only a few eigensolutions to determine the flutter velocity in an automated manner. It is of the k-method type in principle, and the damping is considered as a function of the inverse of the reduced frequency. The technique may be simply outlined as follows: 1) An initial value of the reduced frequency is assumed and the flutter eigenvalue problem solved. 2) The solution is used in a modified Laguerre iteration to predict a value of $1/k$ corresponding to zero damping. Laguerre's iteration requires the first two derivatives of the damping with respect to $1/k$. 3) The eigenvalue problem is again solved at the k value obtained from step 2. 4) Steps 2 and 3 are repeated until the flutter solution is determined within some specified tolerance. The development of the analysis is discussed in the main body of the paper, and some of the mathematical details are presented in the Appendix. In addition, an extension of the basic method to the determination of so-called matched-point density or matched-point flutter condition (condition at which the flutter velocity, air density, and Mach number are consistent for the standard atmosphere) is developed. Results of the application of the method are also presented and discussed.

II. The New Method

A. Flutter Crossings

The characteristic flutter equation in matrix form is usually expressed as

$$[K - \omega^2 M - \rho(v^2/c^2)A]U = 0$$

or

$$[(k^2M + \rho A) - (c^2/v^2)K]U = 0 \qquad (1)$$

The aerodynamic matrix A obtained from the kernel function, the doublet lattice or the supersonic Mach box method is valid only for harmonic motion, and for a fixed Mach number is a function of k. An artificial structural damping g is now introduced, and Eq. (1) rewritten as

$$[(k^2M + \rho A) - (c^2/v^2)(1 + ig)K]U = 0 \qquad (2)$$

This is the usual form in which the flutter equation is expressed in the k-method. Equation (2) can also be expressed in the following form which is more convenient for the purpose of this paper:

$$[K - \lambda[M + \rho(AI)]]U = 0 \qquad (3)$$

where

$$\lambda = \omega^2/1 + ig$$

The solution to Eq. (3) consists of eigenvalues λ_m, $m = 1, NM$, and the corresponding eigenvectors U_m, $m = 1, NM$. A flutter solution is obtained when the imaginary part of one of the eigenvalues λ_m is zero which means that g_m is zero.

When the damping is considered as a function of the independent variable $1/k$ and a search for the roots is made in the $g - 1/k$ domain, the first root obtained corresponding to the lowest $1/k$ is not necessarily the lowest flutter velocity root. Therefore, it is necessary to find all the roots within a selected range of $1/k$. It will be seen that this is not much of a disadvantage even when one is interested only in determining the lowest flutter velocity. Figure 2 illustrates the basic iterative logic of the new method for determining a value of $1/k$ corresponding to a flutter crossing. No a priori assumptions are made regarding the modes which correspond to flutter, or the mode corresponding to the lowest flutter velocity. This is prudent since it is very difficult to always correctly conjecture the behavior of flutter modes without much detailed study and calculations.

A modified Laguerre iteration is used to find the flutter root (the value of $1/k$ corresponding to zero damping) for each mode. The basic Laguerre method[8] is a powerful, rapidly converging method for polynomials all of whose zeros are real and simple; the convergence is of third order. It also converges for multiple zeros, but the convergence is of first order in the neighborhood of a multiple zero. Laguerre's formula for finding roots of g_m can be written as

$$RFI_m = \frac{1}{k} - \frac{n g_m}{(g_m)_{1/k} \pm [(n-1)\{(n-1)[(g_m)_{1/k}]^2 - n g_m (g_m)_{1/k(1/k)}\}]^{1/2}} \qquad (4)$$

where n = the number of zeros of g_m, RFI_m is a predicted value of $1/k$ for which $g_m = 0$

$$(g_m)_{1/k} = \frac{\partial g_m}{\partial(1/k)}, \quad (g_m)_{1/k(1/k)} = \frac{\partial^2 g_m}{\partial(1/k)^2},$$

and m is understood to range from 1 to NM.

The convergence is of third order if the sign outside the square root is the same as the sign of $(g_m)_{1/k}$. It would appear that a value for n would have to be assumed in Eq.

(4). However, g_m can be regarded as a function of the coefficients of the aerodynamic matrix A which are themselves transcendental functions of k. Thus g_m is considered as a transcendental function of $1/k$ and n is taken to be infinity† in Eq. (4) yielding

$$RFI_m = \frac{1}{k} - \frac{g_m}{\pm\{[(g_m)_{1/k}]^2 - g_m(g_m)_{1/k(1/k)}\}^{1/2}} \quad (5)$$

The modified Laguerre iteration expressed by Eq. (5) has a third-order convergence if the undetermined sign in the second term is selected to be the same as the sign of $(g_m)_{1/k}$ since the convergence is independent of n, for real and simple zeros.[9] To select the sign for the second term of Eq. (5), it is apparent that if a positive sign is selected, then the sign of g_m determines if RFI_m is greater than or less than $1/k$. Since a negative g_m means that a root may exist for a larger $1/k$ and since at that root $(g_m)_{1/k}$ would be positive (Fig. 1), a fixed positive sign is selected, thus

$$RFI_m = \frac{1}{k} - \frac{g_m}{\{[(g_m)_{1/k}]^2 - g_m(g_m)_{1/k(1/k)}\}^{1/2}} \quad (6)$$

Equation (6) is used to predict the zero damping roots and has a third-order convergence to a real and simple zero.

The use of Laguerre's method requires evaluating the first and second derivatives of damping. It is desirable to calculate these derivatives from closed-form analytical expressions rather than from finite-difference formulas to avoid considerations of ordering of the eigenvalues in the flutter equation. The analytical expressions for the damping derivatives are now derived following the approach of Refs. 3-5, and then a procedure for implementing Eq. (6) will be described.

Equation (3) is used as the flutter characteristic equation and is rewritten as

$$\left[K - \frac{1}{\mu_m}[M + \rho(AI)]\right]U_m = 0 \quad (7)$$

where

$$\mu_m = (1/\lambda_m) = (1/\omega_m^2)(1 + ig_m),$$

and U_m is the corresponding eigenvector. Note that $g_m = \mu_m^I/\mu_m^R$. In the following derivation, the associated eigenvectors (so-called left-hand eigenvectors) V_m, $m = 1, NM$ are required. These are obtained from

$$V_m^T\left[K - \frac{1}{\mu_m}[M + \rho(AI)]\right] = 0 \quad (8)$$

The expressions for the first and second derivatives of μ_m are derived in the Appendix, and are

$$(\mu_m)_{1/k} = -\rho k^2 \mu_m [V_m^T(AI)_k U_m] \quad (9)$$

$$(\mu_m)_{1/k(1/k)} = -2k(\mu_m)_{1/k} + \rho k^4 \mu_m [V_m^T(AI)_{kk} U_m] - 2\rho^2 k^4 \mu_m \sum_{\substack{l=1 \\ l\neq 1}}^{NM} \frac{[V_l^T(AI)_k U_m][V_m^T(AI)_k U_l]}{\left(1 - \frac{\mu_m}{\mu_l}\right)} \quad (10)$$

where

$$(\mu_m)_{1/k} = \frac{\partial \mu_m}{\partial(1/k)}, \quad (\mu_m)_{1/k(1/k)} = \frac{\partial^2 \mu_m}{\partial(1/k)^2},$$

$$(AI)_k = \frac{\partial(AI)}{\partial k} \text{ and } (AI)_{kk} = \frac{\partial^2(AI)}{\partial k^2}$$

Since μ_m is a complex variable, it can be written as

†The suggestion for taking $n = \infty$ was made to the author by Robert N. Desmarais of NASA Langley Research Center, Hampton, Va.

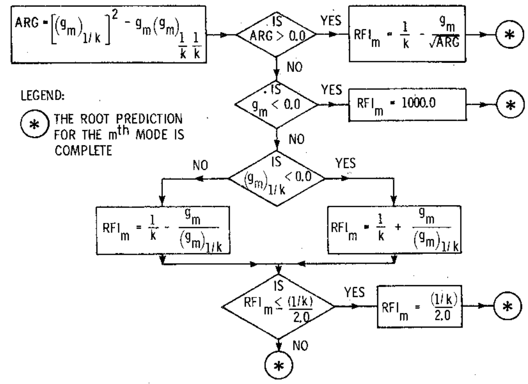

Fig. 3 Block diagram of the logic used for root prediction.

$$\mu_m = \frac{1}{\omega_m^2}(1 + ig_m) = \mu_m^R + i\mu_m^I$$

$$(\mu_m)_{1/k} = (\mu_m^R)_{1/k} + i(\mu_m^I)_{1/k} \quad (11)$$

and

$$(\mu_m)_{1/k(1/k)} = (\mu_m^R)_{1/k(1/k)} + i(\mu_m^I)_{1/k(1/k)}$$

where the superscripts R and I indicate the real and imaginary part, respectively. The derivatives of g_m can be expressed in terms of μ_m and its derivatives by the following:

$$(g_m)_{1/k} = \frac{1}{\mu_m^R}[(\mu_m^I)_{1/k} - g_m(\mu_m^R)_{1/k}] \quad (12)$$

and

$$(g_m)_{1/k(1/k)} = \frac{1}{\mu_m^R}[(\mu_m^I)_{1/k(1/k)} - g_m(\mu_m^R)_{1/k(1/k)} - 2(g_m)_{1/k}(\mu_m^R)_{1/k}] \quad (13)$$

Equations (7-13) allow computation of first and second derivatives of damping for a given value of $1/k$ or k without recourse to finite-difference approximation, and therefore the ordering of eigenvalues of Eq. (7) is not relevant. It should be noted that it is not possible to derive similar expressions for damping derivatives with respect to the velocity for the v-g method since the damping is not a single-valued function of the velocity.

When Eq. (6) is implemented to calculate projected values for the crossings, the argument of the square root may be negative for some modes. This requires special consideration and Fig. 3 briefly presents the manner in which a predicted root from the Laguerre formula is presently handled. Advantage is taken of the fact that if for a given $1/k$, g_m is positive then a crossing exists for a smaller $1/k$. Equation (6) is the primary formula used to predict a crossing, but if it predicts a complex zero of g_m (i.e., a complex RFI_m) then the sign of g_m is checked. If g_m is not positive, an arbitrarily large value is assigned to RFI_m (e.g., $RFI_m = 1000$). If g_m and $(g_m)_{1/k}$ are positive (e.g., at C in Fig. 1b) then a Newton-Raphson projection is used; if only g_m is positive (e.g., at D in Fig. 1b) then RFI_m can be assigned a positive value less than $1/k$. The above precautions are necessary to insure that the program proceeds toward the rth root which requires (Fig. 2) that there should be $(r - 1)$ roots at values of $1/k$ less than or equal to the current $1/k$. It is expected that the logic corresponding to these precautions will seldom be required.

B. Matched-Point Density

When Eq. (7) is solved for an assumed air density ρ, the flutter velocities obtained, if any, will in general not be compatible with the airspeed corresponding to the assumed air density and the fixed Mach number for which

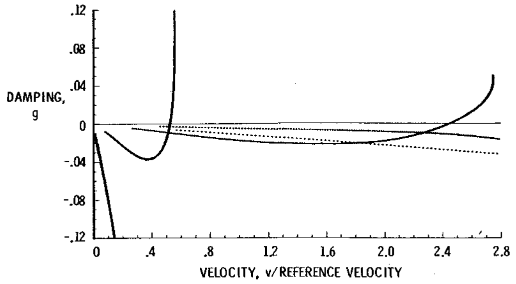

Fig. 4 Variation of damping with velocity for the all-movable control surface.

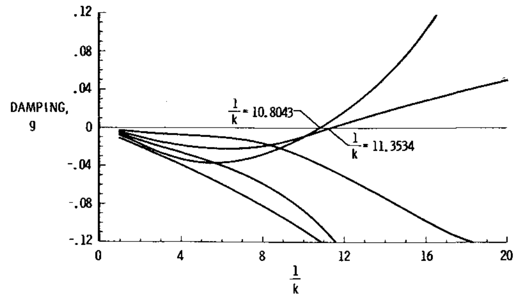

Fig. 5 Variation of damping with 1/reduced frequency for the all-movable control surface.

the aerodynamic matrix A was obtained. Usually, one is interested in finding the matched-point density at which the lowest flutter velocity is compatible with the Mach number. The procedure developed here is general and can be used to determine the matched-point density, if it exists, for a flutter velocity other than the lowest, but the development will be presented here for the lowest flutter velocity.

The procedure used for finding the matched-point density is similar to the one used for determining the flutter crossings. The flutter velocity is considered as a function of $\xi = (\rho_0/\rho)^{1/2}$ where ρ_0 is the sea-level air density and ρ is the air density at any altitude. This functional representation is adopted to take advantage of the fact that for many realistic structures the variation of flutter velocity with ξ is almost linear over a large range of ξ. It should be noted, however, that the method presented here is not restricted to the linear variation of flutter velocity with ξ. The airspeed (speed of sound times Mach number) is a continuous function of ξ but its derivatives with respect to ξ are not continuos at the geometric altitudes of 11,100 meters (36,200 ft), 20,000 meters (65,800 ft), and so forth (U.S. Standard Atmosphere, 1962). A function F is now defined as

$$F = F(\xi) = \text{the lowest flutter velocity} - \text{the airspeed} \tag{14}$$

F is assumed as a continuous function of ξ with continuous first two derivatives except at several known values of ξ. It is apparent that the zeros of F will yield the matched-point densities. If Laguerre's formula is used, then a predicted zero of F, ξ_p, is given by

$$\xi_p = \xi - \frac{nF}{(F)_\xi \pm (H)^{1/2}} \tag{15}$$

where

$$H = (n-1)\{(n-1)[(F)_\xi]^2 - nF(F)_{\xi\xi}\} \tag{16}$$

$$(F)_\xi = (\partial F/\partial\xi) \text{ and } (F)_{\xi\xi} = (\partial^2 F/\partial\xi^2)$$

The sign preceding the square root is taken to be the same as the sign of $(F)_\xi$. Since there would usually be no more than two matched points, n can be assumed to be 2. Thus

$$\xi_p = \xi - \frac{2F}{(F)_\xi + \text{Sign}[(F)_\xi]\{[(F)_\xi]^2 - 2F(F)_{\xi\xi}\}^{1/2}} \tag{17}$$

Equation (17) requires the first two derivatives of F with respect to ξ which can be obtained from the derivatives of the flutter velocity and the airspeed. The derivatives of the airspeed can be easily obtained from representing the speed of sound by different polynomial functions of ξ with each representation corresponding to a range of ξ (or altitude) where the speed of sound is a smooth function. The expressions for the derivatives of velocity are derived following Refs. 3–5, and the final results are given below

$$(k)_\xi = -I1/I2 \tag{18a}$$

$$(\lambda_m)_\xi = \lambda_m[R1 + (k)_\xi]R2 \tag{18b}$$

$$(v_f)_\xi = v_f\left\{\frac{(\lambda_m)_\xi}{2\lambda_m} - \frac{1}{k}(k)_\xi\right\} \tag{18c}$$

$$(k)_{\xi\xi} = -(I3 + I4 + I5)/I6 \tag{19a}$$

$$(\lambda_m)_{\xi\xi} = R3 + R4 + R5 + (k)_{\xi\xi}R6 \tag{19b}$$

and

$$(v_f)_{\xi\xi} = (v_f)_\xi\left[\frac{(\lambda_m)_\xi}{2\lambda_m} - \frac{1}{k}(k)_\xi\right] + v_f\left\{\frac{1}{2\lambda_m}\left[(\lambda_m)_{\xi\xi} - \frac{1}{\lambda_m}[(\lambda_m)_\xi]^2\right] + \frac{1}{k}\left[\frac{1}{k}[(k)_\xi]^2 - (k)_{\xi\xi}\right]\right\} \tag{19c}$$

where

$$R1 + iI1 = \frac{2\rho}{\xi}[V_m{}^T(AI)U_m] \tag{20a}$$

$$R2 + iI2 = -\rho V_m{}^T(AI)_k U_m \tag{20b}$$

$$R3 + iI3 = \frac{2}{\lambda_m}[(\lambda_m)_\xi]^2 \tag{20c}$$

$$R4 + iI4 = 2\rho^2\lambda_m \sum_{\substack{l=1\\ l\neq m}}^{NM}\left(\frac{1}{1-\frac{\lambda_l}{\lambda_m}}\right)\Big\{[(k)_\xi]^2[V_l{}^T(AI)_kU_m][V_m{}^T(AI)_kU_l] - \frac{2(k)_\xi}{\xi}[[V_l{}^T(AI)_kU_m][V_m{}^T(AI)U_l] + [V_l{}^T(AI)U_m][V_m{}^T(AI)_kU_l]] + \frac{4}{\xi^2}[V_l{}^T(AI)U_m][V_m{}^T(AI)U_l]\Big\} \tag{20d}$$

$$R5 + iI5 = -\rho\lambda_m\Big\{\frac{6}{\xi^2}[V_m{}^T(AI)U_m] - \frac{4(k)_\xi}{\xi}[V_m{}^T(AI)_kU_m] + [(k)_\xi]^2[V_m{}^T(AI)_{kk}U_m]\Big\} \tag{20e}$$

$$R6 + iI6 = -\rho\lambda_m[V_m{}^T(AI)_kU_m] \tag{20f}$$

An outline of the derivation of the above equations is included in the Appendix. In Eqs. (18–20), k, λ_m, and v_f correspond to a flutter crossing where g_m has been determined to be zero within a small tolerance, usually of the order of 10^{-3} or 10^{-4}. The above equations seem formidable, but their evaluation of the computer is quite fast since most of the triple products and the derivatives of the aerodynamic inertia matrix can be saved from the

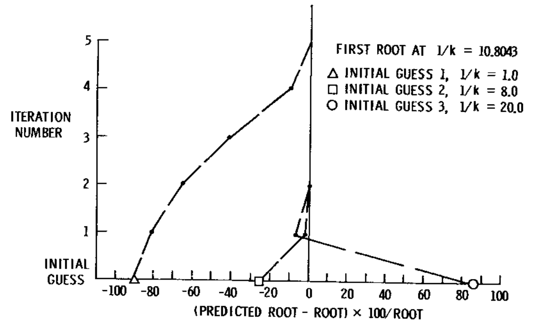

Fig. 6 Convergence to the first root from different initial values of $1/k$ for the all-movable control surface.

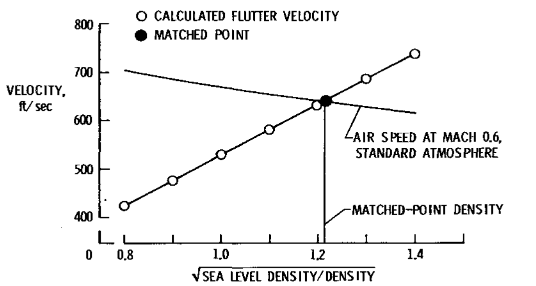

Fig. 7 Matched-point density for the all-movable control surface.

flutter solution program. The iteration defined by Eq. (17) has a third-order convergence and it is expected that in most applications it should not take more than two iterations to determine a matched-point density.

III. Applications and Results

The results of application of the new method are presented in this section. The method has been applied to two realistic configurations. The first is an all-movable control surface with a concentrated mass located at the leading edge of the tip, and the second is a twin-engine jet transport wing. Subsonic kernel function aerodynamics are used in both cases. The aerodynamic matrices were evaluated for a discrete number (40 in both cases) of values of k. Third-order natural spline was then used to interpolate the aerodynamic matrices and the first two derivatives with respect to k at equal increments of $1/k$. For the first case, the aerodynamic matrices were interpolated for a $1/k$ range of 1.0 to 20.0 at 0.05 increment, and for the second case for a $1/k$ range of 0.2 to 5.2 at 0.01 increment. The aerodynamic data are stored in the computer on a random access basis for fast and easy retrieval during program execution. The $1/k$ increment determines the convergence tolerance when the Laguerre iteration searches for a flutter crossing (Fig. 1) since the logical convergence tolerance on $1/k$ is one-half the $1/k$ increment at which aerodynamic data are available.

The variations of damping with velocity and with the inverse of reduced frequency are presented in Figs. 4 and 5, respectively, for the all-movable control surface. Five modes were used in this problem. Note that there are two flutter crossings. The data plotted were obtained by solving the flutter determinant at 381 k values to obtain good graphical representation; in practice, the flutter solution by v-g method may be obtained by solving the flutter determinant for a lesser number of k values depending upon individual judgment. The results of application of the new method for finding the first flutter crossing (corresponding to the lowest $1/k$ root) are shown in Fig. 6. The convergence characteristics of the method are shown for three different initial guesses for $1/k$. If the initial guess for $1/k$ is within about -40% to 85% of the value of $1/k$ at the first crossing, then at most two iterations are required to converge to the first crossing. Since the method always finds the lowest $1/k$ crossing first, the convergence to the second crossing is independent of the initial guess. For this example, the second crossing which is at $1/k$ = 11.3534 was found in one additional iteration after the program converged to the first crossing.

In Fig. 7, the variation of the lowest flutter velocity and the airspeed at Mach number of 0.6 (standard atmosphere) is shown with respect to the square root of density ratio $\xi = (\rho_0/\rho)^{1/2}$. The matched-point density is determined from Fig. 7 to be at $\xi = 1.215$. Using the new method, an initial guess of $\xi = 1$ was used as the starter. The program calculated the lowest flutter velocity (at ξ = 1), and predicted matched-point density at $\xi = 1.212$. The relevant airspeed and the lowest flutter velocity at ξ = 1.212 were then determined and were found to differ by 0.0065%, and the program converged to the matched point and terminated. Similar results were obtained for an initial guess of $\xi = 1.4$.

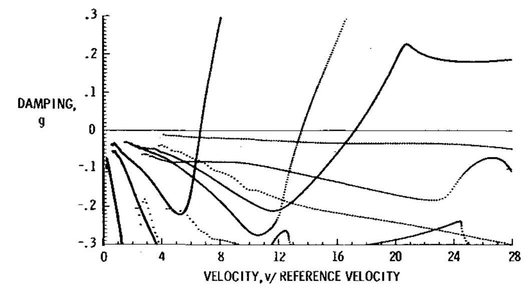

Fig. 8 Variation of damping with velocity for the twin-engine jet transport wing.

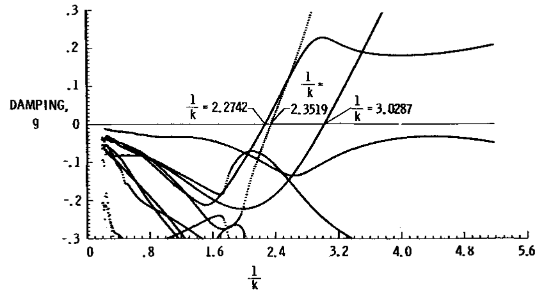

Fig. 9 Variation of damping with 1/reduced frequency for the twin-engine jet transport wing.

Figures 8 and 9 show variation of the damping with the velocity and $1/k$, respectively, for a twin-engine jet transport wing. Although 10 modes were used in the calculations, results from some of the modes are not shown because of the scale selected. The data plotted were obtained by solving the flutter determinant at 501 k values for the two figures. There are three flutter crossings, and the new method is used to determine these crossings. The convergence of the method to the first crossing is graphically shown in Fig. 10. The excellent convergence of the

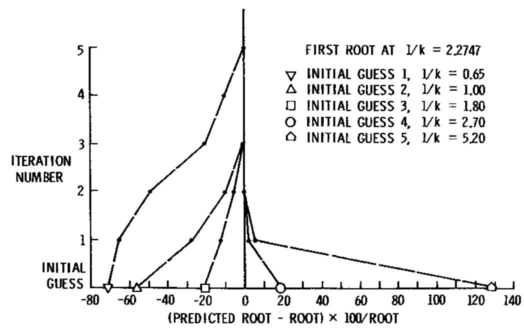

Fig. 10 Convergence to the first root from different initial values of $1/k$ for the twin-engine jet transport wing.

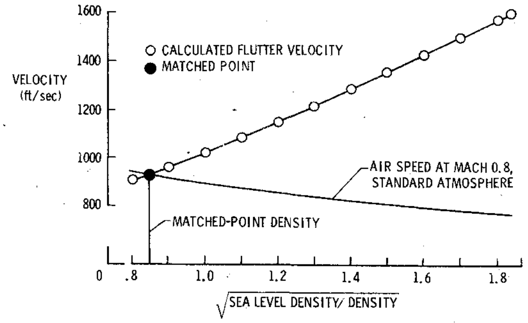

Fig. 11 Matched-point density for the twin-engine jet transport wing.

method is highlighted by the initial guess for $1/k$ which is about 130% more than the actual root. For this example, the method will converge in at most three iterations for initial guesses of $1/k$ which are about −60% to 130% different from the actual root. The convergence to the second crossing at $1/k = 2.3519$ was obtained one additional iteration beyond the first root, and the third crossing at $1/k = 3.0287$ in two additional iterations beyond the second root. The matched point is determined graphically in Fig. 11 at $\xi = 0.847$. (Further accuracy could not be obtained with a graphical solution.) The program was run with the initial density corresponding to $\xi = 1$, and at the first iteration the matched point was predicted at $\xi = 0.8479$. Thus the program converged in one iteration with the lowest flutter velocity and the relevant airspeed differing by 0.0265%. For an initial $\xi = 1.837$, at the first iteration the matched point was predicted at $\xi = 0.8776$, and the program converged at the second iteration to $\xi = 0.8480$ with the lowest flutter velocity and the relevant airspeed differing by 0.0292%.

It is evident from the results obtained that the method has good convergence properties, and the convergence to the zero damping crossings or roots and to the matched-point density is achieved in a small number of iterations.

The CPU time required on the CDC 6600 computer for one iteration to obtain a predicted crossing is approximately 0.145 sec for the 5-mode problem, and 0.945 sec for the 10-mode problem. In this case one iteration involves solving the eigenvalue problem twice, calculating the derivatives, and determining the predicted crossings with the Laguerre formula. The CPU time for calculating a matched-point prediction is about 0.142 sec for the 5-mode problem and about 0.274 sec for the 10-mode problem. Opening of the random access file took 4.866 and 22.378 sec for the 5-mode case and the 10-mode case, respectively. Thus the small number of iterations required, combined with relatively short CPU time per iteration, makes the method very efficient.

IV. Conclusions

The method presented here for the determination of flutter crossings and the matched-point density is shown to be efficient and is completely automated. The calculation of the derivatives of the aerodynamic matrices at different values of the reduced frequency is required in this method. However, interpolation schemes are presently employed in most of the unsteady aerodynamics programs to determine the aerodynamic matrices at large number of reduced frequencies. The additional calculations required to generate the first two derivatives of the aerodynamic matrices with respect to the reduced frequency involve little extra time. This extra time is more than compensated by the shorter computer times required to obtain the flutter solutions and the matched-point density.

In the two examples presented to illustrate the application of the new method, all the flutter crossings in a specified $1/k$ range were determined. It is possible, however, to only determine any desired crossing(s) in the $g - 1/k$ plane. This feature is of special significance for the automated flutter design where it might be desired to only determine the crossing corresponding to the lowest flutter velocity. If it is known from previous analyses that a certain crossing corresponds to the lowest flutter velocity, there may not be any need to determine all the crossings.

Preliminary work done indicates that it may be possible to apply the approach developed here to the *p-k* method[7] of the flutter solution. The resulting savings in the computer time from a successful application of this approach would be even more substantial for the *p-k* method than for the k-method. However, further work is needed to demonstrate the application to the *p-k* method.

References

[1]McComb, H. G., Jr., "Automated Design Methods in Structural Technology," *Proceedings of Conference on Vehicle Technology for Civil Aviation—The Seventies and Beyond,* NASA SP-292, 1971, pp. 225–243.

[2]Turner, M. J., "Optimization of Structures to Satisfy Flutter Requirements," *AIAA Journal,* Vol. 7, No. 5, May 1969, pp. 945–951.

[3]Bhatia, Kumar G., "Optimization of Structures to Satisfy Aeroelastic Constraints," doctoral dissertation, Dec. 1971, Clemson University, Department of Mechanical Engineering, Clemson, S.C.

[4]Rudisill, C. S. and Bhatia, K. G., "Optimization of Complex Structures to Satisfy Flutter Requirements," *AIAA Journal,* Vol. 9, No. 8, Aug. 1971, pp. 1487–1491.

[5]Rudisill, C. S. and Bhatia, K. G., "Second Derivatives of the Flutter Velocity and the Optimization of Aircraft Structures," *AIAA Journal,* Vol. 10, No. 12, Dec. 1972, pp. 1569–1572.

[6]Stroud, W. J., Dexter, C. B., and Stein, M., "Automated Preliminary Design of Simplified Wing Structures to Satisfy Strength and Flutter Requirements," TN D-6534, 1971, NASA.

[7]Hassig, H. J., "An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration," *Journal of Aircraft,* Vol. 8, No. 11, Nov. 1971, pp. 885–889.

[8]Ralston, A., *A First Course in Numerical Analysis,* McGraw-Hill, Inc., New York, 1965, pp. 368–370.

[9]Wilkinson, J. H., *The Algebraic Eigenvalue Problem,* Clarendon Press, Oxford, England, 1965, pp. 443–445.

Appendix: Expressions for the Derivatives

The expressions for the first two derivatives of the damping with respect to the inverse of reduced frequency and of the flutter velocity with respect to the square root of density ratio are derived. The derivation is similar to Refs. 3–5, and the interconnecting steps in the following derivation are kept to the minimum. The basic flutter equations can be expressed as

$$[K - \lambda_m[M + \rho(AI)]]U_m = 0 \tag{A1}$$

$$V_m{}^T[K - \lambda_m[M + \rho(AI)]] = 0 \tag{A2}$$

The eigenvectors are normalized such that

$$U_m{}^T U_m = 1$$

and

$$V_m{}^T[M + \rho(AI)]U_m = 1$$

Differentiating Eq. (A1) with respect to k and premultiplying it by $V_m{}^T$ yields by virtue of Eq. (A2)

$$(\lambda_m)_k = -\lambda_m[V_m{}^T(AF)_k U_m] \tag{A3}$$

where $AF = \rho(AI)$, and the subscript after the parentheses denotes partial derivative of variable in the parentheses with respect to the variable appearing as the subscript, thus

$$(\lambda_m)_k = \frac{\partial \lambda_m}{\partial k}, \text{ etc}$$

From Eq. (A3) one can derive

$$(\mu_m)_{1/k} = -k^2\mu_m[V_m{}^T(AF)_kU_m] \quad \text{(A4)}$$

where $\mu_m = 1/\lambda_m$. To derive an expression for the second derivative of λ_m with respect to k, $(\lambda_m)_{kk}$, the procedure developed in Ref. 5 is followed and the following result obtained

$$(\lambda_m)_{kk} = 2\lambda_m{}^2\sum_{\substack{l=1\\l\neq m}}^{NM}\frac{[V_m{}^T(AF)_kU_l][V_l{}^T(AF)_kU_m]}{(\lambda_m-\lambda_l)} + \frac{2}{\lambda_m}[(\lambda_m)_k]^2 - \lambda_m[V_m{}^T(AF)_{kk}U_m] \quad \text{(A5)}$$

The second derivative of μ_m with respect to $1/k$, $(\mu_m)_{1/k1/k}$, is expressed as

$$(\mu_m)_{1/k(1/k)} = -2k(\mu_m)_{1/k} + \frac{2}{\mu_m}[(\mu_m)_{1/k}]^2 - k^4\mu_m{}^2(\lambda_m)_{kk} \quad \text{(A6)}$$

Substituting $(\lambda_m)_{kk}$ from Eq. (A5) in Eq. (A6) yields after some simplification

$$(\mu_m)_{1/k(1/k)} = -2k(\mu_m)_{1/k} + k^4\mu_m[V_m{}^T(AF)_{kk}U_m] - 2k^4\mu_m\sum_{\substack{l=1\\l\neq m}}^{NM}\frac{[V_l{}^T(AF)_kU_m][V_m{}^T(AF)_kU_l]}{\left(1-\dfrac{\mu_m}{\mu_l}\right)} \quad \text{(A7)}$$

Since density is held constant, Eqs. (A4) and (A7) can be written as

$$(\mu_m)_{1/k} = -\rho k^2\mu_m[V_m{}^T(AI)_kU_m] \quad \text{(A8)}$$

$$(\mu_m)_{1/k(1/k)} = -2k(\mu_m)_{1/k} + \rho k^4\mu_m[V_m{}^T(AI)_{kk}U_m] - 2\rho^2k^4\mu_m\sum_{\substack{l=1\\l\neq m}}^{NM}\frac{[V_l{}^T(AI)_kU_m][V_m{}^T(AI)_kU_l]}{(1-\mu_m/\mu_l)} \quad \text{(A9)}$$

To find the derivatives of the flutter velocity with respect to $\xi = (\rho_o/\rho)^{1/2}$, first $(\lambda_m)_\xi$ and $(\lambda_m)_\xi$ are obtained in exactly the same manner as Eqs. (A3) and (A5). Thus

$$(\lambda_m)_\xi = -\lambda_m[V_m{}^T(AF)_\xi U_m] \quad \text{(A10)}$$

$$(\lambda_m)_{\xi\xi} = \lambda_m{}^2\sum_{\substack{l=1\\l\neq m}}^{NM}\frac{[V_m{}^T(AF)_\xi U_l][V_l{}^T(AF)_\xi U_m]}{(\lambda_m-\lambda_l)} + \frac{2}{\lambda_m}(\lambda_m)_\xi{}^2 - \lambda_m[V_m{}^T(AF)_{\xi\xi}U_m] \quad \text{(A11)}$$

Since

$$AF = \rho(AI),$$

$$(AF)_\xi = -(2\rho/\xi)(AI) + \rho(AI)_k(k)_\xi \quad \text{(A12)}$$

and

$$(AF)_{\xi\xi} = \frac{6\rho}{\xi^2}(AI) - \frac{4\rho}{\xi}(AI)_k(k)_\xi + \rho(AI)_{kk}[(k)_\xi]^2 + \rho(AI)_k(k)_{\xi\xi} \quad \text{(A13)}$$

After substituting $(AF)_\xi$ in Eq. (A10) and $(AF)_\xi$ and $(AF)_{\xi\xi}$ in Eq. (A11), the imaginary parts in these two equations are set to zero yielding formulas for $(k)_\xi$ and $(k)_{\xi\xi}$ expressed by Eqs. (18a) and (19a). This leads directly to Eqs. (18b) and (19b) for $(\lambda_m)_\xi$ and $(\lambda_m)_{\xi\xi}$, respectively. The expressions for $(v_f)_\xi$ and $(v_f)_{\xi\xi}$ are obtained by differentiating $v_f = c\,(\lambda_m/k)^{1/2}$.