potential of the more efficient fluted designs can be established.

## X. Conclusions

A contractual study was initiated to exploit the efficiency of curved elements in the design of panels to cope with the thermal environment. The goal to develop design theory and modify this theory with test data on optimized configurations has been essentially attained for a circular tube configuration. Test data obtained under combined loading on local buckling specimens and on large panels show excellent agreement with theory. The beaded and tubular panel concepts, selected primarily because of their tendency to minimize thermal stresses, are far more efficient than conventional stiffened sheet construction and appear to be competitive when compared with fabrication costs of other panel concepts. The fluted designs, which are still under investigation have the potential to be even more efficient than the circular designs. Successes with the specific concepts developed during this program suggest that much broader applications of the beaded and tubular panel concepts warrant additional considerations.

Boeing has initiated an IR&D program to develop welded circular tube end closures from thin 6A1-4V titanium sheet and to investigate the use of panels with one beaded sheet and one flat sheet in order to further extend the areas which utilize the technologies obtained from this program. Although the work that is being accomplished was initiated to support advanced space vehicle concepts, it is apparent that these developments in beaded and tubular panels should result in increased efficiency in many types of future vehicles. Once again research initially directed toward the space effort will undoubtedly result in even greater non-space benefits.

## References

[1]Shideler, J. L. and Jackson, L. R., "Fuselage and Tank Structures for Hypersonic Aircraft," *Conference on Hypersonic Aircraft Technology*, Ames Research Center, Moffett Field, Calif. SP-148, May 1967, NASA.

[2]Anderson, M. S., Robinson, J. C., and Klich, G. F., "Analysis of Wing Structures for Hypersonic Aircraft," *Conference on Hypersonic Aircraft Technology*, Ames Research Center, Moffett Field, Calif. SP-148, May 1967, NASA.

[3]Plank, P. P., Sakata, I. F., Davis, G. W., and Richie, C. C., "Hypersonic Cruise Vehicle Wing Structure Evaluation," CR-1568, May 1970, NASA.

[4]Card, M. F., Davis, J. G., and Shideler, J. L., "Advanced Design Concepts for Shuttle Airframe Structures," NASA Space Shuttle Technology Conference, San Antonio, Texas, April 12-14, 1972, NASA TM X-2570, July 1972.

[5]Shideler, J. L., Anderson, M. S., and Jackson, L. R., "Optimum Mass-Strength Analysis for Orthotropic Ring-Stiffened Cylinders Under Axial Compression," TN D-6772, July 1972, NASA.

[6]Laakso, J. R., "Design Synthesis of a Boron/Epoxy Reinforced Metal Shear Web," AIAA/ASME/SAE 13th SDM Conference, San Antonio, Texas, April 10-12, 1972.

[7]Jones, R. E., Greene, B. E., "The Force/Stiffness Technique for Nondestructive Buckling Testing," AIAA/ASME/SAE 15th SDM Conference, Las Vegas, Nev., 1974.

[8]Musgrove, M. D., Mortensen, R. E., "Development of Titanium Structural Element Allowables For The Boeing SST," AIAA/ASME 11th SDM Conference, April 1970, Denver, Colo.

[9]Timoshenko S. P., Gere, J. M., *Theory of Elastic Stability*, 2nd ed., McGraw-Hill Book Co., New York, 1961.

[10]"Buckling of Thin-Walled Circular Cylinders," SP-8007, 1965, revised 1968, NASA.

[11]Gerard, G. and Becker, H., "Handbook of Structural Stability, Part III-Buckling of Curved Plates and Shells," TN-3783, 1957, NASA.

[12]Nadai, A., "Theory of Flow and Fracture of Solids," Vol. I, 2nd ed., McGraw-Hill Book Co., New York, 1950.

[13]Mello, R. M., Sherrer, R. E., Musgrove, M. D., "Intermediate Diagonal Tension Field Shear Beam Development For The Boeing SST," AIAA/ASME 12th SDM Conference, April 1971, Anaheim, Calif.

[14]Dykes, B. C., "Analysis of Displacements In Large Plates By The Grid-Shadow Moiré Technique," *Fourth International Conference on Stress Analysis*, April 1970, Cambridge, England.

[15]Plank, P. P., Sakata, I. F., Davis, G. W., and Richie, C. C., "Substantiation Data For Hypersonic Cruise Vehicle Wing Structure Evaluation," CR-66897-2, -3, Feb. 1970, NASA.

FEBRUARY 1974 J. AIRCRAFT VOL. 11, NO. 2

# An Automated Procedure for Computing Flutter Eigenvalues

Robert N. Desmarais* and Robert M. Bennett*
*NASA Langley Research Center, Hampton, Va.*

**A new, fast, and economical automated procedure for implementing the traditional *V-g* method of flutter solution is described. The procedure requires as input the generalized aerodynamic forces for a range of reduced frequencies obtained from an aerodynamic program. These aerodynamic forces are interpolated with respect to reduced frequency using a newly developed, partially tabulated cubic spline that is both fast in execution and economical in storage. The flutter solution is then obtained using an eigenvalue routine that has been developed to take advantage of the parametric nature of the V-g type of solution. Furthermore, the routine takes care of the fundamental and troublesome problem of properly sorting the output eigenvalues. By solving the root-sorting problem, the interpolation for flutter crossings and automatic plotting are accomplished efficiently. The computational techniques used in this new program are described and some sample results are given.**

Presented as Paper 73-393 at the AIAA/ASME/SAE 14th Structures, Structural Dynamics, and Materials Conference, Williamsburg, Va., March 20-22, 1973; submitted July 2, 1973; revision received December 12, 1973.

Index categories: Aircraft Vibration; Nonsteady Aerodynamics; Aeroelasticity and Hydroelasticity.

* Aerospace Engineer, Structures and Dynamics Division. Member AIAA.

## Nomenclature

$A_i$ = tabulated spline coefficients

$A_{ij}$ = generalized aerodynamic force matrix element resulting from the pressure induced by the $j$th mode acting through the displacements of the $i$th mode

$B_i$ = $Y_i - Y_i''/6$ where $Y(X)$ is the function being interpolated

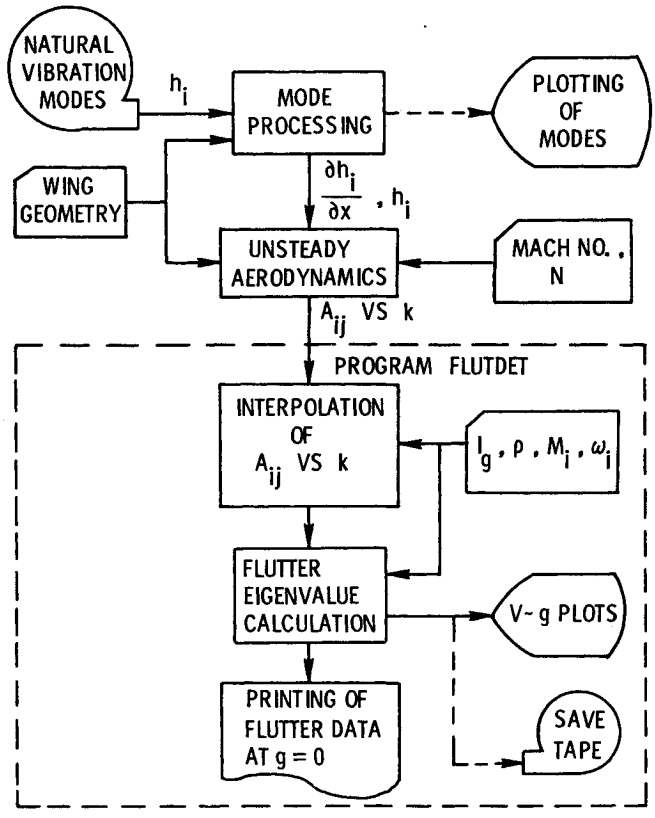



**Fig. 1 Block diagram of typical flutter analysis illustrating portion treated by program FLUTDET.**

$b_0$ = reference length
$C_{ij}$ = components of the flutter matrix as defined by Eq. (12)
$f(\Omega)$ = the characteristic polynomial of the flutter matrix $H$
$g$ = incremental structural damping coefficient
$g_i$ = structural damping coefficient of the $i$th vibration mode
$H$ = upper Hessenberg form of the flutter matrix $C$
$h_i$ = natural vibration mode
$I_g$ = gain of interpolation formula
$i$ = $(-1)^{1/2}$
$k$ = reduced frequency, $\omega b_0/V$
$M_i$ = generalized mass of the $i$th natural vibration mode
$N$ = number of input reduced frequencies
$n$ = number of natural vibration modes
$q_i$ = component of the flutter eigenvector
$S(X)$ = spline interpolation function
$S_1, S_2$ = first and negative second logarithmic derivatives of $f(\bar{\Omega})/\Pi(\bar{\Omega} - \Omega_i)$
$V$ = freestream velocity
$\delta_{ij}$ = kronecker delta (= 1 if $i = j$, and = 0 otherwise)
$\rho$ = freestream density
$\Omega$ = flutter eigenvalue as defined by Eq. (11)
$\bar{\Omega}$ = trial eigenvalue for Laguerre iteration
$\omega$ = vibration frequency
$\omega_0$ = reference frequency
$\omega_i$ = frequency of the $i$th natural vibration mode
$\partial h_i/\partial x$ = streamwise slope of $i$th natural vibration mode

## I. Introduction

FLUTTER prevention can be a significant factor in aircraft design with implications on structural design and thus on weight and performance. With the recent advances in computer technology, computations for a relatively detailed flutter analysis are practical. However, the expenditure of resources or cost can be large for exploring a flight envelope, for multiple configurations, for stores, or for preliminary design. Thus a judicious trade off must be made involving such factors as elapsed time, computer time, man-hours, and confidence in the results. Costs can also be appreciably affected by the overall organization of the flutter analysis software or programs, as well as by the algorithms or computational methods used by the programs. The purpose of this paper is to present some techniques and algorithms that have been developed or adapted specifically for the part of the flutter analysis procedure related to the automated calculation of the flutter eigenvalues. These procedures are designed to reduce costs and thus facilitate flutter analysis.

## II. Statement of the Problem

An overview of the steps in a typical flutter analysis and the portion treated by the program of this paper (FLUTDET) are illustrated by the block diagram of Fig. 1. The portion of the flutter analysis treated by FLUTDET is included in a single computer program with multiple overlays. The overall organization is presented in Fig. 2. The procedure considered here is the conventional $V$-$g$ type of solution using natural vibration modes. The basic steps are closely parallel to those of other methods of flutter solution, however, and the techniques discussed herein may also be applicable to them.

The natural vibration modes, determined either from analysis or from experiment, are required as input data. Generally, the modal data must be processed through a mode processing program to prepare the data in the form required by the unsteady aerodynamics program. The generalized aerodynamic forces for a vibration mode conventionally are functions only of Mach number, planform, and reduced frequency and are expensive to calculate. The aerodynamic and mode processing programs are not treated by this paper, but furnish portions of the input data for the flutter program.

Since the generalized aerodynamic forces are expensive to compute and are smooth functions of reduced frequency $k$, a frequently used cost saving procedure is to calculate the aerodynamic forces for a moderate number of

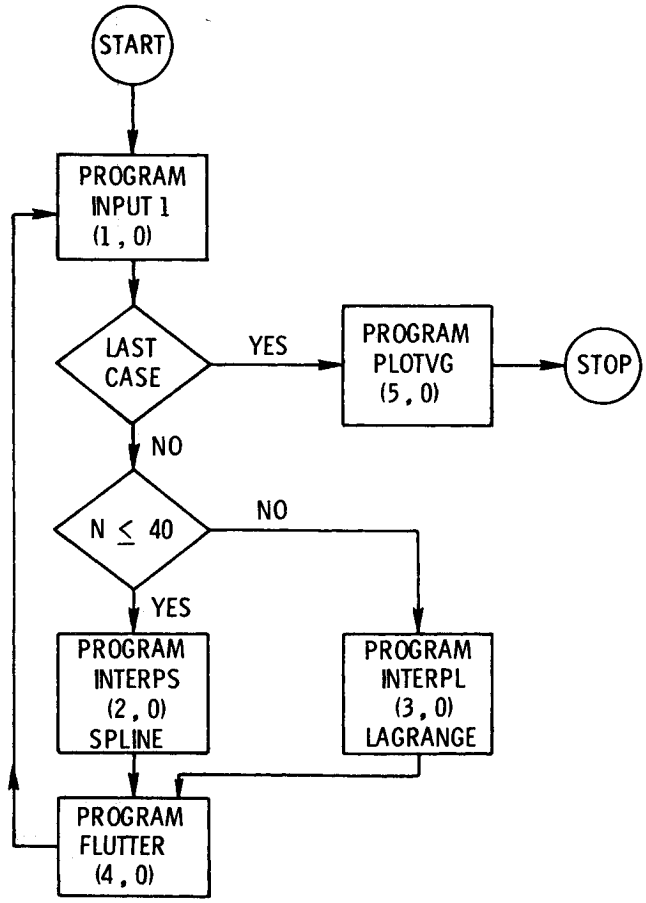



**Fig. 2 Flow diagram of program FLUTDET showing primary overlay levels (in parentheses).**

values of $k$ and interpolate them with respect to $k$ for use in the flutter program. Here a newly developed, partially tabulated cubic spline that is both fast in execution and economical in storage is used and is subsequently described. If more than 40 values of $k$ are input, a simple Lagrange interpolation formula is used.

The $V$-$g$ procedure involves solving a complex eigenvalue problem a large number of times with $k$ as a parameter. Many of the better current eigenvalue programs suffer from two shortcomings for this purpose. First, they start anew for each entry (value of $k$) and do not take advantage of the parametric nature of the problem. This results in a modest increase in time. Second and more important, the output eigenvalues are not sorted such that continuity is maintained as the parameter is varied. Many routines sort the eigenvalues by magnitude or frequency, neither of which will maintain continuity with $k$ for a large flutter problem. Loss of continuity of the roots with $k$ prevents automatic interpolation for flutter crossings, requires large printed output files, and necessitates plotting discrete points which is very slow on mechanical plotters. In order to alleviate these problems, an eigenvalue routine was developed specifically for the flutter problem. Since the aerodynamic forces are interpolated to a small interval in $k$, an iterative technique is used which takes advantage of the parametric nature of the problem to maintain continuity of the eigenvalues with $k$. The description of this eigenvalue routine forms the principal portion of this paper. By maintaining continuity of the eigenvalues with $k$, only the flutter crossings are printed and the $V$-$g$ plots are generated as continuous lines which can be rapidly prepared.

## III. Interpolation of the Generalized Aerodynamic Forces

Figure 3 shows a plot of typical generalized force elements vs reduced frequency. The forces are well-behaved functions of reduced frequency; thus the irregular behavior sometimes exhibited by $g$ vs $V$ plots is due to modal coupling. The generalized forces are relatively expensive to compute so almost all current flutter programs contain a procedure for interpolating these forces. If a large number of forces have been computed, any interpolation procedure is satisfactory. However, if a high-quality interpolation procedure is used, it is possible to use very few generalized forces input for only a few values of $k$. In practice, we have found the most satisfactory procedure to be a natural cubic spline. This spline is the small deflection equation of a uniform beam pinned at the data points. It is the smoothest possible interpolation function in the sense that, of all possible functions passing through the data points, it is the one for which the square of the second derivative, integrated over the interpolation interval, is a minimum.

There are certain problems associated with implementing a spline when the flutter problem contains a large number of degrees of freedom. In order to construct a natural cubic spline, it is necessary to process each force for all values of reduced frequency at the same time. An aerodynamic program will generally compute all of the forces at once for a single reduced frequency. Thus, the aerodynamic program's output file can be interpreted as a large matrix whose column size is twice the square of the number of degrees of freedom and whose row size is the number of reduced frequencies. To construct a natural spline in the conventional manner efficiently, it would be necessary to transpose the matrix described above, spline the transpose, and then transpose the output. This is impractical because, for many problems, the output is too big to fit in memory. The procedure described herein requires much less memory.

A natural cubic spline is a set of piecewise cubic polynomials between data points. Over each interval it is completely defined if the ordinate and second derivative are known at each end. For the first interval the left and right ordinates are known and the left second derivative is zero, so if the right second derivative were known, the first interval would be completely defined. Also, if the right second derivative is known, the right first derivative can be computed, and this equals the left first derivative which, in turn, can be used to compute the right second derivative for the second interval, and so on. This suggests a compact two-pass procedure for generating the interpolation functions. The first pass computes only the second derivative of each generalized aerodynamic force at the right end of the first interval. The output file of the aerodynamic program is then rewound. On an interval-by-interval basis, it is read again and the interpolated forces are computed.

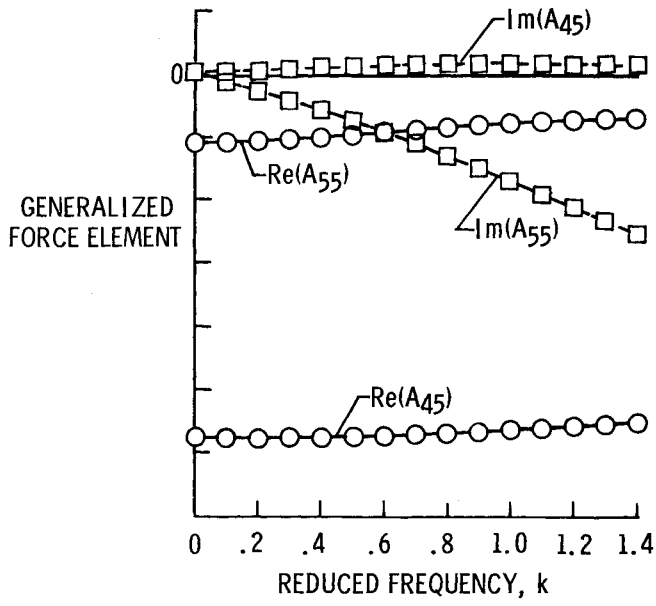



**Fig. 3 Variation of typical generalized aerodynamic force elements with $k$.**

This two-pass procedure is much more efficient if the interpolation abscissas are equispaced. Since it is not always desirable to use equispaced reduced frequencies, the set of reduced frequencies is considered to be a subscripted array and the subscript is used as the interpolation abscissa. This requires only that the reduced frequency be a continuous function of the subscript.

The procedure for computing $Y_i''$ from $Y_i$ when the $X$'s are equispaced is an adaptation of a method developed by Hoskins,[1] from earlier work of Greville,[2] and is implemented as follows:

1. The coefficients $A_i$ are tabulated and stored in memory. They are computed from

$$A_1 = 1$$

$$A_2 = 4$$

$$A_i = 4A_{i-1} - A_{i-2} \qquad i = 3, \ldots, N\text{-}1 \qquad (1)$$

where $N$ is the number of abscissas (i.e., the number of input reduced frequencies).

2. $B_i$ is defined to be $Y_i - Y_i''/6$
$B_1 = Y_1$, and $B_2$ is computed from

$$A_{N-1}B_2 = -A_{N-2} + 6\sum_{I=2}^{N-1}(-1)^I A_{N-I}\,Y_I + (-1)^N A_1 Y_N \qquad (2)$$

3. Over any interval $(X_i, X_{i+1})$ the interpolation function S(X) is the cubic

$$S(X) = (X_{i+1} - X)B_i + (X - X_i)B_{i+1} + (X_{i+1} - X)^3(Y_i - B_i) + (X - X_i)^3(Y_{i+1} - B_{i+1}) \qquad (3)$$

4. Subsequent values of $B_{i+1}$ are computed from

$$B_{i+1} = 6Y_i - 4B_i - B_{i-1} \tag{4}$$

as the $Y_i$ are read.

This method requires very little computer memory and executes very rapidly. It is arithmetically unstable, however, and the instability must be controlled for large $N$. Inspection of Eq. (2) shows that the contribution of $Y_N$ to $B_2$ is multiplied by $1/A_{N-1}$. Any roundoff error contained in this contribution, when carried back to the right end of the interpolation interval by using Eq. (4), is multiplied by $A_{N-1}$, which is $5.8 \times 10^{20}$ for $N = 40$, for example. This implies that Eq. (4) generates an accuracy loss of almost two binary digits each time it is applied (because $A_i \approx 4A_{i-1}$). This round-off error growth is controlled in the program by performing some of the calculations in Eqs. (2) and (4) in double precision. Also, if $N$ is greater than 40, the reduced frequency interval is very small so a simpler four-point Lagrange interpolation is used.

The user selects the gain of the interpolation formula described above. The gain of the interpolation $I_g$, is the ratio of the number of intervals between adjacent reduced frequencies after interpolation to that before interpolation. That is, $\bar{N} = 1 + (N - 1) I_g$ where $\bar{N}$ is the number of reduced frequencies and associated generalized aerodynamic force matrices after interpolation and $N$ is the number before.

## IV. Computation of the Flutter Eigenvalues

For $n$ natural vibration modes $h_i$ the flutter equations are

$$[\ \omega^2 - \omega_i^2 (1 + ig_i + ig)\ ] M_i q_i + \sum_{j=1}^{n} A_{ij} q_j = 0$$
$$i = 1, \ldots, n \tag{5}$$

where

$$M_i = \iint_W m(x,y) h_i^2(x,y) dx\, dy \tag{6}$$

$$A_{ij} = \iint_W h_i(x,y)\, \Delta p_j\,(x,y) dx\, dy \tag{7}$$

and where $m(x,y)$ is the mass per unit area on the lifting surface $W$, $h_i(x,y)$ are the displacements in the $i$th natural vibration mode, and $\Delta p_j(x,y)$ is the lifting pressure over $W$ induced by the downwash associated with the $j$th natural vibration mode. The quantity $g_i$ is the structural damping coefficient for mode $i$ and can vary with $i$. The modal-independent incremental damping coefficient $g$ is used in the $V$-$g$ solution. Note that

$$1 + ig_i + ig \sim (1 + ig_i)(1 + ig) \tag{8}$$

as $g \to 0$ so the error introduced by substituting the product vanishes at $g = 0$. Thus the dimensionless form of Eq. (5) can be written

$$\sum_{j=1}^{n} (C_{ij} - \Omega \delta_{ij})\, q_j = 0 \tag{9}$$

where

$$k = \omega b_0 / V \tag{10}$$

$$\Omega = (\omega_0 b_0 / V)^2 (1 + ig) \tag{11}$$

$$C_{ij} = \left(\frac{\omega_0}{\omega_i}\right)^2 (1 + ig_i)^{-1} \left(\frac{b_0^2 \rho}{2M_i} \cdot \frac{A_{ij}}{\frac{1}{2}\rho V^2} + \delta_{ij} k^2\right) \tag{12}$$

where the quantity $i = (-1)^{1/2}$ is not to be confused with the modal index $i$. This is a conventional linear complex eigenvalue problem that can be solved for any combination of Mach number, reduced frequency, and density.†

† Equation (9) assumes that all modes are elastic. However the program also permits use of rigid body modes.

The incremental damping, $g$ was introduced to make the eigenvalue problem linear and the only solutions desired are those for which $g = 0$.

In Eq. (9) $C_{ij}$ is a function of $k$ and the $A_{ij}$'s have been computed for a large number of closely spaced reduced frequencies. This means that when $\Omega$ or col$(q_i)$ is to be computed, a very close approximation is available; namely, the value at the previous reduced frequency. Because of this an iterative polynomial root finder is more efficient than a global technique such as the $QR$ algorithm.[3]

Several iterative polynomial root finders were considered. We found that of the methods investigated, Laguerre iteration worked best because it permitted the largest interval between successive values of reduced frequency without losing the continuity of the eigenvalues.

The Laguerre iteration procedure is applied as follows:

1) A set of trial eigenvalues $\bar{\Omega}$ are computed by linearly extrapolating the solutions obtained for the two previous values of the reduced frequency.

2) The matrix $C$ is transformed to upper Hessenberg form $H$ by gaussian elimination.

3) The trial eigenvalues are improved by Laguerre iteration as follows.

The quantity $f(\bar{\Omega}) = |H - \bar{\Omega} I|$ and its first two derivatives $f'$ and $f''$ are computed from Hyman's recurrence relations.[4]

Each eigenvalue is computed by iterating the trial eigenvalue using the Laguerre iteration formula

$$\Omega = \bar{\Omega} - n_l \left\{ S_1 \left[ 1 + \sqrt{(n_l - 1)(-1 + n_l S_2 / S_1^2)} \right] \right\}^{-1} \tag{13}$$

where

$$n_l = n - l$$

$$S_1 = \frac{f'(\bar{\Omega})}{f(\bar{\Omega})} - \sum_{i=1}^{l} \frac{1}{(\bar{\Omega} - \Omega_i)} \tag{14}$$

$$S_2 = \left[\frac{f'(\bar{\Omega})}{f(\bar{\Omega})}\right]^2 - \frac{f''(\bar{\Omega})}{f(\bar{\Omega})} - \sum_{i=1}^{l} \frac{1}{(\bar{\Omega} - \Omega_i)^2} \tag{15}$$

and where $l$ is the number of eigenvalues that have already been computed.

The iteration is terminated when a user-selected number of iterations have been performed.

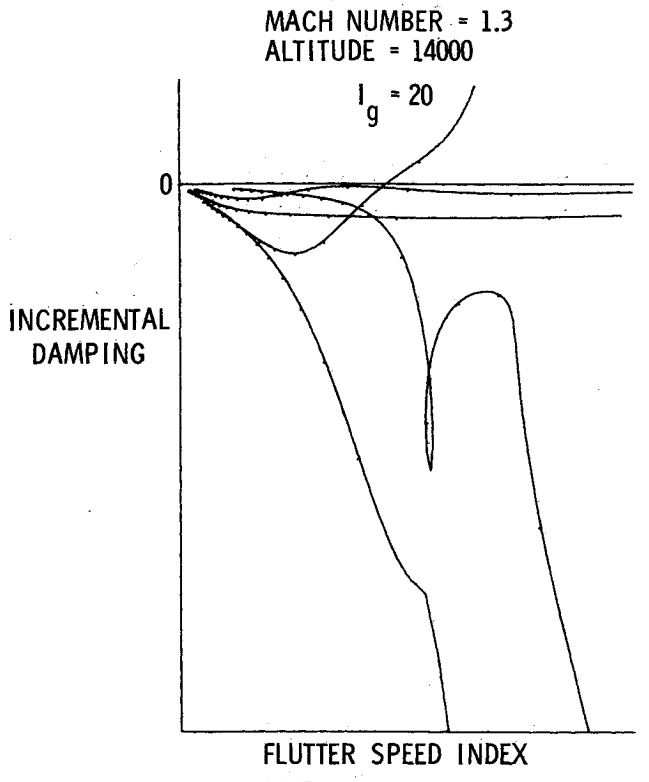
4) If the imaginary part of an eigenvalue changes sign when stepping from one reduced frequency to another, then the values of $k$ and $Re(\Omega)$ for which $Im(\Omega) = 0$ are computed by linear interpolation.

Inspection of Eqs. (14) and (15) shows that partial deflation is used to prevent a trial eigenvalue from converging to a previously computed eigenvalue. For partial deflation, the function whose zeros are being computed is divided by the product of the linear factors associated with all computed zeros and the remainder is not discarded. Partial deflation (unlike total deflation, in which the remainder is discarded) introduces no round-off error.

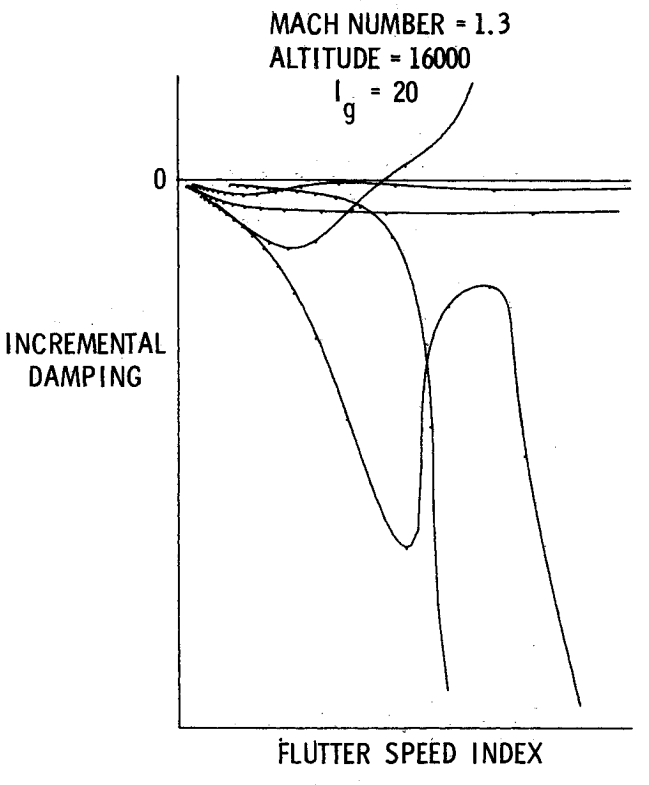
The decision to use a preassigned number of iterations rather than to have the program iterate to convergence was made because there is no single convergence criterion that is reliable, and because a convergence test itself always requires an additional iteration.

It should be noted that, since the generalized forces are interpolated against frequency, it is always possible to trade off the gain of the interpolation procedure against the number of eigenvalue iterations. If the gain is set high enough to generate good quality plots of $g$ vs $V$ and to make the probability of the eigenvalue getting out of order almost zero, then only one iteration is needed. This is the manner in which the program usually is used.

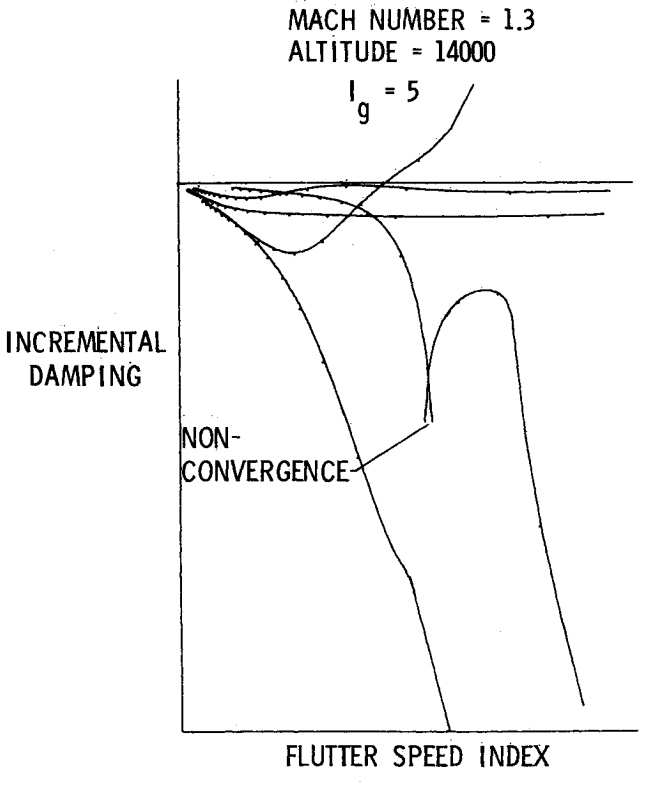
This method of computing eigenvalues is faster then the



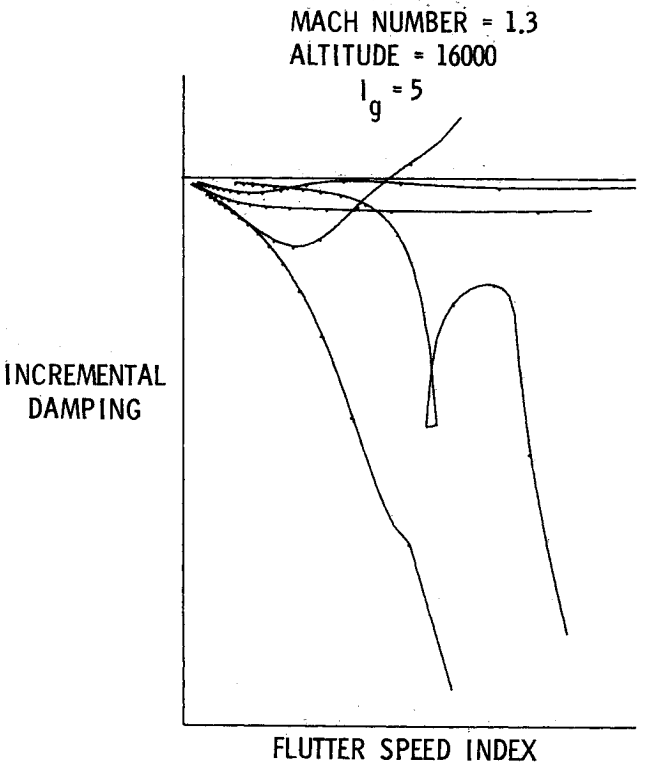

(a) $I_g = 20$.




(b) $I_g = 20$.




(c) $I_g = 5$.




(d) $I_g = 5$.

**Fig. 4 Sample *V-g* plots.**

*QR* algorithm previously used. Both the *QR* algorithm and the method described here start with a preliminary reduction to upper Hessenberg form, requiring $0.5n^3$ multiplications. The rest of the procedure, for both the *QR* algorithm and this method, requires a number of multiplications that is proportional to $n^2$. This means that as $n$ increases the percentage time difference will approach zero. The reason for using the procedure described here is not primarily to save time, but rather to maintain the continuity of the eigenvalues as functions of reduced frequency. It is this fact that makes this an automatic flutter eigenvalue program. If the imaginary part of an eigenvalue changes sign when the reduced frequency is changed, there is always a crossing between the reduced frequencies and not merely some accidental reordering of the eigenvalues.

## V. Preparation of $g$ vs $V$ Plots

The output from the flutter eigenvalue program is in two parts. The only printed output is the information at each crossing; that is, the computed values of speed and frequency for which the incremental damping $g$ is zero. When the program is used in the automatic mode, say as part of a flutter optimization package, the crossing information is the only output that will usually be used. The other part of the output is a computer generated plot of incremental damping versus flutter speed index with reduced frequency as a parameter for each value of density and Mach number. Figure 4 shows several of these plots. On the plots, the points corresponding to consecutive values of reduced frequency are connected by straight lines. In addition, if the reduced frequency is an input value (that is, not interpolated), a tick mark is drawn on the curve at the point. This means that although the shape of the curve between tick marks may depend upon the gain of the interpolation formula or how the interpolation is performed, the points with tick marks do not depend upon the interpolation in any way.

It was mentioned previously that usually only a single Laguerre iteration was used to compute the eigenvalues. Occasionally this is not enough for convergence. Lack of convergence is indicated on the plot by a gap in the plot at an uninterpolated value of reduced frequency. This occurs because the uninterpolated points are plotted twice; first, using the user-selected number of iterations and again with twice that number. Figure 4c shows an example of this (compare with Fig. 4a). In this example, since the nonconvergence did not occur in any of the modes of interest or in the low damping region, it could safely be ignored. Had it occurred near $g = 0$ the case would have been rerun using either more iterations or with a higher interpolation gain. Figure 4d (compare with Fig. 4b) shows how continuity of certain modes can be affected if the interpolation gain is too small.

## VI. Conclusions

The conventional $V$-$g$ method of computing flutter speeds and frequencies has been improved by using state-of-the-art mathematical techniques to interpolate the generalized aerodynamic forces and to compute the eigenvalues of the flutter determinant. Table 1 shows the resulting savings in computing time and central memory requirement for a representative flutter calculation. The timing information in the table is for five degrees of freedom (program dimensioned for 16), one Mach number, eight altitudes, and 401 reduced frequencies (21 interpolated to 401).

The old method used a nonequispaced natural cubic spline to interpolate the generalized forces, Francis' QR algorithm to compute the eigenvalues, and plotted $g$ vs $V$ as discrete points using computer generated symbols.

**Table 1 Computer utilization**

| Task | New method | Old method |
|---|---|---|
| Interpolation | 4.4 sec (10,000 words) | 7.5 sec (28,000 words) |
| Eigenvalue calculation | 95.9 sec (12,000 words) | 115.3 sec (12,000 words) |
| $V$-$g$ plots | 26.1 sec (11,000 words) | 105.6 sec (8,000 words) |
| Total effort (kiloword hours) | 0.41 | 0.68 |

Although the table shows that the cost savings from the use of this method are significant, the greatest benefit derived from the method is not the cost savings but rather the fact that the continuity of the solution from one reduced frequency to the other is maintained. This permits the program to:

1) Automatically interpolate to compute each reduced frequency for which the incremental damping vanishes.

2) Automatically interpolate on altitude to compute the matched-point altitude; that is, the altitude for which the flutter speed computed by the program equals the airspeed obtained by multiplying the flutter Mach number by the speed of sound at that altitude (as determined from the U.S. Standard Atmosphere, 1962).

3) Draw simple and economical plots of incremental damping *vs* velocity by connecting points with straight lines rather than drawing an individual symbol for each point.

To sum up, a method of obtaining flutter eigenvalues has been developed that maintains the continuity of each eigenvalue as a function of reduced frequency. The continuity is maintained by:

1) Iteratively solving for the eigenvalues using a starting value obtained from previous reduced frequencies.

2) Keeping the starting value very close to the converged value by using a very small interval on reduced frequency made possible by interpolating the generalized aerodynamic forces.

Much of the cost of using a small interval on reduced frequency is made up by the fast convergence of the iterative process that results therefrom.

## References

[1] Hoskins, W. D., "Table for Third Degree Spline Interpolation Using Equispaced Knots," *Mathematics of Computation,* Vol. 25, Oct. 1971, pp. 797–801.

[2] Greville, T. N. E., "Table for Third Degree Spline Interpolation with Equally Spaced Arguments," *Mathematics of Computation,* Vol. 24, Jan. 1970, pp. 179–183.

[3] Francis, J. G. F., "The QR Transformation, Parts I and II," *Computer Journal,* Vol. 4, Oct. 1961, Jan. 1962, pp. 265–271 and 332–245.

[4] Wilkinson, J. H., *The Algebraic Eigenvalue Problem,* Clarendon Press, Oxford, 1967, pp. 426–427.