LARGE AMPLITUDE FLUTTER OF A LOW ASPECT RATIO PANEL AT LOW SUPERSONIC SPEEDS COMPARISON OF THEORY AND EXPERIMENT

by

C. S. Ventres

C. K. Kang

prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Marshall Space Flight Center

Contract NAS 8-28577

August 1973

PRINCETON UNIVERSITY

Department of Aerospace and Mechanical Sciences

The Aeroelastic and Magnetoelastic Laboratory

AMS Report No. 1116

ABSTRACT

Flutter boundaries, as well as flutter limit cycle amplitudes, frequencies and stresses were computed for a panel of length-width ratio 4.48 exposed to applied in-plane and transverse loads. The Mach number range was 1.1 to 1.4 . The method used involved direct numerical integration of modal equations of motion derived from the nonlinear plate equations of von Karman, coupled with linearized potential flow aerodynamic theory. The results obtained were compared to experimental data reported in Ref. 5.

The flutter boundaries agreed reasonably well with experiment, except when the in-plane loading approached the buckling load. Structural damping had to be introduced, to produce frequencies comparable to the experimental values. Attempts to compute panel deflections or stress at a given point met with limited success. There is some evidence, however, that deflection and stress maxima can be estimated with somewhat greater accuracy.

TABLE OF CONTENTS

NOMENCLATURE

a	= panel length
a_n	= modal amplitude
b	= panel width
$B_{ijk\ell}$	= nonlinear elastic terms
c	= speed of sound
D	= panel bending stiffness
G_s	= structural damping factor
$g_s \equiv \frac{G_s}{D}\left(\frac{D}{\rho_m h a^4}\right)^{1/2}$	= dimensionless structural damping factor
h	= panel thickness
H_{ji}, I_{ji}	= aerodynamic admittance functions (Eq. 10)
$K = \omega\left[\frac{\rho_m h a^4}{D}\right]^{1/2}$	= dimensionless flutter frequency
M	= Mach Number
N	= number of modes used (Eq. 3)
N_x	= $R_x/R_x)_{buckle}$
p	= pressure
$q = \frac{1}{2}\rho U^2$	= dynamic pressure
Q_{ji}	= generalized aerodynamic force (eq. 10)
R_x	= streamwise applied in-plane load
$s \equiv (\lambda^*/\mu)^{1/2}\,\tau$	= dimensionless aerodynamic time
t	= time

U	= flow velocity
w	= panel deflection
x,y	= coordinates in plane of plate
z	= coordinate normal to plate

<u>Greek</u>

δ_{ij}	= Kronecker delta
Δp	= static pressure differential
$\Delta P \equiv \frac{\Delta p a^4}{Dh}$	= dimensionless static pressure differential
$\lambda^* \equiv \frac{\rho U^2 a^3}{D}$	= dimensionless flow dynamic pressure
$\mu \equiv \rho a/\rho_m h$	= dimensionless flow density
ν	= Poisson's ratio
ρ	= flow density
ρ_m	= panel density
$\sigma_x,\ \sigma_y$	= panel stresses
$\tau \equiv t\left(\frac{D}{\rho_m h a^4}\right)^{1/2}$	= dimensionless time
ϕ	= velocity potential
Φ	= Airy stress function
ψ_m	= modal function (Eq. 3)
ω	= flutter frequency

I. INTRODUCTION

It is now well established that panel flutter is not, in many cases, an immediately destructive vibration. Hence flutter may be tolerated if it can be established that the flutter amplitude is sufficiently small and the duration of flutter sufficiently short. Unfortunately linear structural and aerodynamic theory is incapable of determining flutter amplitudes. Only by including the important panel nonlinearities can the flutter amplitude be established. Recently, methods have been developed at Princeton for analyzing the large amplitude oscillations of a fluttering plate.[1-4] In the investigation reported here, these methods were used to calculate the flutter behavior of a panel exposed to a static pressure differential (that is, an applied transverse pressure load), and to applied in-plane compressive loads comparable to the buckling load of the panel.

The panel length-width ratio (4.48), and the range of flow Mach number (1.1 to 1.4) were selected to allow comparison with the results of wind tunnel tests reported in Reference 5. These tests were in turn motivated by a desire to investigate the flutter behavior of certain panels mounted on the forward shirt of the S IV-B stage of the Saturn V launch vehicle.[5] During these tests, the frequency and amplitude of the panel motion (if any) were measured as the tunnel dynamic pressure was increased. The tests were carried out at various values of test section Mach number, panel static pressure differential, and applied in-plane load. By this method both the panel flutter boundaries (lowest dynamic pressure at which flutter occurrred) and the severity of the post flutter motion were determined.

The calculations described herein were carried out for the same range of parameters as used in Ref. 5. The method used involves the direct numerical integration of a set of nonlinear differential equations for the panel motion, derived from an approximate modal solution of the von Karman nonlinear plate equations. Because of the range of Mach numbers involved, the popular quasi-steady or piston theory expressions for the aerodynamic pressure on the panel were not applicable. Instead the full linearized inviscid, potential flow theory was employed.

So far as is known, the work reported herein constitutes the first attempt at predicting theoretically the severity of flutter of a low aspect ratio stressed panel in the critical low supersonic Mach number range.

II. THEORETICAL DEVELOPMENT

The equations of motion for a three dimensional plate, von Karman's large deflection equations,[6] are

$$D\nabla^4 w = \frac{\partial^2 \Phi}{\partial y^2}\frac{\partial^2 w}{\partial x^2} - 2\frac{\partial^2 \Phi}{\partial x \partial y}\frac{\partial^2 w}{\partial x \partial y} + \frac{\partial^2 \Phi}{\partial x^2}\frac{\partial^2 w}{\partial y^2} \tag{1}$$

$$- \rho_m h \frac{\partial^2 w}{\partial t^2} - G_s \nabla^4 \frac{\partial w}{\partial t} - (p - p_\infty) + \Delta p = 0$$

$$\frac{\nabla^4 \Phi}{Eh} = \left(\frac{\partial^2 w}{\partial x \partial y}\right)^2 - \frac{\partial^2 w}{\partial x^2}\frac{\partial^2 w}{\partial y^2} \tag{2}$$

where w is the plate deflection and Φ is the Airy stress function. G_s is a structural damping parameter. The reason for including structural damping will be discussed later. Equation (2) and the first three terms on the right hand side of equation (1) constitute the nonlinear elastic coupling between out-of-plane bending and in-plane stretching that ultimately limits the amplitude of flutter.

Equations (1) and (2) are reduced to a set of simultaneous nonlinear differential equations by Galerkin's method. The transverse displacement w is expressed as a linear combination of modal functions that satisfy the appropriate boundary conditions at the edge of the plate (in this case, those for a clamped plate):

$$w/h = \sum_{m=1}^{N} a_m(t)\, \psi_m(x/a)\, \psi_1(y/b)$$

$$\psi_m(\zeta) \equiv \cos(m-1)\zeta - \cos(m+1)\zeta \tag{3}$$

As is described in greater detail in Refs. 1-4, Φ is determined by solving equation (2) with expression (3) inserted for w. The boundary conditions satisfied by Φ on the plate edges (the so-called in-plane boundary conditions) depend on the design of the panel support structure. In Refs. 3 and 4 methods of handling situations corresponding to either complete restraint (no in-plane motion permitted at the edges) or zero restraint (in-plane stresses zero at the edges) are discussed. It is not generally feasible to distinguish between these two alternate sets of boundary conditions beforehand by analyzing the panel support structure (and in fact most practical structures would create a degree of restraint somewhere between the two extremes), so both sets are retained in the developments that follow. With Φ determined, equation (1) is satisfied in the Galerkin sense by computing the integral average of equation (1) weighted successively by each of the modal functions $\psi_i(x/a)\ \psi_1(y/b)$ in expression (3) and setting the result to zero. The resulting system of equations is (in nondimensional form):

$$\begin{aligned} &\sum_j S_{ij}\,(\ddot{a}_j + g_s \dot{a}_j) + \sum_j C_{ij} a_j \\ &+ \sum_j \sum_k \sum_\ell {}_{ijk\ell}\, a_j a_k a_\ell + \lambda^* \sum_j Q_{ji} - \underline{\Delta P}\ \delta_{1i} = 0 \end{aligned} \tag{4}$$

The matrices S and C are the familiar modal mass and elastic stiffness matrices of linear vibration theory. C contains the applied streamwise in-plane tension R_x as a parameter; when R_x decreases below a critical negative value, the plate buckles. The fourth order array B contains the nonlinear terms corresponding to the coupling between in-plane stretching and

and out-of-plane bending referred to previously. Explicit expressions for all of these terms are contained in Ref. 4.

The generalized aerodynamic forces Q_{ji} are defined as

$$Q_{ji} \equiv \int_0^1 \int_0^1 \left[\frac{p_j - p_\infty}{\rho U^2}\right] \psi_i(x/a)\psi_1(y/b) \frac{dx}{a} \frac{dy}{b} \tag{5}$$

where p_j is the pressure on the plate caused by an arbitrary deflection in the jth mode:

$$w \equiv a_j(\tau) \psi_j(x/a)\psi_1(y/b) \tag{6}$$

p_j is given by

$$p_j = -\rho\left(\frac{\partial\phi}{\partial t} + U \frac{\partial\phi}{\partial x}\right)\Bigg|_{z=0} \tag{7}$$

where the velocity potential ϕ must satisfy

$$\nabla^2\phi - \frac{1}{c^2} \left(\frac{\partial}{\partial t} + U \frac{\partial}{\partial x}\right)^2 \phi = 0 \tag{8}$$

subject to the boundary conditions

$$\left.\frac{\partial\phi}{\partial z}\right|_{z=0} = \frac{\partial w}{\partial t} + U \frac{\partial w}{\partial x} \qquad \text{on plate}$$

$$= 0 \qquad \text{off plate} \tag{9}$$

The boundary value problem defined by Eqs. (6-9) has been solved in Ref. 7, where it is shown that

$$Q_{ji} = \frac{1}{M}\left(a_j D_{ji} + \frac{da_j}{ds} S_{ji}\right) + \int_0^s a_j(\sigma)\, H_{ji}(\sigma)\, d\sigma + \int_0^s \frac{da_j(\sigma)}{d\sigma}\, I_{ji}(s-\sigma)\, d\sigma \qquad (10)$$

with

$$s = (\lambda^*/\mu)^{1/2}\tau \qquad (11)$$

See Ref. 7 or Appendix B of Ref. 4 for evaluations of D_{ji}, S_{ji}, $H_{ji}(s)$, and $I_{ji}(s)$. (Beware of slight notational differences between the two.) These functions depend parametrically on M and a/b, but not explicitly on λ^* and μ. If the integrals in (10) are deleted, the Q_{ji} are those given by "piston theory"; that is by a direct substitution of the well known expression

$$p - p_\infty = \frac{\rho U^2}{M}\left(\frac{\partial w}{\partial x} + \frac{1}{U}\frac{\partial w}{\partial t}\right)$$

into equation (1).

Equations (4) and (10) are combined to form a set of coupled nonlinear ordinary integral-differential equations in time, τ. The solution procedure is to specify λ^*, μ, M, Δp, a/b, R_x, g_s, and to determine the modal amplitudes by numerical integration. Given the $a_n(\tau)$, the deflection w/h or stresses σ_x , σ_y at any selected point on the panel may be calculated in a straightforward manner. The computer programs used to carry out these various procedures are listed in the Appendix. These routines are modified and improved version of the programs listed in Ref. 4.

III. COMPUTATIONAL CONSIDERATIONS

The considerations of this section relate to the manner in which the computations were arranged and carried out, and to the way in which the results obtained have been displayed. They have been dictated both by the nature of the wind tunnel experiments reported in Ref. 5, and by the necessity of using a very large number of modes (12 in most cases) in order to properly represent the behavior of the low aspect ratio panel being studied.

In order to save computer time (and hence expense) it was found useful to divide the computations into four distinct steps. These are:

1) Computation of the nonlinear terms. Only the plate length width ration a/b, the Poisson's ratio ν, and the in-plane boundary conditions need be specified in order to determine B. Since the results were to be compared with the data of Ref. 5, only one value of a/b (= 4.48) and ν (= 0.3) were employed. Hence only two sets of nonlinear terms, corresponding to complete and zero in-plane edge restraint, were required. These were computed at the outset, and stored on magnetic tape.

2) Computation of the aerodynamic admittance functions H_{ij}(a/b, M, s) H_{ij}(a/b, M, s) (see Eq. 10). As indicated, these quantities depend on the panel length-width ratio and the flow Mach number as well as the dimensionless aerodynamic time s. Since only four values of M were studied, it was found worthwhile to compute H_{ij} and I_{ij} beforehand as well (distinct sets of values for each of M = 1.1, 1.2, 1.3, and 1.4). They also were stored on magnetic tape.

3) Numerical integration of the panel equations of motion. This operation uses as inputs the data stored from steps 1) and 2) above,

as well as specification of λ^*, μ, M, Δp, R_x, and g_s. Interest centers on the amplitude and frequency of the flutter limit cycle at a given point on the panel:

$$w/h)_p = f(\lambda^*, \mu, M, \Delta p, R_x, g_s)$$

$$K = g(\quad " \quad)$$

(The cross-stream in-plane load R_y was zero in the experiments of Ref. 5. and so was assigned the same value in the present study.)

The dimensionless flow dynamic pressure λ^* and flow density μ are related through the flow velocity:

$$\frac{\lambda^*}{\mu} = \left(\frac{\rho_m h a^2}{D}\right) U^2$$

The quantity in brackets is uniquely defined by the geometric and material specifications of the panel being studied. Furthermore, in a continuous flow wind tunnel the flow velocity U is determined by the test section Mach number M, and the stagnation temperature T_o in the tunnel settling chamber:

$$U = (R T_o) \left(\frac{M^2 T}{T_o}\right)$$

$$\frac{M^2 T}{T_o} = \frac{M^2}{1+\frac{\gamma-1}{2} M^2}$$

The stagnation temperature is held constant during tunnel operation, so U is determined solely by the test section Mach number.

It is therefore convenient to display the results of the flutter amplitude and frequency as functions of q, the (dimensional) dynamic pressure rather then as functions of both λ^* and μ independently:

$$w/h)_p = F(q, M, \Delta p, N_x, g_s)$$

$$K = G(q, M, \Delta p, N_x, g_s)$$

Since the non-dimensionalization of R_x is arbitrary, it has been replaced here by the ratio of R_x to its buckling value:

$$N_x \equiv R_x/R_x)_{buckle}$$

By extrapolating to $w/h)_p \to 0$, it is possible to determine the critical or flutter dynamic pressure q_f and the flutter frequency K_f:

$$q_f \equiv F_1(M, \Delta p, N_x, g_s)$$

$$K_f = G_1(M, p, N_x, g_s)$$

4) Panel stresses during flutter. In the theory of thin plates, normal stresses vary linearly across the plate thickness. The extreme values of stress occur on the upper and lower surfaces of the panel, e.g.

$$\sigma_x = \sigma_x)_{ms} \pm \sigma_x)_b$$

where the + and - sign apply to the upper and lower surfaces, respectively. A similar equation holds for σ_y . The bending stress

$\sigma_x)_b$ is proportional to the local curvature of the plate, and is obtained from the modal amplitudes a_n by differentiating Eq. (3) for w/h. On the other hand, the middle surface or in-plane stress $\sigma_x)_{ms}$ obtained by differentiating the Airy stress function Φ of Eq. (2). As such the in-plane stresses depend not only on the plate deflection $w(x/a, y/b,\tau)$ but also on the in-plane boundary conditions satisfied at the edges of the plate. The computer program listed in the Appendix uses the modal amplitudes $a_n(\tau)$ from step 3) to calculate the in-plane or middle-surface stress for a panel completely restrained at its edges. Since not many flutter calculations were made for the zero edge restraint case, an equivalent program for computing the middle surface stresses in such panels was not written.

IV. NUMERICAL RESULTS

Free Panel Vibrations

In order to explore the extent to which the theoretical model employed mirrors the elastic behavior of the panel, independently of the flutter results, panel natural frequencies were computed as a function of applied static transverse and in-plane loading. The transverse load was equivalent to a pressure differential between the two faces of the panel.

The computations were carried out by integrating the modal equations (4) (with $\lambda^* = \mu = 0$, $g_s > 0$) to determine the equilibrium panel deflection under the assumed loading, and then linearizing the equations about that deflection. The natural frequencies were determined numerically from these linearized equations by solving a classical eigenvalue problem. Representative results are shown in Figures 1 through 4, along with comparable experimental data from Ref. 5.

Figures 1 through 3 show the behavior of modes 1, 2, and 6 under a transverse pressure loading. In each case calculations were made assuming both zero and complete in-plane edge restraint. (The edges of a plate with zero in-plane restraint are free to move in the plane of the plate in response to transverse plate motions, while the edges of a plate with complete in-plane restraint are prevented from making any such movement.) In all three figures there is a systematic discrepancy at zero pressure load. Part of this difference is attributable to imperfect convergence of the solution, but probably not all. The calculated frequencies included in Ref. 5 (Table II) show a similar deviation from the experimental results. Of greater interest, however, are the amounts by which the various frequencies increase when a pressure load is applied. For the lower modes, the assumption of complete edge restraint provides the best agreement with experiment, while for the higher modes, zero edge restraint works best.

Figure 4 shows calculated and experimental results for the behavior of the ninth mode under a compressive in-plane load. Both calculated frequencies drop off much more near the buckling load than does the experimental curve. This may be due to the presence of imperfections in the plate, such as a slight initial curvature or waviness.

It was not possible, on the basis of these results, to eliminate one in-plane boundary condition from further consideration. Therefore flutter calculations were carried out for both cases, although a shortage of time and money limited the number of zero edge restraint runs that could be carried out.

Flutter Calculations - General Nature of Solutions Obtained

The flutter limit cycle was determined by integrating the modal equations (4) until a periodic motion was found. Experience indicates that the initial conditions used to start the integration do not affect the amplitude or frequency of the limit cycle, at least for $N_x < 1$. Because of the large length-width ratio (a/b = 4.48) employed, at least twelve modes were required to obtain an acceptable degree of convergence. Furthermore, the transient portion of the solution survived for the equivalent of many cycles of the ultimate limit cycle motion. (Neither of these statements apply for smaller values of a/b.) As a result, the numerical integration turned out to be costly in terms of both computer memory storage area and computation time.

Initially all calculations were made without introducing any damping other than the aerodynamic damping implicit in the potential flow expression (10) for the generalized forces Q_{ji}. However, it was found that for larger values of the dynamic pressure q the flutter frequency became very high ($\gtrsim$ 900 Hz), with the panel deflection being such that the 9th or 10th mode had the largest amplitude. Cunningham[8] has shown that the flutter frequency and mode shape are both critically sensitive to the amount of structural damping present. Therefore, structural damping was introduced into the panel equations of motion (4) in order to suppress the high frequency flutter.

The structural damping present in the actual panels used in Ref. 5 has not been measured to date. Moreover, if the lower flutter frequency found experimentally is indeed due to the presence of additional damping, the source of that damping need not be structural. It may well be caused by the boundary layer in the airflow over the panel. The capability for dealing with boundary layer effects does exist,[9] but at least for the present the technique is not practical for flutter calculations involving the use of many structural modes. Hence the introduction of structural damping must be viewed as an essentially ad hoc procedure designed to eliminate a physically unrealistic aspect of the flutter behavior.

From a mathematical standpoint, there are many forms of structural damping that can be introduced into the plate equations to describe non-elastic behavior. Of these the traditional and most popular choice is the $(1 + ig)$ type, which is meaningless for non-sinuousoidal motion and is therefore unsuitable for nonlinear plate equations. The most common of the many types that can be used have the general form

$$G_s \nabla^{2n} \frac{\partial w}{\partial t}$$

with $n = 0, 1, 2, \ldots.$

They differ, in the modal formulation used here, in the relative damping ratios given the various modes. Roughly speaking, the damping ratios increase as the mode number raised to the (n-1)st power. For the present work, $n = 2$ was selected because $G_s \nabla^4 \frac{\partial w}{\partial t}$ fits easily into the modal equations (4), and because it provides greater damping in the higher modes whose motion it is intended to suppress.

Most of the results that follow have been calculated with $g_s = .0001$. This provides a damping ratio for the first mode of .025 (2.5% of critical). Cost limitations have made it impossible to present a systematic study of the influence of structural damping over the complete range of Mach number, static pressure differential, and applied in-plane load considered here.

Flutter Boundaries

Figures 5 and 6 show flutter boundaries as a function of Mach number for an unloaded panel ($\Delta p = N_x = 0$). Curves are shown for $g_s = 0.$ and $g_s = .0001$. On the same figures are shown experimental data from Ref. 5. The data were obtained from several different panels (of indentical specification), and for two different boundary layer thicknesses, the thicker one being induced by inserting spring pins in the tunnel wall ahead of the panel, which was mounted flush to the wall. This caused the boundary layer thickness (as measured near the trailing edge of the panel) to increase roughly 7 to 30%, depending on the tunnel Mach number and dynamic pressure.[5]

Both the theoretical and experimental results show a gradual decrease in flutter dynamic pressure with increasing Mach number, but the theory shows no minimum at $M = 1.3$. The theoretical flutter boundary for $g_s = 0.$ agrees best with the experimental results for the smooth wall boundary layer (Figure 5) while the damped flutter boundary agrees best with the results for the rough wall boundary layer (Figure 6). This is the correct qualitative behavior, since the boundary layer introduces a damping effect that increases with boundary layer thickness. The quantitative agreement in Figure 6 is of course fortuitous, since the amount of structural damping introduced is arbitrary, and in any event the damping is of structural origin in the theory and aerodynamic in the experiment.

Flutter frequencies are shown in Fig. 7. As mentioned previously, the frequencies for zero damping are unrealistically high, whereas those for $g_s = .0001$ are comparable to the experimental results. Neither theory nor experiment shows much variation with Mach number.

Figures 8 and 9 show calculated and experimental flutter boundaries for plates exposed to compressive in-plane loads. In both figures the qualitative behavior with N_x is correct, although the rate of decrease in the flutter dynamic pressure is more rapid according to the theory. Near buckling ($N_x = 1.0$), the theoretical result becomes overly conservative. The calculations of flutter boundaries near the buckling load is a difficult matter, since the plate behavior is then especially sensitive to the presence of small initial structural imperfections, to the damping effect of the boundary layer, and so on. The prediction of panel natural frequencies under in -plane loading suffers the same difficulty, as can be recalled from Fig. 4.

Figure 10 shows a limited set of calculations for a panel exposed to a static pressure differential. Both sets of in-plane boundary conditions (zero and complete edge restraint) are included. The line labeled "Exp." is a derived curve taken from Fig. 43 of Ref. 5. The result for zero in-plane restraint shows the better agreement with experiment, in spite of the fact that the panels referred to in Ref. 5 were carefully mounted in a massive supporting structure. This result is consistent with similar comparisons made previously involving panels of smaller length-width ratio at higher Mach numbers.[3,10]

Figs. 11, 12, and 13 contain flutter boundaries for panels exposed to combined loading (both $\Delta p \neq 0.$ and $N_x \neq 0.$). The theoretical results are all for the case of complete edge restraint, and reflect the same behavior as exhibited in Fig. 10, namely, a lack of sensitivity to static pressure differential. Note, however, that the agreement between the slopes of corresponding pairs of flutter boundaries in Fig. 11 improves as the in-plane loading increases.

It would be desirable to carry out calculations equivalent to those shown in Figs. 11-13 for the zero in-plane restraint case.

Panel Displacement in Flutter

A record of panel centerline deflection during approximately one period of the flutter oscillation is shown in Fig. 14. Structural damping (g_s = .0001) was assumed in making the calculation; with no damping, many more zero crossings appear than are shown. The motion portrayed in Fig. 14 is qualitatively similar to that reported in Ref. 5. (See especially Fig. 57 of that report). In particular, the panel deflection is largest near the trailing edge, but not markedly so, and the streamwise variation of the deflection is elaborate, but with relatively few zero crossings at any given instant. The motion has a quasi wave-like character, since the zero crossings (points of zero deflection) move with time, and even appear and disappear.

Relatively little panel displacement data was published in Ref. 5, and what was presented was limited to a case wherein the panel was buckled by the applied in-plane load. This situation is both the most important physically (since at a given dynamic pressure the panel deflection is maximized), and the most difficult to handle analytically. As mentioned previously, buckled panels are especially sensitive to effects that normally are either ignored entirely (such as initial imperfections), or handled very crudely (structural damping).

Panel displacements at three different streamwise locations (but 2.5 inches off the centerline) are shown in Fig. 15. The streamwise locations of probes A, C, and F are shown in Fig. 14. At all three locations, the calculated displacements are considerably larger than their experimental counterparts.

Stresses

The bending strèsses are generally considerably larger than the in-plane or axial stresses during flutter. Since the bending stresses are proportional to local panel curvature, the bending stress distribution generally resembles the panel deflection (see Fig. 16). Attempts to compute stresses at a given point on the panel are therefore hampered by the same difficulty encountered in calculating deflections: a small change in the flutter mode shape causes large errors in the stresses computed at that point.

Fig. 17 shows a comparison of calculated and experimental stresses as a function of flow dynamic pressure for a buckled panel. The open circles are the peak-to-peak bending stress in panel #6 at the location of gauge B1, just off the center line of the panel near its trailing edge. The small triangles connected by straight lines are theoretical peak-to-peak stresses calculated for the same point, as well as for a point on the panel center-line, three quarters of the way back behind the leading edge. This latter location is the point where the maximum stress occurred, according to the theory. It should be noted that the applied in-plane load assumed for the calculations was only 73.5% of the theoretical buckling load, whereas the experiment was carried out with an in-plane load equal to 96% of the buckling load applied. As can be seen, the bending stress computed at the 3/4 chord point agrees better with the experimental result than does the value computed at the position of gauge B1. If the stress measured at B1 is in fact the maximum stress that occurred, then the maximum stress is computed with greater accuracy than is the stress at B1.

Figs. 18 through 21 (each of which is divided into two parts) show similar comparisons. In each figure part (a) shows theoretical and experimental bending stresses (Figs. 18 and 19) or axial stresses (Figs. 20 and 21) at the point referred to above. In part (b) the calculated data is the theoretical maximum stress on the panel, displayed alongside the same experimental data as in part (a). In general, the maximum stress (which is less sensitive to changes in the flutter mode shape) best reflects the experimental trends, at least for small N_x . Near the buckling load, excessively large maximum stresses are predicted, presumably for the same reasons mentioned earlier.

V. CONCLUSIONS

Flutter boundaries were computed numerically as functions of Mach number, in-plane loading, and static pressure differential. Comparison with experimental data indicate reasonably good correlation for Mach number and in-plane loading, except near the buckling load. The influence of static pressure differential depends on the in-plane boundary conditions assumed. Assuming zero restraint (edges free to move in plane) provided the best correlation with experiment, although not enough calculations were made to firmly establish this point.

The flutter mode shapes calculated were in good qualitative agreement with experiment. The flutter frequency, however, proved to be sensitive to the amount of structural damping assumed. With no damping, the coupled flutter frequency was several times higher than the experimental value. Because flutter frequency is an important factor in determining panel fatigue life, future experimental programs should include a determination of panel damping. In addition, the theoretical model employed should be improved to include the damping effect of the boundary layer.

Attempts to compute panel deflection and stresses during flutter met with limited success, particularly for buckled panels. There is some indication, however, that maximum deflections and stresses can be calculated with greater accuracy than deflections on stresses at a specific point. From a practical standpoint, knowledge of the maximum is sufficient to determine panel fatigue life; the stress distribution and mode shape are of lesser significance.

Most of the difficulties encountered in this investigation stem from the large length-width ratio of the panel and the presence of large in-plane loads. In this regard a wind tunnel test program using a carefully constructed high aspect ratio $(a/b < 1)$ panel would be very helpful. Stream-wise buckling loads might well be included in the test program, but an extensive set of data should also be collected with little or no in-plane loading present. Such data would be of great help in assessing current theoretical methods without the perplexing but not fundamental difficulties associated with low aspect ratio and panel buckling.

REFERENCES

1. Dowell, E. H., "Nonlinear Oscillations of a Fluttering Plate," AIAA Journal, Vol. 4, No. 7, July 1966, pp. 1267-1275.

2. Dowell, E. H., "Nonlinear Oscillations of a Fluttering Plate II," AIAA Journal, Vol. 5, No. 10, 1967, pp. 1856-1862.

3. Ventres, C. S. and Dowell, E. H., "Comparison of Theory and Experiment for Nonlinear Flutter of Loaded Plates," AIAA Journal, Vol. 8, No. 11, 1970, pp. 2022-2030.

4. Ventres, C. S., "Nonlinear Flutter of Clamped Plates," Ph.D. thesis, Department of Aerospace and Mechanical Sciences, Princeton University, 1969.

5. Kappus, H. P., Lemley, C. E. and Zimmerman, N. H., "An Experimental Investigation of High Amplitude Panel Flutter," NASA CR-1837, May 1971.

6. Bolotin, V. V., "Nonconservative Problems of the Theory of Elastic Stability," (The MacMillan Co., New York, 1963) pp. 274-312.

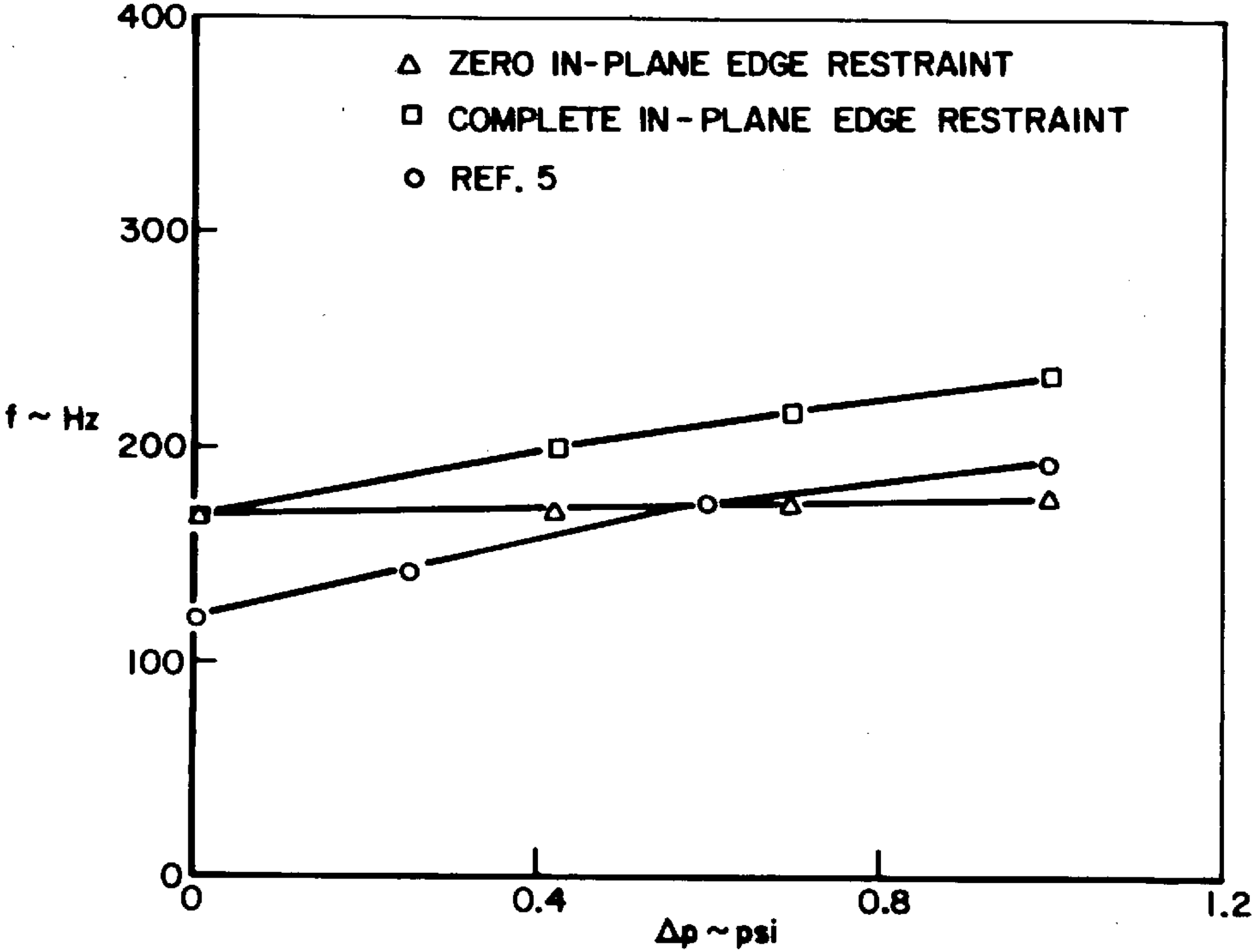
7. Dowell, E. H., "Generalized Aerodynamic Forces on a Flexible Plate Undergoing Transient Motion," Quarterly of Applied Mathematics, Vol. 24, No. 4, 1967, pp. 331-338.

8. Cunningham, H. J., "Flutter Analysis of Flat Rectangular Panels Based on Three-Dimensional Supersonic Unsteady Potential Flow," TR R-256, 1967, NASA.

9. Dowell, E. H., "Generalized Aerodynamic Forces on a Flexible Plate Undergoing Transient Motion in a Shear Flow with an Application to Panel Flutter," AIAA Journal, Vol. 9, No. 5, May 1971, pp. 834-841.

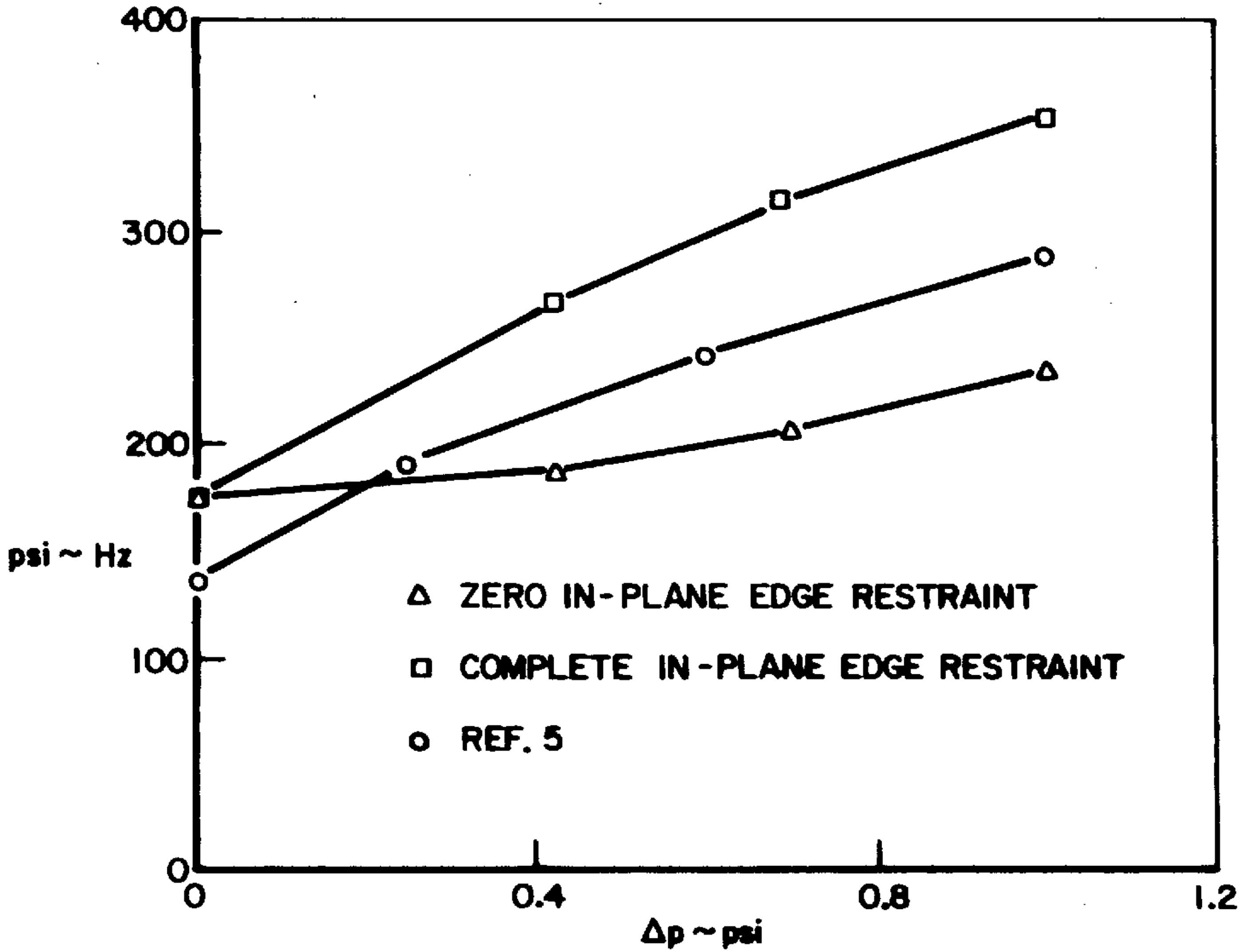
10. Ventres, C. S., "Flutter of a Buckled Plate Exposed to a Static Pressure Differential," AIAA Journal, Vol. 9, No. 5, May 1971, pp. 958-960.

LIST OF FIGURES



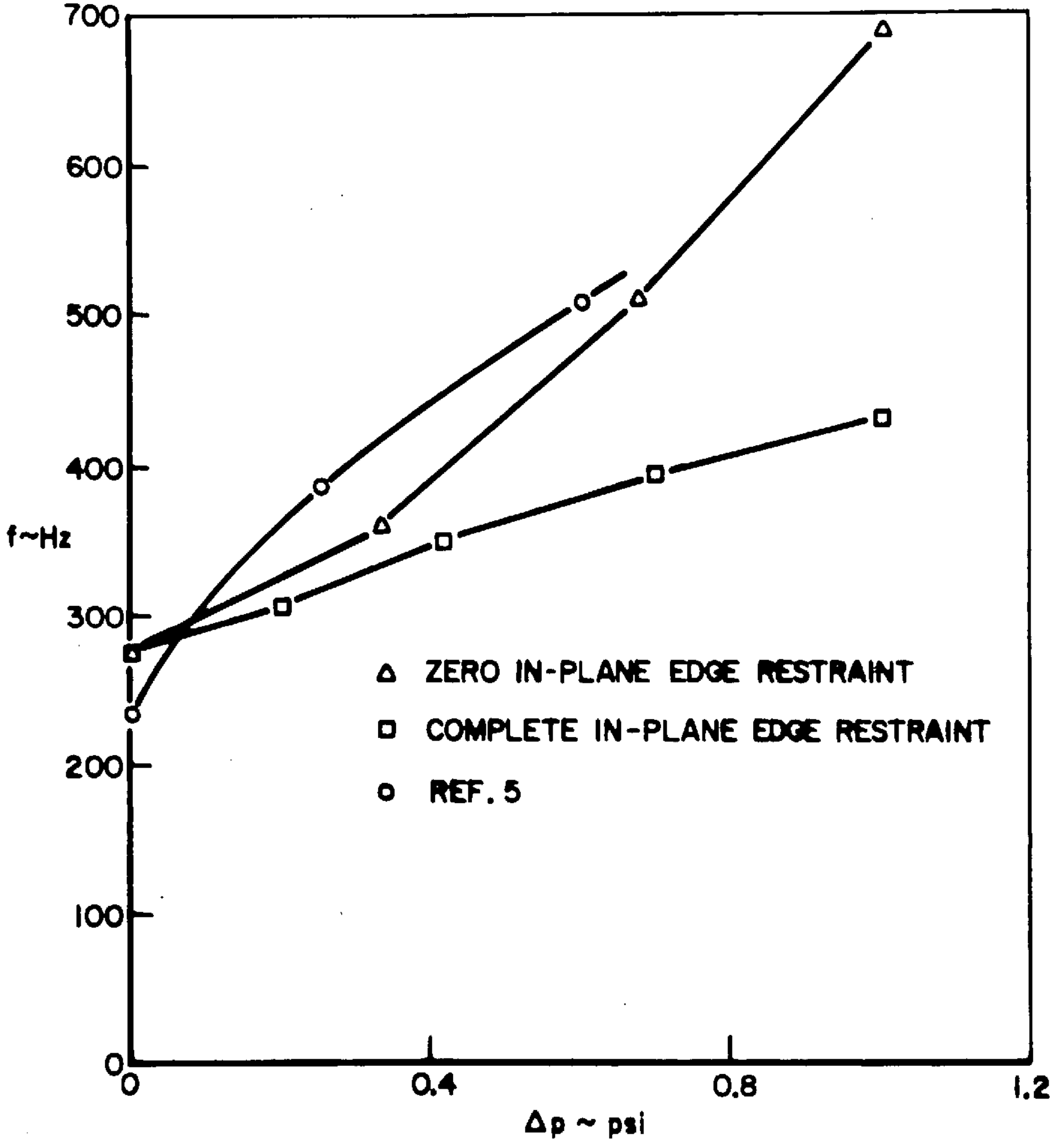
EFFECT OF Δp ON FREQUENCY
SPECTRA 1st MODE

FIGURE 1



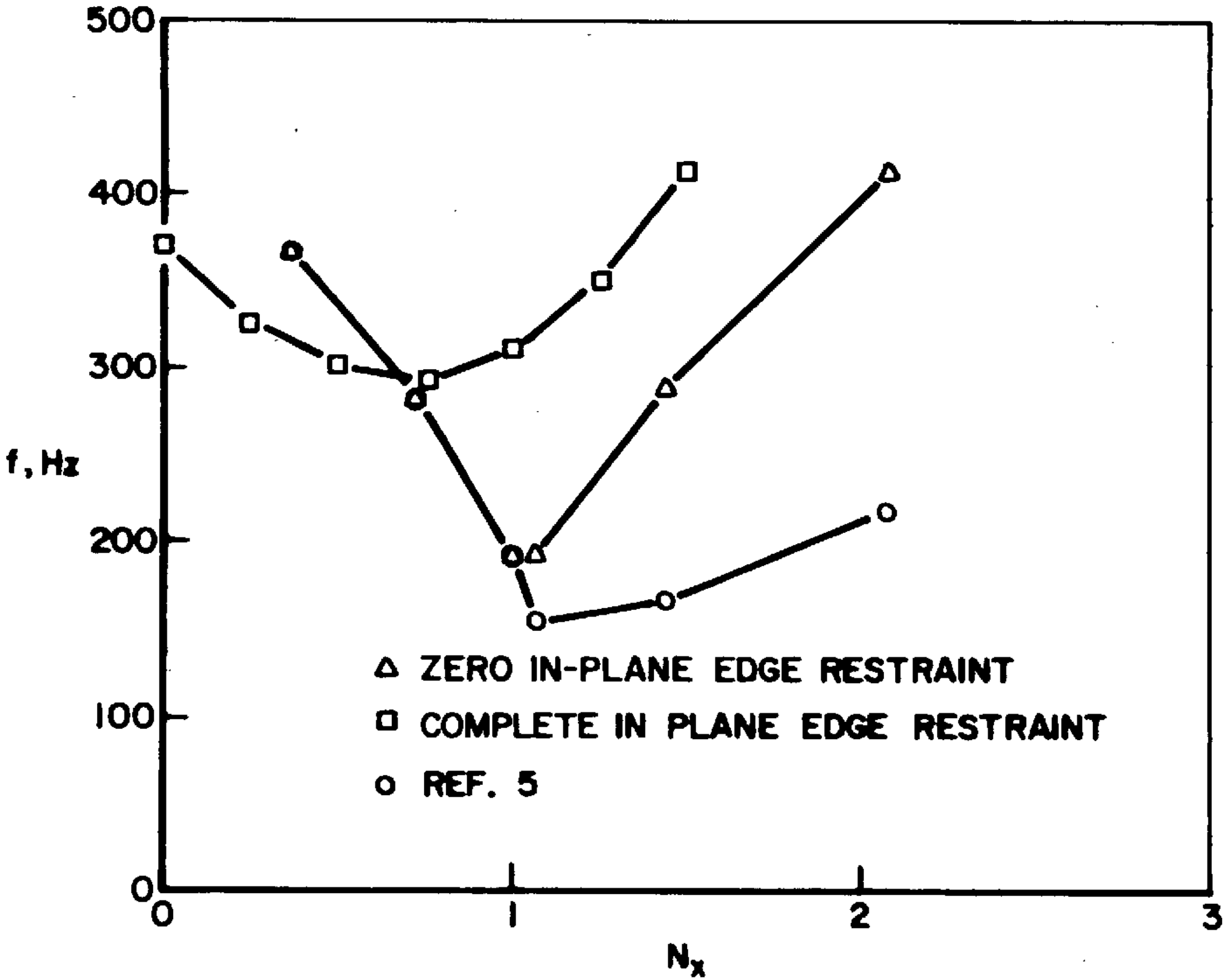
EFFECT OF Δp ON FREQUENCY SPECTRA 2nd MODE

FIGURE 2



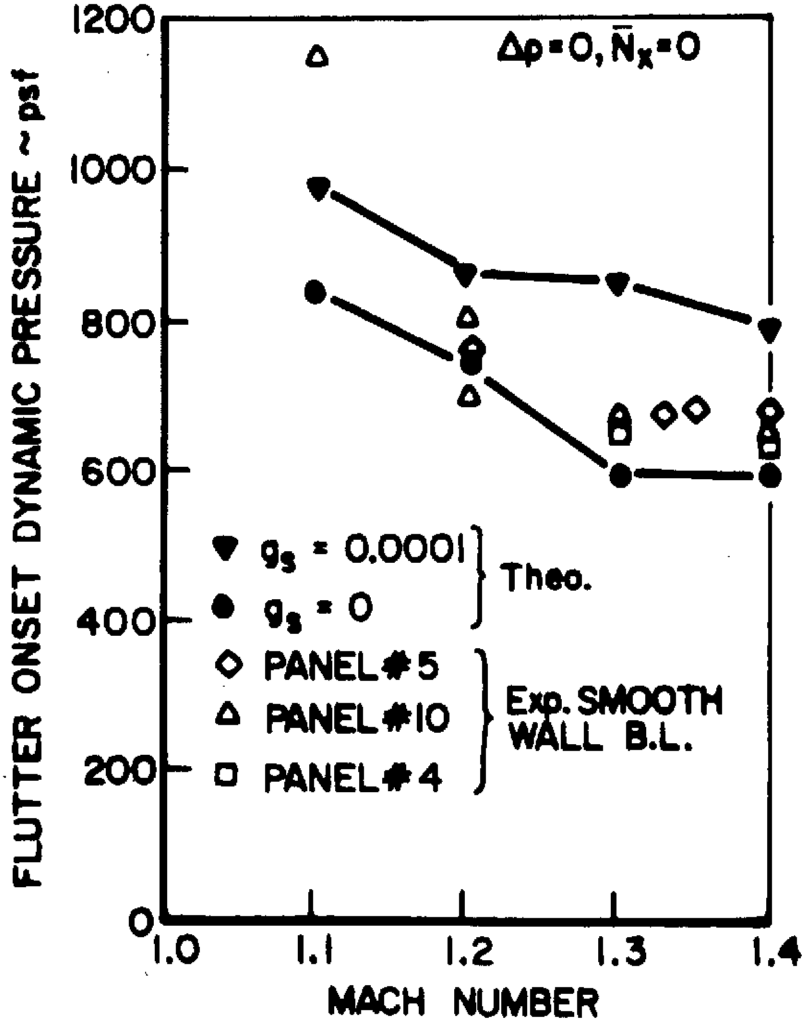
EFFECT OF Δp ON FREQUENCY SPECTRA 6th MODE

FIGURE 3



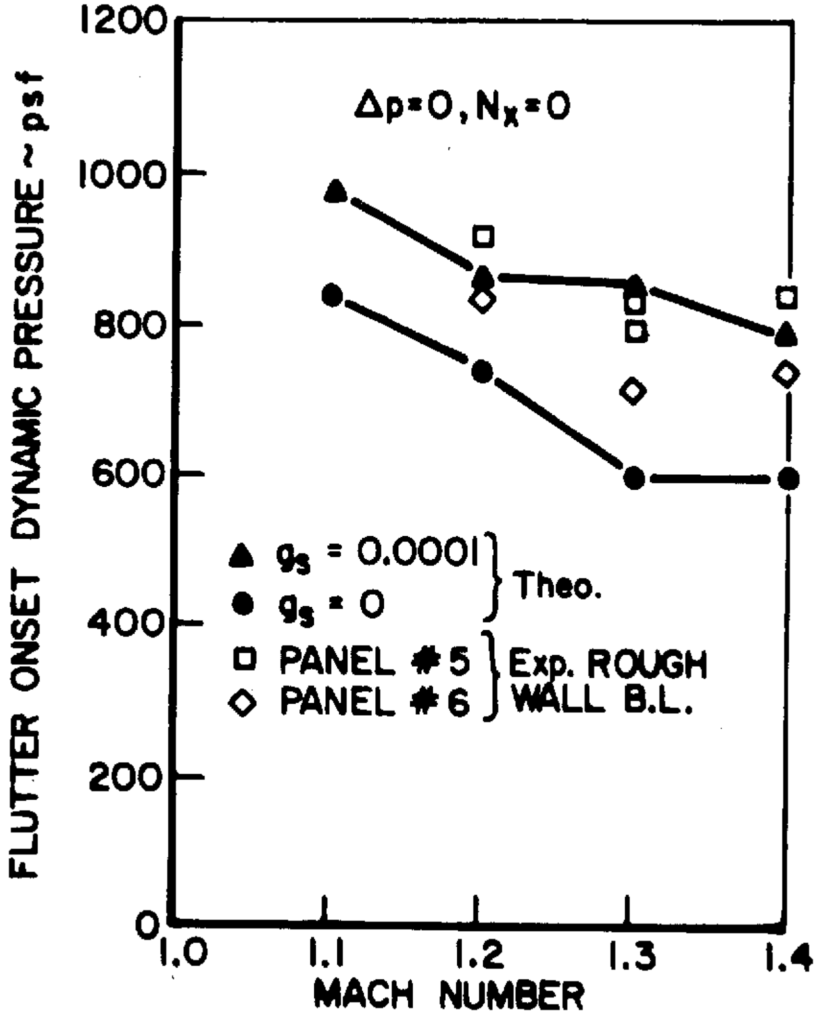
FREQUENCY OF NINTH MODE
vs IN-PLANE LOAD

FIGURE 4



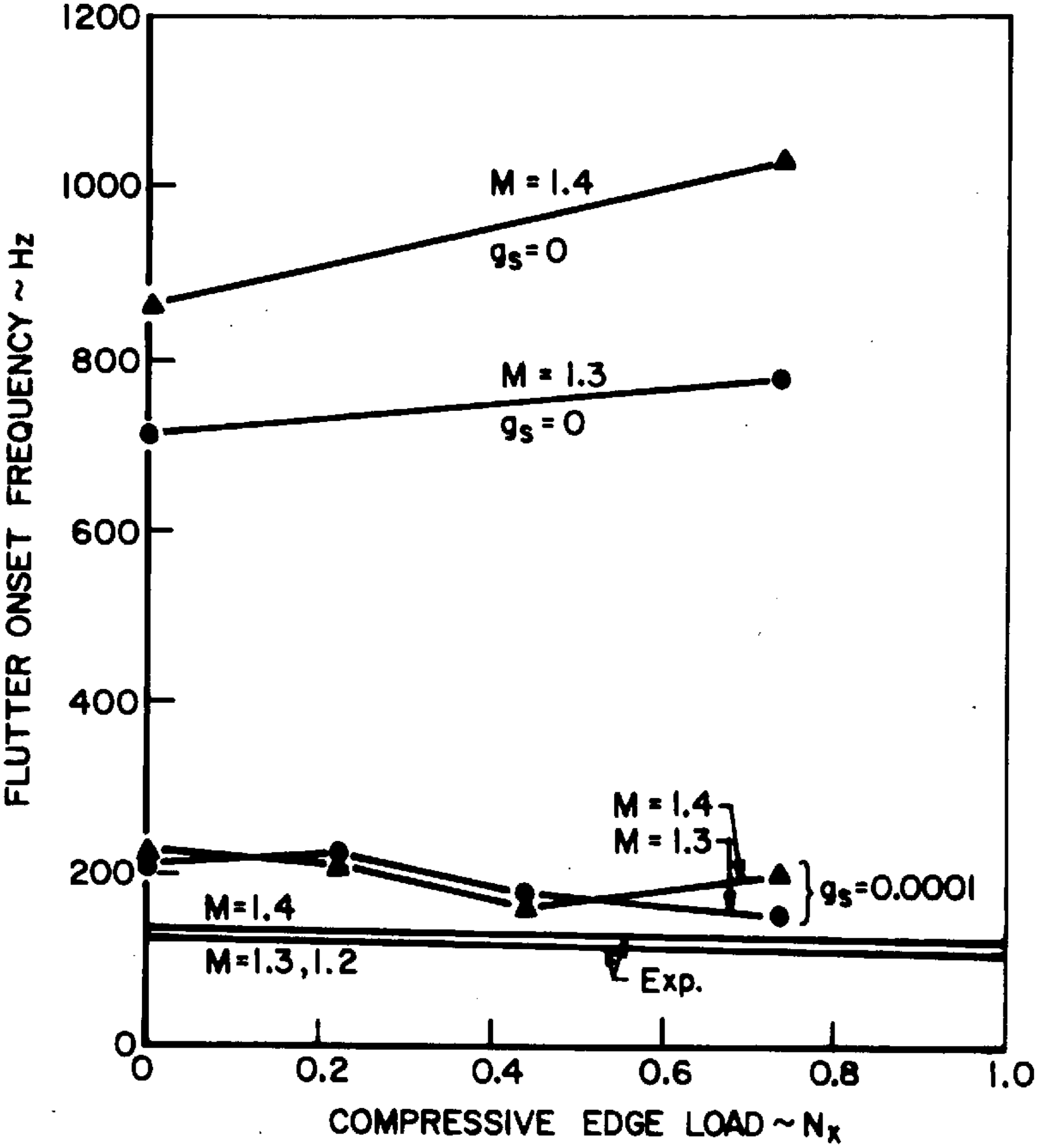
VARIATION OF ONSET DYNAMIC PRESSURE WITH MACH NUMBER

FIGURE 5



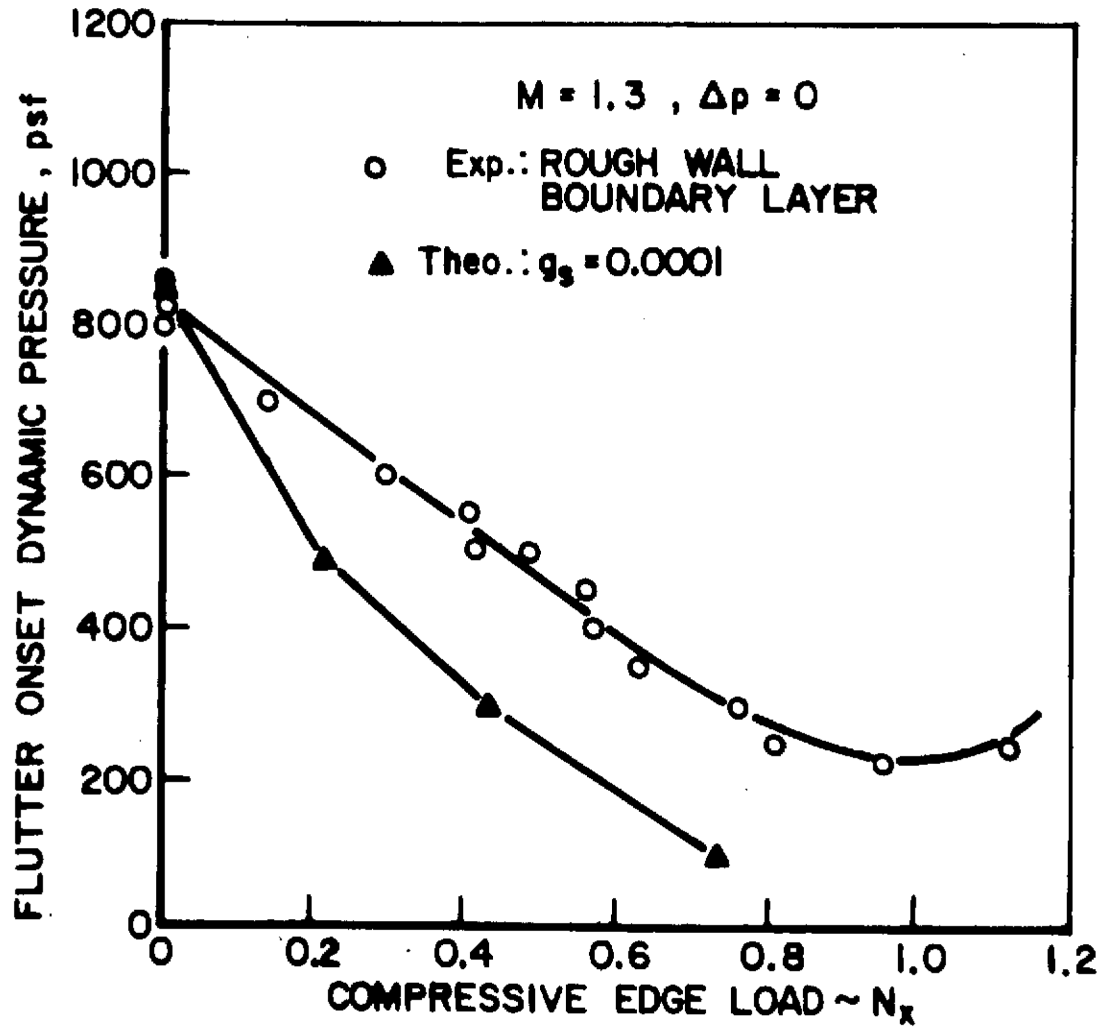
VARIATION OF ONSET DYNAMIC PRESSURE WITH MACH NUMBER

FIGURE 6



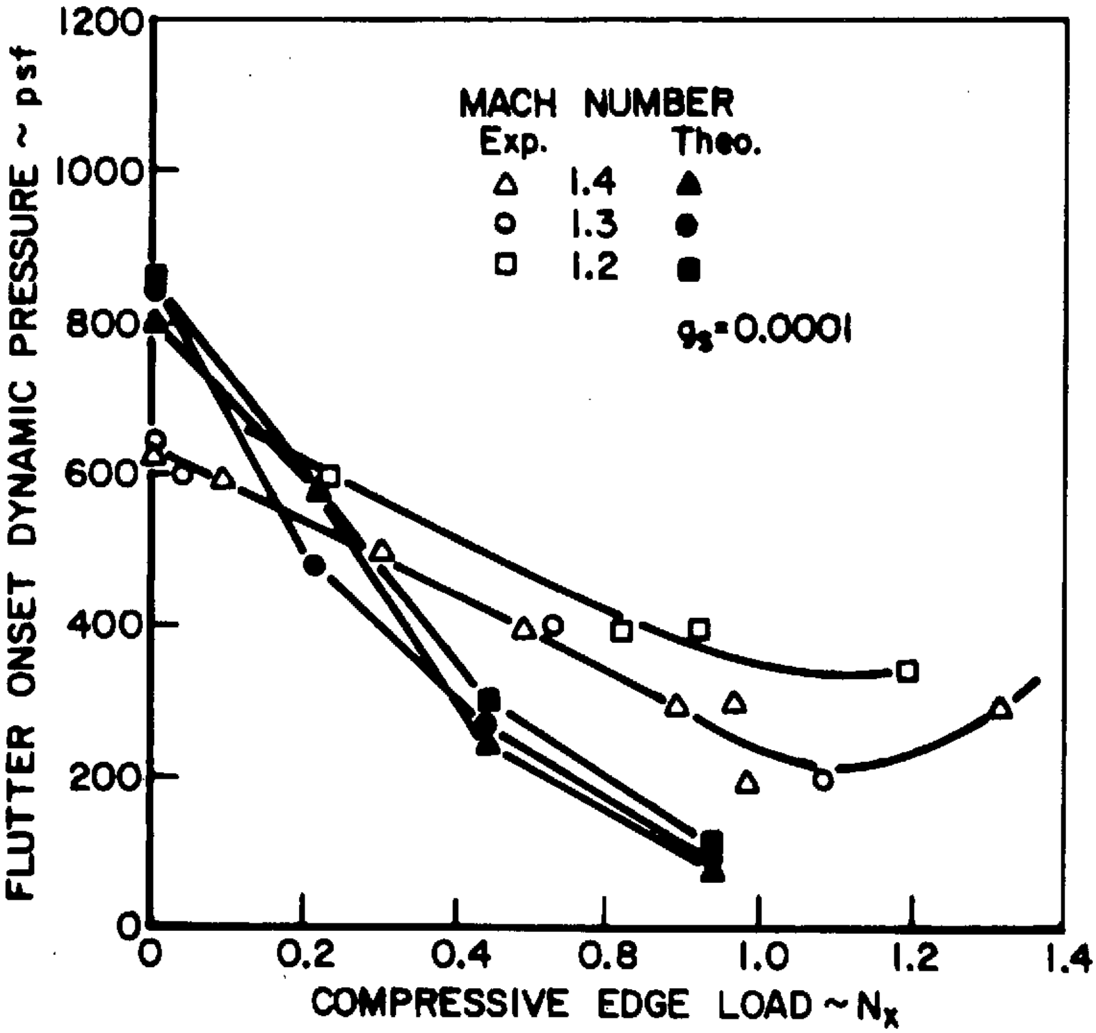
EFFECT OF DAMPING ON FLUTTER ONSET FREQUENCIES

FIGURE 7



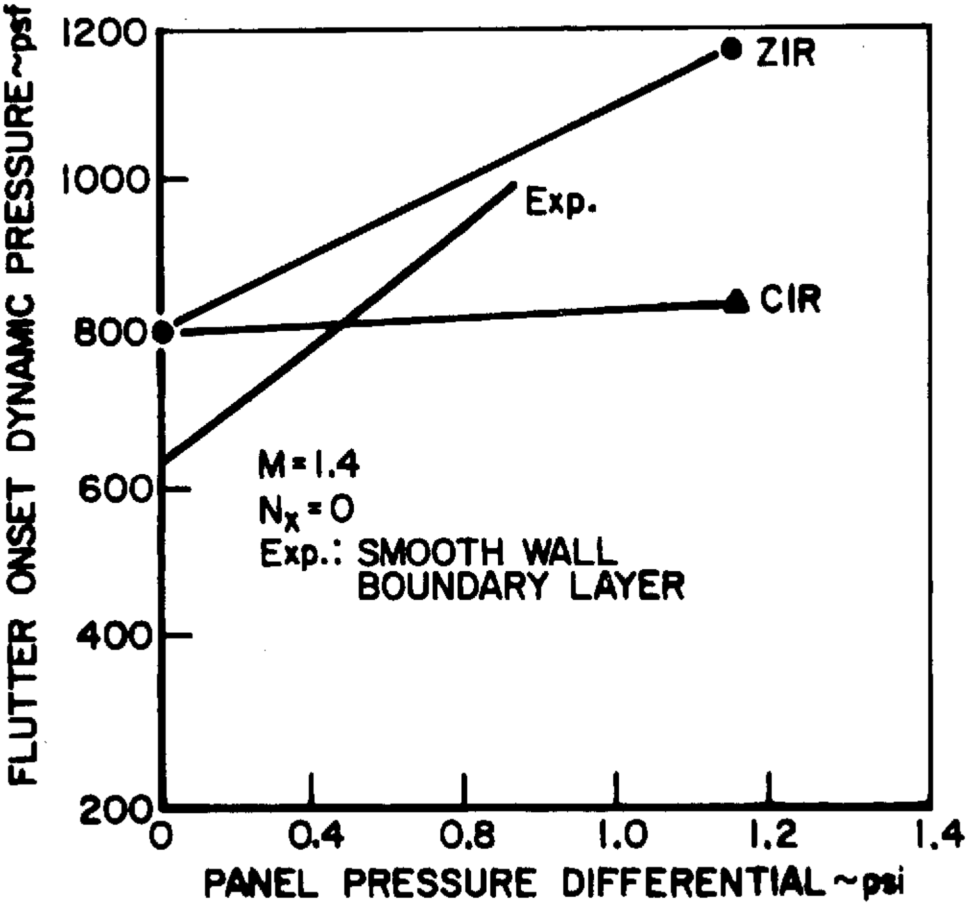
VARIATION OF FLUTTER ONSET DYNAMIC PRESSURE
WITH COMPRESSIVE EDGE LOAD

FIGURE 8



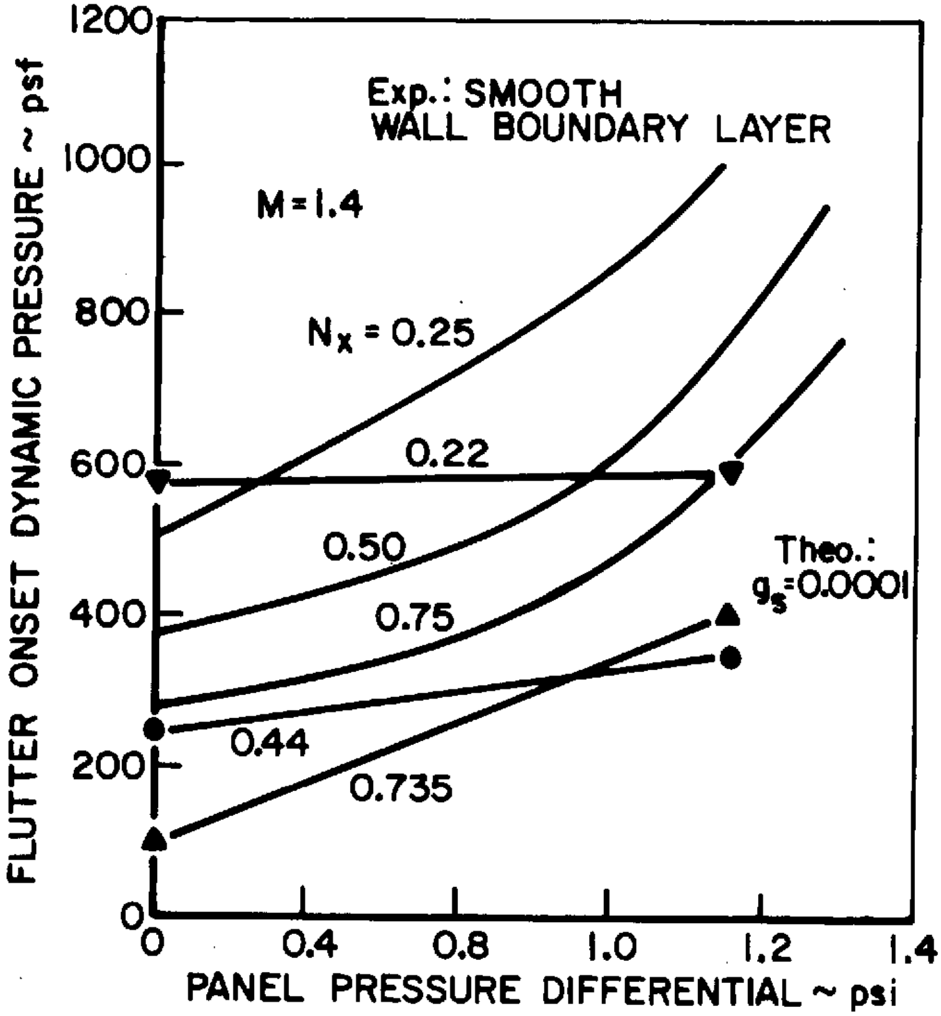
VARIATION OF FLUTTER ONSET DYNAMIC PRESSURE
WITH COMPRESSIVE EDGE LOAD

FIGURE 9



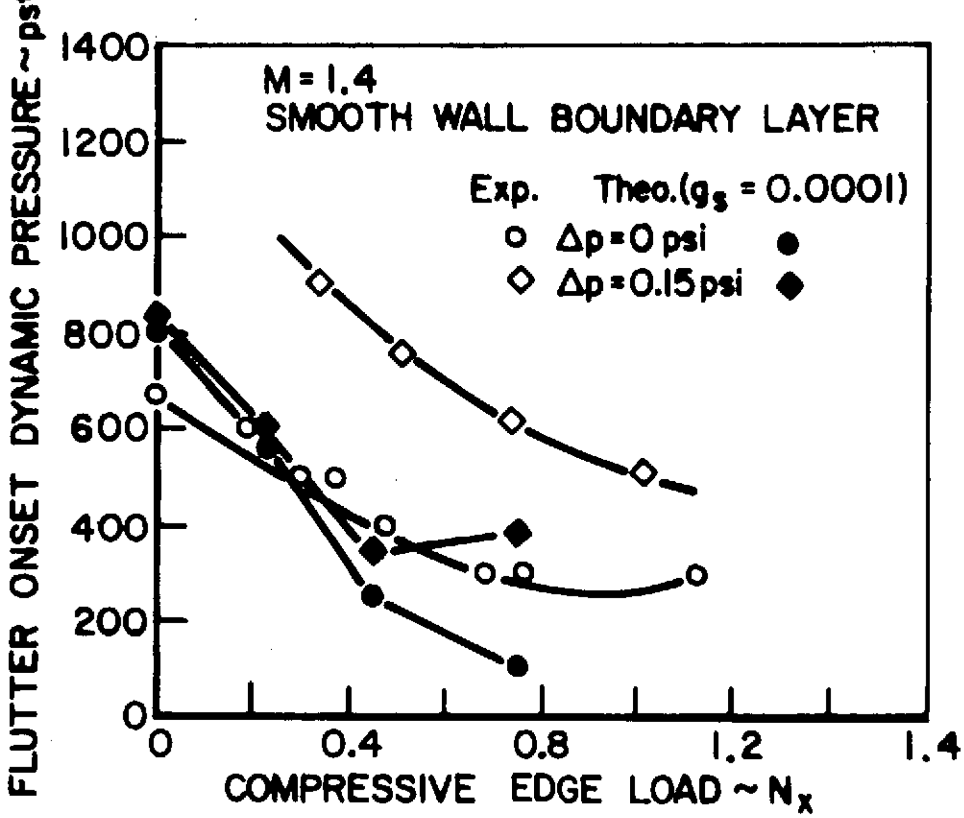
EFFECT OF Δp ON FLUTTER ONSET DYNAMIC PRESSURE (DIFFERENT BOUNDARY CONDITIONS)

FIGURE 10



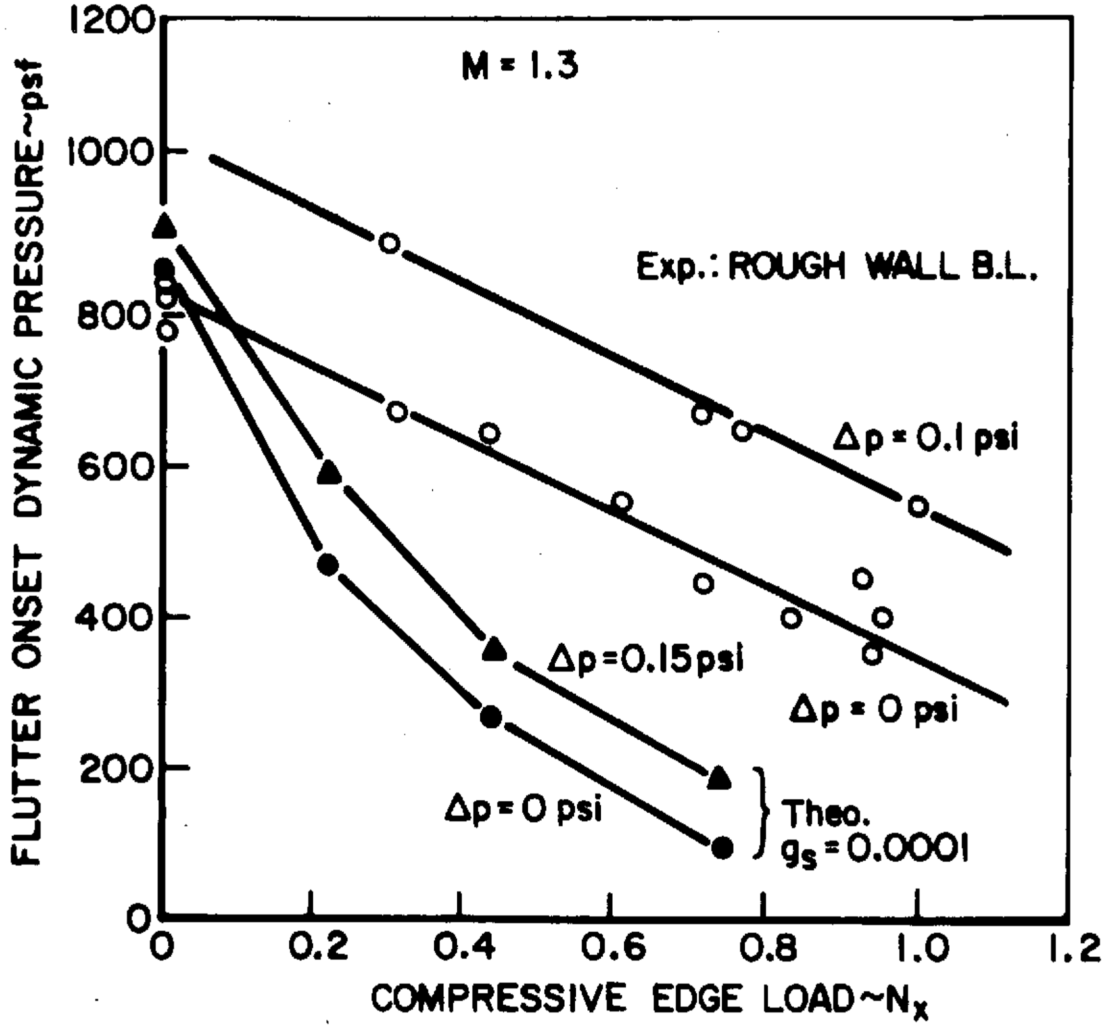
EFFECT OF Δp ON FLUTTER ONSET DYNAMIC PRESSURE (WITH VARIATION IN N_x)

FIGURE 11



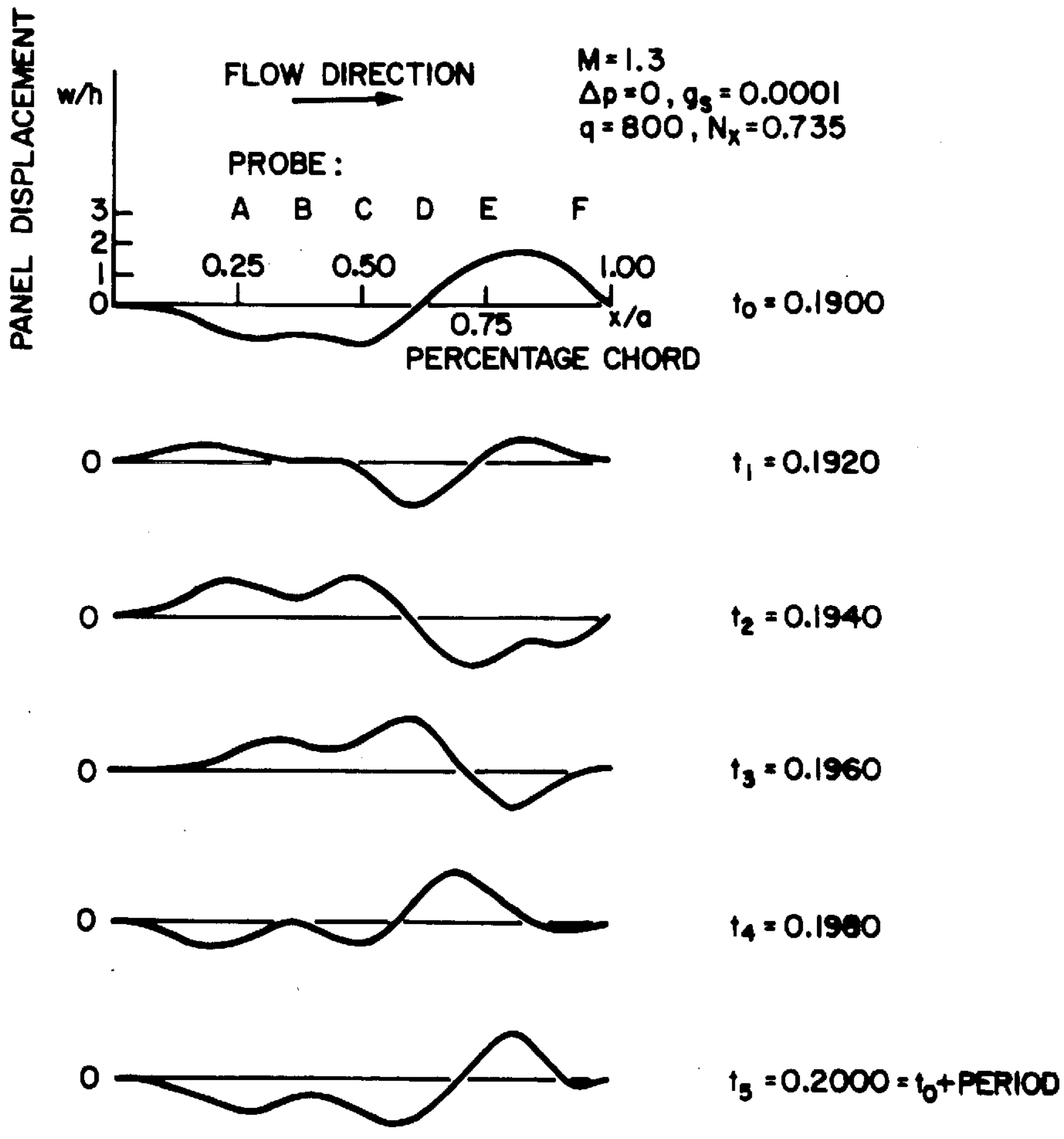
EFFECT OF N_x ON FLUTTER ONSET DYNAMIC PRESSURE

FIGURE 12



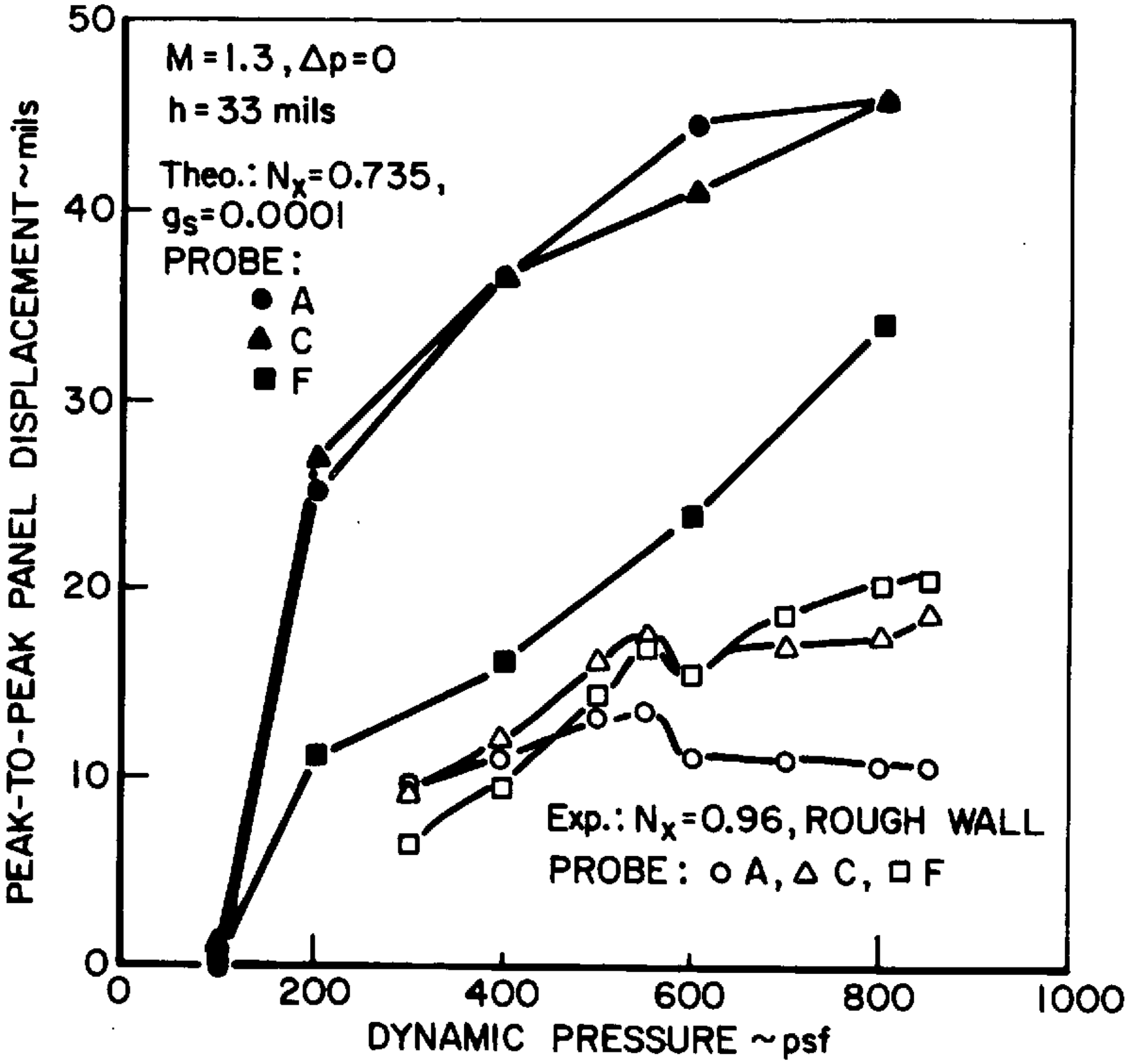
EFFECT OF Δp ON FLUTTER ONSET

FIGURE 13



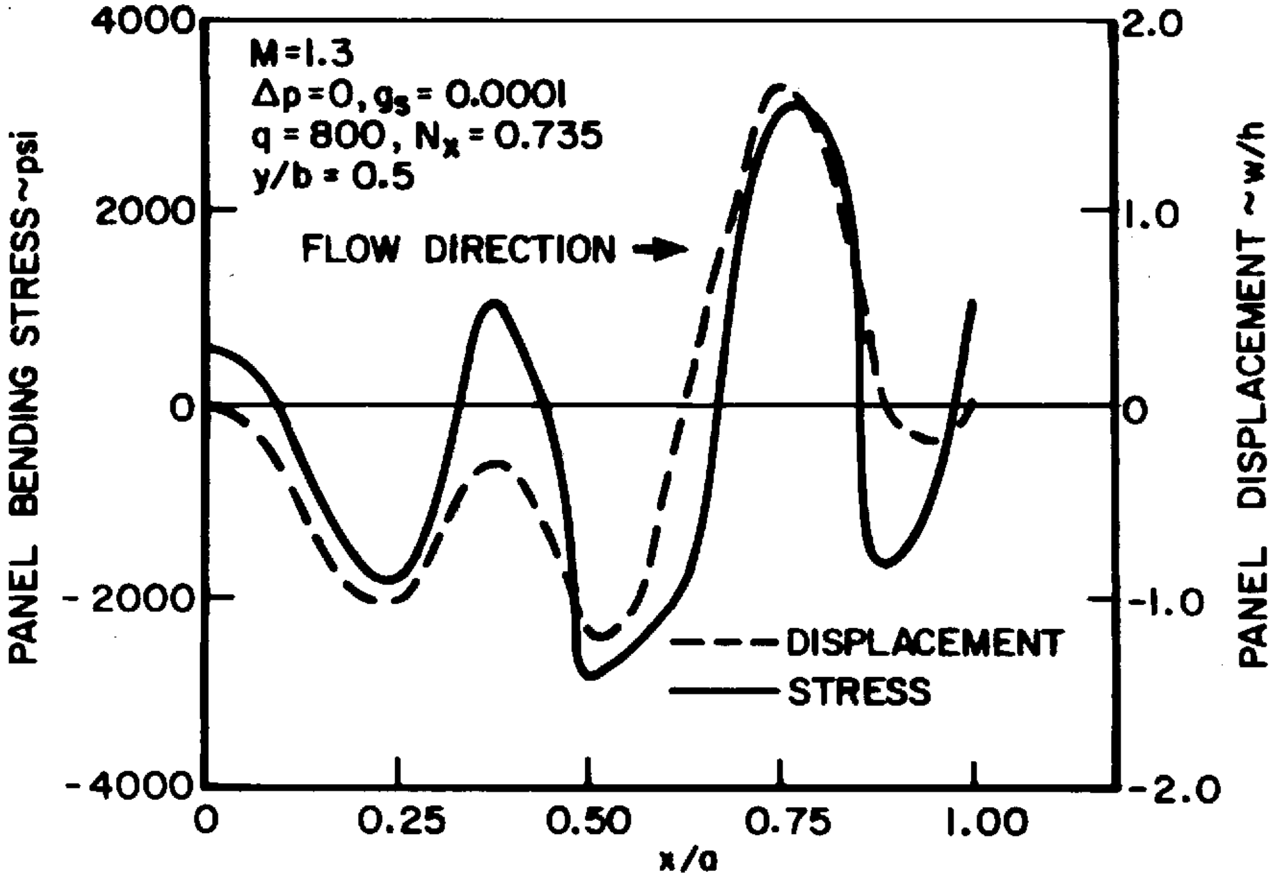
PANEL MOTION DURING FLUTTER

FIGURE 14



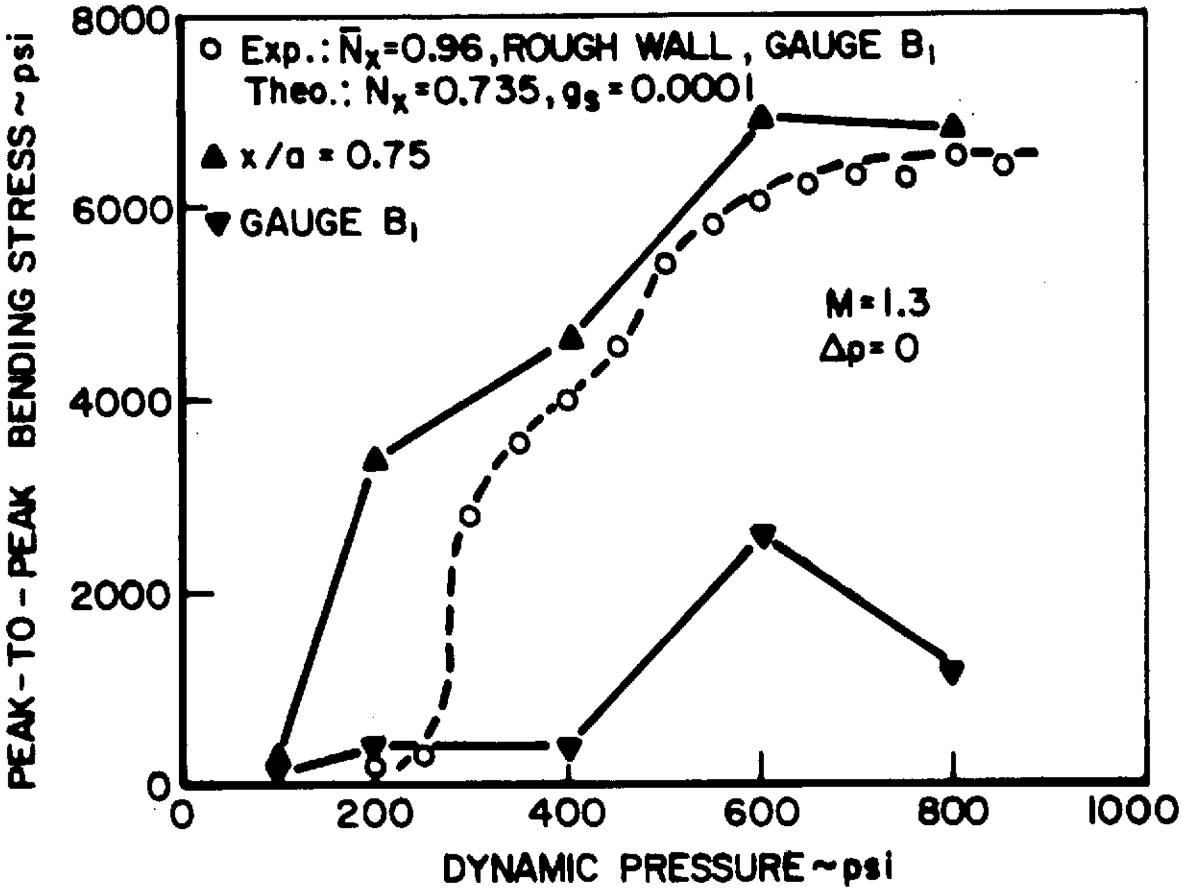
PANEL OSCILLATORY DISPLACEMENT DURING FLUTTER

FIGURE 15



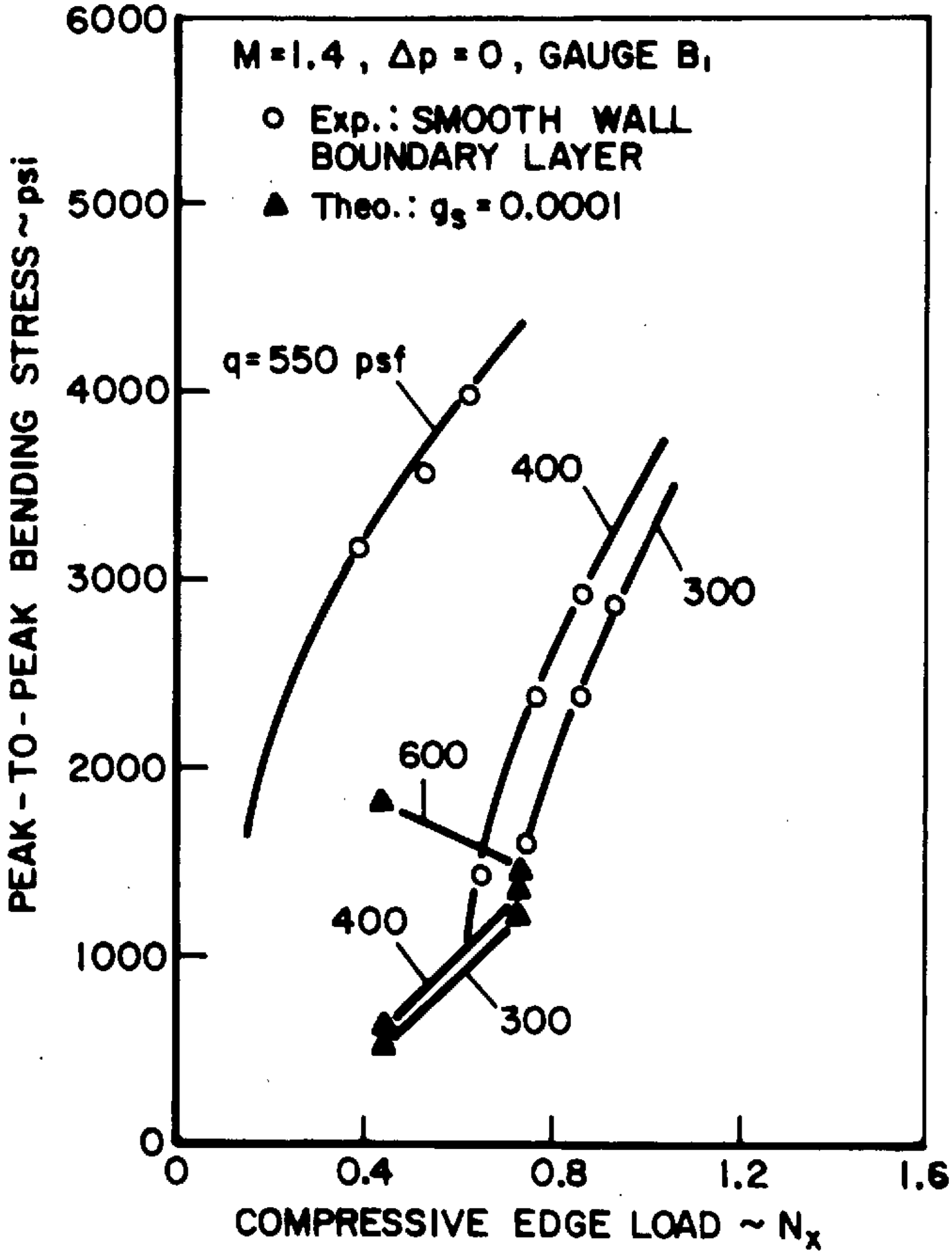
PANEL BENDING STRESS AND DISPLACEMENT

FIGURE 16



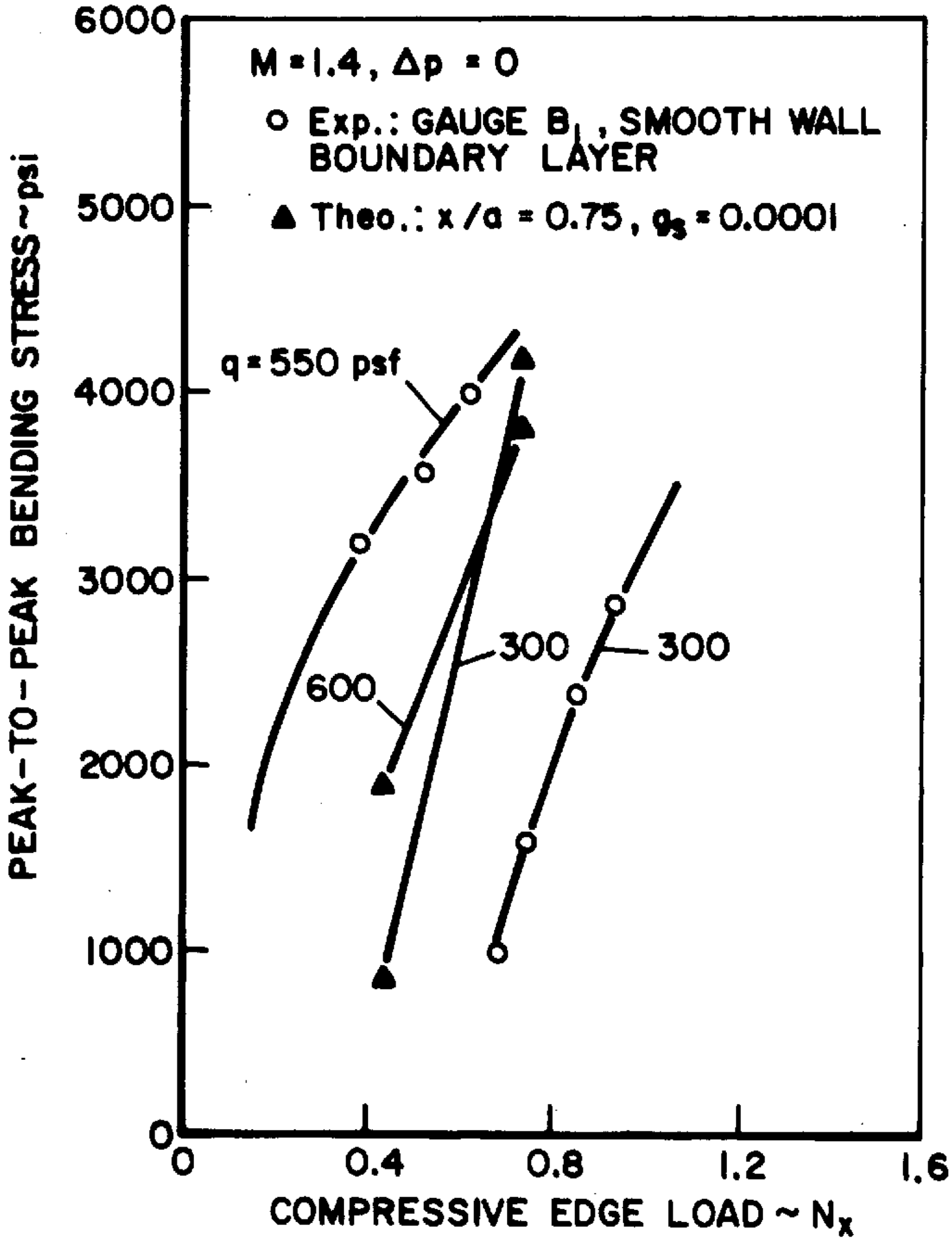
OSCILLATORY BENDING STRESS OF A BUCKLED PANEL DURING FLUTTER

FIGURE 17



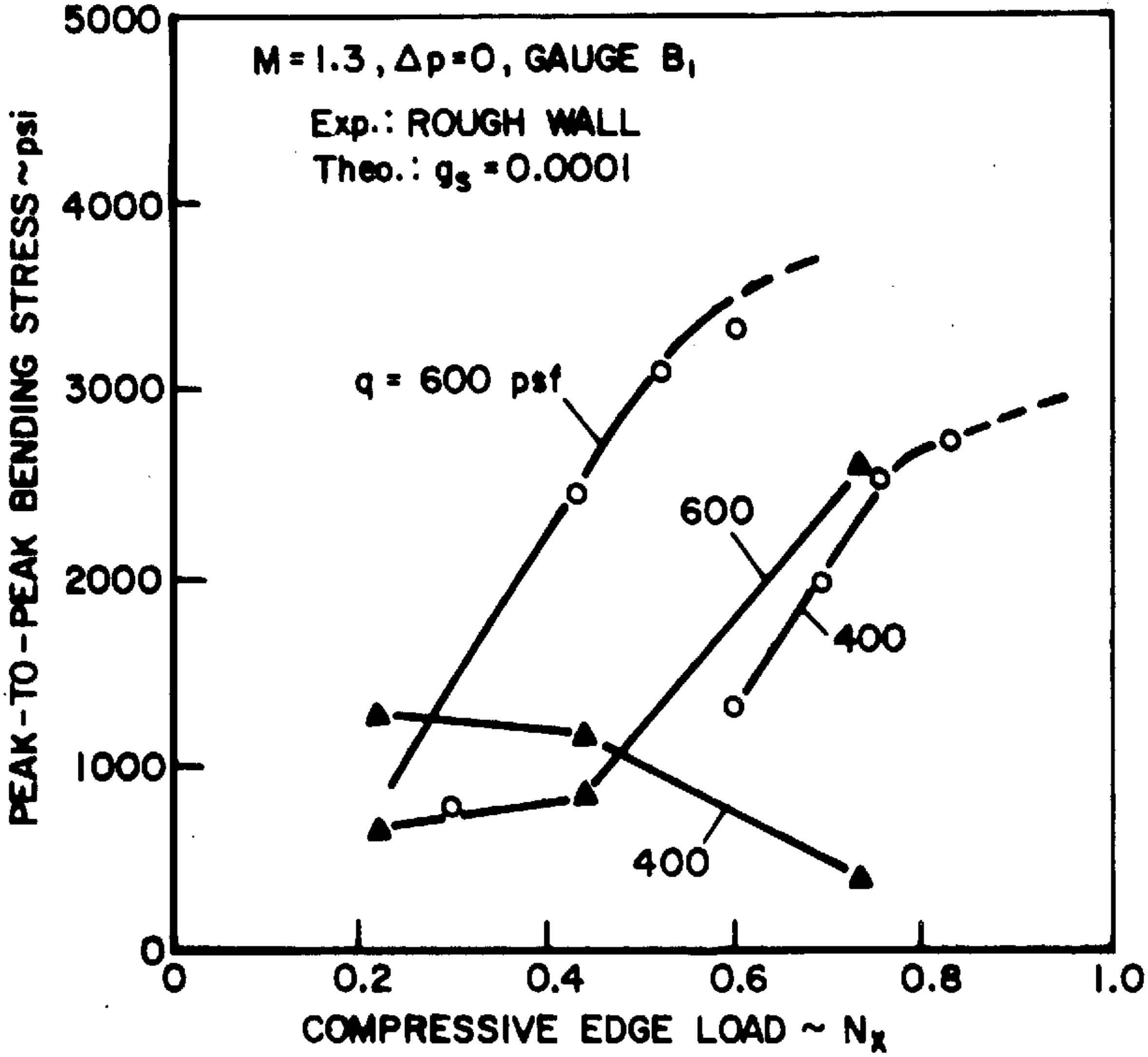
OSCILLATORY BENDING STRESS DURING FLUTTER

FIGURE 18 a



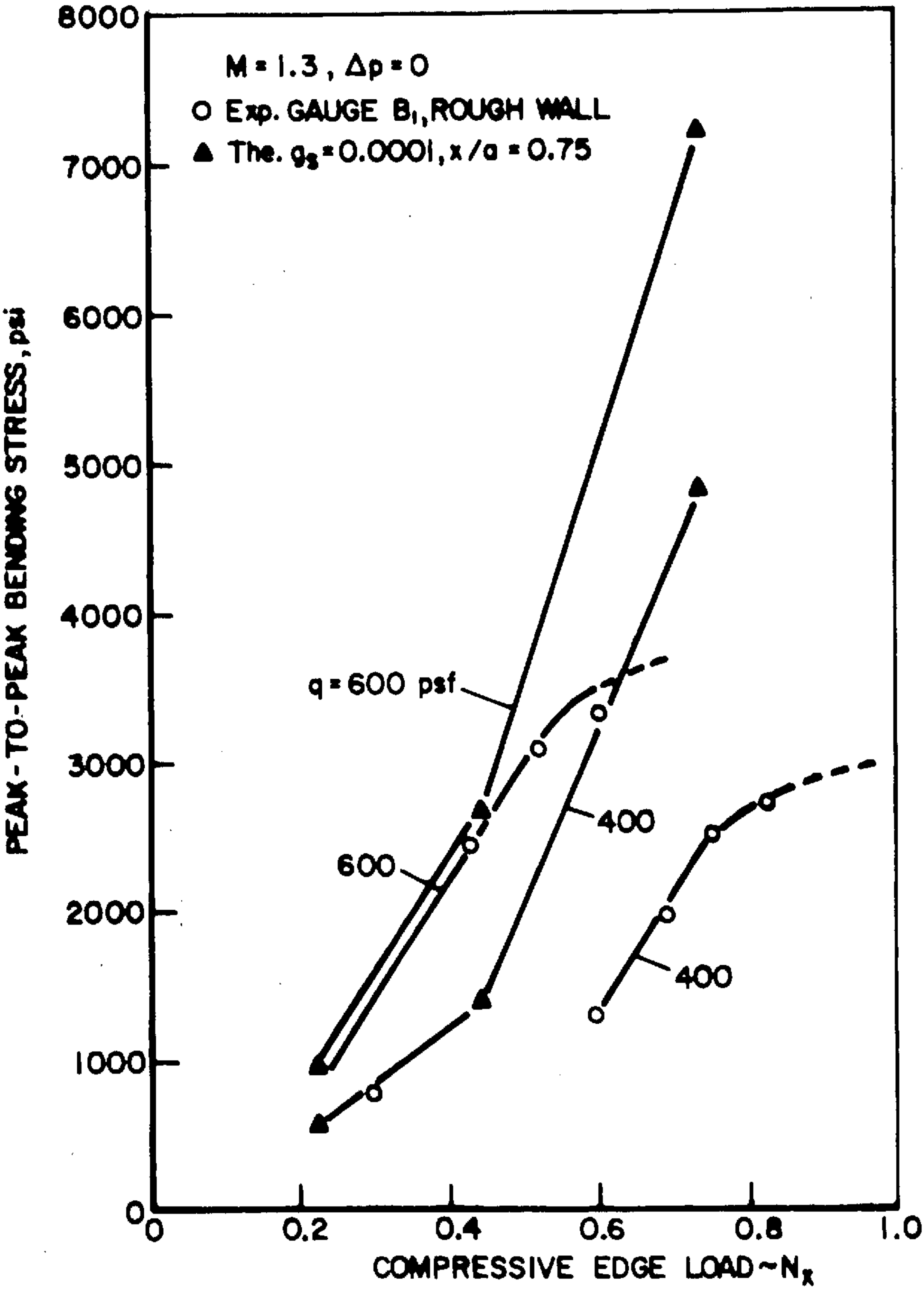
MAXIMUM BENDING STRESS DURING FLUTTER

FIGURE 18b



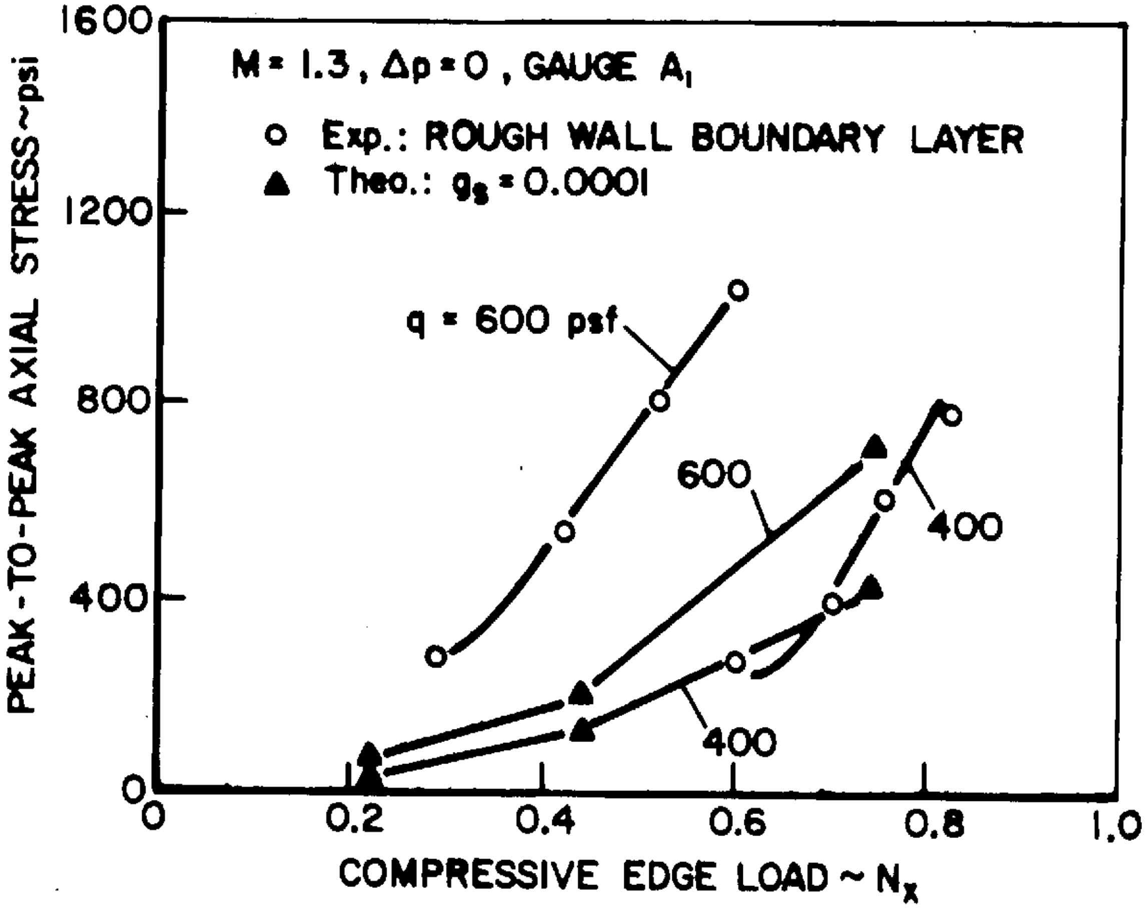
OSCILLATORY BENDING STRESS DURING FLUTTER

FIGURE 19a



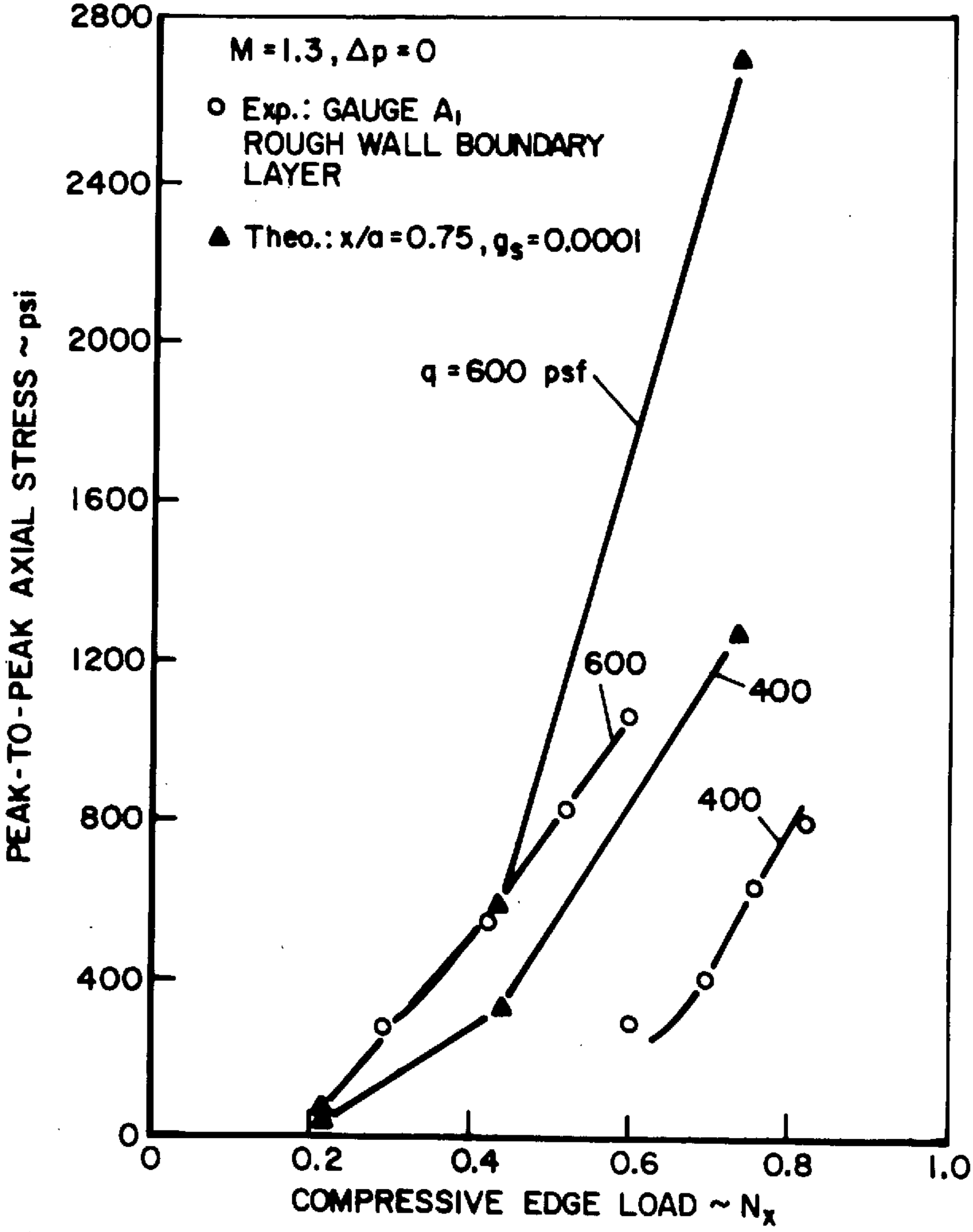
MAXIMUM BENDING STRESS DURING FLUTTER

FIGURE 19 b



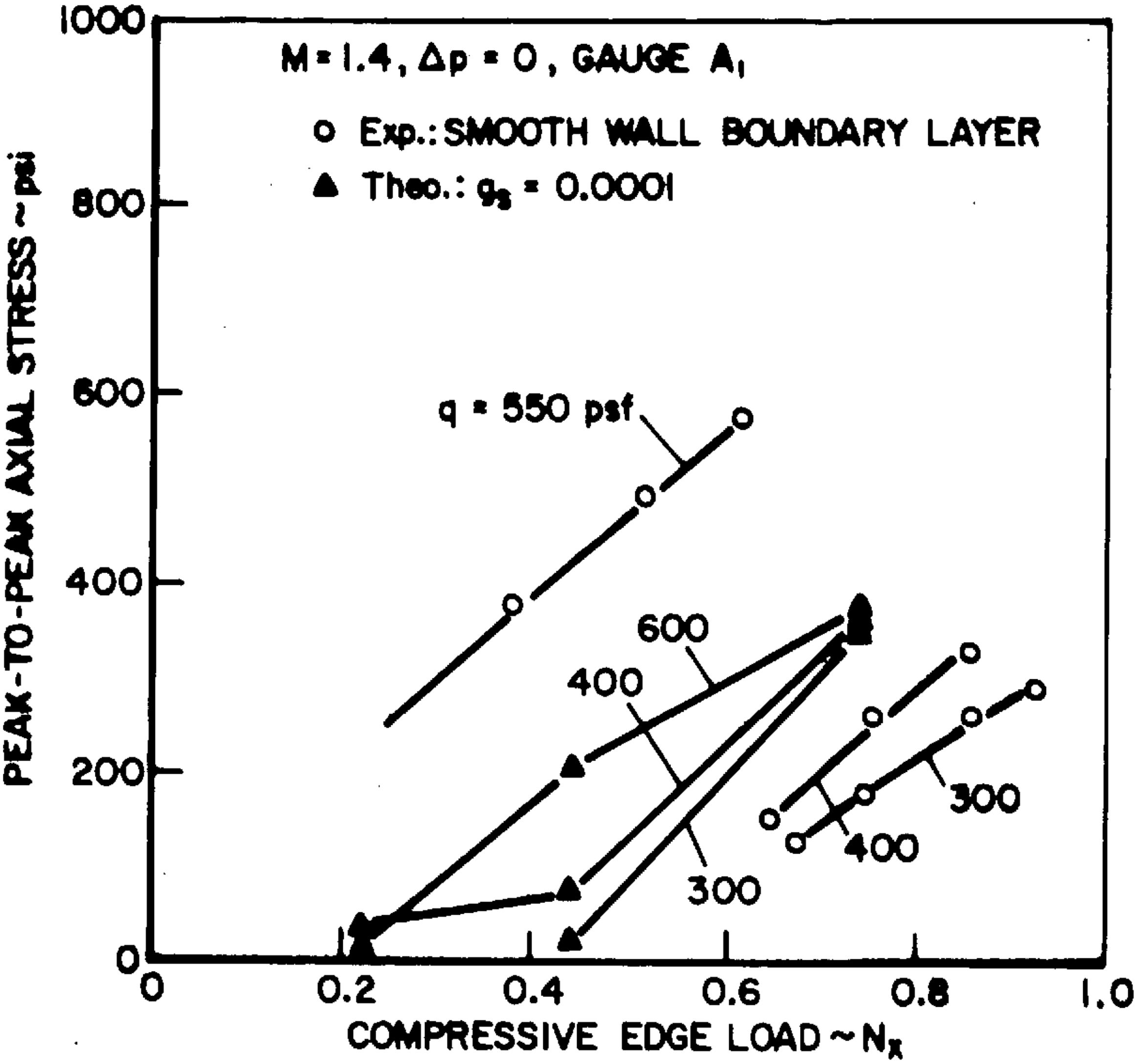
OSCILLATORY AXIAL STRESS DURING FLUTTER

FIGURE 20a



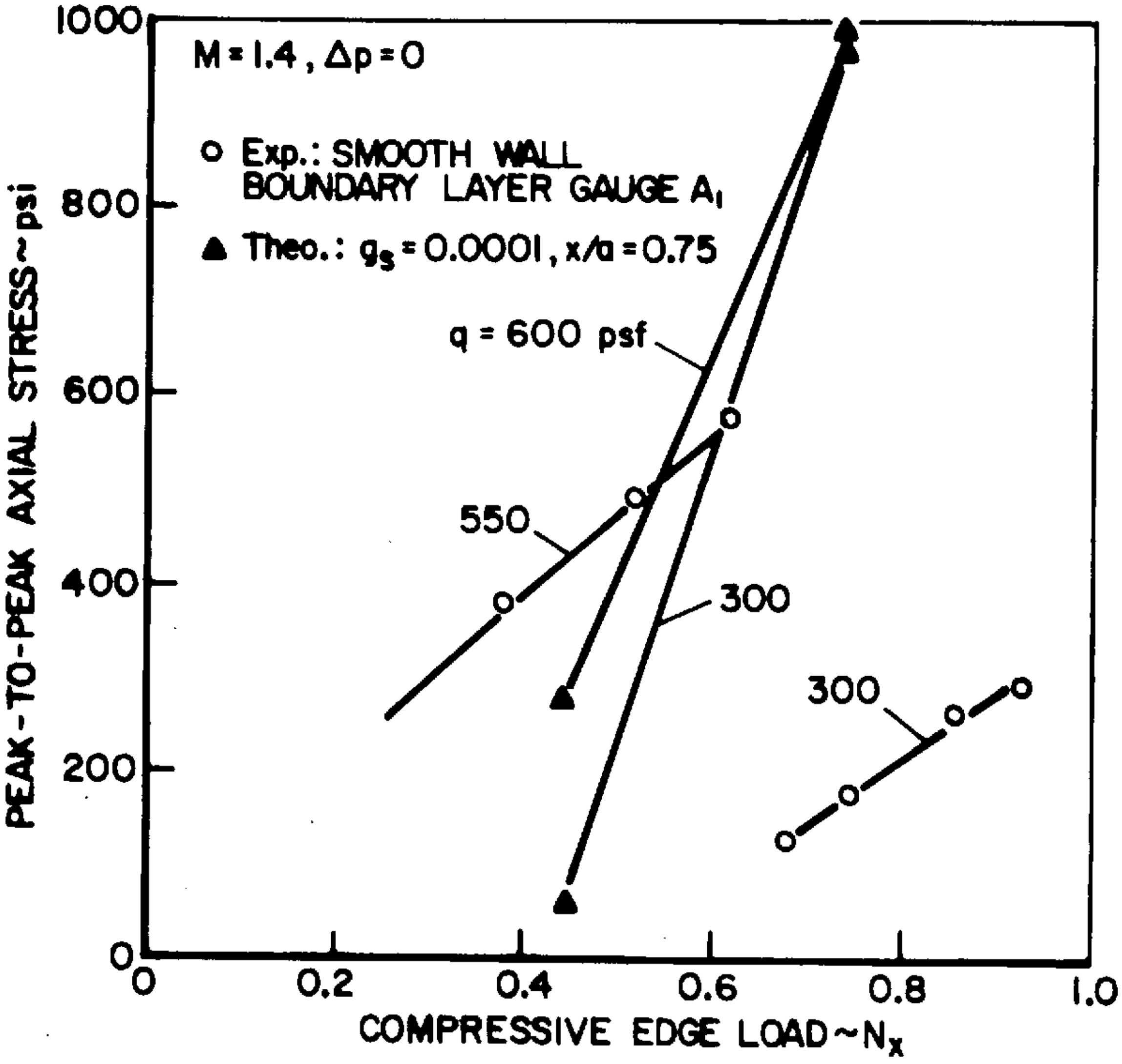
MAXIMUM AXIAL STRESS DURING FLUTTER

FIGURE 20b



OSCILLATORY AXIAL STRESS DURING FLUTTER

FIGURE 21a



MAXIMUM AXIAL STRESS DURING FLUTTER

FIGURE 21b

APPENDIX

Listing of Computer Programs

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
               C        PROGRAM TO COMPUTE NONLINEAR TERMS (COUPLING BETWEEN
               C        IN-PLANE STRETCHING AND OUT OF PLANE BENDING) FOR CLAMPED
               C        PLATE WITH COMPLETE IN-PLANE EDGE RESTRAINT
               C        AB=PLATE LENGTH/WIDTH RATIO,NU=POISSON'S RATIO
               C        NV = # OF MODES
 ISN 0002               REAL NU
 ISN 0003               REAL II
 ISN 0004               DIMENSION B(12,12,12,12)
 ISN 0005               DNA(K) = (FLOAT(K)**2+16.*AB2)**2
 ISN 0006               DNB(K) = (FLOAT(K)**2+4.*AB2)**2
 ISN 0007               CSS(K,L,M) = .5*(CC(K,L-M)-CC(K,L+M))
 ISN 0008               CCC(K,L,M) = .5*(CC(K,L-M) + CC(K,L+M))
 ISN 0009               GG(K,L,M) = CCC(K-1,L-1,M) -CCC(K-1,L+1,M)-CCC(K+1,L-1,M)
                      1 +CCC(K+1,L+1,M)
 ISN 0010               HH(K,L,M) = -PI2*FLOAT(L-1)**2*(CCC(K-1,L-1,M)
                      1 -CCC(K+1,L-1,M))
                      2 + PI2*FLOAT(L+1)**2*(CCC(K-1,L+1,M)-CCC(K+1,L+1,M))
 ISN 0011               II(K,L,M) = -PI*FLOAT(L-1)*(CSS(K-1,L-1,M)-CSS(K+1,L-1,M))
                      1 + PI*FLOAT(L+1)*(CSS(K-1,L+1,M)-CSS(K+1,L+1,M))
 ISN 0012      1        FORMAT (1H0)
 ISN 0013      2        FORMAT (1H1)
 ISN 0014               PI = 3.14159
 ISN 0015               PI2 = PI*PI
 ISN 0016               PI3 = PI2*PI
 ISN 0017               PI4 = PI3*PI
 ISN 0018               READ (5,110) AB,NU
 ISN 0019      110      FORMAT(4E10.3)
 ISN 0020               WRITE(6,1101) AB,NU
 ISN 0021      1101     FORMAT(2E12.3)
 ISN 0022               WRITE (6,1)
 ISN 0023               READ(5,120) NV
 ISN 0024               WRITE(6,120) NV
 ISN 0025      120      FORMAT(I5)
 ISN 0026               WRITE (6,1)
 ISN 0027               AB2 = AB**2
 ISN 0028               AB4 = AB**4
 ISN 0029               X = 1.0E-12
 ISN 0030               DO 42 M = 1,NV
 ISN 0031               DO 42 N = 1,NV
 ISN 0032               MA = M-N
 ISN 0033               MB = M-N-2
 ISN 0034               MC = M-N+2
 ISN 0035               MD = M+N
 ISN 0036               ME = M+N-2
 ISN 0037               MF = M+N+2
 ISN 0038               RM = M
 ISN 0039               RN = N
 ISN 0040               RMA = MA
 ISN 0041               RMB = MB
 ISN 0042               RMC = MC
 ISN 0043               RMD = MD
 ISN 0044               RME = ME
 ISN 0045               RMF = MF
 ISN 0046               BA = -2.*(RM*RMD+2.)/DNA(MA)
```

```
ISN 0047          BB = (RM-1.)*RMA/DNA(MB)
ISN 0048          BC = (RM+1.)*RMA/DNA(MC)
ISN 0049          BD = 2.*(RM*RMA+2.)/DNA(MD)
ISN 0050          BE = -(RM-1.)*RMA/DNA(ME)
ISN 0051          BF = -(RM+1.)*RMA/DNA(MF)
ISN 0052          BG = 4.*(RM**2+1.)/DNB(MA)
ISN 0053          BH = -2.*(RM-1.)**2/DNB(MB)
ISN 0054          BK = -2.*(RM+1.)**2/DNB(MC)
ISN 0055          BL = -4.*(RM**2+1.)/DNB(MD)
ISN 0056          BM = 2.*(RM-1.)**2/DNB(ME)
ISN 0057          BN = 2.*(RM+1.)**2/DNB(MF)
ISN 0058          BP = -2.*RM/(RMA**3+X)
ISN 0059          BQ = (RM-1.)/(RMB**3+X)
ISN 0060          BR = (RM+1.)/(RMC**3+X)
ISN 0061          BS = 2.*RM/RMD**3
ISN 0062          BT = -(RM-1.)/(RME**3+X)
ISN 0063          BU = -(RM+1.)/RMF**3
ISN 0064          DO 42 I = 1,NV
ISN 0065          DO 42 K = I,NV
ISN 0066          IODD = I+K+M+N
ISN 0067          IF(MOD(IODD,2).NE.0) GO TO 38
ISN 0069          BAA = (BA-BG)*HH(I,K,MA) + (BB-BH)*HH(I,K,MB) +
               1 (BC-BK)*HH(I,K,MC) + (BD-BL)*HH(I,K,MD) +
               2 (BE-BM)*HH(I,K,ME) + (BF-BN)*HH(I,K,MF)
ISN 0070          BAA = -48.*(1.-NU**2)*AB2*PI2*BAA
ISN 0071          BBB = RMA*(BA-BG)*II(I,K,MA) + RMB*(BB-BH)*II(I,K,MB)
               1 + RMC*(BC-BK)*II(I,K,MC) + RMD*(BD-BL)*II(I,K,MD) +
               2 RME*(BE-BM)*II(I,K,ME) + RMF*(BF-BN)*II(I,K,MF)
ISN 0072          BBB = -24.*(1.-NU**2)*AB2*PI3*BBB
ISN 0073          BCC = RMA**2*(BA-2.*BG+2.*BP)*GG(I,K,MA)
               1 + RMB**2*(BB-2.*BH+2.*BQ)*GG(I,K,MB)
               3 + RMC**2*(BC-2.*BK+2.*BR)*GG(I,K,MC)
               4 + RMD**2*(BD-2.*BL+2.*BS)*GG(I,K,MD)
               5 + RME**2*(BE-2.*BM+2.*BT)*GG(I,K,ME)
               6 + RMF**2*(BF-2.*BN+2.*BU)*GG(I,K,MF)
ISN 0074          BCC = 12.*(1.-NU**2)*AB2*PI4*BCC
ISN 0075          B(I,K,M,N) = -AB2*(BAA-2.*BBB+BCC)
ISN 0076          GO TO 42
ISN 0077     38   B(I,K,M,N) = 0.
ISN 0078     42   B(K,I,M,N) = B(I,K,M,N)
             C    WRITE NONLINEAR TERMS ONTO TAPE
ISN 0079          WRITE(10) B
ISN 0080          ENDFILE 10
ISN 0081          DO 45 I = 1,NV
ISN 0082          DO 45 J = 1,NV
ISN 0083          DO 45 K = 1,NV
ISN 0084     45   WRITE (6,47) (B(I,J,K,L), L = 1,NV)
ISN 0085     47   FORMAT (10E12.3)
ISN 0086          STOP
ISN 0087          END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002            FUNCTION CC(K,M)
  ISN 0003            CC = 0.
  ISN 0004            IF(K.EQ.M) CC = CC+.5
  ISN 0006            IF(K.EQ.-M) CC = CC+.5
  ISN 0008            RETURN
  ISN 0009            END
```

```
C       PROGRAM TO COMPUTE NONLINEAR TERMS (COUPLING
C       BETWEEN IN-PLANE STRETCHING AND OUT OF PLANE BENDING)
C       FOR PLATES WITH ZERO IN-PLANE EDGE RESTRAINT
C       AB=PLATE LENGTH/WIDTH RATIO,NU=POISSON'S RATIO,
C       NV=# OF MODES
        REAL NU
        DIMENSION B(12,12,12,12)
        DIMENSION AA(30,3,30,3),V(14,30,3,14),G(30,3,14,14)
        DIMENSION AM(90,90),AMW(90,90)
        DIMENSION AVW(180)
        REAL*8 AM,AMW
        REAL*8 AVW
        CSS(K,L,M) = .5*(CC(K,L-M)-CC(K,L+M))
        CCC(K,L,M) = .5*(CC(K,L-M) + CC(K,L+M))
        F(I,J) = CC(I,J)
        FF(N,L) = F(N-1,L-1) - F(N-1,L+1) - F(N+1,L-1) + F(N+1,L+1)
        FFF(K,M) = -FLOAT(M-1)**2*(F(M-1,K-1) - F(M-1,K+1))
      1 + FLOAT(M+1)**2*(F(M+1,K-1) - F(M+1,K+1))
        FFFF(K,M) = FLOAT(M-1)**4*(F(M-1,K-1) - F(M-1,K+1)) -
      1 FLOAT(M+1)**4*(F(M+1,K-1) - F(M+1,K+1))
        GCCA(L,I,J) = .5*(FLOAT((I-1)*(J-1))*(F(I-J,L-1)
      1 - F(I-J,L+1) - F(I+J-2,L-1) + F(I+J-2,L+1))
      2 - FLOAT((I-1)*(J+1))*(F(I-J-2,L-1) - F(I-J-2,L+1)
      3 - F(I+J,L-1) + F(I+J,L+1)) - FLOAT((I+1)*(J-1))*(F(I-J+2,L-1)
      4 - F(I-J+2,L+1) - F(I+J,L-1) + F(I+J,L+1)) +FLOAT((I+1)*(J+1))
      5 *(F(I-J,L-1) - F(I-J,L+1) - F(I+J+2,L-1) + F(I+J+2,L+1)))
        GCCB(L,J,I) = -.5*(FLOAT(I-1)**2*(F(I+J-2,L-1) - F(I+J-2,L+1)
      1 + F(I-J,L-1) - F(I-J,L+1) -F(I+J,L-1) + F(I+J,L+1)
      2 - F(I-J-2,L-1) + F(I-J-2,L+1)) - FLOAT(I+1)**2*
      3 (F(I+J,L-1) - F(I+J,L+1) + F(I-J+2,L-1) - F(I-J+2,L+1)
      4 - F(I+J+2,L-1) + F(I+J+2,L+1) - F(I-J,L-1) + F(I-J,L+1)))
        PI = 3.14159
        PI2 = PI*PI
        PI3 = PI2*PI
        PI4 = PI3*PI
        READ(5,110) AB,NU
  110   FORMAT(4E10.3)
        WRITE(6,1101) AB,NU
 1101   FORMAT(1H0,1P4E10.3)
        WRITE (6,1)
        READ(5,120) NV
        WRITE(6,120) NV
  120   FORMAT(I5)
        WRITE (6,1)
    1   FORMAT (1H0)
    2   FORMAT (1H1)
        AB2 = AB**2
        AB4 = AB**4
C       COMPUTE B
        NX = 2*NV+2
        NY = 3
        DO 10 K = 1,NX
        DO 10 I = 1,NY
        L = 2*I - 1
        DO 10 M = 1,NX
        DO 10 J = 1,NY
```

```
0040              N = 2*J - 1
0041             AA(K,I,M,J) = FFFF(K,M)*FF(L,N) + 2.*AB2*FFF(K,M)*FFF(L,N)
                1 + AB4*FF(K,M)*FFFF(L,N)
0042       10     CONTINUE
           C      INVERT AA AND STORE IN AA
0043              NAM = NX*NY
0044              I = 0
0045              J = 0
0046              DO 512 K = 1,NX
0047              DO 512 L = 1,NY
0048              I = I+1
0049              DO 508 M = 1,NX
0050              DO 508 N = 1,NY
0051              J = J+1
0052       508    AM(I,J) = AA(K,L,M,N)
0053       512    J = 0
0054              CALL MATIN2(AM,NAM,90,AMW,90,AVW,KINV,180)
0055              WRITE (6,1)
0056              WRITE (6,511) KINV
0057       511    FORMAT (I10)
0058              I = 0
0059              J = 0
0060              DO 522 K = 1,NX
0061              DO 522 L = 1,NY
0062              I = I+1
0063              DO 518 M = 1,NX
0064              DO 518 N = 1,NY
0065              J = J+1
0066       518   AA(K,L,M,N) = AM(I,J)
0067       522    J = 0
0068              DO 14 I = 1,NV
0069              DO 14 K = 1,NX
0070              DO 14 J = 1,NY
0071              L = 2*J - 1
0072              DO 14 M = 1,NV
0073              V(I,K,J,M) = AB2*(GCCB(I,K,M)*GCCB(1,1,L) - 2.*GCCA(I,K,M)
                1 *GCCA(1,L,1) + GCCB(I,M,K)*GCCB(1,L,1))*PI4
0074       14     CONTINUE
0075              DO 16 K = 1,NX
0076              DO 16 J = 1,NY
0077              L = 2*J - 1
0078              DO 16 M = 1,NV
0079              DO 16 N = 1,NV
0080              G(K,J,M,N) = 12.*(1. - NU**2)*AB2*(GCCA(K,M,N)*GCCA(L,1,1)
                1 - GCCB(K,N,M)*GCCB(L,1,1))
0081       16     CONTINUE
0082              DO 22 I = 1,NV
0083              DO 22 L = I,NV
0084              IVSUM = I+L
0085              DO 22 J = 1,NV
0086              DO 22 K = 1,NV
0087              IGSUM = J+K
0088              B(I,J,K,L) = 0.
0089              IS = I+J+K+L
0090              M = IS-2*(IS/2)
0091              IF(M.NE.0) GO TO 21
```

```
               IVS = IVSUM-2*(IVSUM/2)+1
               DO 18 IM = IVS,NX,2
               IGS = IGSUM-2*(IGSUM/2)+1
               DO 18 KK = IGS,NX,2
               DO 18 JJ = 1,NY
               DO 18 LL = 1,NY
      18       B(I,J,K,L) = B(I,J,K,L) + V(I,IM,JJ,L)*AA(IM,JJ,KK,LL)
             1 *G(KK,LL,J,K)
               B(I,J,K,L) = -B(I,J,K,L)
      21       CONTINUE
               B(L,J,K,I) = B(I,J,K,L)
      22       CONTINUE
      C        WRITE NONLINEAR TERMS ONTO TAPE
               WRITE(10) B
               ENDFILE 10
               WRITE (6,2)
               DO 45 I = 1,NV
               DO 45 J = 1,NV
               DO 45 K = 1,NV
      45       WRITE (6,47) (B(I,J,K,L), L = 1,NV)
      47       FORMAT(10E12.4)
               STOP
               END
```

```
      FUNCTION CC(K,M)
      CC = 0.
      IF(K.EQ.M) CC = CC+.5
      IF(K.EQ.-M) CC = CC+.5
      RETURN
      END
```

```
0001                 SUBROUTINE MATIN2(A,N1,IA,X,IX,B,INT,N2)
               C     THIS SUBROUTINE INVERTS THE UPPER LEFT N1 BY N1 CORNER OF
               C     MATRIX A, WHICH WHICH HAS AN ACTUAL FIRST DIMENSION OF IA.
               C     X AND B ARE DOUBLE PRECISION MATRICES NEEDED FOR WORKING
               C     SPACE-X MUST BE A DOUBLY DIMENSIONED MATRIX WITH FIRST
               C     DIMENSION IX, B IS SINGLY DIMENSIONED AND SHOULD BE OF
               C     LENGTH AT LEAST 2*N1.  INT IS AN INTEGER VARIABLE WHICH
               C     IS RETURNED EQUAL TO TWO IF THE MATRIX IS TOO ILL
               C     CONDITIONED TO BE INVERTED
               C     MODIFIED JORDAN ELIMINATION
0002                 DOUBLE PRECISION B(N2),X(IX,N1),PIVOT,TEMP,DABS
0003                 DIMENSION A(IA,N1)
0004                 INT=1
0005                 N=N1
0006                 DO 15 I=1,N
0007                 DO 15 J=1,N
0008             15  X(I,J)=A(I,J)
0009                 DO 9 K=1,N
               C     FIND THE PIVOT
0010                 PIVOT=0.
0011                 DO 1 I=K,N
0012                 DO 1 J=K,N
0013                 IF(DABS(X(I,J)).LE.DABS(PIVOT)) GO TO 1
0014                 PIVOT=X(I,J)
0015                 A(1,K)=I
0016                 A(2,K)=J
0017              1  CONTINUE
0018                 IF(K.EQ.1) COMP=DABS(PIVOT)
0019                 IF((K.EQ.1.AND.COMP.LE.1.E-30).OR.
                    1 DABS(PIVOT).LE.1.5E-09*COMP) GO TO 14
               C     EXCHANGE ROWS
0020                 L=A(1,K)+1.E-6
0021                 IF(L.EQ.K) GO TO 3
0022                 DO 2 J=1,N
0023                 TEMP=X(L,J)
0024                 X(L,J)=X(K,J)
0025              2  X(K,J)=TEMP
               C     EXCHANGE COLUMNS
0026              3  L=A(2,K)+1.E-6
0027                 IF(L.EQ.K) GO TO 5
0028                 DO 4 I=1,N
0029                 TEMP=X(I,L)
0030                 X(I,L)=X(I,K)
0031              4  X(I,K)=TEMP
               C     JORDAN STEP
0032              5  DO 8 J=1,N
0033                 J2=N+J
0034                 B(J)=1.D0/PIVOT
0035                 IF(J.NE.K) GO TO 6
0036                 B(J2)=1.D0
0037                 GO TO 7
0038              6  B(J)=-X(K,J)*B(J)
0039                 B(J2)=X(J,K)
0040              7  X(K,J)=0.
0041              8  X(J,K)=0.
0042                 DO 9 I=1,N
```

```
              I2=N+I
              DO 9  J=1,N
        9     X(I,J)=X(I,J)+B(I2)*B(J)
C             REORDER FINAL MATRIX
              DO 13 L=1,N
              K=N+1-L
              J=A(1,K)+1.E-6
              IF(J.EQ.K) GO TO 11
              DO 10 I=1,N
              TEMP=X(I,J)
              X(I,J)=X(I,K)
       10     X(I,K)=TEMP
       11     I=A(2,K)+1.E-6
              IF(I.EQ.K) GO TO 13
              DO 12 J=1,N
              TEMP=X(I,J)
              X(I,J)=X(K,J)
       12     X(K,J)=TEMP
       13     CONTINUE
              DO 25 I=1,N
              DO 25 J=1,N
       25     A(I,J)=X(I,J)
              RETURN
       14     INT=2
              RETURN
              END
```

```
LEVEL 21.6 ( MAY 72 )                                    OS/360  FORTRAN H

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C       PROGRAM TO COMPUTE AERODYNAMIC ADMITTANCE FUNCTIONS
                C       FOR PLATES WITH CLAMPED EDGES
                C       EM IS MACH NUMBER
                C       DO NOT USE EM = 1.0
                C       AB IS PLATE LENGTH/WIDTH RATIO
                C       MMAX IS THE NUMBER OF MODES USED IN THE EXPRESSION
                C       FOR THE PLATE DEFLECTION
                C       IMAX IS THE NUMBER OF POINTS AT WHICH EACH
                C       ADMITTANCE FUNCTION IS TO BE COMPUTED
                C       THE AEH'S AND AEI'S ARE THE ADMITTANCE FUNCTIONS
  ISN 0002              DIMENSION AEH(12,12,100),AEI(12,12,100)
  ISN 0003              DIMENSION FERM(5000)
  ISN 0004              READ(5,11) EM,AB,MMAX,IMAX
  ISN 0005              WRITE(6,11) EM,AB,MMAX,IMAX
  ISN 0006        11    FORMAT (2F10.4, 2I10)
  ISN 0007             DO 1011 I= 1,MMAX
  ISN 0008             DO 1011 J= 1,MMAX
  ISN 0009             DO 1011 K= 1,100
  ISN 0010             AEH(I,J,K)= 0.
  ISN 0011         1011 AEI(I,J,K)= 0.
  ISN 0012              CIMAX=IMAX
  ISN 0013              CMMAX=MMAX
  ISN 0014              PI = 3.14159
  ISN 0015              SIGF = EM/(EM-1.)
  ISN 0016              IF(EM.GT.1.) GO TO 15
  ISN 0018              EMP = EM**2/(1.-EM**2)
  ISN 0019              ABP = (AB**2+1.)/AB**2
  ISN 0020              SIGF = EMP + SQRT(EMP**2+EMP*ABP)
  ISN 0021        15    CONTINUE
  ISN 0022              DELSIG = SIGF/CIMAX
  ISN 0023              WRITE (6,17) SIGF, DELSIG
  ISN 0024        17    FORMAT(2E20.4)
  ISN 0025              GAMMAX=3.*PI
  ISN 0026              ALPMAX=SQRT((PI*(CMMAX+1.))**2+100.)+5.
  ISN 0027              DO 24 I=1,IMAX
  ISN 0028              CI=I
  ISN 0029              S=CI*DELSIG
  ISN 0030              DELGAM=PI/4.
  ISN 0031              DELALP=PI/(4.*(1.+.2*S*(EM+1.)/EM))
  ISN 0032              DEL = -DELGAM*DELALP/(PI*EM)**2
  ISN 0033              NGAM=GAMMAX/DELGAM
  ISN 0034              NALP=ALPMAX/DELALP
  ISN 0035              WRITE(6,99) NALP,NGAM
  ISN 0036         99   FORMAT(2I20)
  ISN 0037             X= DELALP/2.
  ISN 0038              DO 22 L=1,NALP
  ISN 0039              GERM=0.
  ISN 0040              GAM=.01
  ISN 0041              DO 23 K=1,NGAM
  ISN 0042              SQ=SQRT (X**2+GAM**2*AB**2)
  ISN 0043              Z=SQ*S/EM
  ISN 0044              CALL GMR(GAM,1,1,GR,GI)
  ISN 0045         45   C=GR
  ISN 0046              GERM=GERM+C*SQ*BJ1(Z)
```

```
ISN 0047      23        GAM=GAM+DELGAM
ISN 0048                FERM(L)=GERM
ISN 0049      22        X=X+DELALP
ISN 0050                DO 21 M=1,MMAX
ISN 0051                DO 21 MR=M,MMAX
ISN 0052               ALPH= DELALP/2.
ISN 0053                TERMH=0.
ISN 0054                TERMI=0.
ISN 0055                DO 10 J=1,NALP
ISN 0056                CALL GMR(ALPH,M,MR,GR,GI)
ISN 0057                TERMH=TERMH+(GR*SIN (ALPH*S)-GI*COS (ALPH*S))
                     1  *FERM(J)*ALPH
ISN 0058                TERMI=TERMI+(GR*COS (ALPH*S)+GI*SIN (ALPH*S))*FERM(J)
ISN 0059      10        ALPH=ALPH+DELALP
ISN 0060                AEH(M,MR,I)=TERMH*DEL
ISN 0061                AEI(M,MR,I)=TERMI*DEL
ISN 0062                WRITE(6,12) AEH(M,MR,I),AEI(M,MR,I),M,MR,I
ISN 0063     12           FORMAT (2E20.3,3I10)
ISN 0064      21        CONTINUE
ISN 0065      24        CONTINUE
ISN 0066     199        FORMAT(6E20.3)
            C           WRITE ADMITTANCE FUNCTIONS ONTO TAPE
ISN 0067               WRITE (10) AEH,AEI,EM,AB,SIGF,IMAX
ISN 0068                ENDFILE 10
ISN 0069               STOP
ISN 0070                END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
ISN 0002              SUBROUTINE GMR(X,M,N,GR,GI)
            C         CLAMPED PLATE
ISN 0003              XX = X
ISN 0004              PI = 3.14159
ISN 0005              AM = M
ISN 0006              AN = N
ISN 0007              A = PI*(AM-1.)
ISN 0008              B = PI*(AM+1.)
ISN 0009              C = PI*(AN-1.)
ISN 0010              D = PI*(AN+1.)
ISN 0011    14        CONTINUE
ISN 0012              DENOM = (X**2-A**2)*(X**2-B**2)*(X**2-C**2)
                    1 *(X**2-D**2)
ISN 0013              IF(ABS(DENOM).LT.1.0E-10) GO TO 12
ISN 0015              GR = AMP*((1.+(-1.)**(M+N))+((-1.)**M+(-1.)**N)
                    1 *COS(X))
ISN 0016              GI = AMP*SIN (X)*((-1.)**N - (-1.)**M)
ISN 0017              X = XX
ISN 0018             RETURN
ISN 0019    12        CONTINUE
ISN 0020              X = X+.01
ISN 0021              GO TO 14
ISN 0022              END
```

```
         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                            SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002                 FUNCTION BJ1(X)
              C           POLYNOMIAL APPROXIMATION FOR BESSEL FUNCTION
 ISN 0003                 IF (X-3.) 1,2,2
 ISN 0004          1      Y=(X/3.)**2
 ISN 0005                 BJ1=X*(.5-.56249985*Y+.21093573*Y**2-.03954289*Y**3+
                       1 .00443319*Y**4-.00031761*Y**5+.00001109*Y**6)
 ISN 0006                 GO TO 3
 ISN 0007          2      Y=3./X
 ISN 0008                 F1=.79788456+.00000156*Y+.01659667*Y**2+.00017105*Y**3-
                       1 .00249511*Y**4+.00113653*Y**5-.00020033*Y**6
 ISN 0009                 TH1=X-2.35619449+.12499612*Y+.00005656*Y**2-.00637879*Y**3+
                       1 .00074348*Y**4+.00079824*Y**5-.00029166*Y**6
 ISN 0010                 BJ1=F1*COS(TH1)/SQRT(X)
 ISN 0011          3      CONTINUE
 ISN 0012                 RETURN
 ISN 0013                 END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
             C      FLUTTER PROGRAM FOR CLAMPED-EDGE PLATES USING
             C      LINEARIZED POTENTIAL FLOW AERODYNAMICS
             C      LAMDA=DYNAMIC PRESSURE,MU=FLOW DENSITY,MACH=MACH NUMBER
             C      AB=PLATE LENGTH/WIDTH RATIO,RXA,RYA=APPLIED IN-PLANE
             C      LOAD (POSITIVE IN TENSION),PSTAT=STATIC PRESSURE
             C      DIFFERENTIAL, CAVITY=CAVITY ACOUSTIC PARAMETER
             C      DAMP=STRUCTURAL DAMPING FACTOR
             C      NV=#OF MODES,H=INTEGRATION STEP INTERVAL,TPRINT=PRINT-OUT
             C      INTERVAL,TFINAL=TIME AT WHICH INTEGRATION STOPS
             C      SCALE=MAXIMUM ANTICIPATED DEFLECTION (FOR GRAPH ROUTINE)
             C      X, ,. = ALPHANUMERIC CHARACTERS FOR GRAPH ROUTINE
             C      THE A'S ARE MODAL AMPLITUDES
             C      THE W'S ARE THE PANEL DEFLECTION AT 15 EVENLY SPACED POINTS
             C      ALONG THE PANEL CENTERLINE
             C
             C
             C      REMOVE CARDS #177 THROUGH 189 FOR ZERO EDGE RESTRAINT
             C      CALCULATION
             C
             C
 ISN 0002           REAL LAMDA, MU
 ISN 0003           REAL NU
 ISN 0004           REAL MACH
 ISN 0005           DIMENSION B(12,12,12,12)
 ISN 0006           DIMENSION AEH(12,12,100),AEI(12,12,100)
 ISN 0007           DIMENSION AS(500,12),DAS(500,12)
 ISN 0008           DIMENSION S(12,12),C(12,12),D(12,12),PHIX(12,12),PHIY(12,12)
 ISN 0009           DIMENSION A(12),DA(12),DDA(12),DDAS(4,12)
 ISN 0010             DIMENSION Q(12,12)
 ISN 0011           DIMENSION W(15), F(12)
 ISN 0012           DIMENSION ST(12,12)
 ISN 0013           DIMENSION WM(12,12)
 ISN 0014           DIMENSION WV(24)
 ISN 0015           DIMENSION RINE(61)
 ISN 0016           REAL*8 WM
 ISN 0017           REAL*8 WV
 ISN 0018           PP(I,M) = CC(I-1,M-1)-CC(I-1,M+1)-CC(I+1,M-1)+CC(I+1,M+1)
 ISN 0019           PPX(I,J) = -PI*FLOAT(J-1)*(CS(I-1,J-1)-CS(I+1,J-1))
                  1 + PI*FLOAT(J+1)*(CS(I-1,J+1)-CS(I+1,J+1))
 ISN 0020           PPXX(I,M) = -PI2*FLOAT(M-1)**2*(CC(I-1,M-1)-CC(I+1,M-1))
                  1 + PI2*FLOAT(M+1)**2*(CC(I-1,M+1)-CC(I+1,M+1))
 ISN 0021           PPXXXX(I,M) = PI4*FLOAT(M-1)**4*(CC(I-1,M-1)-CC(I+1,M-1))
                  1 - PI4*FLOAT(M+1)**4*(CC(I-1,M+1)-CC(I+1,M+1))
 ISN 0022      1    FORMAT (1H0)
 ISN 0023      2    FORMAT (1H1)
 ISN 0024           PI = 3.14159
 ISN 0025           PI2 = PI*PI
 ISN 0026           PI3 = PI2*PI
 ISN 0027           PI4 = PI3*PI
 ISN 0028           NU = .3
 ISN 0029           READ (5,701) CROSS, BLANK, DOT, SCALE
 ISN 0030    701    FORMAT (3A1, F7.1)
 ISN 0031           WRITE (6,700) CROSS, BLANK, DOT, SCALE
 ISN 0032    700    FORMAT (1X, 3A1, F7.2)
```

```
ISN 0033              SCALE = 30./SCALE
ISN 0034              DO 702 I = 1,61
ISN 0035     702      RINE(I) = BLANK
             C        READ NONLINEAR TERMS FROM TAPE
ISN 0036              READ(10) B
ISN 0037             REWIND 10
             C        READ AERODYNAMIC ADMITTANCE FUNCTIONS FROM TAPE
ISN 0038              READ(12) AEH,AEI,MACH,AB,SIGF,ISMAX
ISN 0039             REWIND 12
ISN 0040              WRITE (6,1)
ISN 0041              WRITE(6,13) MACH,AB,NV,ISMAX,SIGF
ISN 0042     13       FORMAT(2F10.4,2I10,E10.3)
ISN 0043     5        FORMAT (1P8E9.2)
ISN 0044     8        FORMAT (10E12.3)
ISN 0045              WRITE (6,1)
ISN 0046              READ (5,110) LAMDA, MU, PSTAT, CAVITY
ISN 0047     110      FORMAT(4E10.3)
ISN 0048              WRITE(6,1101) LAMDA,MU,PSTAT,CAVITY
ISN 0049     1101     FORMAT(1H0,1P4E10.3)
ISN 0050              READ (5,110) RXA,RYA
ISN 0051              WRITE(6,1101) RXA,RYA
ISN 0052              WRITE (6,1)
ISN 0053             READ (5,116) DAMP
ISN 0054     116      FORMAT(E10.3)
ISN 0055             WRITE (6,117) DAMP
ISN 0056     117      FORMAT(' STRUCTURAL DAMPING= ',E12.3)
ISN 0057              WRITE (6,1)
ISN 0058             READ  (5,1113) NV,H,TPRINT,TFINAL
ISN 0059             WRITE (6,1113) NV,H,TPRINT,TFINAL
ISN 0060      1113 FORMAT (I10,3E10.3)
ISN 0061              WRITE (6,1)
ISN 0062              READ (5,113) (A(I), I = 1,NV)
ISN 0063              WRITE (6,115) (A(I), I = 1,NV)
ISN 0064              READ (5,113) (DA(I), I = 1,NV)
ISN 0065              WRITE (6,115) (DA(I), I = 1,NV)
ISN 0066     113      FORMAT(6E10.3)
ISN 0067     115      FORMAT(1X,6E12.3)
ISN 0068              AB2 = AB**2
ISN 0069              AB4 = AB**4
ISN 0070              RISMAX = ISMAX
ISN 0071              DELS = SIGF/RISMAX
ISN 0072              ROOT = SQRT(LAMDA/MU)
ISN 0073              HAERO = ROOT*H
ISN 0074              NAERO = DELS/HAERO
ISN 0075              IF(NAERO.LT.1) GO TO 404
ISN 0077              IF(NAERO.GT.20) NAERO = 20
ISN 0079              DELSIG = NAERO*HAERO
ISN 0080              IMAX = SIGF/DELSIG
ISN 0081              GO TO 406
ISN 0082     404      CONTINUE
ISN 0083              NAERO = 1
ISN 0084              DELSIG = DELS
ISN 0085              H = DELSIG/ROOT
ISN 0086              IMAX = ISMAX
ISN 0087     406      CONTINUE
ISN 0088              HP = DELSIG/ROOT
```

```
ISN 0089              WRITE (6,1)
ISN 0090              WRITE(6,401) H,DELSIG,NAERO,IMAX
ISN 0091      401     FORMAT(2E20.3,2I5)
ISN 0092              IF(IMAX.GT.100) STOP
ISN 0094              NSTORE = NAERO*IMAX
ISN 0095             DO 3 I= 1,NV
ISN 0096             DO 3 J= 1,NV
ISN 0097             DO 3 K= 1,ISMAX
ISN 0098             AEH(I,J,100-K+1)= AEH(I,J,ISMAX-K+1)
ISN 0099             AEI(I,J,100-K+1)= AEI(I,J,ISMAX-K+1)
ISN 0100           3 CONTINUE
ISN 0101              IP = 100-ISMAX+1
ISN 0102              DO 400 M = 1,NV
ISN 0103              DO 400 N = M,NV
ISN 0104              DO 400 I = 1,IMAX
ISN 0105              X = FLOAT(I)*DELSIG/DELS
ISN 0106              J = INT(X)
ISN 0107              P=X-AINT(X)
ISN 0108              JP=IP-1+J
ISN 0109              IF (J) 300,300,301
ISN 0110      300     AEH(M,N,I)=AEH(M,N,JP+1)*P
ISN 0111              AEI(M,N,I)=AEI(M,N,JP+1)*P
ISN 0112              GO TO 400
ISN 0113      301     AEH(M,N,I)  = AEH(M,N,JP)*(1.-P)  + AEH(M,N,JP+1)*P
ISN 0114              AEI(M,N,I)  = AEI(M,N,JP)*(1.-P)  + AEI(M,N,JP+1)*P
ISN 0115      400     CONTINUE
ISN 0116              DO 410 M = 1,NV
ISN 0117              DO 410 N = M,NV
ISN 0118              DO 410 I = 1,IMAX
ISN 0119              AEH(N,M,I)  =  (-1.)**(M+N)*AEH(M,N,I)
ISN 0120              AEI(N,M,I)  =  (-1.)**(M+N)*AEI(M,N,I)
ISN 0121      410     CONTINUE
ISN 0122              DO 30 I = 1,NV
ISN 0123              DO 30 J = 1,NV
ISN 0124              S(I,J)  = PP(I,J)*PP(1,1)
ISN 0125              ST(I,J)  = S(I,J)
ISN 0126              C(I,J)  = PPXXXX(I,J)*PP(1,1)  + 2.*AB2*PPXX(I,J)*PPXX(1,1)
                    1 + AB4*PP(I,J)*PPXXXX(1,1)
ISN 0127              D(I,J)  = PPX(I,J)*PP(1,1)
ISN 0128              PHIX(I,J)  =-PPXX(I,J)*PP(1,1)
ISN 0129              PHIY(I,J)  =-PP(I,J)*PPXX(1,1)
ISN 0130      30      CONTINUE
ISN 0131              WRITE(6,1)
ISN 0132              DO 910 I = 1,NV
ISN 0133      910     WRITE(6,573)  (    S(I,J),  J = 1,NV)
ISN 0134              WRITE  (6,1)
ISN 0135              DO 580 I = 1,NV
ISN 0136      580     WRITE (6,573) (C(I,J), J = 1,NV)
ISN 0137      573     FORMAT (8E16.4)
              C       INVERT S AND STORE IN S
ISN 0138              CALL MATIN2(S,NV,12,WM,12,WV,KINVRT,24)
ISN 0139              WRITE(6,1)
ISN 0140              WRITE(6,5067) KINVRT
ISN 0141      5067    FORMAT(I5)
ISN 0142              WRITE(6,1)
ISN 0143              DO 912 I = 1,NV
```

```
ISN 0144    912    WRITE(6,573) (   D(I,J), J = 1,NV)
ISN 0145           WRITE(6,1)
ISN 0146           DO 914 I = 1,NV
ISN 0147    914    WRITE(6,573) (PHIX(I,J), J = 1,NV)
ISN 0148           WRITE(6,1)
ISN 0149           DO 916 I = 1,NV
ISN 0150    916    WRITE(6,573) (PHIY(I,J), J = 1,NV)
ISN 0151           WRITE (6,1)
            C      SET UP INTEGRATION
ISN 0152           T = 0.
ISN 0153           TP = 0.
ISN 0154           DO 70 I = 1,500
ISN 0155           DO 70 J = 1,NV
ISN 0156           AS(I,J) = 0.
ISN 0157    70     DAS(I,J) = 0.
ISN 0158           DO 74 I = 1,4
ISN 0159           DO 74 J = 1,NV
ISN 0160    74     DDAS(I,J) = 0.
ISN 0161           IMAXP = IMAX - 1
ISN 0162           HHH = H/24.
ISN 0163           WRITE (6,2)
            C      PREDICTOR ROUTINE
ISN 0164    56     CONTINUE
ISN 0165           IF(A(1).GT.10.) STOP
            C      COMPUTE SECOND DIRIVATIVES
ISN 0167           DO 490 I = 1,NV
ISN 0168           DO 490 J = 1,NV
ISN 0169           Q(I,J) = (D(I,J)*A(J) + ST(I,J)*DA(J)/ROOT)/MACH
ISN 0170           DO 488 K = 1,IMAXP
ISN 0171           KP = NAERO*K
ISN 0172           Q(I,J) = Q(I,J)+DAS(KP,J)*AEI(J,I,K)*HP
                 1 +AS(KP,J)*AEH(J,I,K)*DELSIG
ISN 0173    488    CONTINUE
ISN 0174    490    CONTINUE
ISN 0175           SX = 0.
ISN 0176           SY = 0.
            C      REMOVE THE FOLLOWING 12 CARDS FOR A ZERO EDGE RESTRAINT
            C      CALCULATION
ISN 0177           SY = .5*A(1)**2
ISN 0178           DO 212 M = 1,NV
ISN 0179           RM = M
ISN 0180           SX = SX + (RM**2+1.)*A(M)**2
ISN 0181    212    SY = SY + A(M)**2
ISN 0182           NVP = NV-2
ISN 0183           IF(NVP.LT.1) GO TO 215
ISN 0185           DO 214 M = 1,NVP
ISN 0186           RM = M
ISN 0187           SX = SX - (RM+1.)**2*A(M)*A(M+2)
ISN 0188    214    SY = SY - A(M)*A(M+2)
ISN 0189    215    CONTINUE
ISN 0190           AB2RXB = 12.*PI2*(.75*SX + NU*AB2*SY) + RXA
ISN 0191           RYB = 12.*PI2*(AB2*SY + NU*.75*SX) + RYA
ISN 0192           DO 250 I = 1,NV
ISN 0193           F(I) = 0.
ISN 0194           DO 216 M = 1,NV
ISN 0195    216    F(I) = F(I) - AB2RXB*PHIX(I,M)*A(M) - AB2*RYB*PHIY(I,M)*A(M)
```

```
ISN 0196              DO 220 J = 1,NV
ISN 0197              DO 220 K = 1,NV
ISN 0198              LSUM = I+J+K
ISN 0199              LS = 2*(LSUM/2)-LSUM+2
ISN 0200              DO 220 L = LS,NV,2
ISN 0201     220      F(I) = F(I) - B(I,J,K,L)*A(J)*A(K)*A(L)
ISN 0202              DO 230 J = 1,NV
ISN 0203     230      F(I) = F(I) - C(I,J)*A(J) - LAMDA*Q(I,J)-DAMP*C(I,J)*DA(J)
ISN 0204     250      CONTINUE
ISN 0205              F(1) = F(1) - CAVITY*A(1) + PSTAT
ISN 0206              DO 240 I = 1,NV
ISN 0207              DDA(I) = 0.
ISN 0208              DO 240 J = 1,NV
ISN 0209              DDA(I) = DDA(I) + S(I,J)*F(J)
ISN 0210     240      CONTINUE
             C        PRINT OUTPUT
ISN 0211              IF(T.LT.TP) GO TO 350
ISN 0213              TP = TP + TPRINT
             C        PRINT MODAL AMPLITUDES, VELOCITIES, AND ACCELERATIONS
ISN 0214              WRITE (6,345) T
ISN 0215     345      FORMAT(6H TIME=,F7.4)
ISN 0216              WRITE (6,347) (A(I), I = 1,NV)
ISN 0217              WRITE (6,347) (DA(I), I = 1,NV)
ISN 0218              WRITE(6,347) (DDA(I), I = 1,NV)
ISN 0219     347      FORMAT(6E11.3)
ISN 0220              DO 348 I = 1,15
ISN 0221              W(I) = 0.
ISN 0222              THETA = PI*FLOAT(I)/16.
ISN 0223              DO 346 J = 1,NV
ISN 0224     346      W(I) = W(I) + 2.*A(J)*(COS(FLOAT(J-1)*THETA) - COS(FLOAT(J+1)
                    1 *THETA))
ISN 0225     348      CONTINUE
             C        PRINT PLATE DEFLECTION AT 15 EQUALLY SPACED POINTS ALONG THE
             C        CENTERLINE OF THE PANEL
ISN 0226              WRITE (6,349) (W(I), I = 1,15)
ISN 0227     349      FORMAT (8F7.2)
ISN 0228              RINE(1) = DOT
ISN 0229              RINE(31) = DOT
ISN 0230              RINE(61) = DOT
ISN 0231              L = SCALE*W(12)
ISN 0232              LP = 31+L
ISN 0233              IF(IABS(L).LE.30) RINE(LP) = CROSS
             C        GRAPH DEFLECTION OF POINT ON LATERAL CENTERLINE OF PANEL
             C        3/4 OF WAY BACK FROM LEADING EDGE
ISN 0235              WRITE (6,703) (RINE(I), I = 1,61)
ISN 0236     703      FORMAT (68X, 61A1)
ISN 0237              IF(IABS(L).LE.30) RINE(LP) = BLANK
ISN 0239              IF(T.GE.TFINAL) GO TO 57
ISN 0241     350      CONTINUE
             C        STORE VARIABLES
ISN 0242              DO 24 J = 2,NSTORE
ISN 0243              K = NSTORE-J+2
ISN 0244              KP = K-1
ISN 0245              DO 24 I = 1,NV
ISN 0246              AS(K,I) = AS(KP,I)
ISN 0247              DAS(K,I) = DAS(KP,I)
```

```
ISN 0248      24     CONTINUE
ISN 0249             DO 26 J = 2,4
ISN 0250             K = 6-J
ISN 0251             KP = K-1
ISN 0252             DO 26 I = 1,NV
ISN 0253             DDAS(K,I) = DDAS(KP,I)
ISN 0254      26     CONTINUE
ISN 0255             DO 28 I = 1,NV
ISN 0256             AS(1,I) = A(I)
ISN 0257             DAS(1,I) = DA(I)
ISN 0258             DDAS(1,I) = DDA(I)
ISN 0259      28     CONTINUE
              C      PREDICT
ISN 0260             DO 20 I = 1,NV
ISN 0261             A(I) = A(I)+HHH*(55.*DAS(1,I)-59.*DAS(2,I)
                   1 +37.*DAS(3,I)-9.*DAS(4,I))
ISN 0262             DA(I) = DA(I)+HHH*(55.*DDAS(1,I)-59.*DDAS(2,I)
                   1 +37.*DDAS(3,I)-9.*DDAS(4,I))
ISN 0263      20     CONTINUE
ISN 0264             T = T+H
ISN 0265             GO TO 56
ISN 0266      57     CONTINUE
ISN 0267             STOP
ISN 0268             END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002                  SUBROUTINE MATIN2(A,N1,IA,X,IX,B,INT,N2)
                  C         THIS SUBROUTINE INVERTS THE UPPER LEFT N1 BY N1 CORNER OF
                  C         MATRIX A, WHICH WHICH HAS AN ACTUAL FIRST DIMENSION OF IA.
                  C         X AND B ARE DOUBLE PRECISION MATRICES NEEDED FOR WORKING
                  C         SPACE-X MUST BE A DOUBLY DIMENSIONED MATRIX WITH FIRST
                  C         DIMENSION IX, B IS SINGLY DIMENSIONED AND SHOULD BE OF
                  C         LENGTH AT LEAST 2*N1.  INT IS AN INTEGER VARIABLE WHICH
                  C         IS RETURNED EQUAL TO TWO IF THE MATRIX IS TOO ILL
                  C         CONDITIONED TO BE INVERTED
                  C         MODIFIED JORDAN ELIMINATION
  ISN 0003                  DOUBLE PRECISION B(N2),X(IX,N1),PIVOT,TEMP,DABS
  ISN 0004                  DIMENSION A(IA,N1)
  ISN 0005                  INT=1
  ISN 0006                  N=N1
  ISN 0007                  DO 15 I=1,N
  ISN 0008                  DO 15 J=1,N
  ISN 0009           15     X(I,J)=A(I,J)
  ISN 0010                  DO 9 K=1,N
                  C         FIND THE PIVOT
  ISN 0011                  PIVOT=0.
  ISN 0012                  DO 1 I=K,N
  ISN 0013                  DO 1 J=K,N
  ISN 0014                  IF(DABS(X(I,J)).LE.DABS(PIVOT)) GO TO 1
  ISN 0016                  PIVOT=X(I,J)
  ISN 0017                  A(1,K)=I
  ISN 0018                  A(2,K)=J
  ISN 0019            1     CONTINUE
  ISN 0020                  IF(K.EQ.1) COMP=DABS(PIVOT)
  ISN 0022                  IF((K.EQ.1.AND.COMP.LE.1.E-30).OR.
                           1 DABS(PIVOT).LE.1.5E-09*COMP) GO TO 14
                  C         EXCHANGE ROWS
  ISN 0024                  L=A(1,K)+1.E-6
  ISN 0025                  IF(L.EQ.K) GO TO 3
  ISN 0027                  DO 2 J=1,N
  ISN 0028                  TEMP=X(L,J)
  ISN 0029                  X(L,J)=X(K,J)
  ISN 0030            2     X(K,J)=TEMP
                  C         EXCHANGE COLUMNS
  ISN 0031            3     L=A(2,K)+1.E-6
  ISN 0032                  IF(L.EQ.K) GO TO 5
  ISN 0034                  DO 4 I=1,N
  ISN 0035                  TEMP=X(I,L)
  ISN 0036                  X(I,L)=X(I,K)
  ISN 0037            4     X(I,K)=TEMP
                  C         JORDAN STEP
  ISN 0038            5     DO 8 J=1,N
  ISN 0039                  J2=N+J
  ISN 0040                  B(J)=1.D0/PIVOT
  ISN 0041                  IF(J.NE.K) GO TO 6
  ISN 0043                  B(J2)=1.D0
  ISN 0044                  GO TO 7
  ISN 0045            6     B(J)=-X(K,J)*B(J)
  ISN 0046                  B(J2)=X(J,K)
  ISN 0047            7     X(K,J)=0.
```

```
ISN 0048        8     X(J,K)=0.
ISN 0049              DO 9 I=1,N
ISN 0050              I2=N+I
ISN 0051              DO 9  J=1,N
ISN 0052        9     X(I,J)=X(I,J)+B(I2)*B(J)
               C      REORDER FINAL MATRIX
ISN 0053              DO 13 L=1,N
ISN 0054              K=N+1-L
ISN 0055              J=A(1,K)+1.E-6
ISN 0056              IF(J.EQ.K) GO TO 11
ISN 0058              DO 10 I=1,N
ISN 0059              TEMP=X(I,J)
ISN 0060              X(I,J)=X(I,K)
ISN 0061       10     X(I,K)=TEMP
ISN 0062       11     I=A(2,K)+1.E-6
ISN 0063              IF(I.EQ.K) GO TO 13
ISN 0065              DO 12 J=1,N
ISN 0066              TEMP=X(I,J)
ISN 0067              X(I,J)=X(K,J)
ISN 0068       12     X(K,J)=TEMP
ISN 0069       13     CONTINUE
ISN 0070              DO 25 I=1,N
ISN 0071              DO 25 J=1,N
ISN 0072       25     A(I,J)=X(I,J)
ISN 0073              RETURN
ISN 0074       14     INT=2
ISN 0075              RETURN
ISN 0076               END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              FUNCTION CS(M,N)
  ISN 0003              CS = 0.
  ISN 0004              PI = 3.14159
  ISN 0005              RM = M
  ISN 0006              RN = N
  ISN 0007              IF(MOD(M+N,2).NE.0) CS = -2.*RN/(PI*(RM**2-RN**2))
  ISN 0009              RETURN
  ISN 0010              END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              FUNCTION CC(K,M)
  ISN 0003              CC = 0.
  ISN 0004              IF(K.EQ.M) CC = CC+.5
  ISN 0006              IF(K.EQ.-M) CC = CC+.5
  ISN 0008              RETURN
  ISN 0009              END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                              SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
              C         PROGRAM TO COMPUTE STRESSES DUE TO PLATE DEFLECTION FOR
              C         CLAMPED PLATE WITH COMPLETE IN-PLANE EDGE RESTRAINT
              C         AB=PLATE LENGTH/WIDTH RATIO,NV=# OF MODES,NU=POISSON'S RATIO
              C         XA,YB=COORDINATES OF POINT AT STRESSES ARE COMPUTED
              C         (XA,YB ARE NONDIMENSIONALIZED BY PANEL LENGTH & WIDTH)
              C         XA=N/16,N=0,1,2,...17
              C         YB IS SPECIFIED AS INPUT DATA
              C         THE A'S ARE THE MODAL AMPLITUDES (WHICH DEFINE THE PLATE
              C         DEFLECTION)
  ISN 0002              REAL NU
  ISN 0003              DIMENSION A(12)
  ISN 0004              DENOM1(K) =  (FLOAT(K**2)  +  16.*AB2)**2
  ISN 0005              DENOM2(K) =  (FLOAT(K**2)  +  4.*AB2)**2
  ISN 0006              B1(M,N) = -FLOAT(2*M*(M+N)  +  4)/DENOM1(M-N)
  ISN 0007              B2(M,N) = FLOAT((M-1)*(M+N))/DENOM1(M-N-2)
  ISN 0008              B3(M,N) = FLOAT((M+1)*(M+N))/DENOM1(M-N+2)
  ISN 0009              B4(M,N) = FLOAT(2*M*(M-N)  +  4)/DENOM1(M+N)
  ISN 0010              B5(M,N) = -FLOAT((M-1)*(M-N))/DENOM1(M+N-2)
  ISN 0011              B6(M,N) = -FLOAT((M+1)*(M-N))/DENOM1(M+N+2)
  ISN 0012              B7(M,N) = FLOAT(4*M**2  +  4)/DENOM2(M-N)
  ISN 0013              B8(M,N) = -FLOAT(2*(M-1)**2)/DENOM2(M-N-2)
  ISN 0014              B9(M,N) = -FLOAT(2*(M+1)**2)/DENOM2(M-N+2)
  ISN 0015              B10(M,N)  = -FLOAT(4*M**2  +  4)  /DENOM2(M+N)
  ISN 0016              B11(M,N)  = FLOAT   (2*(M-1)**2)/DENOM2(M+N-2)
  ISN 0017              B12(M,N)  = FLOAT(2*(M+1)**2)/DENOM2(M+N+2)
  ISN 0018              CS(M,X) = COS(FLOAT(M)*PI*X)
  ISN 0019              READ (5,697) AB, NV
  ISN 0020              WRITE (6,697) AB, NV
  ISN 0021      697     FORMAT (F10.3, I10)
  ISN 0022              READ (5,697) YB
  ISN 0023              WRITE (6,697) YB
  ISN 0024              READ (5,699)  (A(I), I = 1,NV)
  ISN 0025              WRITE (6,699)  (A(I), I = 1,NV)
  ISN 0026      699     FORMAT (6F10.4)
  ISN 0027              PI = 3.14159
  ISN 0028              PI2 = PI**2
  ISN 0029              PI4 = PI**4
  ISN 0030              AB2 = AB**2
  ISN 0031              AB4 = AB**4
  ISN 0032              NU = .3
  ISN 0033              XA = 0.
  ISN 0034              CHI = 1. - CS(2,YB)
  ISN 0035              DO 710 II = 1,17
  ISN 0036              WRITE (6,1)
  ISN 0037      1       FORMAT (1H0)
  ISN 0038              WRITE (6,701) XA, YB
  ISN 0039      701     FORMAT (2E20.3)
  ISN 0040              W = 0.
  ISN 0041              WXX = 0.
  ISN 0042              WYY = 0.
  ISN 0043              DO 704 M = 1,NV
  ISN 0044              PSI = CS(M-1,XA)  - CS(M+1,XA)
  ISN 0045              RM = M
  ISN 0046              PSIXX = -(RM-1.)**2*PI2*CS(M-1,XA)  +  (RM+1.)**2*PI2*
```

```
                     1  CS(M+1,XA)
ISN 0047                W = W + A(M)*PSI
ISN 0048                WXX = WXX + PSIXX*A(M)
ISN 0049       704      WYY = WYY + PSI*A(M)
ISN 0050                W = W*CHI
ISN 0051                WXX = WXX*CHI
ISN 0052                WYY = 4.*PI2*CS(2,YB)*WYY
ISN 0053                WRITE (6,701) W
ISN 0054                WRITE (6,701) WXX, WYY
ISN 0055                SIGMAX = -.5*(WXX + NU*AB2*WYY)
ISN 0056                SIGMAY = -.5*(AB2*WYY + NU*WXX)
               C        PRINT PLATE BENDING STRESSES AT POINT(XA,YB)
ISN 0057                WRITE (6,701) SIGMAX, SIGMAY
ISN 0058                SX = 0.
ISN 0059                SY = .5*A(1)**2
ISN 0060                DO 706 M = 1,NV
ISN 0061                RM = M
ISN 0062                SX = SX + (RM**2+1.)*A(M)**2
ISN 0063       706      SY = SY + A(M)**2
ISN 0064                IF(NV.LE.2) GO TO 712
ISN 0066                NVP = NV-2
ISN 0067                DO 708 M = 1,NVP
ISN 0068                RM = M
ISN 0069                SX = SX - (RM+1.)**2*A(M)*A(M+2)
ISN 0070       708      SY = SY - A(M)*A(M+2)
ISN 0071       712      CONTINUE
ISN 0072                AB2RXB = 12.*PI2*(.75*SX + NU*AB2*SY)
ISN 0073                RYB = 12.*PI2*(AB2*SY + .75*NU*SX)
ISN 0074                PHIPXX = 0.
ISN 0075                PHIPYY = 0.
ISN 0076                DO 702 M = 1,NV
ISN 0077                DO 702 N = 1,NV
ISN 0078                X = B1(M,N)*CS(M-N,XA) + B2(M,N)*CS(M-N-2,XA)
                     1  + B3(M,N)*CS(M-N+2,XA) + B4(M,N)*CS(M+N,XA)
                     2  + B5(M,N)*CS(M+N-2,XA) + B6(M,N)*CS(M+N+2,XA)
ISN 0079                YMN = -16.*PI2*CS(4,YB)*X
ISN 0080                X = B7(M,N)*CS(M-N,XA) + B8(M,N)*CS(M-N-2,XA)
                     1  + B9(M,N)*CS(M-N+2,XA) + B10(M,N)*CS(M+N,XA)
                     2  + B11(M,N)*CS(M+N-2,XA) + B12(M,N)*CS(M+N+2,XA)
ISN 0081                YMN = YMN - 4.*PI2*CS(2,YB)*X
ISN 0082                RM = M
ISN 0083                RN = N
ISN 0084                X = (RM-RN)**2*B1(M,N)*CS(M-N,XA) + (RM-RN-2.)**2*B2(M,N)*
                     1  CS(M-N-2,XA) + (RM-RN+2.)**2*B3(M,N)*CS(M-N+2,XA)
                     2  + (RM+RN)**2*B4(M,N)*CS(M+N,XA) + (RM+RN-2.)**2*B5(M,N)
                     3  *CS(M+N-2,XA) + (RM+RN+2.)**2*B6(M,N)*CS(M+N+2,XA)
ISN 0085                XMN = -PI2*CS(4,YB)*X
ISN 0086                X = (RM-RN)**2*B7(M,N)*CS(M-N,XA) + (RM-RN-2.)**2*B8(M,N)*
                     1  CS(M-N-2,XA) + (RM-RN+2.)**2*B9(M,N)*CS(M-N+2,XA)
                     2  + (RM+RN)**2*B10(M,N)*CS(M+N,XA) + (RM+RN-2.)**2*B11(M,N)
                     3  *CS(M+N-2,XA) + (RM+RN+2.)**2*B12(M,N)*CS(M+N+2,XA)
ISN 0087                XMN = XMN - PI2*CS(2,YB)*X
ISN 0088                X = (RM-RN)**2*B13(M,N)*CS(M-N,XA) + (RM-RN-2.)**2*B14(M,N)*
                     1  CS(M-N-2,XA) + (RM-RN+2.)**2*B15(M,N)*CS(M-N+2,XA)
                     2  + (RM+RN)**2*B16(M,N)*CS(M+N,XA) + (RM+RN-2.)**2*B17(M,N)
                     3  *CS(M+N-2,XA) + (RM+RN+2.)**2*B18(M,N)*CS(M+N+2,XA)
```

```
ISN 0089              XMN = XMN - PI2*X
ISN 0090              PHIPXX = PHIPXX + XMN*A(M)*A(N)
ISN 0091      702     PHIPYY = PHIPYY + YMN*A(M)*A(N)
ISN 0092              PHIPXX = 12.*(1.-NU**2)*AB2*PHIPXX
ISN 0093              PHIPYY = 12.*(1.-NU**2)*AB2*PHIPYY
ISN 0094              SIGMAX = AB2RXB + AB2*PHIPYY
ISN 0095              SIGMAY = RYB + PHIPXX
ISN 0096              SIGMAX = SIGMAX/12.
ISN 0097              SIGMAY = SIGMAY/12.
              C       PRINT IN-PLANE STRESSES AT POINT(XA,YB)
ISN 0098              WRITE (6,701) SIGMAX, SIGMAY
ISN 0099              XA = XA + 1./16.
ISN 0100      710     CONTINUE
ISN 0101              RETURN
ISN 0102              END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              FUNCTION B13(M,N)
  ISN 0003              B13 = 0.
  ISN 0004              IF(M-N .NE. 0) B13 = -FLOAT(2*M)/FLOAT((M-N)**3)
  ISN 0006              RETURN
  ISN 0007             END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002             FUNCTION B14(M,N)
 ISN 0003             B14 = 0.
 ISN 0004             IF(M-N-2 .NE. 0) B14 = FLOAT (M-1)/FLOAT((M-N-2)**3)
 ISN 0006             RETURN
 ISN 0007             END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              FUNCTION B15(M,N)
  ISN 0003              B15 = 0.
  ISN 0004              IF(M-N+2 .NE. 0) B15 = FLOAT (M+1)/FLOAT((M-N+2)**3)
  ISN 0006              RETURN
  ISN 0007             END
```

```
       COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002         FUNCTION B16(M,N)
 ISN 0003         B16 = 0.
 ISN 0004         IF(M+N .NE. 0) B16 = FLOAT(2*M)/FLOAT((M+N)**3)
 ISN 0006         RETURN
 ISN 0007        END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
 ISN 0002             FUNCTION B17(M,N)
 ISN 0003             B17 = 0.
 ISN 0004             IF(M+N-2 .NE. 0) B17 = -FLOAT(M-1)/FLOAT((M+N-2)**3)
 ISN 0006             RETURN
 ISN 0007            END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002              FUNCTION B18(M,N)
  ISN 0003              B18 = 0.
  ISN 0004              IF(M+N+2 .NE. 0) B18 = -FLOAT(M+1)/FLOAT((M+N+2)**3)
  ISN 0006              RETURN
  ISN 0007              END
```