NASA CR-144967

CORRECTION FACTOR TECHNIQUES FOR IMPROVING AERODYNAMIC PREDICTION METHODS

By

J. P. Giesing

T. P. Kalman

W. P. Rodden

MAY 1976

Prepared under Contract No. NASI-13835 By

MCDONNELL DOUGLAS CORPORATION

DOUGLAS AIRCRAFT COMPANY

LONG BEACH, CALIFORNIA

For

National Aeronautics And Space Administration

CONTENTS

Page

CORRECTION FACTOR TECHNIQUES FOR IMPROVING AERODYNAMIC PREDICTION METHODS*

By Joseph P. Giesing, Terez P. Kalman
and William P. Rodden**
Douglas Aircraft Company

SUMMARY

This report describes a method for correcting lifting surface theory so that it reflects known experimental data. Specifically the theoretical pressure distribution is modified such that imposed constraints are satisfied (e.g., lift, moment, etc.) while minimizing the change to the theoretical pressure distribution. It is assumed that a finite element or discretized lifting surface method is used, such as either the Doublet or Vortex Lattice Methods.

There are several ways in which the theoretical pressures are modified. One is a direct application of a set of correction factors to the pressures. This is accomplished by premultiplying the pressures with a diagonal matrix of correction factors. A second approach to correcting the theory is to modify the downwash. This modification can be accomplished by either multiplying the downwash by a diagonal matrix of correction factors or by adding an incremental downwash (which is proportioned to the pressure) to the theoretical downwash. In any case the correction factors are adjusted so that the imposed experimental constraints are satisfied by the corrected pressure distribution while the changes in the pressure distribution are minimized.

There are several features that have been built into the basic method and these include: (1) the ability to consider together experimental data

---

* The authors wish to acknowledge Dr. Edward Albano for interesting discussions of alternate formats (non-diagonal) of correction matrices and their potential derivations.

** Consulting Engineer

Prepared under Contract No. NAS1-13835

from more than one mode (e.g., control surface rotation, pitch, camber, etc.), (2) the ability to limit the excursions of the correction factors (i.e., establish minimum and maximum values for them) and (3) the ability to use correction factor mode shapes (i.e., construct correction factors from known distributions or functions).

The methods developed have been implemented on the computer and many correlations and calculations made. Specifically cases involving all three Mach Number ranges are considered. For instance in the subsonic speed range a swept wing with an oscillating partial span flap and a swept wing with a leading edge droop are discussed. In the transonic speed range a two-dimensional symmetric airfoil with an oscillating flap is treated in detail. An arrow wing with and without camber is used in the supersonic analysis.

The computer program used to generate the correction factors for these cases is also fully described in this report and test cases are provided.

Finally, a new, simple method for accounting for transonic effects in the lifting surface theory is described and correlated for the two-dimensional case. Basically, a transformed distance between the sending element and receiving point is employed. The transformation depends on the time delay encountered by a signal traveling from the sending point to the receiving point.

INTRODUCTION

Wind tunnel data have provided the basis for semi-empirical methods of aeroelastic analysis for many years, whether in the estimation of stability and control characteristics, the calculation of structural loads, or in flutter analysis by modified strip methods. These semi-empirical methods have been tailored to aerodynamic lifting-line theory or to strip theory and not to the more general (and more accurate) lifting-surface methods. The use of a diagonal correction matrix to be applied as a premultiplying factor to matrices of aerodynamic influence coefficients obtained from lifting surface theory has been considered by a number of authors. A premultiplier may be regarded as a correction to the pressure distribution; as an alternative, a postmultiplier would be regarded as a correction to the downwash to account for thickness effects and for camber induced by boundary-layer displacement effects. Rodden and Revell (refs. 1 - 4) considered a real correction matrix derived from static wind tunnel measurements and theoretical load predictions. Bergh and Zwaan (refs. 5 and 6) investigated a complex correction matrix derived from oscillatory wind-tunnel pressure measurements and theoretical predictions. These authors assumed measurements were available only for a single mode, a steady angle of attack or an oscillatory pitching (or yawing) mode.

Current interest in using actively controlled aerodynamic surfaces to minimize aeroelastic response requires an improvement in accuracy in predicting unsteady aerodynamic characteristics of lifting surfaces equipped with control surfaces. The correction matrix provides one means of improving the accuracy but it requires experimental data on control surface characteristics in addition to the angle of attack characteristics of the surface. Hence, an extension of references 1 - 4 is necessary to obtain the correction matrix for more than one aerodynamic mode. Furthermore, the discrepancies between theory and experiment in predicting trailing-edge control surface loads are most likely caused by boundary-layer displacement effects on the effective downwash. Hence, another extension is necessary to obtain a postmultiplying correction matrix. These two extensions are considered in the present development. The diagonal format has been retained* and complex pre- and postmultiplying correction matrices have been derived which satisfy the constraints of matching experimental data from multiple aerodynamic downwash modes.

The use of correction matrices is in the time-honored engineering tradition of empirical correction factors. It retains the generality of the theory while approaching the limiting values of the test results. Such a posteriori adjustment obviously cannot be regarded as addressing any of the fundamental causes of the discrepancies. Other possibilities exist whereby empirical corrections can be introduced directly into the theoretical solution. Ashley (ref. 7) has discussed two such "irrational correction methods". The first of these is of interest here and concerns the calculation of the downwash boundary condition and the pressure distribution by "local linearization" in terms of the local velocity V_L rather than the free stream velocity U_∞. The dimensionless downwash then becomes

$$\frac{w}{U_\infty}(x,y,0,t) = \frac{1}{U_\infty}\left[\frac{\partial h}{\partial t} + V_L(x,y)\frac{\partial h}{\partial x}\right]$$

where $h(x,y,t)$ is the deflection of the mean surface. Applications of this "local linearization" to the downwash boundary condition (but not to the kernel function nor pressure coefficient) have been made for control surfaces by Ashley and Rowe (ref. 8) and by Rowe, Winther, and Redman (ref. 9), and improved correlations have been obtained. Tijdeman and Zwaan (ref. 10) have also employed the local linearization of the downwash boundary condition but have suggested another modification for use in the Doublet-Lattice Method for high subsonic flows, viz., that the free stream Mach number be replaced by a mean Mach number M_{jl} for each panel (lifting element or box). The downwash induced at box i by the lifting pressure on box j then becomes

---

* The format of a full matrix was briefly investigated but it was found to destroy the distributional character of the theoretical aerodynamic influence coefficients, and was not considered further.

$$\frac{w_i}{U_\infty} = D_{ij} (M_{j1}, k_r) \Delta C_{pj}$$

where $D_{ij} (M_{j1}, k_r)$ is the downwash influence coefficient between the jth lifting element and the ith downwash collocation point and its functional dependence on M_{j1} and the reduced frequency k_r is indicated. Tijdeman and Zwaan denote the freestream Mach number by M_∞ and the local Mach number at the surface of box i by M_{i2}, so that the locally linearized downwash for harmonic motion becomes

$$\frac{w_i}{U_\infty} = \frac{M_{i2}}{M_\infty} \frac{\partial h_i}{\partial x} + i \frac{\omega}{U_\infty} h_i$$

The values of M_1 and M_2 for a certain box are not equal, in general, because M_1 has to reflect the influence of the Mach number distribution normal to the surface ranging from M_2 at the surface to the freestream Mach number far away from the wing. Preliminary results from NLR calculations have shown that M_1 can be chosen simply to be the average value of M_2 and M_∞.

LIST OF SYMBOLS

A	Matrix that gives pressures in terms of downwash. Inverse of D
a_∞	Speed of sound in the free stream
$\tilde{a}$	Constraining power of a constraint. If $\tilde{a} = 1$ constraint is 100% effective. If $\tilde{a} = 0$ constraint has no effect at all
B	Wing bending moment (about x-axis)
$b/2$	Semi span of wing
C	Aerodynamic coefficient (e.g. lift or moment coefficient). C_e is used as an experimental constraint
C_L	Lift coefficient
C_M	Moment coefficient
C_B	Wing root bending moment coefficient
$\bar{c}$	Reference chord length
c_ℓ	Section lift coefficient
c_m	Section moment coefficient
c_h	Section hinge moment coefficient
D	Matrix that gives downwash (normalwash) in terms of pressures
h	Deflection normal to lifting surface
i_a	Unit vector in direction of axis
k_r	Reduced frequency $\frac{\omega\bar{c}}{2U_\infty}$
M_1	An average Mach Number between M_2 and M_∞

M_2	Local Mach Number at wing surface
M_∞	Free stream Mach Number
$\bar{M}_\infty$	Average Mach Number between sending and receiving point
R	Compressible radius (see Eqn. 73)
S	Matrix that integrates pressures into aerodynamic parameters (e.g., C_L, C_M, c_ℓ, etc.)
S_p	$[S]\,[\Delta C_{p_t}]$ see Equation (9)
S_w	$[S]\,[A][w]$ see Equation (30)
$\bar{S}$	See Equation (45)
$\bar{\bar{S}}$	$[S]\left[\begin{array}{c\|c}\phi & 0\\ \hline 0 & I\end{array}\right]$ See Equation (54)
S_p^*	See Equation (67)
T	Weights given to the correction factors $\bar{\varepsilon}$ for the minimization process, $\sum T\,\bar{\varepsilon}^2 = \min$
t	Time
U_∞	Free stream velocity
V_L	Local surface speed
W	Correction factor = $1 + \varepsilon$ (Called CF in computer program)
w	Downwash (or normalwash) (Called W in computer program)
wT	Weights in the minimization process for estimates
x,y,z	Cartesian coordinates right handed system x aft, y lateral (starboard), z vertical
α	Angle-of-attack, also direction cosine for force or moment axis
β	$\sqrt{1 - M_\infty^2}$
γ	Direction cosine for force or moment axis

$\bar{\gamma}$	Dihedral of lifting surface
ΔC_p	Lower surface minus upper surface pressure coefficient
ΔA	Box area
ε	Incremental correction factors = W - 1
$\bar{\varepsilon}$	Generalized incremental correction factors $\varepsilon = \phi\bar{\varepsilon}$
$\tilde{\varepsilon}$	$\varepsilon \sqrt{T}$
ϕ	Correction factor mode shapes
ϕ_d	Doublet potential function
ω	Circular Frequency

Subscripts and Superscripts

a	Stands for either p or w
d	Designated or known correction factors
e	Experimental
q	Identifies estimates as opposed to constraints
H	Hermetian transpose
mod.	Modified values
p	Identifies pressure modifying terms in the correction factor procedure
t	Theoretical values
u	Undesignated or unknown correction factors
w	Identifies downwash modifying terms
1	Deflection mode 1
2	Deflection mode 2
3/4	Three quarter chord point
1/4	One quarter chord point

Matrix Notation

{ }	Column Matrix
[ ]	Rectangular
⌈ ⌋	Diagonal

THEORETICAL DEVELOPMENT

Basic Method

Premultiplying Correction Factor Matrix. - A derivation of a real premultiplying correction matrix constrained to match aerodynamic data from a single downwash mode is presented in references 1 - 4. The use of Lagrange multipliers considerably simplifies the derivation so we present this alternative derivation here. As an introduction we will rederive the same case first, i.e., the premultiplier for a single mode; then we will consider multiple modes and the postmultiplier. Whether the correction matrix is real or complex depends only on the experimental data: static data lead to a real matrix and oscillatory data lead to a complex matrix.

Assume that we have a matrix [A] of theoretical aerodynamic influence coefficients (AIC's) that relates the theoretical pressures $\{C_{p_t}\}$ on a set of aerodynamic finite elements to the dimensionless downwashes $\{w\}$ at the same aerodynamic elements by

$$\{\Delta C_{p_t}\} = [A]\{w\} \tag{1}$$

The AIC's correspond to the reduced frequency of the experimental data and, hence, are real for static data and complex for oscillatory data. The premultiplying correction matrix $\lceil W_p \rfloor$ is used to obtain an estimate of the experimental pressure distribution $\{\Delta C_{p_e}\}$ from the theoretical distribution from

$$\{\Delta C_{p_e}\} = \lceil W_p \rfloor \{\Delta C_{p_t}\} \tag{2}$$

The subscript p refers to modification of the pressure distribution. The experimental force distribution is usually not known from the test data but only the integrated force and moment coefficients $\{C_e\}$ are measured. An integration matrix [S] relates the experimental force distribution to the measured force coefficients through;

$$\{C_e\} = [S]\,\{\Delta C_{p_e}\} \tag{3}$$

Combining equations (1) - (3) yields

$$\{C_e\} = [S]\,\lceil W_p \rfloor\,[A]\,\{w\} \tag{4}$$

which is the equation to be solved for the correction matrix $\lceil W_p \rfloor$ given all the remaining terms in the equation. The remaining terms are all known: $\{C_e\}$ and $\{w\}$ are obtained from the test data, and [S] and [A] are known from the mathematical model and the theoretical aerodynamic analysis of the configuration. In general, equation (4) is underdetermined, i.e., there are many more unknowns than equations. The method of least squares provides a solution. We require that changes in the theoretical load distribution shall be as uniform as possible or, in least-squares terminology, the weighted sum of the squares of the deviations shall be a minimum, where the deviation $\{\varepsilon_p\}$ is defined as the difference between the correction factors and unity.

$$\{\varepsilon_p\} = \{W_p - I\} \tag{5}$$

We denote the weighting function by T_p; it will be discussed below. The weighted least-squares condition then becomes

$$\begin{aligned} \sum T_p\, \varepsilon_p^{\,2} &= \{\varepsilon_p\}^H \lceil T_p \rfloor \{\varepsilon_p\} \\ &= \text{a minimum} \end{aligned} \tag{6}$$

where H denotes a Hermitian (complex conjugate) transpose. The Lagrange multipliers may be introduced by defining the error functional

$$f_p = (1/2)\,\{\varepsilon_p\}^H \lceil T_p \rfloor \{\varepsilon_p\} \tag{7}$$

and rewriting the measured generalized force coefficients (the constraints) as

$$\{C_e\} = [S] \lceil 1 + \varepsilon_p \rfloor \{\Delta C_{p_t}\}$$

$$= [S] \{\Delta C_{p_t}\} + [S] \lceil \Delta C_{p_t} \rfloor \{\varepsilon_p\}$$

The term $[S] \{\Delta C_{p_t}\}$ is just the theoretical integrated pressures which are the theoretical coefficients, $\{C_t\}$. Thus

$$\{C_t\} = [S] \{\Delta C_{p_t}\} \tag{8}$$

Introducing also the following

$$[S_p] = [S] \lceil \Delta C_{p_t} \rfloor \tag{9}$$

$$\{\Delta C_e\} = \{C_e\} - \{C_t\}$$

gives finally

$$\{\Delta C_e\} = [S_p] \{\varepsilon_p\} \tag{10}$$

The variation of the error functional f_p is

$$\delta f_p = \{\varepsilon_p\}^H \lceil T_p \rfloor \{\delta\varepsilon_p\} \tag{11}$$

and the variations of the incremental constraints, ΔC_e, given by equation (10) are

$$\{\delta\Delta C_e\} = [S_p] \{\delta\varepsilon_p\} \tag{12}$$

$$= 0$$

The condition for the minimum subject to the constraints is then a linear combination of equations (11) and (12) set to zero in which the linear factors are the Lagrange multipliers λ_p,

$$\delta f_p + \{\lambda_p\}^H \{\delta \Delta C_e\} = 0 \tag{13}$$

Substituting equations (11) and (12) into equation (13) yields

$$(\{\varepsilon_p\}^H \lceil T_p \rfloor + \{\lambda_p\}^H [S_p]) \{\delta \varepsilon_p\} = 0$$

Since the variation $\{\delta \varepsilon_p\}$ is arbitrary,

$$\{\varepsilon_p\}^H \lceil T_p \rfloor + \{\lambda_p\}^H [S_p] = 0$$

or, after Hermitian transposition.

$$\lceil T_p \rfloor \{\varepsilon_p\} + [S_p]^H \{\lambda_p\} = 0 \tag{14}$$

The simultaneous solution of equations (5), (10) and (14) yields the desired solution. The simultaneous solution leads first to the Lagrange multipliers and then to ε_p as follows:

$$\{\lambda_p\} = -([S_p] \lceil T_p \rfloor^{-1} [S_p]^H)^{-1} \{\Delta C_e\} \tag{15}$$

$$\{\varepsilon_p\} = -\lceil T_p \rfloor^{-1} [S_p]^H \{\lambda_p\} \tag{16}$$

and the correction factors are then

$$\{W_p\} = \{I\} + \{\varepsilon_p\} \tag{17}$$

The premultiplying correction factors are written in a diagonal format for use in subsequent aeroelastic analyses as in equation (2).

The above results can be restated in summary form as follows:

$$\text{Solution of} \rightarrow \{\Delta C_e\} = [S_p] \{\varepsilon_p\} \tag{18}$$

$$\text{subject to} \rightarrow \sum \varepsilon_p^2 T_p = \text{min.} \tag{19}$$

$$\text{is} \qquad \rightarrow \{\tilde{\varepsilon}\} = [\tilde{S}_p]^H \left([\tilde{S}_p][\tilde{S}_p]^H\right)^{-1} \{\Delta C_e\} \tag{20}$$

$$\text{where} \quad [\tilde{S}_p] = [S_p] \left\lceil \sqrt{T_p} \right\rfloor^{-1} \tag{21}$$

$$\text{and} \quad \{\varepsilon_p\} = \{\tilde{\varepsilon}\} / \sqrt{T_p} \tag{22}$$

The weighting function T_p is arbitrary; the only requirement on it is that it should be positive. However, engineering judgment provides some guidance: if only a single constraint, e.g., the lift curve slope, is available, one would prefer all the correction factors to be simply the ratio of its experimental value to its theoretical estimate. Accordingly, the recommended choice for the weighting function for the premultiplier is

$$\{T_p\} = |[A]\{I\}| \tag{23}$$

However, other choices may be deserving of further investigation.

<u>Multiple Modes</u>. - We next consider multiple experimental downwash modes. The derivation will be presented using two modes $\{w_1\}$ and $\{w_2\}$. For these two modes, the theoretical pressure distributions are

$$\{\Delta C_{p_{t1}}\} = [A]\{w_1\}$$

$$\{\Delta C_{p_{t2}}\} = [A]\{w_2\}$$

so the incremental experimental force coefficients ΔC_{e_1} and ΔC_{e_2} become

$$\{\Delta C_{e1}\} = \{C_e\} - [S] \left\lceil \Delta C_{p_{t1}} \right\rfloor \{W_p\}$$

$$\equiv \{C_e\} - [S_{p1}]\{W_p\} \tag{24}$$

$$\{\Delta C_{e2}\} = \{C_e\} - [S] \lceil \Delta C_{p_{t2}} \rfloor \{W_p\}$$

$$\equiv \{C_e\} - [S_{p2}] \{W_p\} \qquad \text{(Continued)(24)}$$

These two equations may be combined into one set as follows:

$$\{\Delta C_e\} = [S_p] \{\varepsilon_p\} \tag{25}$$

where

$$\{\Delta C_e\} = \left\{ \begin{array}{c} \Delta C_{e_1} \\ \hline \Delta C_{e_2} \end{array} \right\}$$

and

$$[S_p] = [S_{p_1} \,\vdots\, S_{p_2}] \tag{26}$$

If again we impose the minimization condition

$$\sum \varepsilon_p^2 T_p = \min \tag{27}$$

then the solution is identical to equation (20) since equations (25) and (27) are identical to (18) and (19).

Postmultiplying Correction Factor Matrix. - Now we consider the postmultiplying correction matrix. It is only necessary to consider a single downwash mode; the multiple mode case can be generalized by reference to equations (25) and (26). Since the postmultiplier modifies the downwash mode it defines an effective experimental downwash given by

$$\{w_e\} = \lceil W_w \rfloor \{w\}$$

in which the subscript w refers to modification of the downwash. Our new estimate of the experimental pressure distribution becomes:

$$\{\Delta C_{p_e}\} = [A]\{w_e\}$$

$$= [A]\,[W_w]\,\{w\} \tag{28}$$

The experimental generalized forces or force coefficients are again given by equation (3) which with equation (28) becomes

$$\{C_e\} = [S]\,[A]\,\lceil 1+\varepsilon_w \rfloor\,\{w\}$$

$$= [S]\,\{\Delta C_{p_t}\} + [S]\,[A]\,\lceil w \rfloor\,\{\varepsilon_w\} \tag{29}$$

And again noting that $[S]\,\{\Delta C_{p_t}\} = \{C_t\}$ and introducing

$$[S_w] = [S]\,[A]\,\lceil w \rfloor \tag{30}$$

gives

$$\{C_e\} - \{C_t\} = [S_w]\,\{\varepsilon_w\}$$

or since
$$\{C_e\} - \{C_t\} = \{\Delta C_e\}$$

$$\{\Delta C_e\} = [S_w]\,\{\varepsilon_w\} \tag{31}$$

Again the minimization condition is imposed,

$$\sum \varepsilon_w^2\,T_w = \text{min.} \tag{32}$$

The solution for ε_w is then identical to equation (20) since equations (31) and (32) are identical to equations (18) and (19). Again the correction factors are calculated from ε_w as follows:

$$\{W_w\} = \{I\} + \{\varepsilon_w\} \tag{33}$$

The weighting function T_w is also arbitrary. However, the considerations that led to the recommendation of equation (23), in the premultiplying case, also lead to

$$\{T_w\} = | \{I\}^T [A] | \tag{34}$$

which is to say that the weighting function is the lift coefficient induced by a unit downwash at each lifting element. Equation (34) is the recommended choice for the weighting function in the postmultiplying case, although other choices may still warrant further investigation.

For multiple modes, say two, equations (31) and (32) provide the following:

$$\{\Delta C_{e_1}\} = [S_{w_1}] \{\varepsilon_w\} \tag{35}$$

$$\{\Delta C_{e_2}\} = [S_{w_2}] \{\varepsilon_w\} \tag{36}$$

$$\sum \varepsilon_w^2 T_w = \min \tag{37}$$

Again equations (35) and (36) can be combined into one as follows:

$$\{\Delta C_e\} = [S_w] \{\varepsilon_w\} \tag{38}$$

where

$$\{\Delta C_e\} = \left\{ \begin{array}{c} \Delta C_{e_1} \\ - - - \\ \Delta C_{e_2} \end{array} \right\}$$

and where

$$[S_w] = [S_{w_1} \mid S_{w_2}] \tag{39}$$

Equations (38) and (37) are now identical to equations (18) and (19) respectively and thus the solution is the same as before, i.e., equation (20).

Modifications to the Basic Method

In some instances correction factors become unrealistic. In order to correct this situation when it occurs or to minimize the probability of its occurrence initially, various modifications can be introduced. Three such modifications are discussed here, i.e., estimates, correction factor modes and limits. Estimates are like constraints except that the "constraining power" can be varied. Correction factor modes constrain the distribution of correction factors such that the final distribution is a superposition of a limited set of well behaved, user input mode shapes. The "limit" feature constrains the correction factors (or any subset of them) to be above a given minimum and below a given maximum.

<u>Estimates</u>. - In some instances data will be available in the form of estimates. These estimates can be based on past data, data from related configurations, two-dimensional data, empirical methods, or just past experience. In any case they do not have equal weight with the experimental data considered so far. Consider the case where some experimental data are available, leading to $\{\Delta C_e\}$, then the usual equation applies to these data:

$$\{\Delta C_e\} = [S_a]\{\varepsilon_a\}$$

where the subscript "a" stands for either p or w. If estimates exist, leading to $\{\Delta C_g\}$, then it is desirable to minimize the difference between these estimates and the modified theoretical values. Let this difference be termed $\{\varepsilon_g\}$, then:

$$\{\Delta C_g\} = [S_g]\{\varepsilon_a\} + \{\varepsilon_g\}$$

where $\{\Delta C_g\} = \{C_g - C_t\}$ and S_g is analogous to S_a with the exception that it refers to the estimates C_g and not the constraints C_e. Thus we wish to minimize both $\{\varepsilon_a\}$ and $\{\varepsilon_g\}$ together and this is done as follows:

$$\left\{ \begin{array}{c} \Delta C_e \\ \hline \Delta C_g \end{array} \right\} = \left[\begin{array}{c|c} S_a & 0 \\ \hline S_g & I \end{array} \right] \left\{ \begin{array}{c} \varepsilon_a \\ \hline \varepsilon_g \end{array} \right\}$$

This equation can then be solved in the usual manner producing the following result:

$$\sum T\varepsilon_a^2 + \sum \varepsilon_g^2 = \min$$

If it is desired to give the ε_g values more or less weight in the minimization scheme, then the ε_g values must be weighted.

$$\sum T\varepsilon_a^2 + \sum (wT\ \varepsilon_g)^2 = \min. \tag{40}$$

The equations for the constraints then become

$$\left\{ \begin{array}{c} \Delta C_e \\ \hline \Delta C_g \end{array} \right\} = \left[\begin{array}{c|c} S_a & 0 \\ \hline S_g & \frac{1}{wT} \end{array} \right] \left\{ \begin{array}{c} \varepsilon_a \\ \hline \varepsilon_{wT} \end{array} \right\} \tag{41}$$

where

$$\{\varepsilon_{wT}\} = [wT]\ \{\varepsilon_g\} \tag{42}$$

and where the values wT are the weights assigned to the errors ε_g. If the estimates are of high quality then the weights will be large. In the limit as $wT \to \infty$, ΔC_g becomes a constraint instead of an estimate and equation (41) reduces to the form of equation (18). Equation (41) can also be cast into the same form as equation (18) for the general case, i.e., wT finite, as follows:

$$\{\Delta C_e\} = [\bar{S}]\ \{\varepsilon\} \tag{43}$$

with

$$\{\Delta C_e\} = \left\{ \begin{array}{c} \Delta C_e \\ \hline \Delta C_g \end{array} \right\} \tag{44}$$

$$[\bar{S}] = \left[\begin{array}{c|c} S_a & 0 \\ \hline S_g & \frac{1}{wT} \end{array} \right] \tag{45}$$

$$\{\varepsilon\} = \left\{ \begin{array}{c} \varepsilon_a \\ \hline \varepsilon_{WT} \end{array} \right\} \tag{46}$$

Equation (40) can be written in terms of ε as:

$$\sum T^* \varepsilon^2 = \min \tag{47}$$

where

$$T^* = \begin{cases} T \text{ for constraints} \\ \frac{1}{(wT)^2} \text{ for estimates} \end{cases}$$

Thus equation (43) and (47) are formally identical to equation (18) and (19) and thus have the same solution, i.e., equation (20).

Currently the term $\frac{1}{wT}$ is obtained from a term $\tilde{a}$ where

$$\frac{1}{wT} = \frac{1-\tilde{a}}{\tilde{a}} \tag{48}$$

$$10^{-4} \leq \tilde{a} \leq 1.0$$

where $\tilde{a}$ is called the constraining power of the estimate C_g.

Correction Factor Modes. - The correction factors can be expressed in terms of a set of modes ϕ as follows:

$$\{\varepsilon\} = [\phi] \{\bar{\varepsilon}\} \tag{49}$$

Placing equation (49) into (43) gives

$$\begin{aligned} \{\Delta C_e\} &= [\bar{S}] [\phi] \{\bar{\varepsilon}\} \\ &= [\bar{\bar{S}}] \{\bar{\varepsilon}\} \end{aligned} \tag{50}$$

where

$$[\bar{\bar{S}}] = [\bar{S}] [\phi] \tag{51}$$

If the minimization process is applied to $\bar{\varepsilon}$ as usual

$$\sum \bar{\varepsilon}^2 = \min \tag{52}$$

Here the weight T is missing since it is usually not used with modes. Equations (50) and (52) are then identical to equations (18) and (19) and thus have the same solution, i.e., equation (20). A similar expression exists for the postmultiplying correction factors. This approach allows a bias based on experience and past tests and physical reasoning, to be built into the correction factors. When estimates are considered equation (50) must be altered since the ε_{wT} are not fitted with correction factor modes. Thus

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_a \\ \hline \varepsilon_{wT} \end{Bmatrix} = \left[\begin{array}{c|c} \phi & 0 \\ \hline 0 & I \end{array}\right] \begin{Bmatrix} \bar{\varepsilon}_a \\ \hline \varepsilon_{wT} \end{Bmatrix} \tag{53}$$

and thus

$$[\bar{\bar{S}}] = [\bar{S}] \left[\begin{array}{c|c} \phi & 0 \\ \hline 0 & I \end{array}\right] \tag{54}$$

and then the solution proceeds as before.

Limits. - If certain basic properties of the weight factors are known, they could be limited to fall within a given set of bounds. If for instance the sign of an incremental weight factor is known to be positive, then it could be constrained to be positive. Also, for practical reasons, the maximum value of the weight factors should be limited and thus the incremental weight factors are constrained to lie below this maximum. In general, the weight factors can be constrained to lie between a maximum and a minimum.

$$\bar{\varepsilon}_{min} \leq \bar{\varepsilon} \leq \bar{\varepsilon}_{max} \tag{55}$$

Notice that the generalized incremental correction factors, $\bar{\varepsilon}$, are the ones limited in the solution and not the actual ones, ε. The values of $\bar{\varepsilon}$ are the coefficients of the correction factor modes, ϕ, and not the incremental correction factors themselves.

The basic procedure would be to set any generalized incremental weight factor to its maximum or minimum value if it exceeded these limits. This would require a multistep operation: (1) solving for the factors, (2) checking and setting those that exceeded the limits to the limit values, and (3) resolving. Before this can be done a capability must exist for assigning weight factors special values. This is easily accomplished as follows:

$$\{\Delta C_e\} = [\bar{\bar{S}}_u | \bar{\bar{S}}_d] \left\{ \begin{array}{c} \bar{\varepsilon}_u \\ \hline \bar{\varepsilon}_d \end{array} \right\} \tag{56}$$

where $[\bar{\bar{S}}]$ is defined in equation (51). The subscript u indicates those factors that are undesignated and d indicates those that are designated. This equation can be solved for $\{\varepsilon_u\}$ in terms of the known quantities:

$$\{\Delta C_e\} - [\bar{\bar{S}}_d] \{\bar{\varepsilon}_d\} = [\bar{\bar{S}}_u] \{\bar{\varepsilon}_u\} \tag{57}$$

Equation (57) effectively eliminates the designated factors from the minimization process. This equation can then be solved in the usual manner for $\{\bar{\varepsilon}_u\}$ since $\{\bar{\varepsilon}_d\}$ is given. Specifically

$$\left\{ \Delta C_{e_{mod}} \right\} = [\bar{\bar{S}}_u] \{\bar{\varepsilon}_u\} \qquad (58)$$

where

$$\left\{ \Delta C_{e_{mod}} \right\} = \{\Delta C_e\} - [\bar{\bar{S}}_d] \{\bar{\varepsilon}_d\} \qquad (59)$$

The minimization scheme is then

$$\sum T^* \bar{\varepsilon}_u^{\,2} = \min \qquad (60)$$

Equations (58) and (60) are now formally identical to equations (18) and (19) and thus the solution is identical to equation (20). In the computer program the final $\bar{\varepsilon}$ array that are modified or have reached their limits, is called $\bar{\bar{\varepsilon}}$.

A New Postmultiplying Correction Factor Matrix

The postmultiplying correction factor matrix developed in a previous section has been applied successfully to wings operating in pitch. Problems arise however when control surface modes are used. The discussion to be presented in the "Correlation Studies" section describes some of these problems. As a result of these, a new postmultiplying correction factor matrix was developed and it is derived here.

Viscous effects on airfoils can be thought of in terms of a displacement thickness added to the airfoil. The difference between the upper and lower surface displacement thicknesses produces a "decambering" of the airfoil or a change in the downwash w.

$$w_e = w + \delta w \tag{61}$$

The changes in downwash, δw, exist over the entire airfoil or wing and not just in the region where w is non zero. These changes are a function of the pressure distribution on the airfoil. Usually the displacement thickness at a point is an integral function of the pressure distribution upstream of that point. If the general case of correction factor mode shapes is assumed then the downwash correction δw can be expressed as:

$$\{\delta w\} = [\phi] \{\delta \bar{e}\} \tag{62}$$

where $\{\delta \bar{e}\}$ is proportional to the integrated pressures, $\lceil \ell \rfloor$.

$$\{\delta \bar{e}\} = \lceil \ell \rfloor \{\bar{\varepsilon}\} \tag{63}$$

where $\lceil \ell \rfloor$ is given in terms of an integration matrix [N] and the pressures $\{\Delta C_p\}$.

$$\{\ell\} = [N] \{\Delta C_p\} \tag{64}$$

Combining equations (63) with (62) and using the result to obtain the corrected pressures leads to

$$\{\Delta C_{p_e}\} = [A] \{w_e\}$$

$$= [A] \{w + [\phi] \lceil \ell \rfloor \{\bar{\varepsilon}\}\} \qquad (65)$$

The constraints $\{C_e\}$ are obtained by integrating $\{\Delta C_{p_e}\}$ as follows:

$$\{C_e\} = [S] \{\Delta C_{p_e}\}$$

$$= [S] \{\Delta C_{p_t}\} + [S_p*] \{\varepsilon_w\} \qquad (66)$$

where

$$[S_p*] = [S] [A] [\phi] \lceil \ell \rfloor \qquad (67)$$

Noting that $\{\Delta C_e\} = \{C_e\} - \{C_t\}$ equation (65) can be written as:

$$\{\Delta C_e\} = [S_p*] \{\varepsilon_w\} \qquad (68)$$

Equation (68) has a form identical to that of equation (18) except $[S_p]$ is replaced by $[S_p*]$ and thus has the same solution, i.e., equation (20). Once found, $\{\varepsilon_w\}$ can be placed into the expression for $\{\Delta C_{p_e}\}$, in equation (65), and the desired modified pressure found.

Currently in the computer program the matrix [N] is simply either the identity matrix or the matrix $[\phi]^T$. The identity matrix implies that the correction to the downwash is proportional to the local lifting pressure. In addition the above derivation is good for only one mode and thus the multiple mode option can not be used with the new postmultiplier. The new postmultiplier can be extended to multiple modes by simply replacing

$$\{\Delta C_e\} \text{ with } \left\{\begin{array}{c}\Delta C_{e_1} \\ \hline \Delta C_{e_2}\end{array}\right\} \qquad (69)$$

and

$$[S_p^*] \text{ with } \left[\begin{array}{c} S_{p_1}^* \\ \hline S_{p_2}^* \end{array}\right] \tag{70}$$

but this has not yet been tried.

Transonic Effects Using Local Mach Number

Empirical modification of theory is most meaningful if the theory qualitatively matches experimental data. If the theory misses an important feature of the data the modified theory will also usually miss it. Transonic effects fall in this category. The classic lifting surface theory makes no provision for transonic effects and it is the purpose of this section to investigate some simple modifications to help remedy this situation.

Direct Application of Local Mach Number. - Several methods based on the steady local Mach Number distribution have been tried and the results are discussed in later sections.

One of these methods, discussed in the Introduction, consists of making a simple substitution of a local Mach Number distribution in place of its free stream value both in the kernel function and in the boundary conditions and pressure equation (see refs. 7 and 10). The local Mach Number distribution is taken from steady flow results. For the kernel function, application of a Mach Number distribution that lies somewhere in between the surface values and the free stream value was used. Tijdeman and Zwaan (ref. 10) suggest a local Mach Number distribution that lies half way in between the actual local and the free stream values. The reason for this is that acoustic signals propagate to the surface along various paths out in the fluid and thus propagate at some average between the surface value and free stream value. For the kernel, the local receiving point value of Mach Number and the free stream values were averaged and used in place of the free stream value.

For the normalwash boundary condition and pressure evaluation the local Mach Number on the surface was used. Specifically, if $M_2(x)$ is the local surface steady Mach Number distribution then the normalwash boundary condition w is (see ref. 10):

$$\frac{w}{U_\infty} = \frac{M_2(x)}{M_\infty} \frac{\partial h}{\partial x} + i \frac{\omega}{U_\infty} h \tag{71}$$

The second order Bernoulli Equation for steady flow is (see ref. 7):

$$\Delta C_{p_{so}} = \Delta C_p \left[1 + \beta^2 \left(\frac{M_2(x)}{M_\infty} - 1\right)\right] \tag{72}$$

where ΔC_p and $\Delta C_{p_{so}}$ are the first and second order pressures respectively. This method did not prove to be all that was hoped for and thus a second method was investigated.

A New Transonic Effects Method. - In this section a derivation of the newly developed Douglas transonic effects method is presented. The basic method was conceived under the McDonnell-Douglas IRAD program however its implementation in two-dimensions and its application to the airfoil-control surface problem was done under the current contract.

The lifting surface method is based on the following expression for the potential of a doublet, ϕ_d:

$$\phi_d = \frac{\partial}{\partial n} \left\{ \frac{e^{i\omega t} e^{i\lambda[M_\infty(x-\xi)-R]}}{R} \right\} \tag{73}$$

where

$$\lambda = \frac{\omega M_\infty}{\beta^2 U_\infty} , \quad R = \sqrt{(x-\xi)^2 + \beta^2 r^2}$$

and n is the direction normal to the lifting surface. This expression can be rewritten as:

$$\phi_d = \frac{\partial}{\partial n} \left\{ \frac{e^{i\omega(t-\tau)}}{R} \right\} \tag{74}$$

where

$$\tau = \frac{M_\infty}{\beta^2 U_\infty} [R - M_\infty(x-\xi)] \tag{75}$$

It can be shown that τ has a physical significance. The term τ is the acoustic time delay between the sending and receiving points. That is, τ is the time it takes an acoustic signal, originating at the point ξ, η, ζ to reach the point x, y, z as the acoustic pulse washes downstream.

This statement can be illustrated by the example of figure 1.

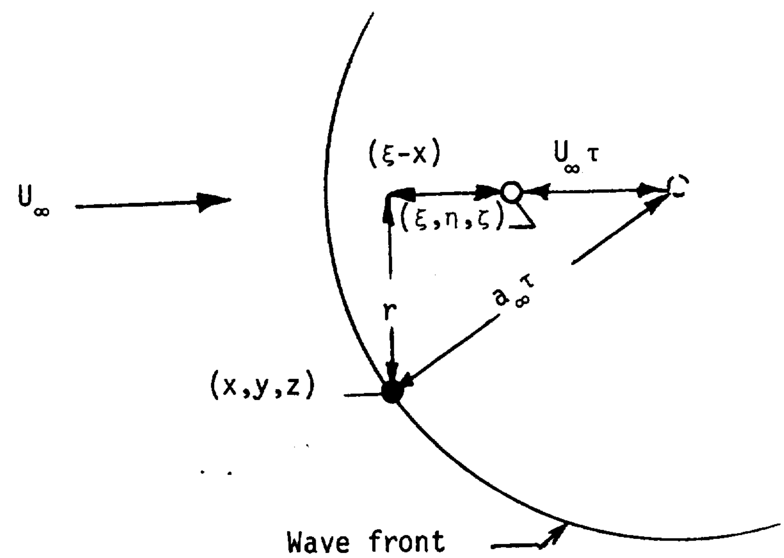


Figure 1

A signal is emitted at (ξ, η, ζ) at time τ = 0. At time τ the wave front, traveling at the speed of sound, a_∞, has reached the receiving point at (x, y, z). During this time the wave center has travelled a distance $U_\infty\tau$. Using the right triangle relations gives:

$$r^2 + (\xi - x + U_\infty \tau)^2 = a_\infty^2 \tau^2 \tag{76}$$

Solving for τ using the solution for a quadratic equation gives:

$$\tau = \frac{M_\infty}{U_\infty \beta^2} (M_\infty(\xi - x) + R)$$

which is exactly what is given in equation 75.

Thus τ in the expression for ϕ_d has physical significance and it is the acoustic time delay between the sending and receiving points in a fluid moving with uniform velocity. This physical insight can form the basis of a correction factor for the theory. For instance if the wave is in a flow field whose velocity varies in the longitudinal direction then the distance d travelled by the wave center is not $U_\infty \tau$ but is:

$$d = \int_0^\tau U(t)\, dt \tag{77}$$

If we consider U(t) to be made up of $U_\infty + \delta U(t)$ then d also can be so split.

$$d = U_\infty \tau + \delta d \tag{78}$$

$$\delta d = \int_0^\tau \delta U(t) dt \tag{79}$$

The wave center velocity U is being discussed, however this is not the velocity of the fluid particle located at the wave center as is the case for a uniform flow. The velocity U actually reflects the wave front speed and location and is the speed of an imagined wave center for the wave that strikes the receiving point. The wave front speed varies around the circumference of the wave, but the most important part of the wave is that part that strikes the receiving point. Thus the velocity U(t) is the time history of the wave center corresponding to that part of the wave that strikes the receiving point (x, y, z). As an approximation to the location of this part of the wave, it is assumed that it lies along a line connecting the receiving and sending points (shown dotted in figure 2). Thus δU is the difference between the local velocity and the free stream velocity along the dotted line. If a

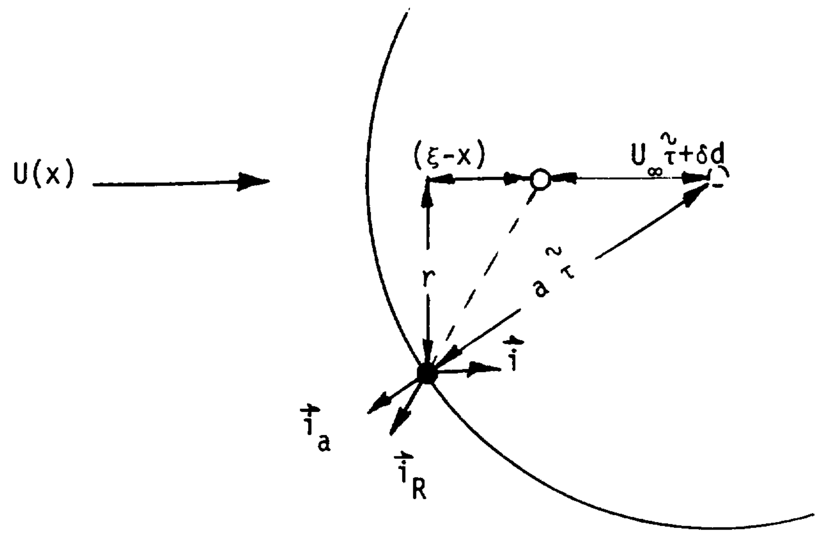


Figure 2

coordinate $\bar{R}$ is defined lying along this line then the integral in time of equation (79) can be converted into a space integral in $\bar{R}$ as follows:

$$\delta d = \int_0^R \delta U(\bar{R}) \frac{dt}{d\bar{R}} \, d\bar{R}$$

where $\bar{R}$ is defined below equation (73) and where $\frac{d\bar{R}}{dt}$ is the speed with which the wave front moves along the radial coordinate.

$$\frac{d\bar{R}}{dt} = (a_\infty \vec{i}_a + U(\bar{R})\vec{i}) \cdot \vec{i}_R$$

The unit vectors $\vec{i}_a$, $\vec{i}$ and $\vec{i}_R$ are defined in figure 2.

For example, in the two-dimensional analysis for coplanar surfaces:

$$\delta d = \int_{x}^{\xi} \Delta U \frac{dt}{d\bar{x}} \, d\bar{x}$$

$$\frac{d\bar{x}}{dt} = a - U(\bar{x}) \tag{80}$$

Thus

$$\delta d = \int_{\xi}^{x} \frac{M(\bar{x}) - U_\infty/a}{1 - M(\bar{x})} \, d\bar{x} \tag{81}$$

where $M(\bar{x})$ is the local Mach Number distribution. As an approximation set $U_\infty/a = U_\infty/a_\infty = M_\infty$.

The time $\tilde{\tau}$ can now be calculated using the right-triangle relations and the quadratic formula solution.

$$[U_\infty\tilde{\tau} + \delta d + \xi - x]^2 + r^2 = a_\infty^2\tilde{\tau}^2$$

Solving for $\tilde{\tau}$ gives:

$$\tilde{\tau} = \frac{M_\infty}{U_\infty\beta^2} \{M_\infty(\xi - \tilde{x}) + \tilde{R}\} \tag{82}$$

where

$$\tilde{R}^2 = (\xi - \tilde{x})^2 + \beta^2 r^2 \tag{83}$$

$$\tilde{x} = x - \delta d \tag{84}$$

It is immediately evident that $\tilde{\tau}$ has exactly the same form as τ (see equation (75)) except that x is replaced, in the expression for τ, by $x - \delta d$ in the expression for $\tilde{\tau}$. In essence then the receiving and sending points have increased their separation (in the x-direction) as far as the acoustic time delay τ is concerned. Is there any reason to carry this increase in distance to other parts of the potential function, specifically, to the radius term in the denominator? It seems so. First, this radial distance is already modified in the expression for $\tilde{\tau}$, see equation (82). Second it is known that transonic effects exist in steady flow ($\omega = 0$) where τ has no effect; i.e., $\phi_d = \frac{\partial}{\partial n}(1/R)$. Thus it seems appropriate to add $-\delta d$ to all $x - \xi$ terms. Thus

$$\tilde{\phi}_d(x - \xi, y - \eta, z - \zeta, \omega, M_\infty) = \phi_d(x - \xi - \delta d, y - \eta, z - \zeta, \omega, M_\infty) \quad (85)$$

where $\tilde{\phi}_d$ is the potential modified for transonic effects.

This method has been implemented for the two-dimensional case and the results are discussed subsequently.

One variation of this method that is possible is to use an average Mach Number between sending and receiving points and define δU as the difference between the local value of velocity and this average. Thus the term M_∞ is replaced with $\bar{M}_\infty$ where

$$\bar{M}_\infty = \frac{1}{(x-\xi)} \int_{\xi}^{x} M(\bar{x})d\bar{x} \quad (86)$$

This method has also been tried and results using this variation are also discussed subsequently.

A final consideration is the determination of the local Mach Number distribution, $M(\bar{x})$. Tijdeman and Zwaan (ref. 10) note that the local surface Mach Number distribution should not be used but that some average between it and the free stream Mach Number should be used. This is because the signal arriving at a point has traveled both in the vicinity of the airfoil and out in the flow field. The recommendation of Tijdeman and Zwaan has been adopted in the present method.

Other work by Tijdeman and Bergh (ref. 11) can also be brought to bear on this work. Specifically a fully two-dimensional acoustic solution of a source pulse located at the control surface hinge was calculated for the case of a nonuniform flow field. This solution produced the exact time lag $\tilde{\tau}$ from the hinge line to all other points on the airfoil. The equivalent distance $\tilde{x}$, and also $\tilde{R}$, from the hinge line to the receiving point can then be calculated using this information and the equation relating $\tilde{\tau}$ to $\tilde{x}$. Thus

$$\xi - \tilde{x} = \sqrt{\tilde{\tau}^2 U_\infty^2 \beta^2 - r^2}$$

If acoustic solutions were obtained for all other sending points then all the necesary $\tilde{x}$ for this theory would be available. This method would be accurate, however it would require many expensive acoustic solutions. It appears that each of these solutions requires a computing effort comparable to a direct solution, by finite difference, of the original problem. This conclusion however, remains to be seen and further study is required.

CORRELATION STUDIES

Local Mach Number Studies

Several methods of accounting for local steady Mach Number variations in the oscillatory lifting surface theory have been studied for the two-dimensional case. A mathematical description of these methods has been presented in the Theoretical Development section. The cases considered here are for a two-dimensional symmetric airfoil (NACA 65A006) with an oscillating 25% chord flap. The local Mach Number variations over the airfoil at zero angle of attack are given in reference 11.

The first and simplest of the methods studied involves simply making a direct substitution of the steady local Mach Number in place of its free stream value. In general, this approach does not produce substantial changes in the pressures from their classical values.

In figure 3 the symbols marked by triangles indicate the pressures calculated using the local Mach Number in the downwash boundary condition. The pressures are reduced from their classical values (indicated by dots and a dashed line) as expected, but not by very much.

The circles indicate the pressures calculated using the local Mach Number in the downwash as well as in the kernel function. In the kernel function the average between the local receiving point Mach Number and the free stream value is used. The pressures again are generally reduced but not by any substantial amount.

The use of local Mach Number in the second order Bernoulli equation (Ashley, ref. 7) also produces very little change. In figure 5 this change is observed as the difference between the circles and triangles. This change is about the same order of magnitude as the other changes except it is generally in the opposite direction.

The second method studied is new and is described in the Theoretical

Development section. The basic idea of this method is to transform the longitudinal distance between sending and receiving points depending on the time it takes an acoustic signal to travel that distance. A variation of this method is simply to replace the free stream Mach Number, M_∞, by $\bar{M}_\infty$, an integrated average of the Mach Number distribution between sending element and receiving point defined in equation (86).

On the face of it the new method gives the best correlation when M_∞ and not $\bar{M}_\infty$ is used. Figure 4 presents a comparison between the two methods at a Mach Number of 0.875. Near the leading edge the basic method designated "Present Method (M_∞)" and indicated by triangles produces the best agreement between experiment and calculation. Near the position of the steady shock wave however the peak pressure is better predicted by the variation of the basic method designated "Present Method ($\bar{M}_\infty$) and indicated by circles. The location of the calculated peak is slightly forward of the experimentally observed peak. Either method, however, is better than the classic theory (indicated by dots) for predicting pressures as comparisons with experimental data shown.

Two features of the experimental pressure distribution illustrate transonic effects. One of these is the reduced leading edge pressure levels, and a second is the bump or peak in pressure near the location of the steady shock wave location. The new method qualitatively reproduces these features. However there is reason to suspect that the basic version of the new method (triangle) under-predicts the leading edge pressure. The reason for this lies in the fact that, even though the calculated and experimental pressures agree near the leading edge, viscous effects have not yet been accounted for and these effects reduce the calculated loads even further. A drop in the leading edge loading caused by application of viscous effects to the ($\bar{M}_\infty$) variation of the basic method (circles), renders this method more acceptable than before. However these effects are not large enough to bring the calculated pressures in line with the experimental values (see fig. 38). Further study is required in this area to decide which method is best or to discover other more accurate variations of the basic method.

The application of the new transonic method to a lower Mach Number, (0.85), is shown in figure 5. The agreement is good near the leading edge but only a slight indication of the shock bump is given by the theory. Also shown in this figure is the effect of local Mach Number on the Bernoulli equation (see equation (72)). The difference between the circles and triangles indicates this effect.

All applications thus far have been for the steady case. Figure 6 presents a comparison of the Present Method (new transonic theory) for the case of the control surface oscillating at a reduced frequency $k_r = \frac{\omega c}{2U_\infty} = 0.059$. Also shown in this figure is a calculation done using the Traci et al method (ref. 12) and a calculation done using the classic theory. The finite element theory of Traci et al predicts the bump at the shock wave fairly accurately however is not as good as the Present Method elsewhere. One part of the pressure distribution that does not seem to be predicted by any of the theories is the depth of the dip in pressure behind the shock.

Figure 7 presents a comparison similar to that in Figure 6 but at a lower Mach Number (0.85). Also instead of in phase (Real) and out of phase (Imaginary) parts given, amplitude $\Delta C_p = \sqrt{(\text{Re}\Delta C_p)^2 + (\text{Im}\Delta C_p)^2}$ and phase angle = $\tan^{-1}$ $(\text{Im}\Delta C_p/\text{Re}\Delta C_p)$ are presented. Again, as in the steady case the bump at the shock is barely noticeable in the Present Method and of course absent altogether in the classic theory.

Figure 8 presents a comparison of the Present Method, classic theory and experimental data for a case similar to that presented in figure 7 except that the Mach Number is 0.875 and the reduced frequency, $k_r = 0.176$. Again pressure amplitude and phase angle are shown. The two variations of the Present Method are in better agreement with the experimentally obtained pressure amplitudes than is the classic theory. However the same can not be said of the phase angles. The Present Method follows the experimental phase angle curve from about the 40% chord on to the trailing edge. However none of the theories follows the curve forward of that point. Tijdeman and Bergh (ref. 11) present a modified phase angle curve based on a full two-dimensional acoustic solution of a pulse located at the control surface leading edge (see section on New Transonic Effects Method). This approach gives very good agreement with the experimental phase angle data (see figure 31 of reference 11). The correction was applied only to the phase angles and not the pressure amplitudes. The calculated phase angle was simply corrected using the additional time lag over and above that experienced in uniform flow. This additional phase lag was not used internal to the theory but applied after the theoretical calculation was completed. The section of this report entitled "A New Transonic Effects Method" describes how this acoustic type of information can be used internally with the theory so that the pressure amplitudes are also effected. This approach has not yet been tried.

A possible explanation of the phase angle differences between theory and experiment might be due to the fact that signal fronts, which emanated from the control surface, do not exactly travel normal to the flow as assumed in the Present Method. Tijdeman and Bergh have shown that the wave fronts are actually inclined to the flow to a considerable degree, within the supersonic zone. This being the case the wave fronts impinge on the forward part of the airfoil (forward of the shock wave) with very little longitudinal time delay. This would explain the flattening of the phase angle curve in front of the shock.

Thus far detailed pressure distributions have been discussed. Attention is now focused on the forces and moments these pressures produce. Figures 9 through 12 present comparisons of the present method with the classic theory and experimental data. In figure 9 the Traci, Farr, Albano theory is also plotted. This figure shows that the Present Method (M_∞) is in better agreement with the data than is the classic method. As expected the Present Method and classic theory tend to coalesce at low Mach Numbers, out of the transonic region. The transonic peak lift, predicted by the Present Method, occurs earlier, as Mach Number is increased, than does the experimental data. Also the dip occurring after this peak is not nearly as deep as shown by the experimental data.

Both theories show values of lift coefficient that are higher than the experimental values. This is due to reduced flap effectivity caused by the viscous boundary layers.

Figure 10 presents a comparison similar to the previous figure except that pitching and hinge moment coefficients are considered. Of particular note is the over prediction of hinge moment by both theories. Again this is due to viscous effects on the flap.

The last two figures have dealt with force and moment coefficients in steady flow for an airfoil with a deflected flap. Figures 11 and 12 present the same data for the oscillatory case. (The reduced frequency varies from 0.098 at $M_\infty = 0.5$ to 0.057 at $M_\infty = 0.901$ in these figures.) Generally speaking the same trends and conclusions hold for these figures as for the previous two figures.

Subsonic Cases

Oscillating Wing with Control Surface. - Extensive low speed wind tunnel measurements of static and oscillatory pressure data have been made by Hertrich (refs. 14 and 15) on straight and swept wings with a full span control surface. The wings had no taper and the control surface had a 30% chord; two aspect ratios, 2.5 and 3.1, were tested by changing the exposed span in the tunnel. The swept wings had a sweep angle of 25^{o}. A later oscillatory test of the swept wing was made by Försching, Triebstein, and Wagener (ref. 16) in which the full span control surface was split approximately in half (the inboard flap had 46.59% of the span(and the aspect ratio was set at 2.94.

The pressure data from the first tests (refs. 14 and 15) were integrated by Hertrich to obtain lift and moment coefficients and the static values for the swept wing with aspect ratio 3.1 have been used here as constraints to determine correction factors. The static values corrected for wind-tunnel wall interference are: lift curve slope $C_{L_\alpha} = L/qs = 3.13$ per radian, pitching moment curve slope $C_{m_\alpha} = M/qS\bar{c}\alpha = 0.148$ per radian, flap lift effectiveness $C_{L_\delta} = L/qS\delta = 1.95$ per radian, flap pitching moment effectiveness $C_{m_\delta} = M/qS\bar{c}\delta = -0.432$ per radian, and the flap hinge moment coefficient $C_{h_\delta} = H/qS\bar{c}\delta = -0.0350 \cos 25^{o} = -0.03172$ per radian where the sweep correction is added to give the moment about the hinge axis; the reference area is $S = 0.564\ m^2$, the reference chord is $\bar{c} = 0.6$ m, and the pitch axis is located at 61.5% of the root chord.

Rolling moment coefficients were not derived from the pressure data and neither was the hinge moment due to angle of attack C_{h_α}. The available data permitted a maximum of five constraints, and two sets of constraints were investigated; the first set used two constraints from the angle of attack data, C_{L_α} and C_{m_α}, and the second set used all five constraints. The use of the flap rotation data alone without the angle of attack data was not considered.

The theoretical basis for the correction factors is the Doublet-Lattice Method (DLM) of reference 17. The idealization of the lifting surface consists of 110 boxes resulting from 11 strips and 10 equal chordwise divisions. The chordwise division on each strip result in 7 boxes on the primary surface and 3 boxes on the control surface. The strip widths Δy_i are chosen so that the strip centerlines fall along the lines of pressure taps. The span of 0.940 m is divided into the following strip widths from root to tip: Δy_1 = 0.110 m, Δy_2 = 0.080 m, Δy_3 = 0.075 m, $\Delta y_4 = \Delta y_5 = \Delta y_6 = \Delta y_7 = \Delta y_9 = \Delta y_{10}$ = 0.090 m, and Δy_{11} = 0.045 m. The pressure stations correspond to the strips as follows: pressure station VII is on Strip 1, VI on Strip 4, V on 6, IV on 8, III on 9, and finally station II is on Strip 10. Pressure station I is too close to the tip to permit a meaningful calculation.

The theoretical pressure distributions are compared to the experimental measurements in figures 13 through 24. The theoretical estimates of the five constraint parameters are: C_{L_α} = 3.207462. C_{m_α} = 0.179494, C_{L_δ} = 2.131577, C_{m_δ} = -0.463554, and C_{h_δ} = -0.057784. Three additional parameters are also of interest. These are the locations of the spanwise aerodynamic centers for angle of attack, $\bar{y}_\alpha/s$, and for flap deflection, $\bar{y}_\delta/s$, and the hinge moment coefficient for angle of attack, C_{h_α}. Their theoretical estimates are $\bar{y}_\alpha/s$ = 0.452071, $\bar{y}_\delta/s$ = 0.464614 and C_{h_α} = -0.021034.

A typical set of correction factors is shown in table I; it is for a premultiplying matrix and is based on five constraints. The factors are listed in order from leading edge to trailing edge on each strip beginning at the root; factors 1 to 10 are on Strip 1, and factors 101 to 110 are on Strip 11. The first seven factors on each strip apply to the primary surface and the last three apply to the control surface. The general trends seen in table I are a spanwise increase in factors from root to tip and a chordwise increase toward the hinge line. The minimum correction factor in table I is a 0.255873 for Box No. 10 and the maximum factor is 2.00893 for Box No. 108.

TABLE I

CORRECTION FACTORS ** PREMULTIPLIER CASE

1	0.848278E+00	0.0	2	0.863980E+00	0.0	3	0.877532E+00	0.0
4	0.885935E+00	0.0	5	0.884442E+00	0.0	6	0.863395E+00	0.0
7	0.795167E+00	0.0	8	0.351261E+00	0.0	9	0.377085E+00	0.0
10	0.255873E+00	0.0	11	0.877065E+00	0.0	12	0.894144E+00	0.0
13	0.907865E+00	0.0	14	0.916395E+00	0.0	15	0.916132E+00	0.0
16	0.899734E+00	0.0	17	0.845436E+00	0.0	18	0.431832E+00	0.0
19	0.430533E+00	0.0	20	0.299866E+00	0.0	21	0.901404E+00	0.0
22	0.920260E+00	0.0	23	0.935662E+00	0.0	24	0.946287E+00	0.0
25	0.949498E+00	0.0	26	0.939449E+00	0.0	27	0.897291E+00	0.0
28	0.510927E+00	0.0	29	0.475718E+00	0.0	30	0.330467E+00	0.0
31	0.928828E+00	0.0	32	0.950003E+00	0.0	33	0.967959E+00	0.0
34	0.981803E+00	0.0	35	0.989602E+00	0.0	36	0.986787E+00	0.0
37	0.958193E+00	0.0	38	0.607972E+00	0.0	39	0.530440E+00	0.0
40	0.366155E+00	0.0	41	0.960566E+00	0.0	42	0.984268E+00	0.0
43	0.100520E+01	0.0	44	0.102273E+01	0.0	45	0.103556E+01	0.0
46	0.104067E+01	0.0	47	0.102777E+01	0.0	48	0.721815E+00	0.0
49	0.591196E+00	0.0	50	0.400920E+00	0.0	51	0.993722E+00	0.0
52	0.101983E+01	0.0	53	0.104356E+01	0.0	54	0.106452E+01	0.0
55	0.108209E+01	0.0	56	0.109509E+01	0.0	57	0.109904E+01	0.0
58	0.840227E+00	0.0	59	0.646701E+00	0.0	60	0.424058E+00	0.0
61	0.102803E+01	0.0	62	0.105634E+01	0.0	63	0.108254E+01	0.0
64	0.110650E+01	0.0	65	0.112847E+01	0.0	66	0.114958E+01	0.0
67	0.117262E+01	0.0	68	0.965580E+00	0.0	69	0.692697E+00	0.0
70	0.428853E+00	0.0	71	0.106328E+01	0.0	72	0.109351E+01	0.0
73	0.112173E+01	0.0	74	0.114814E+01	0.0	75	0.117424E+01	0.0
76	0.120438E+01	0.0	77	0.125149E+01	0.0	78	0.110761E+01	0.0
79	0.726251E+00	0.0	80	0.410725E+00	0.0	81	0.109930E+01	0.0
82	0.113103E+01	0.0	83	0.116048E+01	0.0	84	0.118864E+01	0.0
85	0.121911E+01	0.0	86	0.126123E+01	0.0	87	0.134366E+01	0.0
88	0.129296E+01	0.0	89	0.749826E+00	0.0	90	0.379618E+00	0.0
91	0.113593E+01	0.0	92	0.116805E+01	0.0	93	0.119719E+01	0.0
94	0.122676E+01	0.0	95	0.126399E+01	0.0	96	0.132572E+01	0.0
97	0.146760E+01	0.0	98	0.159211E+01	0.0	99	0.791950E+00	0.0
100	0.407107E+00	0.0	101	0.116343E+01	0.0	102	0.119185E+01	0.0
103	0.121999E+01	0.0	104	0.125326E+01	0.0	105	0.130061E+01	0.0
106	0.138609E+01	0.0	107	0.159604E+01	0.0	108	0.200893E+01	0.0
109	0.949436E+00	0.0	110	0.615306E+00	0.0			

TABLE II

Type of Correction Matrix	Number of Constraints	$\bar{y}_\alpha/s$	$\bar{y}_\delta/s$	C_{h_α}	C_{h_δ}
None	0	0.452071	0.464614	-0.021034	-0.057784
Pre-	2	0.456751	0.469814	-0.021746	-0.059959
Pre-	5	0.484939	0.522318	-0.010117	-0.031721
Post-	2	0.453096	0.465632	-0.022409	-0.057711
Post-	5	0.469176	0.491720	+0.013014	-0.031721

The modified static pressure distributions for angle of attack are shown in figures 13 and 14, and for flap deflection are in figures 15 and 16. Perusal of these figures indicates the following results. For the angle of attack loading, both the premultiplying and postmultiplying corrections move the theoretical results slightly away from the experimental data, the post-multiplier causing a little greater change. The effect of five constraints is greater than that of two. For the flap loading, both the pre- and post-multipliers based on two constraints have small effect. The corrections based on five constraints improve the correlations on the control surface but increase the discrepancies on the wing. The postmultiplier causes a much larger change and, although the data show a pressure reversal near the trailing edge, the postmultiplier exaggerates this reversal to the extent that the sign of the hinge moment is reversed. Table II shows the effects of the four correction matrices on the aerodynamic centers and hinge moments. All of the correction matrices resulted in an outboard shift of the aerodynamic centers, the largest shift coming from the 5-constraint premultiplier.

Two constraints did not improve the hinge moment prediction; and the 5-constraint postmultiplier lead to an unreasonable prediction of $C_{n\alpha}$. The effect of additional constraints based on estimates is a topic deserving further investigation.

We can anticipate similar discrepancies when the correction factors derived from static data are applied to the oscillatory cases, and, indeed, they are shown in figures 17 through 20 for the angle of attack oscillating at $k_r = 0.622$, and in figures 21 through 24 for the flap oscillating at $k_r = 0.752$. The theoretical loading for the oscillating angle of attack is not changed significantly by either the premultiplier or the postmultiplier based on two constraints and both the real and imaginary parts are affected about the same. In some regions the theory is shifted toward the data and in others the theory is moved away from the data. The effects of five constraints are more extreme. The 5-constraint premultiplier improves the correlation for the real part but only improves the agreement for the imaginary part on the control surface while diverging on the wing. The 5-constraint postmultiplier is substantially worse in correcting the real part but is no worse than the premultiplier in modifying the imaginary part. Again, the theoretical load distribution from the oscillating control surface is not changed significantly by either the 2-constraint pre- or postmultipliers. However, some improvement is noted with the 5-constraint corrections although it is only slight. As in the static case, an outboard shift in loading occurs with all correction matrices and for both modes of motion.

The above applications of correction matrices have achieved very limited, if any, success. The lack of improvement in the most elementary case, however, is rather puzzling. This was the case of the static loading at angle of attack for which the correction factors were derived using the two constraints of lift and pitching moment. The pitching moment constraint was expected to shift the theoretical chordwise center of pressure in such a manner that the predicted pressure distribution would be closer to the experimental data. Two explanations for the lack of improvement appear possible. The first is that the theoretical loading in the leading edge region differs so much from the data that it dominates the correction factor calculation and results in a distorted loading. The second possibility is that the limited number of pressure taps near the leading edge prevented an accurate evaluation of the leading edge contribution to the pitching moment. A strain gage measurement of pitching moment would have shed some light on this possible difference.

A number of options were not pursued with these data which may have shown better correlation. First, only one configuration was studied here, the swept wing with aspect ratio 3.1; as noted above, straight wings with two aspect ratios and a swept wing with another aspect ratio were also tested. Next, only the reported integrated loads were used as constraints: the two angle of attack coefficients and the three additional control surface coefficients. The three control surface coefficients were not used as constraints by themselves, nor were additional constraints used based on estimates of rolling or bending moments. The new postmultiplying matrix was also not investigated. Finally, it would have been interesting to apply complex correction factors derived from the oscillatory angle of attack data at

k_r = 0.622 to the oscillatory control surface data at k_r = 0.752; however, this would have required integration of the published oscillatory pressure data to obtain the complex constraint coefficients.

Wing With and Without Leading Edge Droop. - Trailing edge control surfaces are studied in several other sections of this report. In this section, an attempt is made to study leading edge control surfaces. Usable data for such devices is very scarce. Several references have been investigated; however, only reference 18 proved in any way useful. The leading edge device described in this reference is a wing droop of 6^0. The droop was applied to the first 19% of the wing chord along its entire span (see fig. 25). The idealization shown in this figure is for the Doublet Lattice Method (DLM). The fuselage was simplified as simply a wing extension to the centerline.

A steady case at M = 0.80 is considered and the uncorrected calculated results using the DLM for $\alpha = 4^0$, (no droop) agree very well with the experimental data (see fig. 25). Only a lateral shift in the center of pressure seems evident. Correction factors were developed for this case to correct this slight deviation in the theory. The constraints used are lift, pitching moment and bending moment coefficients. These coefficients were summed on strips outboard of the station y/(b/2) = 0.11 and are defined as:

$$C_L = \frac{L}{qA}, \quad \frac{A}{c^2_{root}} = 1.28$$

$$C_M = \frac{M}{qA\bar{c}}, \quad c/c_{root} = .815 \text{ (moment taken about } x/c_{root} = 1.0)$$

$$C_B = \frac{B}{qA\ b/2}, \quad b/2/c_{root} = 1.6 \text{ (moment taken about } x\text{-axis)}$$

For the various modes the coefficients are:

	Pitch $\alpha = 4^0$	$\alpha = 10^0$	6^0 Droop
C_L	.2282	.5356	-.0215
C_M	-.0233	-.065	-.008
C_B	.0563	0.128	-.00536

A premultiplier and a postmultiplier (new type) were tried with equally good results on the span loading. Figure 26 illustrates the effect of the correction factors on the spanwise distribution of aerodynamic center. The correction factors increased the accuracy of the aerodynamic center inboard, but decreased it outboard.

The single mode application of both pre- and postmultipliers, also produces good results for leading edge droop span loadings (fig. 27). Notice that the unmodified results are approximately half of the experimental values. The experimental data were difficult to read on the plots (open squares). Thus, the pressure distributions were integrated to produce the darkened squares. The pressure distributions themselves were difficult to integrate accurately since there were down loads at the nose and uploads near the bend in the chord, such that the total loads were small. If the correction factors possessed only a slight variation in the chordwise direction, the balance of integrated load could shift drastically as a percent of the total.

The flow field near the wing changes at approximately $\alpha = 8^0$. Here the flow is to a large extent separated from the upper outboard surface. A comparison of uncorrected theory and experimental data, for the case of $\alpha = 10^0$, no wing droop, in figure 28, shows a loss of lift outboard of the 40% semi-span. Application of both pre- and postmultipliers (New), using C_L, C_M and C_B (bending moment at the centerline), show a much improved prediction of span loading. Also, shown in this figure is the application of the premultiplier correction factors, obtained at $\alpha = 4^0$ to the $\alpha = 10^0$ case (diamond symbols). The span loads are improved which shows that data obtained at one angle of attack can be profitably applied to other angles of attack. The corrections generated at $\alpha = 4^0$ are not as large as those generated at

$\alpha = 10^0$ because flow separation exist in the latter case. However, both corrections are in the same direction. Therefore, application of correction factors for $\alpha = 4^0$ improves the results for the 10^0 case. In general, the reverse may not hold; i.e., the correction factors obtained at $\alpha = 10^0$ (or larger angles) may be too large and an excessive correction may result leaving the corrected data further from the experimental data than there were originally. It does seem safe, however, to apply correction factors obtained at one angle of attack to other nearby angles if the flow is qualitatively similar (e.g., no great changes in flow pattern).

Transonic Cases

In this section applications of the correction factor technique are made to the same cases considered in the "Local Mach Number Studies" section. Specifically a two-dimensional symmetric airfoil (NACA 65A006) with an oscillating 25% chord flap is used.

Figure 29 illustrates the difference in results obtained when the classic theory (subsonic compressible) and the new transonic theory (Present Method (M_∞)) are corrected. A premultiplying set of correction factors were obtained using three constraints; lift, moment (1/4 chord) and control surface hinge-moment (3/4 chord). Each theory was corrected to the proper experimental constraints, i.e.,

$$c_\ell = 4.93 \qquad M_\infty = .875$$

$$c_{m_{1/4}} = -1.57 \qquad k_r = 0.0$$

$$c_{h_{3/4}} = -0.053$$

where the characteristic length is the chord and the downwash over the control surface is unity. The classic theory does not have the bulge in pressure, near the compression (or shock) region for the steady flow as does the experimental data and applying correction factors will not make it appear. Thus correction factors can not make a qualitative feature appear where none existed before. The corrected classic theory does not compare well with the experimental data and the correction factors themselves, $(1 + \varepsilon)$, show fairly large deviations from unity especially near the leading and trailing edges.

The Present Method (M_∞) however possesses a qualitative similarity with the experimental data and thus it is a better candidate for correction. Figure 29 shows such a correction. The bulge in pressure as calculated by the present method is amplified as it should be. The loading on the flap however is reduced, again as it should be, however the shape of the flap load is distorted. The correction factors themselves are better behaved for the

Present Method, (M_∞), showing large deviations from unity only on the flap surface.

Figure 30 presents the results of applying the premultiplying correction factors, obtained for the steady case, to an unsteady case. That is, the correction factors shown at the bottom of figure 29 are applied to the oscillatory results of the Present Method (M_∞) and the classic theory ($k_r = \omega\bar{c}/2U_\infty = 0.059$). Since the correction factors are real they do not effect the phase angles of the pressures but only the amplitudes, $|\Delta C_p|$. Also shown in the figure is a pressure distribution corrected using factors based on the complex lift moment (1/4 chord) and hinge moment obtained for the unsteady case. Specifically

$$c_\ell = 3.5 - i\,1.18 \qquad M_\infty = 0.875$$

$$c_{m_{1/4}} = -1.66 + i\,0.07 \qquad k_r = 0.059$$

$$c_{h_{3/4}} = -0.057 - i\,0.016$$

The correction factors obtained in this manner produce pressures that are close to those obtained using the steady correction factors (except near the flap) even though the constraints in lift are considerably different in the two cases. There is one slight anomaly in the phase angle for the complex constraint case ($k_r = 0.059$) and it exists on the last two pressure points on the flap. The phase angle there is quite large however these angles do not have a large effect since the amplitude of pressures is very small there.

The question arises; to what extent can static correction factors be applied with accuracy to the oscillatory case? Figures 31 and 32 illustrate the effect of static correction factors on lift, moment and hinge moment coefficient versus reduced frequency for a Mach Number of 0.85. Considering first the lift coefficient it is noticed that the accuracy of the imaginary part is increased up to $k_r = 0.2$. Beyond this point application of correction factors decreases the accuracy of the theory. For the real part of the lift the corrected theory is more accurate only below a reduced frequency of 0.06. For the pitching moment and hinge moment the cross over point is roughly

$k_r = 0.1$. On the average then the static correction factors are useful up to about a reduced frequency of 0.1. Beyond this point it is better to use the original theory.

It is probably true that the accuracy of extrapolating correction factors versus reduced frequency depends on Mach Number. Figure 33 gives an indication that as Mach Number is reduced the accuracy increases. Specifically, static correction factors have been applied to the oscillatory case ($k_r = 0.098$) with very good results. Both amplitude and phase angle are improved.

The theory used in figures 31, 32 and 33 is a variation of the new transonic method presented previously. Specifically the variation utilizes an average local Mach Number ($\bar{M}_\infty$) in place of the free stream Mach Number, M_∞. Figures 31 and 32 show the application of two separate types of correction factors; a premultiplier type, the type used in figures 29 and 30, and a postmultiplier type. The postmultiplier is actually an additive viscous type of correction. It can be seen that this correction does not extrapolate to higher frequencies as well as the premultiplier type (as far as the lift coefficient is concerned). As the frequency is increased the experimental data approach the unmodified theory. One interpretation of this fact is that as the frequency is increased viscous effects are reduced.

The postmultiplying correction factor (designated as "New Post") used in figures 31 and 32 is the new postmultiplying correction factor discussed in the Theoretical Development section. The reason a new type of postmultiplier was needed is because the original one seemed to fail when control surfaces are considered. Postmultipliers correct the downwash matrix. When all downwash values are non zero, e.g., wing pitch, the method seems to work. However, when this is not the case, e.g., control surface deflections, the method fails entirely. The corrected downwash values are either large and erratic themselves or they cause large and erratic pressures due to the modified downwash.

It was hoped that the introduction of correction factor mode shapes would smooth out the corrected downwash and produce accurate results. This did not happen. Even though smooth, well behaved functions were used the results were unrealistic. Although not tried, it seems that limiting the maximum and minimum values of the correction factors probably would not help very much either.

This failure of the postmultiplying correction factors led to an interesting investigation and subsequent development of the "New Postmultiplier". The investigation consisted in finding out what downwash in the theory would produce the experimental pressure distribution. Specifically the theoretical influence coefficient matrix was multiplied by the vector of experimental pressures to produce a vector of downwash values.

Figure 34 shows the results of this type of analysis (designated as Experimental) for a steady subsonic case ($M_\infty = 0.5$). Also shown is the theoretical downwash, i.e., unity over the flap. One thing is noticed immediately, there is a change in downwash ahead of the flap even though it is theoretically zero there. This downwash change is like a negative pitch of the entire airfoil. Figure 35 shows the camber (designated $M_\infty = 0.5$) associated with the downwash given in figure 34 and indeed it is like a negative or nose down pitch. This fact suggests that an additive type of correction factor, whereby all downwash values are changed, is necessary. This resulted in the development of the "New Postmultiplier" as described in the Theoretical Development section. This name is somewhat of a misnomer since the correction factor is additive and not multiplicative although the correction factors are proportional to the theoretical pressures.

The results of applying the new postmultiplying correction factors are also shown in figure 34. Again lift, moment (c/4) and hinge moment (c3/4) coefficient were used as the constraints

$$c_\ell = 3.2$$

$$c_{m_{1/4}} = -.70$$

$$c_{h_{3/4}} = -.0528$$

The corrected values of downwash (circular symbols) agree well with the experimentally deduced downwash. The disagreements at the leading edge of the airfoil and ahead of the flap are due to the fact that downwash is a sensitive function of pressure and slight variations cause large variations in downwash. With this in mind the agreement is very good especially over the flap itself.

Applying this corrected downwash to the theory produces the results given in figure 36 for the pressure distribution. The results of the New Postmultiplier agree very well with the experimental pressures. For reference, corrections by a premultiplier are also shown and these are also very good. The uncorrected theory is also presented for reference.

At the low Mach Numbers used in the last few figures ($M_\infty = 0.5$) transonic effects are not present and any differences between theory and experiment are, in all probability, due to viscous effects. Figure 35 has shown that viscous effects modify not only the downwash over the flap but also over the forward part of the airfoil as well. This comes about due to the fact that the deflected flap causes an induced upwash over the forward portion of the airfoil which in turn generates a difference in boundary layer displacement thickness on the upper and lower surfaces. This difference in displacement thicknesses causes an effective nose down pitch.

It stands to reason that the correction factors generated at $M_\infty = 0.5$ could be used to increase the accuracy of the theory at all Mach Numbers since viscous effects are present at all Mach Numbers. Figure 35 shows the effective cambers at $M_\infty = 0.5$ and $M_\infty = 0.875$ using the ($\bar{M}_\infty$) variation of the new transonic theory. Notice that the transonic camber can be thought of as composed of two pieces; one viscous piece very similar to that found at $M_\infty = 0.5$ and one transonic piece with the shape of a bump. This indicates that the accuracy of the corrected camber (or downwash) at transonic Mach Numbers can be increased if the subsonic ($M_\infty = 0.5$) results are known and used since it represents one part of the correction.

Figures 37, 38, 39 and 40 give examples of applying correction factors obtained at $M_\infty = 0.5$ to other Mach Numbers for both pressures and aerodynamic coefficients. Specifically figures 37 and 38 present the results for Mach Numbers of 0.85 and 0.875 respectively. Up to three separate corrected pressure distributions are shown in each figure. One is the result of applying a premultiplying correction factor matrix to the (M_∞) variation of the new transonic method. A second is the result of applying a new postmultiplier to the same theory; and third is the result of applying a new postmultiplier to the ($\bar{M}_\infty$) variation of the new transonic method. The last pressure distribution is seen to be the most accurate and a definite improvement over the unmodified ($\bar{M}_\infty$) theory (see fig. 4).

Figures 39 and 40 give a clear picture of the effect of applying corrections obtained at $M_\infty = 0.5$ to other Mach Numbers. Figure 39 presents the lift coefficients associated with corrected and uncorrected pressure distributions. Two types of corrections are used; both pre- and postmultiplier (New). The theory used is the ($\bar{M}_\infty$) variation of the new transonic method. Figure 40 presents similar results for the pitching moment and hinge moment coefficients. The corrections developed at $M_\infty = 0.5$ greatly improve the theory as far as the lift coefficent is concerned. The pitching moment is not changed much because it was very close to the data to begin with. The hinge moment also is not changed much.

Figures 39 and 40 show that corrections obtained at low Mach Numbers can be applied to the theories to improve accuracy at higher Mach Numbers.

Figure 41 presents the results of correcting the theory with both a postmultiplier (New) and a premultiplier. First the theory is corrected using a new postmultiplier obtained at $M_\infty = 0.5$. This represents a viscous type of correction. A premultiplier is then applied to the previously corrected results to account for transonic effects. This process produces a pressure distribution that approaches the data more closely than any of the others when it is combined with the ($\bar{M}_\infty$) variation of the new transonic method.

Figure 42 presents typical correction factors for the steady two-dimensional cases considered in this section. The theory used is the Present Method (M_∞). There is a greater change in premultiplying correction factors between $M_\infty = 0.85$ and 0.875 than there is between $M_\infty = 0.5$ and 0.85.

Supersonic Case

The arrow wing, shown in figure 43, has been chosen to illustrate the application of the correction factor technique to the supersonic case. The Douglas Supersonic Doublet Method (SDM) (ref. 20) has been used to determine the theoretical loads. The box idealization used is also shown in the figure. Notice that the tip of the wing has been clipped to reduce the number of boxes. Two modes are considered; (1) pitch ($\alpha = 4^0$) and (2) camber. The wing is operating at a Mach Number of 2.05 and a reduced frequency of zero.

Figure 43 presents a comparison of uncorrected theory (dotted line), corrected theory and experimental data. The experimental values lie below the theoretical (uncorrected) values over the entire span.

Four different methods of correcting the theory were tried. A pre- and postmultiplier (New) were applied using the pitch mode only and the results are very encouraging. The only real difference between the theory, corrected in this manner, and the experimental data appears at the wing tip. A multiple mode case was tried using the pitch and camber modes and the results are good but not as good as the previous two corrections. The fourth method is the application, to the pitch case, of a premultiplier correction factor matrix that was derived for the camber case. Thus a correction factor derived for one mode (camber) is applied to another mode (pitch). The results are not very accurate on the inboard part of the wing but agree as well as the other methods on the outboard part. The constraints used are summarized as follows:

	Pitch $\alpha = 4^0$	Camber
C_L	0.1213	0.1297
C_M	-0.035	-0.022
C_B	0.025	0.02196

$C_L = \dfrac{L}{qA}$ $\qquad \dfrac{A}{c_{root}^2} = 0.28$

$C_m = \dfrac{M}{qA\bar{c}}$ $\qquad \bar{c}/c_{root} = 0.665$

Moment about $x/c_{root} = 0.68$

$$C_B = \frac{B}{qAb/2} \qquad (b/2)/c_{root} = 0.564$$

$$B \equiv \text{Moment about x-axis}$$

A similar set of corrections were applied to the camber case and the results are shown in figure 44. Specifically pre- and postmultiplying correction factors were obtained using the camber mode. In addition a multiple mode (pitch, camber) premultiplying correction factor matrix was derived using six constraints; i.e., C_L, C_M, C_B for both modes. All three of these corrections give approximately the same good results except right at the wing tip.

Two other types of correction factors are applied to the theory and these refer to applying the correction factors derived for pitch to the camber mode. On the inboard portion of the wing these correction factors over-correct the theory, but are accurate on the outboard portion of the wing. On the inboard portions of the wing the correction factors move the corrected theory further from the data than it was originally in its uncorrected state.

Figure 45 illustrates how correction factors modify the pressure distribution over the wing (in pitch) using a premultiplier. It seems that the reduction in lift due to the correction factors is taken out at the trailing edge rather than the leading edge as the experimental pressures would indicate. This fact might be explained if the experimental pitching moment were inaccurate.

The postmultiplier does not directly modify the pressures but modifies the downwash. Modified downwash can be expressed in terms of modified camber. It is of interest to know how the postmultiplier (New) modifies the wing camber and figure 46 presents such a modification. The camber is reduced, by the correction factors, over most of the wing just as expected, since the boundary layer and separation regions act to reduce the effective wing camber. The postmultiplier, then, acts in a way that is consistent with physical processes.

RECOMMENDATIONS FOR DATA ACQUISITION

Most of the data utilized in this study were pressure data, and the correction factors were derived using constraints that were obtained in some instances from integrations of the pressures. Certain errors are associated with integrations of pressures to obtain generalized forces, arising primarily from the limited number of pressure pickup points on a practical model. The forces and moments should be measured directly in addition to the pressures. Control surface hinge moments should be measured and so should rolling moments, i.e., root bending moments, because of the importance of the spanwise aerodynamic center location. For swept surfaces it would be desirable to measure not only pitching and rolling moments in the streamwise coordinate system but also root bending moment and torque about some swept coordinate system, e.g., the 25% or 50% chord lines.

Two significant deficiencies were observed in available experimental data besides the absence of combined pressure and force data. One was a lack of any systematic variation in reduced frequency in covering the range from steady flow to high frequency, i.e., k_r of order unity. The correction factors are frequency dependent, and it is not reliable to use factors derived from low frequency data to predict pressure distributions at high frequencies. The second deficient area is the effect of Reynolds number. An important source of discrepancy between theory and test is the neglect of viscosity in the theory. When extensions of oscillatory lifting surface theory are made to account for viscous effects, data will be needed to verify the accuracy of the improved theory. However, these data are also needed to determine the accuracy with which correction factors derived from data at one Reynolds number can be used to predict pressures at another Reynolds number. This is particularly important for trailing edge control surfaces.

A number of suggestions can be made for future wind tunnel tests in addition to those indicated above. Leading edge control surfaces should be tested; spoilers might also be considered. Very little data are available for these configurations. Models should be designed so that components can be tested in their principal modes of motion. Complete models usually have moveable control surfaces but a moveable fin, horizontal tail, and engine pylon should also be considered to distinguish between component loads and interference loads. More oscillatory transonic data are needed. In two-dimensions, pitch data should be measured in addition to control surface data; in three-dimensions, data on both straight and swept wings are required.

CONCLUDING REMARKS

CONCLUSIONS

The basic conclusions arrived at as a result of the calculations and correlations presented are outlined as follows:

(1) <u>One Set of Correction Factors Is Not Good For All Modes</u>

Application of correction factors, determined from one mode, to other modes has not met with much success. Specifically, correction factors obtained using a pitch mode can not be applied to pressures due to control surface deflections. The converse is also true. In addition, application of correction factors, obtained using a pitch mode, to pressures due to a camber mode (and vice versa) have not proved to be very accurate either. Bergh and Zwaan, reference 6, on the other hand have concluded that correction factors calculated using a pitch mode can be applied to a roll mode. These two modes, however, are similar. One of them has a constant angle-of-attack along the span while the other has a linearly varying angle-of-attack along the span.

Further study is required to find out what types of correction factors are required for the various modes encountered in flutter and other dynamic aeroelastic analyses. The practical method of implementing such a correction procedure also requires further study.

The fact that one set of correction factors can not be applied to all modes has certain implications for testing procedures. It may be that more rigid body types of modes will be required (e.g., pylon yaw, wing alone pitch, tail alone pitch, outer wing pitch, inner wing pitch, fuselage alone pitch, etc.) than are now considered.

A second approach to the problem of obtaining one set of correction factors for both a pitch and a control surface mode was attempted using the "multiple mode" capability of the program. This capability allows the theory to be constrained to produce the correct lift and moment coefficient, etc., for each of several modes. The resulting span loading and/or pressures were not improved for either the pitch or control surface modes.

Even within a single mode problems can occur for different amplitudes. For instance high angle-of-attack flow fields can be basically different (separated) from those at low angles-of-attack. However correction factors obtained at low angles-of-attack can result in improved predictions for all angles-of-attack. The basic reason being that the viscous corrections, although smaller for the unseparated case, are still in the same direction as that for the separated case.

(2) A Bending Moment Constraint Is Needed For Swept Wings

Lift and pitching moment constraints are not enough for the swept wing case. A bending moment constraint is also required so that the loading is not shifted outboard to accommodate an aft shift in aerodynamic center of pressure. Without the constraint on the bending moment the correction factors will cause the wing loading to be moved toward the wing tip instead of moving the load aft along the chordline.

(3) Correction Factors Can Be Extrapolated More Accurately To Other Mach Numbers Than To Other Frequencies

When correction factors are determined at low Mach Numbers they are caused primarily by viscous effects. Since viscous effects exist at all Mach Numbers an increase in accuracy will result if the low Mach Number correction factors are applied to the high Mach Number cases.

Extrapolation in frequency has not been as successful as extrapolation in Mach Number. For the two dimensional case studied it seems that extrapolation further than $\Delta k_r = 0.1$ (based on the half chord and a Mach Number of 0.85) will lead to a decrease in accuracy as k_r is increased. It is believed that extrapolation in reduced frequency is more accurate at lower Mach Numbers.

It appears that as the frequency is increased the viscous effects are reduced. This is an important fact and if steady wind tunnel results are to be used for correcting data then a good estimate of the reduction in the viscous effect must be known. One way to accomplish this without testing every configuration is to test a representative sample of configurations over a range of frequencies and construct general trends to be used in conjunction with steady data to estimate the frequency effect on correction factors.

(4) Qualitative Features Missing From The Theory Can Not Be Generated By Correction Factors

Correction factors tend to produce quantitative changes to the theory and not qualitative ones. For instance, in transonic flow, the bulge in pressure at the shock location cannot be induced with correction factors if one did not exist in the basic theory.

(5) Downwash Correction Factors Must Be Additive, Not Multiplicative

It was found that postmultiplying correction factors did not work for control surface modes (they did work for pitch modes however). That is, scaling the downwash to reproduce the imposed experimental constraints (C_L, C_M, etc.) led to unusable results. Smoothing of the results was obtained by introducing correction factor modes; however, the levels of correction were still unrealistic. An analysis was performed to see what downwash was required to produce the experimental pressures and it became evident that the downwash had to be corrected everywhere and not just on the control surface. This suggested an additive downwash correction. A new method was developed and executed successfully.

Downwash correction factors essentially reflect the physical fact that viscous effects tend to change the effective airfoil camber (and thus the downwash). The camber is changed over the entire airfoil.

(6) Premultiplying And (New) Postmultiplying Correction Factors Are Equally Accurate

In the cases studied the accuracy of the corrected theory is improved equally well (approximately) by either type of correction factor matrix. The downwash correcting factors (New Postmultipliers) are physically more meaningful if interpreted as viscous corrections while premultipliers are more meaningfully intepreted as compressbility corrections.

Since transonic experimental data reflect both viscous and compressibility effects a very accurate way to obtain correction factors, if data permit, is to combine pre- and postmultipliers together. First a postmultiplier (New) is developed at low Mach Number where viscous effects dominate. This correction is then applied to the transonic case. The modified theory is then corrected further for transonic effects using a premultiplier. Correction factors produced in this way are more accurate than most.

(7) <u>New Transonic Method Useful But Requires Further Investigation</u>

Various methods of applying local Mach Number were tried. Simple procedures based on the substitution of the local steady Mach Number (or some average between surface and free stream) for the freestream value in the boundary conditions, kernel, and pressure equations have been tried. The results have only shown minor changes and have not even given qualitatively good results.

A new method was developed at Douglas (under the McDonnell Douglas IRAD program) and is based on a transformation of the distance between sending and receiving points based on acoustic travel time between the two points. This method was implemented for the two-dimensional case and correlated in the present study. The results are encouraging since the predicted pressures are qualitatively similar to the experimental data. That is, the new method predicts a bump in the pressure which is centered at the shock wave location and predicts a lowering of the pressure forward of the shock wave. This bump, however, is forward of the experimentally observed bump, and is usually smaller in amplitude.

The theoretically determined phase angles of the pressures are not in good agreement with the experimental data forward of the 40 percent point (for the case of a control surface rotation). One possible reason for this is the fact that the wave fronts emanating from points on the flap tend to move up and over the shock wave and arrive at the forward portions of the airfoil in nearly a horizontal configuration. On the other hand, in the theory, the paths of the wave fronts are assumed to be normal to the free stream flow with the wave fronts vertical, and this difference causes the phase angles to be greater for the theory than for the data. Further investigation of this discrepancy and its solution is required. It may be possible to use a more exact phase lag time computation in the theory, such as the one used by Tijdeman and Bergh in reference 11.

When trying to decide which theory is best it is important to account for viscous effects. Without such a correction the new transonic method, designated as (M_∞), seems best. However, when viscous effects are accounted for, the variation designated (M_∞) is best.

With the current method of computation the new transonic method is probably not reliable past $M_\infty = 0.90$. This may not be too restrictive since most wings are swept which reduces the normal Mach Numbers to values lower than 0.90.

(8) Overview of Conclusions

The concept of correcting theoretical pressure or load distributions so that they reflect associated experimental data works well with the correction factor technique, especially if the proper experimental data are available (e.g. bending moments.) It was hoped however that a set of correction factors, once developed, would be applicable to a wide variety of other cases. The range of applicability however has not been as wide as hoped for. Success in extrapolating correction factors was obtained for Mach Number and to a limited extent for frequency. Attempts to apply correction factors to dissimilar mode shapes however has not met with much success. Therefore more than one set of correction factors is required. The use of several sets of correction factors to correct oscillatory aerodynamic generalized forces for use in dynamic aeroelastic analyses requires further investigation.

Also concluded from the present analysis is that correction factors can not change the character of the load distribution. If a fundamental feature is missing from the theoretical loading then the correction factors will not make it appear. Thus theoretical methods must possess at least qualitative accuracy.

Recommendations for Further Studies

Further studies may be profitably pursued in several areas. The successes and failures of the correction factor technique, presented here, furnish a guide to such studies.

First, it seems advisable to exercise more of the various options in the present method and include more types of force data (integrated from pressure data) for some of the cases treated in this report. For instance such a case would be the Hertrich wing (refs. 14, 15). It would also be desirable to obtain new data similar to that obtained by Hertrich, in which both force and pressure data are available. It would be interesting to compare force data with integrated pressure data.

Second, it is now clear that one set of correction factors is not sufficient for all deflection modes. Thus a method for including multiple sets of correction factors into the determination of generalized oscillatory aerodynamic forces for various modes is required. This method may require special testing procedures whereby each major component or subcomponent is systematically given a rigid body rotation.

Third, the studies on viscous effects initiated in this report should be continued. Specifically the technique of determining theoretical camber lines that reproduce experimental pressure distributions seems valuable and could lead to a semiempirical method for viscous effects when combined with boundary layer theory.

Fourth, the new transonic method illustrated in this report should be refined and extended. Initially the two-dimensional capability should be refined in the areas of; 1) pressure phase angle and, 2) unsteady shock wave motion. Subsequent to this a three-dimensional method should be developed.

In addition, an investigation should be undertaken to explore the possibility of developing a semiempirical transonic method. The local steady Mach Number can be used as an adjustable parameter so as to produce the pressure distribution changes necessary to satisfy experimental constraints (lift moment etc.).

CORRECTION FACTOR COMPUTER PROGRAM

Introduction

As described in the Theoretical Development Section this method generates a set of correction factors that can be applied to a set of data (e.g., theoretical pressure) such that the data satisfies certain imposed (e.g., experimental) constraints.

For convenience this data will be referred to as pressure data since this is the most common application. However the correction factor procedure is not restricted to pressures and can be applied to other data sets (e.g., span loads, etc.).

For this procedure it is assumed that one or more theoretical pressure distributions, ΔC_{p_j}, (j = 1, number of pressure modes) are input. Associated with these pressures are: an area distribution, ΔA, a set of coordinates, (x, y, z), and a dihedral angle distribution $\bar{\gamma}$ which are input via cards, tape or both. As an option the aerodynamic influence coefficient matrix, $[A] = [D]^{-1}$, along with one or more normalwash distributions, w_j, can be input in place of ΔC_{p_j}; and as a matter of fact these must be input for postmultiplier correction matrices (i.e., correction factor matrices for the normalwash).

Constraint data (experimental data) are input as force or moment coefficients. If a force coefficient, C_e, is considered it is defined as

$$C_e = \frac{1}{\tilde{c}} \sum_{a}^{b} \Delta A \, \Delta C_p \, \vec{n} \cdot \vec{i}_a \quad \text{(force coef.)} \tag{87}$$

where $\tilde{c}$ is a constant used to convert the dimensional sum into a coefficient form. For example if $C_e = C_L$ then $\tilde{c}$ is equal to the reference area. The limits of the sum are also input to the program. The unit vector $\vec{i}_a$ is in the direction of an input axis. A set of axes are input for use in the constraining and monitoring features of the program. Each axis can be input in one of two ways; (1) a point and a direction or (2) by two points. The unit vector $\vec{i}_a$ is calculated as follows:

$$\vec{i}_a = \vec{i}\cos\alpha + \vec{j}\cos\beta + \vec{k}\cos\gamma \tag{88}$$

The unit vector $\vec{n}$ is in the direction of the lifting pressure which is given in terms of the dihedral angle of the lifting surface.

$$\vec{n} = \vec{j}(-\sin\bar{\gamma}) + \vec{k}(\cos\bar{\gamma}) \tag{89}$$

where $\vec{j}$ and $\vec{k}$ are unit vectors in the y and z directions respectively and where a right handed system is employed where z is up, y is out the starboard wing and x is aft.

If a moment coefficient, C_e, is considered then it is defined as follows:

$$C_e = \frac{1}{\tilde{c}} \sum_a^b \Delta A\, C_p\, (\vec{r} \times \vec{n}) \cdot \vec{i}_a \quad \text{(moment coef.)} \tag{90}$$

where

$$\vec{r} = (x - \xi^{(1)})\vec{i} + (y - \eta^{(1)})\vec{j} + (z - \zeta^{(1)})\vec{k} \tag{91}$$

and where $\xi^{(1)}$, $\eta^{(1)}$, $\zeta^{(1)}$ are the coordinates of the first end point of the axis considered and where $\vec{i}_a$ is its direction. The constant $\tilde{c}$ for the case of C_M has the dimensions of volume.

The program has various other capabilities and one of these is its ability to monitor the corrected or uncorrected pressures. The integrations performed in equations (87) and (90) can be performed using data without reference to constraints. Thus if span loads are desired for data that has been corrected (or uncorrected) then the proper summations are activated in the program in a manner similar to that for constraining the data.

The program also has the capability to use correction factor modes. That is, the actual correction factors $\{\varepsilon\}$ are related to a set of modal coordinates $\{\bar{\varepsilon}\}$ as follows:

$$\{\varepsilon\} = [\phi]\{\bar{\varepsilon}\} \tag{92}$$

The modal matrix, ϕ, is either input directly by cards or certain built in modes can be activated.

The program has the capability to limit the excursion of any or all correction factors. The upper and lower bounds are simply input for the correction factors that are to be limited. If correction factor modes are used then the limits are placed on the modal coordinates, $\bar{\varepsilon}$, and not on the correction factors themselves.

In addition to limits, a factor $\tilde{a}$ is input for each constraint to indicate its "constraining power". The term $\tilde{a}$ ranges from 0 to 1.0. If $\tilde{a}$ is 1.0 the constraint has full power and is 100 percent effective as a constraint. If $\tilde{a}$ is 0 then the constraining power is zero and the "constraint" has no effect. For values of $\tilde{a}$ anywhere in between the constraint is said to be an estimate.

Finally, the program can be used to apply previously obtained correction factors to input pressure distributions. The program can also be used simply to monitor existing data without any constraints.

One nomenclature problem which might cause confusion is the fact that the normalwash w is called W in the program, while the correction factors W are called CF.

Program Input

The following table provides an overview of the card input data grouped according to their functions in the program. The layout of the input sheets and a detailed description of each input item are also given following the table.

Overview of Input Data

ITEMS	CARD NO.	WHEN NEEDED	COMMENTS
CONTROL DATA	1, 2, 3, 4	Always	Header card, control dimensions and control flags.
GEOMETRY AND PRESSURES	5, 6	If FLAGP=3	Geometry data is input on one card per i, i=1, NP (NP=number of pressures). Pressures are input either 6 real numbers per card (when FLAGI=1), or 3 complex numbers per card (when FLAGI=0). Repeat pressure input per pressure mode, symmetric modes first, antisymmetric modes (if any) last.
AXIS DATA	7, 8	Always	Axis data, 2 cards per i, i=1, NAXIS (NAXIS=number of input axes)
CONSTRAINT DATA	9, 10 11, 12	If NC ≠ 0	Constraint data is input in a minimum of four cards per i, i=1, NC (NC=number of constraints)
MONITOR DATA	13, 14 15, 16	If NMØN ≠ 0	Monitor data is input in a minimum of four cards per i, i=1, NMØN (NMØN= the number of monitored aerodynamic parameters)
LIMITS DATA	17	If NELIMS≠0	Minimum and maximum limit values on ε; repeat per i, i=1, NELIMS (NELIMS=number of min. and max. limiting value pairs)

ITEMS	CARD NO.	WHEN NEEDED	COMMENTS
CORRECTION FACTOR MODES	18, 19 20, 21	If NEM $\neq$ 0	Correction factor modes may be input according to two options depending on the flag TYPE (see detail description of data) either in cards 18, 19 and 21, or in cards 18, 20 and 21. Repeat per i, i = 1, NEM (NEM=number of correction factor modes)
DOWNWASH DATA	22, 23	If FLAGW=1	Downwash data is input in a minimum of two cards per mode (see detail description of data). Input symmetric modes first, antisymmetric modes (if any) last.

Computer program requires less than 200K OCTAL storage.

Input Sheets

ENGINEER ____________ PHONE ____________ DATE ____________

PUNCH IN ALL CARDS

CASE (columns 63 64 65 66 67 68)

PROGRAM NO. (columns 73 74 75 76): E 1 G C

± 1–10	± 11–20	± 21–30	± 31–40	± 41–50	± 51–60		SEQ. NO. 77–80
		HEADER				CONTRØL DATA	1
NP	NC	NEM	NELIMS	NMØN	NAXIS		2
FLAGB	FLAGP	FLAGT	FLAGW	FLAGI	IPRINT		3
NMSYM	NMASYM						4
x_i	y_i	z_i	γ_i	ΔA_i ,	$i=1,\ NP$	GEOMETRY AND PRESSURES	5
		ΔC_{p_i} ,	$j=1,\ NP$				6
IAX	IFA					AXIS DATA	7
XI1	ETA1	ZETA1	XI2/cosα	ETA2/cosβ	ZETA2/cosγ		8
JAX	IFF					CONSTRAINT DATA	9
NDI	MI	AII	CII	Re CIE	Im CIE		10

DIRECTIONS FOR KEYPUNCH — DO NOT PUNCH BLANK COLUMNS ____________ NO UNDERPUNCHES IN SIGN FIELDS.

ENGINEER ______________ PHONE ______________ DATE ______________

PUNCH IN ALL CARDS

CASE (53 64 65 66 67 68)

PROGRAM NO. (73 74 75 76) E 1 G C

± 1–10	± 11–20	± 21–30	± 31–40	± 41–50	± 51–60		SEQ. NO. 77 78 79 80
LIMI1	LIMI2					CONSTRAINT DATA	11
- 1							12
NAX	IFN					MONITØR DATA	13
NDN	MN	ANT	CNT	LABEL			14
LIMN1	LIMN2						15
- 1							16
LIMK1	LIMK2	Re $\bar{\varepsilon}_{min}$	I_m $\bar{\varepsilon}_{min}$	Re $\bar{\varepsilon}_{max}$	I_m $\bar{\varepsilon}_{max}$	ε LIMITS	17
MØDENØ	ITYPE	NL	AL	BL		CØRRECTIØN FACTOR MØDES	18
J	Re $\emptyset_j$	I_m $\emptyset_j$	J+1	Re $\emptyset_{j+1}$	I_m $\emptyset_{j+1}$		19
LIML1	LIML2						20

DIRECTIONS FOR KEYPUNCH   DO NOT PUNCH BLANK COLUMNS ______   NO UNDERPUNCHES IN SIGN FIELDS.

ENGINEER ______________ PHONE ______________ DATE ______________

PUNCH IN ALL CARDS

CASE (63 64 65 66 67 68): [| | | | |]

PROGRAM NO. (73 74 75 76): E 1 G C

± 1–10	± 11–20	± 21–30	± 31–40	± 41–50	± 51–60		SEQ. NO. 77–80
- 1						CORR. FACTØR MØDES	21
MØDE	IDELW	LIMW1	LIMW2	RealW	Im W	W (NØRMAL-WASH) DATA	22
- 1							23

DIRECTIONS FOR KEYPUNCH DO NOT PUNCH BLANK COLUMNS ______ NO UNDERPUNCHES IN SIGN FIELDS.

Description of Input Data

Control Data

These data items are required for all cases. They consist of a header, control numbers, flags, and tape (or scratch unit) numbers.

CARD	ITEM	MNEMONIC	DESCRIPTION
1	Header	HEADER	Alpha-numeric description of case in card columns 1 through 60
2	NP	NP	Number of ΔC_p elements, where ΔC_p may represent any type of quantity (NP $\leq$ 350)
2	NC	NC	Number of constraints to be applied to the ΔC_p values (NC $\leq$ 35)
2	NEMØDES	NEM	Number of correction factor modes if any (NEM $\leq$ 100)
2	NELIMS	NELIMS	Number of input cards giving the minimum and maximum values of ε (NELIMS $\leq$ 100).
2	NMONITOR	NMØN	Number of sets of monitoring data used to integrate ΔC_p into aerodynamic parameters (NMON $\leq$ 35)
2	NAXIS	NAXIS	Number of axes input for use in integrating the ΔC_p data into forces and moments for constraint and monitoring purposes (NAXIS $\leq$ 25)
3	FLAGB	FLAGB, IFB	FLAGB=0, correction matrix calculation FLAGB=1, monitor data only FLAGB=2, apply input correction factor matrices to input pressure distribution
3	FLAGP	FLAGP, IFP	FLAGP=0, geometry data and ΔC_p are input from tapes; calculate <u>pre</u>multiplying correction factors. (See Tape Description section for format) FLAGP=1, geometry data and D^{-1} (inverse aero matrix) are input from tapes, W (normalwash) input either from tape or on cards (see FLAGW below), calculate <u>post</u>-multiplying correction factors

CARD	ITEM	MNEMONIC	DESCRIPTION
			FLAGP=2, input as for FLAGP=1; but calculate pre-multiplying correction factors FLAGP=3, geometry data and ΔC_p input on cards; calculate pre-multiplying correction factors FLAGP=4, geometry data and D^{-1} input from tapes, W input either from tape or on cards; calculate modified post-multiplying correction factors
3	FLAGT	FLAGT	FLAGT=0, weights for minimization process are absolute values of forces for unit deflections FLAGT=1, weights are unity
3	FLAGW	FLAGW	FLAGW=0, normalwash matrix, W*, is input from tape, if needed FLAGW=1, normalwash matrix, W, is input on cards
3	FLAGI	FLAGI	FLAGI=0, ΔC_p values are input as complex numbers (either from tape or on cards) FLAGI=1, ΔC_p values are input as real numbers (i.e. not complex)
3	IPRINT	IPRINT	Detail print flag; IPRINT = 1, print rows of the SAI matrix, and rows of the SAN matrix (if any) IPRINT = 0, bypass printing of same

* Cap. W is normalwash in the computer program where correction factors are called CF.

CARD	ITEM	MNEMONICS	DESCRIPTION
4	TS	NMSYM	Number of symmetric pressure modes (NMSYM, NMASYM $\leq$ 10)
4	TA	NMASYM	Number of antisymmetric pressure modes
			Note that all data items in cards 2 through 4 are input as integers, right-justified* in their respective fields of ten card columns each (format I10) as shown on the input sheets.

* Right justified means input ending in the last (or right-most) card column of the field.

Geometry and Pressure Data

Data items defining the geometry of a case are usually available on tape; similarly, pressure data (if needed; see item FLAGP under Control Data) are usually input from tape. However, if this is not the case, these data items may be input from cards by specifying FLAGP=3, as shown below.

CARD	ITEM	MNEMONICS	DESCRIPTION
			The following two cards are input only if FLAGP=3.
5	x_i	X	
5	y_i	Y	x, y, z coordinates of pressure point i
5	z_i	Z	
5	$\bar{\gamma}_i$	GMA	Dihedral angle of pressure point i
5	ΔA_i	DELA	Area of box over which the pressure acts. Repeat card 5 for all points, i=1, NP
6	ΔC_p	DCP	Array of the ΔC_p values (lifting pressures) either 3 complex numbers per card (when FLAGI=0), or 6 real numbers per card (when FLAGI=1; see Control Data)
			The format used for cards 5 and 6 is 6F10.0.

Axis Data

The following data items are required for all cases. These input data are used to describe an axis in space. Axes can be described by either two endpoints or by one endpoint and a set of direction cosines. These axes are used in the integration of the pressures into force or moment coefficients. Forces are resolved in the direction of the axis, while moments are taken about the axis.

CARD	ITEM	MNEMONIC	DESCRIPTION
7	Axis number	IAX	Axis number
7	Axis type	IFA	IFA=0, axis endpoints are input IFA=1, a point and direction cosines are input
8	$\xi 1$ $\eta 1$ $\zeta 1$	XI1 ETA1 ZETA1	Axis endpoint coordinates
8	$\xi 2/\cos\alpha$ $\eta 2/\cos\beta$ $\zeta 2/\cos\gamma$	XI2 ETA2 ZETA2	Second axis endpoint coordinates if IFA=0; direction cosines if IFA=1
			Card 7 format is 6I10; card 8 format is 6F10.0

Constraint Data

The correction factors modify the theoretical values of ΔC_p by a minimum amount so that specified forces and moments are reproduced. For example, if the total lift is known experimentally, then several data items must be input specifying the actual value of the lift coefficient and describing the way the ΔC_p values are to be integrated to obtain this coefficient. The lift coefficient is then called a constraint on the theoretical data.

CARD	ITEM	MNEMONIC	DESCRIPTION
9	Axis number	JAX	Number of the axis to be used for calculating the constraint force or moment
	F-M Flag	IFF	IFF=0, the constraint, C_e, is a force in the direction of the axis; IFF=1, the constraint, C_e, is a moment about the axis (right-hand-rule). Card format is 6I10
10	δ	NDI	δ=1, symmetric pressure mode to be used; δ=-1, antisymmetric pressure mode to be used
10	Press. mode	MI	Pressure mode number to be used with constraint C_e
10	$\tilde{a}$	AIT	Constraining effectiveness of C_e; $0 \le \tilde{a} \le 1$. If $\tilde{a} = 1$, C_e is a constraint; if $\tilde{a} < 1$, C_e is only an estimate, and the resulting weighted (corrected) theory will only approximately reproduce C_e. If $\tilde{a}$=0, then C_e will not affect data.
10	$\tilde{c}$	CIT	Constant used to nondimensionalize integrated data. If C_e is a force, $\tilde{c}$ = Area; if C_e is a moment, $\tilde{c}$ = Area x length.

CARD	ITEM	MNEMONIC	DESCRIPTION
10	C_e	CIE	Experimental (or any other) constraint on the data. Card format is 2I10, 4F10.0.
11	LIMI1, LIMI2	LIMI(1) LIMI(2)	Identification of a range of ΔC_p values (or boxes) from LIMI1 to LIMI2 defining the limits of integration for the pressures. There may be as many sets of ranges input as needed. Card format is 6I10.
12	-1	-1	The number -1; end indicator for the sets of data LIMI1, LIMI2
			Cards 9 through 12 are repeated for all constraints, i.e. NC times.

Monitor Data

The following data are input only if the control data item NMØNITØR has a value different from zero. These data are used for the integration of the ΔC_p values into some meaningful parameters as a check on the effect of the correction factors on theory, whenever FLAGB = 0. Since often it is desirable to monitor the unmodified data as well, the setting FLAGB = 1 is designed to integrate the ΔC_p values into parameters without calculating the correction factors. Another monitoring option can be activated by the setting FLAGB = 2; in this case the correction factors are input from tape (FT16) saved in a previous run and the weighted ΔC_p values are integrated into parameters as specified by the monitor data.

CARD	ITEM	MNEMONIC	DESCRIPTION
13	Axis number	NAX	Axis number used in the integration of the ΔC_p values into forces and moments
13	F-M flag	IFN	IFN=0, parameter to be determined is a force IFN=1, parameter to be determined is a moment. Card format is 6I10.
14	δ	NDN	δ=1, symmetric pressure mode used δ=-1, antisymmetric pressure mode used
14	Press. mode	MN	Pressure mode number
14	$\tilde{a}$	ANT	Not used
14	$\tilde{c}$	CNT	Constant used to nondimensionalize integrated data
14	LABEL	LABEL	Alphameric identifier of the integrated parameter (ten characters long) Card format is 2I10, 2F10.0, 10A1

CARD	ITEM	MNEMONIC	DESCRIPTION
15	LIMN1, LIMN2	LIMN(1) LIMN(2)	Identification of a range of ΔC_p values defining the limits of integration for the pressures. There may be as many sets of ranges input as needed. Card format is 6I10.
16	-1	-1	The number -1; end indicator for the sets of data LIMN1, LIMN2
			Cards 13 through 16 are repeated for all parameters, i.e., NMØNITØR times

Limits Data

It is sometimes desirable to place a restriction on the range of values of ε by specifying a minimum and a maximum bound on ε. In this case the control data item NELIMS is input as the number of ε limit pairs to be supplied, which are input as shown below. Note that this input (card 17) is omitted when NELIMS = 0.

CARD	ITEM	MNEMONIC	DESCRIPTION
17	LIMε1	LIMK(1)	A range of boxes, or ΔC_p elements,
	LIMε2	LIMK(2)	over which a limit is placed on ε
17	$\bar{\varepsilon}_{min}$	EBMIN	The minimum value allowed for ε
	$\bar{\varepsilon}_{max}$	EBMAX	The maximum value allowed for ε
			Card format is 2I10, 4F10.0
			Card 17 is repeated for all sets of ranges, i.e., NELIMS times.

Correction Factor Modes Data

In many instances it is desirable to restrict the incremental correction factors $\{\varepsilon\}$ to a linear combination of a set of modes, $\{\varepsilon\} = [\phi]\ \{\varepsilon_g\}$. The mode shapes $[\phi]$ can be input directly per box, and per mode, or the mode shapes may be selected from a set of functions. This set of data is input only if the control data item NEM is different from zero.

CARD	ITEM	MNEMONIC	DESCRIPTION
18	ε MODE NUMBER	MØDENØ	Weight factor mode number
18	TYPE	ITYPE	TYPE=1, use $(x-a)^n$ TYPE=2, use $(y-a)^n$ TYPE=3, use $(z-a)^n$ TYPE=4, use $\exp[b(x-a)^n]$ TYPE=5, use $\exp\ [b(y-a)^n]$ TYPE=6, use $\exp\ [b(z-a)^n]$ as mode equation If TYPE = 0, the ϕ values are input in card 19, and the following 3 items are not used
18	n	NL	Constants used in the mode equation
18	a	AL	Card format is 2I10, 4F10.0
18	b	BL	
19	J	J	Card 19 is input only if TYPE = 0. Box (or element) number for which the $\phi(J)$ applies
19	$\phi(J)$	PHI(J)	The modal value of the J-th value of ε
19	J + 1	J + 1	Another set of ϕ-data
19	$\phi(J+1)$	PHI(J+1)	Card format is 2(I10, 2F10.0). Repeat as needed, 2 sets of data per card. Note that only the non-zero elements need be input.

CARD	ITEM	MNEMONIC	DESCRIPTION
			Omit card 20 if TYPE = 0.
20	LIML1 LIML2	LIML(1) LIML(2)	Range of boxes or ε's over which the current ε-mode applies. There may be as many sets of ranges input as needed. Card format is 6I10.
21	-1	-1	The number -1; end indicator of data set for the current ε-mode (MØDENØ)
			Repeat cards 18 through 21 for all ε modes, i.e., NEM times.

Normalwash Data

If the normalwash matrix [W] is needed (see control data FLAGP), and it is not available on tape, the control flag FLAGW must be input as 1, and then the normalwash values are card input as shown below.

CARD	ITEM	MNEMONIC	DESCRIPTION
22	MØDE	MØDE(J)	Mode number for the current set of W values
22	δ	IDELW	Symmetry flag to aid in identifying the mode; note that δ=1 type values are expected to precede the δ=-1 type values
22	LIMW1 LIMW2	LIMW(1) LIMW(2)	A range of boxes over which the W value applies
22	W	WIN	Normalwash, W, for the above range of boxes. Card format is 4I10, 2F10.0. Repeat card 22 as needed. Note that only the non zero W values need be input.
23	-1	-1	The number -1; end indicator for the normalwash input data

Tape Description

Program E1GC uses a minimum of four, and a maximum of twelve tapes and/or utility (scratch) units depending on the type of the case considered. In addition NPIT = 5 and NPØT = 6 are used throughout the program as the system input/output units respectively. These, as well as all tapes and utility units are defined in subroutine WEYT by means of a DATA statement specification under their respective names. The following table gives a summary of tape names and their use; the formats of those tapes that may be specified as input/output units are described in subsequent tables.

Summary of Tape Units

NAME	UNIT	WHEN NEEDED	USER SUBROUTINES	DESCRIPTION OF CONTENTS
NUTL1	1	Always	WEYT, WSWA, SDBL, EPSJ	Miscellaneous intermediate solutions
NUTL2	2	Always	WEYT, SDBL, DCPT, CEMN	
NTSAIJ	3	If NC≠0	WEYT, SAIJ, DELC, SDBL	SAI matrix rows
NTSANJ	4	If NMØN≠0	WEYT, SAIJ, CEMN	SAN matrix rows
NTPHIJ	8	If NEM≠0	WEYT, PHIJ, SDBL, EPSJ	ϕ matrix columns
MASTSB	9	Always	WEYT, SDBL, GINV	S matrix rows
NEWTSB	10	If NELIMS ≠ 0	WEYT, MØDF, GINV	The modified S matrix rows
NTGEØM	11	If FLAGP≠3	WEYT	Geometry arrays; input tape
NTDCP	12	Always	WEYT, DCPB	ΔC_p matrix columns; either input tape or scratch unit depending on FLAGP
NTAPW	13	If FLAGP ≠ 0, 3	WEYT, WSWA	W (normalwash) columns; input tape if NTAPW = 0, scratch unit otherwise
NTAPDI	14	If FLAGP ≠ 0, 3	WEYT, DCPB, SDBL, DCPT	Inverse downwash factor matrix $[D]^{-1}$
NEWDCP	15	If FLAGI=1 and FLAGP = 0	WEYT, DCPB	Complex ΔC_p columns, when ΔC_p is input as a real matrix
NTAPCF	16	If FLAGB≠1	WEYT, DCPT	CF, the correction factor matrix; NTAPCF is output tape (or scratch unit) for FLAGB = 0 cases; NTAPCF is an input tape for FLAGB = 2 cases

Input Tape NTGEØM

RECORD	WORD	ITEM	DESCRIPTION
1	1	LENGTH	Length of arrays in records 2 through 6 (LENGTH = NP)
2	1 - NP	X	x-coordinate array
3	1 - NP	Y	y-coordinate array
4	1 - NP	Z	z-coordinate array
5	1 - NP	GMA	Dihedral angle ($\bar{\gamma}$) array
6	1 - NP	DELA	Array of box areas

Input Tape (or Scratch Unit) NTDCP

RECORD	WORD	ITEM	DESCRIPTION
1	1	NP	Row dimension of the ΔC_p matrix (column length)
	2	NSYM	Number of ΔC_p columns for symmetric modes
	3	NASYM	Number of ΔC_p columns for antisymmetric modes
2	1 - NP	DCP	ΔC_p column for first symmetric mode
⋮	⋮	⋮	⋮
1+NSYM +NASYM	1 - NP	DCP	ΔC_p column for last antisymmetric mode*

* Note that if NASYM = 0, the last ΔC_p column refers to last symmetric mode

Input Tape (or Scratch Unit) NTAPW

RECORD	WORD	ITEM	DESCRIPTION
1	1	NP	Row dimension of the W (normalwash) matrix
	2	NSYM	Numbers of W columns for symmetric modes
	3	NASYM	Number of W columns for antisymmetric modes
2	1 - NP	W	W column for first symmetric mode
⋮	⋮	⋮	⋮
1 + NSYM +NASYM	1 - NP	W	W column for last anti-symmetric mode (if any)

Input Tape NTAPDI

RECORD	WORD	ITEM	DESCRIPTION
1	1	NP	Row dimension of matrix DI
	2	NP	Column dimension of matrix DI
	3	NCØL2	NCØL2 = NP if both symmetric and antisymmetric DI matrices are on tape; NCØL2 = 0 otherwise
2	1 - NP	DI	First row of DI, the inverse downwash factor matrix, $[D^{-1}]$ for symmetry
⋮	⋮	⋮	⋮
1 + NP	1 - NP	DI	Last DI-row for symmetry
			The following records may be omitted when antisymmetric modes are not desired.
2 + NP	1 - NP	DI	First DI-row for antisymmetry
⋮	⋮	⋮	⋮
1 + 2NP	1 - NP	DI	Last DI-row for antisymmetry

Input/Output Tape NTAPCF

RECORD	WORD	ITEM	DESCRIPTION
1	1	CØDE	Alphameric identifier of tape, 4 characters in length, left justified; CØDE = PRE, for pre-multiplier cases, CØDE = PØST for post-multiplier cases
2	1	LENGTH	Length of array CF. LENGTH should be equal to NP.
	2	NMSYM	Number of symmetric modes for case
	3	NMASYM	Number of antisymmetric modes for case
			Note that the last two items are not used when tape NTAPCF is an input tape
3	1 - NP	CF	Array of the complex correction factors

Test Cases. - The use of the program will be illustrated by two test cases. The first will be a premultiplier and will exercise most features of the program so that their use can be illustrated. The second test case will illustrate the use of the new postmultiplier, tape input, and downwash input on cards.

The theoretical pressures are taken from a two-dimensional analysis of an airfoil with a 25% chord flap. The new transonic procedure discussed previously will be used for the airfoil operating at a Mach Number of 0.875 and a reduced frequency of 0.0. Figure 47 illustrates the geometry, pressures and axes data for the airfoil for control surface rotation (Mode 1) and pitch (Mode 2). Also shown on the figure are the theoretical and experimental values of c_ℓ, $c_{m1/4}$ and $c_{h1/4}$ for mode 1 and c_ℓ for mode 2. The experimental values are used as the constraints. An experimental value for c_ℓ for mode 2 is not available thus an estimate is given in its place in the figure.

Test Case 1. - Table III presents the input cards for the first test case. The number of pressures, NP, is 19; the number of constraints, NC, is 4. For this case 19 correction factor modes, ϕ, will be used, thus NEM = 19. In addition limits will be placed on the values of $\bar{\varepsilon}$. These limits will be described by one card thus NELIMS = 1. The number of axes, NAXIS, is 3. The program is able to monitor the corrected data, and in this test case the number of coefficients to be monitored, NMON, is 4 and they are c_ℓ, $c_{m1/4}$ and $c_{h3/4}$ for mode 1 and c_ℓ for mode 2. Thus the monitored coefficients should reproduce the input constraints. This, in fact, is the case as the output shows in Table IV.

Since correction factors are to be calculated rather than data monitored only, FLAGB = 0. Also since the geometrical data and pressure data are to be card input and a premultiplier is to be calculated FLAGP = 3. The usual weight factor T, (the absolute value of the force on an element) is not used, thus FLAGT = 1. Normalwash values are not input thus FLAGW = 0. Only real values of pressure are used thus FLAGI = 1; the detail print flag is input as IPRINT = 1. In this example there are two modes (call them symmetric) thus

NMSYM = 2 and NMASYM = 0. This marks the end of the control data.

The geometry data are taken from figure 47 and are given on cards designated as type 5. The 1/4-chord point of each box is input along with its area, $\Delta A = \Delta X$. The pressures at each 1/4-chord point of each box and for each mode are taken from figure 47 and are given on cards designated as type 6.

The axis data are encountered next. IAX identifies the axis number and IFA identifies how it is input (whether by two points or a point and a direction). In this case a point $(\xi^{(1)}, \eta^{(1)}, \zeta^{(1)})$ and direction $(\cos\alpha, \cos\beta, \cos\gamma)$ are input thus IFA = 1. These points and directions are taken from figure 47 and are input on card designated as type 8.

The constraint data is next. Input are four constraints c_ℓ, $c_{m1/4}$ and $c_{h3/4}$ for mode 1 and c_ℓ for mode 2 taken from the experimental values of these parameters given on figure 47. Each constraint has a 9 and 10 type card. JAX identifies the axis to be used with the constraint (axis 1 for c_ℓ, axis 2 for $c_{m1/4}$ and axis 3 for $c_{h3/4}$). The flag IFA identifies the coefficient type to be calculated whether the force type (IFA = 0) or moment type (IFA = 1). The terms MI and NDI denote the mode to be used. In this case modes 1 and 2 are symmetric. The constraining power AIT is taken as 1.0 for the constraints of mode 1 to ensure a full constraint. However AIT for c_ℓ of mode 2 is taken as .95 since this is an estimate. The nondimensionalizing constant CIT is the chord for the c_ℓ constraint and the chord squared for $c_{m1/4}$ and $c_{h3/4}$. The limits of integration LIMI1, LIMI2 span the entire surface for c_ℓ and $c_{m1/4}$ (from box 1 to box 19) but only range over the control surface (box 13 to box 19) for $c_{h3/4}$.

The monitor data found on card types 13, 14, 15, 16 are almost identical to that of the constraint data because in this case c_ℓ, $c_{m1/4}$ and $c_{h3/4}$ are the parameters to be monitored. Of course they could be any quantity or for that matter no quantities if monitoring is not desired. The only real difference between monitor data and constraint data is that an alpha-numeric identifier is input in place of the constraints for the monitor data.

As an example of the use of limiting values on $\bar{\varepsilon}$, card type 17 is input for this test case. Specifically it is required that

$$-0.7 \leq \varepsilon \leq 1.5$$

hold for all values of $\bar{\varepsilon}$, 1 through 19 (LIMK1 = 1, LIMK2 = 19).

As a simple example of the use of correction factor mode shapes, ϕ, an identity matrix will be used;

$$[\phi] = \lceil I \rfloor \quad (\text{ITYPE} = 0)$$

Card types 19 and 21 are used to input these modes.

The program output for this case is given in Table IV. The printed output, which fits on 8 1/2 x 11 sheets, contains most of the input. Integration matrices are then printed along with other intermediate steps in the process of solution. At the end of the printout a summary of the geometry data, incremental correction factors, ε, and modified pressures are printed. Next are the correction factors $\varepsilon + 1$ and finally the aerodynamic parameters, calculated using the modified pressures, that have been monitored by the program.

Test Case 2. - Table V presents the input sheets for the second test case. This test case is the same as the test case 1 with the following exceptions: (1) the geometry and $[D]^{-1}$ are input from tape; (2) a new postmultiplier is developed; (3) one mode is used with three constraints; (4) correction factor modes are not used and (5) limits on incremental correction factors, ε, are not imposed.

For this case changes from test case 1 occur in the control data (cards 1 through 4). First, no correction factor modes (NEM = 0) are to be used. Second, the card giving limits on ε is omitted, thus NELIMS = 0. Third, $[D]^{-1}$ and the geometry are to be read in on tapes and the new postmultiplying correction factor is desired, thus FLAGP = 4. In this case normalwash values are to be card read and so FLAGW = 1. Also only one mode is to be used (control surface rotation), thus NMSYM = 1. The geometry data remains the same as in test case 1.

Finally the normalwash is input on card type 22. The mode is a control surface rotation, thus W^* = 1.0 over boxes 13 through 19, i.e., LIMW1 = 13, LIMW2 = 19. The program output is given in Table VI.

---

* Remember W is downwash in this computer program

TABLE III

INPUT DATA - TEST CASE I

```
TEST CASE NO. 1  PRE-MULT.,  CARD INPUT, M=.875, K=0       CONTROL DATA     1
           19        4        19        1        4        3                  2
            0        3         1        0        1        1                  3
            2        0                                                       4
-.99358   0.0      0.0       0.0      0.0256               GEOMETRY DATA    5
-0.94291  0.0      0.0       0.0      0.075
-0.84502  0.0      0.0       0.0      0.119
-0.70657  0.0      0.0       0.0       .155
-0.537    0.0      0.0       0.0      0.181
-0.3479   0.0      0.0       0.0      0.194
 -0.1521  0.0      0.0       0.0      0.194
0.03701   0.0      0.0       0.0      0.181
 0.20657  0.0      0.0       0.0      0.155
0.34501   0.0      0.0       0.0       0.119
0.44291   0.0      0.0       0.0      0.075
0.49358   0.0      0.0       0.0      0.0256
0.50627   0.0      0.0       0.0       0.0248
 0.5545   0.0      0.0       0.0       0.0694
0.64153   0.0      0.0       0.0       0.1
0.75      0.0      0.0       0.0      0.111
0.8585    0.0      0.0       0.0       0.1
 0.9455   0.0      0.0       0.0       0.0694
0.99373   0.0      0.0       0.0       0.0248
7.01      2.45     1.61      1.28     1.14     1.267       PRESSURE DATA    6
2.484      9.69    8.94      10.24    12.65    17.0
17.88     12.5     8.42      5.6      3.41     1.79
.5614
71.7527   24.5187  15.4279   11.2879  8.77386  7.56652     PRESSURE DATA    6
8.20533   14.2879  10.5817   8.97737  7.92188  6.79906
7.03546   6.31302  5.50708   4.08903  2.72631  1.52178
.433745
          1        1                                       AXIS DATA        7
0.0       0.0      0.0       0.0      0.0      1.0                          8
          2        1
-0.5      0.0      0.0       0.0      1.0      0.0
          3        1
0.5       0.0      0.0       0.0      1.0      0.0         CONSTRAINT DATA  9
          1        0                                                        10
          1        1 1.0     2.0      4.93     0.0                          11
          1        19                                                       12
-1
          2        1
          1        1 1.0     4.0      -1.57    0.0
          1        19
-1
          3        1
          1        1 1.0     4.0      -.053    0.0
         13        19
-1
          1        0
          1        2 0.95    2.0      8.8      0.0
          1        19
-1
```

```
             1          0                                                      MONITOR DATA       13
             1          1 1.0          2.0        CL                                              14
             1         19                                                                         15
     -1                                                                                           16
             2          1
             1          1 1.0          4.0        CM-1/4
             1         19
     -1
             3          1
             1          1 1.0          4.0        CH-3/4
            13         19
     -1
             1          0
             1          2 1.0          2.0        CL-ALPHA
             1         19
     -1
             1         19-0.7          0.0        1.5          0.0             EPS-LIMITS         17
             1          0 0.0          0.0        0.0                          CORRECTION-        18
             1 1.0                                                             FACTOR MODES       19
     -1                                                                                           21
             2          0 0.0          0.0        0.0
             2 1.0
     -1
             3          0 0.0          0.0        0.0
             3 1.0
     -1
             4          0 0.0          0.0        0.0
             4 1.0
     -1
             5          0 0.0          0.0        0.0
             5 1.0
     -1
             6          0 0.0          0.0        0.0
             6 1.0
     -1
             7          0 0.0          0.0        0.0
             7 1.0
     -1
             8          0 0.0          0.0        0.0
             8 1.0
     -1
             9          0 0.0          0.0        0.0
             9 1.0
     -1
            10          0 0.0          0.0        0.0
            10 1.0
     -1
            11          0 0.0          0.0        0.0
            11 1.0
     -1
            12          0 0.0          0.0        0.0
            12 1.0
     -1
```

```
          13          0 0.0      0.0      0.0
          13 1.0
   -1
          14          0 0.0      0.0      0.0
          14 1.0
   -1
          15          0 0.0      0.0      0.0
          15 1.0
   -1
          16          0 0.0      0.0      0.0
          16 1.0
   -1
          17          0 0.0      0.0      0.0
          17 1.0
   -1
          18          0 0.0      0.0      0.0
          18 1.0
   -1
          19          0 0.0      0.0      0.0
          19 1.0
   -1
```

TABLE IV

ØUTPUT LISTING - TEST CASE 1

```
      TEST CASE NO. 1  PRE-MULT.,  CARD INPUT, M=.875, K=0

CONTROL FLAGS          --

FLAGB = 0   CORRECTION FACTORS CALCULATED
FLAGP = 3   PREMULTIPLIER  -  PRESSURE AND GEOMETRY TAKEN FROM CARDS
FLAGT = 1   WEIGHTS = 1.0
FLAGW = 0   NORMALWASH TAKEN FROM TAPE       (IF NEEDED)
IPRINT= 1   (DETAIL PRINT FLAG)

CONTROL DIMENSIONS   --

NP     =  19
NC     =   4
NEM    =  19
NELIMS=    1
NMON   =   4
NAXIS  =   3

LIST OF INPUT/OUTPUT TAPES  --

GEOMETRY  TAPE = 11
DELTA-CP  TAPE = 12
W         TAPE = 13
D-INVERSE TAPE = 14
CORR. FACTORS  = 16
```

CARD-READ OPTION IS SPECIFIED

I	X	Y	Z	GAMMA	DELTA-A
1	-0.993580	0.0	0.0	0.0	0.025500
2	-0.942910	0.0	0.0	0.0	0.075000
3	-0.845020	0.0	0.0	0.0	0.119000
4	-0.706570	0.0	0.0	0.0	0.155000
5	-0.537000	0.0	0.0	0.0	0.181000
6	-0.347900	0.0	0.0	0.0	0.194000
7	-0.152100	0.0	0.0	0.0	0.194000
8	0.037010	0.0	0.0	0.0	0.181000
9	0.206570	0.0	0.0	0.0	0.155000
10	0.345010	0.0	0.0	0.0	0.119000
11	0.442910	0.0	0.0	0.0	0.075000
12	0.493580	0.0	0.0	0.0	0.025500
13	0.506270	0.0	0.0	0.0	0.024800
14	0.554500	0.0	0.0	0.0	0.069400
15	0.641530	0.0	0.0	0.0	0.100000
16	0.750000	0.0	0.0	0.0	0.111000
17	0.858500	0.0	0.0	0.0	0.100000
18	0.945500	0.0	0.0	0.0	0.069400
19	0.993730	0.0	0.0	0.0	0.024800

SYMMETRIC DELTA-CP MATRIX

COLUMN 1

7.010000	0.0	2.450000	0.0	1.610000	0.0	1.280000	0.0
1.140000	0.0	1.267000	0.0	2.484000	0.0	9.690000	0.0
8.940000	0.0	10.240000	0.0	12.650000	0.0	17.000000	0.0
17.880005	0.0	12.500000	0.0	8.420000	0.0	5.600000	0.0
3.410000	0.0	1.790000	0.0	0.561400	0.0		

COLUMN 2

71.752701	0.0	24.518707	0.0	15.427900	0.0	11.287900	0.0
8.773860	0.0	7.566520	0.0	8.205330	0.0	14.287900	0.0
10.581700	0.0	8.977370	0.0	7.921880	0.0	6.799060	0.0
7.035460	0.0	6.813020	0.0	5.507080	0.0	4.089030	0.0
2.726310	0.0	1.521780	0.0	0.489745	0.0		

THE 3 SETS OF INPUT DATA FOR ALL AXES

AXISNO(R)	FLAGA(R)	XI1(R)	ETA1(R)	ZETA1(R)	XI2(R) (COSA(R))	ETA2(R) (COSB(R))	ZETA2(R) (COSG(R))
1	1	0.0	0.0	0.0	0.0	0.0	1.000000
2	1	-0.500000	0.0	0.0	0.0	1.000000	0.0
3	1	0.500000	0.0	0.0	0.0	1.000000	0.0

THE 4 SETS OF INPUT DATA FOR ALL CONSTRAINTS

AXISNO(I)	FLAGF(I)	DELTA	M	LIM1	LIM2	A-WIG(I)	C-WIG(I)	C-E(I) REAL	C-E(I) IMAG.
1	0	1	1	1	19	1.000000	0.200000E+01	4.930000	0.0
2	1	1	1	1	19	1.000000	0.400000E+01	-1.570000	0.0
3	1	1	1	13	19	1.000000	0.400000E+01	-0.053000	0.0
1	0	1	2	1	19	0.950000	0.200000E+01	8.800000	0.0

THE 4 SETS OF INPUT DATA FOR MCNITORING

AXISNO(N)	FLAGF(N)	DELTA	M	LIM1	LIM2	A-WIG(N)	C-WIG(N)	LABEL
1	0	1	1	1	19	1.000000	0.200000E+01	CL
2	1	1	1	1	19	1.000000	0.400000E+01	CM-1/4
3	1	1	1	13	19	1.000000	0.400000E+01	CH-3/4
1	0	1	2	1	19	1.000000	0.200000E+01	CL-ALPHA

SAI MATRIX ROWS

0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		
0.003159	0.0	0.008305	0.0	0.010264	0.0	0.008005	0.0
0.001674	0.0	-0.007377	0.0	-0.016873	0.0	-0.024300	0.0
-0.027380	0.0	-0.025139	0.0	-0.017680	0.0	-0.006359	0.0
-0.006239	0.0	-0.018296	0.0	-0.028538	0.0	-0.034688	0.0
-0.033962	0.0	-0.025079	0.0	-0.009261	0.0		
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.000039	0.0	-0.000946	0.0	-0.003538	0.0	-0.006938	0.0
-0.008963	0.0	-0.007729	0.0	-0.003061	0.0		
0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		

SAN MATRIX ROWS

0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		
0.003159	0.0	0.008305	0.0	0.010264	0.0	0.008005	0.0
0.001674	0.0	-0.007377	0.0	-0.016873	0.0	-0.024300	0.0
-0.027380	0.0	-0.025139	0.0	-0.017680	0.0	-0.006359	0.0
-0.006239	0.0	-0.018296	0.0	-0.028538	0.0	-0.034688	0.0
-0.033962	0.0	-0.025079	0.0	-0.009261	0.0		
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.000039	0.0	-0.000946	0.0	-0.003538	0.0	-0.006938	0.0
-0.008963	0.0	-0.007729	0.0	-0.003061	0.0		
0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		

EPSILON LIMITS --

K	LIM-1(K)	LIM-2(K)	EPSILON-BAR-MIN REAL	EPSILON-BAR-MIN IMAG.	EPSILON-PAR-MAX REAL	EPSILON-PAR-MAX IMAG.
1	1	19	-0.700000	0.0	1.500000	0.0

-PHI- MATRIX

COLUMN 1

1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 2

0.0	0.0	1.000000	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 3

0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 4

0.0	0.0	0.0	0.0	0.0	0.0	1.000000	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 5

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 6

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.000000	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 7

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 8

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.000000	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 9

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 10

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.000000	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 11

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 12

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.000000	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 13

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 14

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.000000	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 15

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

COLUMN 16

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
0.0        0.0        0.0        0.0        0.0        0.0
COLUMN 1[illegible]
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
1.000000   0.0        0.0        0.0        0.0        0.0
COLUMN 18
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        1.000000   0.0        0.0        0.0
COLUMN 19
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        1.000000   0.0
```

THE 2 COLUMNS OF THE DELTA-CP-BAR MATRIX

COLUMN 1

1	0.701000E+01	0.0	2	0.245000E+01	0.0	3	0.161000E+01	0.0
4	0.128000E+01	0.0	5	0.114000E+01	0.0	6	0.126700E+01	0.0
7	0.248400E+01	0.0	8	0.969000E+01	0.0	9	0.894000E+01	0.0
10	0.102400E+02	0.0	11	0.126500E+02	0.0	12	0.170000E+02	0.0
13	0.178800E+02	0.0	14	0.125000E+02	0.0	15	0.842000E+01	0.0
16	0.560000E+01	0.0	17	0.341000E+01	0.0	18	0.179000E+01	0.0
19	0.561400E+00	0.0						

COLUMN 2

1	0.717527E+02	0.0	2	0.245187E+02	0.0	3	0.154279E+02	0.0
4	0.112879E+02	0.0	5	0.877386E+01	0.0	6	0.756652E+01	0.0
7	0.820533E+01	0.0	8	0.142879E+02	0.0	9	0.105817E+02	0.0
10	0.897737E+01	0.0	11	0.792188E+01	0.0	12	0.679906E+01	0.0
13	0.703546E+01	0.0	14	0.681302E+01	0.0	15	0.550708E+01	0.0
16	0.408903E+01	0.0	17	0.272631E+01	0.0	18	0.152178E+01	0.0
19	0.489745E+00	0.0						

DELTA-C

-0.411495	0.0	0.420184	0.0	0.074273	0.0	-1.207082	0.0

THEORETICAL C-VALUES

5.341496	0.0	-1.990184	0.0	-0.127273	0.0	10.007082	0.0

ONE S-BAR ROW

0.089728	0.0	0.091875	0.0	0.095795	0.0	0.099200	0.0
0.103170	0.0	0.122899	0.0	0.240948	0.0	0.876945	0.0
0.692850	0.0	0.609280	0.0	0.474375	0.0	0.217600	0.0
0.221712	0.0	0.433750	0.0	0.421000	0.0	0.310800	0.0
0.170500	0.0	0.062113	0.0	0.006961	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

ONE S-BAR ROW

0.022144	0.0	0.020346	0.0	0.016526	0.0	0.010246	0.0
0.001909	0.0	-0.009346	0.0	-0.041913	0.0	-0.235464	0.0
-0.244773	0.0	-0.257424	0.0	-0.223646	0.0	-0.108101	0.0
-0.111551	0.0	-0.228695	0.0	-0.240292	0.0	-0.194250	0.0
-0.115812	0.0	-0.044892	0.0	-0.005199	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

ONE S-BAR ROW

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.000695	0.0	-0.011820	0.0	-0.029792	0.0	-0.038850	0.0
-0.030562	0.0	-0.013836	0.0	-0.001719	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0		

ONE S-BAR ROW

0.918435	0.0	0.919451	0.0	0.917960	0.0	0.874812	0.0
0.794034	0.0	0.733952	0.0	0.795917	0.0	1.293055	0.0
0.820082	0.0	0.534154	0.0	0.297070	0.0	0.087028	0.0
0.087240	0.0	0.236412	0.0	0.275354	0.0	0.226941	0.0
0.136315	0.0	0.052806	0.0	0.006073	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.052632	0.0		

THE 4 BY 23 S-DOUBLE-BAR MATRIX

ROW 1

1	0.897280E-01	0.0	2	0.918750E-01	0.0	3	0.957950E-01	0.0
4	0.992000E-01	0.0	5	0.103170E+00	0.0	6	0.122899E+00	0.0
7	0.240948E+00	0.0	8	0.876945E+00	0.0	9	0.692850E+00	0.0
10	0.609280E+00	0.0	11	0.474375E+00	0.0	12	0.217600E+00	0.0
13	0.221712E+00	0.0	14	0.433750E+00	0.0	15	0.421000E+00	0.0
16	0.310800E+00	0.0	17	0.170500E+00	0.0	18	0.621130E-01	0.0
19	0.696136E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 2

1	0.221440E-01	0.0	2	0.203462E-01	0.0	3	0.165256E-01	0.0
4	0.102459E-01	0.0	5	0.190865E-02	0.0	6	-0.934647E-02	0.0
7	-0.419129E-01	0.0	8	-0.235464E+00	0.0	9	-0.244773E+00	0.0
10	-0.257424E+00	0.0	11	-0.223646E+00	0.0	12	-0.108101E+00	0.0
13	-0.111551E+00	0.0	14	-0.228695E+00	0.0	15	-0.240292E+00	0.0
16	-0.194250E+00	0.0	17	-0.115812E+00	0.0	18	-0.448922E-01	0.0
19	-0.519919E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 3

1	0.0	0.0	2	0.0	0.0	3	0.0	0.0
4	0.0	0.0	5	0.0	0.0	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0	9	0.0	0.0
10	0.0	0.0	11	0.0	0.0	12	0.0	0.0
13	-0.695066E-03	0.0	14	-0.118197E-01	0.0	15	-0.297921E-01	0.0
16	-0.388500E-01	0.0	17	-0.305621E-01	0.0	18	-0.138357E-01	0.0
19	-0.171852E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 4

1	0.918435E+00	0.0	2	0.919451E+00	0.0	3	0.917960E+00	0.0
4	0.874812E+00	0.0	5	0.794034E+00	0.0	6	0.733952E+00	0.0
7	0.795917E+00	0.0	8	0.129305E+01	0.0	9	0.820082E+00	0.0
10	0.534154E+00	0.0	11	0.297070E+00	0.0	12	0.870280E-01	0.0
13	0.872397E-01	0.0	14	0.236412E+00	0.0	15	0.275354E+00	0.0
16	0.226941E+00	0.0	17	0.136315E+00	0.0	18	0.528058E-01	0.0
19	0.607283E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.526316E-01	0.0			

-S- MATRIX

0.255609E+01	0.0	-0.975168E+00	0.0	-0.359801E-01	0.0
0.322784E+01	0.0	-0.975168E+00	0.0	0.422131E+00	0.0
0.216555E-01	0.0	-0.886313E+00	0.0	-0.359801E-01	0.0
0.216555E-01	0.0	0.366550E-02	0.0	-0.247821E-01	0.0
0.322784E+01	0.0	-0.886313E+00	0.0	-0.247821E-01	0.0
0.804018E+01	0.0				

-DC- COLUMN

-0.411495E+00	0.0	0.420184E+00	0.0	0.742729E-01	0.0
-0.120708E+01	0.0				

SOLUTION OF MATRIX EQ.

0.109207E+02	0.0	0.230577E+02	0.0	-0.227097E+02	0.0
-0.206261E+01	0.0				

OUTPUT OF GEN. INVERSE (EPS-TILDA)

-0.403892	0.0	-0.423996	0.0	-0.466205	0.0	-0.484818	0.0
-0.467087	0.0	-0.387223	0.0	0.023238	0.0	1.480520	0.0
0.230980	0.0	-0.383590	0.0	-0.589009	0.0	-0.295733	0.0
-0.315017	0.0	-0.755526	0.0	-0.834346	0.0	-0.670626	0.0
-0.395491	0.0	-0.151506	0.0	-0.017357	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.108558	0.0		

J	EPSILON-BAR		EPSILON-LIMIT		EPSILON-LAST	
1	-0.403892	0.0	0.0	0.0	-0.403892	0.0
2	-0.423996	0.0	0.0	0.0	-0.423996	0.0
3	-0.466205	0.0	0.0	0.0	-0.466205	0.0
4	-0.484818	0.0	0.0	0.0	-0.484818	0.0
5	-0.467087	0.0	0.0	0.0	-0.467087	0.0
6	-0.387223	0.0	0.0	0.0	-0.387223	0.0
7	0.023238	0.0	0.0	0.0	0.023238	0.0
8	1.480520	0.0	0.0	0.0	1.480520	0.0
9	0.230980	0.0	0.0	0.0	0.230980	0.0
10	-0.383590	0.0	0.0	0.0	-0.383590	0.0
11	-0.589009	0.0	0.0	0.0	-0.589009	0.0
12	-0.295733	0.0	0.0	0.0	-0.295733	0.0
13	-0.315017	0.0	0.0	0.0	-0.315017	0.0
14	-0.755526	0.0	-0.700000	0.0	-0.700000	0.0
15	-0.834346	0.0	-0.700000	0.0	-0.700000	0.0
16	-0.670626	0.0	0.0	0.0	-0.670626	0.0
17	-0.395491	0.0	0.0	0.0	-0.395491	0.0
18	-0.151506	0.0	0.0	0.0	-0.151506	0.0
19	-0.017357	0.0	0.0	0.0	-0.017357	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0
23	-0.108558	0.0	0.0	0.0	-0.108558	0.0

THE 4 BY 23 S-DOUBLE-BAR MATRIX

ROW 1

1	0.897280E-01	0.0	2	0.918750E-01	0.0	3	0.957950E-01	0.0
4	0.992000E-01	0.0	5	0.103170E+00	0.0	6	0.122899E+00	0.0
7	0.240948E+00	0.0	8	0.876945E+00	0.0	9	0.692850E+00	0.0
10	0.609280E+00	0.0	11	0.474375E+00	0.0	12	0.217600E+00	0.0
13	0.221712E+00	0.0	14	0.0	0.0	15	0.0	0.0
16	0.310800E+00	0.0	17	0.170500E+00	0.0	18	0.621130E-01	0.0
19	0.696136E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 2

1	0.221440E-01	0.0	2	0.203462E-01	0.0	3	0.165256E-01	0.0
4	0.102459E-01	0.0	5	0.190865E-02	0.0	6	-0.934647E-02	0.0
7	-0.419129E-01	0.0	8	-0.235464E+00	0.0	9	-0.244773E+00	0.0
10	-0.257424E+00	0.0	11	-0.223646E+00	0.0	12	-0.108101E+00	0.0
13	-0.111551E+00	0.0	14	0.0	0.0	15	0.0	0.0
16	-0.194250E+00	0.0	17	-0.115812E+00	0.0	18	-0.448922E-01	0.0
19	-0.519919E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 3

1	0.0	0.0	2	0.0	0.0	3	0.0	0.0
4	0.0	0.0	5	0.0	0.0	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0	9	0.0	0.0
10	0.0	0.0	11	0.0	0.0	12	0.0	0.0
13	-0.695066E-03	0.0	14	0.0	0.0	15	0.0	0.0
16	-0.388500E-01	0.0	17	-0.305621E-01	0.0	18	-0.138357E-01	0.0
19	-0.171852E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 4

1	0.918435E+00	0.0	2	0.919451E+00	0.0	3	0.917960E+00	0.0
4	0.874812E+00	0.0	5	0.794034E+00	0.0	6	0.733952E+00	0.0
7	0.795917E+00	0.0	8	0.129305E+01	0.0	9	0.820082E+00	0.0
10	0.534154E+00	0.0	11	0.297070E+00	0.0	12	0.870280E-01	0.0
13	0.872397E-01	0.0	14	0.0	0.0	15	0.0	0.0
16	0.226941E+00	0.0	17	0.136315E+00	0.0	18	0.528058E-01	0.0
19	0.607283E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.526316E-01	0.0			

DELTA-C-MOD

0.186830	0.0	0.091894	0.0	0.045145	0.0	-0.848846	0.0

-S- MATRIX

0.219071E+01	0.0	-0.774809E+00	0.0	-0.183109E-01	0.0
0.300937E+01	0.0	-0.774809E+00	0.0	0.312089E+00	0.0
0.117937E-01	0.0	-0.766082E+00	0.0	-0.183109E-01	0.0
0.117937E-01	0.0	0.263823E-02	0.0	-0.137844E-01	0.0
0.300937E+01	0.0	-0.766082E+00	0.0	-0.137844E-01	0.0
0.790847E+01	0.0				

-DC- COLUMN

0.186830E+00	0.0	0.918936E-01	0.0	0.451447E-01	0.0
-0.848846E+00	0.0				

SOLUTION OF MATRIX EQ.

0.112057E+02	0.0	0.237958E+02	0.0	-0.224892E+02	0.0
-0.210554E+01	0.0				

OUTPUT OF GEN. INVERSE (EPS-TILDA)

-0.401396	0.0	-0.422258	0.0	-0.466105	0.0	-0.486531	0.0
-0.470355	0.0	-0.390596	0.0	0.026817	0.0	1.501185	0.0
0.212602	0.0	-0.422853	0.0	-0.631618	0.0	-0.317233	0.0
-0.338055	0.0	0.0	0.0	0.0	0.0	-0.743718	0.0
-0.444962	0.0	-0.172254	0.0	-0.019850	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-0.110818	0.0		

J	EPSILON-BAR		EPSILON-LIMIT		EPSILON-LAST	
1	-0.401396	0.0	0.0	0.0	-0.401396	0.0
2	-0.422258	0.0	0.0	0.0	-0.422258	0.0
3	-0.466105	0.0	0.0	0.0	-0.466105	0.0
4	-0.486531	0.0	0.0	0.0	-0.486531	0.0
5	-0.470355	0.0	0.0	0.0	-0.470355	0.0
6	-0.390596	0.0	0.0	0.0	-0.390596	0.0
7	0.026817	0.0	0.0	0.0	0.026817	0.0
8	1.501185	0.0	1.500000	0.0	1.500000	0.0
9	0.212602	0.0	0.0	0.0	0.212602	0.0
10	-0.422853	0.0	0.0	0.0	-0.422853	0.0
11	-0.631618	0.0	0.0	0.0	-0.631618	0.0
12	-0.317233	0.0	0.0	0.0	-0.317233	0.0
13	-0.338055	0.0	0.0	0.0	-0.338055	0.0
14	0.0	0.0	-0.700000	0.0	-0.700000	0.0
15	0.0	0.0	-0.700000	0.0	-0.700000	0.0
16	-0.743718	0.0	-0.700000	0.0	-0.700000	0.0
17	-0.444962	0.0	0.0	0.0	-0.444962	0.0
18	-0.172254	0.0	0.0	0.0	-0.172254	0.0
19	-0.019850	0.0	0.0	0.0	-0.019850	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0
23	-0.110818	0.0	0.0	0.0	-0.110818	0.0

THE 4 BY 23 S-DOUBLE-BAR MATRIX

ROW 1

1	0.897280E-01	0.0	2	0.918750E-01	0.0	3	0.957950E-01	0.0
4	0.992000E-01	0.0	5	0.103170E+00	0.0	6	0.122899E+00	0.0
7	0.240948E+00	0.0	8	0.0	0.0	9	0.692850E+00	0.0
10	0.609280E+00	0.0	11	0.474375E+00	0.0	12	0.217600E+00	0.0
13	0.221712E+00	0.0	14	0.0	0.0	15	0.0	0.0
16	0.0	0.0	17	0.170500E+00	0.0	18	0.621130E-01	0.0
19	0.696136E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 2

1	0.221440E-01	0.0	2	0.203462E-01	0.0	3	0.165256E-01	0.0
4	0.102459E-01	0.0	5	0.190865E-02	0.0	6	-0.934647E-02	0.0
7	-0.419129E-01	0.0	8	0.0	0.0	9	-0.244773E+00	0.0
10	-0.257424E+00	0.0	11	-0.223646E+00	0.0	12	-0.108101E+00	0.0
13	-0.111551E+00	0.0	14	0.0	0.0	15	0.0	0.0
16	0.0	0.0	17	-0.115812E+00	0.0	18	-0.448922E-01	0.0
19	-0.519919E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 3

1	0.0	0.0	2	0.0	0.0	3	0.0	0.0
4	0.0	0.0	5	0.0	0.0	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0	9	0.0	0.0
10	0.0	0.0	11	0.0	0.0	12	0.0	0.0
13	-0.695066E-03	0.0	14	0.0	0.0	15	0.0	0.0
16	0.0	0.0	17	-0.305621E-01	0.0	18	-0.138357E-01	0.0
19	-0.171852E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.0	0.0			

ROW 4

1	0.918435E+00	0.0	2	0.919451E+00	0.0	3	0.917960E+00	0.0
4	0.874812E+00	0.0	5	0.794034E+00	0.0	6	0.733952E+00	0.0
7	0.795917E+00	0.0	8	0.0	0.0	9	0.820082E+00	0.0
10	0.534154E+00	0.0	11	0.297070E+00	0.0	12	0.870280E-01	0.0
13	0.872397E-01	0.0	14	0.0	0.0	15	0.0	0.0
16	0.0	0.0	17	0.136315E+00	0.0	18	0.528058E-01	0.0
19	0.607283E-02	0.0	20	0.0	0.0	21	0.0	0.0
22	0.0	0.0	23	0.526316E-01	0.0			

DELTA-C-MOD

-0.911027	0.0	0.309115	0.0	0.017950	0.0	-2.629568	0.0

```
-S-   MATRIX
 0.132509E+01     0.0        -0.507947E+00    0.0        -0.623628E-02    0.0
 0.180490E+01     0.0        -0.507947E+00    0.0         0.218913E+00    0.0
 0.424704E-02     0.0        -0.417531E+00    0.0        -0.623628E-02    0.0
 0.424704E-02     0.0         0.112890E-02    0.0        -0.496776E-02    0.0
 0.180490E+01     0.0        -0.417531E+00    0.0        -0.496776E-02    0.0
 0.618498E+01     0.0

-DC-  COLUMN
-0.911027E+00     0.0         0.309115E+00    0.0         0.179497E-01    0.0
-0.262957E+01     0.0

SOLUTION OF MATRIX EQ.
 0.113088E+02    -0.0         0.240208E+02    0.0        -0.213291E+02    0.0
-0.212086E+01     0.0

OUTPUT OF  GEN. INVERSE  (EPS-TILDA)
 -0.401236      0.0      -0.422297      0.0      -0.466577     0.0      -0.487404     0.0
 -0.471456      0.0      -0.391275      0.0       0.030033     0.0       0.0          0.0
  0.216393      0.0      -0.426141      0.0      -0.637582     0.0      -0.320454     0.0
 -0.342438      0.0       0.0           0.0       0.0          0.0       0.0          0.0
 -0.490984      0.0      -0.192810      0.0      -0.022389     0.0       0.0          0.0
  0.0           0.0       0.0           0.0      -0.111624     0.0
```

SYMMETRIC DELTA-CP-TILDA , MODE 1

J	X(J)	Y(J)	Z(J)	GAMMA(J)	DELTA-A(J)	EPS(J) REAL	EPS(J) IMAG.	DCP-TILDA(J) REAL	DCP-TILDA(J) IMAG.
1	-0.9936	0.0	0.0	0.0	0.0256	-0.401236	0.0	4.197338	0.0
2	-0.9429	0.0	0.0	0.0	0.0750	-0.422297	0.0	1.415373	0.0
3	-0.8450	0.0	0.0	0.0	0.1190	-0.466577	0.0	0.858811	0.0
4	-0.7066	0.0	0.0	0.0	0.1550	-0.487404	0.0	0.656123	0.0
5	-0.5370	0.0	0.0	0.0	0.1810	-0.471456	0.0	0.602541	0.0
6	-0.3479	0.0	0.0	0.0	0.1940	-0.391275	0.0	0.771254	0.0
7	-0.1521	0.0	0.0	0.0	0.1940	0.030033	0.0	2.558602	0.0
8	0.0370	0.0	0.0	0.0	0.1810	1.500000	0.0	24.224991	0.0
9	0.2066	0.0	0.0	0.0	0.1550	0.216393	0.0	10.874557	0.0
10	0.3450	0.0	0.0	0.0	0.1190	-0.426141	0.0	5.876318	0.0
11	0.4429	0.0	0.0	0.0	0.0750	-0.637582	0.0	4.584589	0.0
12	0.4936	0.0	0.0	0.0	0.0256	-0.320454	0.0	11.552276	0.0
13	0.5063	0.0	0.0	0.0	0.0248	-0.342438	0.0	11.757212	0.0
14	0.5545	0.0	0.0	0.0	0.0694	-0.700000	0.0	3.750000	0.0
15	0.6415	0.0	0.0	0.0	0.1000	-0.700000	0.0	2.526000	0.0
16	0.7500	0.0	0.0	0.0	0.1110	-0.700000	0.0	1.679999	0.0
17	0.8585	0.0	0.0	0.0	0.1000	-0.490984	0.0	1.735744	0.0
18	0.9455	0.0	0.0	0.0	0.0694	-0.192810	0.0	1.444869	0.0
19	0.9937	0.0	0.0	0.0	0.0248	-0.022389	0.0	0.548831	0.0

SYMMETRIC DELTA-CP-TILDA , MODE 2

J	X(J)	Y(J)	Z(J)	GAMMA(J)	DELTA-A(J)	EPS(J) REAL	EPS(J) IMAG.	DCP-TILDA(J) REAL	DCP-TILDA(J) IMAG.
1	-0.9936	0.0	0.0	0.0	0.0256	-0.401236	0.0	42.962952	0.0
2	-0.9429	0.0	0.0	0.0	0.0750	-0.422297	0.0	14.164542	0.0
3	-0.8450	0.0	0.0	0.0	0.1190	-0.466577	0.0	8.229603	0.0
4	-0.7066	0.0	0.0	0.0	0.1550	-0.487404	0.0	5.786134	0.0
5	-0.5370	0.0	0.0	0.0	0.1810	-0.471456	0.0	4.637374	0.0
6	-0.3479	0.0	0.0	0.0	0.1940	-0.391275	0.0	4.605927	0.0
7	-0.1521	0.0	0.0	0.0	0.1940	0.030033	0.0	8.451761	0.0
8	0.0370	0.0	0.0	0.0	0.1810	1.500000	0.0	35.719742	0.0
9	0.2066	0.0	0.0	0.0	0.1550	0.216393	0.0	12.871511	0.0
10	0.3450	0.0	0.0	0.0	0.1190	-0.426141	0.0	5.151746	0.0
11	0.4429	0.0	0.0	0.0	0.0750	-0.637582	0.0	2.871033	0.0
12	0.4936	0.0	0.0	0.0	0.0256	-0.320454	0.0	4.620271	0.0
13	0.5063	0.0	0.0	0.0	0.0248	-0.342438	0.0	4.626251	0.0
14	0.5545	0.0	0.0	0.0	0.0694	-0.700000	0.0	2.043905	0.0
15	0.6415	0.0	0.0	0.0	0.1000	-0.700000	0.0	1.652123	0.0
16	0.7500	0.0	0.0	0.0	0.1110	-0.700000	0.0	1.226708	0.0
17	0.8585	0.0	0.0	0.0	0.1000	-0.490984	0.0	1.387734	0.0
18	0.9455	0.0	0.0	0.0	0.0694	-0.192810	0.0	1.228365	0.0
19	0.9937	0.0	0.0	0.0	0.0248	-0.022389	0.0	0.478780	0.0

CORRECTION FACTORS ** PREMULTIPLIER CASE

1	0.598764E+00	0.0	2	0.577703E+00	0.0	3	0.533423E+00	0.0
4	0.512596E+00	0.0	5	0.528544E+00	0.0	6	0.608725E+00	0.0
7	0.103003E+01	0.0	8	0.250000E+01	0.0	9	0.121639E+01	0.0
10	0.573859E+00	0.0	11	0.362418E+00	0.0	12	0.679546E+00	0.0
13	0.657562E+00	0.0	14	0.300000E+00	0.0	15	0.300000E+00	0.0
16	0.300000E+00	0.0	17	0.509016E+00	0.0	18	0.807190E+00	0.0
19	0.977611E+00	0.0						

SYMMETRIC CE-N MODE 1

N	LABEL	CE(N) REAL	CE(N) IMAG.
1	CL	0.493003E+01	0.0
2	CM-1/4	-0.157000E+01	0.0
3	CH-3/4	-0.530002E-01	0.0
4	CL-ALPHA	0.493003E+01	0.0

SYMMETRIC CF-N MODE 2

N	LABEL	CE(N) REAL	IMAG.
1	CL	0.880589E+01	0.0
2	CM-1/4	-0.145306E+01	0.0
3	CH-3/4	-0.398661E-01	0.0
4	CL-ALPHA	0.880589E+01	0.0

TABLE V

INPUT DATA - TEST CASE 2

```
TEST CASE NO. 2  NEW POST-MULT., TAPE INPUT, M=.875, K=0   3CONTROL DATA      1
         19         3          0          0        3       1                  2
          0         4          1          1        0                          3
          1         0                                                         4
          1         1                                       AXIS DATA         7
 0.0         0.0       0.0        0.0        0.0       1.0                    8
          2         1
-0.5         0.0       0.0        0.0        1.0       0.0
          3         1
 0.5         0.0       0.0        0.0        1.0       0.0  CONSTRAINT DATA   9
          1         0                                                        10
          1         1  1.0        2.0        4.93      0.0                   11
          1        19                                                        12
-1
          2         1
          1         1  1.0        4.0       -1.57      0.0
          1        19
-1
          3         1
          1         1  1.0        4.0       -.053      0.0
         13        19
-1                                                          MONITOR DATA     13
          1         0                                                        14
          1         1  1.0        2.0        CL                              15
          1        19                                                        16
-1
          2         1
          1         1  1.0        4.0        CM-1/4
          1        19
-1
          3         1
          1         1  1.0        4.0        CH-3/4
         13        19
-1
          1         1          13         19-1.0                NORMALWASH DATA  22
-1                                                                               23
```

TABLE VI

ØUTPUT LISTING - TEST CASE 2

```
TEST CASE NO. 2  NEW POST-MULT., TAPE INPUT, M=.875, K=0


CONTROL FLAGS          --

FLAGB = 0    CORRECTION FACTORS CALCULATED
FLAGP = 4    NEW POSTMULTIPLIER   -   DT INVERSE AND GEOMETRY TAKEN FROM TAPE
FLAGT = 1    WEIGHTS = 1.0
FLAGW = 1    NORMALWASH TAKEN FROM CARDS      (IF NEEDED)
IPRINT= 1    (DETAIL PRINT FLAG)


CONTROL DIMENSIONS    --

NP     =  19
NC     =   3
NEM    =   0
NELIMS=    0
NMON   =   3
NAXIS  =   3


LIST OF INPUT/OUTPUT TAPES   --

GEOMETRY  TAPE = 11
DELTA-CP  TAPE = 12
W         TAPE = 13
D-INVERSE TAPE = 14
CORR. FACTORS  = 16
```

THE 3 SETS OF INPUT DATA FOR ALL AXES

AXISNO(R)	FLAGA(R)	XI1(R)	ETA1(R)	ZETA1(R)	XI2(R) (COSA(R))	ETA2(R) (COSB(R))	ZETA2(R) (COSG(R))
1	1	0.0	0.0	0.0	0.0	0.0	1.000000
2	1	-0.500000	0.0	0.0	0.0	1.000000	0.0
3	1	0.500000	0.0	0.0	0.0	1.000000	0.0

THE 3 SETS OF INPUT DATA FOR ALL CONSTRAINTS

AXISNO(I)	FLAGF(I)	DELTA	M	LIM1	LIM2	A-WIG(I)	C-WIG(I)	C-F(I) REAL	C-F(I) IMAG.
1	0	1	1	1	19	1.000000	0.200000E+01	4.930000	0.0
2	1	1	1	1	19	1.000000	0.400000E+01	-1.570000	0.0
3	1	1	1	13	19	1.000000	0.400000E+01	-0.053000	0.0

THE 3 SETS OF INPUT DATA FOR MONITORING

AXISNO(N)	FLAGE(N)	DELTA	M	LIM1	LIM2	A-WIG(N)	C-WIG(N)	LABEL
1	0	1	1	1	19	1.000000	0.200000E+01	CL
2	1	1	1	1	19	1.000000	0.400000E+01	CM-1/4
3	1	1	1	13	19	1.000000	0.400000E+01	CH-3/4

SAI MATRIX ROWS

0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		
0.003159	0.0	0.008305	0.0	0.010264	0.0	0.008005	0.0
0.001674	0.0	-0.007377	0.0	-0.016873	0.0	-0.024300	0.0
-0.027380	0.0	-0.025139	0.0	-0.017680	0.0	-0.036359	0.0
-0.006239	0.0	-0.018296	0.0	-0.028538	0.0	-0.034687	0.0
-0.033962	0.0	-0.025079	0.0	-0.009261	0.0		
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.000039	0.0	-0.000946	0.0	-0.003538	0.0	-0.006937	0.0
-0.008962	0.0	-0.007729	0.0	-0.003061	0.0		

SAN MATRIX ROWS

0.012800	0.0	0.037500	0.0	0.059500	0.0	0.077500	0.0
0.090500	0.0	0.097000	0.0	0.097000	0.0	0.090500	0.0
0.077500	0.0	0.059500	0.0	0.037500	0.0	0.012800	0.0
0.012400	0.0	0.034700	0.0	0.050000	0.0	0.055500	0.0
0.050000	0.0	0.034700	0.0	0.012400	0.0		
0.003159	0.0	0.008305	0.0	0.010264	0.0	0.008005	0.0
0.001674	0.0	-0.007377	0.0	-0.016873	0.0	-0.024300	0.0
-0.027380	0.0	-0.025139	0.0	-0.017680	0.0	-0.006359	0.0
-0.006239	0.0	-0.018296	0.0	-0.028538	0.0	-0.034687	0.0
-0.033962	0.0	-0.025079	0.0	-0.009261	0.0		
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-0.000039	0.0	-0.000946	0.0	-0.003538	0.0	-0.006937	0.0
-0.008962	0.0	-0.007729	0.0	-0.003061	0.0		

```
--W--   IS CARD INPUT
MODE    DELTA      LIMITS                    - W -
                                      REAL              IMAG.
  1        1        13     19      -1.000000         0.0
```

THE 1 COLUMNS OF THE SYMMETRIC W MATRIX

COLUMN 1

1	0.0	0.0	2	0.0	0.0	3	0.0	0.0
4	0.0	0.0	5	0.0	0.0	6	0.0	0.0
7	0.0	0.0	8	0.0	0.0	9	0.0	0.0
10	0.0	0.0	11	0.0	0.0	12	0.0	0.0
13	-0.100000E+01	0.0	14	-0.100000E+01	0.0	15	-0.100000E+01	0.0
16	-0.100000E+01	0.0	17	-0.100000E+01	0.0	18	-0.100000E+01	0.0
19	-0.100000E+01	0.0						

THE 1 COLUMNS OF THE DELTA-CP-BAR MATRIX

COLUMN 1

1	0.716455E+01	-0.731030E+00	2	0.250015E+01	-0.250801E+00	3	0.165207E+01	-0.159399E+00
4	0.131495E+01	-0.118636E+00	5	0.117077E+01	-0.949423E-01	6	0.130501E+01	-0.868268E-01
7	0.257012E+01	-0.109281E+00	8	0.100245E+02	-0.220436E+00	9	0.922275E+01	-0.156961E+00
10	0.104787E+02	-0.129675E+00	11	0.127522E+02	-0.110500E+00	12	0.154990E+02	-0.924698E-01
13	0.196969E+02	-0.944285E-01	14	0.131619E+02	-0.883011E-01	15	0.908173E+01	-0.676048E-01
16	0.614658E+01	-0.472467E-01	17	0.387184E+01	-0.297328E-01	18	0.208988E+01	-0.158927E-01
19	0.661877E+00	-0.499649E-02						

DELTA-C

-0.630051	0.122550	0.519804	-0.018031	0.087867	-0.001059

THEORETICAL C-VALUES

5.560051	-0.122550	-2.083804	0.018031	-0.140867	0.001059

ONE S-BAR ROW

-0.035068	0.000612	-0.142536	0.002546	-0.302465	0.005638	-0.488549	0.009541
-0.673939	0.013722	-0.746680	0.015803	-0.612399	0.012899	-0.414697	0.008227
-0.437368	0.008864	-0.357380	0.007347	-0.204170	0.004221	-0.031777	0.000653
-0.195771	0.004057	-0.404510	0.008493	-0.636052	0.013583	-0.883069	0.019254
-1.143060	0.025356	-1.426368	0.032111	-0.871221	0.019694	0.0	0.0
0.0	0.0	0.0	0.0				

ONE S-BAR ROW

-0.022906	-0.000047	-0.088104	-0.000163	-0.170660	-0.000274	-0.239268	-0.000325
-0.266833	-0.000338	-0.212600	-0.000411	-0.085731	-0.000688	0.005131	-0.000793
0.039684	-0.000944	0.054618	-0.000838	0.039468	-0.000504	0.006702	-0.000080
0.043300	-0.000501	0.102894	-0.001087	0.191509	-0.001820	0.311120	-0.002737
0.453494	-0.003788	0.607809	-0.004997	0.379680	-0.003102	0.0	0.0
0.0	0.0	0.0	0.0				

ONE S-BAR ROW

-0.000135	-0.000005	-0.000569	-0.000020	-0.001298	-0.000043	-0.002319	-0.000069
-0.003637	-0.000094	-0.004861	-0.000103	-0.005300	-0.000088	-0.005218	-0.000065
-0.007625	-0.000069	-0.008914	-0.000056	-0.007270	-0.000031	-0.001456	-0.000005
-0.010606	-0.000030	-0.020665	-0.000059	-0.018662	-0.000093	0.003496	-0.000144
0.042317	-0.000217	0.085811	-0.000315	0.059176	-0.000202	0.0	0.0
0.0	0.0	0.0	0.0				

-S- MATRIX

0.776227E+01	0.0	-0.163754E+01	-0.168500E-01	-0.181983E+00	-0.273930E-02
-0.163754E+01	0.168500E-01	0.108984E+01	0.0	0.909225E-01	0.326475E-03
-0.181983E+00	0.273930E-02	0.909225E-01	-0.326475E-03	0.138484E-01	0.0

-DC- COLUMN

-0.630051E+00	0.122550E+00	0.519804E+00	-0.180307E-01	0.878674E-01	-0.105876E-02

SOLUTION OF MATRIX EQ.

0.951395E-01	0.167979E-01	-0.304458E-01	-0.258064E-02	0.779167E+01	0.180770E+00

OUTPUT OF GEN. INVERSE (EPS-TILDA)

-0.003680	-0.000575	-0.015273	-0.002359	-0.033606	-0.005084	-0.057116	-0.008387
-0.084121	-0.011875	-0.102196	-0.013584	-0.096750	-0.011589	-0.080137	-0.008222
-0.102091	-0.009162	-0.105001	-0.008042	-0.077203	-0.005017	-0.014564	-0.000841
-0.102517	-0.005489	-0.202497	-0.011175	-0.211533	-0.015178	-0.065944	-0.015800
0.207560	-0.013562	0.514896	-0.010769	0.366935	-0.005312	0.0	0.0
0.0	0.0	0.0	0.0				

SYMMETRIC DELTA-CP-TILDA , MODE 1

J	X(J)	Y(J)	Z(J)	GAMMA(J)	DELTA-A(J)	EPS(J) REAL	EPS(J) IMAG.	DCP-TILDA(J) REAL	DCP-TILDA(J) IMAG.
1	-0.9936	0.0	0.0	0.0	0.0256	-0.003680	-0.000575	8.731648	-0.201151
2	-0.9429	0.0	0.0	0.0	0.0750	-0.015273	-0.002359	3.205733	-0.042868
3	-0.8450	0.0	0.0	0.0	0.1190	-0.033606	-0.005084	2.304436	0.003822
4	-0.7066	0.0	0.0	0.0	0.1550	-0.057116	-0.008387	1.993573	0.031236
5	-0.5370	0.0	0.0	0.0	0.1810	-0.084121	-0.011875	1.857959	0.045195
6	-0.3479	0.0	0.0	0.0	0.1940	-0.102196	-0.013584	1.897981	0.036609
7	-0.1521	0.0	0.0	0.0	0.1940	-0.096750	-0.011589	2.755267	-0.010252
8	0.0370	0.0	0.0	0.0	0.1810	-0.080137	-0.008222	9.003983	-0.069108
9	0.2066	0.0	0.0	0.0	0.1550	-0.102091	-0.009162	8.396429	-0.024435
10	0.3450	0.0	0.0	0.0	0.1190	-0.105001	-0.008042	9.362289	-0.019947
11	0.4429	0.0	0.0	0.0	0.0750	-0.077203	-0.005017	11.350240	-0.014470
12	0.4936	0.0	0.0	0.0	0.0256	-0.014564	-0.000841	14.212420	0.003620
13	0.5063	0.0	0.0	0.0	0.0248	-0.102517	-0.005489	18.871185	0.040578
14	0.5545	0.0	0.0	0.0	0.0694	-0.202497	-0.011175	12.228393	0.065479
15	0.6415	0.0	0.0	0.0	0.1000	-0.211533	-0.015178	7.173484	0.053617
16	0.7500	0.0	0.0	0.0	0.1110	-0.065944	-0.015800	2.928123	0.019568
17	0.8585	0.0	0.0	0.0	0.1000	0.207560	-0.013562	0.046162	-0.011998
18	0.9455	0.0	0.0	0.0	0.0694	0.514896	-0.010769	-0.950595	-0.024853
19	0.9937	0.0	0.0	0.0	0.0248	0.366935	-0.005312	0.634418	-0.029181

CORRECTION FACTORS ** POSTMULTIPLIER CASE

1	0.996320E+00	-0.574841E-03	2	0.984727E+00	-0.235866E-02	3	0.966394E+00	-0.508433E-02
4	0.942884E+00	-0.838712E-02	5	0.915879E+00	-0.118752E-01	6	0.897804E+00	-0.135835E-01
7	0.903250E+00	-0.115889E-01	8	0.919863E+00	-0.822229E-02	9	0.897909E+00	-0.916240E-02
10	0.894999E+00	-0.804217E-02	11	0.922797E+00	-0.501745E-02	12	0.985436E+00	-0.841083E-03
13	0.897483E+00	-0.548857E-02	14	0.797503E+00	-0.111750E-01	15	0.788467E+00	-0.151780E-01
16	0.934056E+00	-0.158000E-01	17	0.120756E+01	-0.135618E-01	18	0.151490E+01	-0.107688E-01
19	0.136693E+01	-0.531235E-02						

FLAGP = 4 -- TAPE CONTAINS THE EPSILON-BAR (EB) VALUES

1	-0.368013E-02	-0.574841E-03	2	-0.152729E-01	-0.235866E-02	3	-0.336063E-01	-0.508433E-02
4	-0.571160E-01	-0.838712E-02	5	-0.841213E-01	-0.118752E-01	6	-0.102196E+00	-0.135835E-01
7	-0.967499E-01	-0.115889E-01	8	-0.801373E-01	-0.822229E-02	9	-0.102091E+00	-0.916240E-02
10	-0.105001E+00	-0.804217E-02	11	-0.772026E-01	-0.501745E-02	12	-0.145645E-01	-0.841083E-03
13	-0.102517E+00	-0.548857E-02	14	-0.202497E+00	-0.111750E-01	15	-0.211533E+00	-0.151780E-01
16	-0.659443E-01	-0.158000E-01	17	0.207560E+00	-0.135618E-01	18	0.514896E+00	-0.107688E-01
19	0.366935E+00	-0.531235E-02						

SYMMETRIC CF-N MODE 1

N	LABEL	CF(N) REAL	IMAG.
1	CL	0.492999E+01	-0.431901E-06
2	CM-1/4	-0.157000E+01	0.181608E-07
3	CH-3/4	-0.530001E-01	-0.276486E-09

Subroutine Description

The computer program for generating correction factors (E1GC) consists of twenty subroutines. The MAIN of this program reads and writes the header card and reads the control dimensions for a case; the latter are used for dimensioning most of the complex arrays that are passed into Subroutine WEYT via the argument list. Subroutine WEYT is the actual working main of the program, which calls all the major subroutines, supplying these with the necessary information via their argument lists. The following is a detailed description of all subroutines of program E1GC including their flow charts, where applicable, given in alphabetical order. The computer program is written in the FORTRAN IV programming language.

SUBRØUTINE CEMN(NPØT, IGØ, MØDE, NTAPSA, NP, NMON, LABEL, NUTL, SAI, DCPTIL, CE)

Functional Description

This subroutine integrates the corrected pressures, ΔC_{p_e}, into coefficients, C_e, which are used to monitor the results (See Eq. (3)). The integration procedure is identical to that required for obtaining the imposed constraints.

$$\{C_e\} = [S]\{\Delta C_{p_e}\}$$

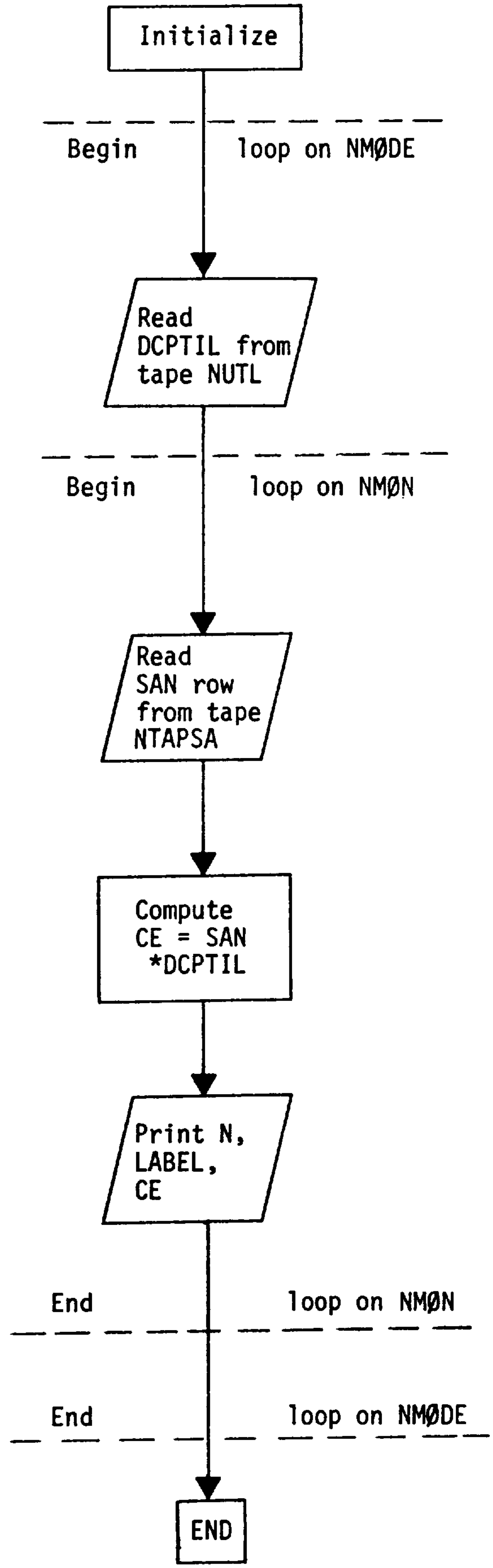
The coefficients C_e are part of the printed output.

Description of Argument List

NPØT	Data set number of the system output data set
IGØ	1 for symmetric modes
	2 for antisymmetric modes
MØDE	Mode number
NTAFSA	Data set (tape) number of tape containing the integration matrix [S]
NP	Number of rows in the ΔC_p matrix
NMØN	Number of integration rows used for monitoring
LABEL	Alphanumeric label describing the aerodynamic parameters
NUTL	Data set (tape) number of tape containing columns of the weighted pressures, $\Delta \tilde{C}_{p_e}$
SAI	A row of the integration matrix [S]
DCPTIL	A column of the weighted pressures $\{\Delta \tilde{C}_{p_e}\}$
CE	A column of the aerodynamic parameters $\{C_e\}$

Calling Subroutine WEYT

Flow Chart


Initialize
Begin loop on NMØDE
Read DCPTIL from tape NUTL
Begin loop on NMØN
Read SAN row from tape NTAPSA
Compute CE = SAN *DCPTIL
Print N, LABEL, CE
End loop on NMØN
End loop on NMØDE
END

SUBRØUTINE DCPB(NTDCP, NTAPW, NTAPDI, IGØ, IFP, IFW, NRØW, NCØL, NMAX, DCP, CØL, WØRK)

Functional Description

This subroutine computes the theoretical pressure distribution if it is not input. Specifically

$$\{\Delta C_{p_t}\} = [D]^{-1} \{W\}$$

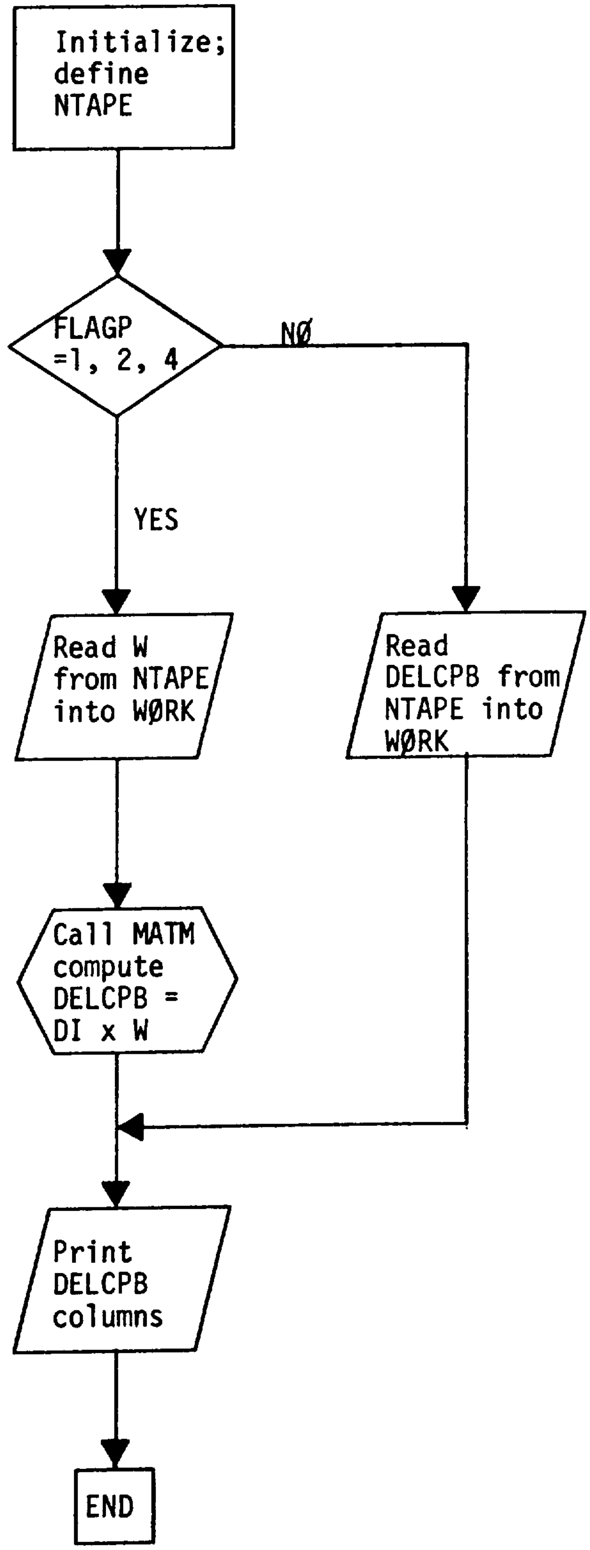
where W is the normalwash and $[D]^{-1}$ is the inverse of the aerodynamic influence coefficient matrix. This corresponds to Equation (1) where $[D]^{-1}$ = [A]. ΔC_{p_t} is called DELCPB in this subroutine.

Description of Argument List

NTDCP	Tape number containing the matrix of pressure coefficients, $[\Delta C_{p_t}]$, in column order
NTAPW	Tape number containing the normalwash matrix, [W], in column order
NTAPDI	Tape number containing the inverse-D matrix, ([A]) [DI], in row order
IGØ	1 for symmetric modes, 2 for antisymmetric modes
IFP	Control flag (see input flag FLAGP). IFP = 0, 2, 3 means premultiplying correction factors, IFP = 1, 4 means post-multiplying correction factors
IFW	Normalwash flag. IFW = 0 means normalwash is tape input (if any), IFW = 1 means normalwash is card input
NRØW	Number of rows in the ΔC_{p_t} matrix
NCØL	Number of columns in the ΔC_{p_t} matrix
NMAX	Maximum number of columns in the ΔC_{p_t} matrix
DCP	One column of the ΔC_{p_t} matrix (complex)
CØL	Temporary work array (complex)
WØRK	The NROW $\times$ NMAX complex array containing the ΔC_{p_t} matrix

Calling Subroutine WEYT

Flow Chart


Initialize;
define
NTAPE
FLAGP
=1, 2, 4
NØ
YES
Read W
from NTAPE
into WØRK
Read
DELCPB from
NTAPE into
WØRK
Call MATM
compute
DELCPB =
DI x W
Print
DELCPB
columns
END

SUBROUTINE DCPT(NPØT, FLAGB, IGØ, MØDE, NP, NSCRCH, NUTL, NTAPDI, NTAPW, NTAPCF, X, Y, Z, GMA, DELA, NMAX, NEM, W, DI, EPS, DCPBAR, DCPTIL, WØRK, EB)

Functional Description

This subroutine modifies the theory with the calculated correction factors. If a premultiplier is used the theoretical pressure, ΔC_{p_t}, is modified to produce the modified pressures ΔC_{p_e} (see eqs.(2) and (5)).

$$\left\{ \Delta C_{p_e} \right\} = [1+\epsilon]\left\{\Delta C_{p_t}\right\}$$

If a postmultiplier is used then the downwash, W, is modified to produce the corrected pressures ΔC_p (see eqs. (5) and (28)).

$$\left\{\Delta C_{p_e}\right\} = [D]^{-1} \lceil 1+\epsilon \rfloor \{W\}$$

If the new postmultiplier is used then

$$\left\{\Delta C_{p_e}\right\} = [D]^{-1} \{W + [\bar{\phi}] \{\bar{\epsilon}\}\}$$

$$[\bar{\phi}] = [\phi] \lceil \ell \rfloor$$

$$\{\ell\} = [\phi]^T \{\Delta C_{p_t}\}$$

(See eq. 65)

Also the correction factors, CF, are written on tape where

$$CF = 1 + \epsilon$$

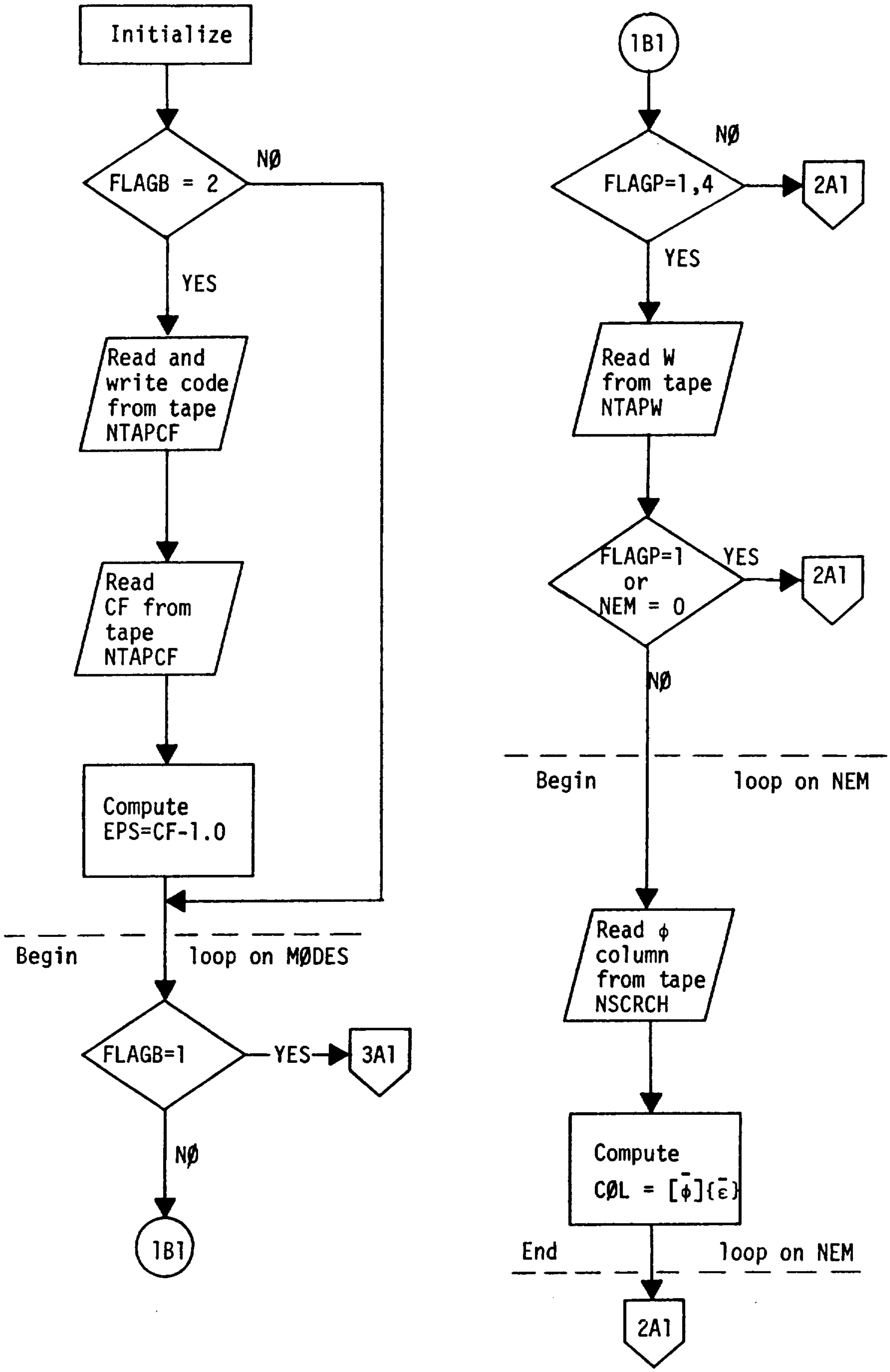
Description of Arguments

NPØT	Data set number of the system output data set
FLAGB	Option flag for correction matrix calculation and/or monitoring of data
IGØ	1 for symmetric modes 2 for antisymmetric modes
MØDE	Mode number
NP	Number of row elements in the ΔC_p matrix
NSCRCH	Data set (tape) number containing the $\bar{\phi}$ matrix in row order (for FLAGP=4 cases only)
NUTL	Data set (tape) number on which the ΔC_{p_e} columns are saved
NTAPDI	Data set (tape) number containing the D^{-1} matrix rows (if needed) $(D^{-1} = A)$
NTAPW	Data set (tape) number containing the W matrix columns (if needed)
NTAPCF	Data set (tape) number on which the matrix of correction factors, CF, is saved in column order
X	x coordiantes } of the pressure points or the ΔC_{p_t}
Y	y coordinates } of the pressure points or the ΔC_{p_t}
Z	z coordinates } of the pressure points or the ΔC_{p_t}
GMA	Dihedral angle array of the boxes over which the pressures act
DELA	Array of box areas
NMAX	Column dimension of the two-dimensional complex array WØRK
NEM	Number of correction factor modes

W	A column of the W matrix (complex)
DI	A row of the D^{-1} matrix (complex), ([A] matrix)
EPS	ε array
DCPBAR	A column of the $\Delta\bar{C}_{p_t}$ matrix (complex)
DCPTIL	A column of the ΔC_{p_e} matrix (complex)
WØRK	Two dimensional complex array containing the ΔC_{p_t} matrix
EB	$\bar{\varepsilon}$ array $(\varepsilon = \phi\bar{\varepsilon})$

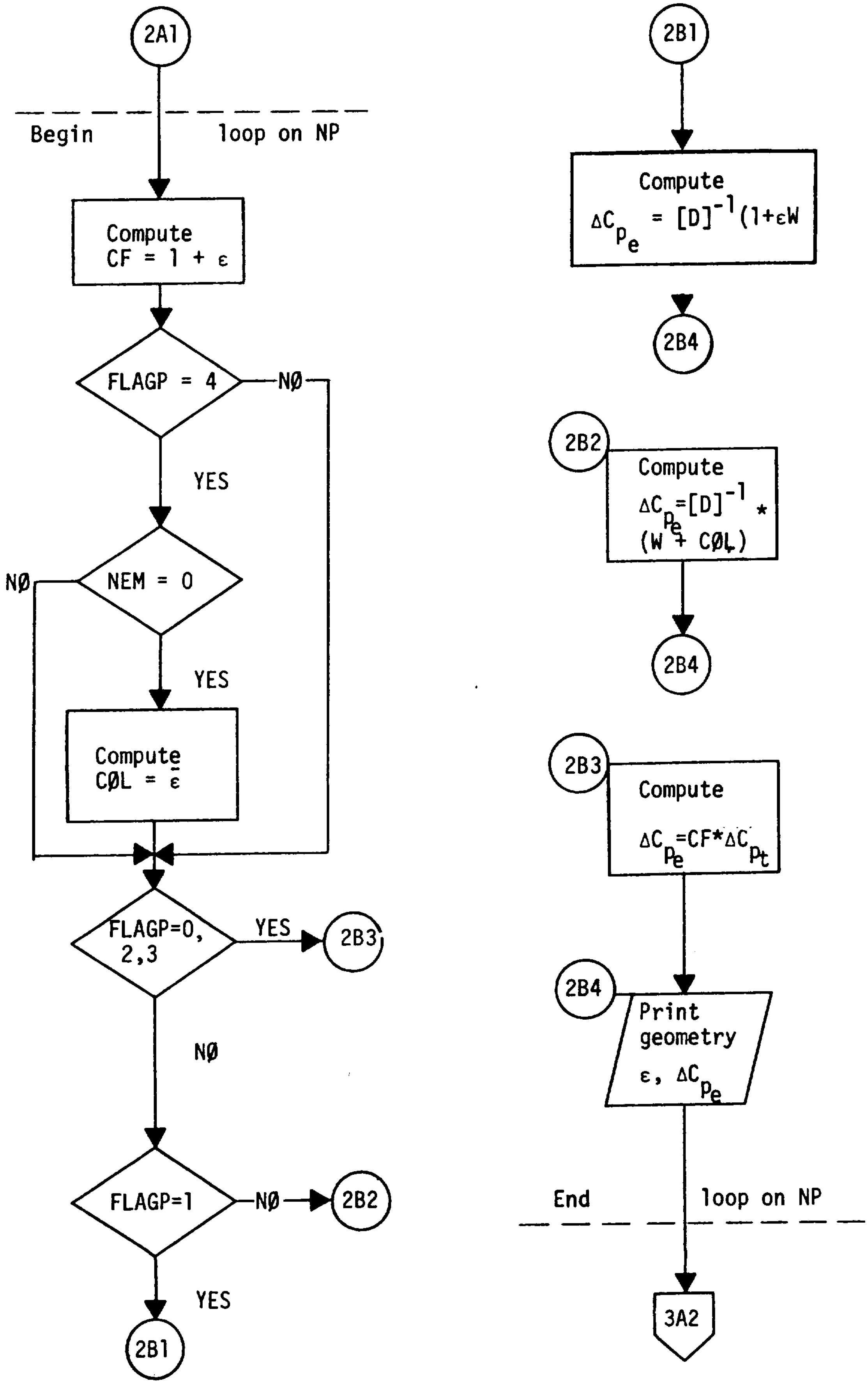
<u>Calling Subroutine</u> WEYT

Flow Chart



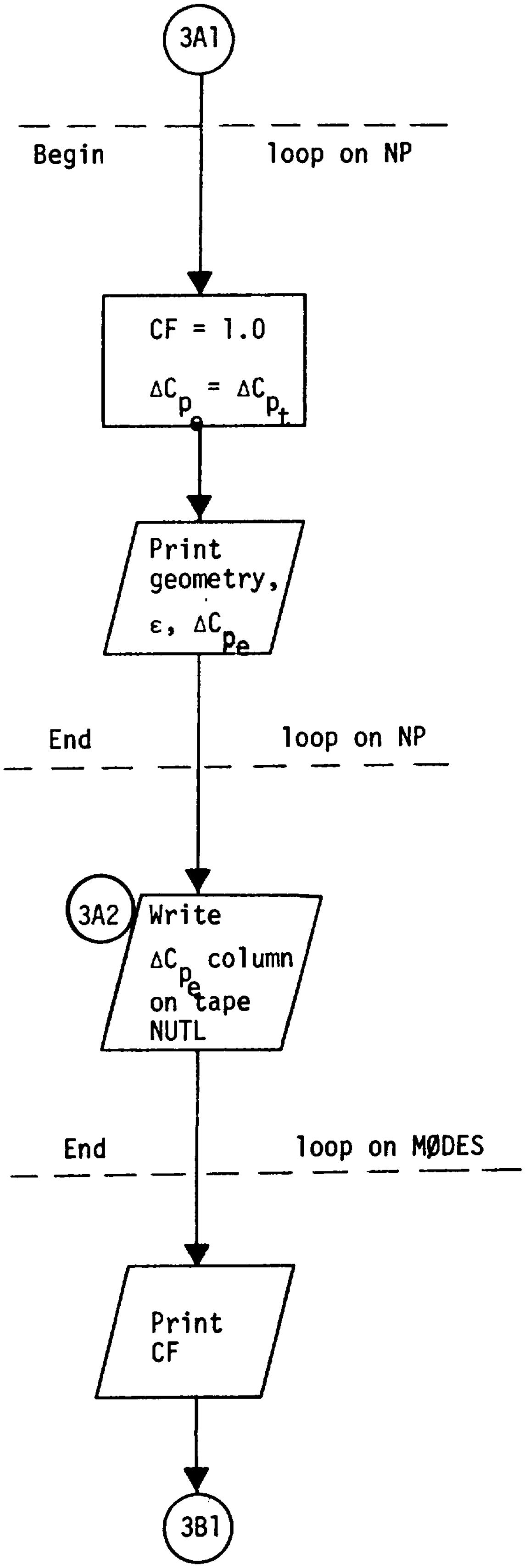
Initialize
FLAGB = 2
NØ
YES
Read and write code from tape NTAPCF
Read CF from tape NTAPCF
Compute EPS=CF-1.0
Begin loop on MØDES
FLAGB=1
YES
3A1
NØ
1B1
1B1
FLAGP=1,4
NØ
2A1
YES
Read W from tape NTAPW
FLAGP=1 or NEM = 0
YES
2A1
NØ
Begin loop on NEM
Read φ column from tape NSCRCH
Compute CØL = [φ̄]{ε̄}
End loop on NEM
2A1

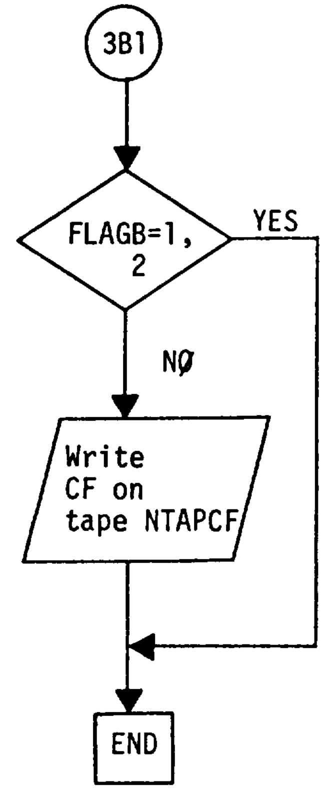
Flow Chart



2A1
Begin
loop on NP
Compute
CF = 1 + ε
FLAGP = 4
NØ
YES
NEM = 0
NØ
YES
Compute
CØL = ε
FLAGP=0, 2,3
YES
2B3
NØ
FLAGP=1
NØ
2B2
YES
2B1
2B1
Compute
ΔCpe = [D]^-1 (1+εW
2B4
2B2
Compute
ΔCpe=[D]^-1 *
(W + CØL)
2B4
2B3
Compute
ΔCpe=CF*ΔCpt
2B4
Print
geometry
ε, ΔCpe
End
loop on NP
3A2

Flow Chart


3A1
Begin loop on NP
CF = 1.0
$\Delta C_{p_e} = \Delta C_{p_t}$
Print geometry, ε, ΔC_{p_e}
End loop on NP
3A2
Write ΔC_{p_e} column on tape NUTL
End loop on MØDES
Print CF
3B1


3B1
FLAGB=1, 2
YES
NØ
Write CF on tape NTAPCF
END

SUBROUTINE DELC(NTAPE, NPØT, NC, NP, NMØDE, NMAX, CIE, DCI, SAI, WØRK)

Functional Description

This subroutine forms the difference between the theoretical, C_t, and the experimental (constrained) C_e coefficients (see Equations (9) and (10)).

$$\{\Delta C_e\} = \{C_e\} - [S]\{\Delta C_{p_t}\}$$

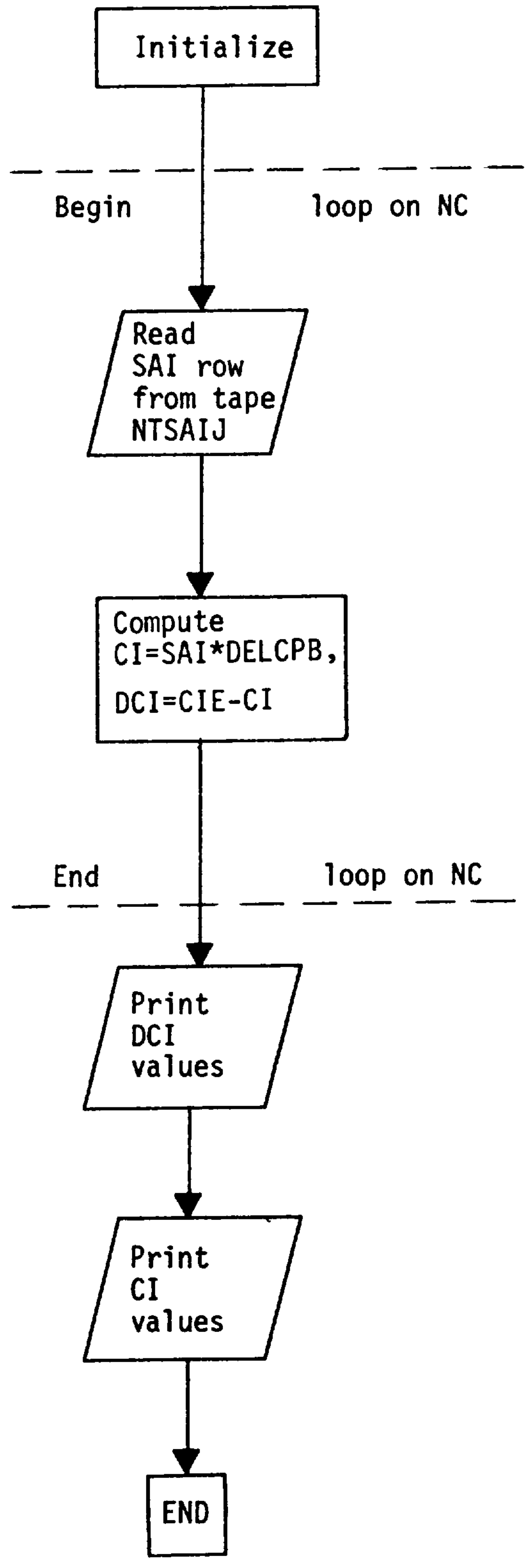
It also prints out C_t ($= [S]\{\Delta C_{p_t}\}$) and ΔC.

Description of Argument List

NTAPE	Tape containing rows of the integration matrix, [S]
NPØT	Data set number of the system output data set
NC	Number of constraints applied to ΔC_p values
NP	Number·of row elements in the ΔC_p matrix
NMØDE	Number of modes
NMAX	Maximum number of columns in the two-dimensional WØRK array
CIE	Array containing the input values C_e (experimental constraints)
DCI	The $[\Delta C_e]$ matrix
SAI	A row of the integration matrix [S.]
WORK	The NP by NMAX complex array containing the ΔC_{p_t} matrix

Calling Subroutine WEYT

Flow Chart



Initialize
Begin loop on NC
Read
SAI row
from tape
NTSAIJ
Compute
CI=SAI*DELCPB,
DCI=CIE-CI
End loop on NC
Print
DCI
values
Print
CI
values
END

SUBRØUTINE EDBL(NPØT, NELIMS, NP, NS, LIMK, JARR, NSMØD, EBMIN,

EBMAX, EB, ELIM)

Functional Description

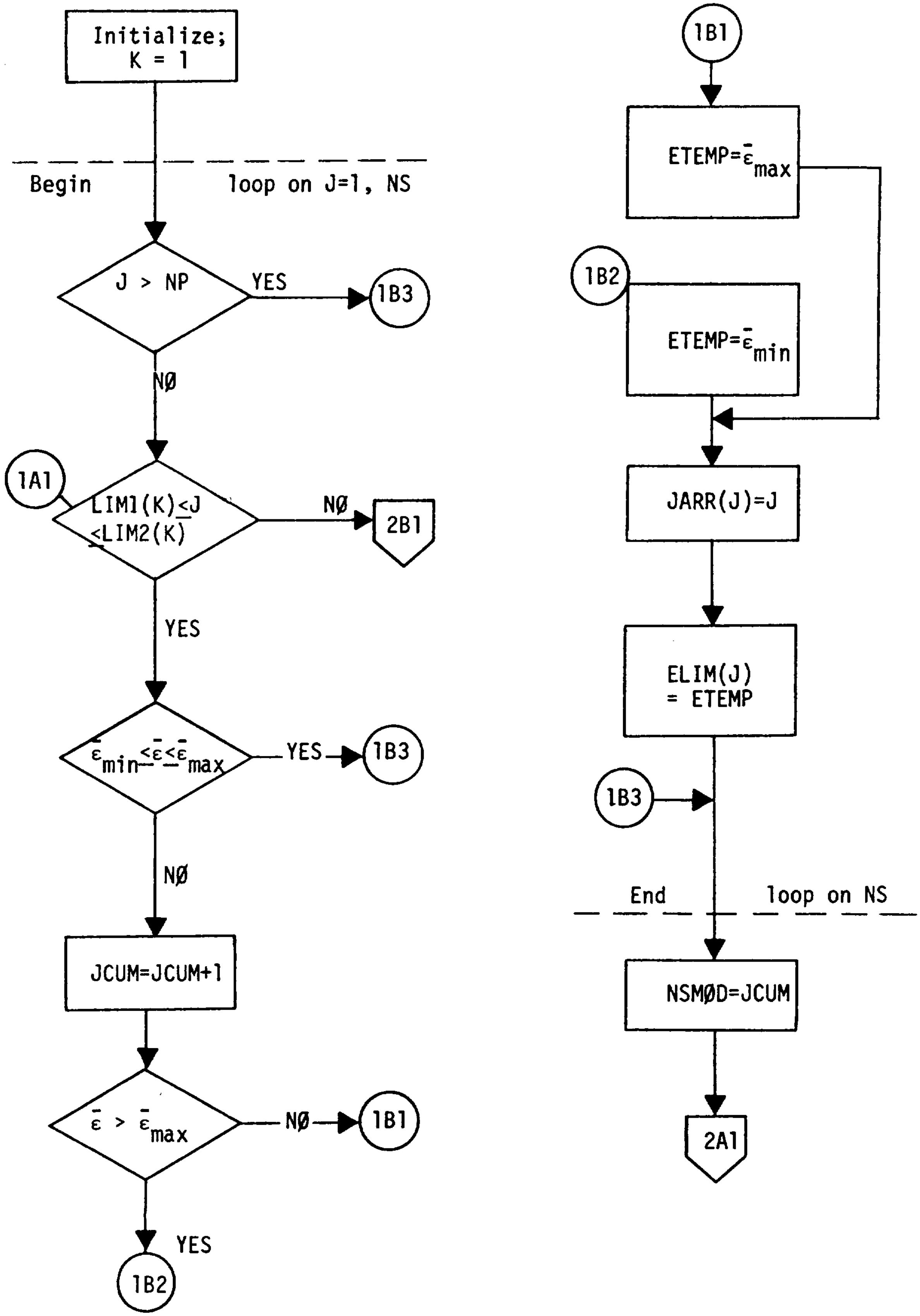
This subroutine compares the correction factors, $\bar{\varepsilon}$, with the input limits $\bar{\varepsilon}_{min}$, $\bar{\varepsilon}_{max}$. If any $\bar{\varepsilon}$ falls outside of the limits it replaces $\bar{\varepsilon}$ with the closest limit. (The values of $\bar{\varepsilon}$ are correction factors if correction factor modes do not exist). This subroutine forms the final correction factor array $\bar{\bar{\varepsilon}}$ (see paragraph below Eq. (60)).

Description of Argument List

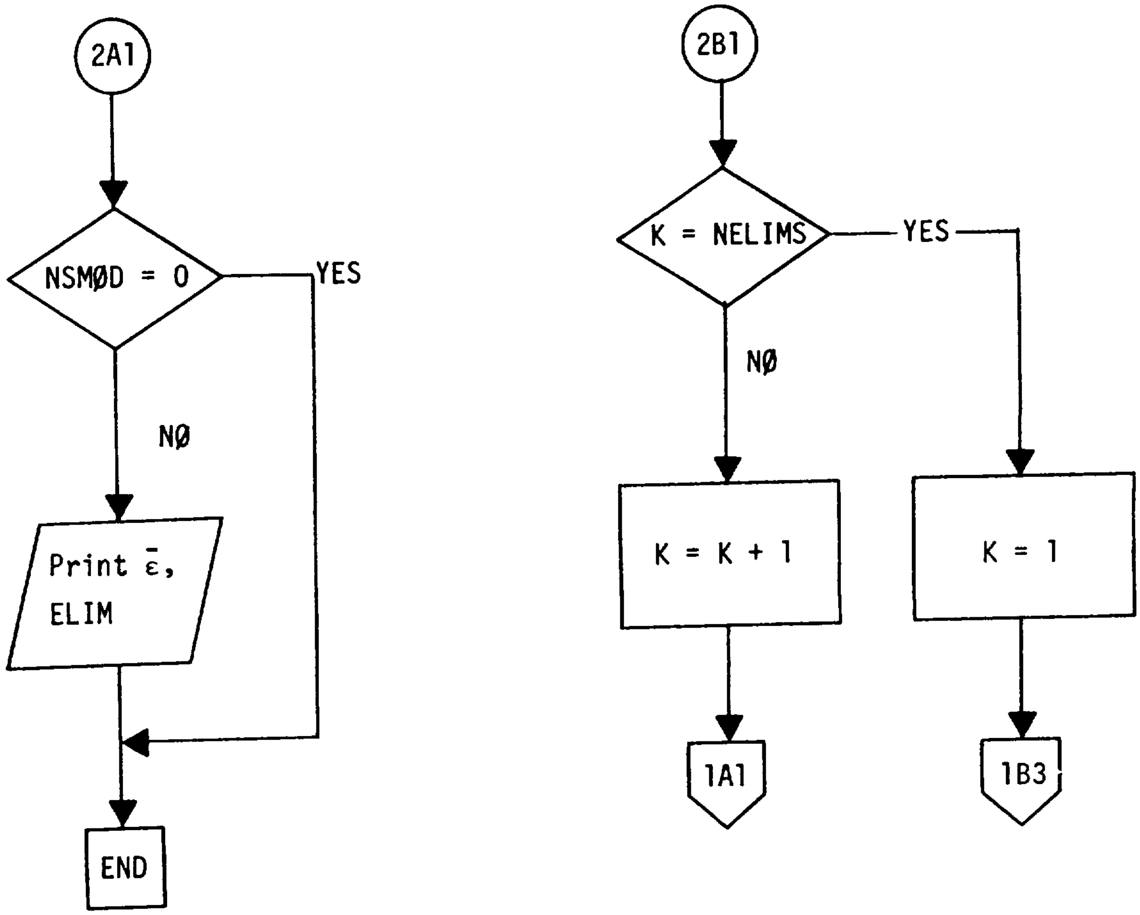
NPØT	Data set number of the system output data set
NELIMS	Number of input cards for EBMIN and EBMAX - see below
NP	Number of row elements in the ΔC_p matrix
NS	NS = NP + NC when NEM $\leq$ NP NS = NEM + NC when NEM > NP
LIMK	A two-dimensional array containing the first- and last box numbers that define a range of boxes (or ΔC_p) over which a limit is to be placed on ε
JARR	Array of the box numbers for which the ε values are modified
NSMØD	The number of ε values which are modified due to the limits placed on these
EBMIN EBMAX	The minimum- and maximum value allowed for the values of ε for boxes (or ΔC_p) in the range defined by LIMK
EB	Array of the calculated $\bar{\varepsilon}$ values
ELIM	Array of the ε values that were modified due to the $\bar{\varepsilon}_{min}$, $\bar{\varepsilon}_{max}$ restrictions

Calling Subroutine WEYT

Flow Chart


Initialize;
K = 1
Begin
loop on J=1, NS
J > NP
YES
1B3
NØ
1A1
LIM1(K)≤J
≤LIM2(K)
NØ
2B1
YES
ε̄min≤ε̄≤ε̄max
YES
1B3
NØ
JCUM=JCUM+1
ε̄ > ε̄max
NØ
1B1
YES
1B2
1B1
ETEMP=ε̄max
1B2
ETEMP=ε̄min
JARR(J)=J
ELIM(J)
= ETEMP
1B3
End
loop on NS
NSMØD=JCUM
2A1

Flow Chart



2A1
NSMØD = 0
YES
NØ
Print $\bar{\varepsilon}$,
ELIM
END
2B1
K = NELIMS
YES
NØ
K = K + 1
K = 1
1A1
1B3

SUBRØUTINE EPSJ(NTPHIJ, NP, NEM, NS, EB, EPS, PHI)

Functional Description

This subroutine relates ε to $\bar{\varepsilon}$, as in Equation (53).

$$\{\varepsilon\} = [\phi]\{\bar{\varepsilon}\}$$

where $[\phi]$ are correction factor modes and where

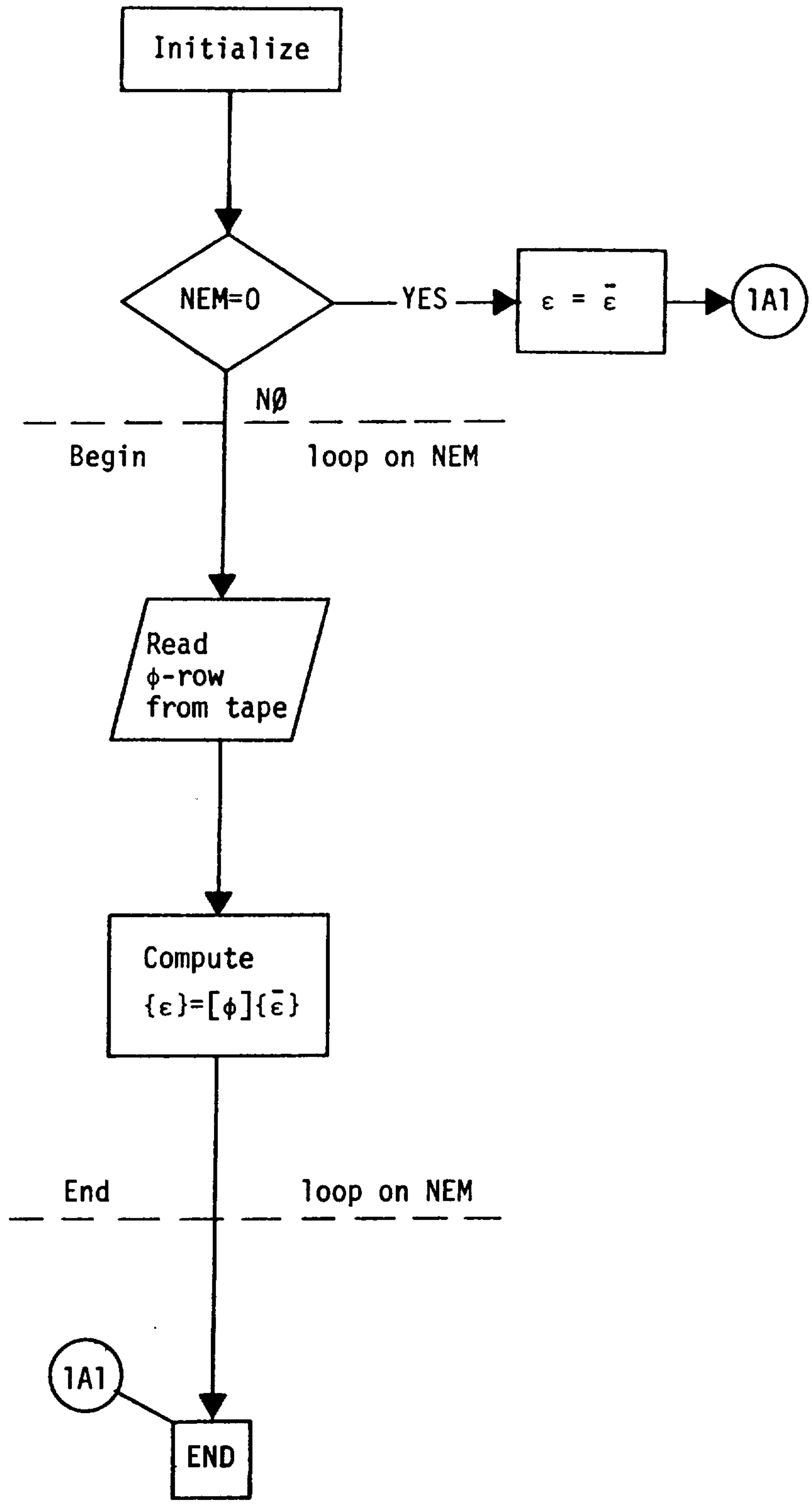
$$\varepsilon = \begin{cases} \varepsilon_p \text{ for premultiplying correction factors} \\ \varepsilon_w \text{ for postmuliplying correction factors} \end{cases}$$

Description of Argument List

NTPHIJ	Tape number containing the ϕ matrix
NP	Number of row elements in the ϕ matrix
NEM	Number of correction factor modes
NS	NS = the greater of (NP+NC) and (NEM+NC)
EB	Array of the $\bar{\varepsilon}$ values
EPS	The final ε array
PHI	A column of the matrix of weight factor mode shapes, ϕ

Calling Subroutine WEYT

Flow Chart


Initialize
NEM=0
YES
ε = ε̄
1A1
NØ
Begin
loop on NEM
Read
ϕ-row
from tape
Compute
{ε}=[ϕ]{ε̄}
End
loop on NEM
1A1
END

SUBRØUTINE GINV(NPØT, NTAPSB, NC, NS, NX, DC, EB, B, S, SBB)

Functional Description

This subroutine provides a general inverse of the following set of equations:

$$NC\updownarrow \quad \{\Delta C_e\} = [\tilde{\bar{S}}]\{\tilde{\varepsilon}\} \updownarrow NS$$

When NC = NS (Direct Solution)

$$\{\tilde{\varepsilon}\} = [\tilde{\bar{S}}]^{-1}\{\Delta C_e\}$$

When NC < NS (Minimization Solution, $\sum \tilde{\varepsilon}^2$ = min (see Eq. (20))

$$\{\tilde{\varepsilon}\} = [\tilde{\bar{S}}]^H\{\lambda\}$$

$$\{\lambda\} = [[\tilde{\bar{S}}][\tilde{\bar{S}}]^H]^{-1}\{\Delta C_e\}$$

When NC > NS (Least Squares Solution $\sum \Delta C_e^2$ = min.)

$$\{\tilde{\varepsilon}\} + [[\tilde{\bar{S}}]^H[\tilde{\bar{S}}]]^{-1}\{\Delta C_e\}$$

$$\{\lambda\} = [\tilde{\bar{S}}]^H\{\Delta C_e\}$$

In the above;

$$[\tilde{\bar{S}}] = [\bar{\bar{S}}][\sqrt{T_p}]^{-1}$$

where $[\bar{\bar{S}}]$ given in Eq. (51). The term $\tilde{\varepsilon}$ is given in Eq. (22). The superscript H indicates the Hermitian transpose.

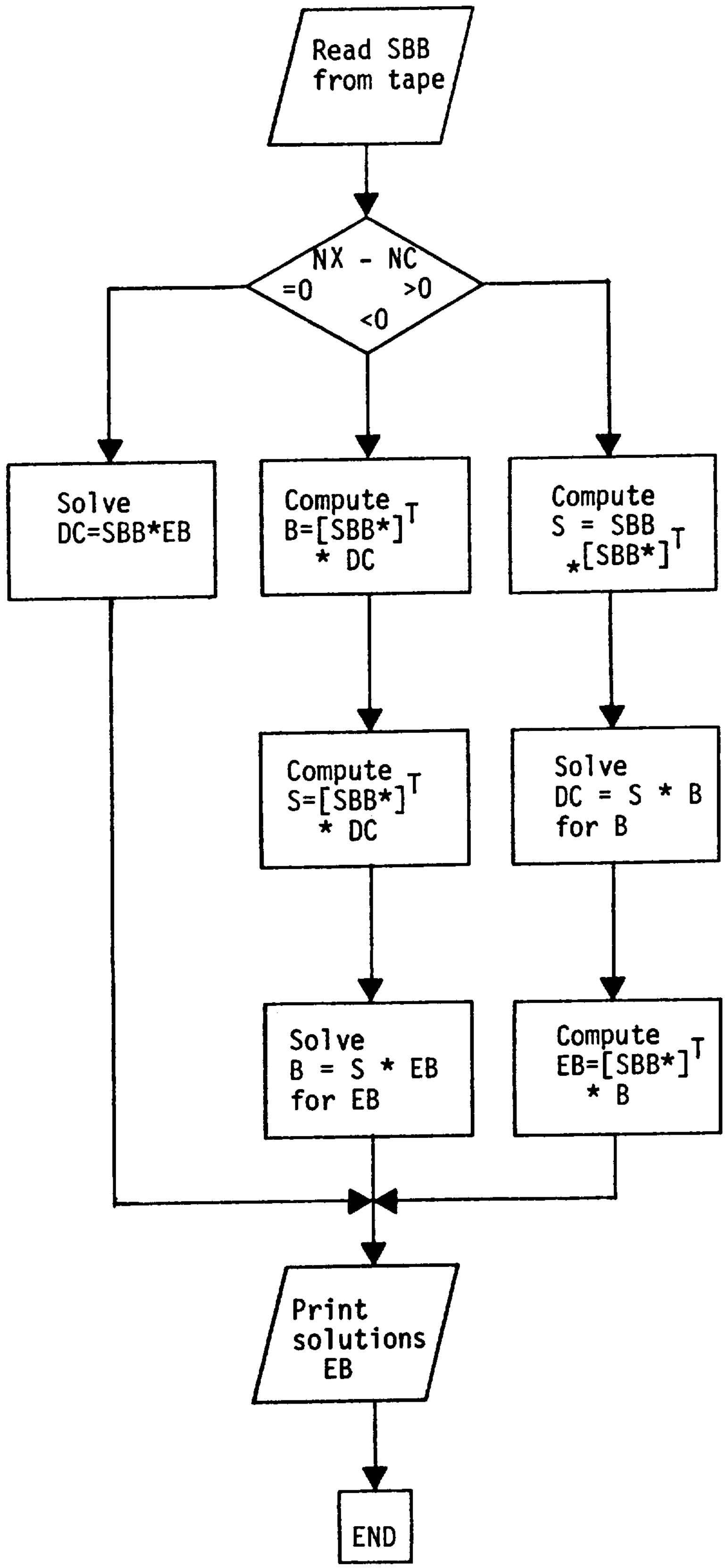
Description of Argument List

NPØT	Data set number of the system output data set
NTAPSB	Tape number containing the $\tilde{S}$ matrix
NC	Number of constraints; number of rows in the $\tilde{S}$ matrix (SBB)
NS	NS = the greater of (NP+NC) and (NEM+NC); number of columns in the $\tilde{S}$ matrix
NX	NX=NS if NEM=0, NX=NEM + NC otherwise
DC	The complex ΔC_e array
EB	The complex array $\{\tilde{\varepsilon}\}$, output of subroutine GINV
B	An array of intermediate solutions (complex)
S	A complex two-dimensional work array of dimension NC by NC
SBB	The complex NC by NS matrix, $\tilde{S}$

Calling Subroutine   WEYT

Called Subroutine   MIS2, the standard IBM system subroutine for solving complex matrix equations

Flow Chart



Read SBB from tape
NX - NC
=0
>0
<0
Solve DC=SBB*EB
Compute B=[SBB*]T * DC
Compute S = SBB *[SBB*]T
Compute S=[SBB*]T * DC
Solve DC = S * B for B
Solve B = S * EB for EB
Compute EB=[SBB*]T * B
Print solutions EB
END

SUBRØUTINE MATM(NT, IGØ, NR, NC, NMAX, A, C, B)

Functional Description

Subroutine MATM is essentially a matrix multiplication routine. It obtains the DI matrix ([A]) rows from tape NT and the W matrix from the two-dimensional array B. The results of the matrix multiplication, ΔC_{p_t}, are saved in array B which is returned to the calling routine, DCPB, via the argument list.

Description of Argument List

NT	Tape number containing the inverse-D matrix
IGØ	1 for symmetric modes
	2 for antisymmetric modes
NR	Number of rows in the ΔC_{p_t} matrix
NC	Number of columns in the ΔC_{p_t} matrix
NMAX	Maximum number of columns in the ΔC_{p_t} matrix
A	A row of the inverse-D matrix, [A]
C	Complex work arry
B	Two-dimensional complex array in which the ΔC_{p_t} matrix is stored

Calling Subroutine DCPB

SUBRØUTINE MØDF(NC, NS, MASTSB, NEWTSB, JARR, SQRTT, ELIM, SBB, DCI, DCMØD)

Functional Description

When some of the values of $\bar{\varepsilon}$ have exceeded their limits and have been replaced by the limit values, these new values of $\bar{\varepsilon}$ (called ε_d in Equation (56) are then considered fixed and known. However the constraints are now not satisfied and a change in the constraint ΔC_e, i.e., ΔC_{mod}, is calculated (see Equation (59)).

$$\Delta C_{mod} = \Delta C_e - [\bar{\bar{S}}_d] \{\varepsilon_d\}$$

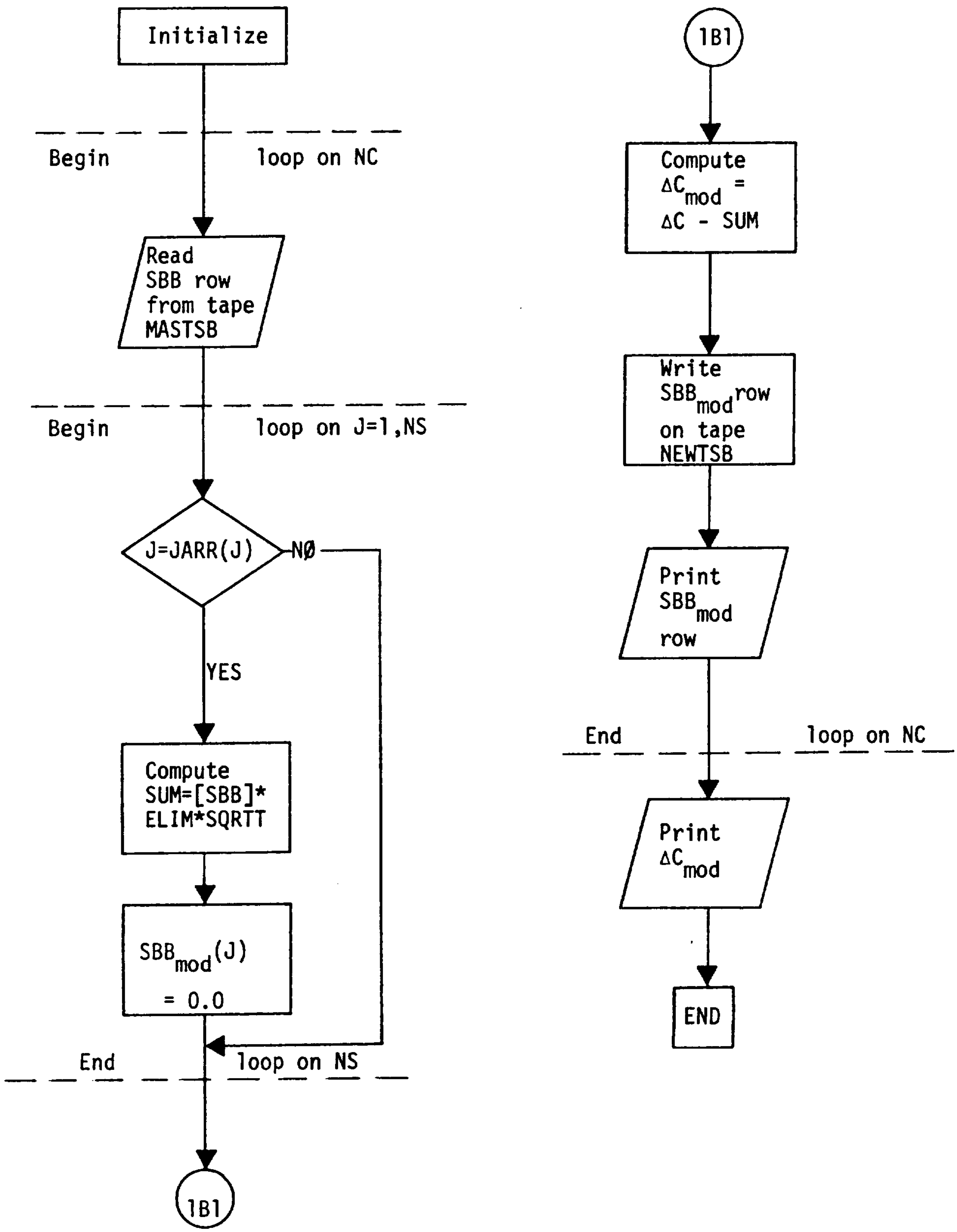
Since the new values of $\bar{\varepsilon}$, i.e., ε_d, can not influence solution further the $\bar{\bar{S}}$ matrix must be changed to delete the influence of ε_d. Thus the elements of $\bar{\bar{S}}$ that give the influence of ε_d, i.e., $\bar{\bar{S}}_d$, must be eliminated resulting in $\bar{\bar{S}}_u$. This subroutine forms $\bar{\bar{S}}_u$, or in the notation of the computer program $[\bar{\bar{S}}_{mod}]$.

Description of Argument List

NC	Number of constraints - dimension of the complex arrays DCI and DCMØD
NS	Dimension of the complex array SBB
MASTSB	Tape number containing SBB arrays (i.e., rows of the $\bar{\bar{S}}$ matrix)
NEWTSB	Tape number containing the modified SBB arrays (i.e. rows of the $\bar{\bar{S}}_{mod}$ matrix) $\bar{\bar{S}}_{mod} = \bar{\bar{S}}_u$
JARR	Array of the element numbers for which the $\bar{\bar{S}}$-values are replaced by zeroes
SQRTT	$\sqrt{T_j}$ - see Equations (23) and (34)
ELIM	ε_{limj}, array of the modified ε values
SBB	Complex array containing rows of the $\bar{\bar{S}}$ matrix
DCI	Complex array containing ΔC_e
DCMØD	Complex array containing ΔC_{mod}

Calling Subroutine WEYT

Flow Chart


Initialize
Begin loop on NC
Read SBB row from tape MASTSB
Begin loop on J=1,NS
J=JARR(J)
NØ
YES
Compute SUM=[SBB]* ELIM*SQRTT
SBB$_{mod}$(J) = 0.0
End loop on NS
1B1
1B1
Compute ΔC$_{mod}$ = ΔC - SUM
Write SBB$_{mod}$ row on tape NEWTSB
Print SBB$_{mod}$ row
End loop on NC
Print ΔC$_{mod}$
END

SUBRØUTINE PHIJ(NPIT, NPØT, NTPHIJ, NEM, NP, KØDE, MØDES, X, Y, Z, PHI)

Functional Description

This subroutine forms the correction factor modes. If ϕ is input element by element (TYPE = 0) then this subroutine simply arranges the data into arrays. If TYPE = 0 then modes are calculated as follows:

$(x_j - a_\ell)^{n_\ell}$	TYPE = 1
$(y_j - a_\ell)^{n_\ell}$	2
$(z_j - a_\ell)^{n_\ell}$	3
$\exp[b_\ell(x_j - a_\ell)^{n_\ell}]$	4
$\exp[b_\ell(y_j - a_\ell)^{n_\ell}]$	5
$\exp[b_\ell(z_j - a_\ell)^{n_\ell}]$	6

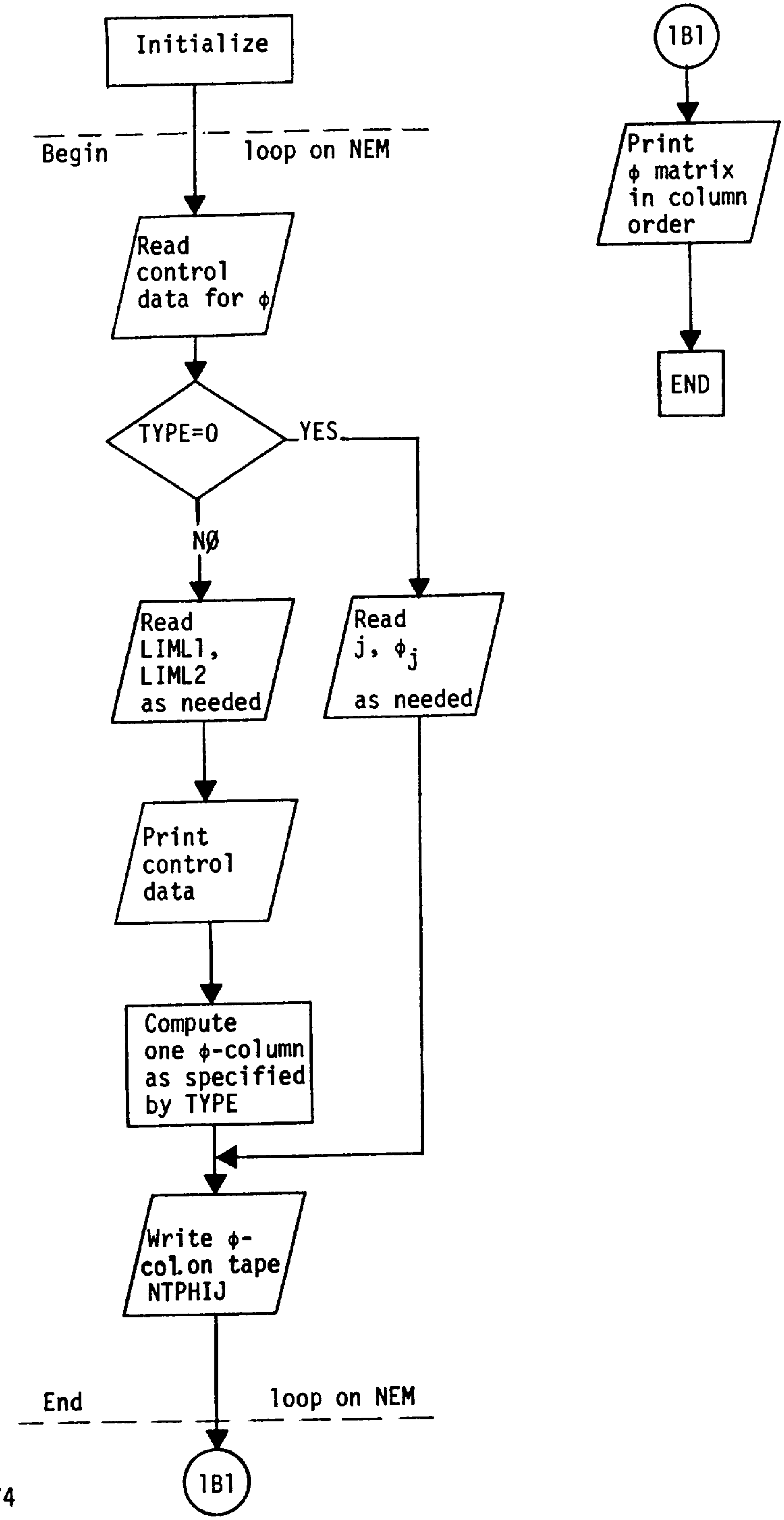
where a b n are input per mode and where $\phi_{j\ell} = 0$ over boxes are not considered.

Description of Argument List

NPIT	Data set number of the system input data set
NPØT	Data set number of the system output data set
NTPHIJ	Tape number containing columns of the NP by NEM ϕ matrix
NEM	Row dimension of the ϕ matrix
NP	Column dimension of the ϕ matrix
KØDE	-1 (end indicator of card input sets)
MØDES	not used
X	x
Y	y coordinates of the pressure points or
Z	z of the ΔC_p's
PHI	Complex array containing one column of the ϕ matrix

Calling Subroutine WEYT

Flow Chart


Initialize
Begin loop on NEM
Read control data for ϕ
TYPE=0
YES
NØ
Read LIML1, LIML2 as needed
Read j, ϕ_j as needed
Print control data
Compute one ϕ-column as specified by TYPE
Write ϕ-col.on tape NTPHIJ
End loop on NEM
1B1
1B1
Print ϕ matrix in column order
END

SUBRØUTINE PØSN(NT, IGØ)

Functional Description

This subroutine positions tapes of a certain uniform format for reading; see Tape Description for NTDCP, NTAPW and NTAPDI.

Description of Argument List

NT	Tape number to be positioned for reading
IGØ	1 for symmetric modes, 2 for antisymmetric modes

Calling Subroutines DCPB, DCPT, MATM, SBAR

SUBRØUTINE RECD(NTAPE, A, N)

Functional Description

This subroutine reads arrays of real numbers A of length N from tape NTAPE one record at a time. It is used for the reading of the geometry arrays when these are input from tape NTGEØM.

Description of Argument List

NTAPE	Tape number
A	Array to be read from tape
N	Length of array A

Calling Subroutine WEYT

SUBRØUTINE SAIJ(NPIT, NPØT, NTSAIJ, NTSANJ, NC, NP, NMØN, NAXIS,
AIT, CIE, X, Y, Z, CG, SG, DELA, FLAGA, FLAGF,
KØDE, IPRINT, LABEL, SAI)

Functional Description

This subroutine sets up proper argument lists for SROW so that integration matrices, [S], can be calculated for both constraining and monitoring purposes.

Description of Argument List

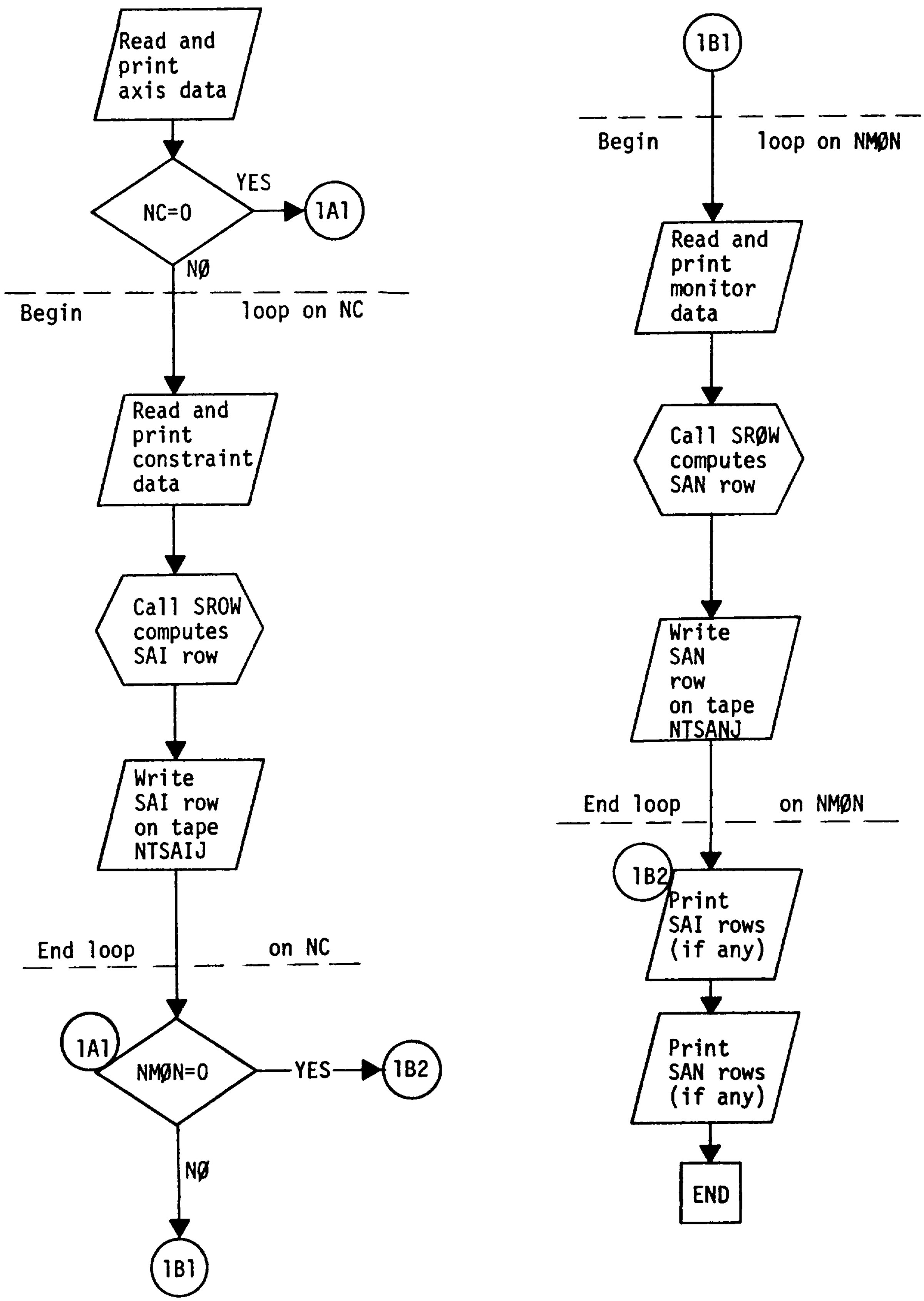
NPIT	Data set number of the system input data set
NPØT	Data set number of the system output data set
NTSAIJ	Tape number containing the integration matrix rows, SA_{ij}, for constraints
NTSANJ	Tape number containing the inegration matrix rows, SA_{nj}, for monitoring
NC	Number of constraints
NP	Number of ΔC_p elements
NMØN	Number of sets of the monitoring data
NAXIS	Number of axes
AIT	Constraining effectiveness of the experimental data, $\tilde{a}_i$
CIE	Experimental (or any other) constraint on the data, C_e
X, Y, Z	Coordinates of the pressure points (boxes)
CG, SG	Cosine-, sine of box dihedral angles
DELA	Box areas
FLAGA	Axis flag = 0, axis endpoints are input = 1, direction cosines are input
FLAGF	Force/moment flag = 0, C_e is a force in the direction of axis = 1, C_e is a moment about axis

KØDE	= -1
IPRINT	Detail print flag IPRINT = 1, print SA_{ij} and SA_{nj} rows IPRINT = 0, bypass print
LABEL	Alphameric identifier of the integrated parameters
SAI	Complex array containing a row of either one of the integration matrices SA_{ij} or SA_{nj}

<u>Calling Subroutine</u> WEYT

<u>Called Subroutine</u> SRØW

Flow Chart



Read and print axis data
NC=0
YES
1A1
NØ
Begin
loop on NC
Read and print constraint data
Call SROW computes SAI row
Write SAI row on tape NTSAIJ
End loop
on NC
1A1
NMØN=0
YES
1B2
NØ
1B1
1B1
Begin
loop on NMØN
Read and print monitor data
Call SRØW computes SAN row
Write SAN row on tape NTSANJ
End loop
on NMØN
1B2
Print SAI rows (if any)
Print SAN rows (if any)
END

SUBRØUTINE SBAR(NTSBIJ, NTAPDI, NC, NP, NS, FLAGP, FLAGT, FLAGW, I, IGØ, SQRTT, AIW, SAI, DI, W, DELCPB, SBI)

Functional Description

This subroutine solves for the matrix SBI where

$$SBI = [\bar{S}] \lceil \sqrt{T^*} \rfloor^{-1}$$

The matrix $[\bar{S}]$ contains all the capability of the program except modes and limits. This capability is outlined in Eqs. (45), (26) and (9) for premultipliers and (45), (39) and (30) for postmultipliers. The weights T^* are defined below Equation (47).

$$SBI = \begin{cases} \overline{SA}_{ij} & \text{when } j = 1, 2\cdots NP,\ i = 1, 2\cdots NC \\ WT_i & \text{when } j = NP + i \\ 0 & \text{otherwise} \end{cases}$$

where NP = number of pressure values and NC = number of constraints.

$$\overline{SA}_{ij} = \begin{cases} SA_{ij}\, \Delta C_{p_j} \Big/ \sqrt{T_j} & \text{when } \Delta C_p \text{ available} \\ \sum SA_{i_\ell}\, DI_{\ell_j}\, W_j \Big/ \sqrt{T_j} & \text{when } \Delta C_{p_j} \text{ not available} \end{cases}$$

$$WT_i = \begin{cases} (1 - \tilde{a}_i)/\tilde{a}_i & \tilde{a}_i > 0.0001 \\ 10^4 & \tilde{a}_i \leq .0001 \end{cases}$$

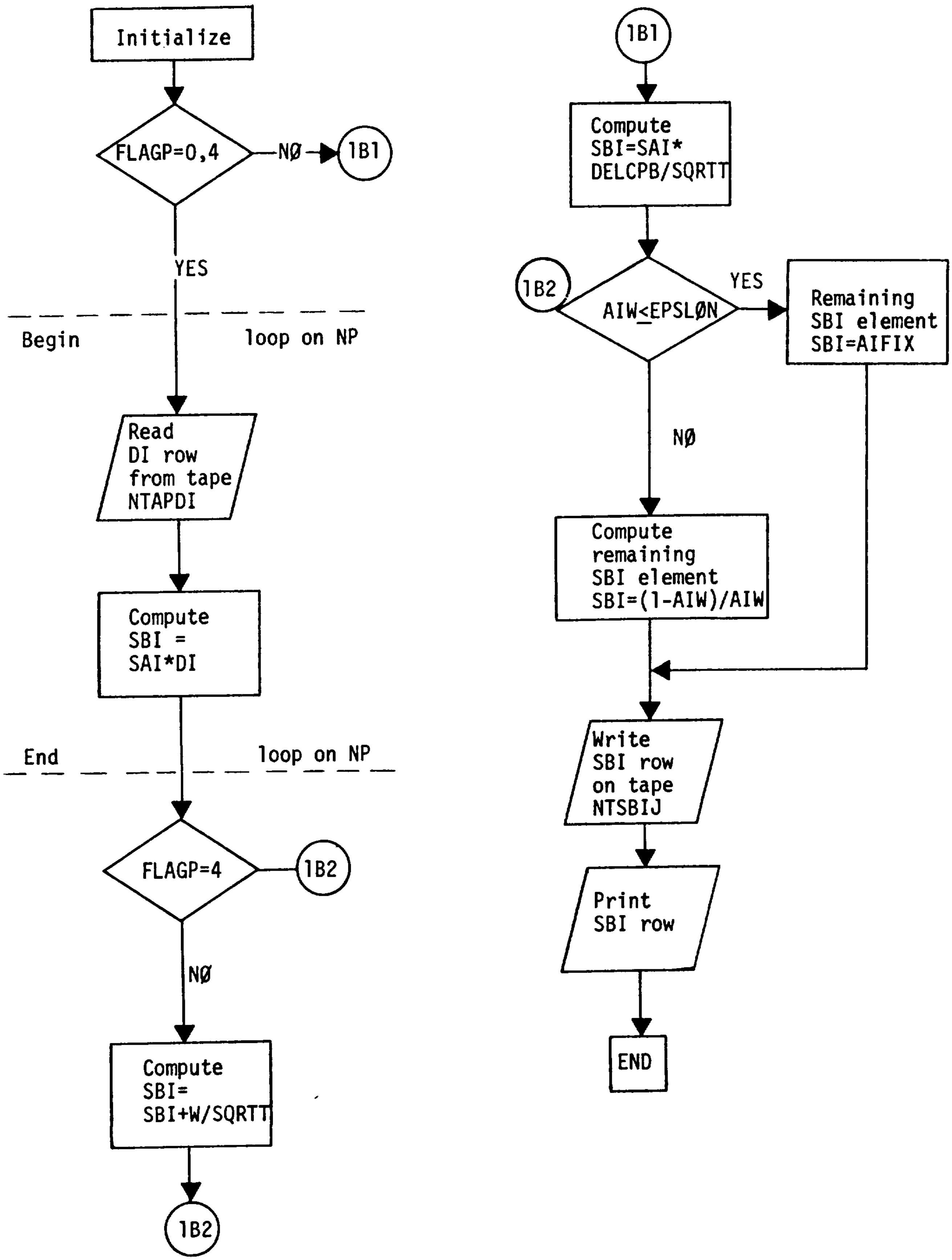
$$T_j = \begin{cases} \left.\begin{array}{ll} |\Delta C_{p_t}|\ \Delta A & FLAGP \neq 1 \\ \text{or} & \\ |W| & FLAGP = 1 \end{array}\right\} & FLAGT = 0 \\ 1.0 \qquad\qquad FLAGT = 1 & \end{cases}$$

Description of Argument List

NTSBIJ	Tape number containing rows of the integration matrix without weight factor modes, $\bar{S}$
NTAPDI	Tape number containing rows of the inverse-D matrix
NC	Number of constraints
NP	Number of ΔC_p elements
NS	Length of the SBI matrix rows
FLAGP	Option flag for ΔC_p and/or pre- or post-multiplying correction factors
FLAGT	Option flag for weights
FLAGW	Option flag for normalwash input
I	An intermediate index
IGØ	1 for symmetric modes 2 for antisymmetric modes
SQRTT	$= \sqrt{T}$; see equations
AIW	Constraining effectiveness $\tilde{a}_i$
SAI	A row of the integration matrix S'_{ij}
DI	A row of the inverse-D matrix
W	A column of the normalwash matrix
DELCPB	A column of the ΔC_{p_e} matrix (lifting pressure coefficients, either input, ΔC_p, or computed as $[D]^{-1}\{W\}$)
SBI	A row of the integration matrix without weight factor modes, $[\bar{S}]\lceil\sqrt{T^*}\rfloor^{-1}$, $T^* = \begin{cases} T & \text{for constraints} \\ \frac{1-\tilde{a}}{\tilde{a}} & \text{for estimates} \end{cases}$

Calling Subroutine SDBL

Flow Chart



Initialize
FLAGP=0,4
NØ
1B1
YES
Begin
loop on NP
Read
DI row
from tape
NTAPDI
Compute
SBI =
SAI*DI
End
loop on NP
FLAGP=4
1B2
NØ
Compute
SBI=
SBI+W/SQRTT
1B2
1B1
Compute
SBI=SAI*
DELCPB/SQRTT
1B2
AIW≤EPSLØN
YES
Remaining
SBI element
SBI=AIFIX
NØ
Compute
remaining
SBI element
SBI=(1-AIW)/AIW
Write
SBI row
on tape
NTSBIJ
Print
SBI row
END

SUBROUTINE SDBL(NSCRCH, NUTL, MASTSB, NTPHIJ, NTAPW, NTSAIJ,
NTAPDI, IGØ, FLAGW, FLAGP, FLAGT, NC, NP, NS,
NEM, SQRTT, AIT, DELA, SBB, SBI, SAI, DI, W, PHI, DELCP)

Functional Description

This subroutine calculates $[\bar{\bar{S}}]$ described in Equation (54). The quantity calculated in this routine includes estimates and thus

$$[\bar{\bar{S}}] = [SBB] = [\bar{S}] \left[\begin{array}{c|c} \phi & 0 \\ \hline 0 & I \end{array}\right]$$

Description of Argument List

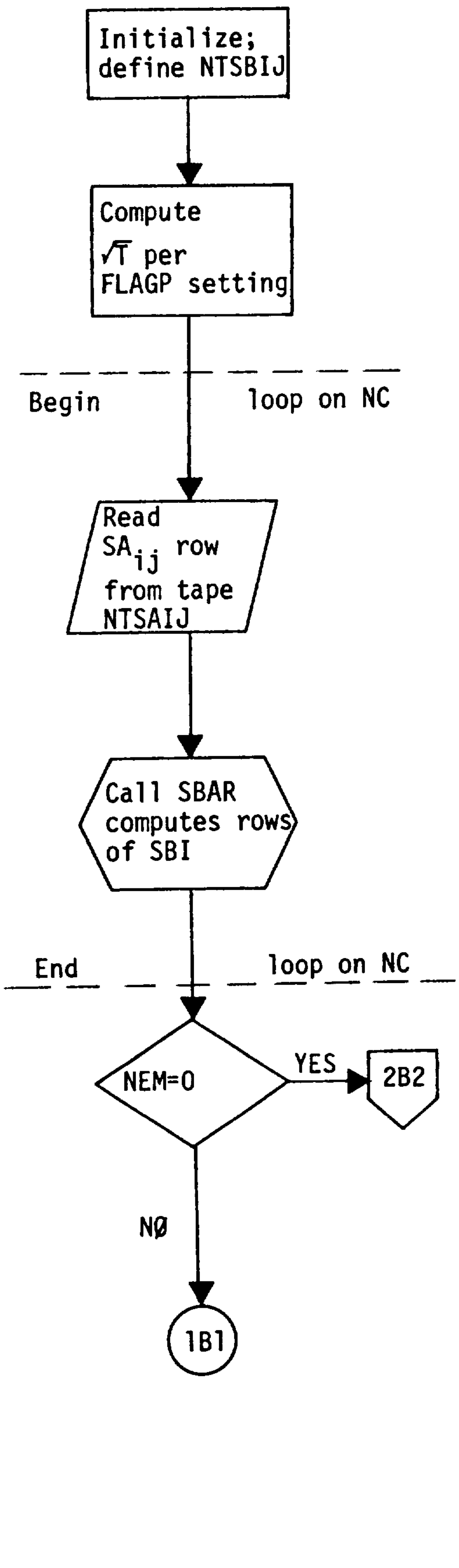
NSCRCH	Tape number containing columns of the $\bar{\phi}$ matrix (if any)
NUTL	Utility (scratch) tape number
MASTSB	Tape number containing the $\bar{\bar{S}}$ matrix rows
NTPHIJ	Tape number containing the ϕ matrix columns
NTAPW	Tape number containing columns of the normalwash matrix
NTSAIJ	Tape number containing rows of the integration matrix, SA_{ij}
NTAPDI	Tape number containing rows of the inverse-D matrix
IGØ	1 for symmetric modes 2 for antisymmetric modes
FLAGW	Option flag for normalwash input
FLAGP	Option flag for ΔC_p input and/or pre- or post-multiplying corrections
FLAGT	Option flag for weights
NC	Number of constraints
NP	Number of ΔC_p elements
NS	= max (NP+NC, NEM+NC)
NEM	Number of correction factor modes
SQRTT	$\sqrt{T_j}$, see equations
AIT	Constraining effectiveness $\tilde{a}_i$
DELA	Box areas
SBB	A row of the $\bar{\bar{S}}$ matrix (integration matrix with weight factor modes)
SBI	A row of the $[\bar{S}] \lceil \sqrt{T^*} \rfloor^{-1}$ matrix
SAI	A row of the S_{ij} elements

DI	A row of the inverse-D matrix
W	A column of the normalwash matrix
PHI	A column of the ϕ matrix (weight factor mode shapes)
DELCPB	A column of the ΔC_{p_t} matrix

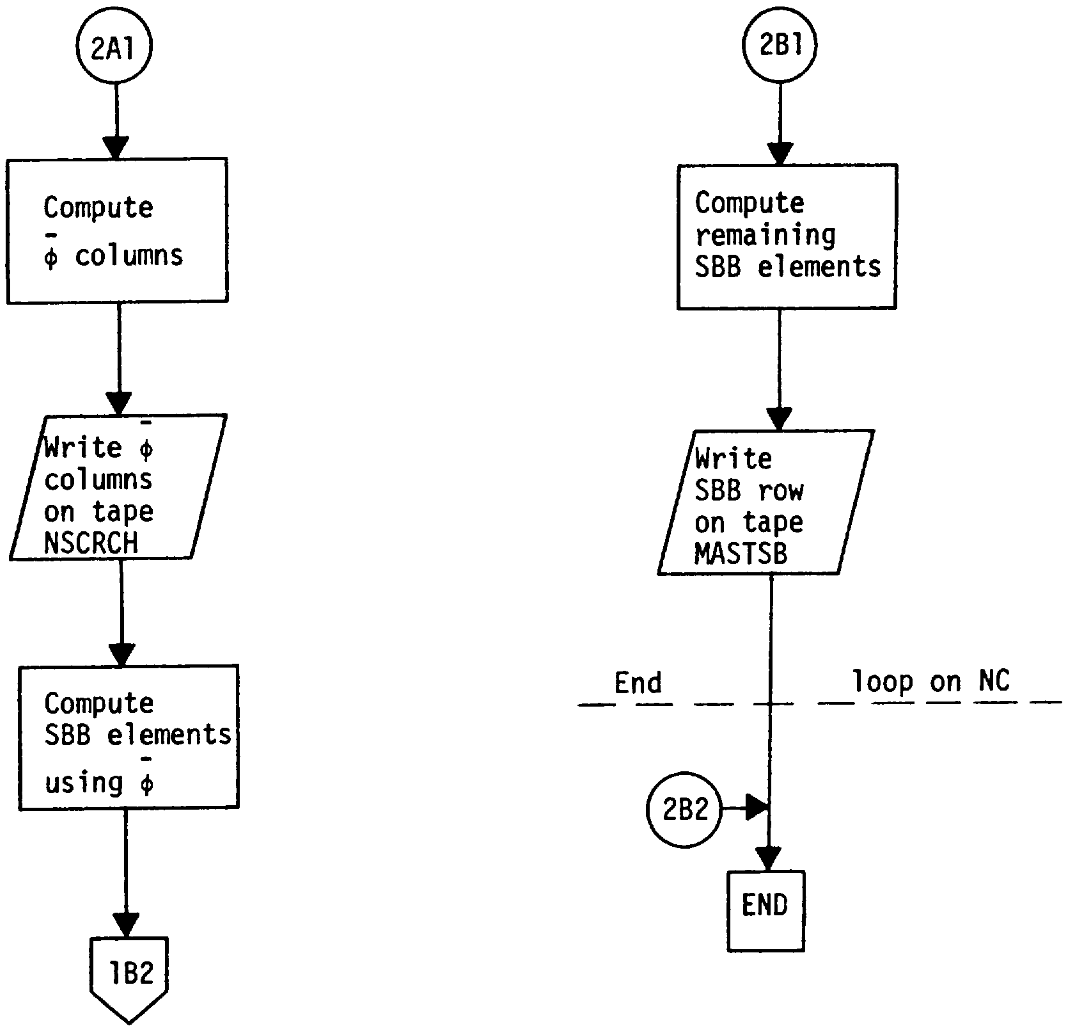
<u>Calling Subroutine</u> WEYT

<u>Called Subroutine</u> SBAR

Flow Chart


Initialize; define NTSBIJ
Compute
√T per
FLAGP setting
Begin
loop on NC
Read
SA_ij row
from tape
NTSAIJ
Call SBAR
computes rows
of SBI
End
loop on NC
NEM=0
YES
2B2
NØ
1B1
1B1
Begin
loop on NC
Initialize
SBB row
Read SBI
row from
tape
NTSBIJ
Begin
loop on NEM
Read
φ-column
from tape
NTPHIJ
FLAGP = 4
YES
2A1
NØ
Compute
SBB elements
using φ
1B2
End
loop on NEM
2B1

Flow Chart


2A1
Compute
$\bar{\phi}$ columns
Write $\bar{\phi}$
columns
on tape
NSCRCH
Compute
SBB elements
using $\bar{\phi}$
1B2
2B1
Compute
remaining
SBB elements
Write
SBB row
on tape
MASTSB
End loop on NC
2B2
END

SUBRØUTINE SRØW(FLAGA, FLAGF, XI1, ETA1, ZETA1, CG, SG, CTIL, X, Y, Z, DELA, LIMI, IIMAX, I, NP, IR, XI2, ETA2, ZETA2, SAI)

Functional Description

This subroutine constructs the integration matrix [S] described in Equation (3) a row at a time.

$$S_{ij} = \text{SAIJ} = \begin{cases} A_{ij}\ \Delta A_j & \text{for force calc.} \\ B_{ij}\ \Delta A_j & \text{for moment calc.} \end{cases}$$

$$A_{ij} = [\ -\cos\beta_i\ \sin\bar{\gamma}_j + \cos\gamma_i\ \cos\bar{\gamma}_j]/\tilde{c}_i$$

$$\begin{aligned} B_{ij} = \{\cos\alpha_i\ &[(y_j-\eta_i^{(1)})(\cos\bar{\gamma}_j) + (z_j-\zeta_i^{(1)})\sin\bar{\gamma}_j] \\ &- \cos\beta_i\ (x_j-\xi_i^{(1)})\cos\bar{\gamma}_j \\ &- \cos\gamma_i (x_j-\xi_i^{(1)})\sin\bar{\gamma}_j\}/\tilde{c}_i \end{aligned}$$

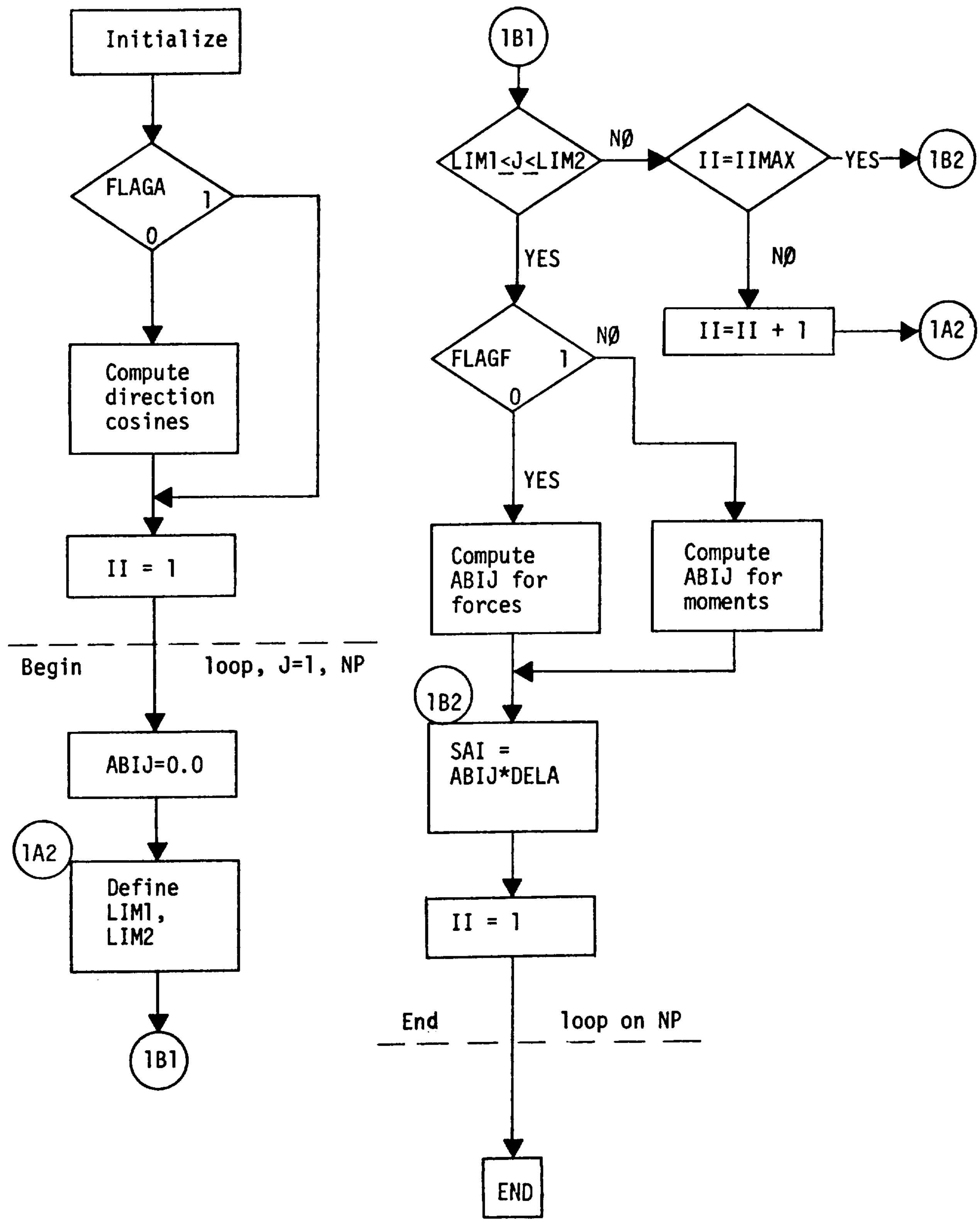
where $\cos\alpha_i$, $\cos\beta_i$, $\cos\gamma_i$ are the direction cosines of the input axis and where x_j, y_j, z_j, $\bar{\gamma}_j$ are the coordinates and dihedral of the aerodynamic box and $\xi_i^{(1)}$ $\eta_i^{(1)}$ $\zeta_i^{(1)}$ are the coordinates of the one edge of the input axis. SAIJ is of course zero on boxes that are not to be integrated.

Description of Argument List

FLAGA	Axis input option flag
	= 0, axis endpoints are input
	= 1, direction cosines are input
FLAGF	Force/moment flag
	= 0, $C_i^{(e)}$ is a force in direction of axis
	= 1, $C_i^{(e)}$ is a moment about axis
XI1, ETA1, ZETA1	Axis endpoint coordinates, $\xi^{(1)}$, $\eta^{(1)}$, $\zeta^{(1)}$
CG, SG	Cosine-, sine of box dihedral angles
CTIL	Constant used to nondimensionalize integrated data ($\tilde{c}$)
X, Y, Z	Coordinates of the pressure points (boxes)
DELA	Box areas
LIMI	First-, last box numbers for the integration of the ΔC_p values
IIMAX	Number of LIMI sets input for one constraint
I	Intermediate index
NP	Number of ΔC_p values
IR	Row index of the S_{ij} matrix
XI2, ETA2, ZETA2	Second axis endpoint coordinates when FLAGF = 0, direction cosines when FLAGF = 1
SAI	A row of the integration matrix S

Calling Subroutine SAIJ

Flow Chart


Initialize
FLAGA
1
0
Compute direction cosines
II = 1
Begin loop, J=1, NP
ABIJ=0.0
1A2
Define LIM1, LIM2
1B1
1B1
LIM1≤J≤LIM2
NØ
II=IIMAX
YES
1B2
YES
NØ
FLAGF
1
0
NØ
II=II + 1
1A2
YES
Compute ABIJ for forces
Compute ABIJ for moments
1B2
SAI = ABIJ*DELA
II = 1
End loop on NP
END

SUBRØUTINE WEYT(NP, NC, NEM, NELIMS, NMØN, NAXIS, NMIN, NMAX, NS, NPIT, NPØT, W, DI, DCP, EPS, PHI, SAI, DCPTIL, DELCPB, CØL, CIE, DCMØD, EBMIN, EBMAX, EB, ELIM, SBB, SBI, S, SBMAT, DCI, WØRK)

Functional Description

This subroutine is the core of the correction factor method. All logic for the method is established here. This subroutine uses input to decide what is to be done and sets up the argument lists for and executes the calls to all required subroutines. The following flow charts document the logic flow of this subroutine. This subroutine sets up the logic for various forms of input data and various types of calculations. The input data ranges over geometry, pressures, downwashes, aerodynamic influence matrices, previously generated correction factors, integration matrix data, etc. This data can enter the program by cards, tapes or both.

There are three basic computational branches; (1) correction factor calculation, (2) monitoring of data (integration of pressures into aerodynamic parameters) and (3) application of previously generated correction factors to pressure distributions. Within branch (1) there exists a choice of what type of correction factors to generate, premultiplier, postmuliplier and new postmultiplier. Also a choice as to the type of weighting to be used (i.e. the T) is available. The program also tests to see if limits are placed on the correction factors and if modes are used. The constraining power $\tilde{a}$ is always input since a constraint is simply $\tilde{a} = 1.0$.

Description of Argument List

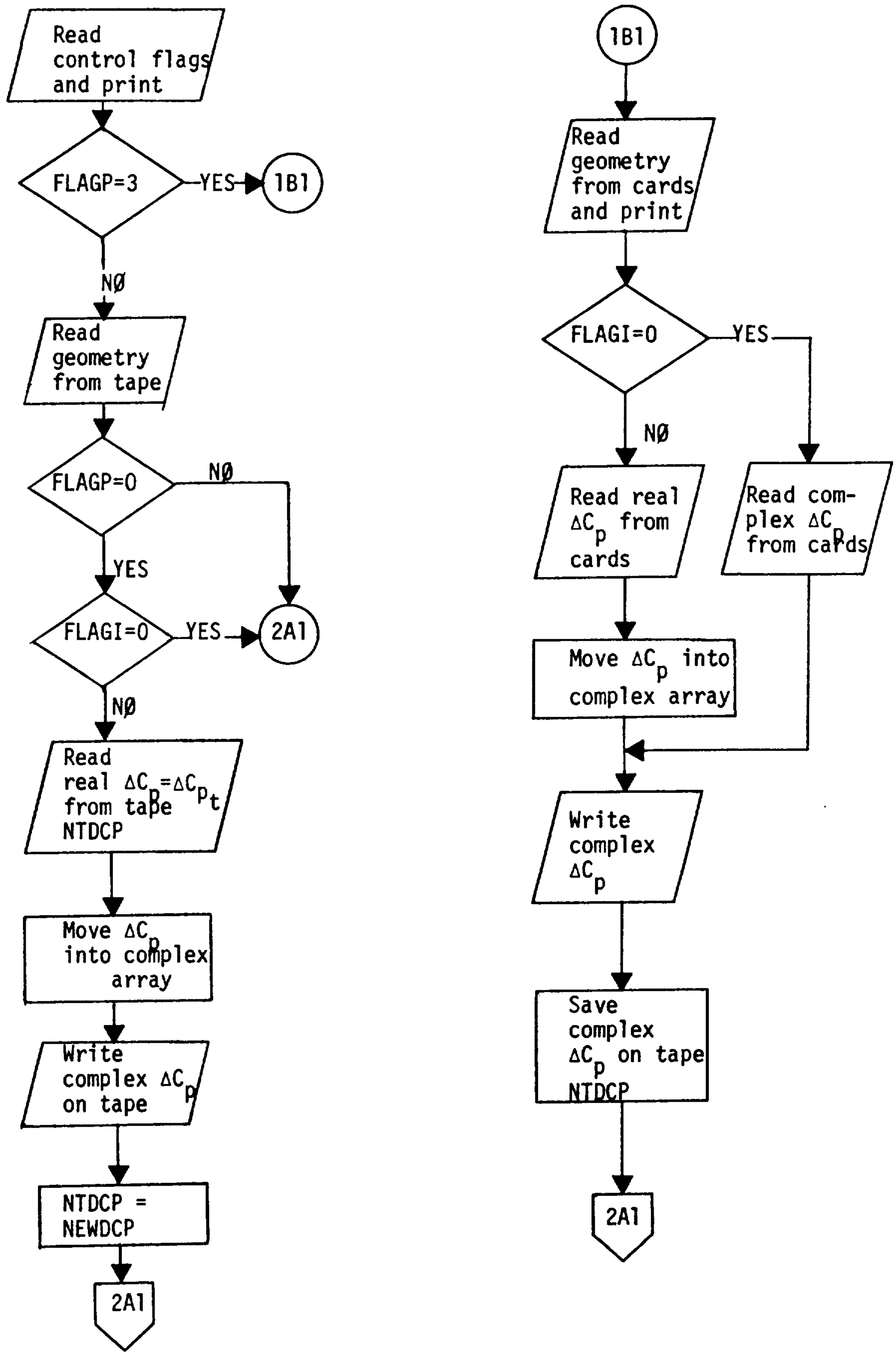
NP	Number of ΔC_p elements
NC	Number of constraints
NEM	Number of correction factor modes
NELIMS	Number of input cards for the EBMIN, EBMAX pairs
NMØN	Number of sets of monitoring data
NAXIS	Number of axes for use in the inegration of the ΔC_{p_t} values
NMIN	= max (1, NELIMS)
NMAX	= max (NC, NMIN, 10)
NS	= max (NP+NC, NEM+NC)
NPIT	Data set number of the system input data set
NPØT	Data set number of the system output data set
W	A column of the normalwash matrix
DI	A row of the inverse-D matrix , (A matrix)
DCP	A column of the theoretical ΔC_{p_t} matrix
EPS	Incremental weight factors array, ε
PHI	A column of the weight factor mode shape matrix, ϕ
SAI	Integration matrix row array , [S]. SAN = [S] for monitoring
DCPTIL	A column of pressures modified by weight matrix, $\Delta\tilde{C}_{p_e}$
DELCPB	A column of the unmodified lifting pressure coefficients, ΔC_{p_t} (either input, ΔC_{p_t}, or computed as $[D]^{-1}\{W\}$)

CØL	Complex array for intermediate use
CIE	Array of the experimental constraints, C_{e_i}
DCMØD	Array of the modified ΔC_e values
EBMIN, EBMAX	Minimum-, maximum values allowed for the ε array to take
EB	$\bar{\varepsilon}$ array (incremental weight factors) $(\varepsilon = \phi\bar{\varepsilon})$
ELIM	Array of the modified ε values
SBB	A row of the $\bar{\bar{S}}$ matrix
SBI	A row of the $[\bar{S}] \lceil \sqrt{T^*} \rfloor^{-1}$ matrix
S	A two-dimensional complex work array of dimension NC by NC
SBMAT	$\bar{\bar{S}}$ matrix of maximum dimension NC by NS
DCI	Array of the ΔC_e values
WØRK	A two-dimensional complex array of dimension NP by NMAX in which the ΔC_{p_t} matrix is stored

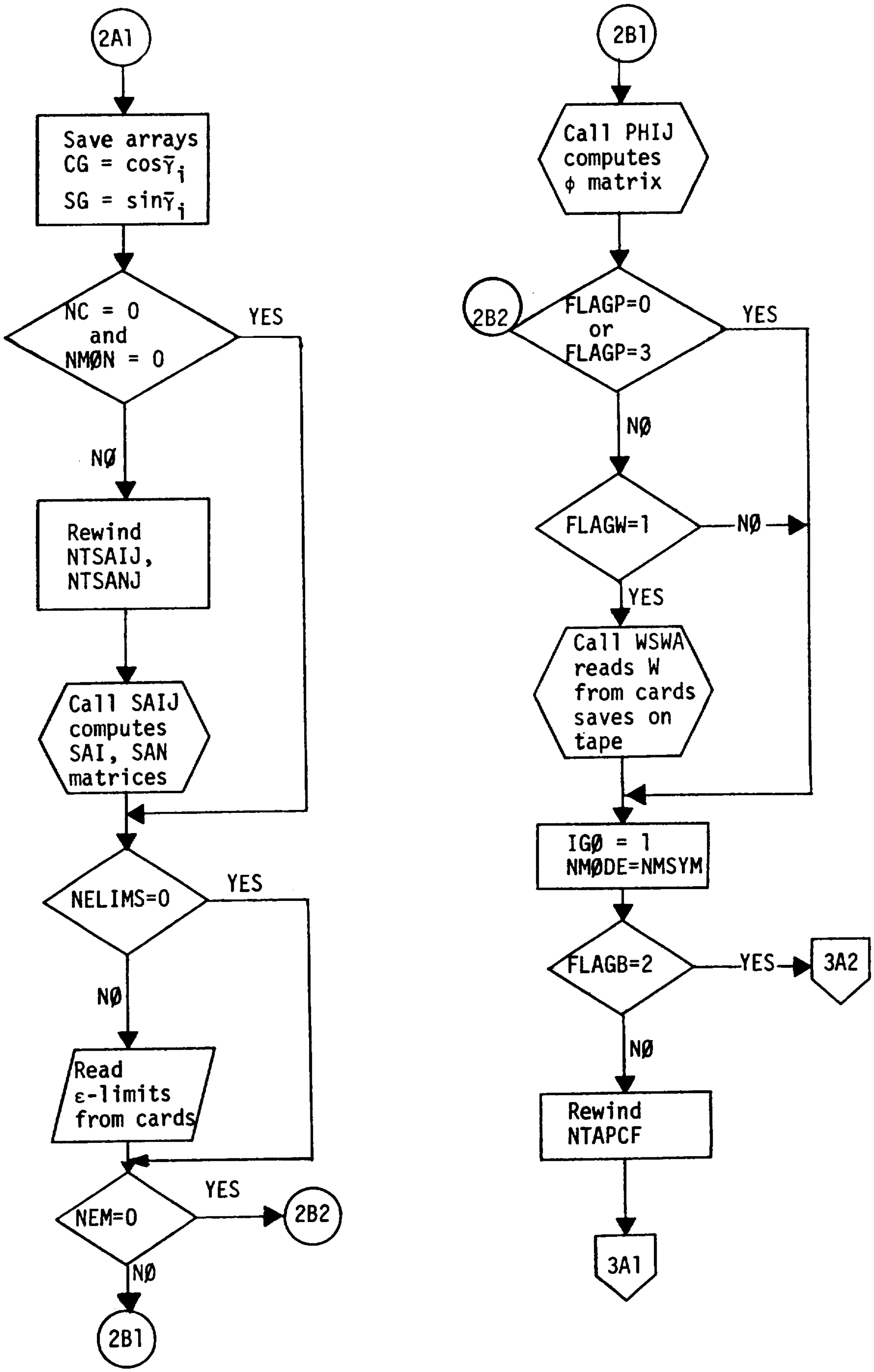
<u>Calling Subroutine</u> MAIN

<u>Called Subroutines</u> CEMN, DCPB, DCPT, DELC, EDBL, EPSJ, GINV, MØDF, PHIJ, RECD, SAIJ, SDBL, WSWA

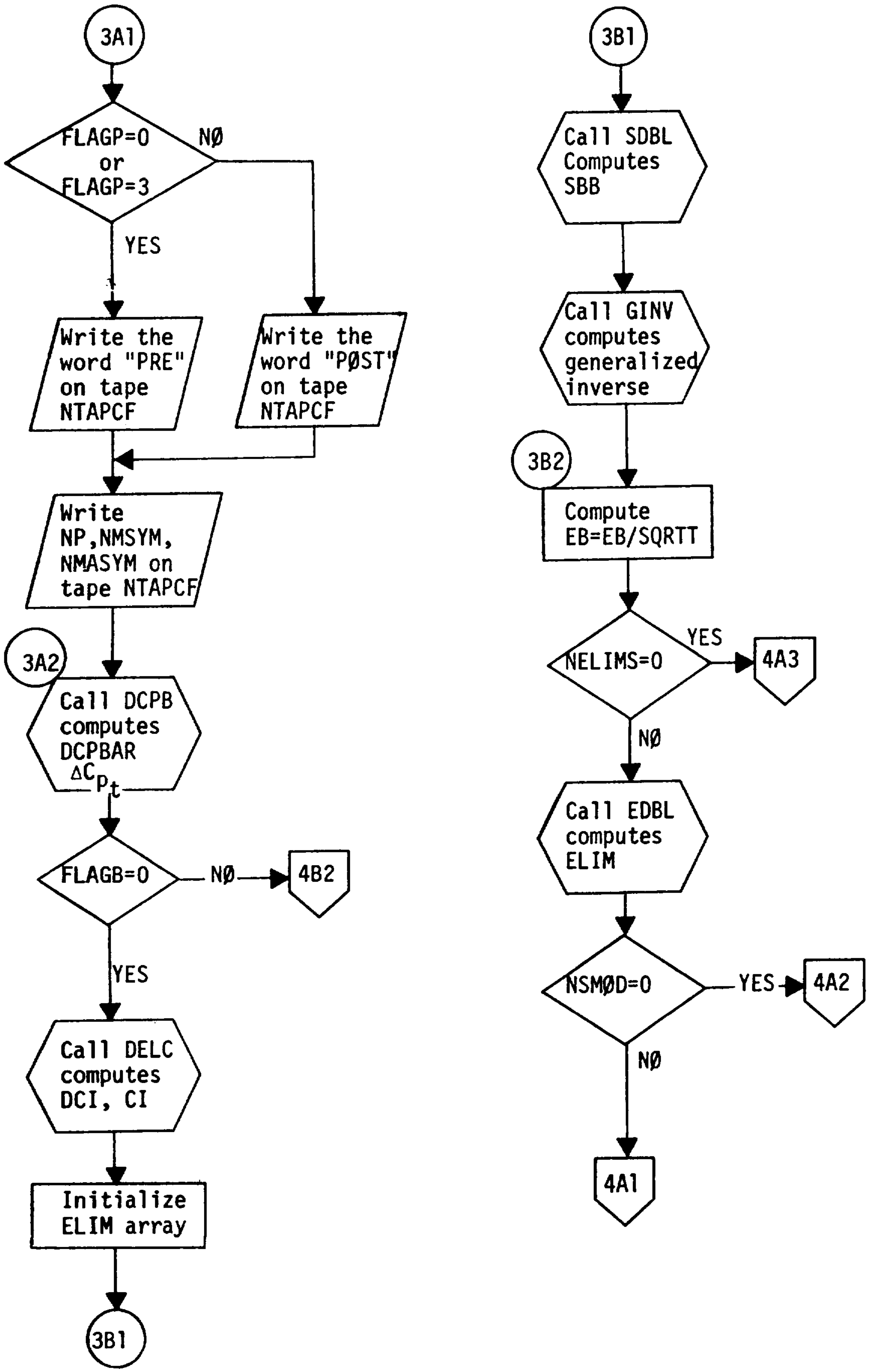
Flow Charts (Read in geometry and pressures)



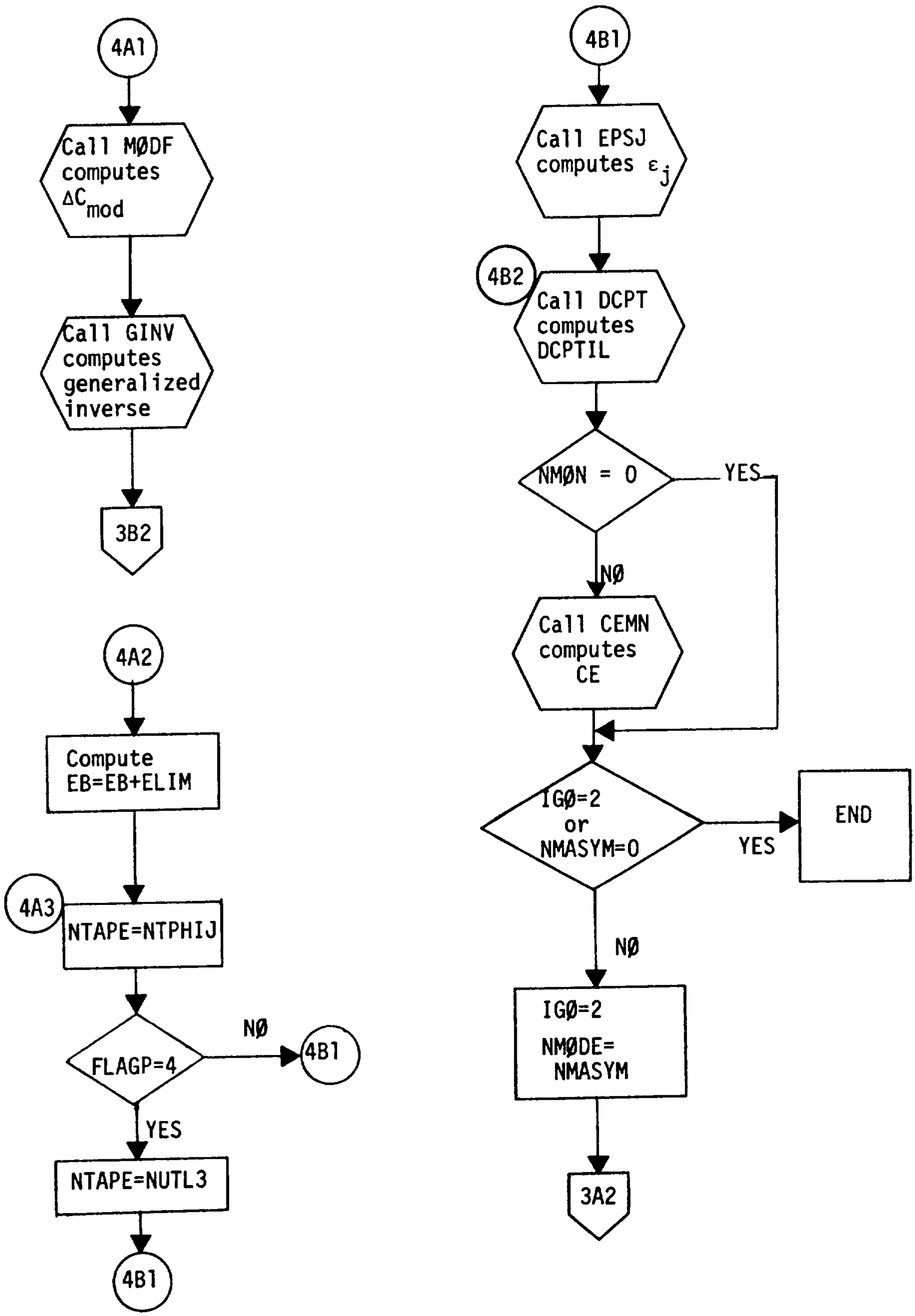
Flow Charts (Calculation of S, ϕ, W)



Flow Charts (Basic method)



Flow Charts (Accounting for limits)



SUBROUTINE WSWA(NPIT, NPØT, NUTL1, NTAPW, KØDE, NP, NCØL,
NMAX, NMSYM, NMASYM, W)

Functional Description

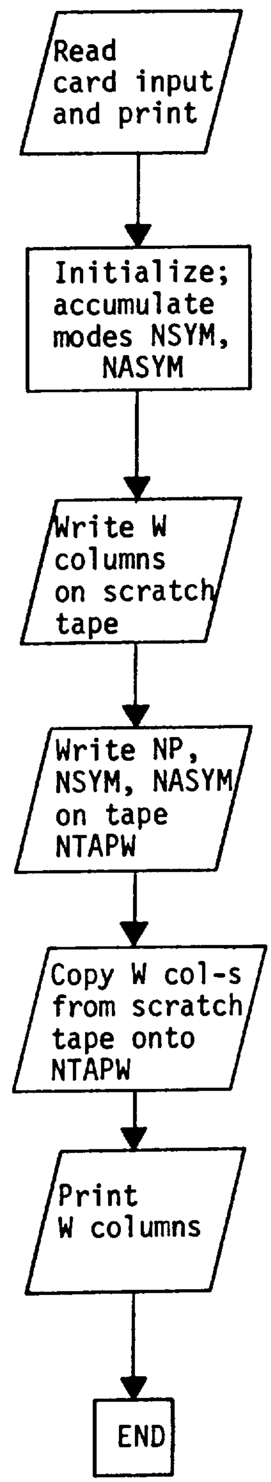
Subroutine WSWA is called from WEYT only if the input flag FLAGW = 1. It reads and prints the mode number and symmetry flag identifying the mode, the range of boxes over which the input W value applies, and the normalwash, W for this range of boxes. This card input is repeated for all ranges as needed, but only the non-zero W values are required as input. The complete W matrix is assembled from the input and is saved on tape NTAPW in column order.

Description of Argument List

NPIT	Data set number of the system input data set
NPØT	Data set number of the system output data set
NUTL1	Utility (scratch) tape number
NTAPW	Tape number containing columns of the W matrix
KØDE	= -1
NP	Number of row elements in the W matrix
NCØL	Number of columns in the W matrix
NMAX	Maximum number of columns in the W matrix
NMSYM	Number of symmetric modes
NMASYM	Number of antisymmetric modes
W	Two-dimensional complex array containing the W matrix

Calling Subroutine WEYT

Flow Chart


Read card input and print
Initialize; accumulate modes NSYM, NASYM
Write W columns on scratch tape
Write NP, NSYM, NASYM on tape NTAPW
Copy W col-s from scratch tape onto NTAPW
Print W columns
END

SUBRØUTINE ZERØUT(WØRK, LENGTH, LØØP, ITAPE)

Functional Description

This subroutine initializes a complex array WØRK of length LENGTH to zeroes. In addition to this, when the argument ITAPE $\neq$ 0, the complex zeroes stored in WØRK are written on tape ITAPE as many times as specified by the argument LØØP.

Description of Argument List

WØRK	Complex array to be initialized to zeroes
LENGTH	Length of the complex array WØRK
LØØP	Number of times the array WØRK is to be written on tape ITAPE (only if ITAPE $\neq$ 0)
ITAPE	Tape number on which the array WØRK is saved LØØP-times (if any)

Calling Subroutines   EPSJ, MAIN, MATM, SDBL

LISTING

CORRECTION FACTORS (E1GC) 02/03/76

```
      SUBROUTINE CEMN(NPOT, IGO, NMODE, NTAPSA,  NP, NMON, LABEL, NUTL, CEMN0040
     1               SAI      , DCPTIL,       CE                      ) CEMN0050
C                                                                       CEMN0060
C          NPOT      DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET      CEMN0070
C          IGO       1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES  CEMN0080
C          NMODE     NUMBER OF MODES                                    CEMN0090
C          NTAPSA    TAPE NUMBER CONTAINING THE INTEGRATION MATRIX  SA  CEMN0100
C          NP        NUMBER OF ROWS IN THE  DELTA-CP  MATRIX            CEMN0110
C          NMON      NUMBER OF INTEGRATION ROWS USED FOR MONITORING     CEMN0120
C          LABEL     LABEL DESCRIBING THE AERODYNAMIC PARAMETERS        CEMN0130
C          NUTL      TAPE NUMBER CONTAINING THE WEIGHTED PRESSURE COLUMNSCEMN0132
C          SAI       INTEGRATION MATRIX ROW                             CEMN0140
C          DCPTIL    ARRAY OF THE WEIGHTED PRESSURES                    CEMN0150
C          CE        AERODYNAMIC PARAMETERS                             CEMN0160
C                                                                       CEMN0170
      DIMENSION  LABEL(10, 35)                                          CEMN0180
      COMPLEX    SAI( NP),  DCPTIL( NP),  CE(  35 )                     CEMN0190
   10 FORMAT   ( 1H1/// 40H        SYMMETRIC    CE-N          MODE, I3 )CEMN0200
   20 FORMAT   ( 1H1/// 40H   ANTISYMMETRIC    CE-N          MODE, I3 )CEMN0210
   30 FORMAT   ( / 40H    N        LABEL              CE(N)             CEMN0220
     1           / 48H                          REAL           IMAG./)  CEMN0230
   40 FORMAT   ( I4, 8X, 10A1, 2E16.6 )                                 CEMN0240
C                                                                       CEMN0250
      REWIND  NUTL                                                      CEMN0252
      DO  120  MODE = 1, NMODE                                          CEMN0254
      READ    (NUTL)  IGON, MD, DCPTIL                                  CEMN0256
      GO TO   (50,60),  IGO                                             CEMN0260
   50 WRITE   (NPOT,10)  MODE                                           CEMN0270
      GO TO   70                                                        CEMN0280
   60 WRITE   (NPOT,20)  MODE                                           CEMN0290
   70 WRITE   (NPOT,30)                                                 CEMN0300
      REWIND  NTAPSA                                                    CEMN0310
      DO  110  N = 1, NMON                                              CEMN0320
      READ    (NTAPSA)  NDN, MN , SAI                                   CEMN0330
      CE(N)     =  (0.0, 0.0)                                           CEMN0340
      DO  100  I = 1, NP                                                CEMN0400
      CE(N)     =  CE(N) + SAI(I) * DCPTIL(I)                           CEMN0410
  100 CONTINUE                                                          CEMN0420
C                                                                       CEMN0430
      WRITE  (NPOT,40)  N  ,  (LABEL(M, N), M = 1, 10), CE(N)           CEMN0440
  110 CONTINUE                                                          CEMN0450
  120 CONTINUE                                                          CEMN0452
C                                                                       CEMN0454
      RETURN                                                            CEMN0460
      END                                                               CEMN0470
      SUBROUTINE DCPB(NTDCP , NTAPW , NTAPDI, IGO,  IFP , IFW , NROW,   DCPB0040
     1               NCOL, NMAX,  DCP   , COL   ,           WORK      ) DCPB0050
C                                                                       DCPB0060
C          NTDCP     TAPE NUMBER CONTAINING THE PRESSURE COEFFICIENTS   DCPB0070
C          NTAPW     TAPE NUMBER CONTAINING THE NORMALWASH MATRIX       DCPB0080
C          NTAPDI    TAPE NUMBER CONTAINING THE INVERSE -D- MATRIX      DCPB0090
C          IGO       1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES  DCPB0100
C          IFP       DELTA-CP-OPTION FLAG  (0, 1, 2, 3 OR 4)            DCPB0110
```

```
                  CORRECTION FACTORS  (E1GC)  02/03/76
                  ------------------------------------
C       IFW       W-OPTION FLAG  (0  OR  1)                           DCPB0120
C       NROW      NUMBER OF ROWS IN THE DELTA-CP MATRIX               DCPB0130
C       NCOL      NUMBER OF COLUMNS IN THE DELTA-CP MATRIX            DCPB0140
C       DCP       ONE COLUMN OF THE DELTA-CP-MATRIX     (COMPLEX)     DCPB0150
C       COL       TEMPORARY WORK ARRAY                  (COMPLEX)     DCPB0160
C       NMAX      MAXIMUM NUMBER OF COLUMNS IN THE DELTA-CP MATRIX    DCPB0170
C       WORK      THE  2-D  DELTA-CP-BAR  MATRIX        (COMPLEX)     DCPB0180
C                                                                     DCPB0190
      COMPLEX    DCP(NROW), COL(NROW), WORK(NROW, NMAX)               DCPB0200
C                                                                     DCPB0210
C                                                                     DCPB0220
      NTAPE     =  NTDCP                                              DCPB0230
      IF  (IFP.EQ.1.OR.IFP.EQ.2.OR. IFP .EQ. 4)  NTAPE = NTAPW        DCPB0240
      CALL         POSN(  NTAPE,  IGO )                               DCPB0250
C                                                                     DCPB0260
      DO  10  J = 1, NCOL                                             DCPB0270
      READ   (NTAPE)  (WORK(I, J),   I = 1, NROW)                     DCPB0280
   10 CONTINUE                                                        DCPB0290
      IF  (IFP.EQ.0.OR.IFP.EQ.3)                     GO TO  30        DCPB0300
   20 CONTINUE                                                        DCPB0310
      CALL         MATM(NTAPDI, IGO, NROW, NCOL, NMAX, DCP , COL , WORK ) DCPB0320
C                                                                     DCPB0330
   30 CONTINUE                                                        DCPB0340
C                                                                     DCPB0350
C         WORK  NOW CONTAINS THE ENTIRE  DELTA-CP-BAR  MATRIX         DCPB0360
C                                                                     DCPB0370
      WRITE  (6,50)  NCOL                                             DCPB0380
      DO  40  J = 1, NCOL                                             DCPB0390
      WRITE  (6  ,60)  J  , (I, WORK(I, J),   I = 1, NROW)            DCPB0400
      IF  (J .NE. NCOL)  WRITE  (6, 70)                               DCPB0402
   40 CONTINUE                                                        DCPB0410
   50 FORMAT  ( 1H1 /// 6H   THE , I4, 55H COLUMNS OF THE   DELTA-CP-BARDCPB0420
     1   MATRIX                             / )                       DCPB0430
   60 FORMAT  ( // 9H   COLUMN , I4 // ( 3 ( I6, 2E14.6 ) ) )          DCPB0440
   70 FORMAT  ( 1H1 // )                                              DCPB0442
      RETURN                                                          DCPB0450
      END                                                             DCPB0460
      SUBROUTINE DCPT(NPOT, LINES, IGO, FLAGB, FLAGP, MODES, NP, NSCRCH,DCPT0040
     1      NUTL, NTAPDI, NTAPW, NTAPCF, X , Y , Z , GMA, DELA,   NMAX,  DCPT0050
     2      NEM , W , DI, EPS,  DCPBAR,   DCPTIL,  WORK    ,  EB      ) DCPT0060
C                                                                     DCPT0070
C       NPOT      DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET       DCPT0080
C       FLAGB     OPTION FLAG FOR DATA MONITORING                     DCPT0090
C       IGO       1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES   DCPT0100
C       MODES     NUMBER OF MODES                                     DCPT0110
C       NP        LENGTH OF THE   DCPBAR   ARRAY                      DCPT0120
C       NSCRCH    TAPE NUMBER CONTAINING THE  PHI-BAR  MATRIX (IF ANY)DCPT0130
C       NUTL      TAPE NUMBER ON WHICH THE  DELTA-CP-TILDA  MATRIX    DCPT0140
C                 COLUMNS ARE SAVED                                   DCPT0150
C       NTAPDI    TAPE NUMBER CONTAINING THE  INVERSE-D  MATRIX ROWS  DCPT0160
C                 (IF NEEDED)                                         DCPT0170
C       NTAPW     TAPE NUMBER CONTAINING THE  W  MATRIX COLUMNS       DCPT0180
C                 (IF NEEDED)                                         DCPT0190
```

```
                CORRECTION FACTORS   (E1GC)   02/03/76
C                -------------------------------------
C       NTAPCF   TAPE NUMBER ON WHICH THE CORRECTION FACTOR MATRIX    DCPT0200
C                IS SAVED                                             DCPT0210
C       X, Y, Z  COORDINATES OF THE  PRESSURE POINTS                  DCPT0220
C       GMA      DIHEDRAL ANGLE ARRAY OF THE BOXES OVER WHICH THE     DCPT0230
C                PRESSURES ACT                                        DCPT0240
C       DELA     ARRAY OF BOX AREAS                                   DCPT0250
C       NMAX     COLUMN DIMENSION OF THE TWO-DIMENSIONAL ARRAY  WORK  DCPT0260
C       NEM      NUMBER OF CORRECTION FACTOR MODES                    DCPT0270
C       W        A COLUMN OF THE  W  MATRIX                           DCPT0280
C       DI       A ROW OF THE  INVERSE-D  MATRIX                      DCPT0290
C       EPS      EPSILON ARRAY                                        DCPT0300
C       DCPBAR   A COLUMN OF THE  DELTA-CP-BAR  MATRIX                DCPT0310
C       DCPTIL   A COLUMN OF THE  DELTA-CP-TILDA  MATRIX              DCPT0320
C       WORK     TWO-DIMENSIONAL COMPLEX ARRAY CONTAINING THE         DCPT0330
C                DELTA-CP-BAR  MATRIX                                 DCPT0340
C       EB       EPSILON-BAR  ARRAY (INCREMENTAL WEIGHT FACTORS)      DCPT0350
C                                                                     DCPT0360
      INTEGER    FLAGB, FLAGP                                         DCPT0370
      DIMENSION  X(NP),   Y(NP),   Z(NP),  GMA(NP),  DELA(NP)         DCPT0380
      COMPLEX    ONE,   EPS(NP),  DCPBAR(NP),   DCPTIL(NP), CF(350),  DCPT0390
     1           W(NP),  DI(NP),  WORK(NP, NMAX )  ,  EB( 1 )         DCPT0400
      COMPLEX    COL(350), PHIBAR(350)                                DCPT0410
      DATA       COL / 350*0.0 /                                      DCPT0420
      DATA       WORD / 4H     /                                      DCPT0430
C                                                                     DCPT0440
   10 FORMAT  ( 1H1/// 38H      SYMMETRIC DELTA-CP-TILDA , MODE, I3 /) DCPT0450
   20 FORMAT  ( 1H1/// 38H  ANTISYMMETRIC DELTA-CP-TILDA , MODE, I3 /) DCPT0460
   30 FORMAT  ( 97H    J   X(J)     Y(J)      Z(J)   GAMMA(J)   DELTA-A(J)DCPT0470
     1          EPS(J)           DCP-TILDA(J)          /    58X,      DCPT0480
     2   42HREAL          IMAG.       REAL          IMAG.        /  )  DCPT0490
   40 FORMAT  ( I4, 4F9.4,  F12.4,  4F12.6 )                          DCPT0500
   50 FORMAT  ( 1H1/// 34H   FLAGB = 1,  MONITOR-STEP  ONLY       /  ) DCPT0510
   60 FORMAT  ( 1H1 /// 50H     CORRECTION FACTORS  **   PREMULTIPLIER DCPT0520
     1CASE  / )                                                       DCPT0530
   70 FORMAT  ( 1H1 /// 50H     CORRECTION FACTORS  **  POSTMULTIPLIER DCPT0540
     1CASE  / )                                                       DCPT0550
   80 FORMAT  ( 1H1 /// 32H   FLAGB = 2   --   MONITOR DATA  //       DCPT0560
     1          43H   CORRECTION FACTOR (TAPE INPUT) CODE IS  , 1A4 / ) DCPT0570
   82 FORMAT  ( 1H1/// 67H   FLAGP = 4   --   TAPE CONTAINS THE  EPSILONDCPT0572
     1-BAR   ( EB )  VALUES    /  )                                   DCPT0574
   90 FORMAT  (                        //  3 ( I6, 2E14.6 ) )         DCPT0580
      IF   (FLAGB .EQ. 1)  WRITE  (NPOT,50)                           DCPT0590
      IF   (FLAGP .EQ. 4)  REWIND  NSCRCH                             DCPT0600
      IFP        =  FLAGP + 1                                         DCPT0610
      ONE        =  (1.0, 0.0)                                        DCPT0620
      REWIND  NUTL                                                    DCPT0630
      IF   (FLAGB .NE. 2)  GO TO  110                                 DCPT0640
      REWIND  NTAPCF                                                  DCPT0650
      READ    (NTAPCF)  WORD                                          DCPT0660
      WRITE   (NPOT,80) WORD                                          DCPT0670
      READ    (NTAPCF)                                                DCPT0680
      READ    (NTAPCF)   (CF(I),  I = 1, NP)                          DCPT0690
      IF  (FLAGP .EQ. 4)  WRITE  (NPOT, 82)                           DCPT0692
```

```
      WRITE  (NPOT, 90)  (I, CF(I),  I = 1, NP)                         DCPT0694
      IF  (FLAGP .EQ. 4)  GO TO  102                                    DCPT0696
      DO  100  I = 1, NP                                                DCPT0700
      EPS(I)   =  CF(I) - ONE                                           DCPT0710
  100 CONTINUE                                                          DCPT0720
      GO TO  110                                                        DCPT0722
  102 CONTINUE                                                          DCPT0724
      DO  104  I = 1, NP                                                DCPT0726
  104 EB(I)    =  CF(I)                                                 DCPT0728
  110 CONTINUE                                                          DCPT0730
      DO  310  MODE = 1, MODES                                          DCPT0740
      IF  (IGO .EQ. 2)  GO TO  120                                      DCPT0750
      WRITE  (NPOT,10)  MODE                                            DCPT0760
      GO TO  130                                                        DCPT0770
  120 WRITE  (NPOT,20)  MODE                                            DCPT0780
  130 CONTINUE                                                          DCPT0790
      WRITE  (NPOT,30)                                                  DCPT0800
C                                                                       DCPT0810
      L        =  1                                                     DCPT0820
      IF  (FLAGB .EQ. 1)  GO TO  280                                    DCPT0830
      GO TO  (180,140,180,180,140),             IFP                     DCPT0840
  140 CONTINUE                                                          DCPT0850
      CALL        POSN( NTAPW , IGO )                                   DCPT0860
      CALL        POSN( NTAPDI, IGO )                                   DCPT0870
      DO  150  M = 1, MODE                                              DCPT0880
  150 READ   (NTAPW )  W                                                DCPT0890
      IF  (FLAGP .EQ. 1 .OR. NEM .EQ. 0)  GO TO  180                    DCPT0900
      IF  (MODE .GT. 1)  GO TO  180                                     DCPT0902
      REWIND  NSCRCH                                                    DCPT0910
      DO  170  K = 1, NEM                                               DCPT0920
      READ   (NSCRCH)  KK,  (PHIBAR(M),  M = 1, NP)                     DCPT0930
      DO  160  M = 1, NP                                                DCPT0940
      COL(M)   =  COL(M) + PHIBAR(M) * EB(K)                            DCPT0950
  160 CONTINUE                                                          DCPT0960
  170 CONTINUE                                                          DCPT0970
C                                                                       DCPT0980
  180 CONTINUE                                                          DCPT0990
      DO  270  I = 1, NP                                                DCPT1000
      DCPTIL(I)=  (0.0, 0.0)                                            DCPT1010
      IF  (FLAGB .NE. 2)  CF(I) = ONE + EPS(I)                          DCPT1020
      GO TO  (250,220,250,250,190),             IFP                     DCPT1030
  190 IF  (NEM .NE. 0)  GO TO  210                                      DCPT1040
      DO  200  N = 1, NP                                                DCPT1050
  200 COL(N)   =  EB(N)                                                 DCPT1060
  210 CONTINUE                                                          DCPT1070
      IF  (FLAGB .NE. 2 .OR. MODE .GT. 1)  GO TO  220                   DCPT1072
      DO  212  M = 1, NP                                                DCPT1074
  212 EPS(M)   =  COL(M)                                                DCPT1076
  220 CONTINUE                                                          DCPT1080
      READ   (NTAPDI)  DI                                               DCPT1090
      DO  240  J = 1, NP                                                DCPT1100
      IF  (FLAGP .EQ. 1)  GO TO  230                                    DCPT1110
C                                                                       DCPT1120
```

```
      DCPTIL(I)=  DCPTIL(I) + DI(J)*(W(J)+COL(J))                      DCPT1130
      GO TO  240                                                       DCPT1140
  230 CONTINUE                                                         DCPT1150
C                                                                      DCPT1160
      DCPTIL(I)=  DCPTIL(I) + DI(J)*(ONE+EPS(J))*W(J)                  DCPT1170
  240 CONTINUE                                                         DCPT1180
      GO TO  260                                                       DCPT1190
  250 CONTINUE                                                         DCPT1200
      DCPTIL(I)=  CF(I) * WORK(I, MODE)                                DCPT1210
  260 CONTINUE                                                         DCPT1220
      WRITE  (NPOT,40)  I, X(I), Y(I), Z(I), GMA(I), DELA(I)           DCPT1230
     1              ,  EPS(I),          DCPTIL(I)                      DCPT1240
      IF  ((I/L) .LT. LINES)  GO TO  270                               DCPT1250
      L           =  L + 1                                             DCPT1260
      IF  (IGO .EQ. 1)  WRITE  (NPOT,10)  MODE                         DCPT1270
      IF  (IGO .EQ. 2)  WRITE  (NPOT,20)  MODE                         DCPT1280
      WRITE  (NPOT,30)                                                 DCPT1290
C                                                                      DCPT1300
  270 CONTINUE                                                         DCPT1310
C                                                                      DCPT1320
      GO TO  300                                                       DCPT1330
  280 CONTINUE                                                         DCPT1340
      DO  290  I = 1, NP                                               DCPT1350
      DCPTIL(I)=  WORK(I, MODE)                                        DCPT1360
      CF(I)     =  ONE + EPS(I)                                        DCPT1370
      WRITE  (NPOT,40)  I, X(I), Y(I), Z(I), GMA(I), DELA(I)           DCPT1380
     1              ,  EPS(I),          DCPTIL(I)                      DCPT1390
  290 CONTINUE                                                         DCPT1400
  300 CONTINUE                                                         DCPT1410
      WRITE  (NUTL )  IGO, MODE, DCPTIL                                DCPT1420
C                                                                      DCPT1430
  310 CONTINUE                                                         DCPT1440
      IF  (FLAGB .EQ. 2 .AND. FLAGP .EQ. 4)  RETURN                    DCPT1442
      GO TO  (330,320,330,330,320),            IFP                     DCPT1450
  320 CONTINUE                                                         DCPT1460
      WRITE  (NPOT,70)                                                 DCPT1470
      GO TO  340                                                       DCPT1480
  330 CONTINUE                                                         DCPT1490
      WRITE  (NPOT,60)                                                 DCPT1500
  340 CONTINUE                                                         DCPT1510
C                                                                      DCPT1520
      WRITE  (NPOT,90)         ( I, CF(I),  I = 1, NP)                 DCPT1530
      IF  (FLAGP .EQ. 4)  GO TO  350                                   DCPT1532
      IF  (FLAGB .EQ. 0)                                               DCPT1540
     1WRITE  (NTAPCF)  (CF(I),  I = 1, NP)                             DCPT1550
      REWIND  NUTL                                                     DCPT1560
C                                                                      DCPT1570
      RETURN                                                           DCPT1580
  350 WRITE  (NTAPCF)  (EB(I),  I = 1, NP)                             DCPT1582
C                                                                      DCPT1584
      WRITE  (NPOT, 82)                                                DCPT1586
      WRITE  (NPOT, 90)  (I, EB(I),  I = 1, NP)                        DCPT1588
      REWIND  NUTL                                                     DCPT1590
```

CORRECTION FACTORS (E1GC) 02/03/76

```
      RETURN                                                            DCPT1600
      END                                                               DCPT1610
      SUBROUTINE DELC(NTAPE, NPOT,  NC, NP, NMODE, NMAX ,               DELC0040
     1          CIE, DCI, SAI, WORK                                   ) DELC0050
C                                                                       DELC0060
C        NTAPE      TAPE NUMBER CONTAINING THE INTEGRATION MATRIX  SA   DELC0070
C        NPOT       DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET       DELC0080
C        NC         NUMBER OF CONSTRAINTS                               DELC0090
C        NP         LENGTH OF THE  DELTA-CP  COLUMNS                    DELC0100
C        NMODE      NUMBER OF MODES                                     DELC0110
C        NMAX       MAXIMUM NUMBER OF COLUMNS IN THE  DELTA-CP  MATRIX  DELC0120
C        CIE        ARRAY OF THE INPUT VALUES  C-I(E)                   DELC0130
C        DCI        THE  DELTA-C  ARRAY                                 DELC0132
C        SAI        A ROW OF THE INTEGRATION MATRIX  SA                 DELC0134
C        WORK       THE  NP-BY-NMAX  COMPLEX ARRAY CONTAINING THE       DELC0136
C                   DELTA-CP-BAR  MATRIX                                DELC0140
C                                                                       DELC0150
      COMPLEX     CIE(NC), DCI(NC      ), WORK(NP, NMAX), SAI(NP), CI(35)DELC0160
      WRITE  (NPOT,40)                                                  DELC0170
      REWIND  NTAPE                                                     DELC0180
      DO  20  I = 1, NC                                                 DELC0190
      DCI(I    )=  (0.0, 0.0)                                           DELC0200
      READ   (NTAPE)  NDI, MI, SAI                                      DELC0210
      DO  10  J = 1, NP                                                 DELC0220
      DCI(I    )=  DCI(I    )+SAI(J)* WORK(J, MI)                       DELC0230
   10 CONTINUE                                                          DELC0240
      CI(I)     =  DCI(I    )                                           DELC0250
      DCI(I    )=  CIE(I) - DCI(I    )                                  DELC0260
   20 CONTINUE                                                          DELC0270
C                                                                       DELC0280
      WRITE  (NPOT,50)  (DCI(I    ),  I = 1, NC)                        DELC0290
      WRITE  (NPOT,60)                                                  DELC0300
      WRITE  (NPOT,50)  (CI(I),  I = 1, NC)                             DELC0310
   30 CONTINUE                                                          DELC0320
C                                                                       DELC0330
   40 FORMAT  ( 1H1 /// 12H   DELTA-C       /  )                        DELC0340
   50 FORMAT  ( 8F13.6 )                                                DELC0350
   60 FORMAT  ( /// 25H   THEORETICAL  C-VALUES   / )                   DELC0360
C                                                                       DELC0370
      RETURN                                                            DELC0380
      END                                                               DELC0390
      SUBROUTINE EDBL(NPOT, NELIMS,  NP ,  NS , LIMK , JARR , NSMOD   , EDBL0040
     1           EBMIN   , EBMAX ,  EB , ELIM                         ) EDBL0050
C                                                                       EDBL0060
C        NPOT       DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET       EDBL0070
C        NELIMS     NUMBER OF INPUT CARDS FOR THE  EBMIN, EBMAX  PAIRS  EDBL0080
C        NP         LENGTH OF THE  DELTA-CP  COLUMNS                    EDBL0090
C        NS         =  MAX (NP+NC, NEM+NC)                              EDBL0100
C        LIMK       FIRST AND LAST BOX NUMBERS FOR THE  EBMIN, EBMAX    EDBL0110
C        JARR       ARRAY OF BOX NUMBERS FOR WHICH THE EPSILON VALUES   EDBL0120
C                   WERE MODIFIED                                       EDBL0122
C        NSMOD      NUMBER OF MODIFIED EPSILON VALUES                   EDBL0124
C        EBMIN      MINIMUM VALUE ALLOWED FOR EPSILON                   EDBL0130
```

```
                CORRECTION FACTORS  (E1GC)  02/03/76
                ------------------------------------
C         EBMAX     MAXIMUM VALUE ALLOWED FOR EPSILON                    EDBL0140
C         EB        ARRAY OF THE CALCULATED EPSILON-BAR VALUES           EDBL0150
C         ELIM      ARRAY OF THE MODIFIED EPSILON VALUES  (LIMITS)       EDBL0160
C                                                                        EDBL0170
      DIMENSION  LIMK(2,  100 ),  JARR(  350 )                           EDBL0180
      COMPLEX    EBMIN( NELIMS), EBMAX(NELIMS),    EB(NS)   ,            EDBL0190
     1           ELIM (NS)   ,  ETEMP                                    EDBL0200
   10 FORMAT  ( 1H1 /// 8H        J , 7X, 11HEPSILON-BAR , 17X,          EDBL0210
     1          14HEPSILON-LIMIT ,  14X, 12HEPSILON-LAST        /  )     EDBL0220
   20 FORMAT  ( I8, 6F14.6 )                                             EDBL0230
      JCUM       =  0                                                    EDBL0240
      NSMOD      =  0                                                    EDBL0250
      J          =  1                                                    EDBL0260
   30 K          =  1                                                    EDBL0270
C                                                                        EDBL0280
   40 CONTINUE                                                           EDBL0290
      LIM1       =  LIMK(1, K)                                           EDBL0300
      LIM2       =  LIMK(2, K)                                           EDBL0310
C                                                                        EDBL0320
      IF   (J .GT. NP)  GO TO  80                                        EDBL0330
      IF   (J .LT. LIM1 .OR. J .GT. LIM2)  GO TO  70                     EDBL0340
C                                                                        EDBL0350
C          THE INDEX  J  IS BETWEEN THE TWO PRESCRIBED LIMITS            EDBL0360
C                                                                        EDBL0370
      EB1        =  REAL(EBMIN(K))                                       EDBL0380
      EB2        =  REAL(EBMAX(K))                                       EDBL0390
      ABSEB      =  REAL(EB(J))                                          EDBL0400
C                                                                        EDBL0410
      IF  (ABSEB .GE. EB1 .AND. ABSEB .LE. EB2)  GO TO  80               EDBL0420
      JCUM       =  JCUM + 1                                             EDBL0430
      IF  (ABSEB .GT. EB2)  GO TO  50                                    EDBL0440
      ETEMP      =  EBMIN(K)                                             EDBL0450
      GO TO  60                                                          EDBL0460
C                                                                        EDBL0470
   50 CONTINUE                                                           EDBL0480
      ETEMP      =  EBMAX(K)                                             EDBL0490
   60 CONTINUE                                                           EDBL0500
      JARR(J)    =  J                                                    EDBL0510
      ELIM(J)    =  ETEMP                                                EDBL0520
      GO TO  90                                                          EDBL0530
C                                                                        EDBL0540
   70 CONTINUE                                                           EDBL0550
      IF   (K .EQ. NELIMS)  GO TO  80                                    EDBL0560
      K          =  K + 1                                                EDBL0570
      GO TO  40                                                          EDBL0580
C                                                                        EDBL0590
   80 CONTINUE                                                           EDBL0600
C                                                                        EDBL0610
   90 CONTINUE                                                           EDBL0620
      IF   (J .EQ. NS)  GO TO  100                                       EDBL0630
      J          =  J + 1                                                EDBL0640
      GO TO  30                                                          EDBL0650
  100 CONTINUE                                                           EDBL0660
```

```
      NSMOD     =  JCUM                                                 EDBL0670
      IF  (NSMOD .EQ. 0)  RETURN                                        EDBL0680
      WRITE  (NPOT,10)                                                  EDBL0690
C                                                                       EDBL0700
      DO  110  J = 1, NS                                                EDBL0710
      ETEMP     =  EB(J)                                                EDBL0720
      IF  (J .EQ. JARR(J))  ETEMP = ELIM(J)                             EDBL0730
      WRITE  (NPOT,20)  J, EB(J), ELIM(J), ETEMP                        EDBL0740
  110 CONTINUE                                                          EDBL0750
C                                                                       EDBL0760
      RETURN                                                            EDBL0770
      END                                                               EDBL0780
      SUBROUTINE EPSJ(NTPHIJ, NP , NEM , NMIN, EB   ,  EPS   ,  PHI    ) EPSJ0040
C                                                                       EPSJ0050
C         NTPHIJ    TAPE NUMBER CONTAINING THE  PHI  MATRIX COLUMNS     EPSJ0060
C         NP        NUMBER OF ROW ELEMENTS IN THE  PHI  MATRIX          EPSJ0070
C         NEM       NUMBER OF COLUMNS      IN THE  PHI  MATRIX          EPSJ0080
C         NS        =  MAX (NP+NC, NEM+NC)                              EPSJ0090
C         EB        ARRAY OF THE  EPSILON-BAR  VALUES                   EPSJ0100
C         EPS       THE FINAL  EPSILON  ARRAY                           EPSJ0110
C         PHI       A COLUMN OF THE  PHI  MATRIX  (WEIGHT FACTOR MODE   EPSJ0120
C                   SHAPES)                                             EPSJ0122
C                                                                       EPSJ0130
      COMPLEX     EPS(NP)  ,   EB( 1)  ,  PHI(NP)                       EPSJ0140
      CALL        ZEROUT( EPS  ,  NP   ,  0  ,  0  )                    EPSJ0150
      IF  (NEM .EQ. 0)  GO TO  30                                       EPSJ0160
      REWIND  NTPHIJ                                                    EPSJ0170
      DO  20  J = 1, NEM                                                EPSJ0190
      READ   (NTPHIJ)  MODENO,  PHI                                     EPSJ0210
C                                                                       EPSJ0230
      DO  10  I = 1, NP                                                 EPSJ0240
      EPS(I)    =  EPS(I) + PHI(I) * EB(J)                              EPSJ0250
   10 CONTINUE                                                          EPSJ0260
C                                                                       EPSJ0270
   20 CONTINUE                                                          EPSJ0280
      RETURN                                                            EPSJ0290
C                                                                       EPSJ0300
   30 CONTINUE                                                          EPSJ0310
      DO  40  I = 1, NP                                                 EPSJ0320
      EPS(I)    =  EB(I)                                                EPSJ0330
   40 CONTINUE                                                          EPSJ0340
C                                                                       EPSJ0350
      RETURN                                                            EPSJ0360
      END                                                               EPSJ0370
      SUBROUTINE GINV(NPOT, NTAPSB, NC, NS, NX  , DC, EB,               GINV0040
     1           B , S , SBB                                          ) GINV0050
C                                                                       GINV0060
C         NPOT      DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET       GINV0070
C         NTAPSB    TAPE NUMBER CONTAINING THE  S-DOUBLE-BAR MATRIX ROWSGINV0080
C         NC        NUMBER OF CONSTRAINTS                               GINV0090
C         NS        =  MAX (NP+NC, NEM+NC)                              GINV0100
C         NX        NX=NS  IF NEM=0,  NX=NEM+NC  OTHERWISE              GINV0110
C         DC        THE COMPLEX  DELTA-C  ARRAY                         GINV0120
```

```
                     CORRECTION FACTORS  (E1GC)  02/03/76
                     ----------------------------------------
C         EB         THE COMPLEX  EPSILON-BAR  ARRAY, OUTPUT OF  GINV     GINV0130
C         B          ARRAY OF INTERMEDIATE SOLUTIONS                      GINV0140
C         S          2-D COMPLEX WORK ARRAY,   NC-BY-NC                   GINV0150
C         SBB        THE COMPLEX  NC-BY-NS   S-DOUBLE-BAR  MATRIX         GINV0160
C                                                                         GINV0170
C                                                                         GINV0180
      COMPLEX    DC(NC), EB(NS), B(NC), S(NC, NC), SBB(NC, NS), SCALER   GINV0190
      COMPLEX    SBBCT                                                    GINV0192
      DATA       NERR, M, SCALER / 0, 1, (1.0,0.0) /                      GINV0200
C                                                                         GINV0210
C         NOTE THAT  -EB-  STANDS FOR  EPSILON-TILDA                      GINV0220
C                                                                         GINV0230
      NSMNC     =  NX - NC                                                GINV0240
      DO  10  I = 1, NC                                                   GINV0250
      READ   (NTAPSB)   (SBB(I, J), J = 1, NS)                            GINV0260
      IF   (NSMNC .LT. 0)   GO TO  10                                     GINV0270
      EB(I)     =   DC(I)                                                 GINV0280
   10 CONTINUE                                                            GINV0290
C                                                                         GINV0300
      IF   (NSMNC)90,20,30                                                GINV0310
   20 CONTINUE                                                            GINV0320
C                                                                         GINV0330
C         NS = NC   BRANCH  ---  SOLVE THE EQUATION   DC = SBB * EB       GINV0340
C         FOR   EB   USING   MIS2                                         GINV0350
C                                                                         GINV0360
      CALL        MIS2( SBB , NS , NC , EB , M , NERR, SCALER )           GINV0370
C                                                                         GINV0380
      GO TO  140                                                          GINV0390
C                                                                         GINV0400
   30 CONTINUE                                                            GINV0410
C                                                                         GINV0420
C         NS-GREATER-THAN-NC   BRANCH  ---                                GINV0430
C         COMPUTE    S = SBB * (SBB-CONJUGATE-TRANSPOSE)  AND             GINV0440
C         SOLVE THE EQUATION   DC = S * B   FOR   B   USING   MIS2        GINV0450
C                                                                         GINV0460
      DO  60  I = 1, NC                                                   GINV0470
      B(I)       =   DC(I)                                                GINV0480
      DO  50  K = 1, NC                                                   GINV0490
      S(I, K)    =   (0.0, 0.0 )                                          GINV0500
      DO  40  J = 1, NX                                                   GINV0510
      SBBCT      =   CONJG( SBB(K, J) )                                   GINV0512
      S(I, K)    =   S(I, K) + SBB(I, J) *  SBBCT                         GINV0520
   40 CONTINUE                                                            GINV0530
   50 CONTINUE                                                            GINV0540
   60 CONTINUE                                                            GINV0550
      WRITE  (NPOT,180)                                                   GINV0560
      WRITE  (NPOT,200)   S                                               GINV0570
      WRITE  (NPOT,170)                                                   GINV0580
      WRITE  (NPOT,200)   DC                                              GINV0590
C                                                                         GINV0600
      CALL        MIS2( S , NC , NC , B , M , NERR, SCALER )              GINV0610
C                                                                         GINV0620
      WRITE  (6,190)                                                      GINV0630
```

CORRECTION FACTORS   (E1GC)   02/03/76

```
      WRITE  (6,200)  B                                                 GINV0640
C         COMPUTE    EB = (SBB-CONJUGATE-TRANSPOSE) * B                 GINV0650
C                                                                       GINV0660
      DO   80   J = 1, NX                                               GINV0670
      EB(J)      =  (0.0, 0.0)                                          GINV0680
      DO   70   I = 1, NC                                               GINV0690
      SBBCT      =  CONJG( SBB(I, J) )                                  GINV0692
      EB(J)      =  EB(J) +  SBBCT     * B(I)                           GINV0700
   70 CONTINUE                                                          GINV0710
   80 CONTINUE                                                          GINV0720
C                                                                       GINV0730
      GO TO  140                                                        GINV0740
C                                                                       GINV0750
   90 CONTINUE                                                          GINV0760
C                                                                       GINV0770
C         NS-LESS-THAN-NC    BRANCH  ---   THE LEAST SQUARES CASE  ---  GINV0780
C         COMPUTE    B = (SBB-CONJUGATE-TRANSPOSE) * (DELTA-C)          GINV0790
C         COMPUTE    S = (SBB-CONJUGATE-TRANSPOSE) * SBB   ,  AND       GINV0800
C         SOLVE THE EQUATION    B = S * EB    FOR   EB   USING   MIS2   GINV0810
C                                                                       GINV0820
      DO   130   J = 1, NX                                              GINV0830
      EB(J)      =  (0.0, 0.0)                                          GINV0840
      DO   100   L = 1, NC                                              GINV0850
      SBBCT      =  CONJG( SBB(L, J) )                                  GINV0852
      EB(J)      =  EB(J) +  SBBCT     * DC(L)                          GINV0860
  100 CONTINUE                                                          GINV0870
C                                                                       GINV0880
      DO   120   K = 1, NX                                              GINV0890
      DO   110   I = 1, NC                                              GINV0892
      SBBCT      =  CONJG( SBB(I, J) )                                  GINV0894
      S(J, K)    =  S(J, K) +  SBBCT     * SBB(I, K)                    GINV0900
  110 CONTINUE                                                          GINV0910
  120 CONTINUE                                                          GINV0920
  130 CONTINUE                                                          GINV0930
C                                                                       GINV0940
      CALL        MIS2( S , NC , NS , EB , M , NERR, SCALER )           GINV0950
C                                                                       GINV0960
C                                                                       GINV0970
  140 CONTINUE                                                          GINV0980
C                                                                       GINV0990
C                                                                       GINV1000
      WRITE  (NPOT,150)                                                 GINV1010
      WRITE  (NPOT,160)   (EB(I),  I = 1, NX)                           GINV1020
  150 FORMAT  ( /// 40H   OUTPUT OF  GEN. INVERSE  (EPS-TILDA)     /  ) GINV1030
  160 FORMAT  ( 8F13.6 )                                                GINV1040
  170 FORMAT  (       /// 16H   -DC-   COLUMN      / )                  GINV1050
  180 FORMAT  ( 1H1 /// 16H   -S-    MATRIX      / )                    GINV1060
  190 FORMAT  (       /// 28H   SOLUTION OF MATRIX EQ.     /  )         GINV1070
  200 FORMAT  ( 6E16.6 )                                                GINV1080
C                                                                       GINV1100
      RETURN                                                            GINV1110
      END                                                               GINV1120
      PROGRAM E1GC(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE1=512, MAIN0002
```

```
                  CORRECTION FACTORS  (E1GC)  02/03/76
                  ------------------------------------
      1  TAPE2=512 , TAPE3=512 , TAPE4=512 , TAPE8=512 , TAPE9=512 ,     MAIN0004
      2  TAPE10=512, TAPE11=512, TAPE12=512, TAPE13=512, TAPE14=512,     MAIN0006
      3  TAPE15=512, TAPE16=512                                       )  MAIN0008
C                                                                        MAIN0010
C   ***  WEIGHT FACTORS FOR FINITE ELEMENT LIFTING SURFACE THEORY        MAIN0020
C                                                                        MAIN0030
C        PROGRAM  E1GC  CREATES PRE- OR POST-MULTIPLYING MATRICES THAT   MAIN0040
C        MODIFY THEORETICAL DATA TO REFLECT EXPERIMENTALLY KNOWN DATA.   MAIN0050
C        THIS PROGRAM CAN ALSO BE USED TO MONITOR THEORETICAL PRESSURE   MAIN0060
C        DATA.   INPUT TO PROGRAM IS  THEORETICAL PRESSURES,  DATA       MAIN0070
C        NEEDED FOR THE INTEGRATION MATRICES  AND  EXPERIMENTAL          MAIN0080
C        CONSTRAINTS.   OUTPUT TO PROGRAM IS  WEIGHT MATRICES,           MAIN0090
C        MODIFIED PRESSURES  AND  MONITORED DATA.                        MAIN0100
C                                                                        MAIN0110
C                                                                        MAIN0120
      COMPLEX     WORK(10000)                                            MAIN0130
      DIMENSION   HEADER( 20)                                            MAIN0140
      DATA        END / 4HEND  /, LOC, LENGTH             / 1,   10000 / MAIN0150
      DATA        NPIT, NPOT / 5, 6 /,  NMIN, NMAX, NRHS / 1, 2*10 /     MAIN0160
C                                                                        MAIN0170
   10 FORMAT   ( 15A4 )                                                  MAIN0180
   20 FORMAT   ( 1H1 //// 10X, 15A4 / )                                  MAIN0190
   30 FORMAT   ( 8I10 )                                                  MAIN0200
   40 FORMAT   ( /// 46H   --WORK ARRAY DOES NOT FIT INTO CORE-- STOP  ) MAIN0210
   42 FORMAT   ( /// 33H   ***   CORE REQUIRED (NWORK) = , I6            MAIN0212
     1           //  33H   ***   CORE AVAILABLE        = , I6        //) MAIN0220
C                                                                        MAIN0230
   50 CONTINUE                                                           MAIN0240
C                                                                        MAIN0250
C          READ  HEADER CARD                                             MAIN0260
      READ   (NPIT,10)  (HEADER(I),  I = 1, 15)                          MAIN0270
      IF  (HEADER(1) .EQ. END)  STOP                                     MAIN0280
      WRITE  (NPOT,20)  (HEADER(I),  I = 1, 15)                          MAIN0290
C                                                                        MAIN0300
C          READ CONTROL DIMENSIONS                                       MAIN0310
      READ   (NPIT,30)  NP, NC, NEM, NELIMS, NMON, NAXIS                 MAIN0320
      NS        =  NP + NC                                               MAIN0330
      IF  (NEM .GT. NP)  NS = NEM + NC                                   MAIN0340
      NMATR1    =  NC * NC                                               MAIN0350
      NMATR2    =  NC * NS                                               MAIN0360
      IF  (NELIMS .NE. 0)  NMIN = NELIMS                                 MAIN0370
C                                                                        MAIN0380
C          CALCULATE SIZE OF WORK ARRAY                                  MAIN0390
C                                                                        MAIN0400
      NMAX      =  MAX0(NC, NMIN, NRHS )                                 MAIN0410
      NMATR3    =  NC                                                    MAIN0420
      NMATR4    =  NP * NMAX                                             MAIN0430
      NMATRX    =  NMATR1 + NMATR2 + NMATR3 + NMATR4                     MAIN0440
      NWORK     =  9*NP + 2*(NC+NMIN) + 4*NS + NMATRX                    MAIN0450
C                                                                        MAIN0460
      WRITE  (NPOT,42)  NWORK, LENGTH                                    MAIN0510
      IF  (LENGTH .GE.  NWORK)  GO TO  60                                MAIN0520
      WRITE  (NPOT,40)                                                   MAIN0530
      STOP     16
```

```
                                                                       MAIN0540
C                                                                      MAIN0550
   60 CONTINUE                                                         MAIN0560
C                                                                      MAIN0570
      CALL       ZEROUT( WORK(LOC),  NWORK,  1 ,  0  )                 MAIN0580
C                                                                      MAIN0590
      L1         =  LOC                                                MAIN0600
      L2         =  L1  + NP                                           MAIN0610
      L3         =  L2  + NP                                           MAIN0620
      L4         =  L3  + NP                                           MAIN0630
      L5         =  L4  + NP                                           MAIN0640
      L6         =  L5  + NP                                           MAIN0650
      L7         =  L6  + NP                                           MAIN0660
      L8         =  L7  + NP                                           MAIN0670
      L9         =  L8  + NP                                           MAIN0680
      L10        =  L9  + NP                                           MAIN0690
      L11        =  L10 + NC                                           MAIN0700
      L12        =  L11 + NC                                           MAIN0710
      L13        =  L12 + NMIN                                         MAIN0720
      L14        =  L13 + NMIN                                         MAIN0730
      L15        =  L14 + NS                                           MAIN0740
      L16        =  L15 + NS                                           MAIN0750
      L17        =  L16 + NS                                           MAIN0760
      L18        =  L17 + NS                                           MAIN0770
      L19        =  L18 + NMATR1                                       MAIN0780
      L20        =  L19 + NMATR2                                       MAIN0790
      L21        =  L20 + NMATR3                                       MAIN0800
C                                                                      MAIN0810
      CALL       WEYT(NP , NC , NEM , NELIMS, NMON, NAXIS, NMIN,       MAIN0820
     1           NMAX,   NS   , NPIT, NPOT   ,          WORK( L1),     MAIN0830
     2           WORK( L2), WORK( L3), WORK( L4), WORK( L5), WORK( L6),MAIN0840
     3           WORK( L7), WORK( L8), WORK( L9), WORK(L10), WORK(L11),MAIN0850
     4           WORK(L12), WORK(L13), WORK(L14), WORK(L15), WORK(L16),MAIN0860
     5           WORK(L17), WORK(L18), WORK(L19), WORK(L20), WORK(L21))MAIN0870
C                                                                      MAIN0880
      GO TO  50                                                        MAIN0890
      END                                                              MATM0040
      SUBROUTINE MATM( NT , IGO, NR , NC , NMAX,  A,  C,  B          ) MATM0050
C                                                                      MATM0060
C         NT         TAPE NUMBER CONTAINING THE  INVERSE-D  MATRIX ROWS MATM0070
C         IGO        1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES  MATM0080
C         NR         NUMBER OF ROW ELEMENTS IN THE  DELTA-CP-BAR  MATRIX MATM0090
C         NC         NUMBER OF COLUMNS      IN THE  DELTA-CP-BAR  MATRIX MATM0100
C         NMAX       MAXIMUM NUMBER OF COLUMNS IN ARRAY  B             MATM0110
C         A          A ROW OF THE  INVERSE-D  MATRIX                   MATM0120
C         C          COMPLEX WORK ARRAY                                MATM0130
C         B          2-D COMPLEX ARRAY IN WHICH THE  DELTA-CP-BAR  MATRIXMATM0132
C                    IS STORED                                         MATM0140
C                                                                      MATM0150
      COMPLEX    A(NR),  C(NR),  B(NR, NMAX)                           MATM0160
      DO  40  K = 1, NC                                                MATM0170
      CALL       POSN(  NT  , IGO )                                    MATM0180
      CALL       ZEROUT( C  ,  NR,  1 ,  0  )                          MATM0190
      DO  20  I = 1, NR
```

CORRECTION FACTORS (E1GC) 02/03/76

```
      READ   (NT )  A                                                    MATM0200
      DO   10   J =  1, NR                                               MATM0210
      C(I)         =   C(I) + A(J) * B(J, K)                             MATM0220
   10 CONTINUE                                                           MATM0230
   20 CONTINUE                                                           MATM0240
C                                                                        MATM0250
      DO  30   I =  1, NR                                                MATM0260
      B(I, K)   =   C(I)                                                 MATM0270
C                                                                        MATM0280
   30 CONTINUE                                                           MATM0290
   40 CONTINUE                                                           MATM0300
      RETURN                                                             MATM0310
      END                                                                MATM0320
      SUBROUTINE   MODF( NC , NS , MASTSB, NEWTSB, JARR,   SQRTT,        )MODF0040
     1             ELIM, SBB, DCI, DCMOD                                 MODF0050
C                                                                        MODF0060
C         NC          NUMBER OF CONSTRAINTS, LENGTH OF ARRAYS  DCI, DCMOD MODF0070
C         NS          LENGTH OF THE  SBB  ARRAY                          MODF0080
C         MASTSB      TAPE NUMBER CONTAINING ROWS OF THE  S-DOUBLE-BAR   MODF0090
C                     MATRIX  (SBB)                                      MODF0100
C         NEWTSB      TAPE NUMBER CONTAINING ROWS OF THE MODIFIED        MODF0110
C                     S-DOUBLE-BAR  MATRIX                               MODF0112
C         JARR        ARRAY OF BOX NUMBERS (ELEMENTS) FOR WHICH THE  SBB MODF0120
C                     VALUES ARE REPLACED BY ZEROES                      MODF0122
C         SQRTT       SQRT(T)     --    SEE EQUATIONS                    MODF0130
C         ELIM        ARRAY OF THE MODIFIED EPSILON VALUES               MODF0140
C         SBB         A ROW OF THE  S-DOUBLE-BAR  MATRIX  (INTEGRATION   MODF0150
C                     MATRIX WITH WEIGHT FACTOR MODES)                   MODF0152
C         DCI         ARRAY OF  DELTA-C  VALUES                          MODF0160
C         DCMOD       ARRAY OF THE MODIFIED  DELTA-C  VALUES             MODF0170
C                                                                        MODF0180
      DIMENSION  JARR(  350 ), SQRTT(350)                                MODF0190
      COMPLEX    ELIM( 350  ), SBB( NS  ), DCI(NC), DCMOD(NC),  SUM      MODF0200
C                                                                        MODF0210
      REWIND  MASTSB                                                     MODF0220
      REWIND  NEWTSB                                                     MODF0230
C                                                                        MODF0240
      WRITE  (6,30)  NC, NS                                              MODF0242
      DO   20   I = 1, NC                                                MODF0250
      SUM       =  (0.0, 0.0)                                            MODF0260
      READ   (MASTSB)  SBB                                               MODF0270
C                                                                        MODF0280
      DO  10   J = 1, NS                                                 MODF0300
      IF  (J .NE. JARR(J))  GO TO  10                                    MODF0310
      SUM       =   SUM + SBB(J)    * ELIM(J)   *  SQRTT(J)              MODF0320
      SBB(J)    =   (0.0, 0.0)                                           MODF0330
   10 CONTINUE                                                           MODF0340
      DCMOD(I) =  DCI(I) - SUM                                           MODF0350
      WRITE  (NEWTSB)    SBB                                             MODF0360
      WRITE  (6,32)  I, (K, SBB(K),  K = 1, NS)                          MODF0370
   20 CONTINUE                                                           MODF0380
C                                                                        MODF0390
      REWIND  NEWTSB                                                     MODF0400
```

```
      LL          =   LL + 1                                               PHIJ0450
      GO TO   90                                                           PHIJ0460
C                                                                          PHIJ0470
  100 CONTINUE                                                             PHIJ0480
      LLMAX       =   LL - 1                                               PHIJ0490
      WRITE   (NPOT,40)   MODENO, ITYPE    , NL    , AL     , BL           PHIJ0492
     1     ,      ( (LIML(J, LL),   J = 1, 2),   LL = 1, LLMAX)            PHIJ0494
      LL          =   1                                                    PHIJ0500
      DO   220   J =  1, NP                                                PHIJ0510
      PHI(J)      =   (0.0, 0.0)                                           PHIJ0520
  110 CONTINUE                                                             PHIJ0530
      LIM1        =   LIML(1, LL)                                          PHIJ0540
      LIM2        =   LIML(2, LL)                                          PHIJ0550
      IF   (J .GE. LIM1 .AND. J .LE. LIM2)   GO TO   120                   PHIJ0560
      IF   (LL .EQ. LLMAX)                   GO TO   220                   PHIJ0570
      LL          =   LL + 1                                               PHIJ0580
      GO TO   110                                                          PHIJ0590
  120 CONTINUE                                                             PHIJ0600
      PHI(J)      =   (1.0, 0.0)                                           PHIJ0610
      IF   (NL .EQ.0.0) GO TO   210                                        PHIJ0620
      GO TO   (130,140,150,160,170,180),   ITYPE                           PHIJ0630
  130 PHR         =  (ABS(X(J)  - AL))** NL                                PHIJ0640
      GO TO   200                                                          PHIJ0650
  140 PHR         =  (ABS(Y(J)  - AL))** NL                                PHIJ0660
      GO TO   200                                                          PHIJ0670
  150 PHR         =  (ABS(Z(J)  - AL))** NL                                PHIJ0680
      GO TO   200                                                          PHIJ0690
  160 EARG        =   BL * (ABS(X(J)  - AL))  ** NL                        PHIJ0700
      GO TO   190                                                          PHIJ0710
  170 EARG        =   BL * (ABS(Y(J)  - AL))  ** NL                        PHIJ0720
      GO TO   190                                                          PHIJ0730
  180 EARG        =   BL * (ABS(Z(J)  - AL))  ** NL                        PHIJ0740
  190 CONTINUE                                                             PHIJ0750
      PHI(J)      =   CMPLX( EXP( EARG) ,    0.0)                          PHIJ0760
      GO TO   210                                                          PHIJ0770
  200 CONTINUE                                                             PHIJ0780
      PHI(J)      =   CMPLX(PHR, 0.0)                                      PHIJ0790
  210 CONTINUE                                                             PHIJ0800
  220 CONTINUE                                                             PHIJ0810
  230 WRITE   (NTPHIJ)   MODENO,   (PHI(J),   J = 1, NP)                   PHIJ0820
      IF   (L .EQ. NEM)          GO TO   240                               PHIJ0830
      L           =   L + 1                                                PHIJ0840
      GO TO   60                                                           PHIJ0850
  240 CONTINUE                                                             PHIJ0860
C                                                                          PHIJ0870
C          PHI MATRIX COMPLETE    (DIMENSION  NP-BY-NEM)                   PHIJ0880
C          SAVED ON TAPE    NTPHIJ  IN COLUMN ORDER                        PHIJ0890
C                                                                          PHIJ0900
      REWIND   NTPHIJ                                                      PHIJ0910
      WRITE   (NPOT,260)                                                   PHIJ0920
      DO   250   I = 1, NEM                                                PHIJ0930
      READ    (NTPHIJ)   MODENO, PHI                                       PHIJ0940
      WRITE   (NPOT,270)   MODENO                                          PHIJ0950
```

CORRECTION FACTORS (E1GC) 02/03/76

```
30 FORMAT  ( 1H1 /// 6H   THE, I4, 4H  BY, I4, 22H  S-DOUBLE-BAR MATRMODF0410
  1IX  /   )                                                        MODF0412
32 FORMAT  (//8H     ROW, I4 // (3 ( I6, 2E14.6 ) ) )              MODF0420
40 FORMAT  ( 8F13.6 )                                               MODF0420
50 FORMAT  ( /// 20H   DELTA-C-MOD              /  )                MODF0430
   WRITE  (6,50)                                                    MODF0440
   WRITE  (6,40)   (DCMOD(I),  I = 1, NC)                           MODF0450
C                                                                   MODF0460
   RETURN                                                           MODF0470
   END                                                              MODF0480
   SUBROUTINE PHIJ(NPIT, NPOT, NTPHIJ, NEM, NP, KODE, MODES,        PHIJ0040
  1               X    , Y   , Z     ,           PHI           )    PHIJ0050
C                                                                   PHIJ0060
C      NPIT       DATA SET NUMBER OF THE SYSTEM INPUT  DATA SET     PHIJ0070
C      NPOT       DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET     PHIJ0080
C      NTPHIJ     TAPE NUMBER CONTAINING COLUMNS OF THE  PHI  MATRIX PHIJ0090
C      NEM        NUMBER OF COLUMNS IN THE  PHI  MATRIX             PHIJ0100
C      NP         NUMBER OF ROW ELEMENTS IN THE  PHI  MATRIX        PHIJ0110
C      KODE       = -1                                              PHIJ0120
C      MODES      NOT USED                                          PHIJ0130
C      X, Y, Z    COORDINATES OF THE PRESSURE POINTS                PHIJ0140
C      PHI        COMPLEX ARRAY CONTAINING ONE COLUMN OF THE        PHIJ0150
C                 NP-BY-NEM  PHI  MATRIX                            PHIJ0160
C                                                                   PHIJ0170
C                                                                   PHIJ0180
   DIMENSION  LIML(2, 25), X(NP), Y(NP), Z(NP)                      PHIJ0190
   COMPLEX    PHI( NP), PHI1, PHI2                                  PHIJ0200
   REAL       NL                                                    PHIJ0202
10 FORMAT  ( 8I10 )                                                 PHIJ0210
20 FORMAT  ( 2I10, 5F10.0 )                                         PHIJ0220
30 FORMAT  ( 2( I10, 2F10.0 ) )                                     PHIJ0230
40 FORMAT  ( 2I10, F12.4, 2F13.5, 2I10 /(6(58X, 2I8)) )             PHIJ0240
50 FORMAT  ( /// 79H  MODENO(L)    TYPE(L)      N(L)      A(L)      PHIJ0250
  1  B(L)           LIM1       LIM2   /  )                          PHIJ0260
C                                                                   PHIJ0270
   REWIND  NTPHIJ                                                   PHIJ0280
   L         =  1                                                   PHIJ0290
C                                                                   PHIJ0300
60 READ   (NPIT,20)  MODENO, ITYPE   , NL  , AL   , BL              PHIJ0310
C                                                                   PHIJ0320
   IF  (ITYPE .NE. 0)      GO TO  80                                PHIJ0330
   CALL        ZEROUT( PHI   , NP,   1 ,  0  )                      PHIJ0340
70 READ   (NPIT,30)  J  , PHI1   ,   JP1, PHI2                      PHIJ0350
   IF  (J .LE. KODE)  GO TO  230                                    PHIJ0360
   PHI(J)    =  PHI1                                                PHIJ0370
   IF  (JP1 .NE. 0)  PHI(JP1) = PHI2                                PHIJ0380
   GO TO  70                                                        PHIJ0390
C                                                                   PHIJ0400
80 CONTINUE                                                         PHIJ0410
   IF  (L .EQ. 1)  WRITE  (NPOT,50)                                 PHIJ0412
   LL        =  1                                                   PHIJ0420
90 READ   (NPIT,10)  (LIML(J, LL), J = 1, 2)                        PHIJ0430
   IF  (LIML(1, LL) .LE. KODE)  GO TO  100                          PHIJ0440
```

```
      WRITE  (NPOT,280)   PHI                                               PHIJ0960
  250 CONTINUE                                                              PHIJ0970
  260 FORMAT   ( 1H1 /// 16H    -PHI- MATRIX    / )                         PHIJ0980
  270 FORMAT   ( /     9H    COLUMN , I3      / )                           PHIJ0990
  280 FORMAT   ( 8F13.6 )                                                   PHIJ1000
C                                                                           PHIJ1010
      RETURN                                                                PHIJ1020
      END                                                                   PHIJ1030
      SUBROUTINE POSN(   NT,    IGO   )                                     POSN0040
C                                                                           POSN0050
C         NT          TAPE NUMBER TO BE POSITIONED FOR READING              POSN0060
C         IGO         1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES     POSN0070
      REWIND  NT                                                            POSN0080
      READ   (NT)  NROW, NCOL1, NCOL2                                       POSN0090
      IF  (IGO .EQ. 1 .OR. NCOL1 .EQ. 0)   RETURN                           POSN0100
      DO  10   J = 1, NCOL1                                                 POSN0110
      READ     (NT)                                                         POSN0120
   10 CONTINUE                                                              POSN0130
C                                                                           POSN0140
      RETURN                                                                POSN0150
      END                                                                   POSN0160
      SUBROUTINE RECD( NTAPE,     A,     N   )                              RECD0040
C         NTAPE       TAPE NUMBER                                           RECD0050
C         A           ARRAY TO BE READ FROM TAPE                            RECD0060
C         N           LENGTH OF ARRAY  A                                    RECD0070
      DIMENSION  A(     N     )                                             RECD0080
      READ    (NTAPE)  A                                                    RECD0090
      RETURN                                                                RECD0100
      END                                                                   RECD0110
      SUBROUTINE SAIJ(NPIT, NPOT, NTSAIJ, NTSANJ, NC, NP, NMON, NAXIS ,     SAIJ0040
     1 AIT, CIE, X, Y, Z, CG, SG, DELA   , FLAGA , FLAGF , KODE, IPRINT,    SAIJ0050
     2             LABEL,    SAI                                       )    SAIJ0060
C                                                                           SAIJ0070
C         NPIT        DATA SET NUMBER OF THE SYSTEM  INPUT DATA SET         SAIJ0080
C         NPOT        DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET         SAIJ0090
C         NTSAIJ      TAPE NUMBER CONTAINING THE MATRIX ROWS  SAI           SAIJ0100
C         NTSANJ      TAPE NUMBER CONTAINING THE MATRIX ROWS  SAN           SAIJ0110
C         NC          NUMBER OF CONSTRAINTS                                 SAIJ0120
C         NP          NUMBER OF DELTA-CP  ELEMENTS                          SAIJ0130
C         NMON        NUMBER OF SETS OF THE MONITORING DATA                 SAIJ0140
C         NAXIS       NUMBER OF AXES                                        SAIJ0150
C         AIT         CONSTRAINING EFFECTIVENESS OF EXPERIMENTAL DATA       SAIJ0160
C         CIE         EXPERIMENTAL (OR ANY OTHER) CONSTRAINT ON THE DATA    SAIJ0170
C         X, Y, Z     COORDINATES OF THE PRESSURE POINTS  (BOXES)           SAIJ0180
C         CG, SG      COSINE-, SINE OF BOX DIHEDRAL ANGLES                  SAIJ0190
C         DELA        BOX AREAS                                             SAIJ0200
C         FLAGA       0,  AXIS ENDPOINTS ARE INPUT                          SAIJ0210
C                     1,  DIRECTION COSINES ARE INPUT                       SAIJ0220
C         FLAGF       0,  CIE  IS A FORCE IN DIRECTION OF AXIS              SAIJ0230
C                     1,  CIE  IS A MOMENT ABOUT AXIS                       SAIJ0240
C         KODE        = -1                                                  SAIJ0250
C         IPRINT      DETAIL PRINT FLAG                                     SAIJ0252
C                     IPRINT = 1,  PRINT SAI  AND  SAN  ROWS                SAIJ0254
```

```
                    CORRECTION FACTORS   (E1GC)   02/03/76
                    ----------------------------------
C                   IPRINT = 0,  NO  PRINT                              SAIJ0256
C        LABEL      ALPHAMERIC IDENTIFIER OF THE INTEGRATED PARAMETERS  SAIJ0260
C        SAI        COMPLEX ARRAY CONTAINING A ROW OF EITHER ONE OF THE SAIJ0270
C                   INTEGRATION MATRICES  SAI  OR  SAN                  SAIJ0272
C                                                                       SAIJ0280
      INTEGER    FLAGA, FLAGF                                           SAIJ0290
      DIMENSION        X(NP), Y(NP), Z(NP), CG(NP), SG(NP), DELA(NP),SAIJ0300
     1           JAX(35), IFF(35), DELI(35), EMI(35), AIT(35), CIT(35)  SAIJ0310
      DIMENSION  IAX(25), IFA(25), XI1(25), ETA1(25), ZETA1(25),        SAIJ0320
     1    LIMI(2,35), LIMN(2,35),  XI2(25), ETA2(25), ZETA2(25)         SAIJ0330
      DIMENSION  NAX(35), IFN(35), LABEL(10, 35),                       SAIJ0340
     1           DELN(35), EMN(35), ANT(35), CNT(35)                    SAIJ0350
      COMPLEX    CIE(35), SAI(NP)                                       SAIJ0360
C                                                                       SAIJ0370
   10 FORMAT  ( 8I10 )                                                  SAIJ0380
   12 FORMAT  ( 34X, 2I7 )                                              SAIJ0382
   20 FORMAT  ( 8F10.0 )                                                SAIJ0390
   22 FORMAT  ( 1H1 // )                                                SAIJ0392
   30 FORMAT  ( 2I10, 4F10.0 )                                          SAIJ0400
   40 FORMAT  ( 2I10, 2F10.0, 10A1 )                                    SAIJ0410
   50 FORMAT  ( 1H1///6H   THE, I4, 32H SETS OF INPUT DATA FOR ALL AXES SAIJ0420
     1          //102H  AXISNO(R)  FLAGA(R)    XI1(R)        ETA1(R)    SAIJ0430
     2  ZETA1(R)        XI2(R)        ETA2(R)       ZETA2(R)   /  66X,  SAIJ0440
     3     37H(COSA(R))     (COSB(R))     (COSG(R))   /  )              SAIJ0450
   60 FORMAT  ( 2I10, 3F14.6, 1E14.6, 2F14.6 )                          SAIJ0460
   70 FORMAT  ( 2I10, 2I7, 10X, 1F16.6, 1E16.6, 2F14.6 )                SAIJ0470
   80 FORMAT  ( 2I10, 2I7, 10X, 1F16.6, 1E16.6, 6X, 10A1 )              SAIJ0480
   90 FORMAT  ( /// 6H   THE, I4, 44H SETS OF INPUT DATA FOR ALL CONSTRASAIJ0490
     1INTS      // 93H   AXISNO(I)  FLAGF(I)  DELTA      M    LIM1   LIM2 SAIJ0500
     2  A-WIG(I)        C-WIG(I)          C-E(I)   /   82X,             SAIJ0510
     3     20HREAL             IMAG.   /  )                             SAIJ0520
  100 FORMAT  ( /// 6H   THE, I4, 35H SETS OF INPUT DATA FOR MONITORING SAIJ0530
     1          // 90H   AXISNO(N)  FLAGF(N)  DELTA      M    LIM1   LIM2 SAIJ0540
     2  A-WIG(N)        C-WIG(N)        LABEL    /  )                   SAIJ0550
C                                                                       SAIJ0560
      IR          =   1                                                 SAIJ0570
      WRITE   (NPOT,50)   NAXIS                                         SAIJ0580
  110 READ    (NPIT,10)   IAX(IR), IFA(IR)                              SAIJ0590
      READ    (NPIT,20)                      XI1(IR), ETA1(IR), ZETA1(IR) SAIJ0600
     1                                     , XI2(IR), ETA2(IR), ZETA2(IR) SAIJ0610
      WRITE   (NPOT,60)   IAX(IR), IFA(IR), XI1(IR), ETA1(IR), ZETA1(IR) SAIJ0620
     1                                     , XI2(IR), ETA2(IR), ZETA2(IR) SAIJ0630
      IF   (IR .EQ. NAXIS)       GO TO  120                             SAIJ0640
      IR          =   IR + 1                                            SAIJ0650
      GO TO   110                                                       SAIJ0660
C                                                                       SAIJ0670
  120 CONTINUE                                                          SAIJ0680
      IF   (NC .EQ. 0)   GO TO   160                                    SAIJ0690
      WRITE   (NPOT,22)                                                 SAIJ0692
      I           =   1                                                 SAIJ0700
      WRITE   (NPOT,90)   NC                                            SAIJ0710
  130 READ    (NPIT,10)   JAX(I), IFF(I)                                SAIJ0720
      READ    (NPIT,30)   NDI    , MI     , AIT(I), CIT(I), CIE(I)      SAIJ0730
```

```
      WRITE  (NPOT,70)   JAX(I), IFF(I), NDI   , MI      , AIT(I),         SAIJ0740
     1                   CIT(I), CIE(I)                                    SAIJ0750
      II         =  1                                                      SAIJ0760
  140 READ   (NPIT,10)   (LIMI(J, II),  J = 1, 2)                          SAIJ0770
      IF  (LIMI(1, II) .LE. KODE)  GO TO  150                              SAIJ0780
      WRITE  (NPOT,12)   (LIMI(J, II),  J = 1, 2)                          SAIJ0782
      II         =  II + 1                                                 SAIJ0790
      GO TO  140                                                           SAIJ0800
C                                                                          SAIJ0810
  150 CONTINUE                                                             SAIJ0820
      IIMAX      =  II - 1                                                 SAIJ0830
C                                                                          SAIJ0840
C         COMPUTE ELEMENTS OF THE SA-I-J  MATRIX  (DIM.  NC-BY-NP)         SAIJ0850
C         AND SAVE MATRIX ON TAPE  NTSAIJ  IN ROW ORDER                    SAIJ0860
C                                                                          SAIJ0870
      FLAGF      =  IFF(I) + 1                                             SAIJ0880
      IR         =  JAX(I)                                                 SAIJ0890
      FLAGA      =  IFA(IR)+ 1                                             SAIJ0900
C                                                                          SAIJ0910
      IDO        =  1                                                      SAIJ0920
      CALL          SROW(FLAGA, FLAGF, XI1  , ETA1 , ZETA1, CG, SG, CIT    SAIJ0930
     1              ,    X    , Y    , Z    , DELA , LIMI , IIMAX , I      SAIJ0940
     2              ,    NP   , IR   , XI2  , ETA2 , ZETA2,   SAI       )  SAIJ0950
C                                                                          SAIJ0960
C                                                                          SAIJ0970
      WRITE  (NTSAIJ )   NDI   , MI     , SAI                              SAIJ0980
      IF  (I .EQ. NC)          GO TO  160                                  SAIJ0990
      I          =  I + 1                                                  SAIJ1000
      GO TO  130                                                           SAIJ1010
C                                                                          SAIJ1020
  160 CONTINUE                                                             SAIJ1030
      IF  (NMON .EQ. 0)  GO TO  200                                        SAIJ1040
      WRITE  (NPOT,22)                                                     SAIJ1042
C                                                                          SAIJ1050
C         SAI  MATRIX COMPLETE  (DIMENSION  NC-BY-NP)                      SAIJ1060
C         SAVED ON TAPE  NTSAIJ  IN ROW ORDER                              SAIJ1070
C                                                                          SAIJ1080
      IF  (IDO .EQ. 2)  GO TO  200                                         SAIJ1090
      IDO        =  2                                                      SAIJ1100
      REWIND  NTSANJ                                                       SAIJ1110
      N          =  1                                                      SAIJ1120
      WRITE  (NPOT,100)  NMON                                              SAIJ1130
C                                                                          SAIJ1140
  170 READ   (NPIT,10)   NAX(N), IFN(N)                                    SAIJ1150
      READ   (NPIT,40)   NDN    , MN     , ANT(N), CNT(N),                 SAIJ1160
     1                   (LABEL(M, N),  M = 1, 10)                         SAIJ1170
      WRITE  (NPOT,80)   NAX(N), IFN(N), NDN    , MN     , ANT(N),         SAIJ1180
     1                   CNT(N), (LABEL(M, N), M = 1, 10)                  SAIJ1190
C                                                                          SAIJ1200
      NN         =  1                                                      SAIJ1210
  180 READ   (NPIT,10)   (LIMN(J, NN),  J = 1, 2)                          SAIJ1220
      IF  (LIMN(1, NN) .LE. KODE)  GO TO  190                              SAIJ1230
      WRITE  (NPOT,12)   (LIMN(J, NN),  J = 1, 2)                          SAIJ1232
```

```
      NN         =  NN + 1                                                SAIJ1240
      GO TO  180                                                          SAIJ1250
  190 CONTINUE                                                            SAIJ1260
      NNMAX      =  NN - 1                                                SAIJ1270
C                                                                         SAIJ1280
C          COMPUTE ELEMENTS OF THE  SA-N-J  MATRIX  (DIM.  NMON-BY-NP)   SAIJ1290
C          AND SAVE MATRIX ON TAPE  NTSANJ  IN ROW ORDER                  SAIJ1300
C                                                                         SAIJ1310
      FLAGF      =  IFN(N) + 1                                            SAIJ1320
      IR         =  NAX(N)                                                SAIJ1330
      FLAGA      =  IFA(IR)+ 1                                            SAIJ1340
C                                                                         SAIJ1350
      CALL          SROW(FLAGA, FLAGF, XI1  , ETA1 , ZETA1, CG, SG, CNT  SAIJ1360
     1              ,     X    , Y    , Z    , DELA , LIMN , NNMAX , N   SAIJ1370
     2              ,.    NP   , IR   , XI2  , ETA2 , ZETA2,   SAI     ) SAIJ1380
C                                                                         SAIJ1390
      WRITE  (NTSANJ )  NDN    , MN     , SAI                             SAIJ1400
C                                                                         SAIJ1410
      IF  (N .EQ. NMON)       GO TO  200                                  SAIJ1420
      N          =  N + 1                                                 SAIJ1430
      GO TO  170                                                          SAIJ1440
C                                                                         SAIJ1450
  200 CONTINUE                                                            SAIJ1460
C                                                                         SAIJ1470
C          SAN  MATRIX COMPLETE  (DIMENSION  NMON-BY-NP)                  SAIJ1480
C          SAVED ON TAPE  NTSANJ                                          SAIJ1490
C                                                                         SAIJ1500
      IF   (IPRINT .EQ. 0)  GO TO  270                                    SAIJ1502
      IF   (NC .EQ. 0)  GO TO  220                                        SAIJ1510
C                                                                         SAIJ1520
      REWIND  NTSAIJ                                                      SAIJ1530
      WRITE  (NPOT,240)                                                   SAIJ1540
      DO  210  I = 1, NC                                                  SAIJ1550
      READ   (NTSAIJ)   NDI    , MI     , SAI                             SAIJ1560
      WRITE  (NPOT,260)  (SAI(J),  J = 1, NP)                             SAIJ1570
  210 CONTINUE                                                            SAIJ1580
  220 CONTINUE                                                            SAIJ1590
      IF  (NMON .EQ. 0)  GO TO  270                                       SAIJ1600
      REWIND  NTSANJ                                                      SAIJ1610
      WRITE  (NPOT,250)                                                   SAIJ1620
      DO  230  I = 1, NMON                                                SAIJ1630
      READ   (NTSANJ)   NDN    , MN     , SAI                             SAIJ1640
      WRITE  (NPOT,260)  (SAI(J),  J = 1, NP)                             SAIJ1650
  230 CONTINUE                                                            SAIJ1660
      REWIND  NTSAIJ                                                      SAIJ1670
      REWIND  NTSANJ                                                      SAIJ1680
  240 FORMAT  ( 1H1 /// 18H   SAI MATRIX ROWS / )                         SAIJ1690
  250 FORMAT  ( 1H1 /// 18H   SAN MATRIX ROWS / )                         SAIJ1700
  260 FORMAT  ( 8F13.6 )                                                  SAIJ1710
  270 CONTINUE                                                            SAIJ1720
      RETURN                                                              SAIJ1730
      END                                                                 SAIJ1740
      SUBROUTINE SBAR(NTSBIJ, NTAPDI, NC, NP, NS, FLAGP, FLAGT, FLAGW,   SBAR0040
```

```
                  CORRECTION FACTORS   (E1GC)   02/03/76
                  ------------------------------------
     1        I, IGO, SQRTT, AIW, SAI,  DI,  W,  DELCPB,    SBI          ) SBAR0050
C                                                                          SBAR0060
C          NTSBIJ     TAPE NUMBER CONTAINING ROWS OF THE  S-BAR  MATRIX    SBAR0070
C          NTAPDI     TAPE NUMBER CONTAINING ROWS OF THE  INVERSE-D MATRIXSBAR0080
C          NC         NUMBER OF CONSTRAINTS                                SBAR0090
C          NP         NUMBER OF  DELTA-CP  ELEMENTS                        SBAR0100
C          NS         LENGTH OF THE  S-BAR  MATRIX ROWS                    SBAR0110
C          FLAGP      OPTION FLAG FOR  DELTA-CP  AND/OR  PRE-  OR  POST-   SBAR0120
C                     MULTIPLYING CORRECTIONS                              SBAR0130
C          FLAGT      OPTION FLAG FOR WEIGHTS                              SBAR0140
C                                                                          SBAR0150
C          FLAGW      OPTION FLAG FOR NORMALWASH INPUT                     SBAR0160
C          I          INTERMEDIATE INDEX                                   SBAR0170
C          IGO        1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES    SBAR0180
C          SQRTT      SQRT(T)      --     SEE EQUATIONS FOR DEFINITION     SBAR0190
C          AIW        CONSTRAINING EFFECTIVENESS                           SBAR0200
C          SAI        A ROW OF THE  SAI  MATRIX  (INTEGRATES DELTA-CP      SBAR0210
C                     INTO COEFFICIENTS)                                   SBAR0212
C          DI         A ROW OF THE  INVERSE-D  MATRIX                      SBAR0220
C          W          A COLUMN OF THE NORMALWASH  MATRIX                   SBAR0222
C          DELCPB     A COLUMN OF THE LIFTING PRESSURE COEFFICIENTS,       SBAR0230
C                     DELTA-CP-BAR                                         SBAR0232
C          SBI        A ROW OF THE INTEGRATION MATRIX  S-BAR               SBAR0240
      DIMENSION  SQRTT(350)                                                SBAR0250
      COMPLEX    SAI(NP), DI(NP),  W(NP), DELCPB(NP), SBI(NS), SAXDI       SBAR0260
      INTEGER    FLAGP, FLAGT, FLAGW                                       SBAR0270
      DATA       EPSLON, AIFIX / .0001, 10000. /                           SBAR0280
C   ***                                                                    SBAR0290
C                                                                          SBAR0300
      IF  (FLAGP .EQ. 1 .OR. FLAGP .EQ. 4)  GO TO  20                      SBAR0310
C                                                                          SBAR0320
C   ---------------------------------                                      SBAR0330
C          FLAGP = 0, 2, 3  OPTIONS   (PRE -MULTIPLY  BRANCH)              SBAR0340
C                                                                          SBAR0350
      DO  10  J = 1, NP                                                    SBAR0360
      SAXDI    =  SAI(J) * DELCPB(J)                                       SBAR0370
      SBI(J)   =  SAXDI / SQRTT(J)                                         SBAR0380
   10 CONTINUE                                                             SBAR0390
C                                                                          SBAR0400
      GO TO  60                                                            SBAR0410
C                                                                          SBAR0420
C   ---------------------------------------                                SBAR0430
C          FLAGP = 1, 4  OPTIONS      (POST-MULTIPLY  BRANCH)              SBAR0440
C                                                                          SBAR0450
   20 CONTINUE                                                             SBAR0460
      CALL        POSN( NTAPDI,  IGO )                                     SBAR0470
C                                                                          SBAR0480
      DO  40  J = 1, NP                                                    SBAR0490
      READ   (NTAPDI)  DI                                                  SBAR0500
C                                                                          SBAR0510
      DO  30  K = 1, NP                                                    SBAR0520
      SBI(K)   =  SBI(K) + SAI(J) * DI(K)                                  SBAR0530
   30 CONTINUE                                                             SBAR0540
```

```
   40 CONTINUE                                                          SBAR0550
      IF  (FLAGP .EQ. 4)   GO TO  60                                    SBAR0560
      DO   50   J = 1, NP                                               SBAR0570
      SBI(J)    =   SBI(J) * W(J) / SQRTT(J)                            SBAR0580
   50 CONTINUE                                                          SBAR0590
C                                                                       SBAR0600
   60 CONTINUE                                                          SBAR0610
      JJ          =   NP + I                                            SBAR0620
      IF  (AIW .LE. EPSLON) GO TO  70                                   SBAR0630
      SBI(JJ)   =   (1.0 - AIW) / AIW                                   SBAR0640
      GO TO  80                                                         SBAR0650
   70 SBI(JJ)   =   AIFIX                                               SBAR0660
   80 CONTINUE                                                          SBAR0670
C                                                                       SBAR0680
C          ONE ROW OF THE S-BAR-I-J  MATRIX IS COMPLETE    ---          SBAR0690
C          WRITE IT ON TAPE   NTSBIJ                                    SBAR0700
C                                                                       SBAR0710
      WRITE  (NTSBIJ)  SBI                                              SBAR0720
      WRITE  (6    ,90)                                                 SBAR0730
      WRITE  (6    ,100)  (SBI(J),   J = 1, NS)                         SBAR0740
C                                                                       SBAR0750
   90 FORMAT  ( 1H0 /// 18H   ONE  S-BAR  ROW  /  )                     SBAR0760
  100 FORMAT  ( 8F13.6 )                                                SBAR0770
      RETURN                                                            SBAR0780
      END                                                               SBAR0790
      SUBROUTINE SDBL(NSCRCH,NUTL,MASTSB,NTPHIJ, NTAPW, NTSAIJ, NTAPDI, SDBL0040
     1    IGO, FLAGW, FLAGP, FLAGT, NC, NP, NS, NEM,  SQRTT ,   AIT,    SDBL0050
     2   DELA,    SBB, SBI, SAI, DI , W  , PHI, DELCPB,   WORK        ) SDBL0060
C                                                                       SDBL0070
C                                                                       SDBL0080
C          NUTL       UTILITY (SCRATCH) TAPE NUMBER                     SDBL0090
C          MASTSB     TAPE NUMBER CONTAINING THE  SBB  MATRIX ROWS      SDBL0100
C          NTPHIJ     TAPE NUMBER CONTAINING THE  PHI  MATRIX COLUMNS   SDBL0110
C          NTAPW      TAPE NUMBER CONTAINING THE   W   MATRIX COLUMNS   SDBL0120
C          NTSAIJ     TAPE NUMBER CONTAINING THE  SAI  MATRIX ROWS      SDBL0130
C          NTAPDI     TAPE NUMBER CONTAINING THE  INVERSE-D  MATRIX ROWS SDBL0140
C          IGO        1 FOR SYMMETRIC MODES,  2 FOR ANTISYMMETRIC MODES SDBL0150
C          FLAGW      OPTION FLAG FOR NORMALWASH INPUT                  SDBL0160
C          FLAGP      OPTION FLAG FOR  DCP-INPUT AND/OR  PRE- OR POST-  SDBL0170
C          FLAGT      OPTION FLAG FOR  WEIGHTS                          SDBL0180
C          NC         NUMBER OF CONSTRAINTS                             SDBL0190
C          NP         NUMBER OF  DELTA-CP  ELEMENTS                     SDBL0200
C          NS         =  MAX (NP+NC, NEM+NC)                            SDBL0210
C          NEM        NUMBER OF CORRECTION FACTOR MODES                 SDBL0220
C          SQRTT      =  SQRT(T)     --     SEE EQUATIONS FOR DEFINITION SDBL0230
C          AIT        CONSTRAINING EFFECTIVENESS                        SDBL0240
C          DELA       BOX AREAS                                         SDBL0250
C          SBB        A ROW OF THE  S-DOUBLE-BAR  MATRIX                SDBL0260
C          SBI        A ROW OF THE  S-BAR  MATRIX                       SDBL0270
C          SAI        A ROW OF THE  SAI MATRIX                          SDBL0280
C          DI         A ROW OF THE  INVERSE-D  MATRIX                   SDBL0290
C          W          A COLUMN OF THE  W  MATRIX  (NORMALWASH)          SDBL0292
C          PHI        A COLUMN OF THE  PHI  MATRIX                      SDBL0300
```

```
          CORRECTION FACTORS   (EIGC)   02/03/76
          ----------------------------------------
C     NSCRCH    TAPE NUMBER CONTAINING THE PHI-BAR  MATRIX (IF ANY) SDBL0080
C     DELCPB    A COLUMN OF THE  DELTA-CP-BAR  MATRIX               SDBL0310
      INTEGER     FLAGP, FLAGT, FLAGW                               SDBL0320
      DIMENSION   SQRTT(350), AIT(NC),  DELA(350)                   SDBL0330
      COMPLEX     SBB(NS ), SBI(NS), SAI(NP), DI(NP), W(NP), DELCPB(NP) SDBL0340
     1            ,  PHI(NP),  WORK(NP,  1  )                       SDBL0350
      COMPLEX     EL      , PHIBAR(350), COL(350)                   SDBL0360
      NTSBIJ   =  MASTSB                                            SDBL0370
      IFP      =  FLAGP + 1                                         SDBL0380
      IF  (FLAGP .EQ. 4)  REWIND  NSCRCH                            SDBL0390
      IF  (NEM .NE. 0)  NTSBIJ = NUTL                               SDBL0400
      REWIND  NTSAIJ                                                SDBL0410
      REWIND  NTSBIJ                                                SDBL0420
      MI       =  1                                                 SDBL0430
   10 CONTINUE                                                      SDBL0440
      DO  20  J = 1, NP                                             SDBL0450
   20 DELCPB(J)=  WORK(J, MI)                                       SDBL0460
      GO TO  (50,30,50,50,30),         IFP                          SDBL0470
   30 CONTINUE                                                      SDBL0480
      CALL        POSN( NTAPW,  IGO )                               SDBL0490
   40 READ   (NTAPW)  W                                             SDBL0500
   50 CONTINUE                                                      SDBL0510
      DO  110  J = 1, NP                                            SDBL0520
      SQRTT(J) =  1.0                                               SDBL0530
      IF  (FLAGT .EQ. 1)  GO TO  110                                SDBL0540
      GO TO  (100,60,100,100,60),         IFP                       SDBL0550
   60 CONTINUE                                                      SDBL0560
C                                                                   SDBL0570
C          POSTMULTIPLIER TYPE                                      SDBL0580
C                                                                   SDBL0590
      IF  (J .GT. 1)  GO TO  80                                     SDBL0600
      CALL        POSN( NTAPDI, IGO )                               SDBL0610
      DO  70  K = 1, NP                                             SDBL0620
   70 COL(K)   =  (0.0, 0.0)                                        SDBL0630
C                                                                   SDBL0640
   80 READ   (NTAPDI)  DI                                           SDBL0650
      DO  90  N = 1, NP                                             SDBL0660
   90 COL(N)   =  COL(N) + DI(N) * DELA(J)                          SDBL0670
      GO TO  110                                                    SDBL0680
  100 CONTINUE                                                      SDBL0690
C                                                                   SDBL0700
C          PREMULTIPLIER  TYPE                                      SDBL0710
C                                                                   SDBL0720
      SQRTT(J) =  SQRT( CABS ( DELCPB(J)*DELA(J) ) )                SDBL0730
  110 CONTINUE                                                      SDBL0740
C                                                                   SDBL0750
      IF  (FLAGT .EQ. 1)  GO TO  140                                SDBL0760
      GO TO  (140,120,140,140,120),        IFP                      SDBL0770
  120 CONTINUE                                                      SDBL0780
      DO  130  N = 1, NP                                            SDBL0790
  130 SQRTT(N) =  SQRT( CABS( COL(N)) )                             SDBL0800
  140 CONTINUE                                                      SDBL0810
C                                                                   SDBL0820
```

```
      DO   200   I = 1, NC                                               SDBL0830
      AIW         =  AIT(I)                                              SDBL0840
      CALL          ZEROUT( SBI, NS,  1 ,  0 )                           SDBL0850
      READ    (NTSAIJ)  NDI, MI, SAI                                     SDBL0860
C                                                                        SDBL0870
  150 CONTINUE                                                           SDBL0880
      DO   160   J = 1, NP                                               SDBL0890
  160 DELCPB(J)=  WORK(J, MI)                                            SDBL0900
      GO TO  (190,170,170,190,170),   IFP                                SDBL0910
  170 CONTINUE                                                           SDBL0920
      CALL          POSN( NTAPW, IGO )                                   SDBL0930
      DO   180   M = 1, MI                                               SDBL0940
  180 READ    (NTAPW)   W                                                SDBL0950
C                                                                        SDBL0960
  190 CONTINUE                                                           SDBL0970
C                                                                        SDBL0980
      CALL          SBAR(NTSBIJ, NTAPDI, NC, NP, NS, FLAGP, FLAGT, FLAGW, SDBL0990
     1         I, IGO, SQRTT, AIW, SAI, DI , W  , DELCPB,      SBI      ) SDBL1000
C                                                                        SDBL1010
  200 CONTINUE                                                           SDBL1020
      IF   (NEM .EQ. 0)  GO TO  290                                      SDBL1030
C                                                                        SDBL1040
C                                                                        SDBL1050
      REWIND   NUTL                                                      SDBL1060
      REWIND   MASTSB                                                    SDBL1070
      DO   205   J = 1, NP                                               SDBL1072
  205 DELCPB(J)=  WORK(J,   1 )                                          SDBL1074
      NEM1        =  NEM + 1                                             SDBL1080
      NEMNC       =  NEM + NC                                            SDBL1090
      WRITE   (6   ,300)   NC, NEMNC                                     SDBL1100
      DO   280   I = 1, NC                                               SDBL1110
      REWIND   NTPHIJ                                                    SDBL1120
      READ    (NUTL )     SBI                                            SDBL1130
      CALL          ZEROUT( SBB   , NS   ,    1,    0 )                  SDBL1140
      DO   260   K = 1, NEM                                              SDBL1150
      READ    (NTPHIJ)   MODENO, PHI                                     SDBL1160
      IF   (FLAGP .EQ. 4)   GO TO  220                                   SDBL1170
      DO   210   J = 1, NP                                               SDBL1180
      SBB(K)      =  SBB(K) + SBI(J) * PHI(J)                            SDBL1190
  210 CONTINUE                                                           SDBL1200
      GO TO  260                                                         SDBL1210
C                                                                        SDBL1220
  220 CONTINUE                                                           SDBL1230
      EL          =  (0.0, 0.0)                                          SDBL1240
      DO   230   L = 1, NP                                               SDBL1250
      EL          =  EL + PHI(L) * DELCPB(L)                             SDBL1260
  230 CONTINUE                                                           SDBL1270
C                                                                        SDBL1280
      CALL          ZEROUT( PHIBAR,  350 ,  1   ,  0  )                  SDBL1290
      DO   240   M = 1, NP                                               SDBL1300
      PHIBAR(M)=  PHI(M) * EL                                            SDBL1310
  240 CONTINUE                                                           SDBL1320
      WRITE  (NSCRCH)  K,  (PHIBAR(M),  M = 1, NP)                       SDBL1330
```

```
C
      DO   250   J = 1, NP
      SBB(K)     =  SBB(K) + SBI(J) * PHIBAR(J)
  250 CONTINUE
C
  260 CONTINUE
      DO   270   K =NEM1, NEMNC
      NPK        =  NP + K - NEM
      SBB(K)     =  SBI(NPK)
  270 CONTINUE
      WRITE  (MASTSB)    SBB
      WRITE  (6  ,310)   I, (K, SBB(K),   K = 1, NEMNC)
  280 CONTINUE
      REWIND   MASTSB
C
  290 CONTINUE
      REWIND   NTSBIJ
  300 FORMAT   ( 1H1 /// 6H   THE, I4, 4H   BY, I4, 22H   S-DOUBLE-BAR MATR
     1IX    /    )
  310 FORMAT   (//8H      ROW, I4 // (3 ( I6, 2E14.6 ) ) )
      IF   (FLAGP .EQ. 4)  REWIND   NSCRCH
      RETURN
      END
      SUBROUTINE SROW(FLAGA, FLAGF, XI1   , ETA1 , ZETA1, CG, SG, CTIL
     1             ,    X     , Y     , Z     , DELA , LIMI , IIMAX , I
     2             ,    NP    , IR    , XI2   , ETA2 , ZETA2,    SAI        )
C
C         FLAGA     0,  AXIS ENDPOINTS ARE INPUT
C                   1,  DIRECTION COSINES ARE INPUT
C         FLAGF     0,  CIE  IS A FORCE IN DIRECTION OF AXIS
C                   1,  CIE  IS A MOMENT ABOUT AXIS
C
C         XI1, ETA1, ZETA1     AXIS ENDPOINT COORDINATES
C         CG, SG    COSINE-, SINE  OF BOX DIHEDRAL ANGLES
C         CTIL      CONSTANT USED TO NONDIMENSIONALIZE INTEGRATED DATA
C         X, Y, Z   COORDINATES OF THE PRESSURE POINTS  (BOXES)
C         DELA      BOX AREAS
C         LIMI      FIRST- AND LAST BOX NUMBERS FOR THE INTEGRATION OF
C                   THE  DELTA-CP  VALUES
C         IIMAX     NUMBER OF  LIMI  SETS INPUT FOR ONE CONSTRAINT
C         I         INTERMEDIATE INDEX
C         NP        NUMBER OF  DELTA-CP  VALUES
C         IR        ROW INDEX OF  SAI  MATRIX
C
C         XI2, ETA2, ZETA2     SECOND AXIS ENDPOINT COORDINATES WHEN
C                              FLAGF=0,  DIRECTION COSINES WHEN FLAGF=1
C         SAI       A ROW OF THE INTEGRATION MATRIX  (SA-IJ, OR SA-NJ)
C
      INTEGER     FLAGA, FLAGF
      DIMENSION   XI1( 25), ETA1( 25), ZETA1( 25), CG( NP), SG( NP)
     1          , XI2( 25), ETA2( 25), ZETA2( 25), CTIL( 35)
      DIMENSION   X( NP), Y( NP), Z( NP), DELA( NP), LIMI(2, 35)
      COMPLEX     SAI( NP)
```

CORRECTION FACTORS (E1GC) 02/03/76

```
C
      GO TO  (10,20),   FLAGA
   10 XID        =   XI2  (IR) - XI1  (IR)
      ETAD       =   ETA2 (IR) - ETA1 (IR)
      ZETAD      =   ZETA2(IR) - ZETA1(IR)
      P          =   SQRT(XID**2 + ETAD**2 + ZETAD**2)
      COSAI      =   XID   / P
      COSBI      =   ETAD  / P
      COSGI      =   ZETAD / P
      GO TO  30
C
   20 CONTINUE
      COSAI      =   XI2  (IR)
      COSBI      =   ETA2 (IR)
      COSGI      =   ZETA2(IR)
   30 CONTINUE
      II         =   1
      DO  90  J  =   1, NP
      ABIJ       =   0.0
   40 CONTINUE
      LIM1       =   LIMI(1, II)
      LIM2       =   LIMI(2, II)
      IF  (J .GE. LIM1 .AND. J .LE. LIM2)  GO TO  50
      IF  (II .EQ. IIMAX)                  GO TO  80
      II         =   II + 1
      GO TO  40
C
   50 CONTINUE
      GO TO  (60,70),   FLAGF
   60 CONTINUE
      ABIJ       =   (-COSBI * SG(J) + COSGI * CG(J)) / CTIL(I)
      GO TO  80
   70 CONTINUE
      XDIF       =   X(J) - XI1  (IR)
      YDIF       =   Y(J) - ETA1 (IR)
      ZDIF       =   Z(J) - ZETA1(IR)
      ABIJ       =   (COSAI * (YDIF * CG(J) + ZDIF * SG(J))
     1                -COSBI * (XDIF * CG(J))
     2                -COSGI * (XDIF * SG(J))                ) / CTIL(I)
   80 CONTINUE
      II         =   1
      SAI(J)     =   CMPLX(ABIJ*DELA(J)  ,  0.0 )
   90 CONTINUE
C
      RETURN
      END
      SUBROUTINE WEYT( NP  , NC  , NEM , NELIMS, NMON , NAXIS, NMIN  ,
     1                 NMAX  ,  NS  , NPIT, NPOT  ,              W     ,
     2                 DI    ,  DCP       , EPS   ,       PHI  , SAI   ,
     3                 DCPTIL,  DELCPB    , COL   ,       CIE  , DCMOD ,
     4                 EBMIN ,  EBMAX     , EB    ,       ELIM , SBB   ,
     5                 SBI   ,  S         , SBMAT ,       DCI  , WORK     )
C
```

```
C                  CORRECTION FACTORS  (E1GC)  02/03/76
C                  -----------------------------------
C       NP         NUMBER OF  DELTA-CP  ELEMENTS                         WEYT0090
C       NC         NUMBER OF  CONSTRAINTS                                WEYT0100
C       NEM        NUMBER OF  CORRECTION FACTOR MODES                    WEYT0110
C       NELIMS     NUMBER OF  INPUT CARDS FOR  EBMIN, EBMAX  PAIRS       WEYT0120
C       NMON       NUMBER OF  SETS OF MONITORING DATA                    WEYT0130
C       NAXIS      NUMBER OF  AXES FOR USE IN INTEGRATION OF  DELTA-CP   WEYT0140
C       NMIN       =  MAX (  1, NELIMS)                                  WEYT0150
C       NMAX       =  MAX ( NC, NMIN, 10 )                               WEYT0160
C       NS         =  MAX (NP+NC, NEM+NC)                                WEYT0170
C       NPIT       DATA SET NUMBER OF THE SYSTEM  INPUT DATA SET         WEYT0180
C       NPOT       DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET         WEYT0190
C       W          A COLUMN OF THE NORMALWASH MATRIX                     WEYT0200
C       DI         A ROW OF THE  INVERSE-D  MATRIX                       WEYT0210
C       DCP        ARRAY OF  DELTA-CP  ELEMENTS  (THEORETICAL VALUES)    WEYT0220
C       EPS        EPSILON ARRAY  (INCREMENTAL WEIGHT FACTORS)           WEYT0230
C       PHI        A COLUMN OF THE  PHI  MATRIX  (WEIGHT FACTOR MODE     WEYT0240
C                  SHAPES)                                               WEYT0250
C       SAI        INTEGRATION MATRIX ROW                                WEYT0260
C       DCPTIL     PRESSURES MODIFIED BY WEIGHT MATRIX                   WEYT0270
C       DELCPB     LIFTING PRESSURE COEFFICIENTS                         WEYT0280
C       COL        COMPLEX ARRAY FOR INTERMEDIATE USE                    WEYT0290
C       CIE        ARRAY OF THE INPUT VALUES  C-I(E)                     WEYT0300
C       DCMOD      ARRAY OF THE MODIFIED  DELTA-C  VALUES                WEYT0310
C       EBMIN      MINIMUM VALUE ALLOWED FOR  EPSILON                    WEYT0320
C       EBMAX      MAXIMUM VALUE ALLOWED FOR  EPSILON                    WEYT0322
C       EB         EPSILON-BAR ARRAY  (INCREMENTAL WEIGHT FACTORS)       WEYT0330
C       ELIM       ARRAY OF THE MODIFIED EPSILON VALUES                  WEYT0340
C       SBB        A ROW OF THE  S-DOUBLE-BAR MATRIX                     WEYT0350
C       SBI        A ROW OF THE  S-BAR  MATRIX                           WEYT0360
C       S          2-D COMPLEX WORK ARRAY,  NC-BY-NC                     WEYT0370
C       SBMAT      2-D COMPLEX  S-DOUBLE-BAR  MATRIX,  NC-BY-NS          WEYT0380
C       DCI        ARRAY OF THE  DELTA-C  VALUES                         WEYT0390
C       WORK       2-D COMPLEX ARRAY,  NP-BY-NMAX,  IN WHICH THE         WEYT0400
C                  LIFTING PRESSURE COEFFICIENT MATRIX  (DELCPB)         WEYT0402
C                  COLUMNS ARE STORED                                    WEYT0404
C                                                                        WEYT0410
      DIMENSION  X(350), Y(350), Z(350), GMA(350), CG(350), SG(350),    WEYT0420
     1           DELA(350), ROW(350),  AIT( 35),  JARR(350)          ,  WEYT0430
     2           LIMK(2, 100), LIML(2, 100), LABEL(10, 35), SQRTT(350)  WEYT0440
      COMPLEX    W(NP),   DI(NP), DCP(NP), EPS(NP), PHI(NP), SAI(NP),   WEYT0450
     1           DCPTIL(NP), DELCPB(NP), COL(NP), CIE(NC), DCMOD(NC) ,  WEYT0460
     2           EBMIN(NMIN), EBMAX(NMIN), EB(NS), ELIM(NS), SBB(NS) ,  WEYT0470
     3           SBI(NS), S(NC, NC), SBMAT(NC, NS), DCI(NC       )   ,  WEYT0480
     4           WORK(NP, NMAX),      B(35),      CE(35)                WEYT0490
C                                                                        WEYT0500
      INTEGER    FLAGA, FLAGB, FLAGP, FLAGT, FLAGW, FLAGF, FLAGI, FLAGDIWEYT0510
      DATA       KODE  , NUTL1 , NUTL2 , NTSAIJ, NTSANJ, NTPHIJ, MASTSB,WEYT0520
     1           NEWTSB, NTGEOM, NTDCP, NTAPW, NTAPDI, NEWDCP, NTAPCF   WEYT0530
     2           / -1, 1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16   / WEYT0532
      DATA       JARR / 350*0 /, SQRTT / 350*1.0 /, LINES / 48 /        WEYT0540
      DATA       LABEL / 350*1H  /,  PRE , POST / 4HPRE , 4HPOST /      WEYT0550
   10 FORMAT   ( 8I10 )                                                 WEYT0560
   20 FORMAT   ( 2I10, 4F10.0 )                                         WEYT0570
```

CORRECTION FACTORS (E1GC) 02/03/76

```
 30 FORMAT  ( /// 27H   CONTROL FLAGS          --  //   )             WEYT0580
 40 FORMAT  ( 45H    FLAGB = 0   CORRECTION FACTORS CALCULATED       ) WEYT0590
 50 FORMAT  ( 75H    FLAGB = 1   DATA MONITORED ONLY   -  NO CORRECTION WEYT0600
   1FACTORS                   )                                         WEYT0610
 52 FORMAT  ( 75H    FLAGB = 2   DATA MONITORED  -  CORRECTION FACTORS WEYT0612
   1TAKEN FROM TAPE           )                                         WEYT0614
 60 FORMAT  ( 75H    FLAGP = 0   PREMULTUPLIER  -  PRESSURE AND GEOMETRWEYT0620
   1Y TAKEN FROM TAPE         )                                         WEYT0630
 70 FORMAT  ( 75H    FLAGP = 1   POSTMULTIPLIER  -  DT INVERSE AND GEOMWEYT0640
   1ETRY TAKEN FROM TAPE      )                                         WEYT0650
 80 FORMAT  ( 75H    FLAGP = 2   PREMULTIPLIER  -  DT INVERSE AND GEOMEWEYT0660
   1TRY TAKEN FROM TAPE       )                                         WEYT0670
 90 FORMAT  ( 75H    FLAGP = 3   PREMULTIPLIER  -  PRESSURE AND GEOMETRWEYT0680
   1Y TAKEN FROM CARDS        )                                         WEYT0690
 92 FORMAT  ( 78H    FLAGP = 4   NEW POSTMULTIPLIER  -  DT INVERSE AND WEYT0692
   1GEOMETRY TAKEN FROM TAPE     )                                      WEYT0694
100 FORMAT  ( 50H    FLAGT = 0   WEIGHTS ARE A FUNCTION OF THE LOADS ) WEYT0700
110 FORMAT  ( 50H    FLAGT = 1   WEIGHTS = 1.0                       ) WEYT0710
120 FORMAT  ( 60H    FLAGW = 0   NORMALWASH TAKEN FROM TAPE     (IF NEEWEYT0720
   1DED)    )                                                           WEYT0730
130 FORMAT  ( 60H    FLAGW = 1   NORMALWASH TAKEN FROM CARDS    (IF NEEWEYT0740
   1DED)    )                                                           WEYT0750
134 FORMAT  ( 10H   IPRINT=, I2, 23H   (DETAIL PRINT FLAG)  )          WEYT0756
140 FORMAT  ( /// 27H   CONTROL DIMENSIONS    --  //                   WEYT0760
   1          10H   NP    = , I4 / 10H   NC     = , I4  /              WEYT0770
   2          10H   NEM   = , I4 / 10H   NELIMS= , I4  /               WEYT0780
   3          10H   NMON  = , I4 / 10H   NAXIS = , I4  / )             WEYT0790
150 FORMAT  ( /// 34H   LIST OF INPUT/OUTPUT TAPES   --    //          WEYT0800
   1        19H   GEOMETRY  TAPE = , I3/19H   DELTA-CP  TAPE =, I3 /   WEYT0810
   2        19H   W         TAPE = , I3/19H   D-INVERSE TAPE =, I3 /   WEYT0820
   3        19H   CORR. FACTORS  = , I3  /                           ) WEYT0822
160 FORMAT  ( I4, I9, I12, 4F16.6 )                                    WEYT0830
170 FORMAT  ( 1H1///  23H   EPSILON LIMITS    --   //                  WEYT0840
   1          48H   K     LIM-1(K)     LIM-2(K)     EPSILON-BAR-MIN  , WEYT0850
   2          17X, 15HEPSILON-BAR-MAX / 32X, 20H REAL          IMAG.   WEYT0860
   3          12X, 20H REAL              IMAG.  / )                    WEYT0870
180 FORMAT  ( 1H1 /// 32H   CARD-READ OPTION IS SPECIFIED // 6H      I WEYT0880
   1 , 6X,   1HX, 12X, 1HY, 12X, 1HZ, 11X, 5HGAMMA, 8X, 7HDELTA-A  / ) WEYT0890
190 FORMAT  ( 6F10.0 )                                                 WEYT0900
200 FORMAT  ( 8F13.6 )                                                 WEYT0910
210 FORMAT  ( I6, 8F13.6 )                                             WEYT0920
220 FORMAT  ( 1H1 /// 34H        SYMMETRIC DELTA-CP MATRIX    /  )     WEYT0930
230 FORMAT  ( 1H0 /// 34H   ANTISYMMETRIC DELTA-CP MATRIX    /  )      WEYT0940
240 FORMAT  ( / 9H   COLUMN , I3   / )                                 WEYT0950
C                                                                       WEYT1000
C        READ  AND  WRITE  CONTROL FLAGS                               WEYT1010
C                                                                       WEYT1020
      READ   (NPIT,10)  FLAGB, FLAGP, FLAGT, FLAGW, FLAGI, IPRINT      WEYT1030
      WRITE  (NPOT,30)                                                  WEYT1040
      IF   (FLAGB .EQ. 0)  WRITE  (NPOT,40)                             WEYT1050
      IF   (FLAGB .EQ. 1)  WRITE  (NPOT,50)                             WEYT1060
      IF   (FLAGB .EQ. 2)  WRITE  (NPOT,52)                             WEYT1062
      IF   (FLAGP .EQ. 0)  WRITE  (NPOT,60)                             WEYT1070
```

```
                     CORRECTION FACTORS  (E1GC)  02/03/76
                     ------------------------------------
      IF   (FLAGP .EQ. 1)  WRITE  (NPOT,70)                             WEYT1080
      IF   (FLAGP .EQ. 2)  WRITE  (NPOT,80)                             WEYT1090
      IF   (FLAGP .EQ. 3)  WRITE  (NPOT,90)                             WEYT1100
      IF   (FLAGP .EQ. 4)  WRITE  (NPOT,92)                             WEYT1102
      IF   (FLAGT .EQ. 0)  WRITE  (NPOT,100)                            WEYT1110
      IF   (FLAGT .EQ. 1)  WRITE  (NPOT,110)                            WEYT1120
      IF   (FLAGW .EQ. 0)  WRITE  (NPOT,120)                            WEYT1130
      IF   (FLAGW .EQ. 1)  WRITE  (NPOT,130)                            WEYT1140
      WRITE  (NPOT,134)   IPRINT                                        WEYT1144
C                                                                       WEYT1150
C         WRITE  CONTROL DIMENSIONS                                     WEYT1160
C                                                                       WEYT1170
      WRITE  (NPOT,140)  NP ,   NC ,   NEM ,   NELIMS, NMON , NAXIS     WEYT1180
C                                                                       WEYT1190
C         READ  NUMBER OF MODES  AND WRITE LIST OF INPUT/OUTPUT TAPES   WEYT1200
C                                                                       WEYT1210
      READ   (NPIT, 10)  NMSYM, NMASYM                                  WEYT1220
      WRITE  (NPOT,150)  NTGEOM, NTDCP, NTAPW, NTAPDI, NTAPCF           WEYT1230
C                                                                       WEYT1240
      IFP        =  FLAGP + 1                                           WEYT1250
      IF   (FLAGP .EQ. 3)                        GO TO  310             WEYT1260
C                                                                       WEYT1270
C         READ  GEOMETRY TAPE                                           WEYT1280
C                                                                       WEYT1290
      REWIND  NTGEOM                                                    WEYT1300
      READ   (NTGEOM)  LENGTH                                           WEYT1310
C                                                                       WEYT1320
      CALL          RECD( NTGEOM,  X      , LENGTH )                    WEYT1330
      CALL          RECD( NTGEOM,  Y      , LENGTH )                    WEYT1340
      CALL          RECD( NTGEOM,  Z      , LENGTH )                    WEYT1350
      CALL          RECD( NTGEOM,  GMA    , LENGTH )                    WEYT1360
      CALL          RECD( NTGEOM,  DELA   , LENGTH )                    WEYT1370
      IF   (FLAGP .NE. 0)  GO TO  500                                   WEYT1380
      IF   (FLAGI .EQ. 0)  GO TO  500                                   WEYT1390
      REWIND  NTDCP                                                     WEYT1400
      REWIND NEWDCP                                                     WEYT1410
      READ   (NTDCP)  LENGTH, NMSYM, NMASYM                             WEYT1420
      WRITE  (NEWDCP) LENGTH, NMSYM, NMASYM                             WEYT1430
      MODCP     =  NMSYM + NMASYM                                       WEYT1440
      DO  300  J = 1, MODCP                                             WEYT1450
      READ   (NTDCP)  ROW                                               WEYT1460
      DO  290  I = 1, NP                                                WEYT1470
  290 DCP(I)    =  CMPLX(ROW(I), 0.0)                                   WEYT1480
      WRITE  (NEWDCP)  DCP                                              WEYT1490
  300 CONTINUE                                                          WEYT1500
      REWIND NEWDCP                                                     WEYT1510
      NTDCP     =  NEWDCP                                               WEYT1520
      GO TO  500                                                        WEYT1530
  310 CONTINUE                                                          WEYT1540
      REWIND  NUTL1                                                     WEYT1550
      REWIND  NTDCP                                                     WEYT1570
      WRITE  (NPOT,180)                                                 WEYT1580
C                                                                       WEYT1590
```

```
C         READ GEOMETRY ARRAYS  AND  DELTA-CP MATRIX FROM  CARDS          WEYT1600
C                                                                          WEYT1610
      DO  330  I = 1, NP                                                   WEYT1620
      READ   (NPIT,190)      X(I), Y(I), Z(I), GMA(I), DELA(I)             WEYT1630
      WRITE  (NPOT,210)  I,  X(I), Y(I), Z(I), GMA(I), DELA(I)             WEYT1640
C                                                                          WEYT1650
  330 CONTINUE                                                             WEYT1710
      WRITE  (NTDCP)    NP, NMSYM, NMASYM                                  WEYT1720
      IGO       =  1                                                       WEYT1750
      IF  (NMSYM .EQ. 0)  GO TO  390                                       WEYT1760
      MODES     =  NMSYM                                                   WEYT1770
      WRITE  (NPOT,220)                                                    WEYT1780
  340 CONTINUE                                                             WEYT1790
      DO  380  J = 1, MODES                                                WEYT1800
      WRITE  (NPOT,240)  J                                                 WEYT1810
      IF  (FLAGI .NE. 0)  GO TO  350                                       WEYT1820
      READ   (NPIT,190)  (DCP(I),  I = 1, NP)                              WEYT1830
      GO TO  370                                                           WEYT1840
  350 READ   (NPIT,190)  (ROW(I),  I = 1, NP)                              WEYT1850
      DO  360  I = 1, NP                                                   WEYT1860
  360 DCP(I)    =  CMPLX(ROW(I), 0.0)                                      WEYT1870
  370 CONTINUE                                                             WEYT1880
      WRITE  (NPOT,200)  (DCP(I),  I = 1, NP)                              WEYT1890
      WRITE  (NTDCP)       (DCP(I),  I = 1, NP)                            WEYT1900
  380 CONTINUE                                                             WEYT1910
C                                                                          WEYT1920
  390 CONTINUE                                                             WEYT1930
      IF  (IGO .EQ. 2 .OR. NMASYM .EQ. 0)  GO TO  400                      WEYT1940
      IGO       =  2                                                       WEYT1950
      MODES     =  NMASYM                                                  WEYT1960
      WRITE  (NPOT,230)                                                    WEYT1970
      GO TO  340                                                           WEYT1980
  400 CONTINUE                                                             WEYT1990
      LENGTH    =  NP                                                      WEYT2000
C                                                                          WEYT2010
  500 CONTINUE                                                             WEYT2350
C                                                                          WEYT2360
      DO  510  I = 1, LENGTH                                               WEYT2370
      CG(I)     =  COS(GMA(I))                                             WEYT2380
      SG(I)     =  SIN(GMA(I))                                             WEYT2390
  510 CONTINUE                                                             WEYT2400
C                                                                          WEYT2410
      IF  (NC .EQ. 0 .AND. NMON .EQ. 0)  GO TO  520                        WEYT2430
      REWIND  NTSAIJ                                                       WEYT2440
      REWIND  NTSANJ                                                       WEYT2450
C                                                                          WEYT2460
      CALL        SAIJ(NPIT, NPOT, NTSAIJ, NTSANJ, NC, NP, NMON, NAXIS ,   WEYT2470
     1 AIT, CIE, X, Y, Z, CG, SG, DELA  , FLAGA , FLAGF , KODE, IPRINT,    WEYT2480
     2            LABEL ,  SAI                                        )    WEYT2490
C                                                                          WEYT2500
  520 CONTINUE                                                             WEYT2510
      IF  (NELIMS .EQ. 0)  GO TO  540                                      WEYT2520
      K         =  1                                                       WEYT2530
```

```
      WRITE  (NPOT,170)                                                  WEYT2540
  530 READ   (NPIT,20)     LIMK(1, K), LIMK(2, K), EBMIN(K), EBMAX(K)    WEYT2550
      WRITE  (NPOT,160)  K, LIMK(1, K), LIMK(2, K), EBMIN(K), EBMAX(K)   WEYT2560
      IF  (K .EQ. NELIMS)          GO TO  540                            WEYT2570
      K        =  K + 1                                                  WEYT2580
      GO TO  530                                                         WEYT2590
C                                                                        WEYT2600
  540 CONTINUE                                                           WEYT2610
C                                                                        WEYT2620
      IF  (NEM .EQ. 0)  GO TO  550                                       WEYT2630
C                                                                        WEYT2640
      CALL        PHIJ(NPIT, NPOT, NTPHIJ, NEM , NP,  KODE,  MODES,      WEYT2650
     1                 X    , Y   , Z     ,           PHI             )  WEYT2660
C                                                                        WEYT2670
  550 CONTINUE                                                           WEYT2680
      NROW       =  NP                                                   WEYT2690
      IF  (FLAGP .EQ. 0 .OR. FLAGP .EQ. 3)  GO TO  560                   WEYT2700
      IF  (FLAGW .NE. 1)                    GO TO  560                   WEYT2710
C                                                                        WEYT2720
      REWIND  NTAPDI                                                     WEYT2730
      READ   (NTAPDI)                                                    WEYT2740
      NCOL       =  MAXO(NMSYM, NMASYM)                                  WEYT2750
      REWIND  NTAPW                                                      WEYT2760
C                                                                        WEYT2770
      CALL        WSWA(NPIT, NPOT, NUTL1, NTAPW, KODE, NP, NCOL, NMAX,   WEYT2780
     1            NMSYM, NMASYM,                          WORK        )  WEYT2790
C                                                                        WEYT2800
C         THE TAPE  NTAPW  CONTAINS THE TWO  W-MATRICES,  WS  AND  WA,   WEYT2810
C         PRECEDED BY THE NUMBERS   NP, NMSYM, NMASYM                    WEYT2820
C                                                                        WEYT2830
  560 CONTINUE                                                           WEYT2840
      NX         =  NS                                                   WEYT2842
      IF  (NEM .NE. 0)  NX = NEM + NC                                    WEYT2844
      IGO        =  1                                                    WEYT2850
      NMODE      =  NMSYM                                                WEYT2860
      IF  (NMSYM .NE. 0)  GO TO  562                                     WEYT2862
      IGO        =  2                                                    WEYT2864
      NMODE      =  NMASYM                                               WEYT2866
  562 CONTINUE                                                           WEYT2868
      IF  (FLAGB .NE. 0)  GO TO  600                                     WEYT2870
C                                                                        WEYT2872
      REWIND  NTAPCF                                                     WEYT2880
      GO TO  (580,570,580,580,570),            IFP                       WEYT2890
  570 WRITE  (NTAPCF)  POST                                              WEYT2900
      GO TO  590                                                         WEYT2910
  580 WRITE  (NTAPCF)  PRE                                               WEYT2920
  590 CONTINUE                                                           WEYT2930
      WRITE  (NTAPCF)  NP, NMSYM, NMASYM                                 WEYT2940
  600 CONTINUE                                                           WEYT2950
C                                                                        WEYT2960
      CALL        DCPB(NTDCP , NTAPW , NTAPDI, IGO, FLAGP, FLAGW, NROW,  WEYT2970
     1            NMODE, NMAX,         DCP   , COL ,        WORK      )  WEYT2980
      IF  (FLAGB .NE. 0)  GO TO  612                                     WEYT2982
```

```
C                                                                         WEYT2990
      CALL       DELC(NTSAIJ, NPOT   ,  NC ,  NP ,  NMODE,  NMAX ,        WEYT3000
     1           CIE ,  DCI ,  SAI   ,                       WORK       ) WEYT3010
C                                                                         WEYT3020
      DO  610  I = 1, NS                                                  WEYT3030
      ELIM(I)    =  (0.0, 0.0)                                            WEYT3040
  610 CONTINUE                                                            WEYT3050
  612 CONTINUE                                                            WEYT3052
      IF  (FLAGB .EQ. 0)  GO TO  616                                      WEYT3054
      GO TO  (672, 614),  FLAGB                                           WEYT3056
  614 IF  (FLAGP .NE. 4 .OR. NEM .EQ. 0)  GO TO  672                      WEYT3057
  616 CONTINUE                                                            WEYT3058
      CALL       SDBL(NUTL1,NUTL2,MASTSB,NTPHIJ,NTAPW,NTSAIJ,NTAPDI,      WEYT3060
     1    IGO, FLAGW, FLAGP, FLAGT, NC, NP, NS, NEM,  SQRTT ,   AIT,      WEYT3070
     2    DELA,  SBB, SBI, SAI, DI ,  W, PHI,          DELCPB, WORK    )  WEYT3080
      IF  (FLAGB .NE. 0)  GO TO  672                                      WEYT3090
C                                                                         WEYT3100
      CALL       GINV(NPOT, MASTSB, NC ,      NS ,  NX  , DCI          ,  WEYT3110
     1                EB  ,  CIE  , S   ,           SBMAT              )  WEYT3120
C                                                                         WEYT3130
  620 CONTINUE                                                            WEYT3140
      DO  630  I = 1, NP                                                  WEYT3150
      EB(I)      =  EB(I) / SQRTT(I)                                      WEYT3160
  630 CONTINUE                                                            WEYT3170
C                                                                         WEYT3180
      IF  (NELIMS.EQ. 0)  GO TO  670                                      WEYT3190
C                                                                         WEYT3210
      CALL       EDBL(NPOT, NELIMS,  NP , NS , LIMK, JARR, NSMOD       ,  WEYT3220
     1           EBMIN    , EBMAX ,  EB , ELIM                         )  WEYT3230
C                                                                         WEYT3240
      IF  (NSMOD .EQ. 0)  GO TO  650                                      WEYT3250
C                                                                         WEYT3260
      CALL       MODF( NC , NS , MASTSB, NEWTSB, JARR ,  SQRTT ,          WEYT3270
     1           ELIM, SBB, DCI          ,                 DCMOD       )  WEYT3280
C                                                                         WEYT3290
      CALL       GINV(NPOT, NEWTSB, NC,     NS    ,   NX     ,  DCMOD  ,  WEYT3300
     1                EB  ,  B    ,  S  ,            SBMAT             )  WEYT3310
C                                                                         WEYT3320
C                                                                         WEYT3330
      GO TO  620                                                          WEYT3340
C                                                                         WEYT3350
  650 CONTINUE                                                            WEYT3360
      DO  660  IB = 1, NS                                                 WEYT3370
  660 EB(IB)     =  EB(IB) + ELIM(IB)                                     WEYT3380
  670 CONTINUE                                                            WEYT3390
C                                                                         WEYT3400
      NTAPE      =  NTPHIJ                                                WEYT3402
      IF  (FLAGP .EQ. 4)  NTAPE = NUTL1                                   WEYT3404
      CALL       EPSJ(NTAPE ,  NP , NEM , NMAX,                           WEYT3410
     1           EB    ,           EPS ,                   PHI         )  WEYT3420
C                                                                         WEYT3430
  672 CONTINUE                                                            WEYT3432
C                                                                         WEYT3440
```

```
      CALL          DCPT(NPOT, LINES, IGO, FLAGB, FLAGP, NMODE, NP, NUTL1, WEYT3450
     1    NUTL2, NTAPDI, NTAPW, NTAPCF, X , Y , Z , GMA, DELA,   NMAX,   WEYT3460
     2    NEM   , W , DI, EPS,  DELCPB,    DCPTIL,  WORK    , EB       ) WEYT3470
C                                                                       WEYT3480
      IF  (NMON .EQ. 0)  GO TO  690                                     WEYT3500
C                                                                       WEYT3510
      CALL          CEMN(NPOT,  IGO, NMODE, NTSANJ, NP, NMON, LABEL, NUTL2,WEYT3520
     1              SAI      ,  DCPTIL,     CE                          ) WEYT3530
C                                                                       WEYT3540
  690 CONTINUE                                                          WEYT3550
      IF  (IGO .EQ. 2 .OR. NMASYM .EQ. 0)  GO TO  700                   WEYT3560
      IGO      =   2                                                    WEYT3570
      NMODE    =   NMASYM                                               WEYT3580
      GO TO  600                                                        WEYT3590
  700 CONTINUE                                                          WEYT3600
C                                                                       WEYT3610
      RETURN                                                            WEYT3620
      END                                                               WEYT3630
      SUBROUTINE WSWA(NPIT, NPOT, NUTL1, NTAPW, KODE, NP, NCOL, NMAX,   WSWA0040
     1              NMSYM,   NMASYM,                      W            ) WSWA0050
C                                                                       WSWA0060
C         NPIT       DATA SET NUMBER OF THE SYSTEM INPUT  DATA SET      WSWA0070
C         NPOT       DATA SET NUMBER OF THE SYSTEM OUTPUT DATA SET      WSWA0080
C         NUTL1      UTILITY (SCRATH) TAPE NUMBER                       WSWA0090
C         NTAPW      TAPE NUMBER CONTAINING COLUMNS OF THE  W  MATRIX   WSWA0100
C         KODE       = -1                                               WSWA0110
C         NP         NUMBER OF ROW ELEMENTS IN THE  W  MATRIX           WSWA0120
C         NCOL       NUMBER OF COLUMNS      IN THE  W  MATRIX           WSWA0130
C         NMAX       MAXIMUM NUMBER OF COLUMNS IN THE  W  MATRIX        WSWA0140
C         NMSYM      NUMBER OF SYMMETRIC MODES                          WSWA0150
C         NMASYM     NUMBER OF ANTISYMMETRIC MODES                      WSWA0160
C         W          2-D COMPLEX ARRAY CONTAINING THE  W  MATRIX        WSWA0170
C                                                                       WSWA0180
      COMPLEX      WIN(100),    W( NP , NMAX)                           WSWA0190
      DIMENSION    MODE(100), IDELW(100), LIMW(2, 100)                  WSWA0200
   10 FORMAT   ( 4I10, 4F10.0)                                          WSWA0220
   20 FORMAT   ( 1H1/// 23H   --W--   IS CARD INPUT // 26H   MODE    DELTA WSWA0230
     1    LIMITS, 15X, 5H- W - / 35X, 18HREAL         IMAG.      )      WSWA0240
   30 FORMAT   ( I6, 2I8, I6, 4F14.6 )                                  WSWA0250
   40 FORMAT   ( 1H1///6H   THE , I4, 41H COLUMNS OF THE     SYMMETRIC  WWSWA0260
     1  MATRIX                   )                                      WSWA0270
   50 FORMAT   ( /  8H  COLUMN , I4 /  ( 3 ( I6, 2E14.6 ) ) )           WSWA0280
   60 FORMAT   ( 1H1///6H   THE , I4, 41H COLUMNS OF THE ANTISYMMETRIC  WWSWA0290
     1  MATRIX                   )                                      WSWA0300
   62 FORMAT   ( 1H1 // )                                               WSWA0302
C                                                                       WSWA0310
      J            =  1                                                 WSWA0320
      WRITE   (NPOT,20)                                                 WSWA0330
   70 CONTINUE                                                          WSWA0340
      READ    (NPIT,10)  MODE(J), IDELW(J), LIMW(1,J),LIMW(2,J), WIN(J) WSWA0350
      IF   (MODE(J) .LE. KODE)  GO TO  80                               WSWA0360
      WRITE   (NPOT,30)  MODE(J), IDELW(J), LIMW(1,J),LIMW(2,J), WIN(J) WSWA0370
      J            =  J + 1                                             WSWA0380
```

```
      GO TO  70
   80 CONTINUE
      NROW       =   NP
      REWIND  NUTL1
      DO   100   N = 1, NMAX
      DO   90    M = 1, NROW
      W(M, N)   =   (0.0, 0.0)
   90 CONTINUE
  100 CONTINUE
      JMAX       =   J - 1
      NDEL       =   1
      IGO        =   1
  110 CONTINUE
      MDMAX      =   0
      DO   130   J = 1, JMAX
      ND         =   IDELW(J)
      IF   (ND .NE. NDEL)  GO TO  130
      MD         =   MODE(J)
      L1         =   LIMW(1, J)
      L2         =   LIMW(2, J)
      IF   (MDMAX .LT. MD)  MDMAX = MD
      DO   120   L = L1, L2
      W(L, MD)   =   WIN(J)
  120 CONTINUE
C
  130 CONTINUE
C
      WRITE   (NUTL1)   W
      GO TO   (140,150),   IGO
  140 CONTINUE
C
      NMSYM      =   MDMAX
      NDEL       =   -1
      IGO        =   2
      GO TO  110
  150 CONTINUE
      NMASYM     =   MDMAX
      REWIND  NUTL1
      REWIND  NTAPW
      WRITE   (NTAPW)  NROW, NMSYM, NMASYM
C
      NMD        =   NMSYM
      IGO        =   1
      WRITE   (NPOT,40)  NMSYM
  160 CONTINUE
      READ    (NUTL1)  W
      DO   170   NM = 1, NMD
C
      WRITE   (NTAPW)  (W(M, NM),   M = 1, NROW)
      WRITE   (NPOT,50)  NM, ( I, W( I, NM),   I = 1, NROW )
      IF   (NM .NE. NMD)  WRITE   (NPOT,62)
  170 CONTINUE
C
```

```
      GO TO  (180,190),   IGO                                         WSWA0920
  180 CONTINUE                                                        WSWA0930
      IF  (NMASYM .EQ. 0)  GO TO  190                                 WSWA0940
      WRITE  (NPOT,60)  NMASYM                                        WSWA0950
      NMD       =  NMASYM                                             WSWA0960
      IGO       =  2                                                  WSWA0970
      GO TO  160                                                      WSWA0980
C                                                                     WSWA0990
  190 CONTINUE                                                        WSWA1000
      REWIND  NTAPW                                                   WSWA1010
      RETURN                                                          WSWA1020
      END                                                             WSWA1030
      SUBROUTINE ZEROUT( WORK, LENGTH, LOOP, ITAPE )                  ZER00020
C                                                                     ZER00030
C         WORK       COMPLEX ARRAY TO BE INITIALIZED TO  ZEROES       ZER00040
C         LENGTH     LENGTH OF ARRAY  WORK                            ZER00050
C         LOOP       NUMBER OF TIMES THE ARRAY WORK IS TO BE WRITTEN ON ZER00060
C                    TAPE  ITAPE  (ONLY IF  ITAPE  IS NOT  ZERO)      ZER00070
C         ITAPE      TAPE NUMBER ON WHICH THE ARRAY  WORK IS SAVED    ZER00080
      COMPLEX    WORK(LENGTH)                                         ZER00090
C                                                                     ZER00100
      DO  10  I = 1, LENGTH                                           ZER00110
      WORK(I)  =  (0.0, 0.0)                                          ZER00120
   10 CONTINUE                                                        ZER00130
      IF  (ITAPE . EQ . 0)  RETURN                                    ZER00140
C                                                                     ZER00150
      DO  20  L = 1, LOOP                                             ZER00160
      WRITE  (ITAPE)  (WORK(I),  I = 1, LENGTH)                       ZER00170
   20 CONTINUE                                                        ZER00180
      RETURN                                                          ZER00190
      END                                                             ZER00200
```

REFERENCES

1. Revell, J. D., and Rodden, W. P., "A Rational Method for Utilizing Experimental Aerodynamic Data in Flutter Analyses Through the Use of Aerodynamic Influence Coefficient Matrices", North American Aviation, Inc., Report NA-59-867, 30 January 1959.

2. Rodden, W. P., "An Empirical Weighting Matrix for Use with Aerodynamic Influence Coefficients in Aeroelastic Analyses," Northrop Corp., Report NOR-59-320, 1 April 1959.

3. Rodden, W. P., and Revell, J. D., "The Status of Unsteady Aerodynamic Influence Coefficients," Paper FF-33, presented to IAS 30th Annual Meeting, 20-24 January 1962; preprinted as Rept. TDR-930(2230-09)TN-2, 22 November 1961, Aerospace Corp., El Segundo, Ca.

4. Rodden, W. P., "Comment on 'Convergence Proof of Discrete-Panel Wing Loading Theories'", J. Aircraft, Vol. 9, No. 9, September 1972, pp. 686-688.

5. Bergh, H., "Some Aspects of Unsteady Pressure Measurements," AGARD Report 498 Part II, 1965; also Rept. NLR MP.227, National Aerospace Lab., Amsterdam, The Netherlands, 1964.

6. Bergh, H., and Zwaan, R. J., "A Method for Estimating Unsteady Pressure Distributions for Arbitrary Vibration Modes from Measured Distributions for One Single Mode," Rept. NLR-TR-F.250, National Aerospace Lab., Amsterdam, The Netherlands, February 1966.

7. Ashley, H., "Some Considerations Relative to the Prediction of Unsteady Air Loads on Interfering Surfaces," AGARD-CP-80-71, Paper No. 1, AGARD Symposium on Unsteady Aerodynamics for Aeroelastic Analyses of Interfering Surfaces, Tønsberg, Norway, 3-4 November 1970.

8. Ashley, H., and Rowe, W. S., "On the Unsteady Aerodynamic Loading of Wings with Control Surfaces," Z. Flugwiss, Band 18, Heft 9/10, September/Oktober

1970, pp. 321-330.

9. Rowe, W. S., Winther, B. A., and Redman, M. C., "Unsteady Subsonic Aerodynamic Loadings Caused by Control Surface Motions," J. Aircr., Vol. 11, No. 1, January 1974, pp. 45-54.

10. Tijdeman, H., and Zwaan, R. J., "On the Prediction of Aerodynamic Loads on Oscillating Wings in Transonic Flow," AGARD Report No. 612, Supplement to the Manual on Aeroelasticity, Vol. 11, Ch. 10, January 1974. Also Report NLR MP 73026U June 1965.

11. Tijdeman, H., and Bergh, H., "Analysis of Pressure Distributions Measured on a Wing with Oscillating Control Surface in Two-Dimensional High Subsonic and Transonic Flow," National Aerospace Laboratories Report NLR-TR F.253, March 1967.

12. Traci, R. M., Farr, J. L., Albano, E., "Perturbation Method for Transonic Flows About Oscillating Airfoils," AIAA Paper No. 75-877, June 1975.

13. Tijdeman, H., Schippers, P., "Results of Pressure Measurements on an Airfoil with Oscillating Flap in Two-Dimensional High Subsonic and Transonic Flow (Zero Incidence and Zero Mean Flap Position)," National Aerospace Laboratories Report NLR TR 73078U, July 1973.

14. Hertrich, H., "Druckverteilingsmessungen an Halbflügelmodellen mit Ruder in stationärer Unterschallströmung", Bericht 66J12, Aerodynamische Versuchsanstalt, Göttingen, 23 December 1966.

15. Hertrich, H., "Zur experimentellen Prüfung instationärer dreidimensionaler Tragflachentheorien bei inkompressibler Strömung," Bericht 67J02, Aerodynamische Versuchsanstalt, Göttingen, 25 Mar. 1967.

16. Försching, H., Triebstein, H., and Wagener, J., "Pressure Measurements on an Harmonically Oscillating Swept Wing with Two Control Surfaces in Incompressible Flow," AGARD-CP-80-71, Paper No. 15, AGARD Symposium on

Unsteady Aerodynamics for Aeroelastic Analyses of Interfering Surfaces, Tønsberg, Norway, 3-4 November 1970.

17. Giesing, J. P., Kalman, T. P., Rodden, W. P., "Subsonic Steady and Oscillatory Aerodynamics for Multiple Interfering Wings and Bodies," Journal of Aircraft, Vol. 9, No. 10, Oct. 1972.

18. Schmeer, J. W., "The Effect of Leading-Edge Droop upon the Pressure Distribution and Aerodynamic Loading Characteristics of a 45° Sweptback Wing at Transonic Speeds," NACA TM RM L55.I16, Nov. 1955.

19. Carlson, H. W., "Pressure Distributions at Mach Number 2.05 on a Series of Highly Swept Arrow Wings Employing Various Degrees of Twist and Camber," NASA TN D-1264, May 1962.

20. Giesing, J. P., Kalman, T. P., "Oscillatory Supersonic Lifting Surface Theory Using a Finite Element Doublet Representation," AIAA Paper No. 75-761 presented Denver, Colo. May 1975.

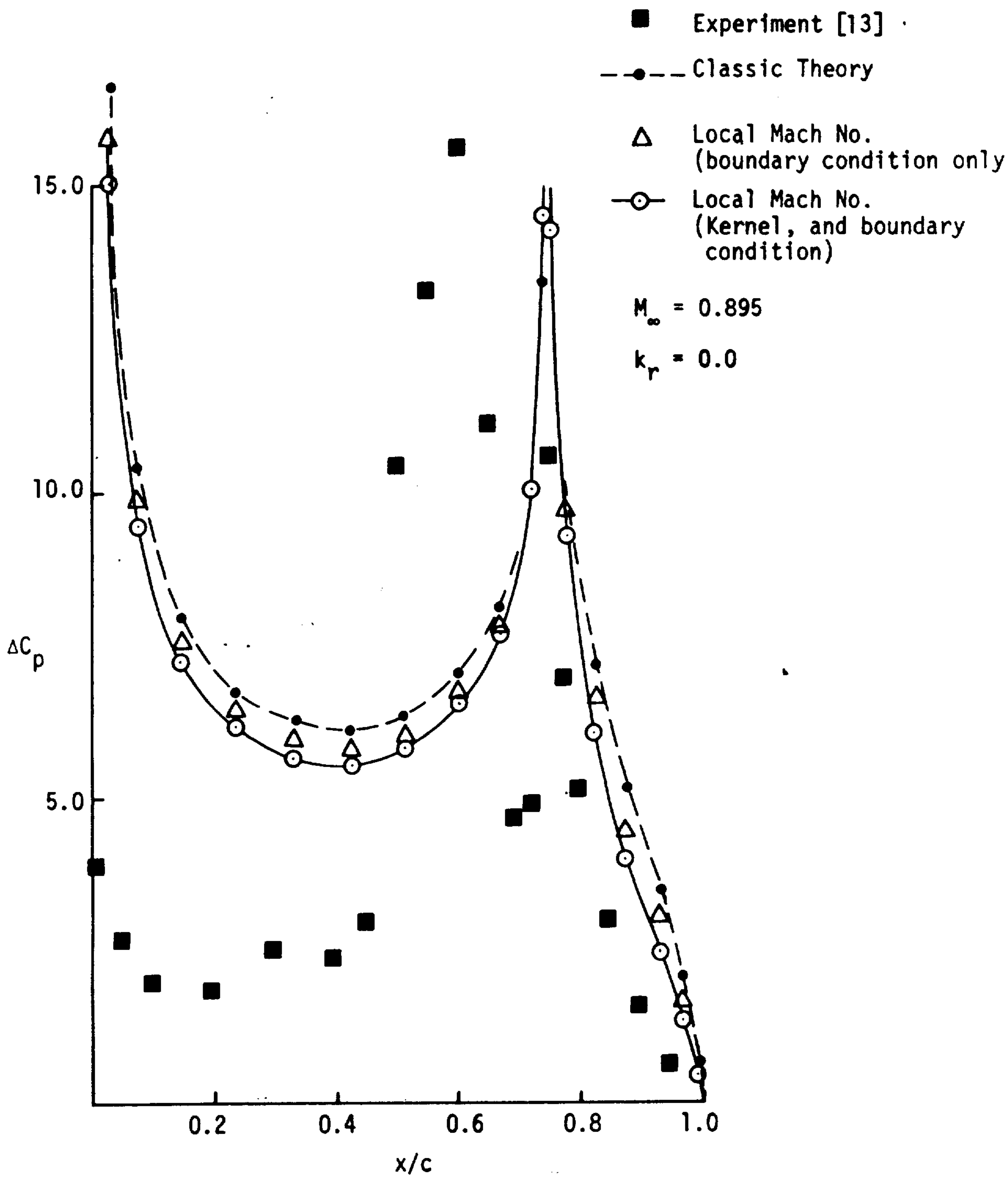


Figure 3. - Effect of applying local Mach Number to the boundary conditions and the kernel of the classic theory.

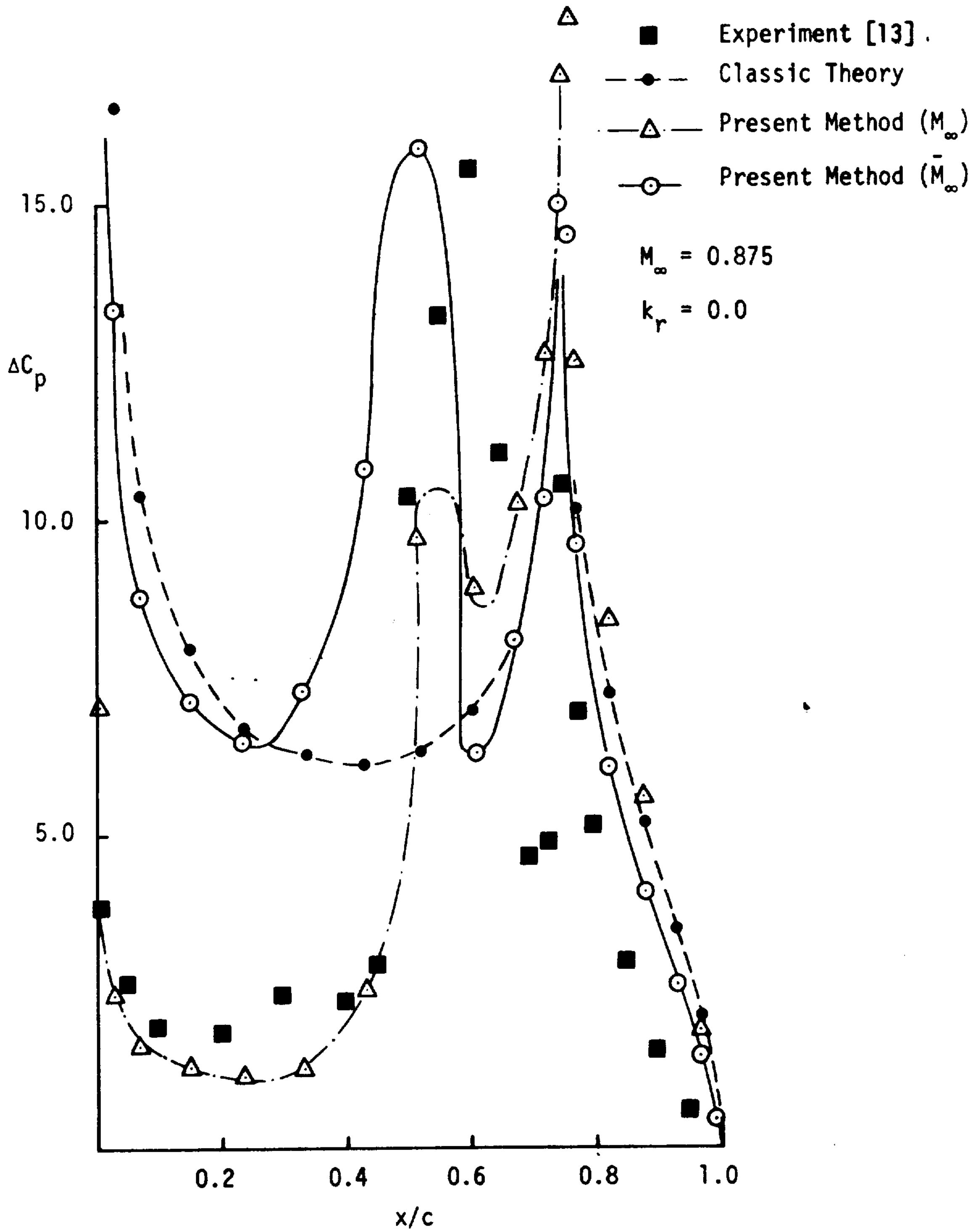


Figure 4. - Comparison of classic theory and two variations of the present method with experimental data for the steady case.

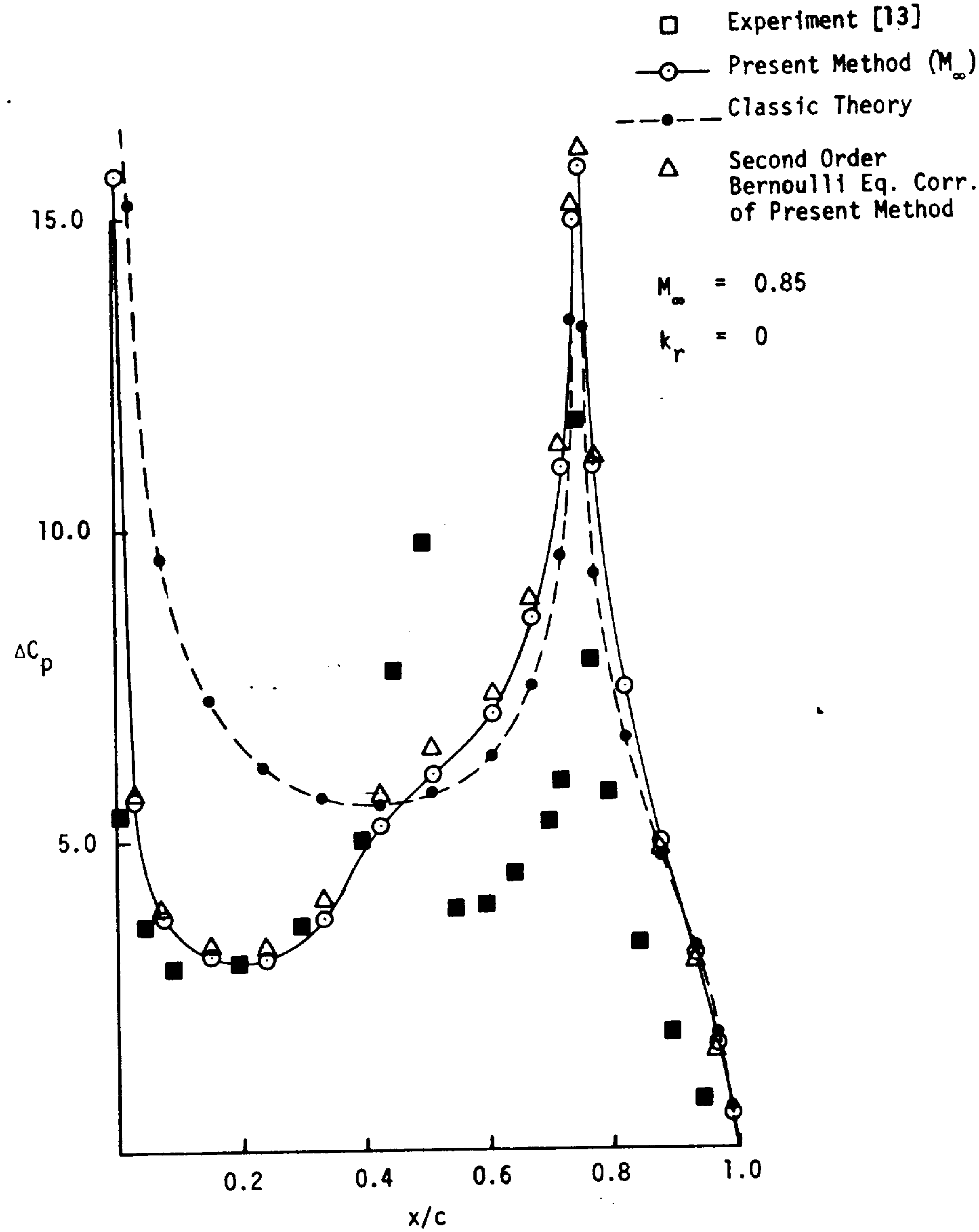


Figure 5. - Comparison of classic theory and the present method (M_∞) (with and without second order Bernoulli correction) with experimental data.

5

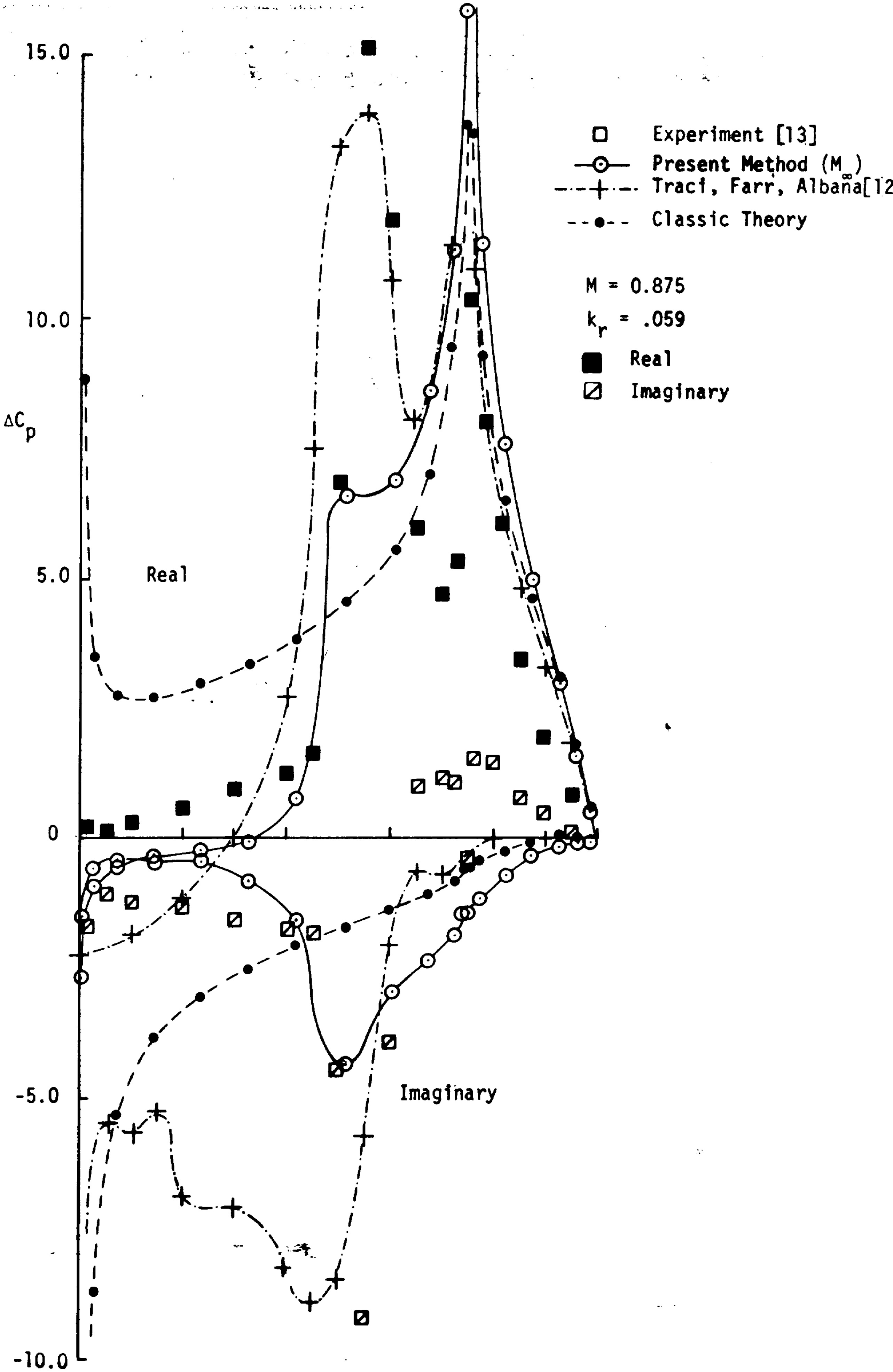


Figure 6. - Comparison of various theories with experimental data for the oscillatory control surface case.

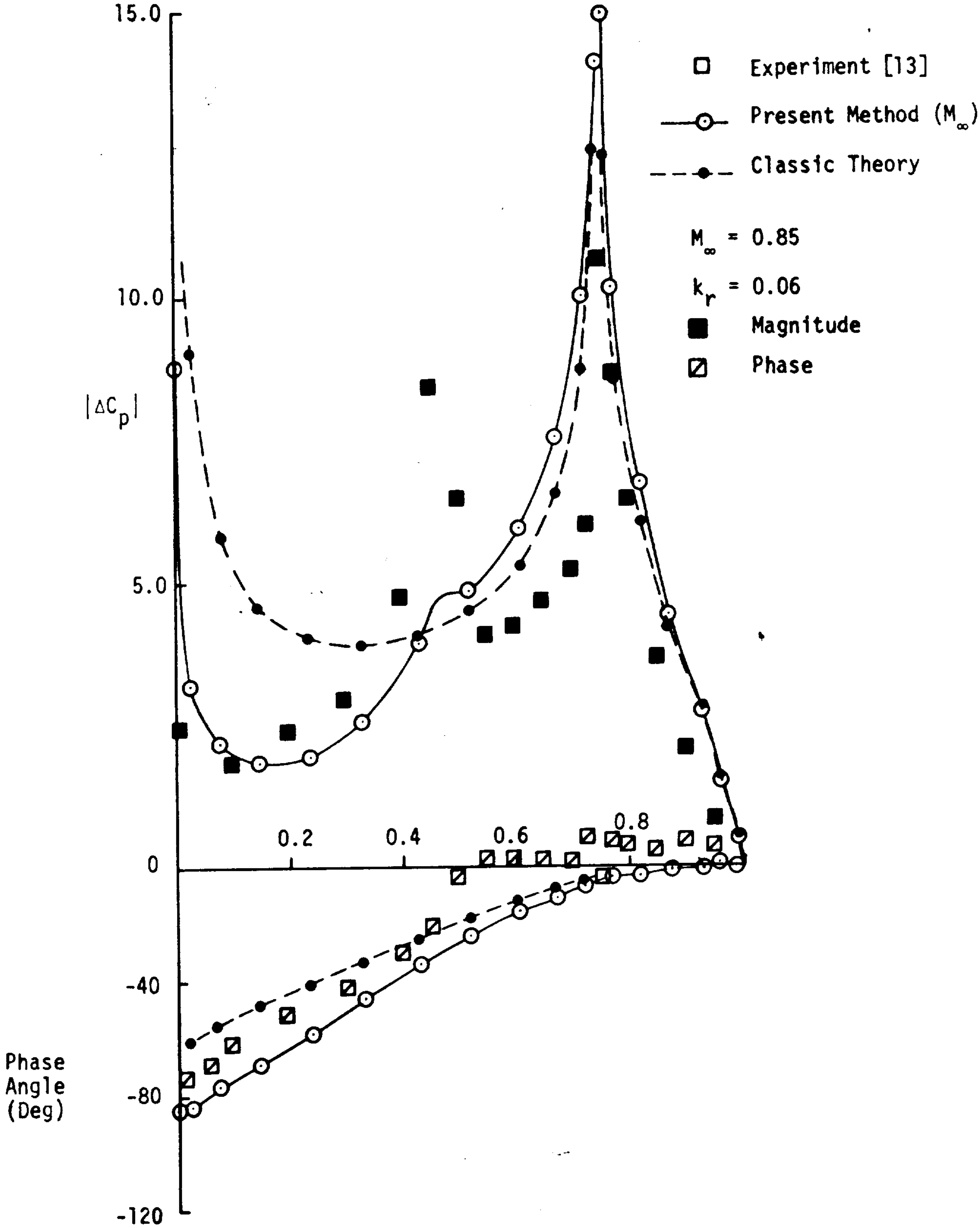


Figure 7. - Comparison of classic theory and the present method (M_∞) with experimental data for the oscillatory case.

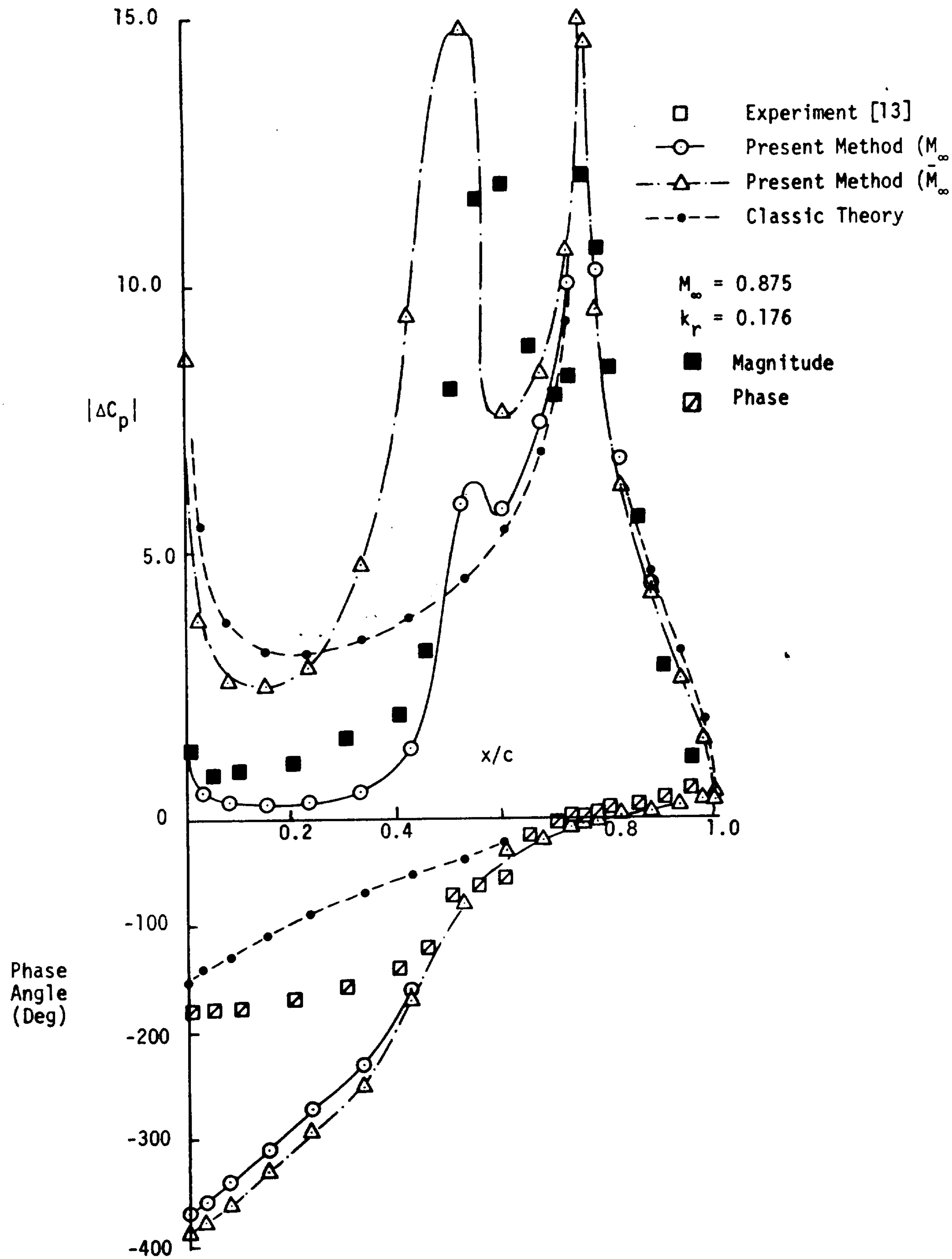


Figure 8. - Comparison of two variations of the Present transonic method with data and classic theory for the oscillatory case.

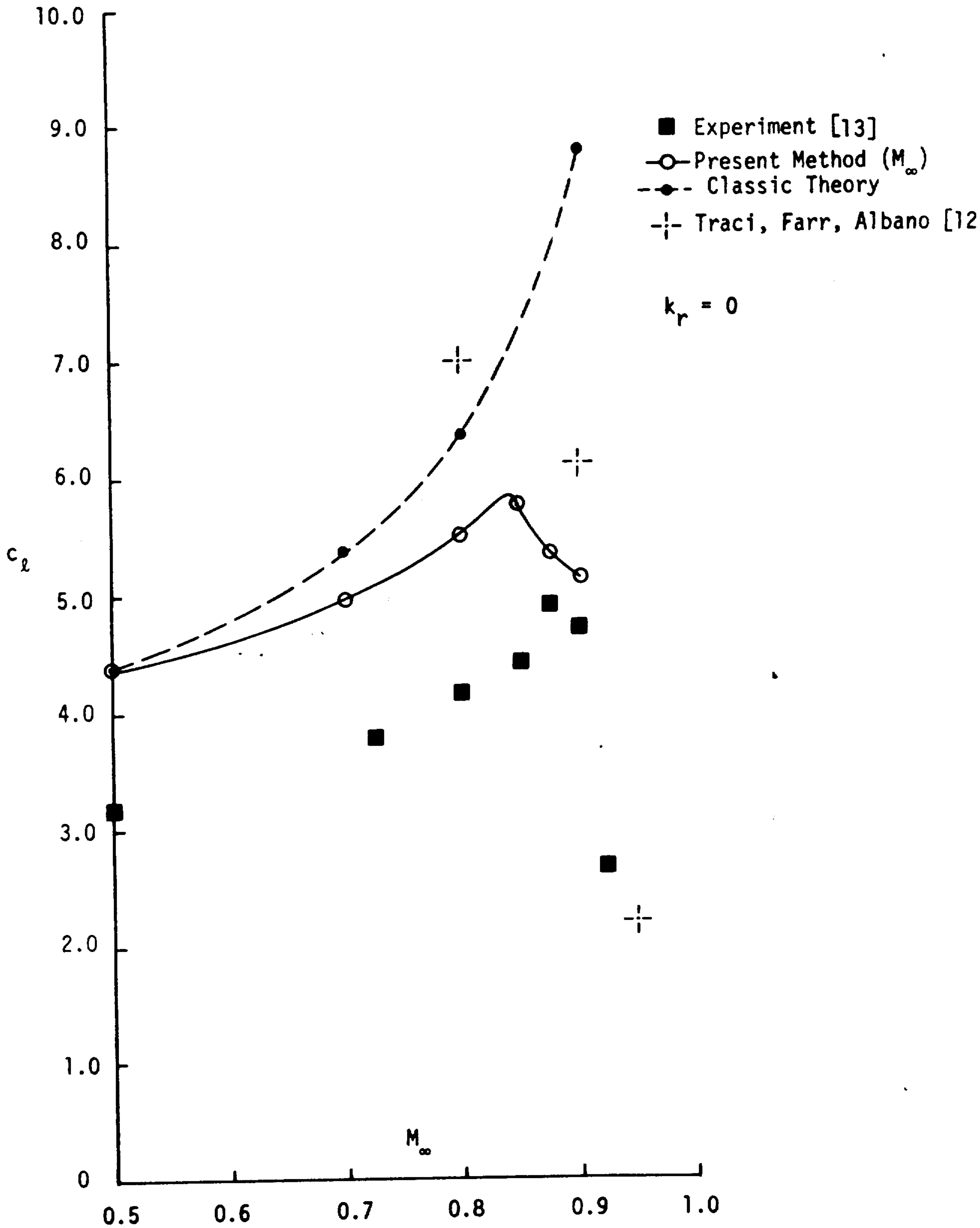


Figure 9. - Comparison of experimental and theoretical lift coefficient for an airfoil with a deflected 25% chord flap.

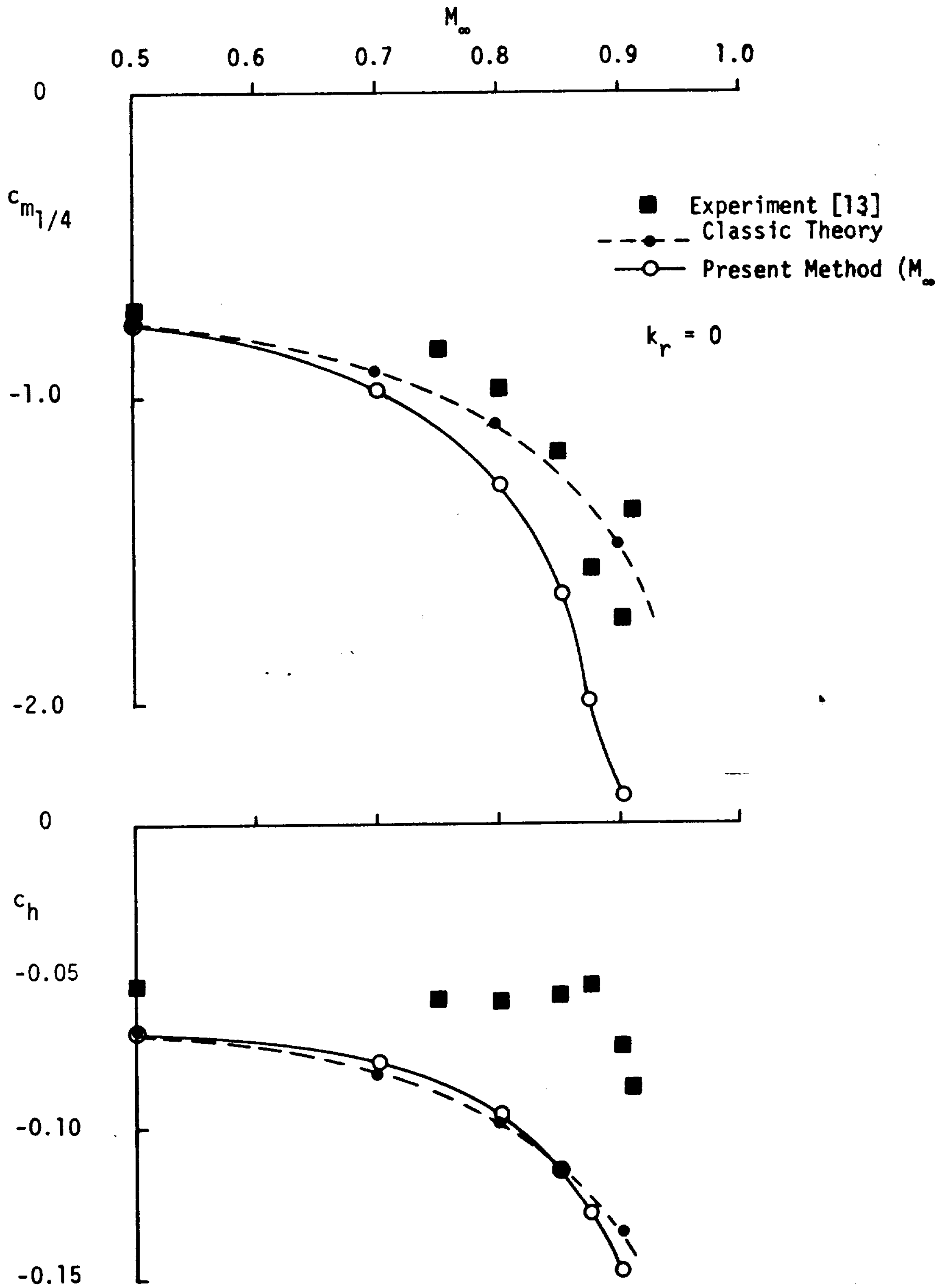


Figure 10 - Comparison of experimental and theoretical pitching (about c/4) and hinge moment (about 3 c/4) coefficient for an airfoil with a deflected 25% chord flap.

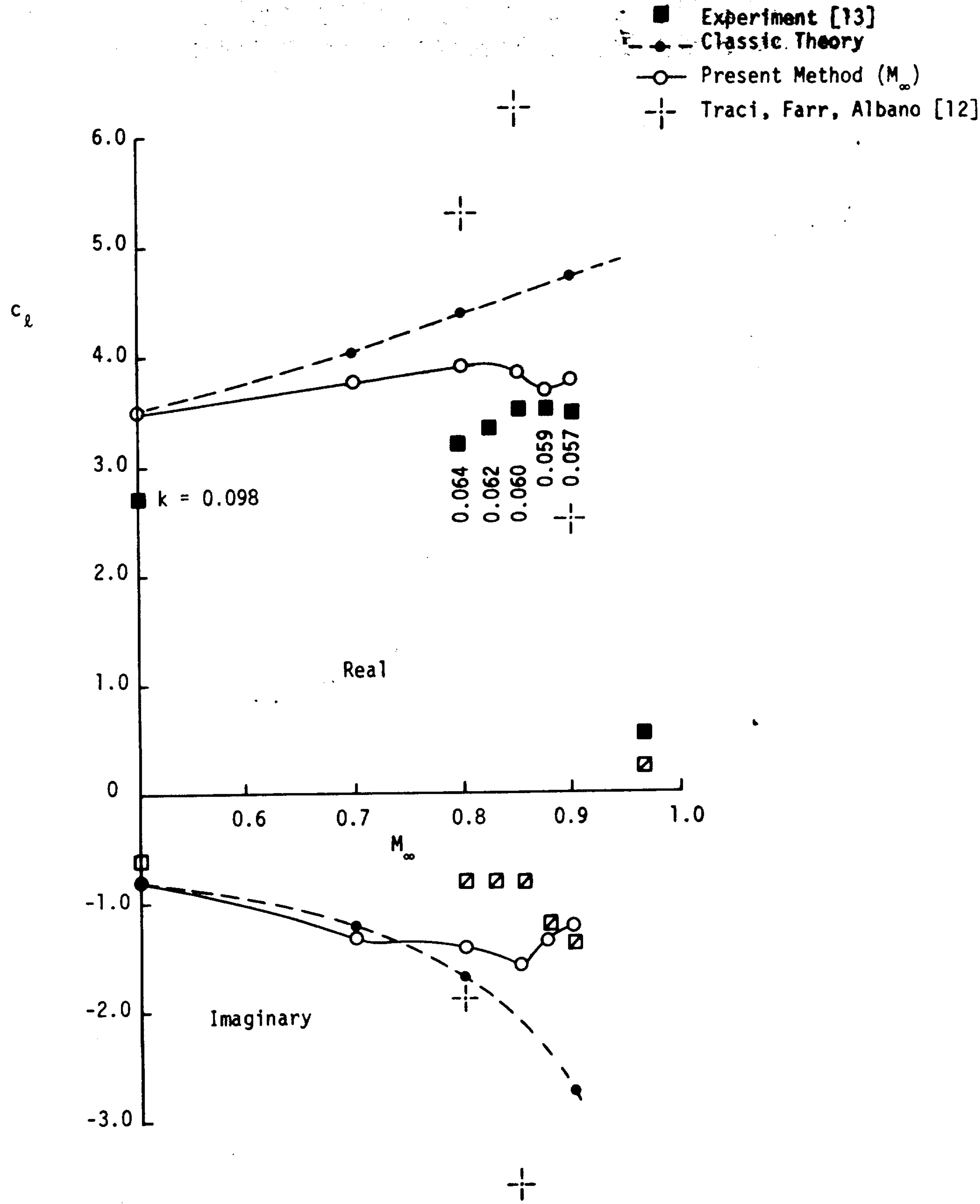


Figure 11. - Comparison of experimental and theoretical lift coefficient for an airfoil with deflected 25% chord flap.

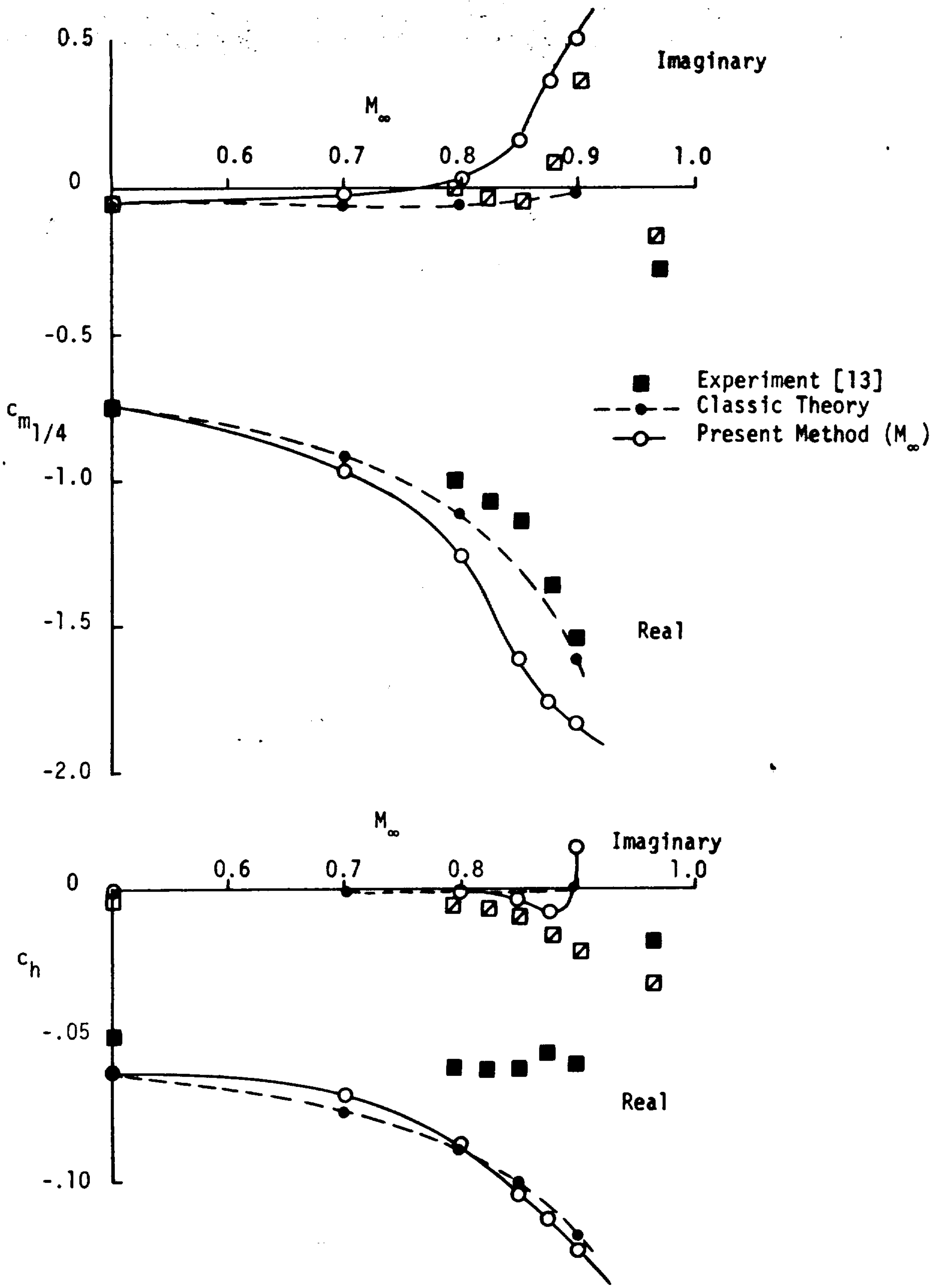


Figure 12. - Comparison of experimental and theoretical pitching moment (about c/4) and hinge moment (about 3c/4) coefficient for an airfoil with an oscillating control surface.

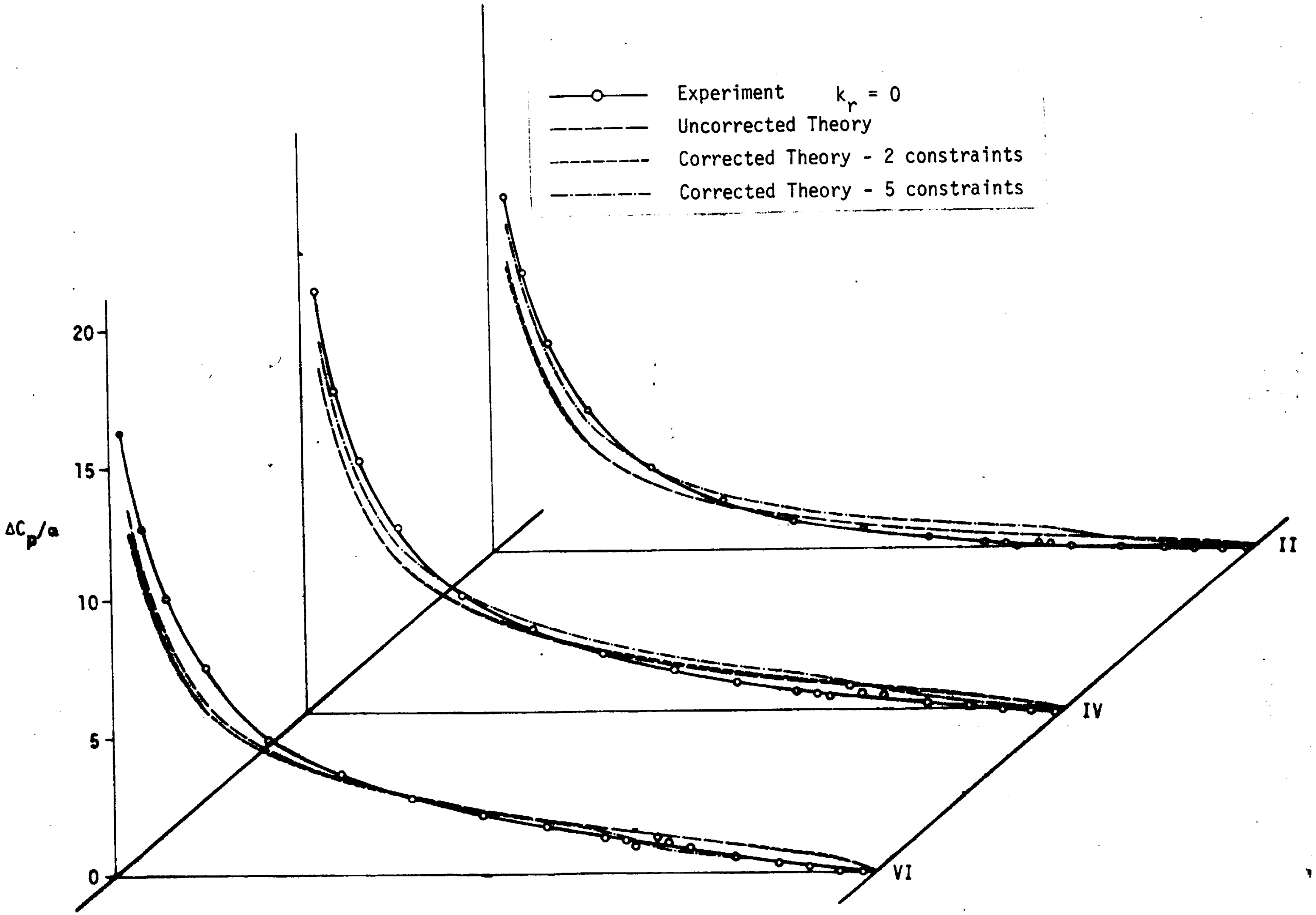


Figure 13. - Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Premultiplying Correction Factors

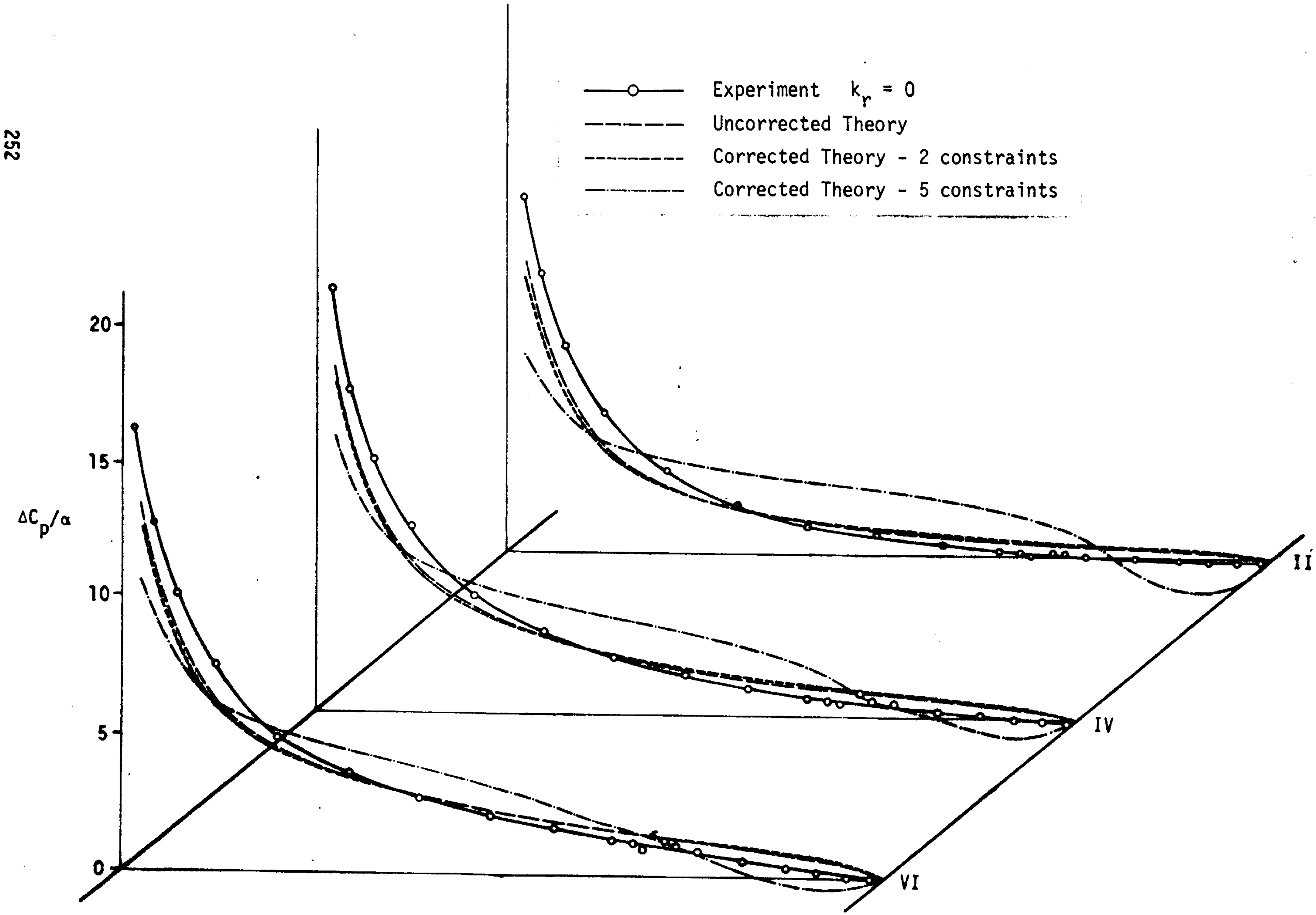


Figure 14. - Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Postmultiplying Correction Factors

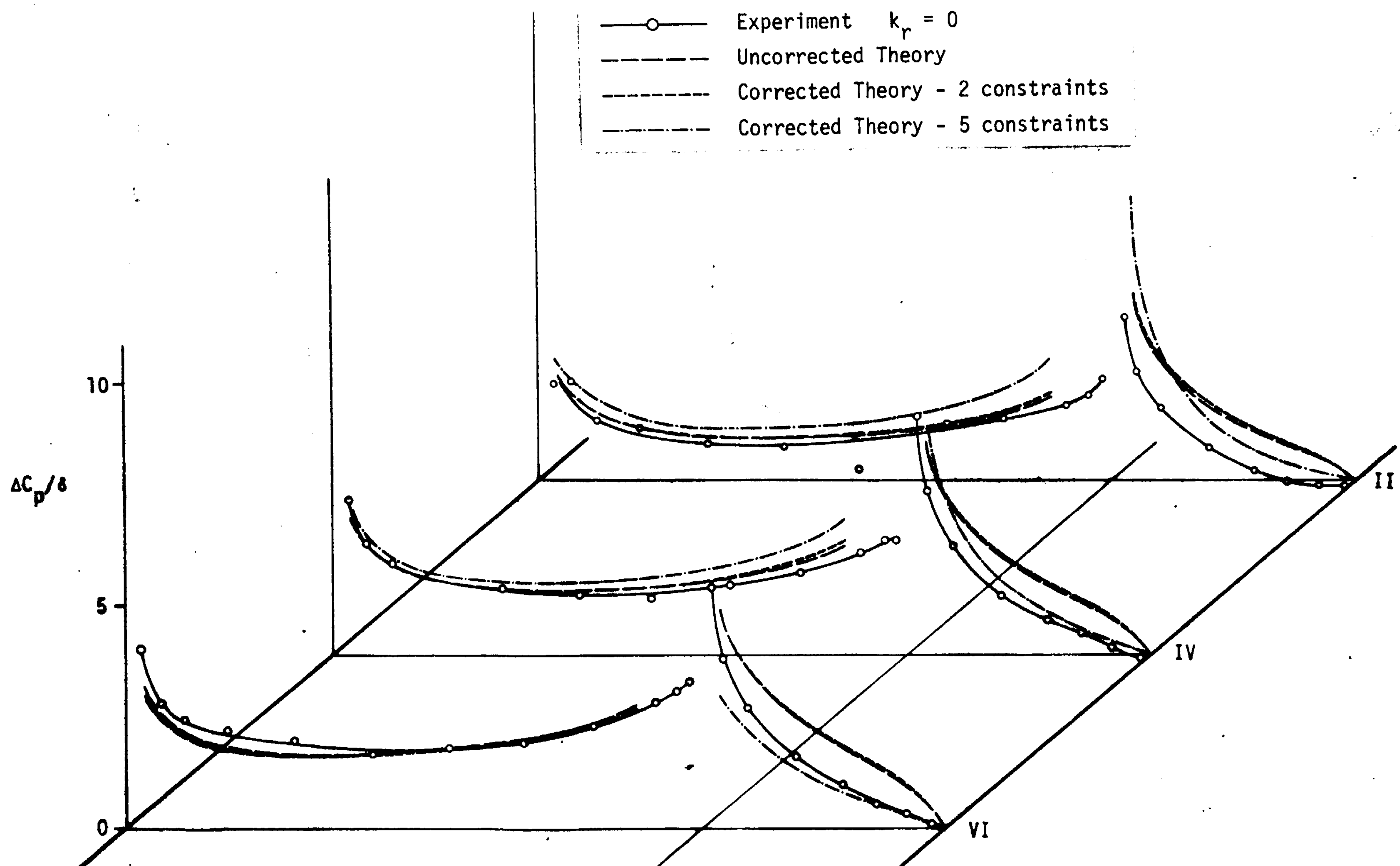


Figure 15. - Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Premultiplying Correction Factors

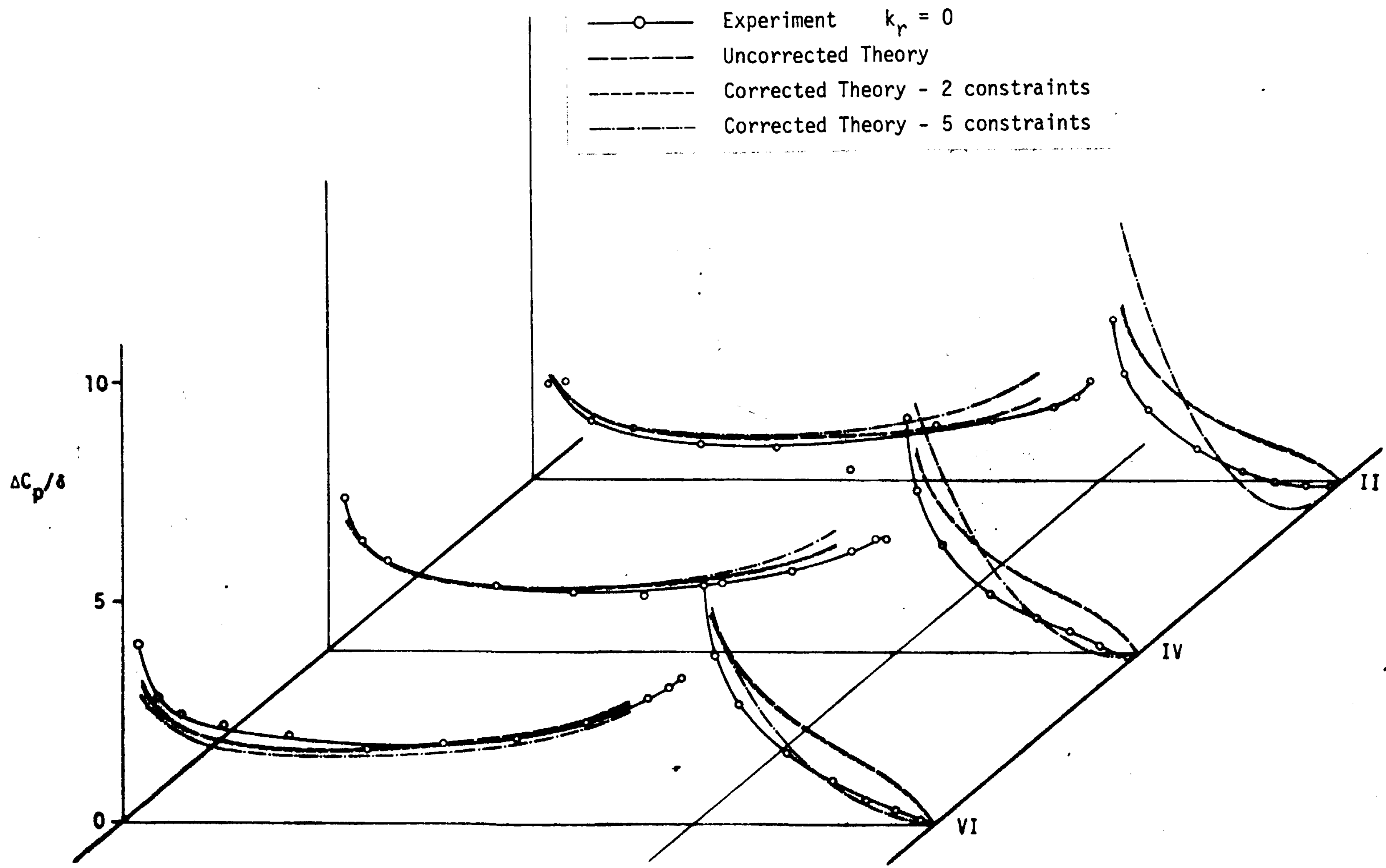


Figure 16. - Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Postmultiplying Correction Factors

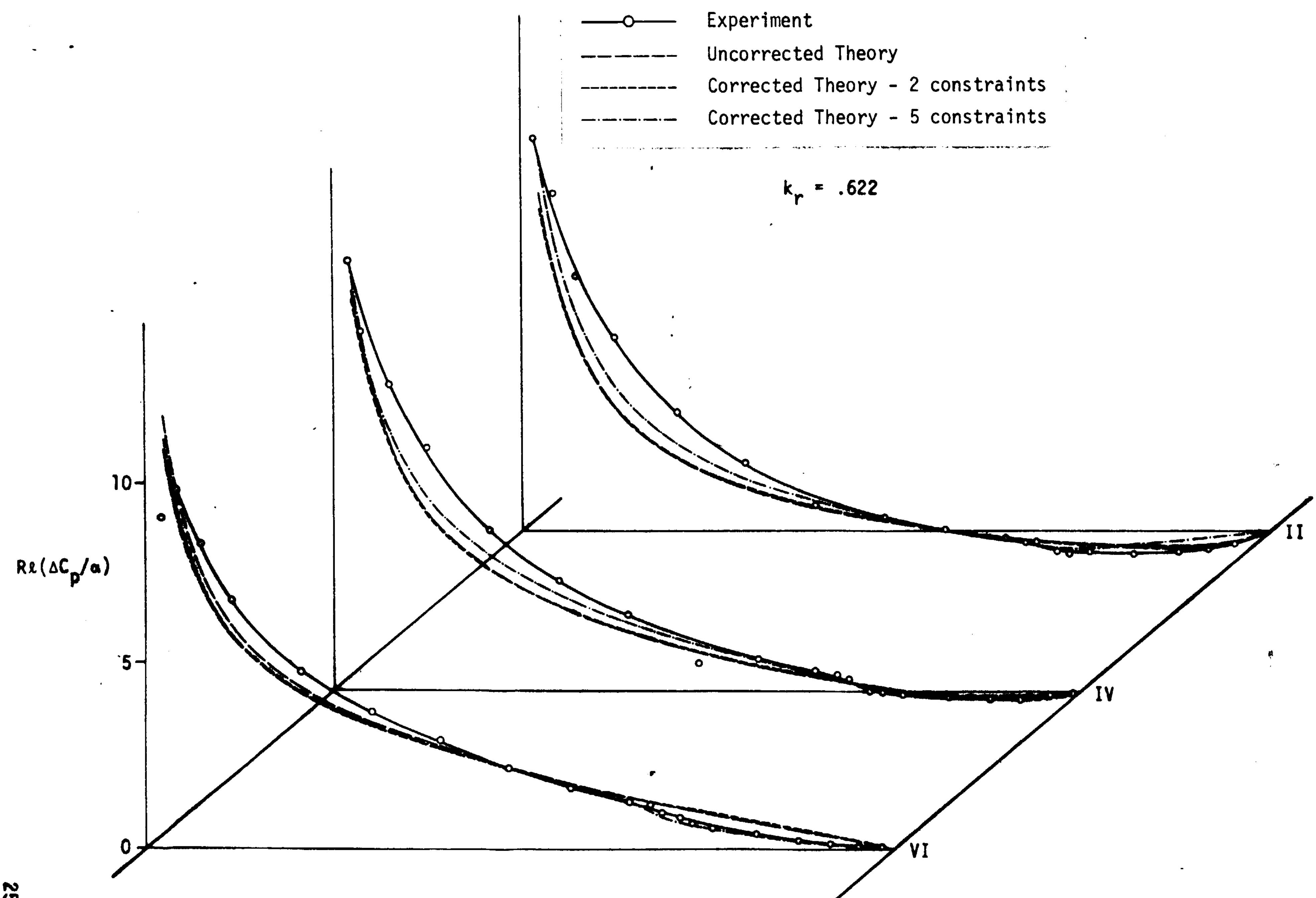


Figure 17. - Real Part of Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Premultiplying Correction Factors

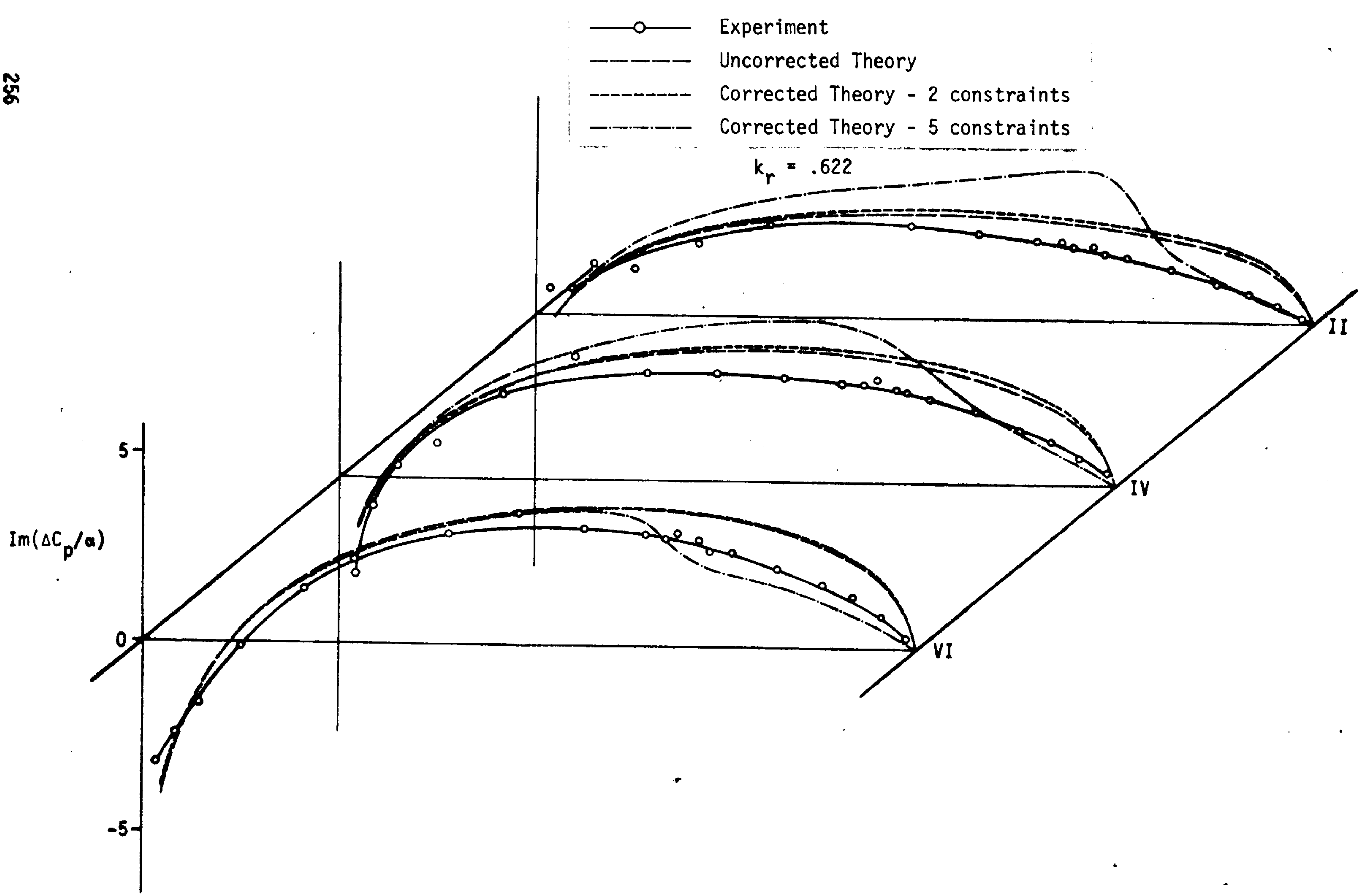


Figure 18. - Imaginary Part of Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Premultiplying Correction Factors

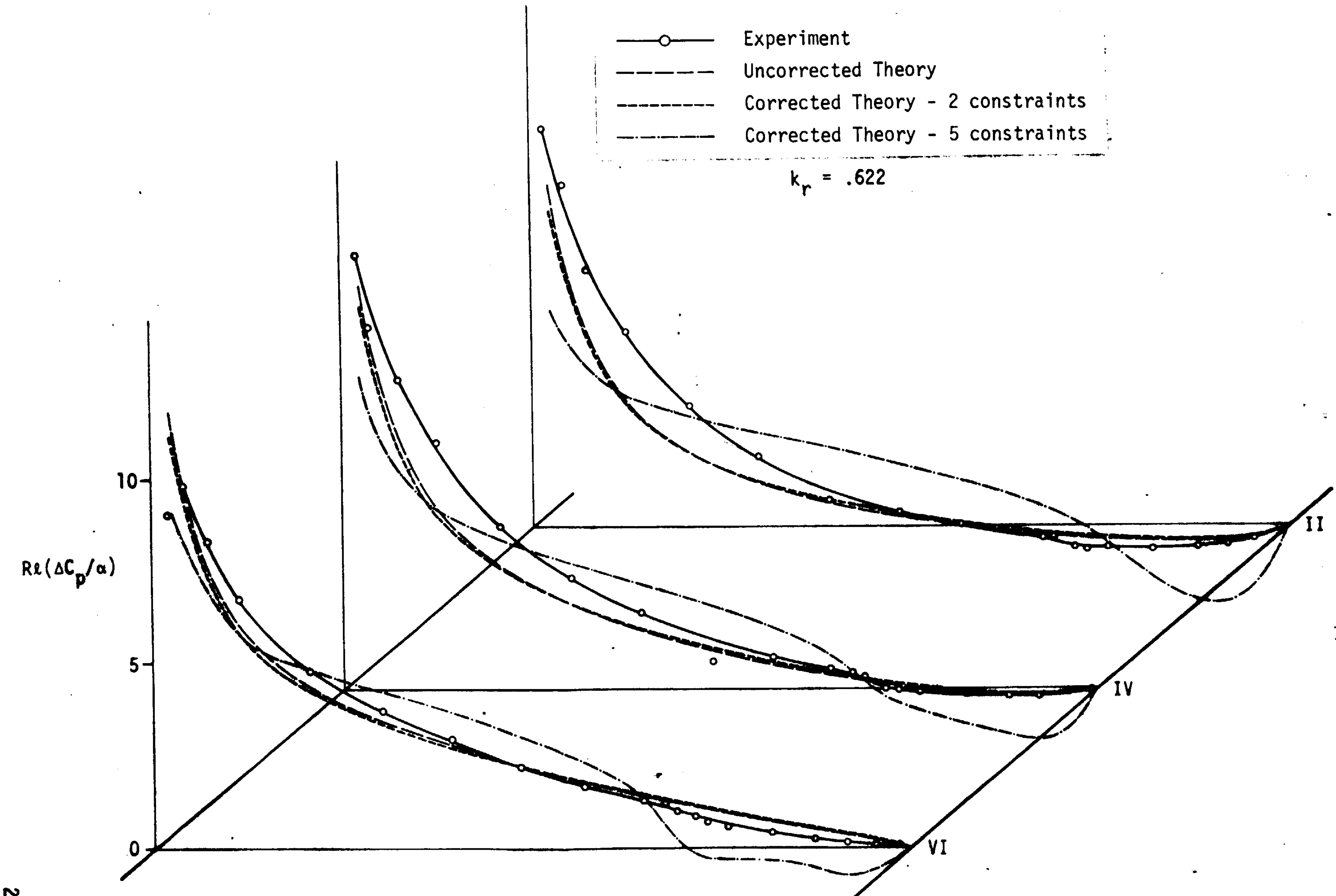


Figure 19. - Real Part of Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Postmultiplying Correction Factors

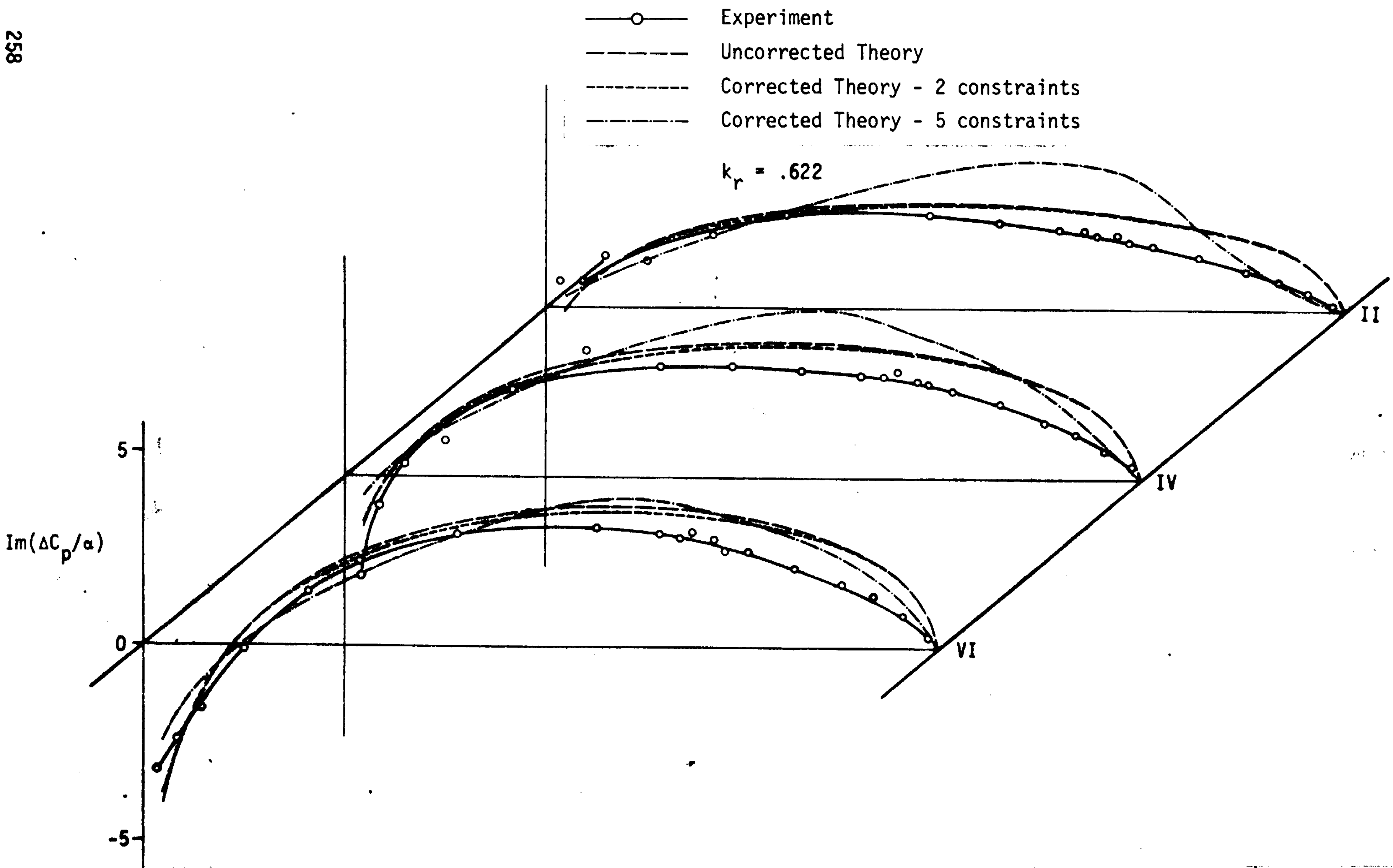


Figure 20. - Imaginary Part of Experimental, Theoretical and Corrected Pressure Loadings for Angle of Attack Using Postmultiplying Correction Factors

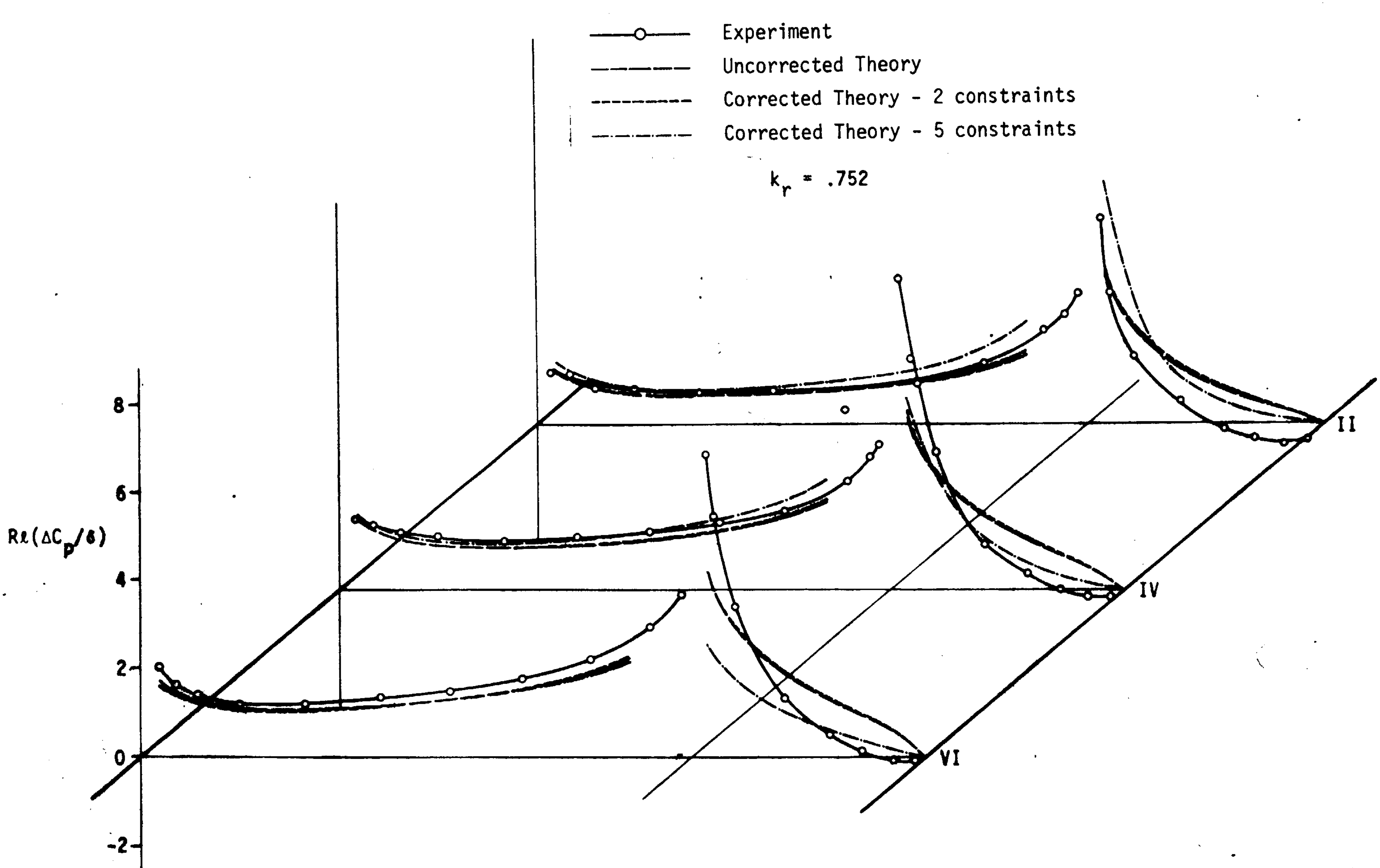


Figure 21. - Real Part of Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Premultiplying Correction Factors

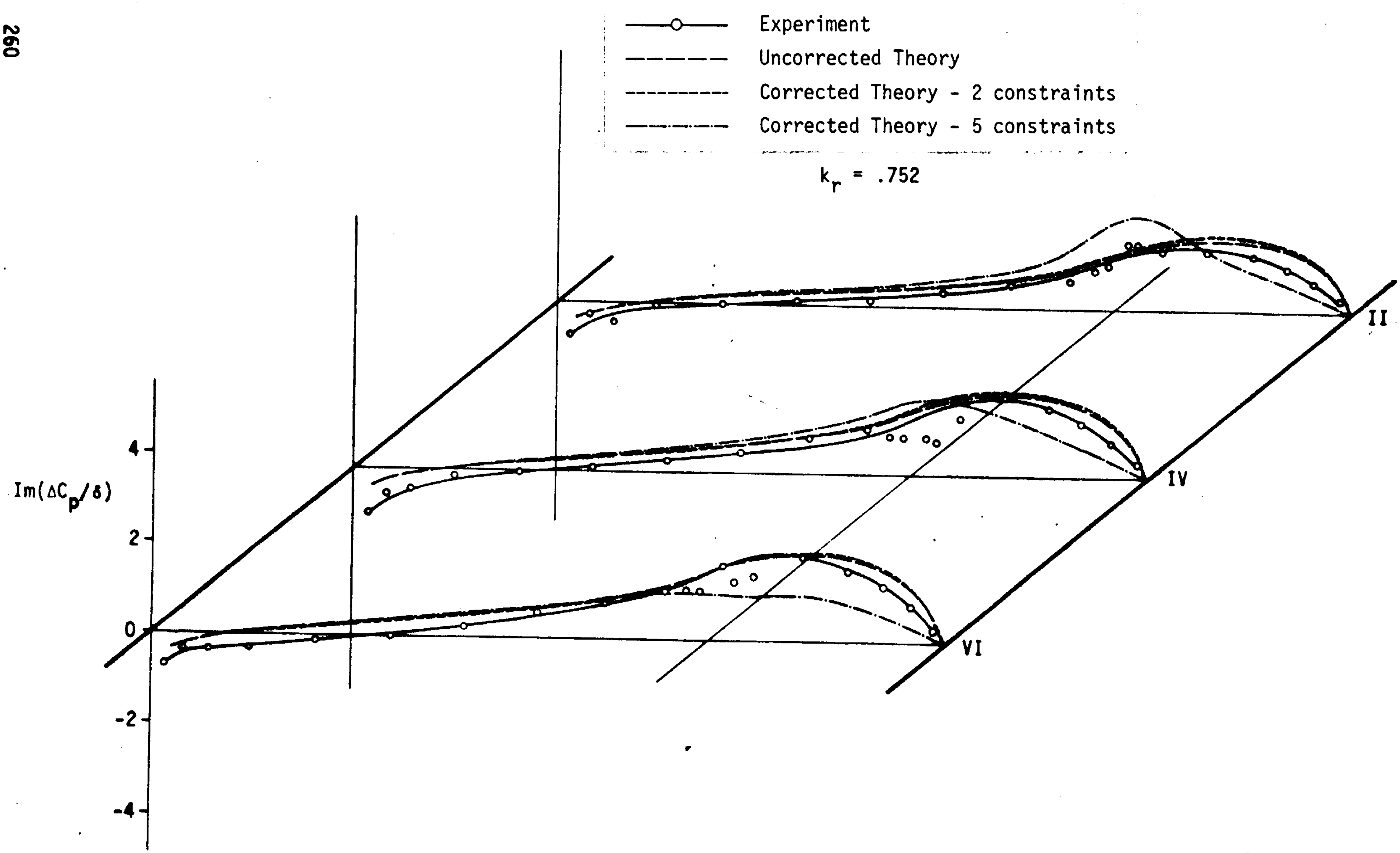


Figure 22. - Imaginary Part of Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Premultiplying Correction Factors

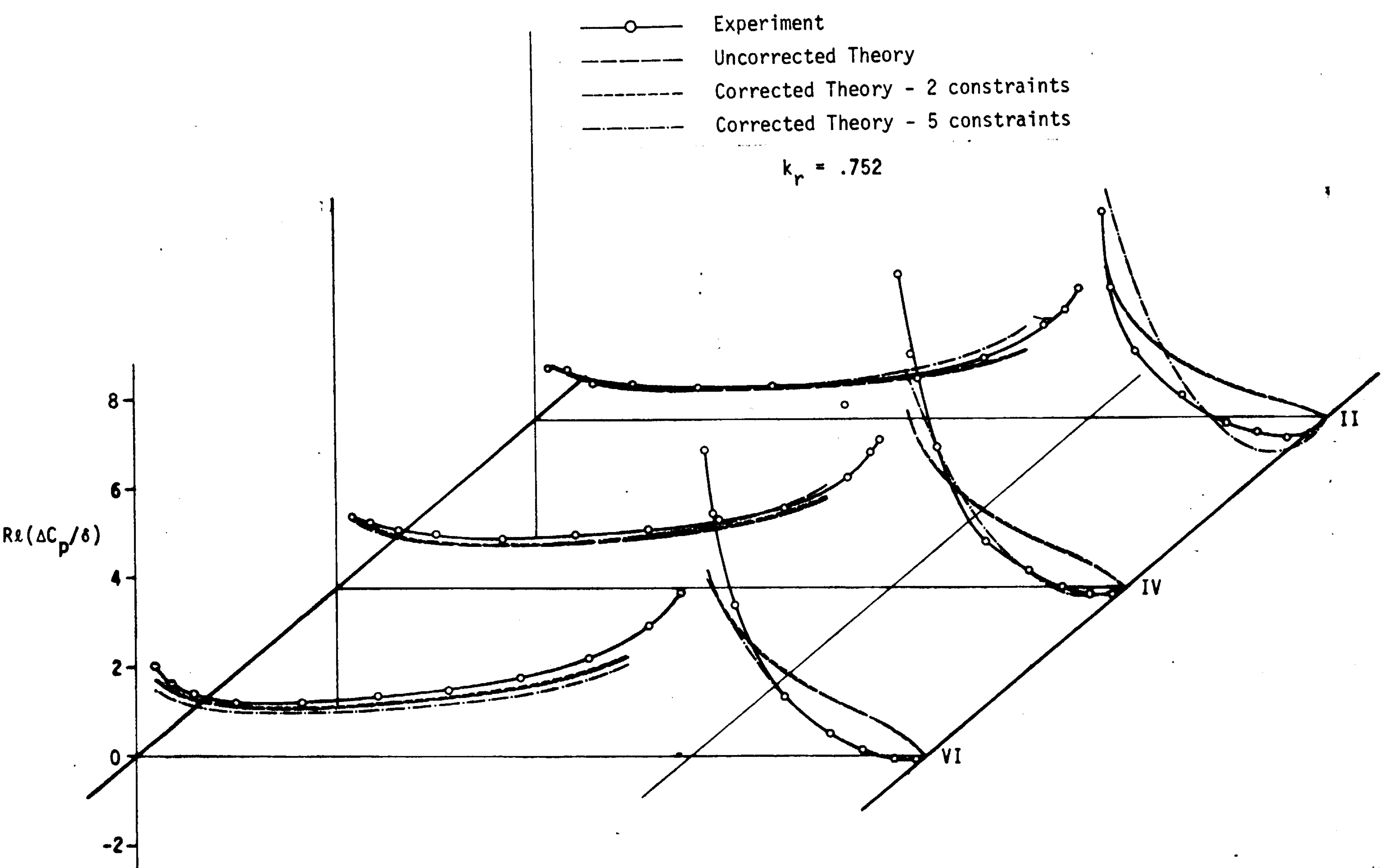


Figure 23. - Real Part of Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Postmultiplying Correction Factors

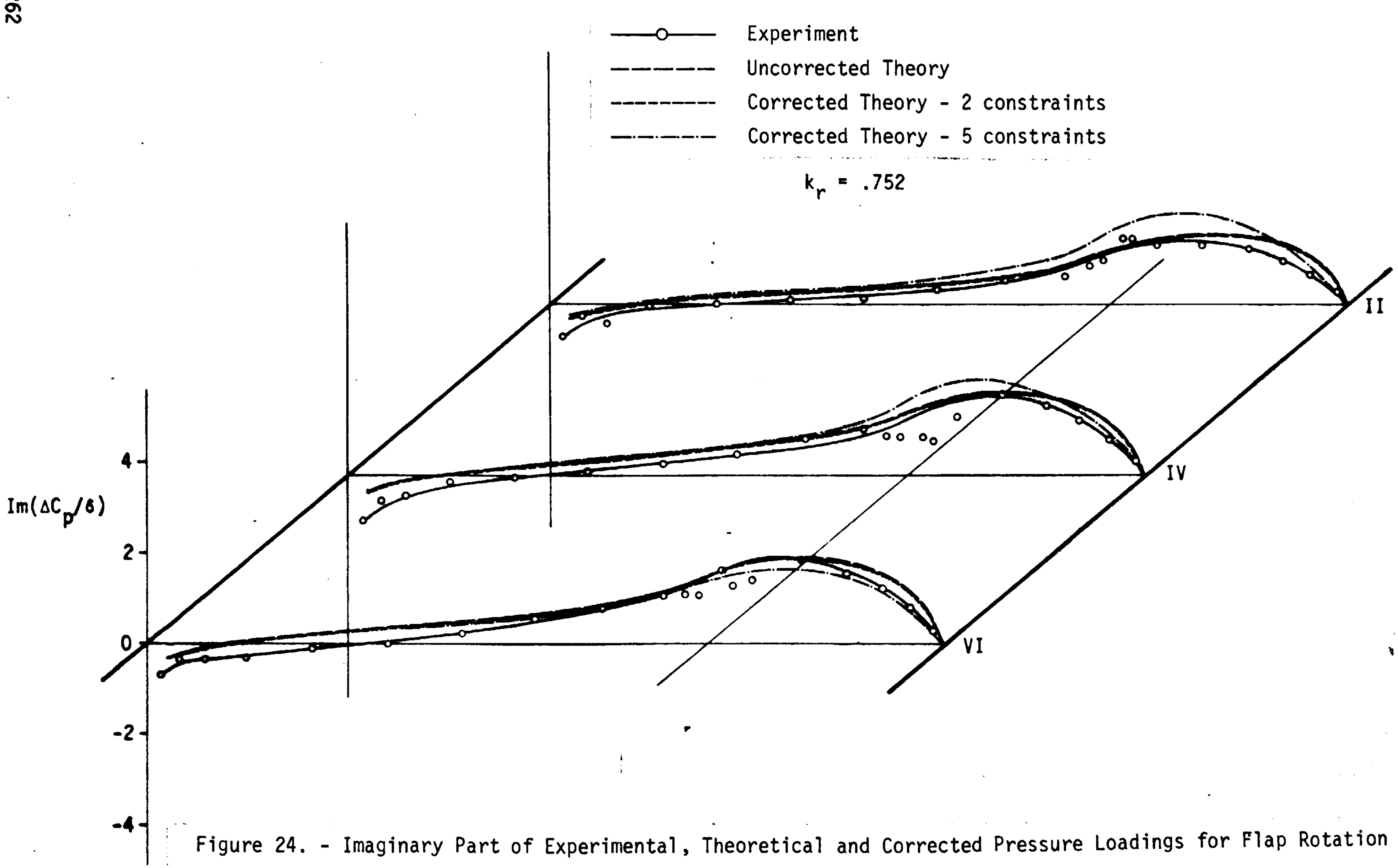


Figure 24. - Imaginary Part of Experimental, Theoretical and Corrected Pressure Loadings for Flap Rotation Using Postmultiplying Correction Factors

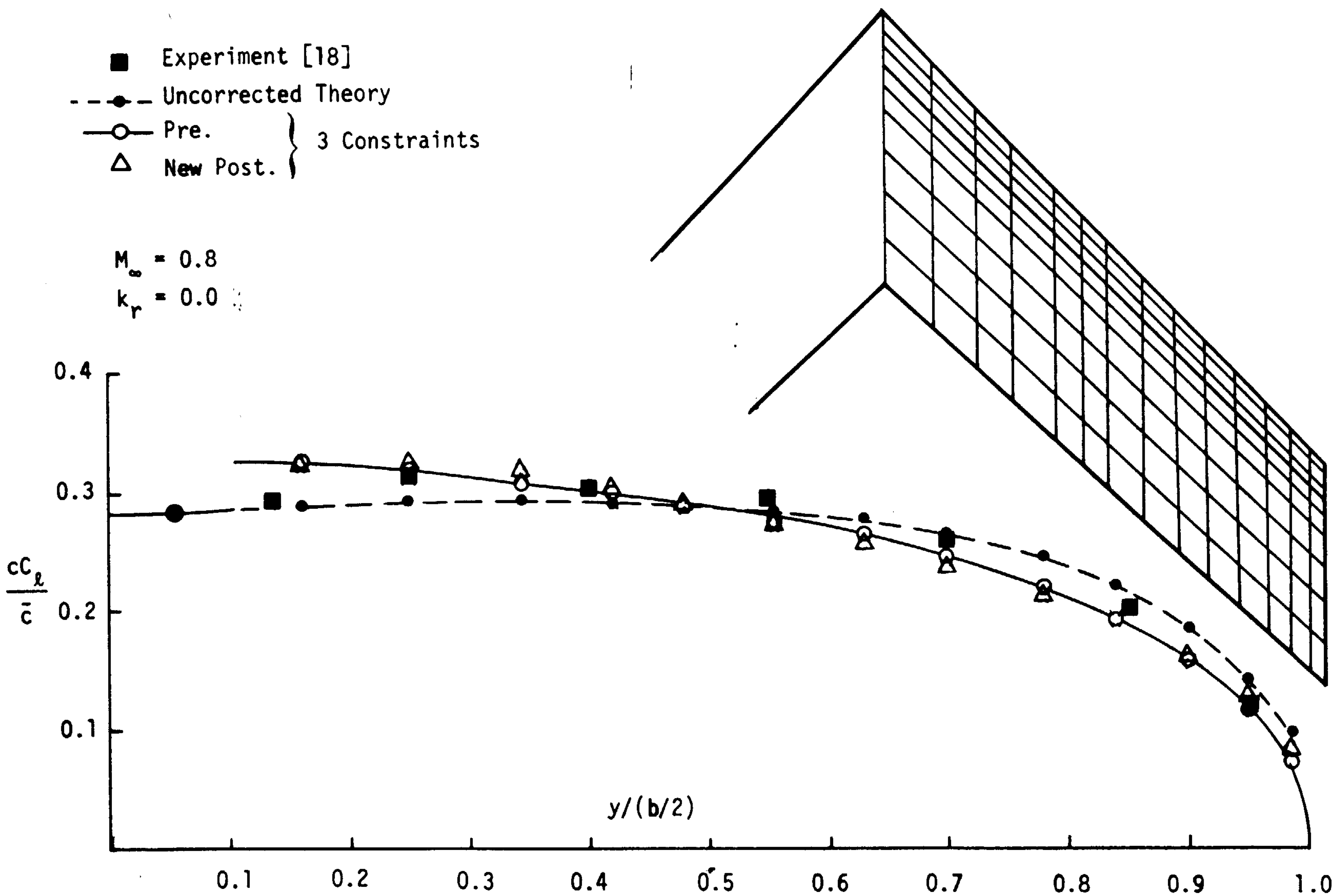


Figure 25. - Comparison of experimental data with corrected and uncorrected theory for a swept wing operating at subsonic speeds and $\alpha = 4^0$.

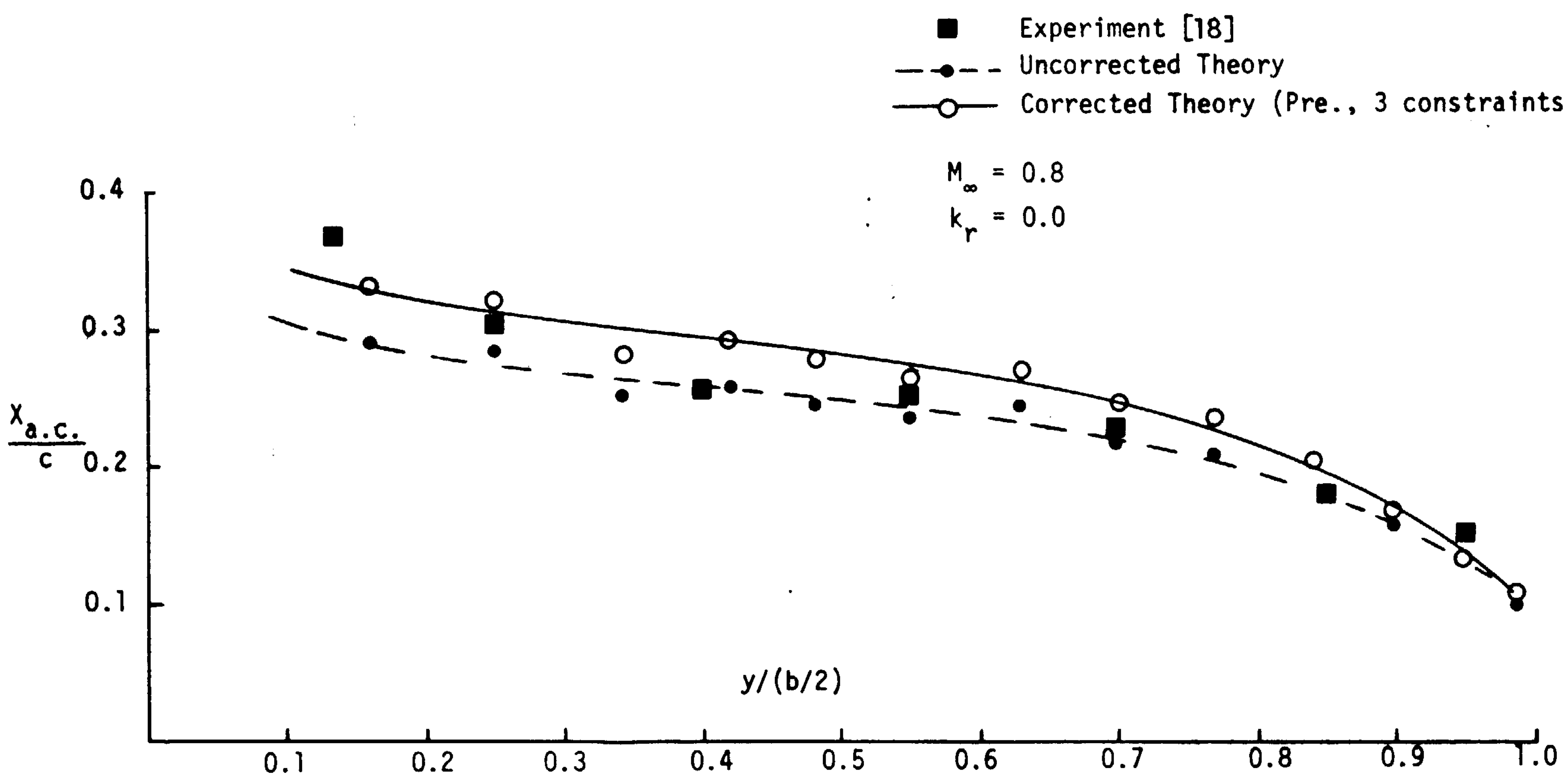


Figure 26. - Comparison of experimental data with corrected and uncorrected theory for a swept wing operating at subsonic speeds at $\alpha = 4^{0}$.

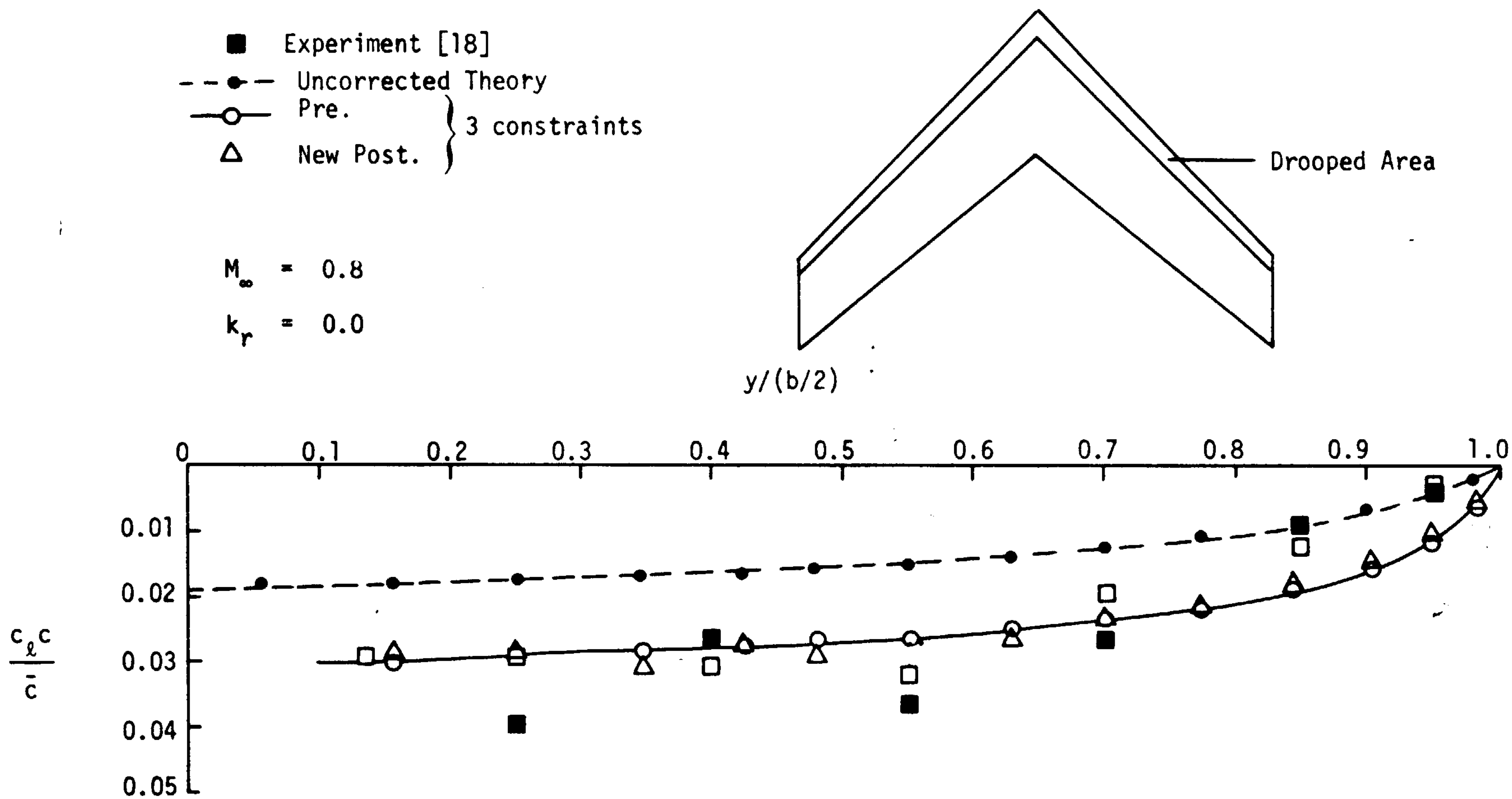


Figure 27. - Comparison of experimental data with corrected and uncorrected theory for a swept wing operating at subsonic speeds with a leading edge droop of 6^0.

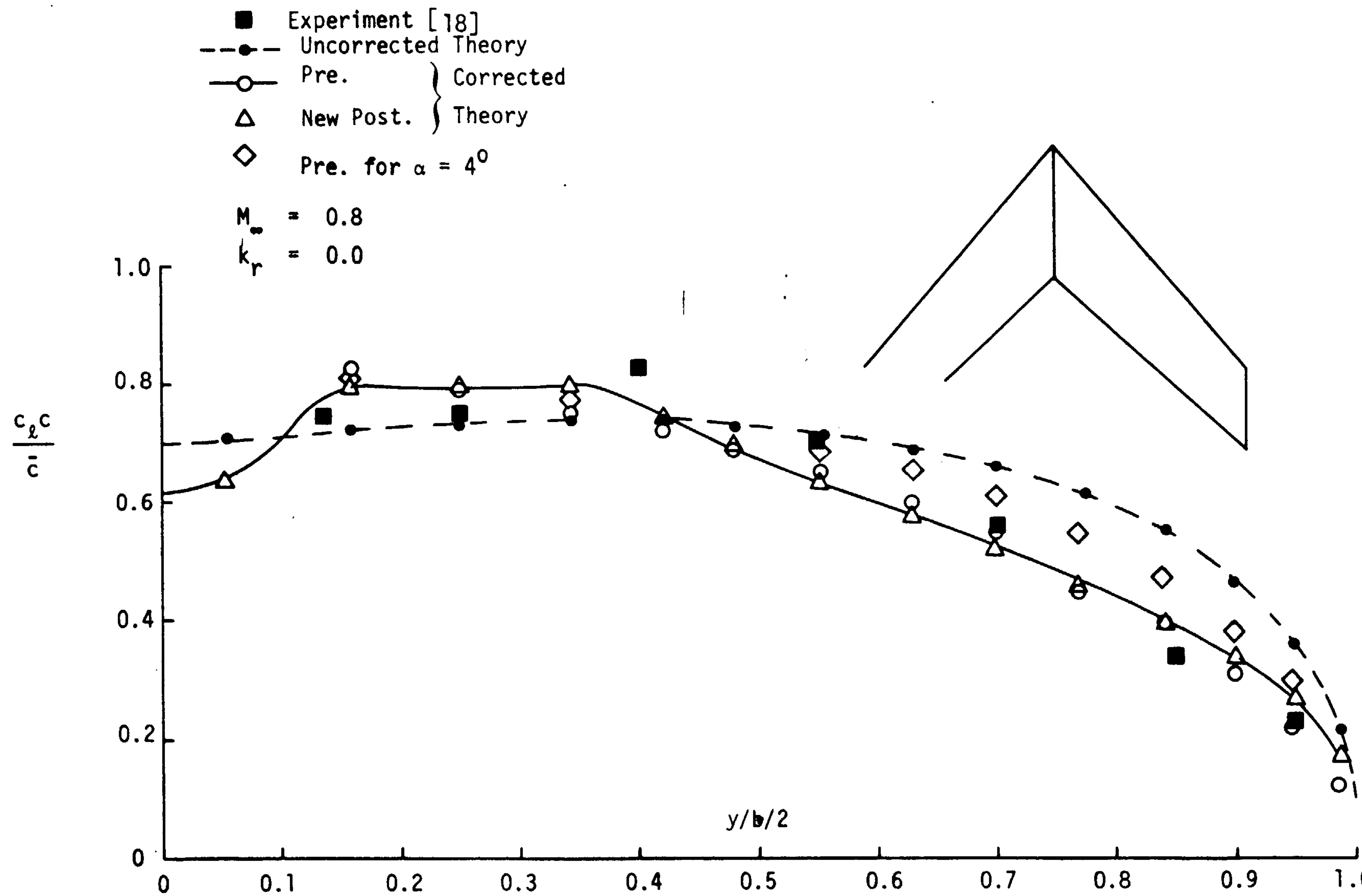


Figure 28. - Comparison of experimental data with corrected and uncorrected theory for a swept wing operating at subsonic speeds at $\alpha = 10^0$.

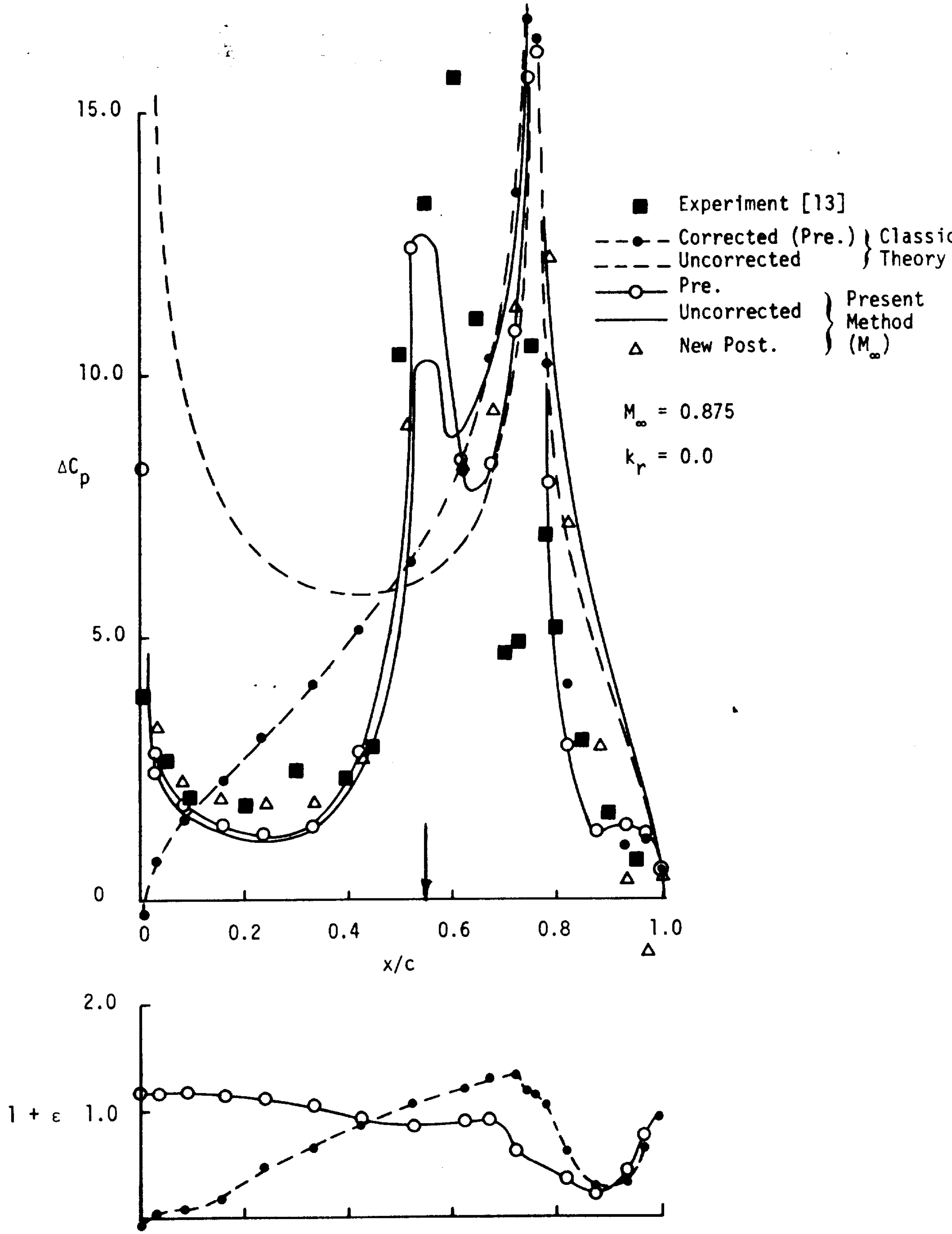


Figure 29. - Comparison of corrected and uncorrected classic theory and Present Method (M_∞) with experimental data.

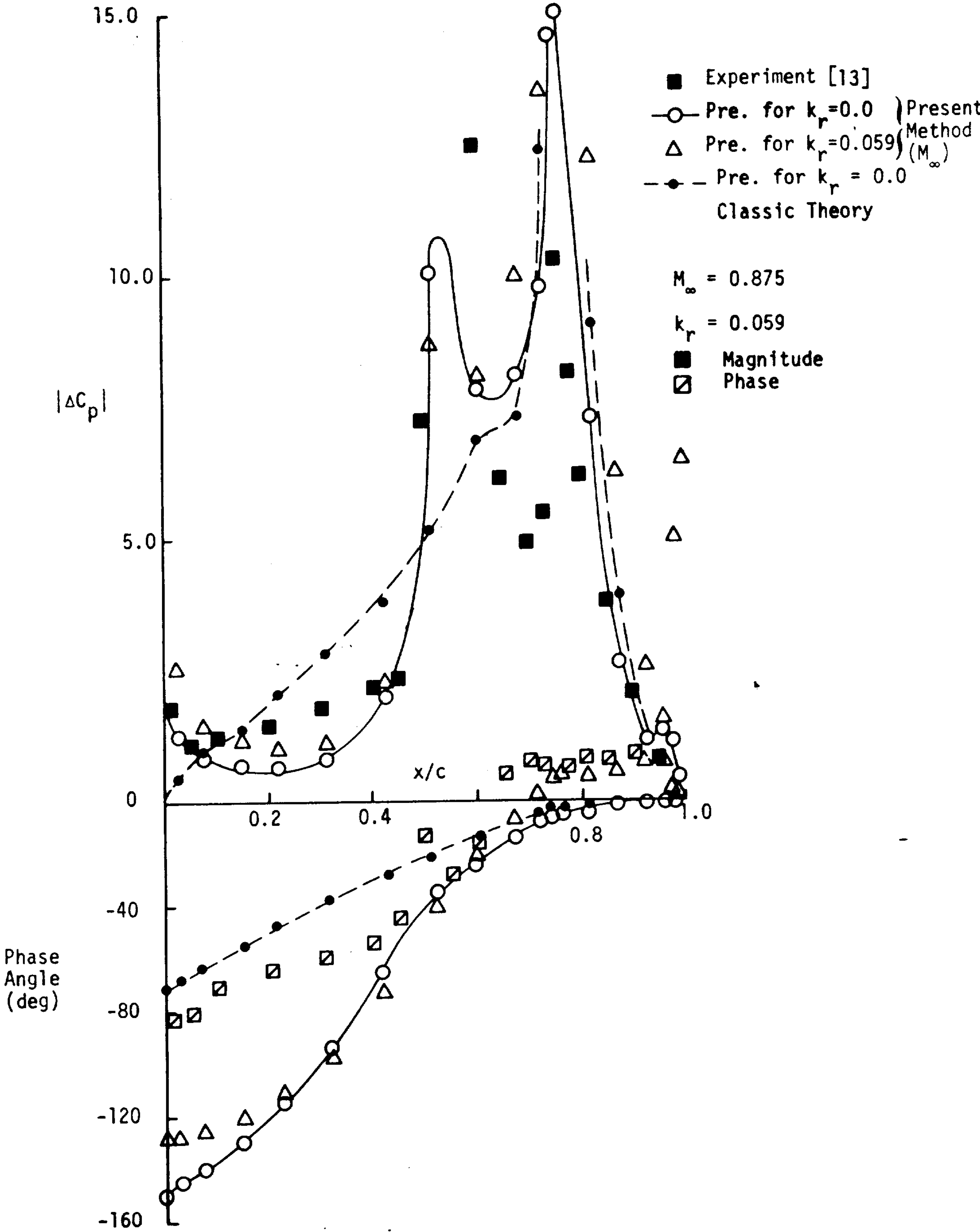


Figure 30. - Comparison of experimental data with corrected theory; the corrections being based on static and oscillatory data.

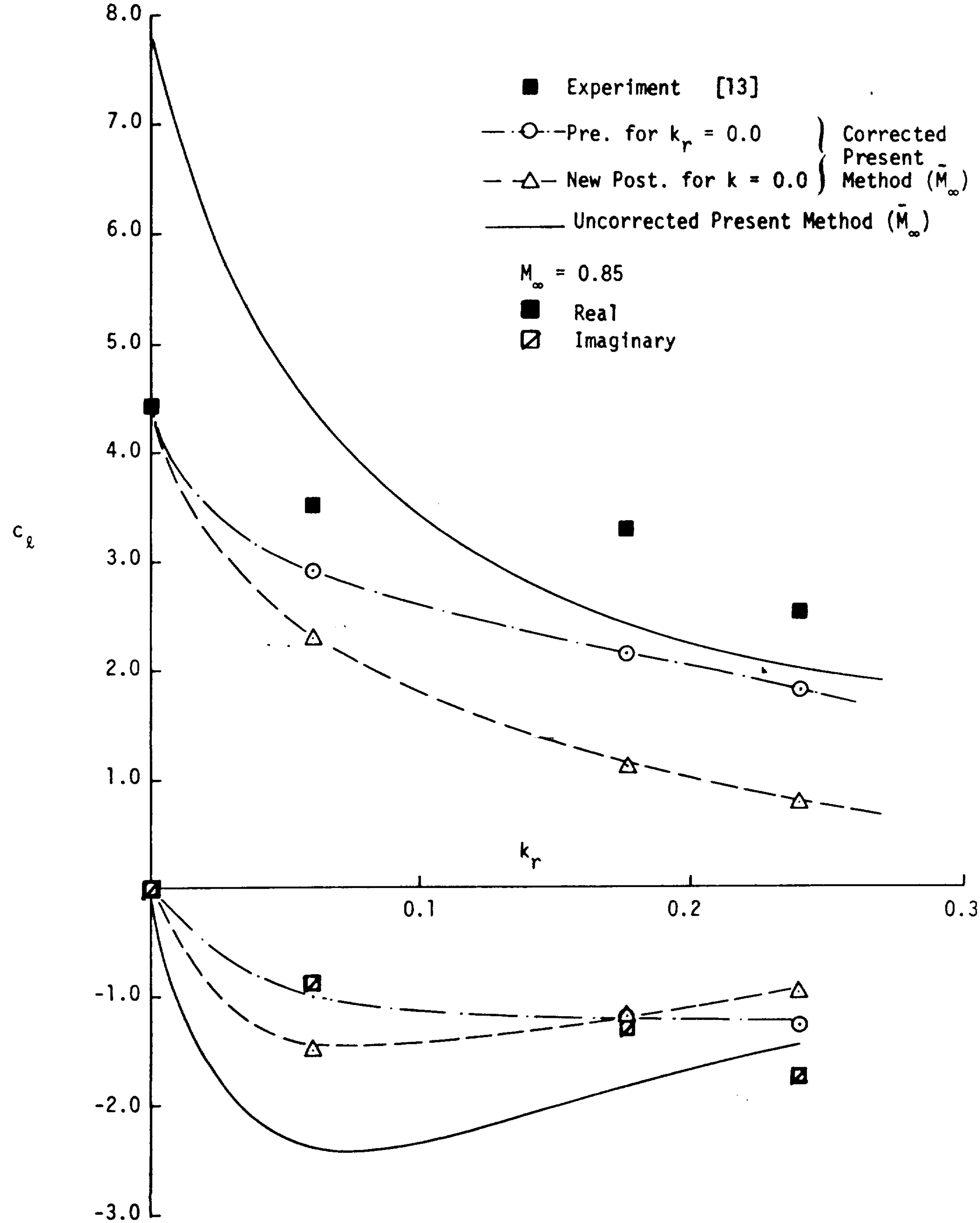


Figure 31. - Effect of applying static correction factors to the Present Method ($\bar{M}_\infty$) for the oscillatory case.

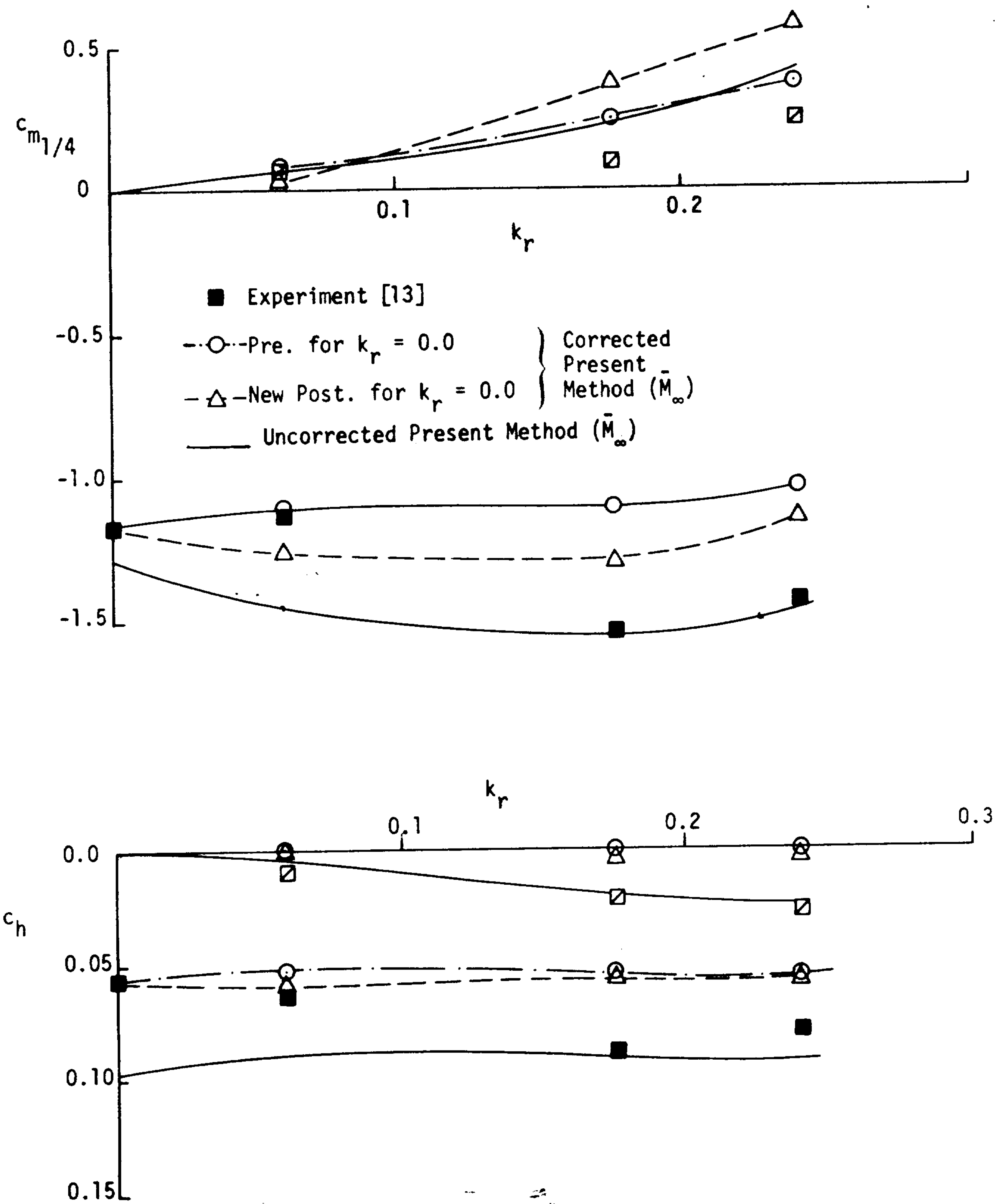


Figure 32. - Effect of applying static correction factors to the Present Method ($\bar{M}_\infty$) for the oscillatory case.

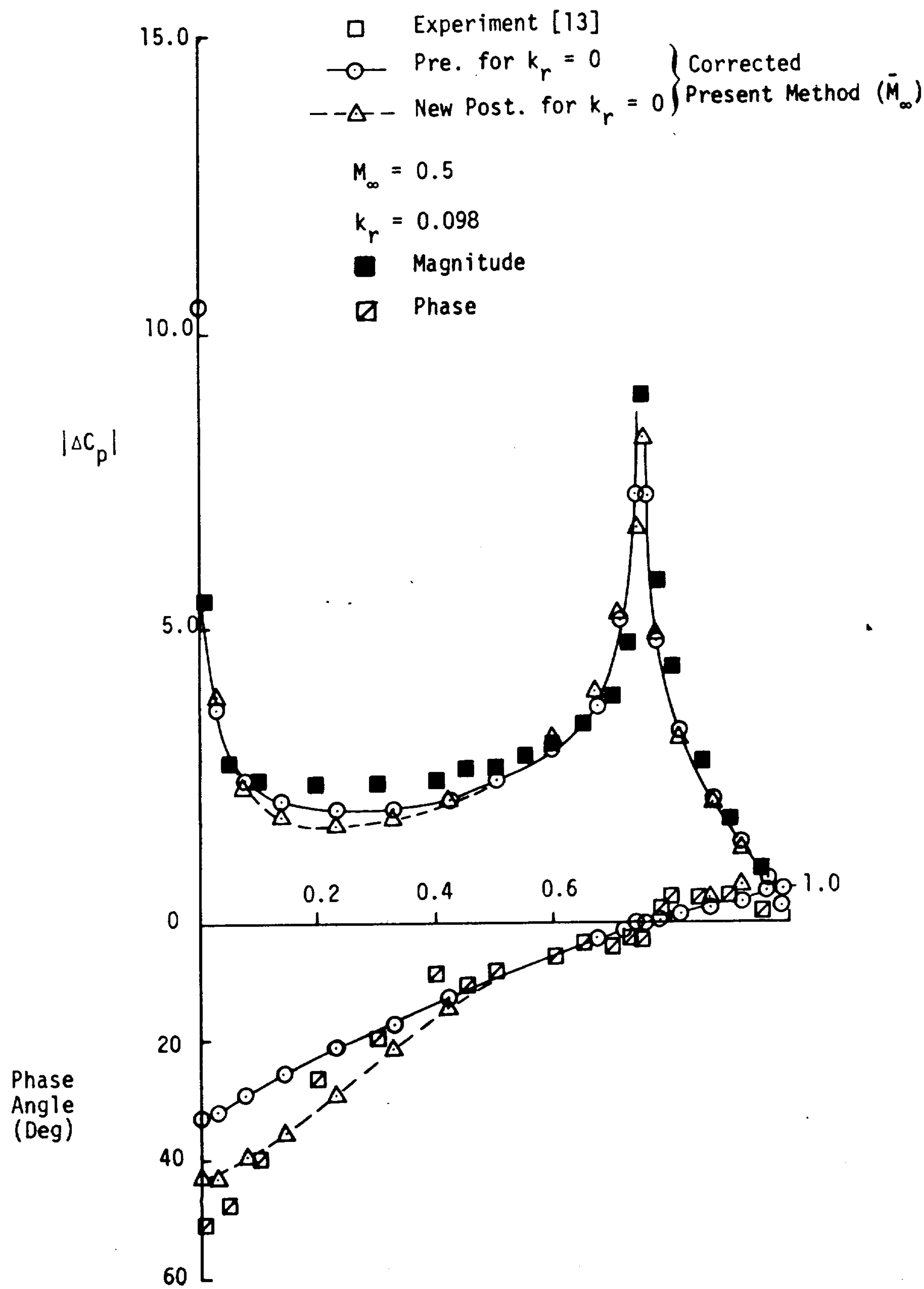


Figure 33. - Application of static correction factors to the oscillatory case.

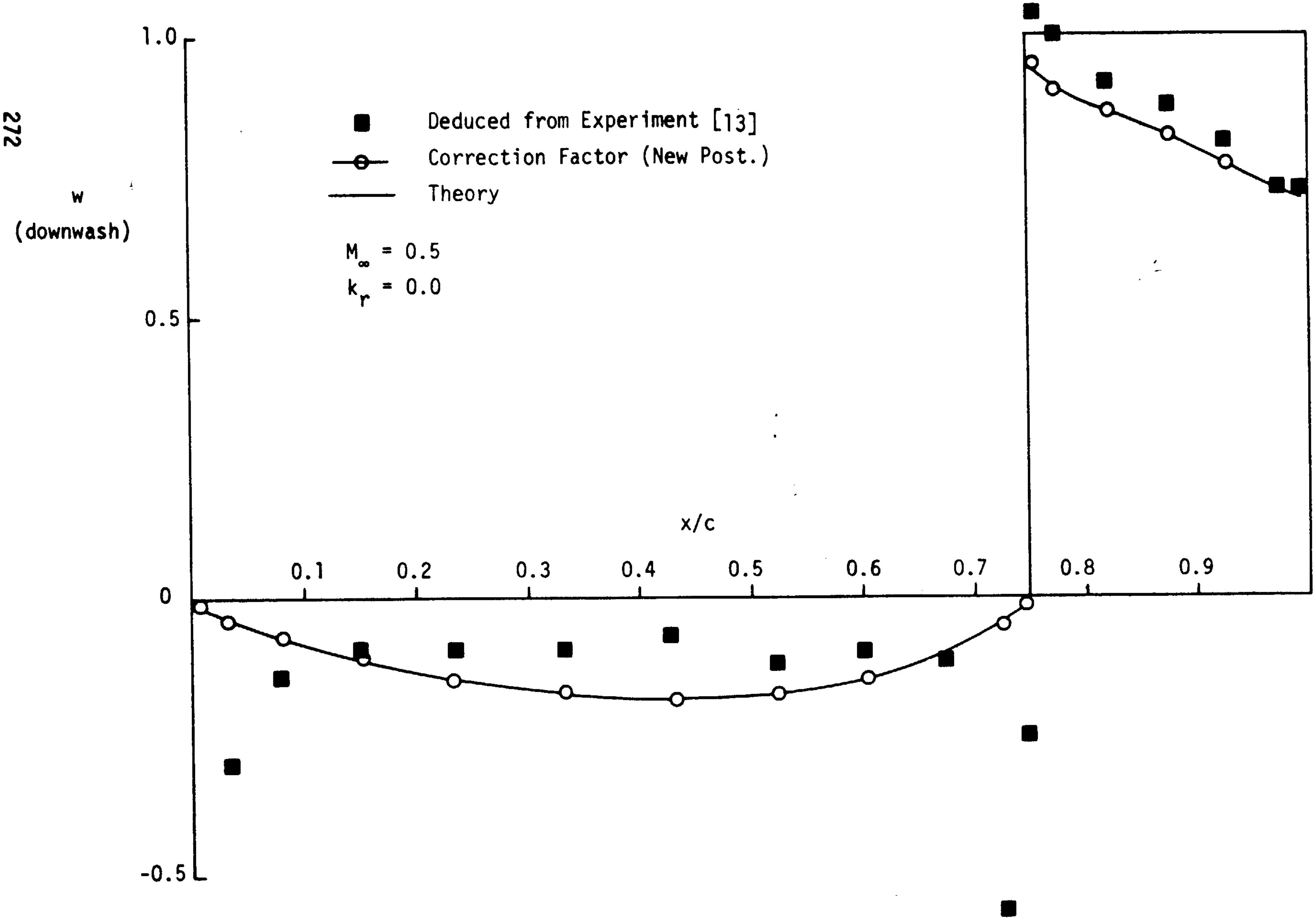


Figure 34. - Comparison of theoretical, corrected, and experimental downwash induced by a control surface deflection on a two-dimensional section.

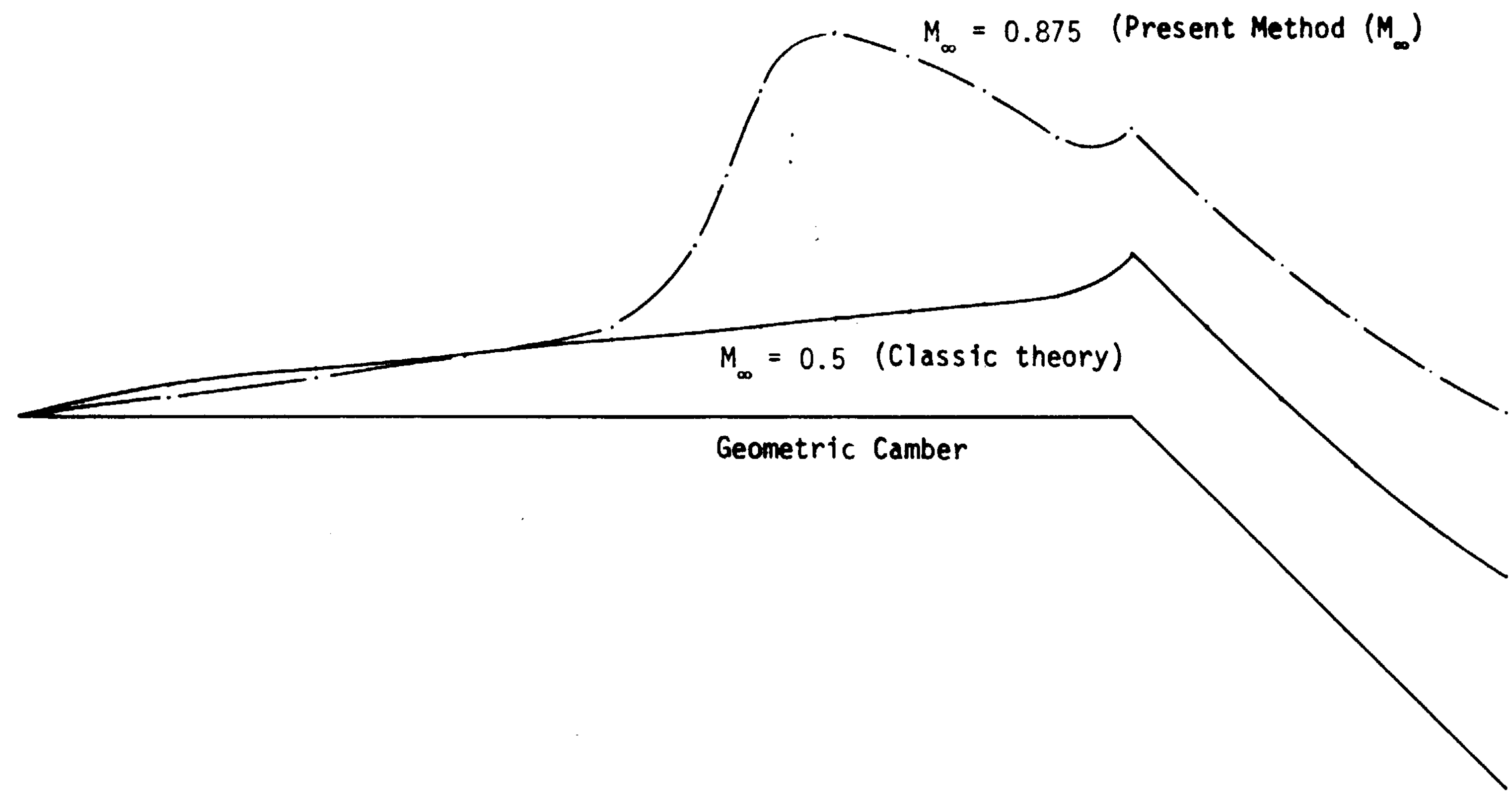


Figure 35. - Comparison of the apparent camber distribution induced at sub- and transonic Mach Numbers with the geometric camber for a control surface deflection on a two-dimensional section.

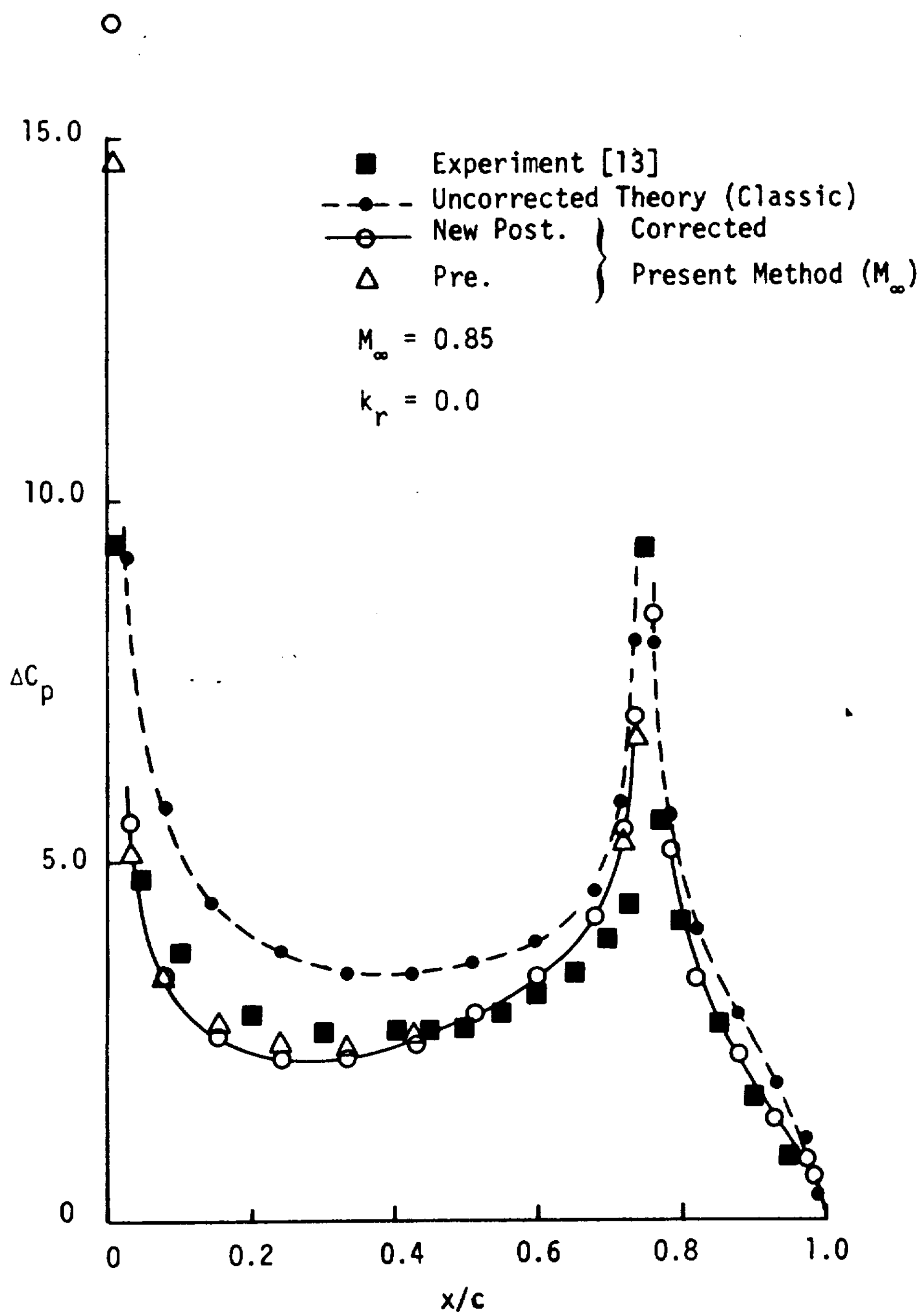


Figure 36. - Comparison of data with premultiplier and postmultiplier (New) corrected theory.

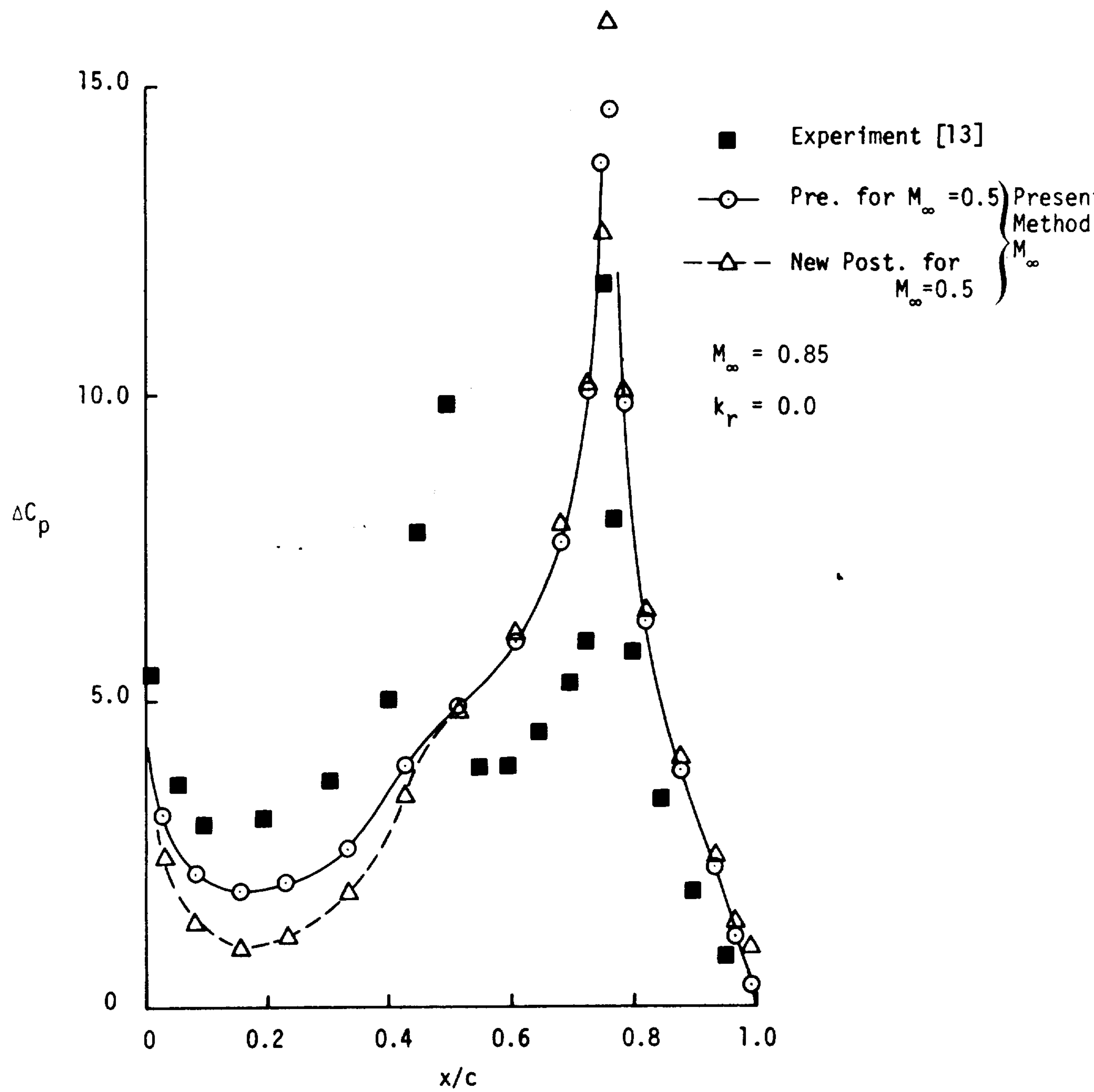


Figure 37. - Application of correction factors obtained at subsonic Mach Number (M_∞ = 0.5) to the transonic case (M_∞ = 0.85).

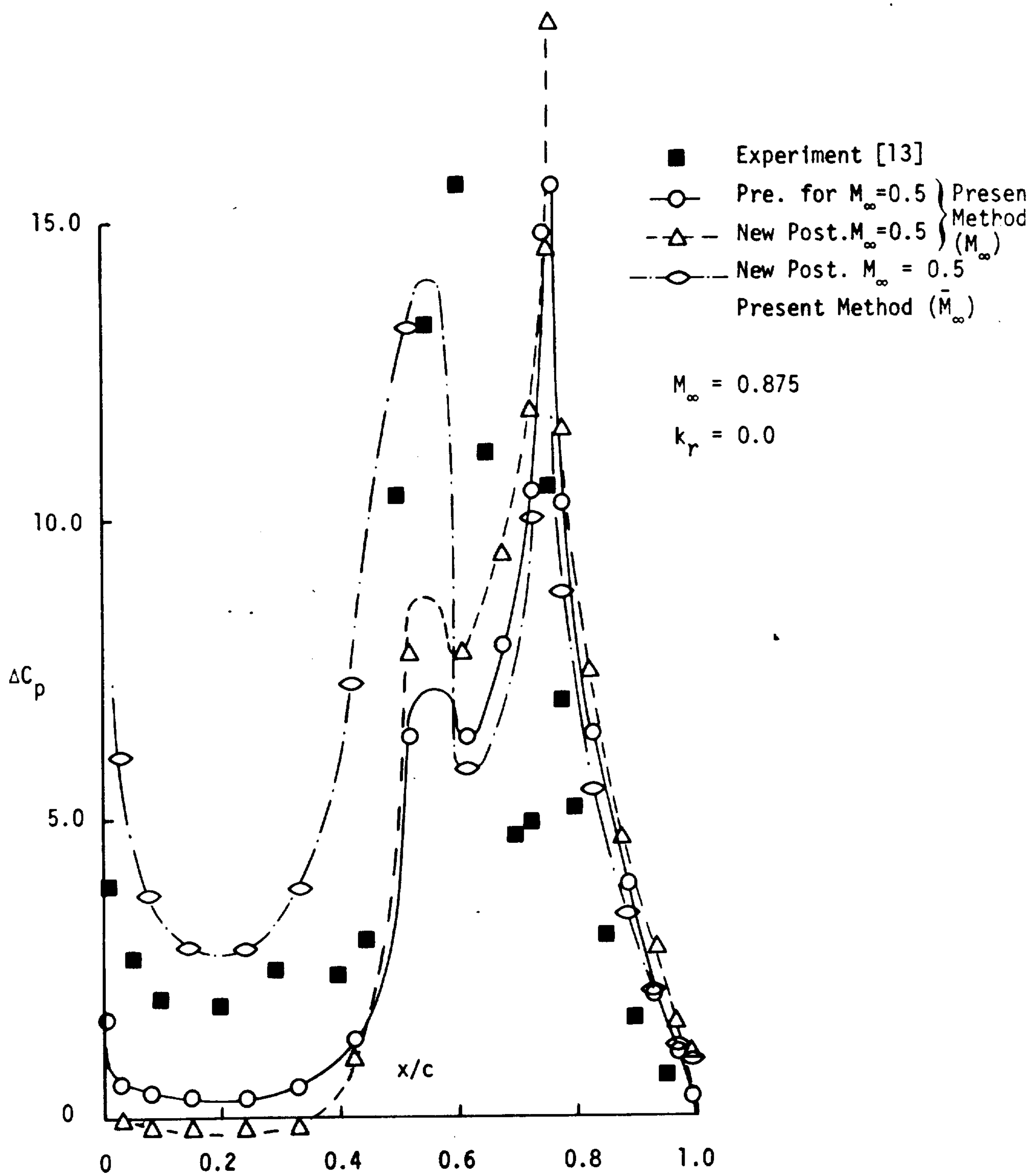


Figure 38. - Application of correction factors obtained at a subsonic Mach Number ($M_\infty = 0.5$) to theory for the transonic case ($M_\infty = 0.875$).

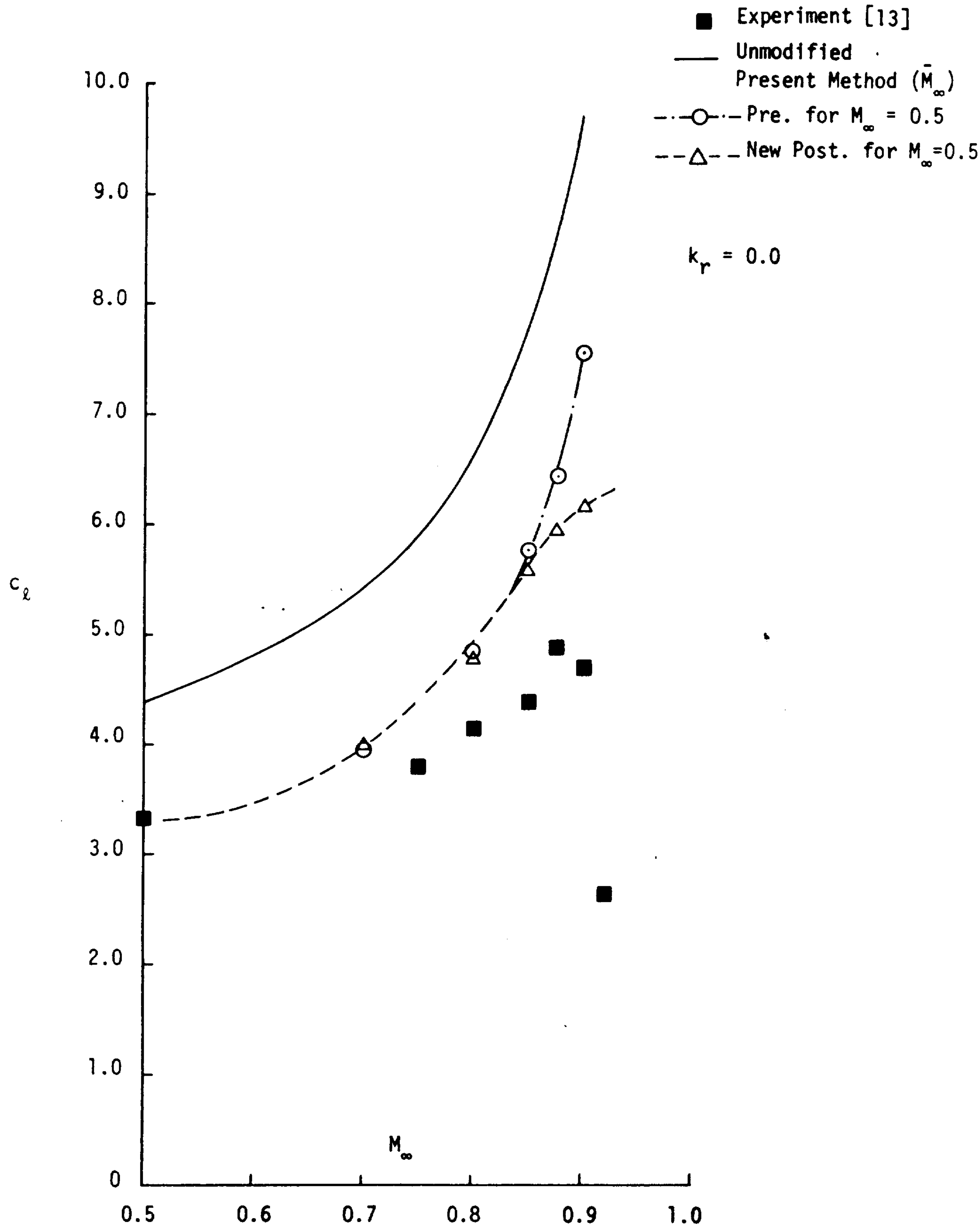


Figure 39. - Effect of applying correction factors obtained at M_∞ = 0.5 to other Mach Numbers for the Present Method ($\bar{M}_\infty$).

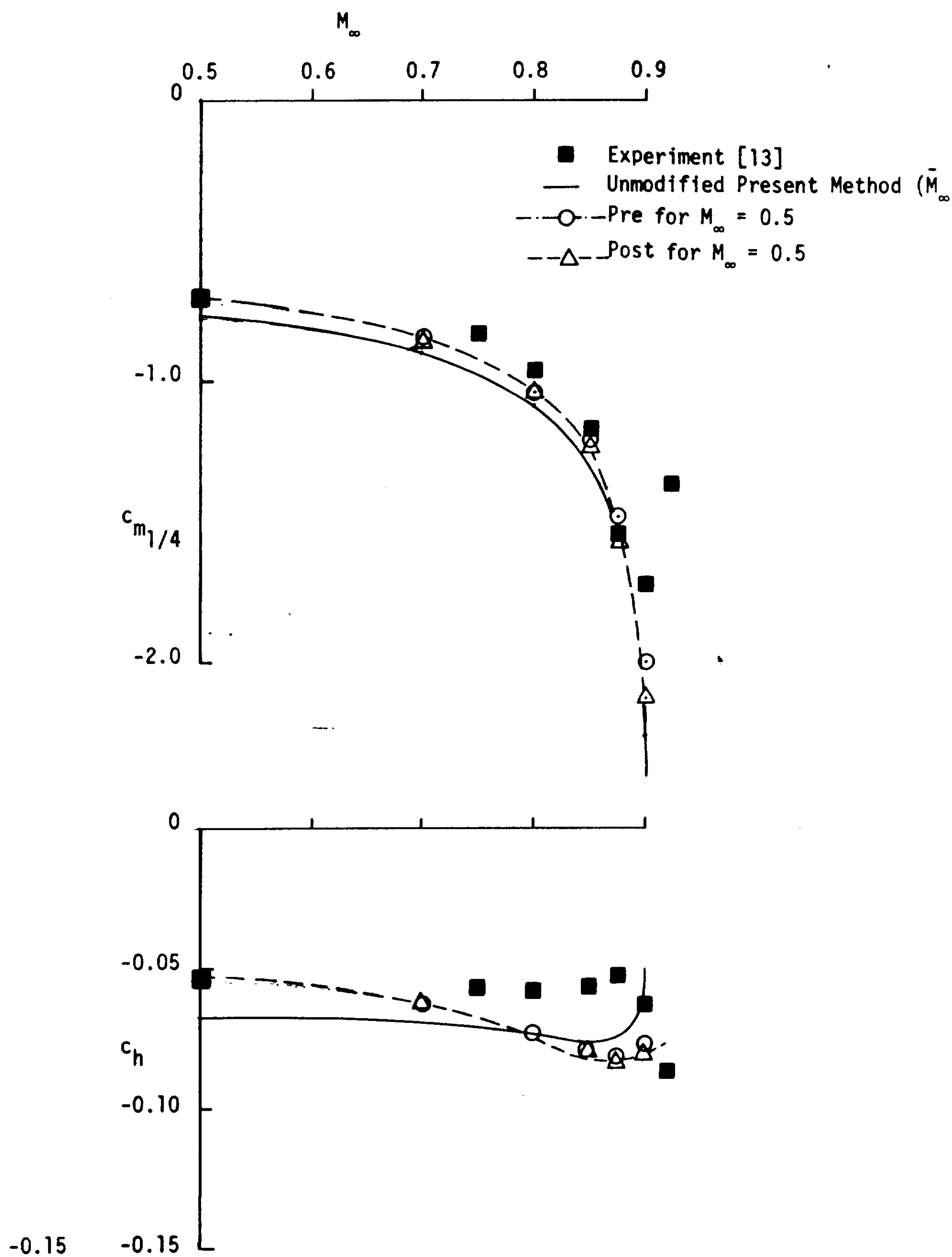


Figure 40. - Effect of applying correction factors obtained at $M_\infty = 0.5$ to other Mach Numbers for the Present Method ($\bar{M}_\infty$)

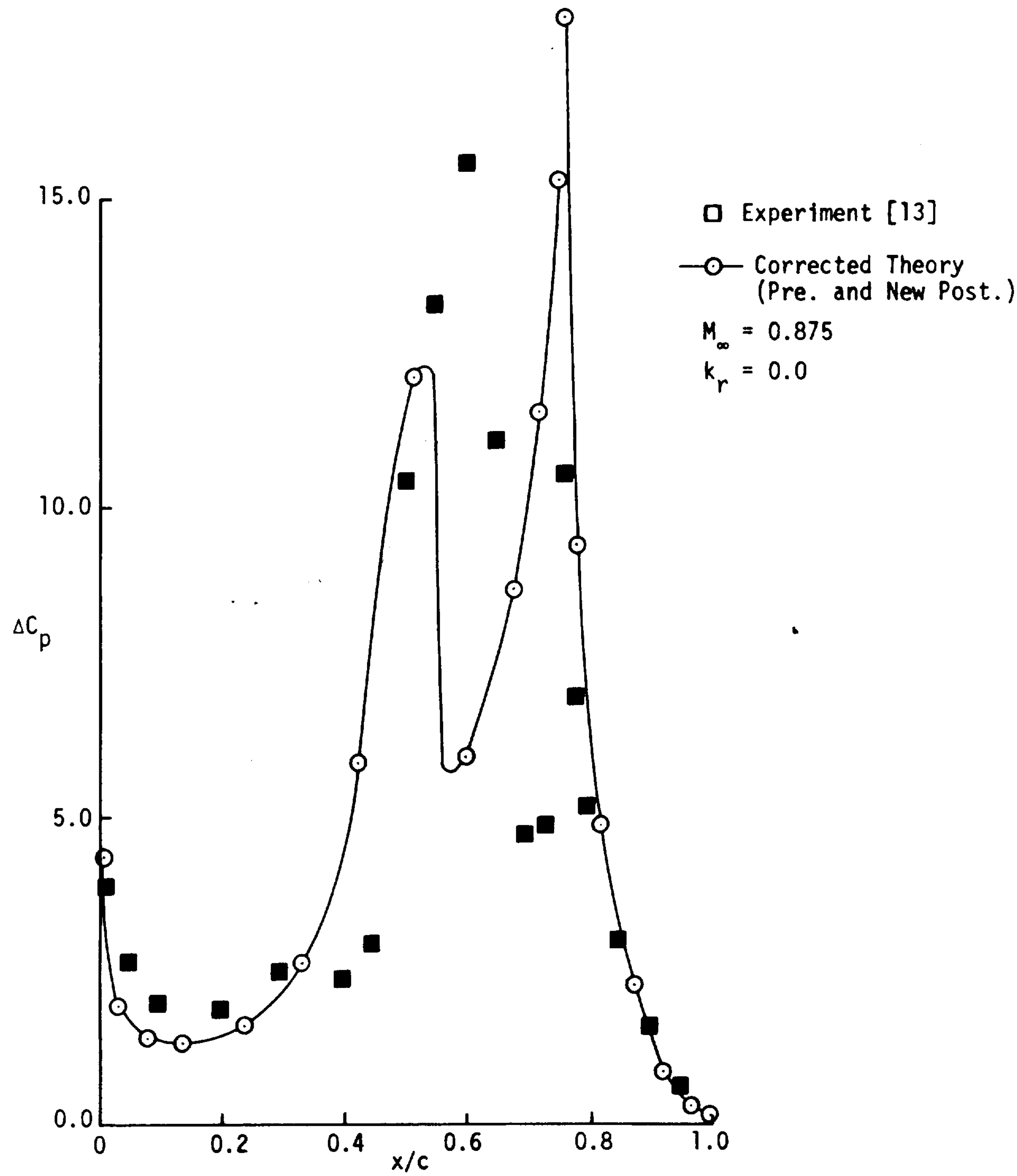


Figure 41. - Results of applying postmultiplying correction factors (New) obtained at M = 0.5 and premultiplying factors obtained at M = 0.875 to theory.

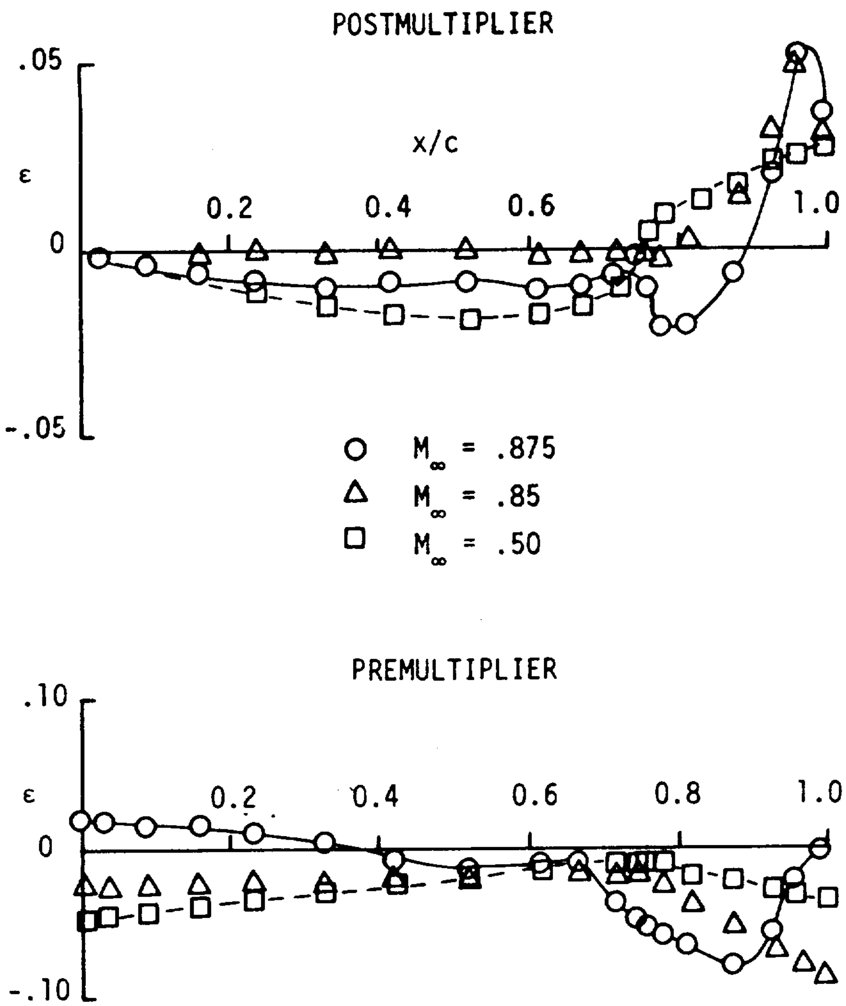


Figure 42. - Comparison of incremental correction factors obtained at various Mach Numbers using the Present Method (M_∞).

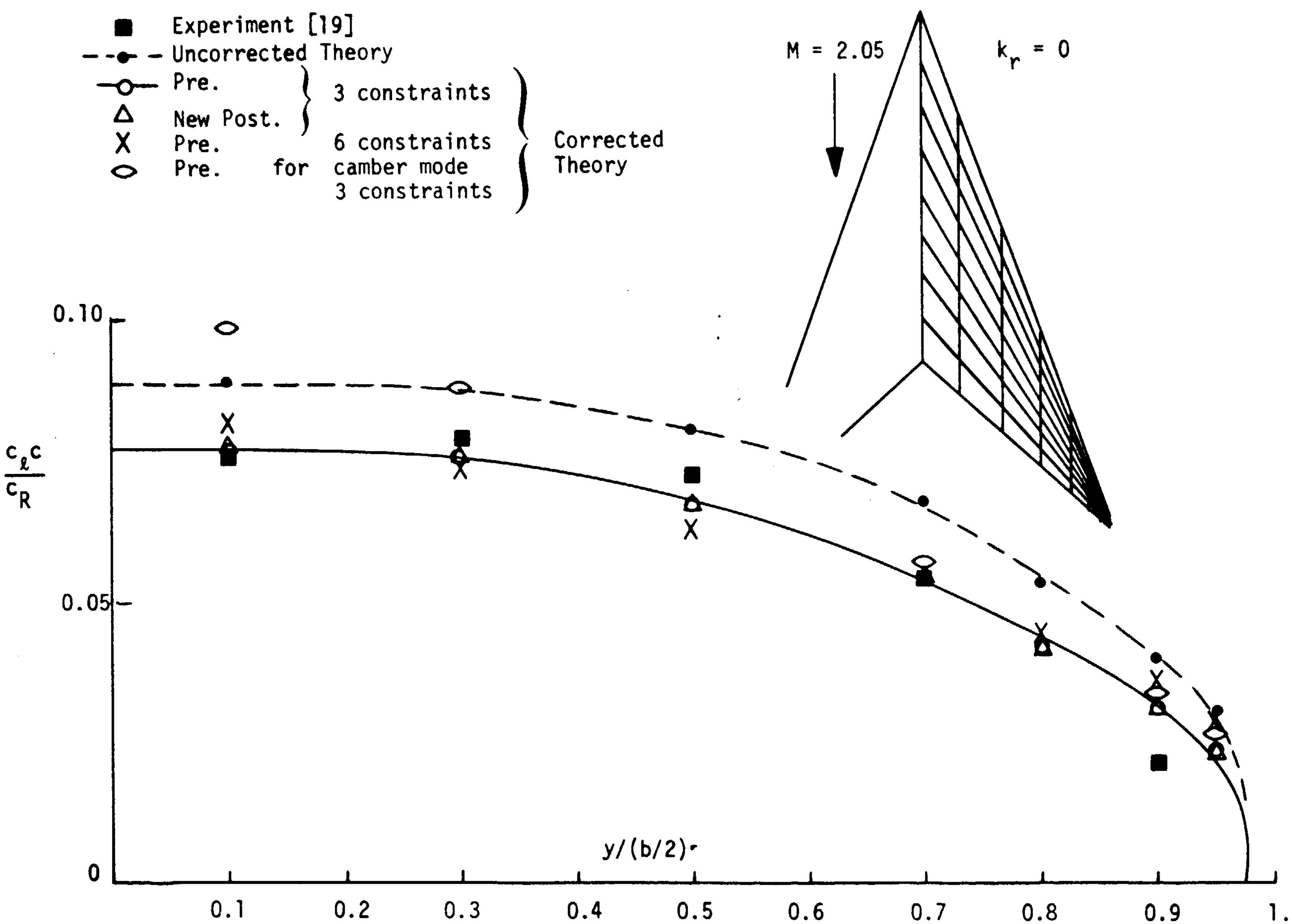


Figure 43. - Comparison of experimental data with corrected and uncorrected theory for an arrow wing in pitch ($\alpha = 4^{o}$) operating at supersonic speeds.

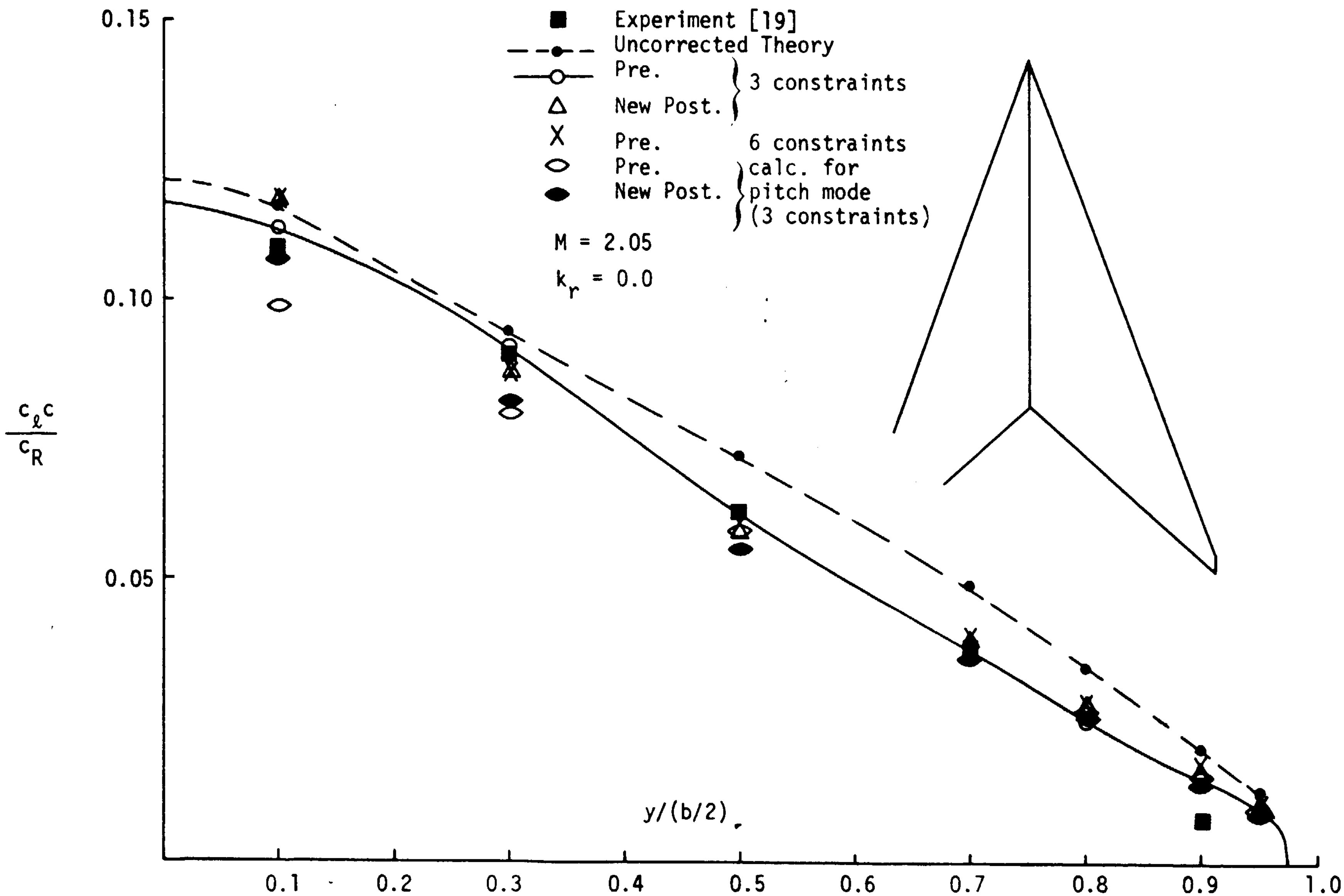


Figure 44. - Comparison of experimental data with corrected and uncorrected theory for an arrow wing with camber operating at supersonic speeds.

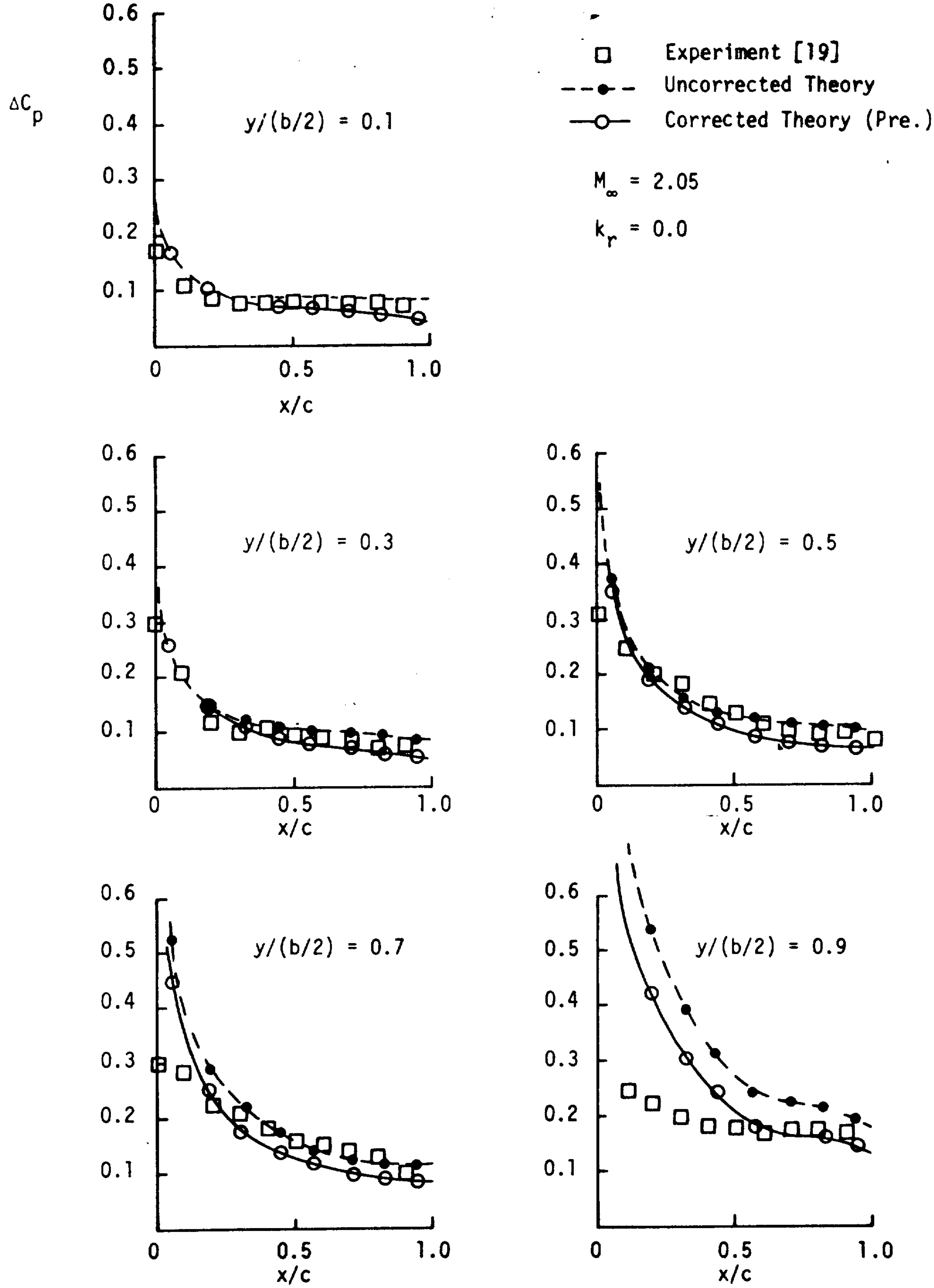


Figure 45. - Comparison of experimental data with corrected and uncorrected tneory for an Arrow wing in pitch.

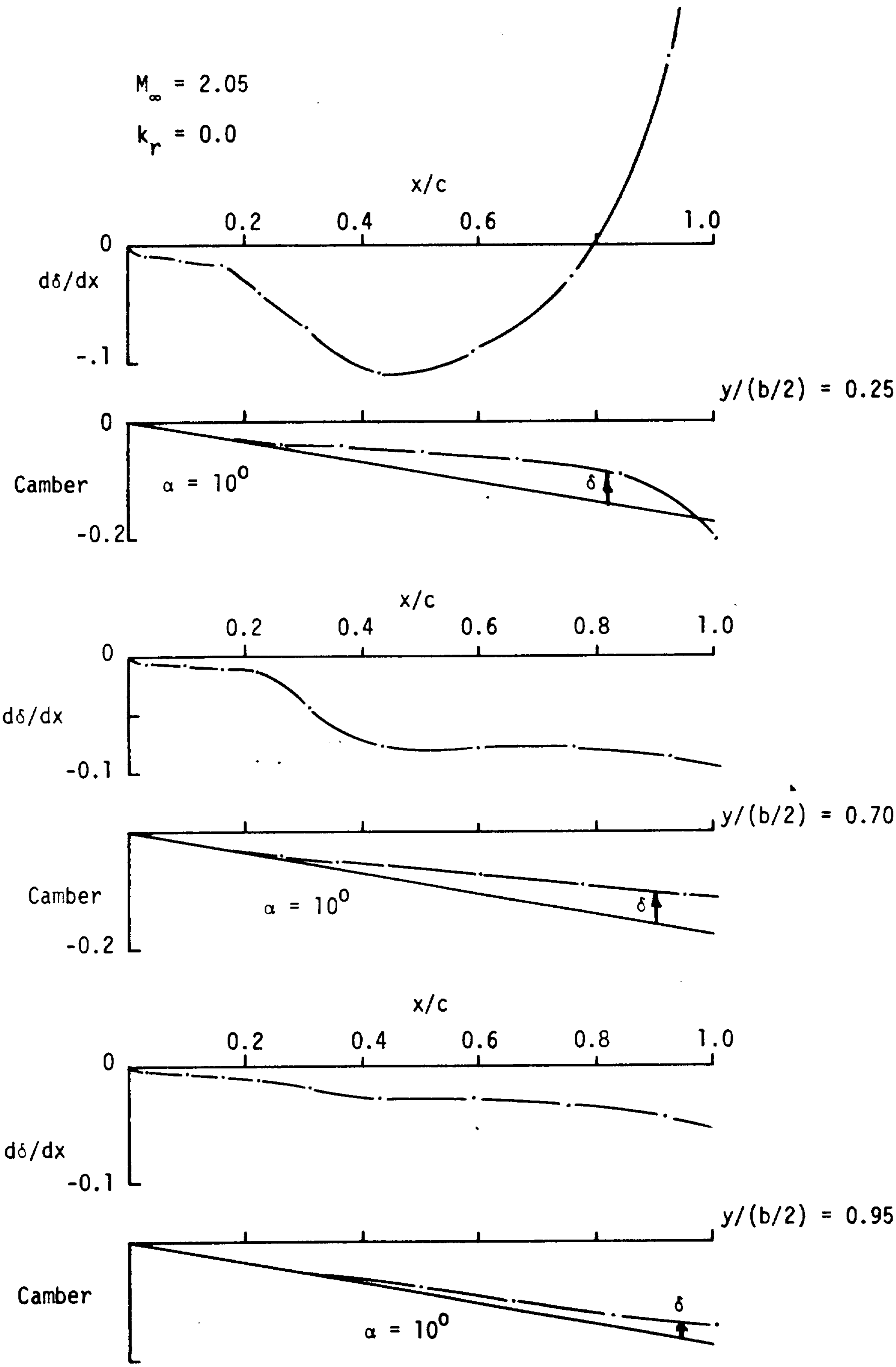
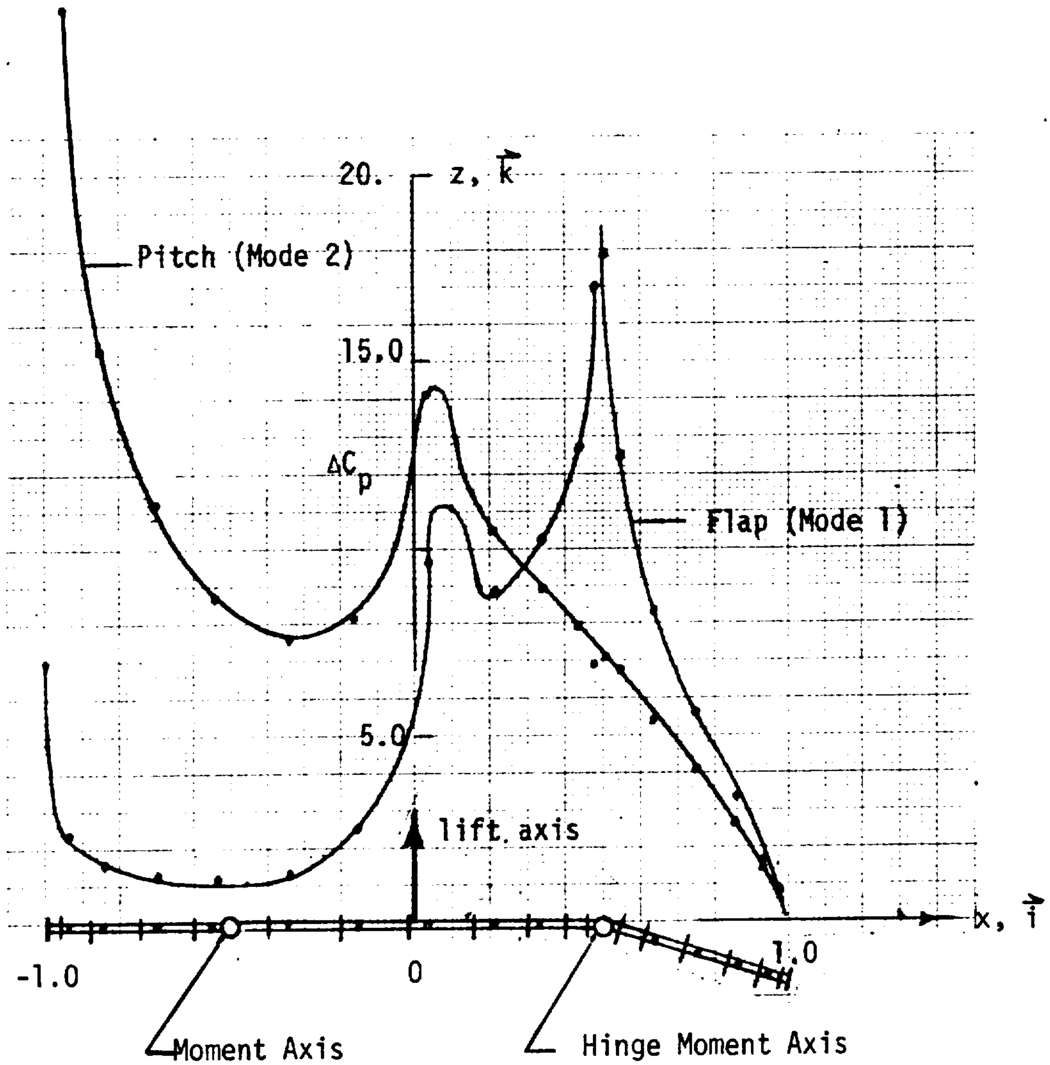


Figure 46. - Camber changes inferred by New Postmultiplying correction factors as applied to an Arrow wing operating at supersonic speeds and $\alpha = 10^o$.



Axes Data

Axis No.	Origin $\xi^{(1)}$	Origin $\eta^{(1)}$	Origin $\zeta^{(1)}$	Direction Cosines $\cos\alpha$	Direction Cosines $\cos\beta$	Direction Cosines $\cos\gamma$
1	0	0	0	0	0	1
2	-.5	0	0	0	1	0
3	+.5	0	0	0	1	0

Lift and Moment Data

	Flap Exp.	Flap Theo.	Pitch Estimate	Pitch Theo.
c_ℓ	4.93	5.56	8.8	10.0
$c_{m1/4}$	-1.57	-2.09		
$c_{h3/4}$	-0.053	- .141		

Figure 47. - Graphical and Tabulated Data for Program Test Cases

1. Report No. NASA CR-144967	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle CORRECTION FACTOR TECHNIQUES FOR IMPROVING AERODYNAMIC PREDICTION METHODS		5. Report Date May 1976
		6. Performing Organization Code
7. Author(s) Joseph P. Giesing, Terez P. Kalman, William P. Rodden		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address McDonnell Douglas Corporation Douglas Aircraft Company 3855 Lakewood Boulevard Long Beach, California 90846		11. Contract or Grant No. NAS1-13835
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C.		14. Sponsoring Agency Code
15. Supplementary Notes Final Report, for NASA Langley Research Center; Project Manager Irving Able		

16. Abstract

A method for correcting discrete element lifting surface theory to reflect given experimental data is presented. Theoretical pressures are modified such that imposed constraints are satisfied (e.g., lift, moment, etc.) while minimizing the changes to the pressures. Several types of correction procedures are presented and correlated; (1) scaling of pressures; (2) scaling of downwash values and (3) addition of an increment to the downwash that is proportioned to pressure. Some special features are included in these methods and they include: (1) consideration of experimental data from multiple deflection modes, (2) limitation of the amplitudes of the correction factors and (3) the use of correction factor mode shapes. These methods are correlated for cases involving all three Mach Number ranges using a FORTRAN IV computer program. Subsonically, a wing with an oscillating partial span control surface and a wing with a leading edge droop are presented. Transonically a two-dimensional airfoil with an oscillating flap is considered. Supersonically an arrow wing with and without camber is analyzed.

In addition to correction factor methods an investigation is presented dealing with a new simplified transonic modification of the two-dimensional subsonic lifting surface theory. Correlations are presented for an airfoil with an oscillating flap.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Correction factors Oscillatory aerodynamics Control surface aerodynamics Semiempirical aerodynamic corrections Transonic flow	Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	285	

* For sale by the National Technical Information Service, Springfield, Virginia 22151