https://ntrs.nasa.gov/search.jsp?R=19740010533 2020-03-23T10:42:52+00:00Z

NASA CONTRACTOR REPORT

NASA CR-2335

NASA CR-2335

INVESTIGATION OF NONLINEAR INVISCID AND VISCOUS FLOW EFFECTS IN THE ANALYSIS OF DYNAMIC STALL

by Peter Crimi

Prepared by
AVCO SYSTEMS DIVISION
Wilmington, Mass.
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1974

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-2335		

4. Title and Subtitle	5. Report Date
INVESTIGATION OF NONLINEAR INVISCID AND VISCOUS FLOW EFFECTS IN THE ANALYSIS OF DYNAMIC STALL	February 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Peter Crimi	--
	10. Work Unit No.
9. Performing Organization Name and Address	
AVCO Systems Division Wilmington, MA	11. Contract or Grant No. NAS 1-11245
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Final Report

16. Abstract

A recently developed method for analyzing unsteady airfoil stall was refined by including nonlinear effects in the representation of the inviscid flow. Certain other aspects of the potential-flow model were reexamined and the effects of varying Reynolds number on stall characteristics were investigated. Refinement of the formulation improved the representation of the flow and chordwise pressure distribution below stall, but substantial quantitative differences between computed and measured results are still evident for sinusoidal pitching through stall. Agreement is substantially improved by assuming the growth rate of the dead-air region at the onset of leading-edge stall is of the order of the component of the free stream normal to the airfoil chordline. The method predicts the expected increase in the resistance to stalling with increasing Reynolds number. Results indicate that a given airfoil can undergo both trailing-edge and leading-edge stall under unsteady conditions.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Dynamic stall analysis, oscillating airfoil, Reynolds number, leading-edge stall	Unlimited - Unclassified

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	112	$4.25

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

INVESTIGATION OF NONLINEAR INVISCID AND VISCOUS FLOW EFFECTS IN THE ANALYSIS OF DYNAMIC STALL

by Peter Crimi
Avco Systems Division

SUMMARY

A recently developed method for analyzing unsteady airfoil stall was refined by including nonlinear effects in the representation of the inviscid flow. Certain other aspects of the potential-flow model were reexamined and the effects of varying Reynolds number on stall characteristics were investigated. Refinement of the formulation improved the representation of the flow and chordwise pressure distribution below stall, but substantial quantitative differences between computed and measured results are still evident for sinusoidal pitching through stall. Agreement is substantially improved by assuming the growth rate of the dead-air region at the onset of leading-edge stall is of the order of the component of the free stream normal to the airfoil chordline. The method predicts the expected increase in the resistance to stalling with increasing Reynolds number. Results indicate that a given airfoil can undergo both trailing-edge and leading-edge stall under unsteady conditions.

INTRODUCTION

Unsteady airfoil stall is a problem of particular concern in the design and operation of helicopter rotors. Periodic stall and unstall of the blades at high advance ratio cause severe oscillatory control loads, increased vibration levels and, under some circumstances, a torsional aeroelastic instability, effectively limiting aircraft performance (Ref. 1). As a result, the problem has been the subject of considerable research (Refs. 2-5, for example).

A method was recently developed for analyzing dynamic stall of a helicopter rotor blade. The method, which is described in Ref. 6, employs a model for each of the basic flow elements involved in the unsteady stall of a two-dimensional airfoil in incompressible flow. The interactions of these elements are analyzed by forward integration in time, with the aid of a digital computer. Results are in good qualitative agreement with measured loads, both dynamic lift overshoot and unstable moment variation being in clear evidence in the computed loading. A number of approximations were employed in the formulations which caused substantial quantitative differences, however.

One of the approximations most open to question was the use of a linearized representation of the potential flow. This study was directed, first, to refining the model of the potential flow by introducing second-order terms. Certain other aspects of the model for a stalled airfoil were also considered. The method was then used to investigate the effects of varying Reynolds number and to determine the relative importance of leading-edge and trailing-edge stall under unsteady conditions.

SYMBOLS

b	airfoil semichord, m
$C_{m\ c/4}$	moment coefficient, $C_{m\ c/4} = m/(2\ \rho U^2 b^2)$
C_n	normal-force coefficient, $C_n = n/(\rho U^2 b)$
C_p	pressure coefficient, $C_p = 2\ (p - p_\infty)/(\rho U^2)$
c	airfoil chord, m
k	reduced frequency, $k = \omega b/U$
l	lift per unit span, N/m
ℓ	length of dead-air region, m
m	moment per unit span about quarter-chord, N
n	force normal to chord line per unit span, N/m
p	static pressure, N/m^2
p_∞	free-stream static pressure, N/m^2
q	fluid speed at airfoil surface, m/s
$Re_{(\ )}$	Reynolds number based on length indicated by subscript
r_o	leading-edge radius, m
T	airfoil section thickness distribution, m
t	time, s
U	free-stream speed, m/s
u_e	fluid speed external to boundary layer
(x, y)	foil-fixed coordinates, with origin at midchord
α	angle of attack, deg or rad
γ	vortex strength, m/s
δ^*	boundary layer displacement thickness, m

θ_p	pitch angle, deg or rad
ρ	fluid density, kg/m^3
σ	source strength, m/s
τ_w	wall shear, N/m^2
ϕ	perturbation velocity potential m^2/s
ω	frequency of pitch oscillation, rad/s

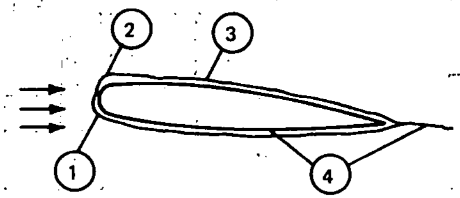
DESCRIPTION OF BASIC METHOD

The previously developed method for analyzing dynamic stall is outlined briefly below. Details can be found in Ref. 6.

When the flow is attached (Figure 1a), the flow elements represented are: (1) a laminar boundary layer extending from the stagnation point over the leading edge; (2) a leading-edge separation bubble (if separation occurs prior to transition); (3) a turbulent boundary layer from the reattachment point of the leading-edge bubble (or the transition point) to the trailing edge; and (4) a potential flow over the airfoil, including the effects of a vortical wake generated by the variation in time of the circulation about the airfoil. If the airfoil undergoes leading-edge stall (Figure 1b), the flow elements modelled are: (1) a laminar boundary layer to the point of separation; (2) a laminar constant-pressure shear layer to the point of transition; (3) a turbulent constant-pressure shear layer; (4) a turbulent pressure-recovery region; and (5) a potential flow over the airfoil and external to the viscous mixing region, again including a vortical wake. If trailing-edge stall occurs (Figure 1c), the flow elements represented are: (1) the laminar boundary layer; (2) the leading-edge bubble (if laminar separation occurs prior to transition); (3) the turbulent boundary layer; (4) a turbulent constant-pressure shear layer; (5) a turbulent pressure-recovery region; and (6) a potential flow with vortical wake. These elements were formulated as follows.

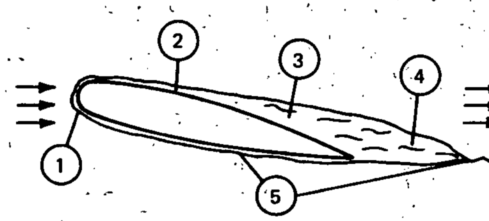
Potential Flow

Given the airfoil section characteristics and motions, together with the distribution of pressure in the dead-air region if the airfoil is stalled, the flow and pressure over the airfoil must be determined to compute the integrated load and analyze the boundary layer. The problem was formulated by imposing linearized boundary conditions of flow tangency and pressure, using a perturbation velocity potential derived from source and vortex distributions. The resulting coupled set of singular integral equations is solved by casting the singularity distributions in series form and solving for the unknown coefficients by imposing boundary conditions at prescribed points.



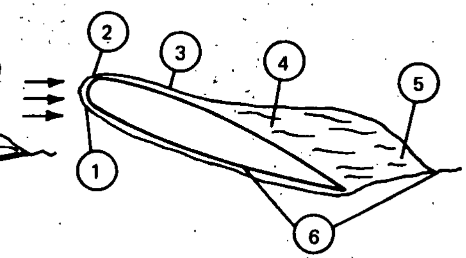
1. LAMINAR BOUNDARY LAYER
2. LEADING-EDGE BUBBLE
3. TURBULENT BOUNDARY LAYER
4. AIRFOIL AND VORTEX WAKE

a. ATTACHED FLOW



1. LAMINAR BOUNDARY LAYER
2. LAMINAR MIXING REGION
3. TURBULENT MIXING REGION
4. TURBULENT REATTACHMENT REGION
5. AIRFOIL AND VORTEX WAKE

b. LEADING-EDGE STALL



1. LAMINAR BOUNDARY LAYER
2. LEADING-EDGE BUBBLE
3. TURBULENT BOUNDARY LAYER
4. TURBULENT MIXING REGION
5. TURBULENT REATTACHMENT REGION
6. AIRFOIL AND VORTEX WAKE

c. TRAILING-EDGE STALL

Figure 1 FLOW ELEMENTS

Boundary Layer

Because the relative importance of the individual elements of the boundary layer flow as they affect dynamic stall could not be established in advance, the representation in Ref. 6 was made as general as possible. The method of finite differences for unsteady flow, with variable step size in both streamwise and normal directions, was employed with the error in each finite-difference approximation of second order. The Cebeci-Smith eddy-viscosity model (Ref. 7) is used to compute turbulent shear.

Dead-Air Region

The function of the model of the dead-air region is to define the streamwise distribution of pressure in that region, given the locations of the separation and recovery points and the pressure at the recovery point. The dead-air region is assumed to consist, in the most general case, of a laminar constant-pressure free shear layer from separation to transition, a turbulent constant-pressure mixing region, and a turbulent pressure-recovery region. The laminar shear layer is analyzed by the method of Ref. 8, assuming quasi-steady flow. The turbulent mixing and pressure-recovery regions are analyzed using the steady-flow momentum integral and first moment equations. Profile parameters in those regions are assumed to be universal functions of a dimensionless streamwise coordinate, with those functions derived from an exact viscous-inviscid interaction calculation. Matching of approximate solutions for the mixing and pressure-recovery regions at their interface completes the analysis.

Leading-Edge Bubble

The leading-edge bubble on an unstalled airfoil is analyzed using the same basic relations employed for the dead-air region. Given the boundary-layer parameters at separation, the length of the bubble and the amount of pressure rise possible, for that length, in the pressure recovery region, are computed. That pressure rise is compared with the rise in pressure in the potential flow over the length of the bubble. If the latter is greater than the former, the bubble is assumed to have burst, and the stall process is initiated.

Loading Calculation Procedure

Calculations proceed by forward integration in time, given the blade motions as a function of time. If, at a given instant, the airfoil is not stalled, the potential flow is computed, and the boundary layer and leading-edge bubble are analyzed to check for bubble bursting. If the airfoil is stalled, the pressure distribution in the dead-air region is computed, the potential flow evaluated, and the boundary layer is analyzed to locate the separation point. The last two steps are repeated iteratively until assumed and computed separation points agree. Rate of growth of the dead-air region is determined from an estimate of the rate of fluid entrainment derived from the potential-flow solution. In the case of leading-edge stall, unstall is determined by first postulating its occurrence and analyzing the leading-edge bubble which would then form to ascertain whether that event did in fact occur. During unstall, the dead-air region is washed off the airfoil. The rate of wash-off is normally taken to be the free-stream speed. There is some indication, though, that the rate should be considerably less than that value, as is discussed subsequently.

FORMULATION OF POTENTIAL FLOW TO SECOND ORDER

The following specifically concerns the derivation of the perturbation velocity potential to second order for unsteady, attached flow. The formulations derived are also used to compute flow and loading when the airfoil is stalled. However, in the latter case, while the solution is uniformly valid to first order, strictly consistent accounting of second-order terms was not attempted. A complete development to second order could not be justified without a corresponding refinement of the analysis of the dead-air region, which would be outside the scope of the present study.

The problem of an airfoil in unsteady flow, including nonlinear effects, has been treated by several investigators. Representative of these studies is the work of Giesing (Ref. 9), who used a finite-element approach to obtain numerical solutions for arbitrary transient motions, and Chen and Wirtz (Ref. 10), who employed an expansion of the acceleration potential to obtain integrated loading to second order for oscillatory pitching and plunging.

The approach taken here was dictated primarily by the requirement for compatibility with the formulation of the linearized problem in Ref. 6. The velocity potential and boundary conditions were systematically expanded, following the general procedure used in Ref. 11 for the steady problem, as follows.

The perturbation velocity potential can be written in the form

$$\phi(x, y, t) = \frac{1}{2\pi}\int_{-b}^{b} \sigma(\xi, t)\tan^{-1}\left(\frac{y}{x-\xi}\right)d\xi$$

$$-\frac{1}{2\pi}\int_{-b}^{x_o} \gamma(\xi, t)\ln\left(\sqrt{(x-\xi)^2 + y^2}\right)d\xi \qquad (1)$$

where coordinates (x, y) are fixed to the airfoil, with origin at midchord. The surface of the airfoil, of chord 2b, is located at $y = \pm T(x)$, $-b \leq x \leq b$ (consideration has been limited to symmetric airfoils). Coordinate x_o in Eq. (1) locates the end of the shed vortex wake.

Wake displacement effects have been omitted. Their contribution to the boundary conditions is of third order. While there is a second-order effect from wake displacement on the pressure at the airfoil surface, the contribution is symmetric and so does not affect integrated load. In any case, that term has reduced frequency as a factor, so for all practical purposes it can be regarded as third order.

The potential is required to satisfy

$$\frac{V_\infty + \partial \phi / \partial y}{U_\infty + \partial \phi / \partial x} = \pm T'(x) , \quad \left\{ \begin{array}{l} y = \pm T(x) \\ -b \leqslant x \leqslant b \end{array} \right. \qquad (2)$$

where U_∞ and V_∞ are apparent free-stream components, including effects of foil motions.

The source and vortex distribution strengths are taken to be sums of terms of ascending order in angle of attack or a similar small parameter, i.e.,

$$\sigma = \sigma_1 + \sigma_2 + \ldots$$

$$\gamma = \gamma_1 + \gamma_2 + \ldots$$

If the derivatives of ϕ in Eq. (2) are expanded in Taylor series about $y = 0$, terms of like order are assembled, and symmetric and antisymmetric contributions separated, in the usual manner, it results that

$$\left. \begin{array}{l} \sigma_1 = 2\, U\, T'(x) \\ \dfrac{1}{2\pi} \displaystyle\int_{-b}^{x_o} \dfrac{\gamma_1(\xi, t)\, d\xi}{x - \xi} = w(x, t) \end{array} \right\} \qquad (3)$$

$$
\left.
\begin{aligned}
\alpha_2 &= \frac{2U}{\pi} \frac{\partial}{\partial x} \int_{-b}^{b} \frac{T'(\xi)\, d\xi}{x - \xi} \\
\frac{1}{2\pi} \int_{-b}^{x_o} \frac{\gamma_2(\xi, t)\, d\xi}{x - \xi} &= - \frac{\partial}{\partial x} (T\,\gamma_1)
\end{aligned}
\right\} \qquad (4)
$$

where U is free-stream speed and w is the downwash imposed by foil incidence and motions.

The second of Eqs. (3) is, of course, the one solved in Ref. 6, using a Glauert-type trigonometric expansion of γ_1. Comparing Eqs. (3) and (4), it is seen that γ_2 can be computed using the same procedure as for γ_1, with $-(T\,\gamma_1)'$ substituted for w.

Some difficulty was encountered in implementing Eqs. (4), because T was originally formulated as a rather slowly converging trigonometric series. As a result, the derivative of $T\,\gamma_1$, in series form, did not converge satisfactorily. The difficulty was resolved by replacing the original series for T with the following more rapidly converging one:

$$
T = b \sin\theta \left[\frac{1}{2} (1 - \cos\theta) \frac{r_o}{b} + \sin\theta \sum_{n=1} t_n \sin n\theta \right]
$$

where r_o is leading-edge radius, and $x = b\cos\theta$.

The same basic procedure as the one developed by Lighthill for steady flow (Ref. 11) was followed in deriving the formula for the flow at the surface, q. The x-component of q, denoted q_x, is of the order of free-stream speed U, while the y-component, q_y, is of first order in whatever expansion parameter one chooses to use. Thus,

$$
\begin{aligned}
q &= q_x \sqrt{1 + (q_y/q_x)^2} \\
&= q_x + \frac{1}{2} \frac{q_y^2}{q_x} + \dots \\
&= q_x + \frac{1}{2} \frac{q_y^2}{U} + \text{higher order terms.}
\end{aligned}
\qquad (5)
$$

while

$$q_x = U \cos \theta_p \pm \frac{1}{2}(\gamma_1 + \gamma_2) + u_s$$

$$+ \frac{1}{2\pi} \int_{-b}^{b} \frac{(\sigma_1 + \sigma_2)\, d\xi}{x - \xi} + UTT'' + \text{third-order terms.}$$

$$q_y = U \sin \theta_p - \frac{1}{2\pi} \int_{-b}^{x_o} \frac{\gamma_1\, d\xi}{x - \xi} \pm v_s \pm UT'$$

$$+ \text{second-order terms.}$$

where θ_p is pitch angle and u_s and v_s are contributions from the source distribution representing the dead-air region when the airfoil is stalled (see Ref. 6). If these expressions for q_x and q_y are substituted in Eq. (5), a term which is singular at the leading edge due to a factor $(b + x)^{-1/2}$ is obtained, just as in the linear approximation. However, there is also a second-order term which is singular due to a factor $(b + x)^{-1}$, namely $U [TT'' + \frac{1}{2}(T')^2]$, which is approximately equal to $-.25\, Ur_o/(b + x)$ near the leading edge. A uniformly valid approximation for q is obtained, using Lighthill's procedure, by subtracting off the singular part of the offending term and multiplying by the factor

$$\left(\frac{b + x}{b + x + r_o/2}\right)^{\frac{1}{2}}$$

which restores the complete expression to its original form, to second order, some distance from the leading edge, and makes the result finite at the leading edge. The complete expression for q, uniformly valid to second order, with attached flow, is then

$$q = \left(\frac{b + x}{b + x + r_o/2}\right)^{\frac{1}{2}} \left\{ U \cos \theta_p \pm \frac{1}{2} (\gamma_1 + \gamma_2) + u_s + UTT'' \right.$$

$$+ \frac{1}{2\pi} \int_{-b}^{b} \frac{(\sigma_1 + \sigma_2)\, d\xi}{x - \xi}$$

$$+ \frac{1}{2U} \left[U \sin \theta_p - \int_{-b}^{x_o} \frac{\gamma_1\, d\xi}{x - \xi} \pm v_s \pm UT' \right]^2$$

$$\left. + \frac{U r_o}{4(b + x)} \right\}$$

The pressure coefficient on the airfoil is computed from

$$C_p(x, \pm T(x), t) = 1 - \left(\frac{q}{U}\right)^2 - \frac{1}{U^2} \frac{\partial}{\partial t} \phi(x, 0^{\pm}, t) \qquad (6)$$

The derivative of ϕ in Eq. (6) is computed by the same formulas as in Ref. 6, except that the second-order corrections are added to the source and vortex strenths. A term of second order which derives from the Taylor expansion of ϕ about y = 0 has been omitted from Eq. (6) for the same reasons wake displacement effects were neglected.

RESULTS OF COMPUTATIONS

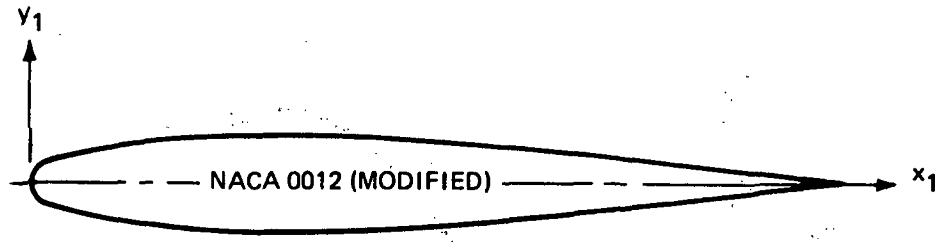
All calculations were performed for a modified NACA 0012 airfoil section. The section and a list of offsets are shown in Fig. 2. Unless otherwise noted, chordal Reynolds number was 2 million.

Preliminary Calculations

Computations were performed for transient pitching at a dimensionless rate $\dot{\theta}b/U$ of .025 and a chordal Reynolds number of 3 million, in order to determine whether quasi-steady flow could be assumed in the analysis of the laminar and turbulent boundary layers, and so effect a substantial savings in computer storage requirements and running time. Results for the laminar boundary layer are shown in Fig. 3, in which are plotted the external flow magnitude u_e, displacement thickness δ^* and wall shear τ_w as a function of distance along the airfoil surface from the stagnation point. Results obtained by omitting time derivatives are seen to be virtually identical to those using the complete boundary-layer equations.

It should be noted here that the point of vanishing wall shear is not generally coincident with the separation point in unsteady laminar flow. It was shown in Ref. 12, though, that a sufficient condition for identifying the separation point is that the boundary layer equations become invalid downstream of the true separation point. In the method of analyzing the boundary layer used here, there is a good indication of when that event occurs, since the solution is obtained by iteration on the wall shear. When that iteration diverges, then presumably there is no solution to the boundary layer equations and unsteady separation has occurred. It was found that, for the results of Fig. 3, the points of vanishing wall shear and separation are nonetheless coincident, for all practical purposes, since the iteration on wall shear diverged at the first streamwise station downstream of the point where the wall shear went to zero.

A slight difference between the quasi-steady and unsteady result was discernable near separation of the turbulent boundary layer downstream of reattachment of the leading-edge bubble, as shown in Fig. 4. However, the difference in separation points, 3.5 percent of chord, is less than the acceptable error of 5% used in the overall



x_1/c	Yu/c
0	0
.0110	.0170
.0220	.0230
.0330	.0270
.0540	.0340
.0760	.0390
.1087	.0445
.1521	.0493
.2065	.0527
.2500	.0542
.3043	.0547
.3478	.0541
.4130	.0520

x_1/c	Yu/c
0	0
.4564	.0499
.5000	.0472
.5434	.0439
.6086	.0383
.6521	.0343
.6955	.0300
.7607	.0230
.8042	.0181
.8477	.0127
.8911	.0070
.9346	.0011
1.000	.0011

ro/c = .0143

Figure 2 AIRFOIL SECTION

iteration for trailing-edge stall. The assumption of quasi-steady flow appears to be justified, then, at least for those flow conditions and airfoil motions normally experienced by a helicopter rotor blade. That assumption was therefore employed for all subsequent calculations.

Effect of Second-Order Terms

In order to assess the importance of the added terms in the potential flow model, a number of calculations were performed for direct comparison with results obtained using

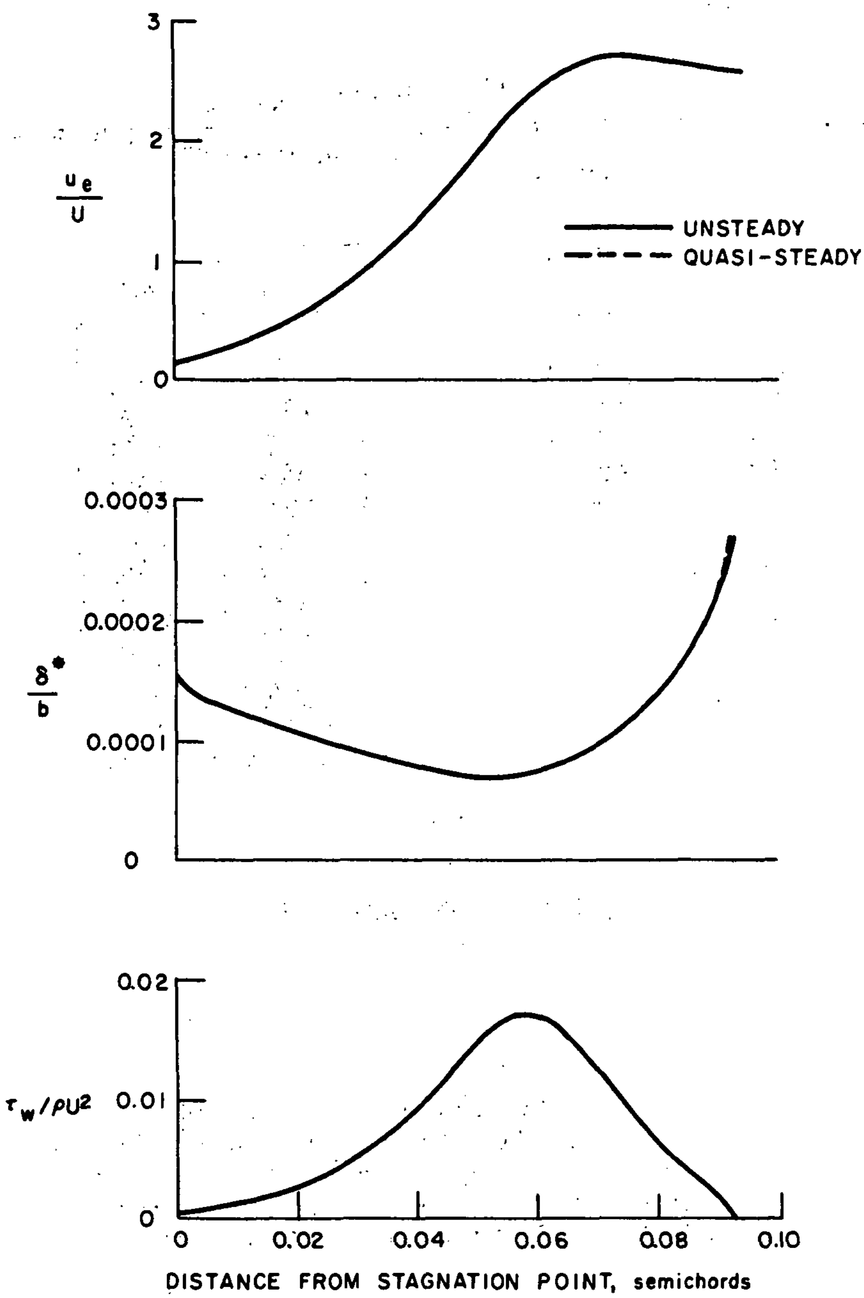


Figure 3 COMPARISON OF RESULTS OF LAMINAR BOUNDARY LAYER ANALYSES

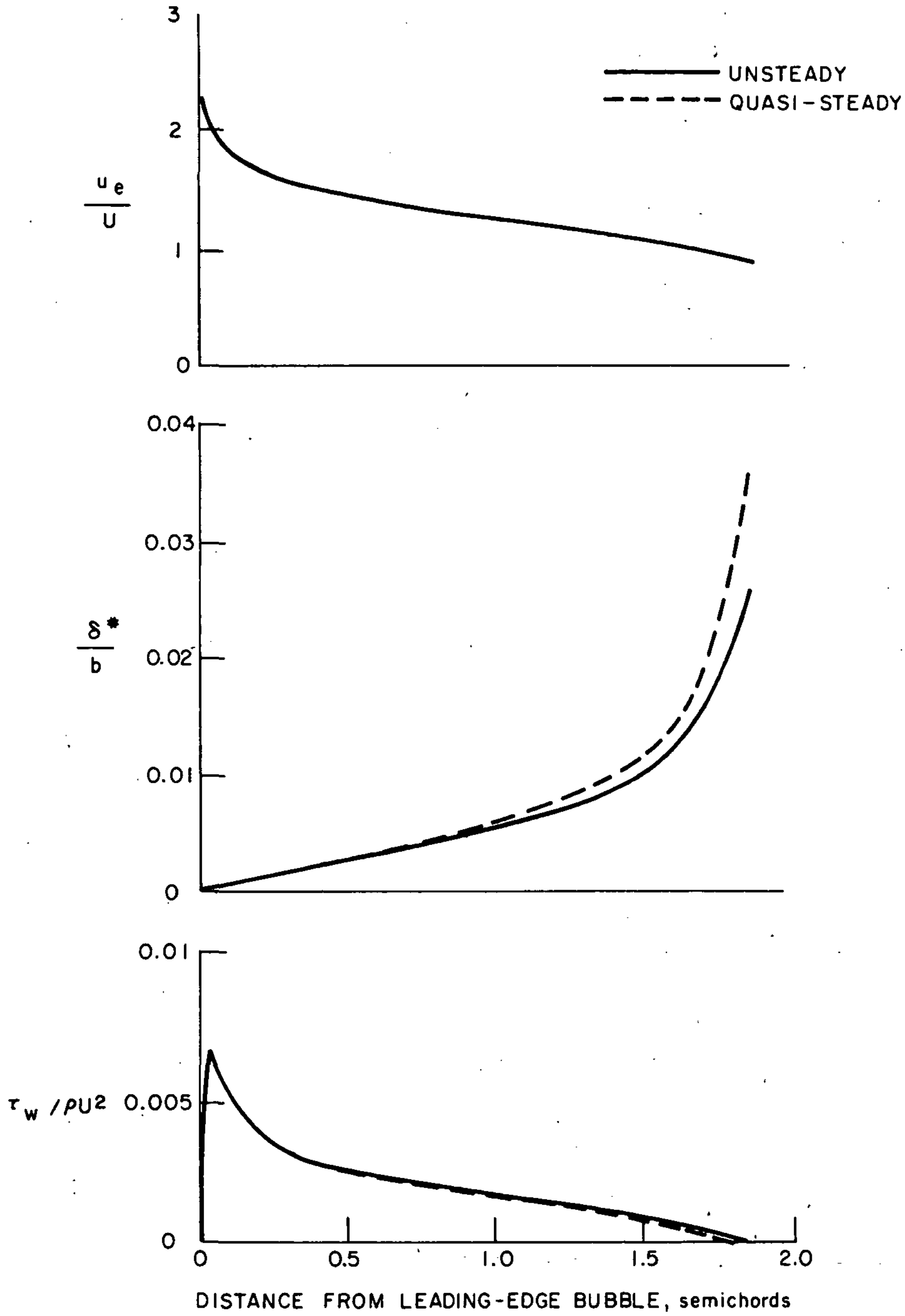


Figure 4 COMPARISON OF RESULTS OF TURBULENT BOUNDARY LAYER ANALYSES

a linearized potential-flow formulation. Cases of both transient and sinusoidal pitching motions were considered.

Time histories of normal force and moment coefficient and length of the dead-air region for a linearly varying pitch angle are shown in Fig. 5. The refined solution is seen to produce somewhat less dynamic overshoot of normal force than the linear formulation. However, the relative overshoot is about the same, the maximum static C_n for the linearized case being somewhat larger than when second-order terms are included.

A comparison of the variation of static normal-force and moment coefficients with angle of attack, generated by series of transient pitch calculations, is shown in Fig. 6. Including second-order terms is seen to reduce both the maximum C_n and the stall angle of attack, and increase the slope of the C_n curve below stall. The increase in $dC_n/d\alpha$ is due to nonlinear thickness effects. The theoretical value of the slope for the airfoil analyzed is 6.82 per radian, while the computed value is 6.76; the slight difference is believed to be due to numerical inaccuracies.

Including second-order terms significantly improved the representation of the pressure distribution. The predicted moment coefficient below stall using the nonlinear potential model is essentially zero, as it should be. The linear solution makes the center of pressure somewhat aft of the quarter-chord, the shift being due to the Lighthill correction which was used to remove the leading-edge singularity.

Results for sinusoidal pitching at a reduced frequency of .13 are compared in Fig. 7, where C_n and $C_{m\ c/4}$ are plotted against instantaneous pitch angle. The dashed curves are the static variations of the coefficients. The corresponding measured loading variation, from Ref. 2, is shown in Fig. 8. Including second-order terms is seen to cause some dynamic overshoot, while none had been obtained with the linear model. The overshoot is still considerably less than what was measured, however. The nonlinear terms are seen to cause unstall to be initiated at a pitch angle somewhat above the static stall angle, while the linear solution has unstall starting slightly below that angle.

Results for pitching at a reduced frequency of .26 are shown in Fig. 9. The comparable measured loading is shown in Fig. 10. The same effects of the second-order terms are evident at the higher frequency, but the relative differences are somewhat less.

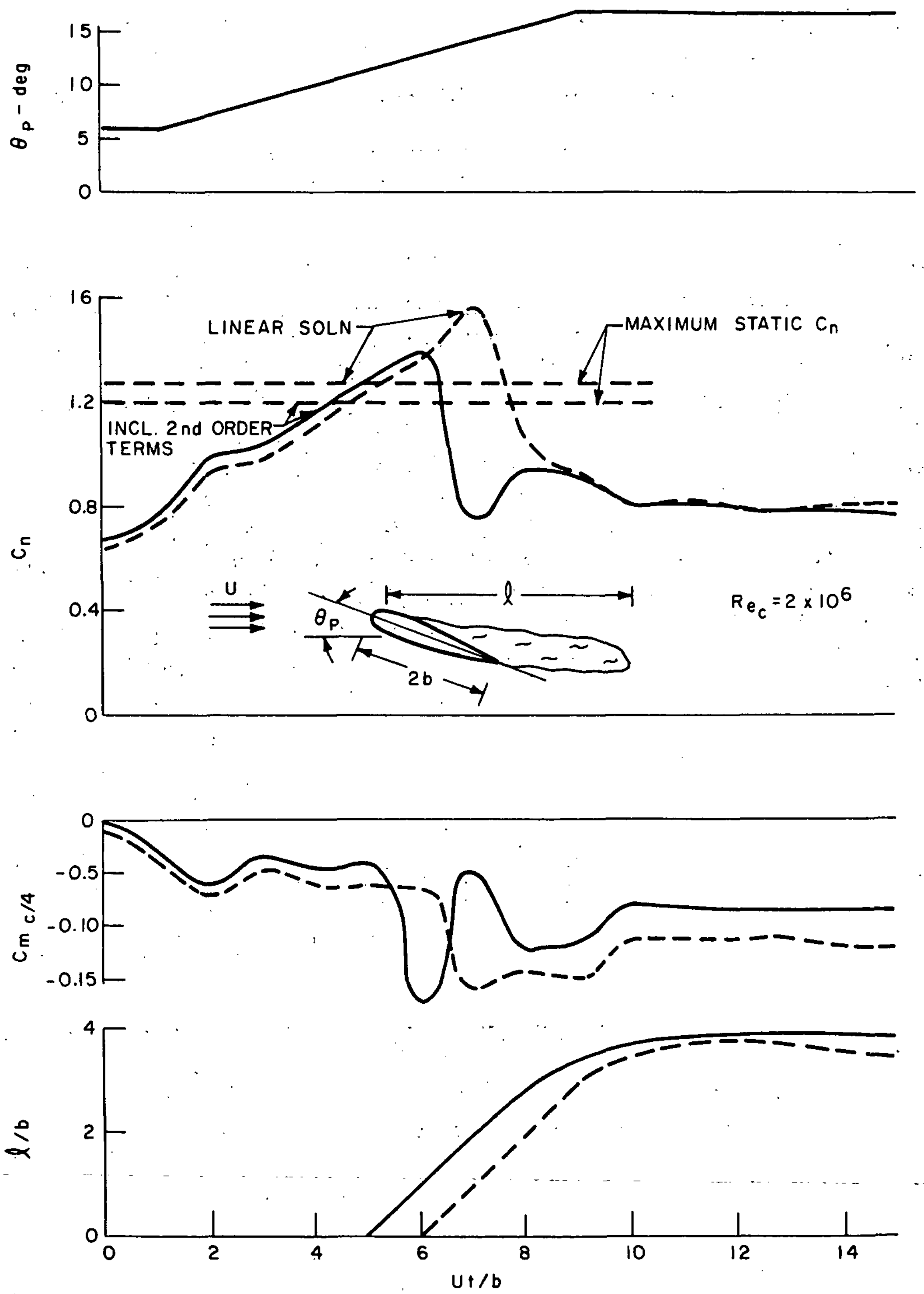


Figure 5 COMPARISON OF COMPUTED LOADING TIME HISTORIES DURING TRANSIENT PITCHING

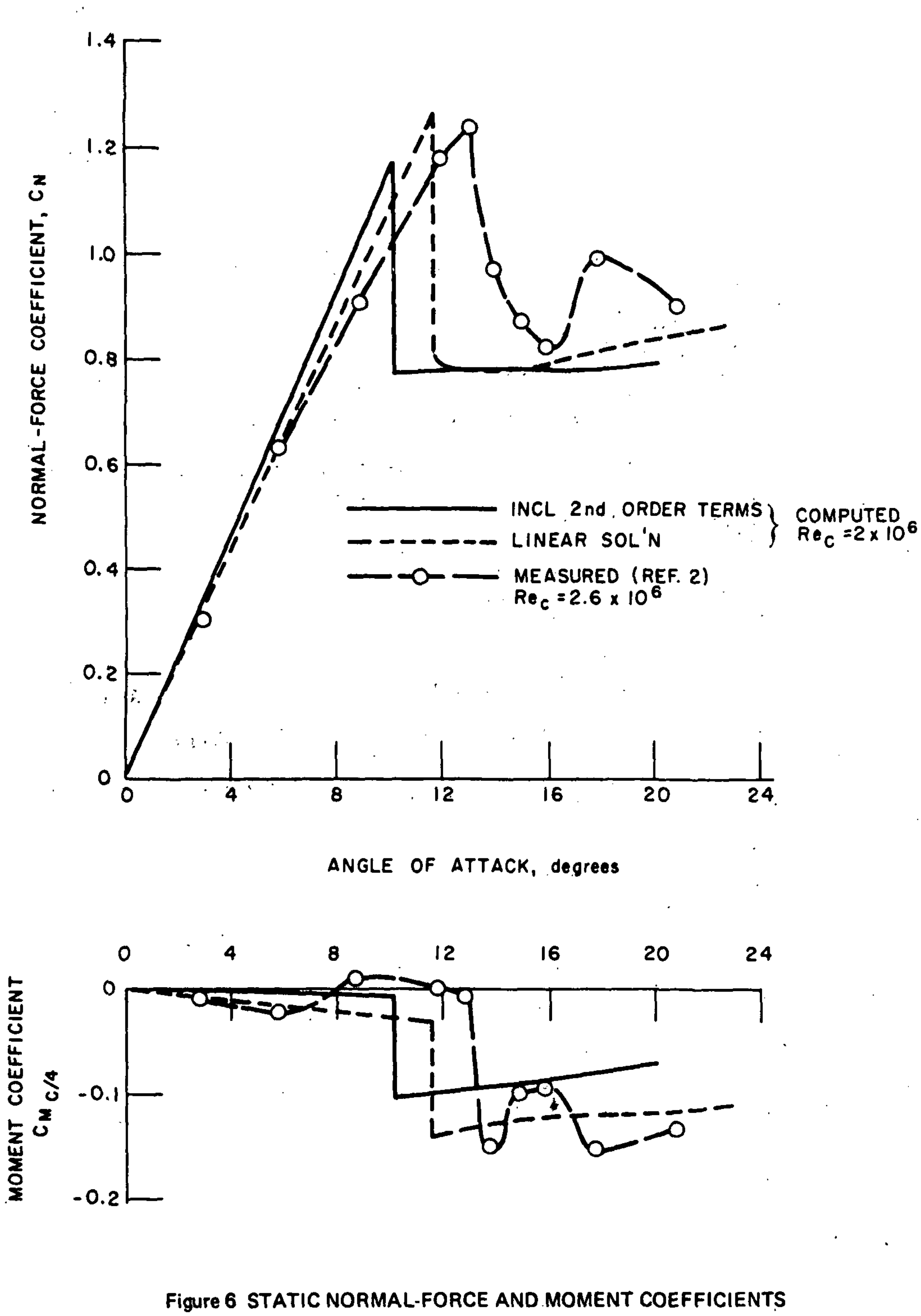


Figure 6 STATIC NORMAL-FORCE AND MOMENT COEFFICIENTS

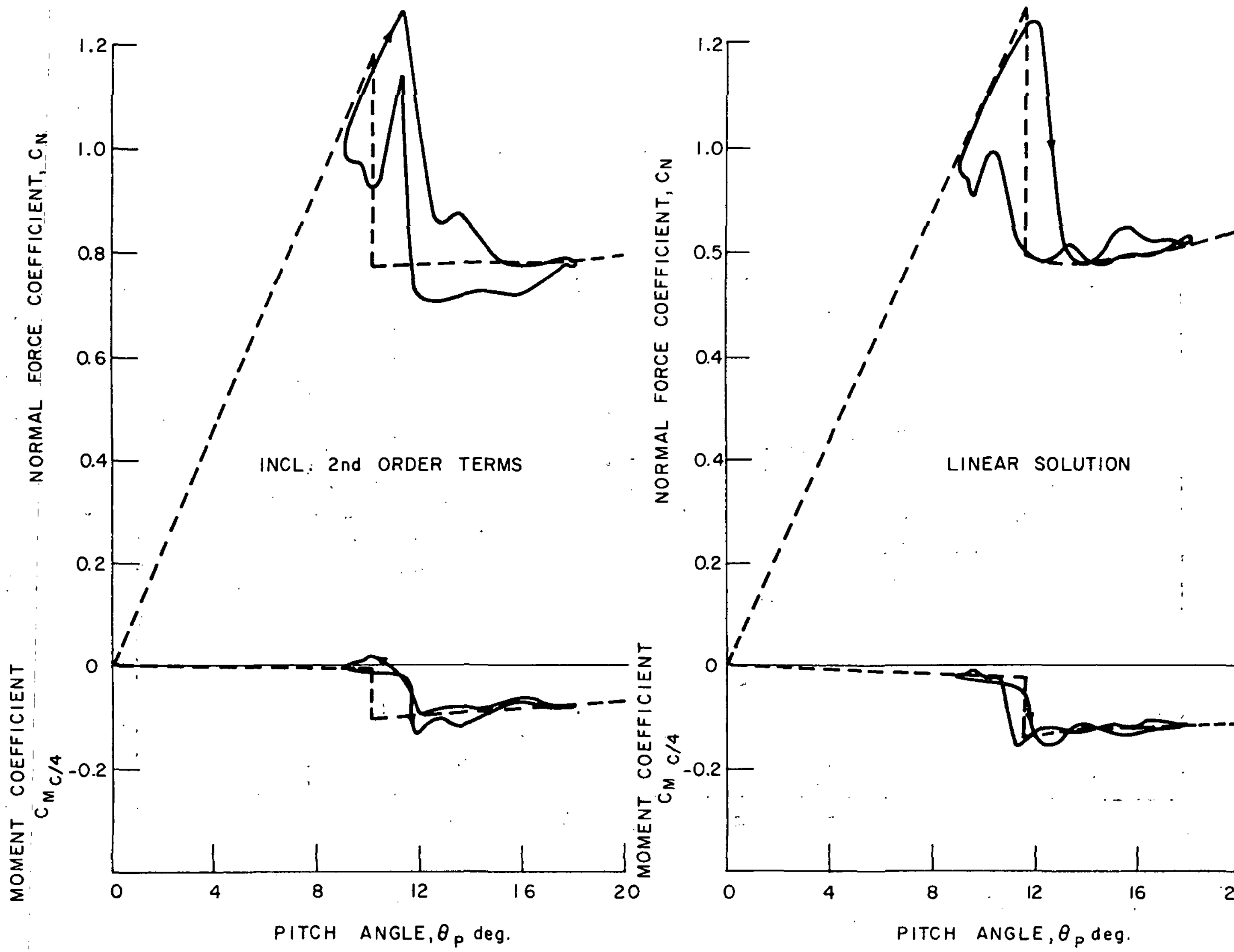


Figure 7 COMPARISON OF COMPUTED LOADINGS DURING SINUSOIDAL PITCHING, k = .13

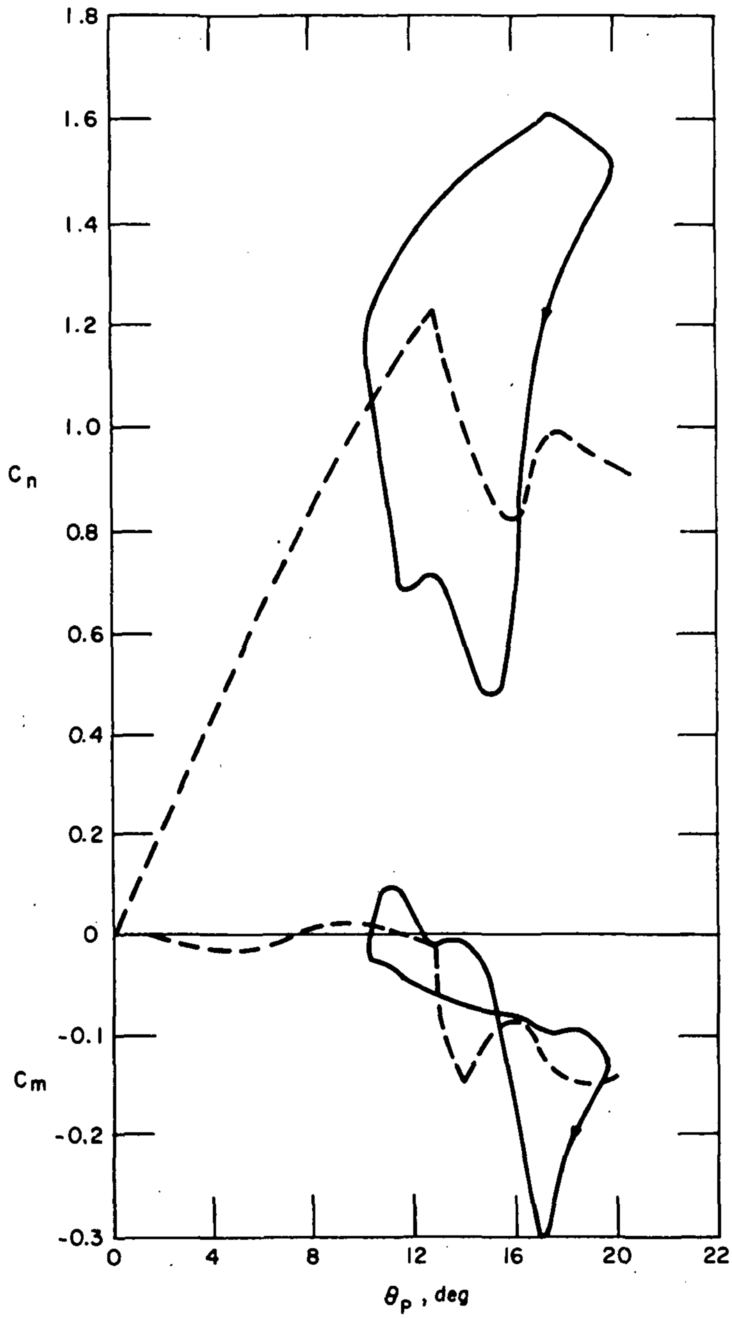


Figure 8 MEASURED LOADING WITH k = .126 (from Ref. 2)

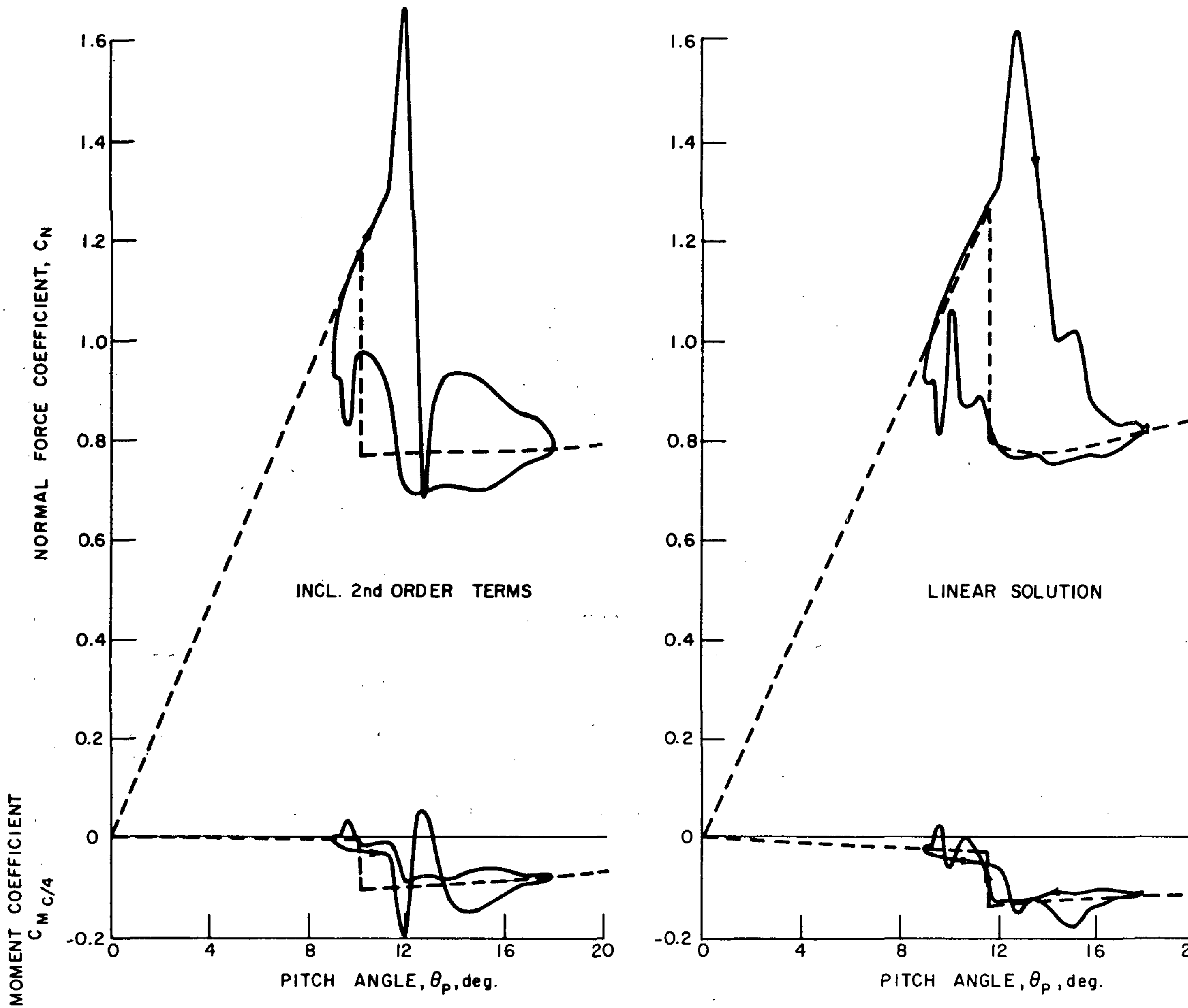


Figure 9 COMPARISON OF COMPUTED LOADINGS DURING SINUSOIDAL PITCHING, k = .26

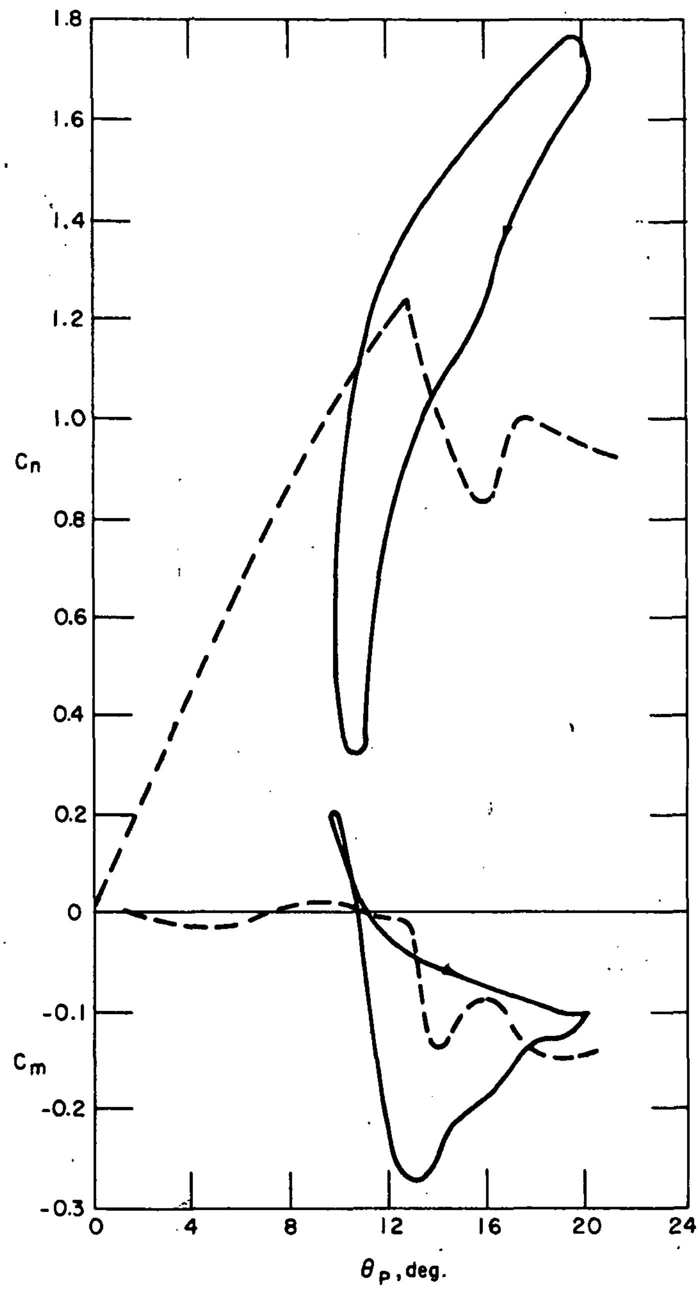


Figure 10 MEASURED LOADING WITH k = .252 (from Ref. 2)

While a nonlinear potential-flow model improves the predicted loading somewhat, substantial quantitative differences between computed and measured results are still evident. The analysis does serve to demonstrate that the errors in the original potential-flow model were not contributing to any great extent to those differences. Attention can now be directed to other factors in improving the method, as discussed in the next section.

Effect of Dead-Air Region Growth Rate

As was noted in the previous description of the basic method, rate of growth of the dead-air region is computed from an estimate of rate of entrainment obtained from the potential-flow solution. That estimate, which is proportional to the net source strength in the dead-air region, can only be reasonably justified when the length of the dead-air region is near its steady state value. However, a value for the growth rate is needed at the onset of leading-edge stall, when the region does not yet extend beyond the trailing edge, and is much less than its steady-state length of between two and three chords. It was beyond the scope of the original study to attempt a definitive analysis of this complex unsteady interaction of viscous and inviscid flows, so the growth rate was arbitrarily made equal to the free stream speed at the onset of leading-edge stall.

Subsequent calculations revealed, though, that one of the factors contributing to lift overshoot is an increment in lift induced on the aft portion of the airfoil when the dead-air region still terminates upstream of the trailing edge. This effect is discussed in more detail in Ref. 6. It would appear, then, that if the initial growth rate of the dead-air region were reduced, more overshoot would result.

A series of transient pitch calculations were carried out, parametrically varying initial growth rate ℓ_0, to verify that this is the case. The loading time histories obtained are shown in Fig. 11. As expected, the overshoot in C_n is increased substantially by reducing ℓ_0. The time to reach steady state is, of course, increased as well.

The greatest differences between computed and measured loading during oscillatory pitch are the amount of lift overshoot and the amounts of lift and moment hysteresis. It would appear, then, that considerable improvement in the agreement would result if ℓ_0 were reduced. The following arguments are put forth as justification for doing that.

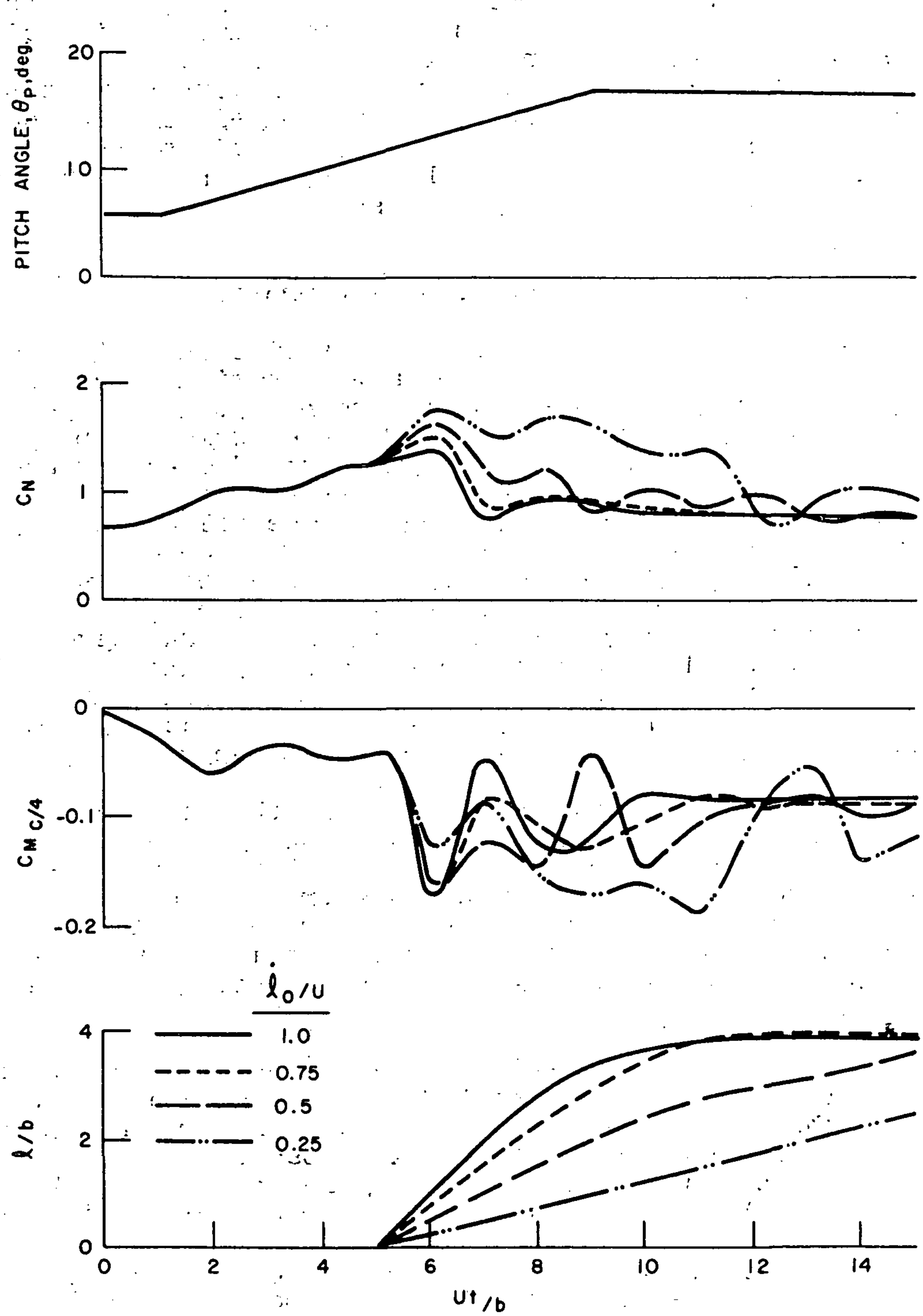
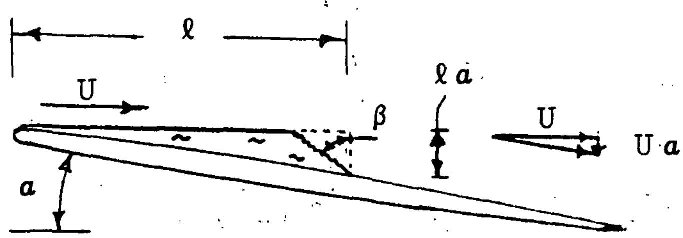


Figure 11 EFFECT OF $\dot{\ell}_0$ ON LOADING DURING TRANSIENT PITCHING

The dead-air region consists of fluid garnered from the free stream by the combined action of viscous shear (mostly turbulent, probably) and pressure gradients. Let it be assumed that this assimilation of fluid mass takes place primarily at the downstream end of the dead-air region, and that the fluid is trapped in the funnel-shaped region subtending an angle β, as sketched below (lengths and velocities shown are approximate, assuming angle of attack α is small).



Let f be the fraction of fluid entering the funnel which ultimately becomes part of the dead-air region. Then the rate of increase of the mass of trapped air is given, approximately, by

$$\dot{m} = f \left[\rho (\ell\alpha \tan \beta)(U \alpha) \right]$$

But $\dot{m}$ is proportional to the rate of increase of area; $\dot{m} \approx \rho \ell \dot{\ell} \alpha$. Equating the two rates, it follows that

$$\dot{\ell} \approx (f \tan \beta) U \alpha$$

Thus, if $f \tan \beta$ is a number of order one, or less, $\dot{\ell}_o$ is of the order of $U\alpha$, rather than U.

Convincing evidence that $\dot{\ell}_o$ must be substantially less than U was obtained by recomputing the loading for oscillatory pitching with k = .13, using a value of .25U for $\dot{\ell}_o$. Presumably the process of unstall, in which the free stream recaptures the fluid in the dead-air region, is similar to that of stall onset, so the rate of wash-off of the dead-air region during unstall was also reduced from U to .25U. The resulting curves of C_n and $C_{m\,c/4}$ vs. pitch angle are shown in Fig. 12. The computed variation of C_n

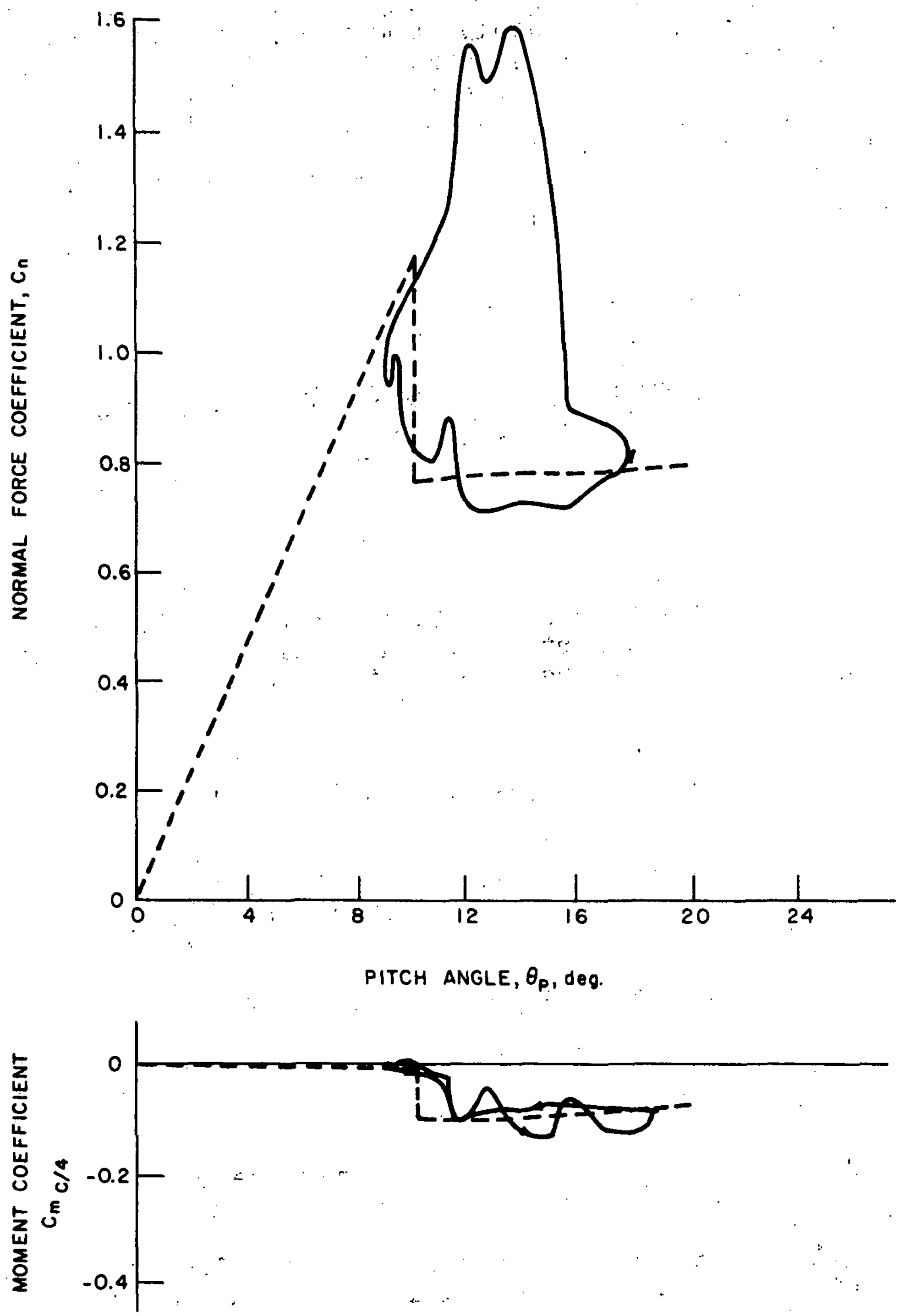


Figure 12 COMPUTED LOADING DURING SINUSOIDAL PITCHING
WITH ℓ_0 = .25 U AND k = .13

is now seen to be in quite good agreement with the measured loading, Fig. 8. Maximum normal force is predicted almost exactly, and the areas within the loops are roughly the same. The moment variations still differ appreciably, the main discrepancy being between when the airfoil starts pitching down and where unstall begins. There is apparently considerably more suction over the aft portion of the airfoil in the dead-air region than what is predicted. The assumption of quasi-steady flow in the dead-air region is the most likely cause for this difference.

The load during oscillatory pitching with $\dot{\ell}_o$ and wash-off rate of .25U was also computed for k = .26, with the results shown in Fig. 13. Again, the agreement with the measured loading, Fig. 10, is much improved, the peak C_n computed being 1.88 while the measured maximum is 1.77. The inadequacy in the model of the dead-air region with θ_p decreasing is also evident at the higher frequency.

It might be conjectured that the wash-off rate should be less than the initial growth rate of the dead-air region, since reentrainment presumably is dominated by viscous shear, rather than pressure gradients. Therefore, the effect of changing the ratio of $\dot{\ell}_o$ to wash-off rate was investigated by repeating the oscillatory pitching calculations using a wash-off rate of .1U, with $\dot{\ell}_o$ again being .25U. The results for k of .13 and .26 are shown in Figs. 14 and 15, respectively. The loading variation is seen to be quite insensitive to wash-off rate, regardless of the reduced frequency or the value of $\dot{\ell}_o$, so making it equal to $\dot{\ell}_o$ would appear to be a reasonable approximation.

Effect of Reynolds Number on Stall Characteristics

Stall characteristics of a given airfoil are necessarily a function of Reynolds number, because of the role played by the boundary layer in the stall process. Specifically, one would expect that an airfoil which is subject to leading-edge stall at a certain Reynolds number would undergo trailing-edge stall at a much higher Reynolds number, since transition would then preclude formation of a leading-edge bubble. At intermediate Reynolds numbers, presumably either type of stall could occur. Calculations were performed to investigate the effect of varying Reynolds number on dynamic and static stall characteristics in this intermediate range of Reynolds numbers.

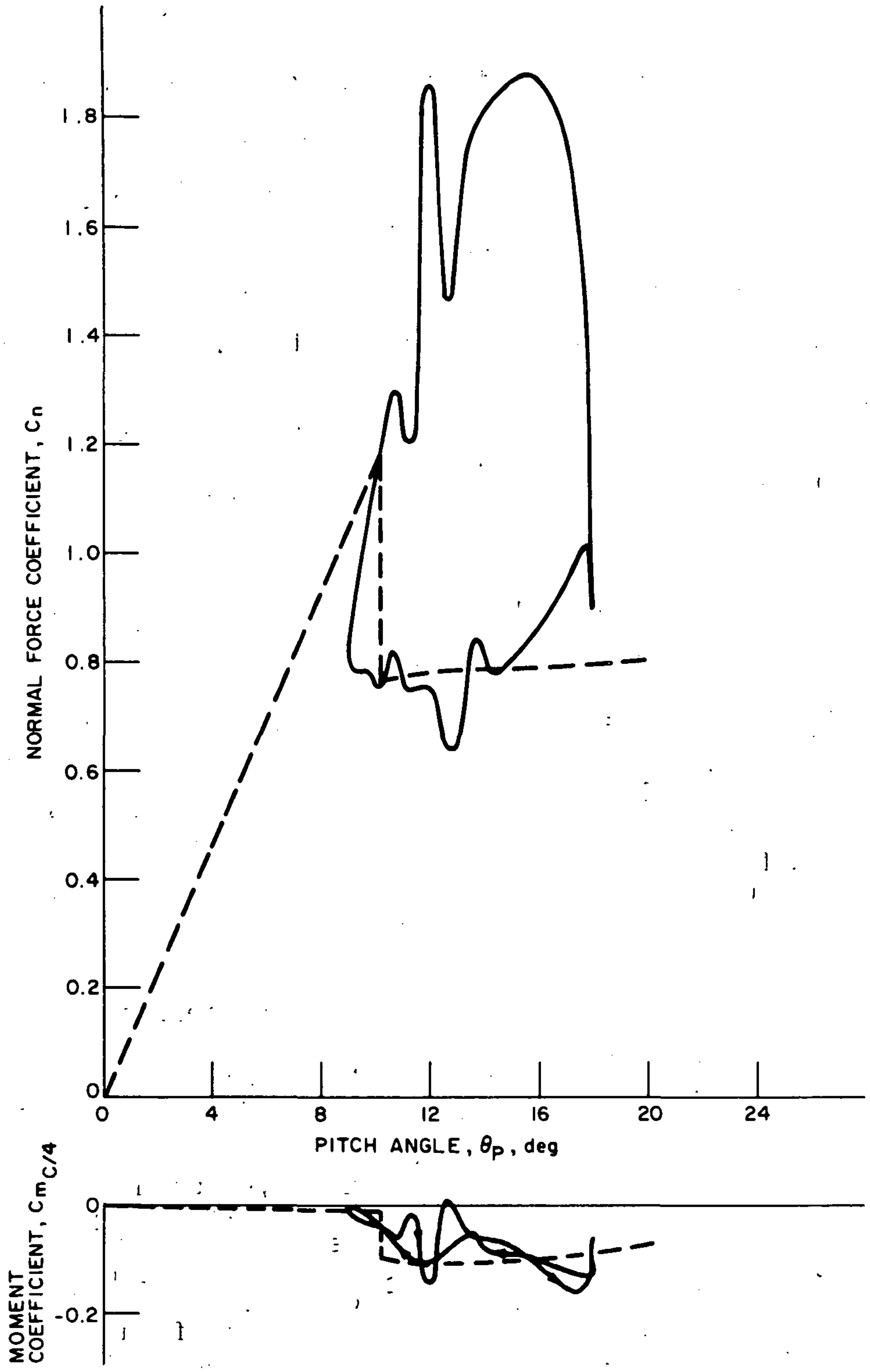


Figure 13 COMPUTED LOADING DURING SINUSOIDAL PITCHING WITH $\dot{\ell}_o = 0.25U$ AND $k = 0.26$

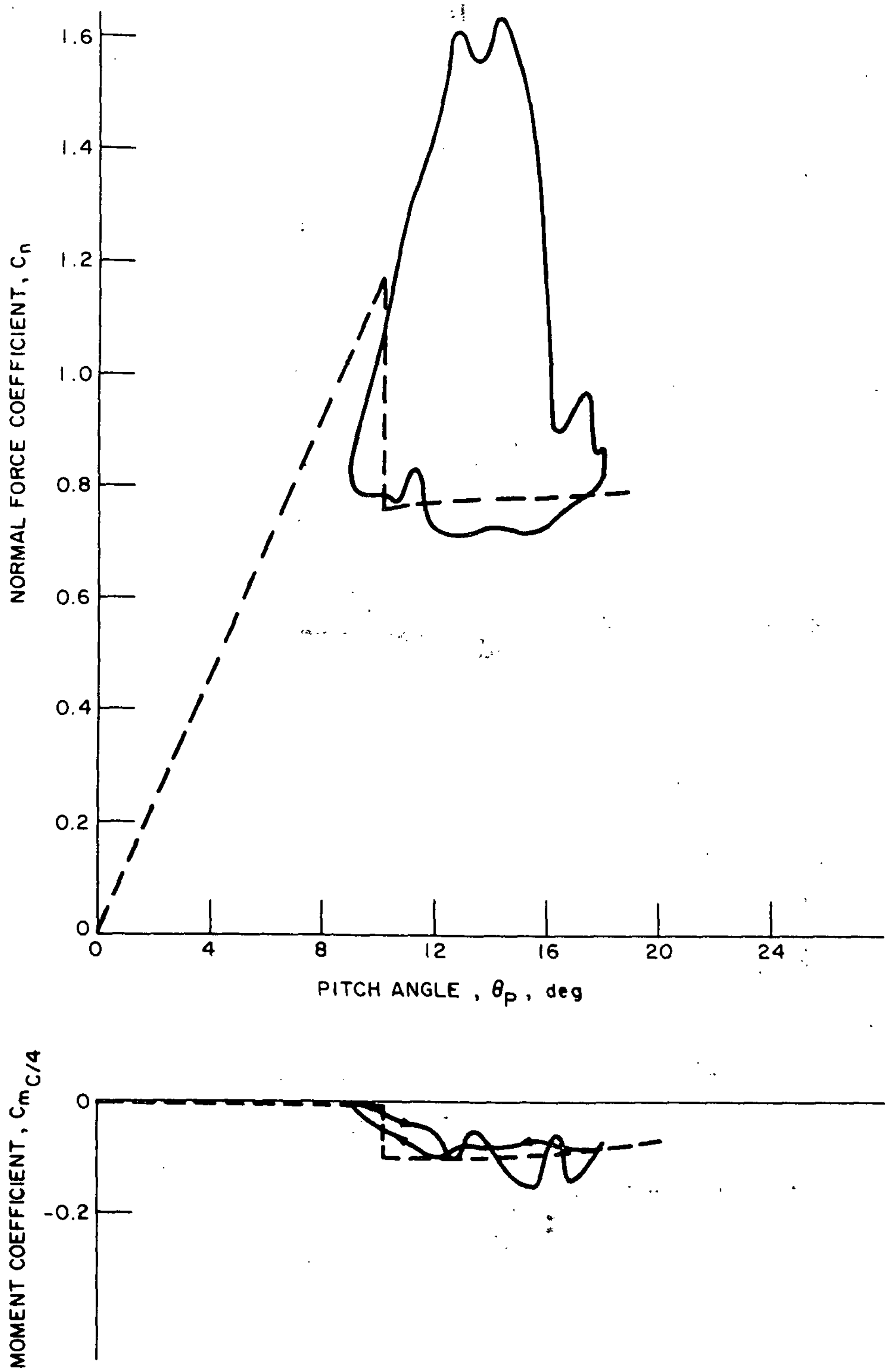


Figure 14 COMPUTED LOADING DURING SINUSOIDAL PITCHING WITH WASH-OFF RATE OF 0.1U, $\dot{\ell}_o$ = 0.25U, k = 0.13

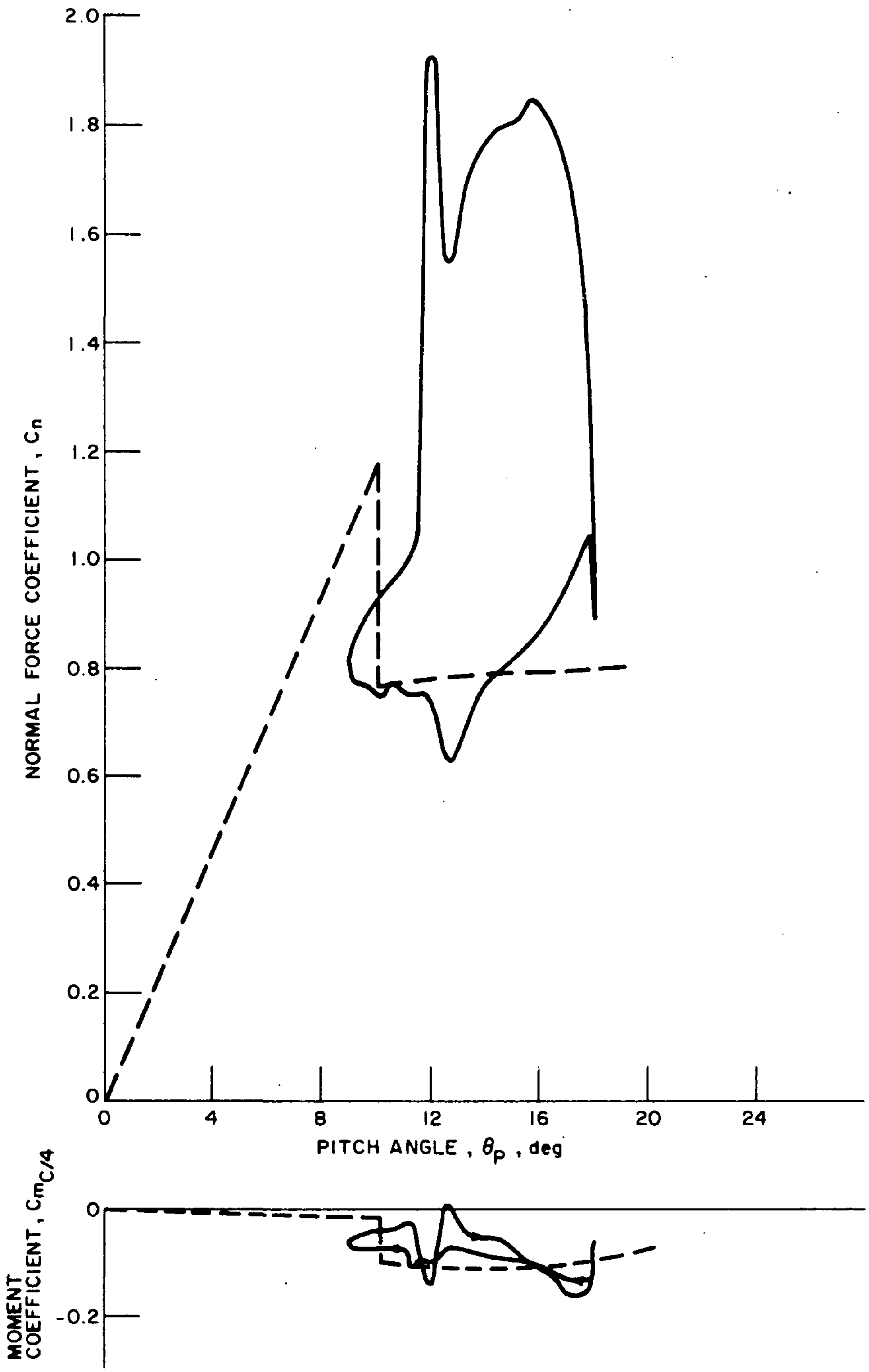


Figure 15 COMPUTED LOADING DURING SINUSOIDAL PITCHING WITH WASH-OFF RATE OF 0.1U, $\dot{\ell}_o$ = 0.25U, k = 0.26

The loading resulting from transient pitching through stall was analyzed for chordal Reynolds numbers of 3 and 6 million, for comparison with the previous results obtained at a Reynolds number of 2 million. Pitch angle was again varied linearly with time up to a prescribed value and then held constant.

Increasing Reynolds number to 3 million caused a marked increase in the resistance to bursting of the leading-edge bubble. The airfoil does not undergo leading-edge stall for pitch angles as high as about 16 degrees, but does experience trailing-edge stall between about 12 and 16 degrees. At a steady-state pitch angle of 15.8 degrees, separation of the turbulent boundary layer has progressed upstream to near the quarter-chord point.

An interesting phenomenon is encountered during pitching through higher angles. As the separation point of the turbulent boundary layer moves up the chord, the resistance of the leading-edge bubble to bursting continuously decreases, even though the circulation and loading on the airfoil are decreasing as well. The reason for this is that the separated region has relatively little effect on the flow in the immediate vicinity of the leading edge, even though it reduces the loading over the rest of the airfoil. At a sufficiently high incidence, the bubble bursts and leading-edge stall ensues. Results for a case in which both trailing-edge and leading-edge stall occur are shown in Fig. 16, where the loading and the separation point location x_s are plotted against time for pitching up to 18 degrees. Note that very little C_n overshoot is predicted in this case.

As expected, a further increase in Reynolds number to 6×10^6 increases the resistance to both leading-edge and trailing-edge stall. When the steady-state pitch angle is 15.8 degrees, the separation point is only about .1c from the trailing edge. At a slightly larger pitch angle, that point moves rapidly up the chord. Figure 17 shows the loading and x_s variations with time for pitching up to 16.3 degrees. The steady-state separation point is seen to be about .35c from the leading edge in this case. No C_n overshoot at all is predicted for this case.

Leading-edge stall ultimately occurs at high pitch angles for $Re_c = 6 \times 10^6$, as well, but the separation point of the turbulent boundary layer very nearly encroaches on the leading-edge bubble before the bubble bursts. The distance separating them is only about .01c.

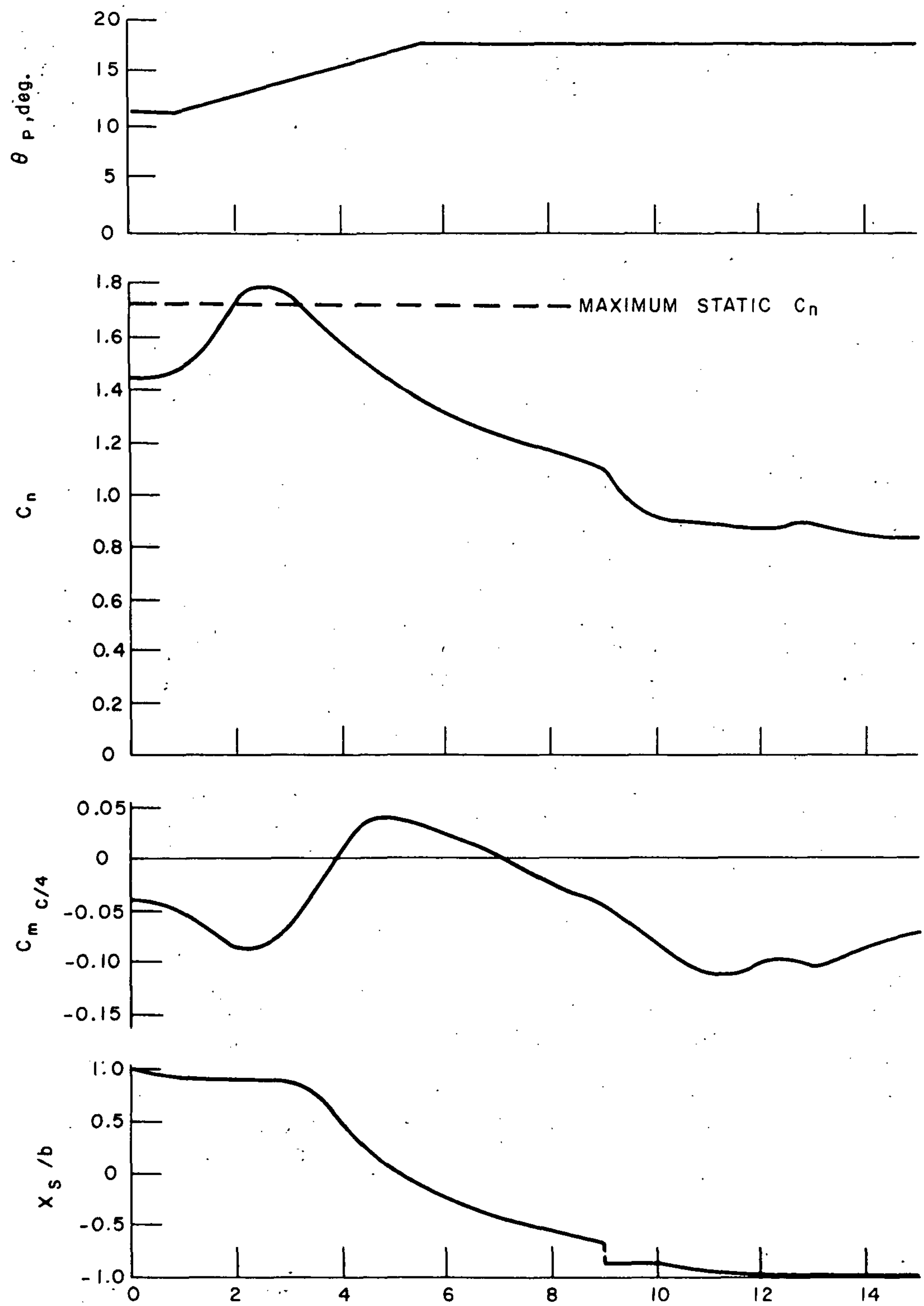


Figure 16 LOADING AND X_s VARIATION DURING TRANSIENT PITCH, $Re_c = 3 \times 10^6$

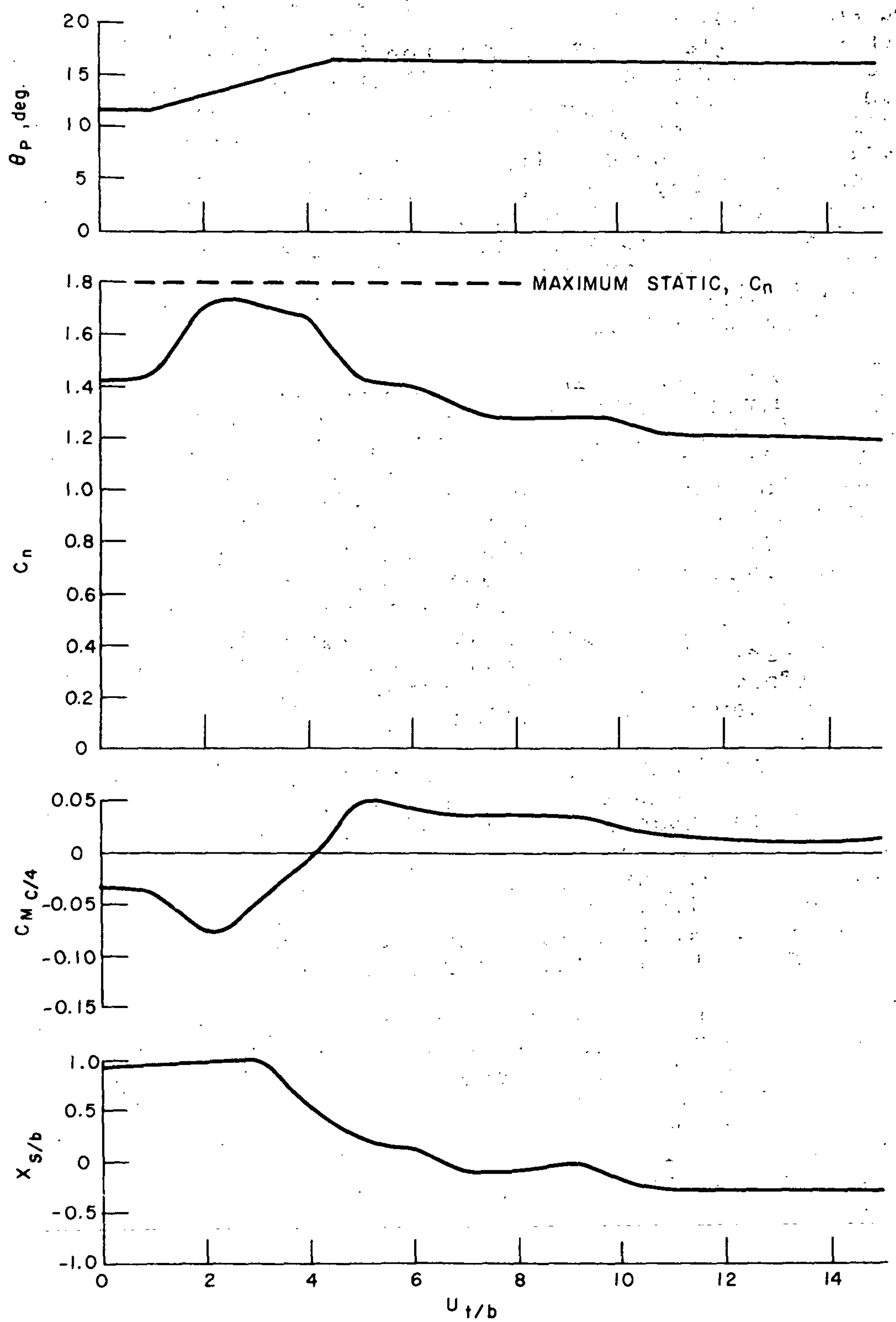


Figure 17 LOADING AND X_s VARIATION DURING TRANSIENT PITCH, $Re_c = 6 \times 10^6$

Results are summarized in Fig. 18, which shows the computed variation of static normal-force and moment coefficients with angle of attack for the three Reynolds numbers considered. A flagged symbol indicates that the airfoil was undergoing trailing-edge stall. Results of measurements for different Reynolds numbers for the section analyzed are not available. However, data in Ref. 13 show that a regular 0012 section at Reynolds numbers between 3 and 6 million has a maximum lift coefficient of about 1.6, generated at an angle of attack of 16 degrees, which agrees fairly well with the computed values of maximum C_n of 1.7 at $Re_c = 3 \times 10^6$ and 1.8 with $Re_c = 6 \times 10^6$, also occurring at about 16 degrees angle of attack.

The rapid falloff in normal force with angle of attack at higher Reynolds numbers is quite different from the behavior of thicker airfoils undergoing trailing-edge stall, the falloff in the latter case being more gradual (see Ref. 14). The reason for the sharp drop-off is apparently that the pressure rise is quite steep near the leading-edge, but relatively flat aft of midchord. Thus, the separation point moves rapidly forward, once incipient separation occurs, until it encounters the region of steep gradients near midchord (note the variation of x_s in Figs. 16 and 17). On thicker airfoils, the pressure increase along the chord is more uniform, allowing the separation point to stabilize at points closer to the trailing edge.

One case of sinusoidal pitching through stall was analyzed with $Re_c = 6 \times 10^6$ and $k = .13$. The C_n and $C_{m\ c/4}$ variations with pitch angle are shown in Fig. 19. Only trailing edge stall occurred during the cycle. As can be seen from Fig. 19, the normal force did not exceed the maximum static value. The moment variation exhibits a fairly large unstable (i.e., clockwise) loop, and the moment undergoes rather large positive excursions.

Since no direct determination of the type of stall has been made in the tests carried out to date, it is not clear whether the lack of lift overshoot in the calculations is symptomatic of trailing-edge stall or is an indication of an inadequacy in the analytic model. If the latter is the case, the most likely source of the problem would seem to be, again, the assumption of quasi-steady flow in the dead-air region.

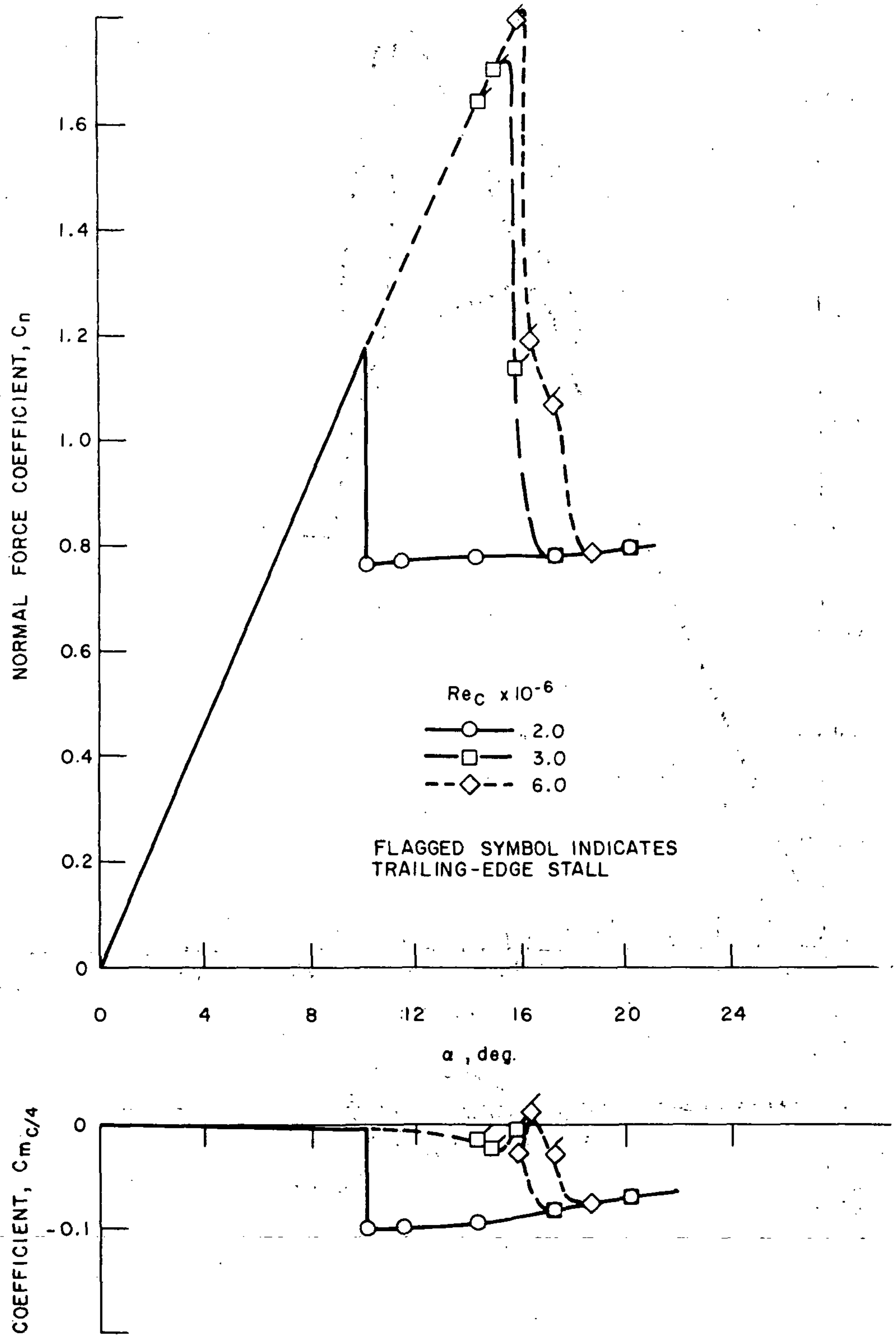


Figure 18 COMPUTED STATIC NORMAL-FORCE AND MOMENT COEFFICIENTS

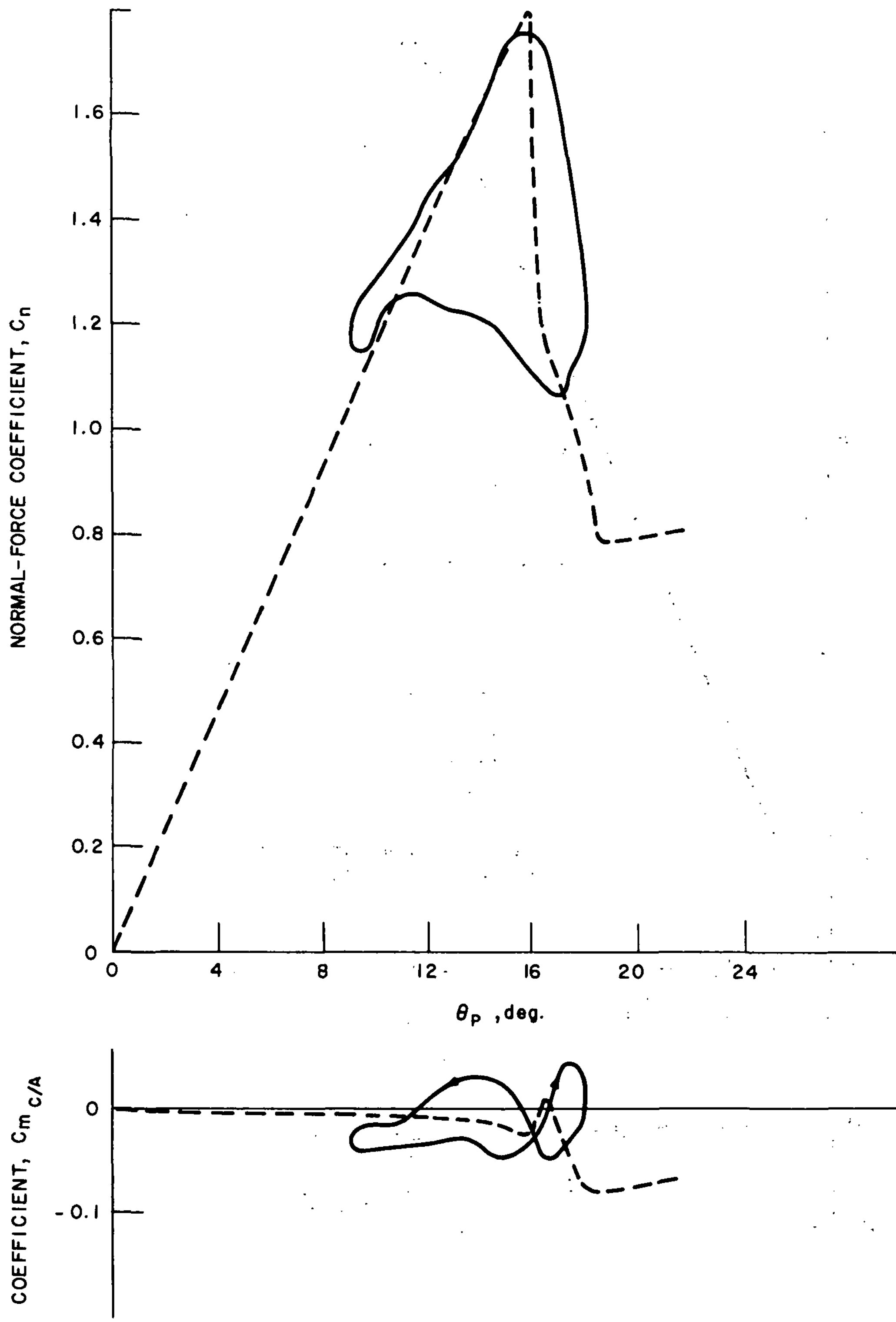


Figure 19 COMPUTED LOADING DURING SINUSOIDAL PITCHING,
k = .13, $Re_c = 6 \times 10^6$

CONCLUDING REMARKS

The representation of the potential flow in a previously developed method for analyzing dynamic stall has been refined by including second-order terms. The effects of growth rate of the dead-air region during leading-edge stall and the effect of Reynolds number on stall characteristics were also investigated. Results can be summarized as follows.

Including second-order terms improves the representation of the flow and chordwise pressure distribution on the airfoil below stall, but does not appreciably reduce the large quantitative differences between computed and measured results for sinusoidal pitching through stall.

The amount of lift overshoot during dynamic stall is a strong function of the rate of growth of the dead-air region at the onset of leading-edge stall. Much improved agreement between theory and test with a smaller growth rate indicates that the growth rate is of the order of the component of the free stream normal to the airfoil chord, but further analytical and experimental study of the stall onset process is needed. In contrast to the strong dependence on growth rate, the loading is quite insensitive to the rate at which the dead-air region is washed off during unstall.

The method predicts an increase in the resistance to both leading-edge and trailing-edge stall with increasing Reynolds number, as expected. Computations indicate that a given airfoil can undergo both trailing-edge and leading-edge stall under unsteady conditions at high Reynolds numbers, and that dynamic lift overshoot is much less when trailing-edge stall occurs. The latter effect may be due to the assumption of quasi-steady flow in the dead-air region.

APPENDIX

PROGRAM LISTING

APPENDIX

PROGRAM LISTING

A listing of the FORTRAN coding of the computer program follows. The program was written in FORTRAN IV for use on an IBM 360/75 computer.

```
C
C   PROGRAM TO ANALYZE UNSTEADY AIRFOIL STALL
C
      DIMENSION USAV(   1,70),U(   1,70,2),CMAT(50,50)
      DIMENSION RMAT(130), SCALE(300,2), V(100,2), UC(100,3), Y(100)
     X,  SCALS(300)
      DIMENSION ALAM(30),VZIP(30),FPRES(100),CAMBR(24),X(300)
      DOUBLE PRECISION CMAT, RMAT
      DIMENSION XGAM(30),XC(300),SBL(300),XBSIG(100)
      DIMENSION ASZ(30),AS(30,30),BS(30,30),ASHZ(30),ASH(30,30),BSH(30,3
     10),AR(30),ARH(30),UE(300,3)
      DIMENSION BLAM(30),FLAM(10),XFLAM(10)
      DIMENSION XU(30),YU(30),XL(30),YL(30),          VZ2(30),AR2(30)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XIW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      EQUIVALENCE                  (ASH(1),SCALS(1))
      DATA IN,MOUT,MD,       NTIME,      ISEPT,IWASH/5,6,250,   0,  0,2/
      DATA PI,TIME,      RENEL,USTOP/3.14159,0.,   6.20E4,3.0/
      DATA FPAST,FUP,VBUB /.1,.4,1./
      DATA VOFF /1./
      DATA  FLAM           /1.75,1.75,1.724,1.527,1.354,1.,.663,.452,.25
     14,.21/
      DATA  XFLAM           /-100.,-11.26,-7.01,-3.48,-1.766,0.,1.888,4.
     103,6.77,7.19/
      DATA NSBL,NZ,NY,            NCORD,LOWER,MSTOP,MOTR,NOTBL,INDV/
     1 150,100,70,        20,1,6,1,1,0/
      DATA ELSIG,FRZ,ARR,AMPLU,FREQU,ALPH1,ALPH2,HEAVE,AROT,FREQF,PHIH,
     1   RY1,DRY,Y(2),TEST,UPRIM,       ISEP/            .1,.06,3.5,0.,
     1 0.00,.1,0.,0.,-.5,0.,0.,1.001,.008,.01,.001,.0001,    0/
      NAMELIST /CASE/ NSBL,NZ,NY,MAXT,NGAM,NSIG,NOFF,NCORD,LOWER,MSTOP,
     1MOTR,NOTBL,INDV,ELSIG,DXI,REB,RDBB,FRZ,ARR,AMPLU,FREQU,ALPH1,ALPH2
     1,HEAVE,AROT,FREQF,PHIH,    RY1,DRY,    Y,TEST,UPRIM,XU,XL,YU,YL,NF,
     1NVOR,SVOR,HVOR,BARG,X1VOR,EMI,TORF,SSPA,CMPA,CMPAS,USTOP,RENEL,
     1 FPAST,FUP,VBUB,MD,VOFF
      NGAM=24
      NSIG=24
      NF=20
      UINF=1.
      NWAKE=999
      DXI=1.E10
      READ(IN,CASE)
      MX=NSBL+NZ-1
      NDIMC=50
      INDV=INDV+1
      RY=RY1
C
C   NOTE - OFFSETS ARE PUT IN AS LISTED IN  THEORY OF WING SECTIONS,  I.E
C          AS A FRACTION OF TOTAL CHORD, X1 BEING MEASURED FROM THE
C          LEADING EDGE.  BE SURE NF IS AN EVEN NUMBER.
C
      WRITE(MOUT,6)
      WRITE(MOUT,CASE)
      GO TO (343,342), INDV
```

```
342  WRITE(MOUT,25) NVOR,SVOR,HVOR,BARG,X1VOR,EMI,TORF,SSPA
     HVOR=HVOR**2
     BARG=BARG/6.2832
343  RDBC = RDBB / 2.
     CALL SECT(XU,YU,XL,YL,NOFF,NF,RDBC,           THICK,CAMBR)
     RDBB = RDBC * 2.
     DO 7875 N = 1, NF
     CAMBR(N) =  2 * CAMBR(N)
7875 THICK(N) = 2. *  THICK(N)
     CR=SQRT(RDBB)/2.
     CALL THCALC(CR,THICK,CRHAT,THAT,NF)
     WRITE(MOUT,4)
     WRITE(MOUT,7) AMPLU,FREQU,ALPH1,ALPH2,HEAVE,AROT,FREQF,RDBB,REB
     WRITE(MOUT,8)
     WRITE(MOUT,9) (N,CAMBR(N),THICK(N),THAT(N),N=1,NF)
     WRITE(MOUT,101) CR,CRHAT
     CALL SCAL(SBL,NSBL,FRZ,ARR,RDBB)
     CALL CORDX(NSBL,NZ,RDBB,SBL,X,XC)
     UTU=CR
     DO 350 N=1,NF
350  UTU=UTU+THICK(N)
     UTU=2.*UTU
     DO 2420 M=1,MX
     IF(XC(M)-1.) 2420,2419,2419
2419 MEND=M-1
     GO TO 2421
2420 CONTINUE
2421 MX=MEND
     XATT = -100.
     MXM1=MX-1
     UE(MX+1,1)=1.
     EPSLE=2.*(X(NZ)-X(NZ-1))
     EPSTE=X(MX)-X(MX-1)
     ALTC=11.50E4/SQRT(REB)
     IF(MD.EQ.1) GO TO 2423
     DO 2422 M=1,MX
     SCALE(M,1)=0.
     SCALE(M,2)=0.
     DO 2422 N=1,NY
     U(M,N,1)=0.
2422 U(M,N,2)=0.
2423 NSIGA=NSIG
     NSIGB=NSIG
     NSIG1=NSIG+1
     MOTR=MOTR+1
     PITCH = ALPH1
     IF( INDV + MOTR .LE. 2) PITCH = PITCH - ALPH2
     UN=SIN(PITCH)
     UT=COS(PITCH)
     NOTBL=NOTBL+1
     XMAX=1.-ELSIG
     CONA=.375*PI/DX1
     ANGS=PI/FLOAT(NSIG)
     CALL SETSX(NSIG1,1.1,2.,XSIG,ANGS)
     XSEP=1.1
```

```
      DO 2430 N=1,NSIG1
      XSIGB(N)=XSIG(N)
 2430 XSIGA(N)=XSIG(N)
      DO 2431 N=1,NSIG
      DO 2431 NU=1,3
 2431 BCAP(N,NU)=0.
      PINT=2./FLOAT(NCORD)
      NCP1=NCORD+1
      THXI=1.5/DXI
      NGP1=NGAM+1
      NWM1=NWAKE-1
      COUNT=0.
      DO 8456 N=1,NWAKE
      GAMAW(N)=0.
      GAMAW2(N) = 0.
      XIW(N)=1.+COUNT
 8456 COUNT=COUNT+DXI
      ANGLE=PI/FLOAT(NGAM)
      COUNT=0.
      DO 1002 M=1,NGP1
      PHIM=COUNT*ANGLE
      XGAM(M)=COS(PHIM)
      DOUNT=2.
      DO 1001 N=2,NGAM
      AS(M,N)=COS(DOUNT*PHIM)
 1001 DOUNT=DOUNT+1.
 1002 COUNT=COUNT+1.
      CALL WASH(XGAM,NGAM,TIME,ALPH1,ALPH2,HEAVE,AROT,FREQF,PHIH,UINF,CA
     1MBR,NF,VZIP,1,1)
      IF(INDV.EQ.2) CALL VWASH(BARG,HVOR,SVOR,NVOR,X1VOR,UINF,VZIP,XGAM,
     1NGP1,DXI)
      DO 8458 M=1,NGP1
      CMAT(M,1)=1.
      RMAT(M)=2.*VZIP(M)
      CMAT(M,2)=XGAM(M)
      DO 8457 N=3,NGP1
 8457 CMAT(M,N)=AS(M,N-1)
 8458 CONTINUE
      CALL ALSOL(NGP1, CMAT, RMAT, NDIMC)
      DO 8459 N=1,NGP1
      ACAP(N,1)=RMAT(N)
 8459 ACAP(N,2)=ACAP(N,1)
      CALL WASH2(VZ2,XGAM,NGAM,CR,THICK,NF,ACAP,0.,0,0.,2.)
      DO 8470 M = 1, NGP1
      RMAT(M)=    VZ2(M)
      CMAT(M,1) = 1.
      CMAT(M,2) = XGAM(M)
      DO 8470 N = 3, NGP1
 8470 CMAT(M, N) = AS(M, N - 1)
      CALL ALSOL(NGP1,CMAT,RMAT,NDIMC)
      DO 8471 M = 1, NGP1
      ACAP2(M,1)=RMAT(M)
 8471 ACAP2(M, 2) = ACAP2(M, 1)
      DO 2784 M=1,MX
      SIGN=1.
```

```
      IF(M-NZ) 2774,2775,2775
 2774 SIGN=-SIGN
 2775 CALL OECAL(UE(M,1),XC(M),SIGN,0)
 2784 JE(M,2)=UE(M,1)
      DO 1004 M=2,NGAM
 1004 BLAM(M)=(1.125*XGAM(M)+.1875*(1.+XGAM(M))*(1.-3.*XGAM(M))*ALOG((1.
     1+XGAM(M))/(1.-XGAM(M))))/DXI
      BLAM(NGP1)=-1.125/DXI
C
C  INDEXING IN TIME IS CARRIED OUT AT THIS POINT.
C
 9999 CONTINJE
      GO TO (196,195),MOTR
 195  READ(IN,2) ALPH1,ALPH2,HEAVE
C
C  NOTE - FOR READ-IN OF FOIL MOTIONS, MAKE ALPH1 = ALPHA,
C         ALPH2 = ALPHA-DOT, AND HEAVE = H-DOT.
C
 196  GO TO (198,197), INDV
 197  CALL ELPIT(ALPH1,ALPH2,EMI,TORF,SSPA,UINF,DXI,CMPA,CMPAS)
      X1VOR=X1VOR+DXI*UINF
 198  NITS=1
      TIME=TIME+DXI
      PITCH = ALPH1
      IF(INDV + MOTR .LE. 2) PITCH = PITCH - ALPH2*COS(FREQF*TIME)
      NTIME=NTIME+1
      NWAKE=NTIME+2
      IF(NWAKE-998) 202,201,201
 201  NWAKE=998
 202  IF(MAXT-NTIME) 8989,8800,8800
 8800 SAVEU=JINF
      UINF=1.+AMPLU*SIN(FREQU*TIME)
      UBUB=UINF*VBUB
      UN=UINF*SIN(PITCH)
      UT=UINF*COS(PITCH)
      UDOT=FREQU*AMPLU*COS(FREQU*TIME)
      STEPX=.5*DXI*(UINF+SAVEU)
      DO 1003 J=2,NWAKE
      JC=NWAKE-J+2
      GAMAW(JC)=GAMAW(JC-1)
      GAMAW2(JC) = GAMAW2(JC-1)
 1003 XIW(JC)=XIW(JC-1)+STEPX
      IF(ISEP) 2009,2009,2007
 2007 DO 2008 N=1,NSIG
      BCAP(N,3)=BCAP(N,2)
 2008 BCAP(N,2)=BCAP(N,1)
      DO 4433 N=1,NSIG1
      XSIGB(N)=XSIGA(N)
 4433 XSIGA(N)=XSIG(N)
      PRECS=PREC
      GO TO 2010
 2009 DEADL=0.
      ELDOT=UBUB
 2010 DO 1014 M=1,MX
      UE(M,3)=UE(M,2)
```

```
1014 UE(M,2)=UE(M,1)
     DEAD1=DEADL
     ELD1=ELDOT
     ALAM(1)=(1.125+.75*ALOG(STEPX*.5))/DXI
     DO 1005 M=2,NGP1
1005 ALAM(M)=BLAM(M)+.75*(1.+(1.-XGAM(M))/STEPX)*ALOG((1.+STEPX-XGAM(M)
    1)/(1.-XGAM(M)))/DXI
     DO 2006 M=1,NGP1
     ACAP2(M,3) = ACAP2(M,2)
     ACAP2(M,2) = ACAP2(M,1)
     ACAP(M,3)=ACAP(M,2)
2006 ACAP(M,2)=ACAP(M,1)
     AFACT=8.*(ACAP(1,2)+.5*ACAP(2,2))-2.*(ACAP(1,3)+.5*ACAP(2,3))
     AFAC2 =8.*(ACAP2(1,2)+.5*ACAP2(2,2))-2.*(ACAP2(1,3)+.5*ACAP2(2,3))
     ALPHS=VZIP(1)
     CALL WASH(XGAM,NGAM,TIME,ALPH1,ALPH2,HEAVE,AROT,FREQF,PHIH,UINF,CA
    1MBR,NF,VZIP,MOTR,INDV)
     IF(INDV.EQ.2) CALL VWASH(BARG,HVOR,SVOR,NVOR,X1VOR,UINF,VZIP,XGAM,
    1NGP1,DXI)
     DO 1006 M=1,NGP1
     ASZ(M)=1.+2.*ALAM(M)
     AS(M,1)=XGAM(M)+ALAM(M)
     SUM=0.
     SUM2=0.
     NWM1=NWAKE-1
     DO 4343 J=2,NWM1
     JP=J+1
     DLX=XIW(J)-XGAM(M)
     COX=DLX/(XIW(JP)-XIW(J))
     ALX=ALOG((XIW(JP)-XGAM(M))/DLX)
     SUM=SUM+(GAMAW(J)+(GAMAW(J)-GAMAW(JP))*COX)*ALX
4343 SUM2=SUM2+(GAMAW2(J)+(GAMAW2(J)-GAMAW2(JP))*COX)*ALX
     ELX=1.-XGAM(M)
     IF(M.EQ.1) ELX=1.
     ALX=(1.-XGAM(M))*ALOG(1.+STEPX/ELX)/STEPX
     AR2(M)=ALAM(M)*AFAC2/3.+(SUM2-GAMAW2(2)*ALX)/PI
1006 AR(M)=2.*VZIP(M)+ALAM(M)*AFACT/3.+(SUM-GAMAW(2)*ALX)/PI
C
C  THE FOLLOWING CALCULATIONS, THROUGH STATEMENT 4444, ARE PERFORMED
C   ONLY IF THE AIRFOIL IS STALLED.  THE AIRFOIL IS DESIGNATED TO BE
C   STALLED IF INTEGER ISEP IS NONZERO.
C
     IF(ISEP) 3247,4444,3247
3247 GO TO (3344,3345),IWASH
3344 XSEP=XSEP+DXI*VOFF
     IF(XSEP-XMAX) 3248,3347,3347
3347 IWASH=2
     ISEP=0
     XSEP=1.1
     DO 3015 K=1,3
     DO 3015 N=1,NSIG
3015 BCAP(N,K)=0.
     GO TO 4444
3345 IF(INDT) 3348,3348,3248
3348 IF(NITS-1) 3248,3349,3248
```

```
3349 IF(INDV.EQ.2) GO TO 6349
      IF(VZIP(1)-ALPHS) 6349,6348,6348
6348 NITS=2
      GO TO 3248
6349 CALL UNPOP(NGP1,AR,ALAM,AFACT,RMAT,CMAT,XGAM,AS,MX,NZ,XC,UE,NDIMC,
     1 AFAC2,AR2)
      GO TO 2785
3248 XATT=XSEP+DEAD1+.5*(ELD1+ELDOT)*DXI
      IF(INDT.EQ.1.AND.XATT.LT.1.2) XATT=1.2
      DEADL=XATT-XSEP
      DIFF=1.-XATT
      CALL SETSX(NSIG1,XSEP,XATT,XSIG,ANGS)
      DO 4434 N=1,NSIG
4434 XBSIG(N)=.5*(XSIG(N)+XSIG(N+1))
      DO 3086 M=1,NGP1
      DO 3086 N=1,NSIG
3086 BS(M,N)=0.
      IF(DIFF-1.E-6) 5005,5006,5006
5005 PREC=0.
      GO TO 5007
5006 CALL ATTPR(PREC,ASZ,AS,AR,CMAT,RMAT,NGP1,AR2,NDIMC,XGAM,UTU,PRECS)
5007 CALL MIXER(FPRES,PREC,UINF,UDOT,THICK,NF,XBSIG,NSIG,INDT,DEL1,THET
     11,REB,USEP,X4,CP1,RDBB)
      DO 3487 M=1,NGP1
      IF(XGAM(M)-XSEP) 3088,3088,3089
3089 IF(XATT-XGAM(M)) 3187,3087,3091
3091          CONTINUE
        ACOSX=ARCOS( ( 2.*XGAM(M) - XSEP-XATT) /DEADL)
         DO 30 N= 2, NSIG
30      BS(M,N)= SIN( FLOAT(N-1)* ACOSX)
      BS(M,1)=SQRT((XGAM(M)-XSEP)/(XATT-XGAM(M)))
3088 IF(DIFF-1.E-6) 3087,3098,3098
3098 BS(M,1)=BS(M,1)+DIFF**(-1.5)*SQRT(DEADL)*(2.*DIFF+(SQRT((1.-XGAM(M
     1))/(XATT-XGAM(M)))-1.)*(4.*XGAM(M)-1.-3.*XATT))
      GO TO 3087
3187 BS(M,1)=DIFF**(-1.5)*SQRT(DEADL)*(3.+    XATT-4.*XGAM(M))
3087 CONTINUE
3487 CONTINUE
C
C    SET-UP OF THE SECOND SET OF EQUATIONS STARTS HERE.
C
      DO 4350 K=1,NSIG
      IF(XBSIG(K)-1.) 4348,4349,4349
4348 COSK=XBSIG(K)
      SINK=SQRT(1.-COSK*COSK)
      THETK=ARCT(COSK)
      TANT=SIN(.5*THETK)/COS(.5*THETK)
      ASHZ(K)=TANT+CONA*(1.+COSK)*(1.-3.*COSK)/UINF+THXI*(PI-THETK+SINK+
     1CONA*(1.+COSK)*SINK**2)/UINF
      ASH(K,1)=.5*(ASHZ(K)-TANT)+SINK
      COUNT=1.
      DO 4355 N=2,NGAM
      COUNT=COUNT+1.
4355 ASH(K,N)=SIN(COUNT*THETK)+.75*(SIN((COUNT+1.)*THETK)/(COUNT+1.)-SI
     1N((COUNT-1.)*THETK)/(COUNT-1.))/(DXI*UINF)
```

```
      GO TO 4350
 4349 ASHZ(K)=0.
      DO 4359 N=1,NGAM
 4359 ASH(K,V)=0.
 4350 CONTINJE
      CPOT=CP1
      DO 4800 K=1,NSIG
      CORD=XBSIG(K)
         BSH(K,1)=-1.
        ACOSX=ARCOS( ( 2.*CORD     - XSEP-XATT)/DEADL)
        DO 4808 N= 2, NSIG
 4808      BSH(K,N)= COS ( FLOAT(N-1) * ACOSX)
        ARH(K)= FPRES(K)
      IF(CORD-1.) 5008,4799,4799
 5008 CALL EGAMI(2,NGAM,ACAP,BCAP(1,2),XSIGA(1),XSIGA(NSIGA+1),GAMAW(2),
     1CORD,VAL1)
      CALL EGAMI(3,NGAM,ACAP,BCAP(1,3),XSIGB(1),XSIGB(NSIGB+1),GAMAW(3),
     1CORD,VAL2)
      ARH(K)=ARH(K)+(2.*VAL1-.5*VAL2)/(DXI*UINF)+.0625*AFACT*PI*(1.+CORD
     1)*(1.-3.*CORD+THXI*(1.-CORD*CORD))/(DXI*UINF)
 4799 CONTINJE
 4800 CONTINJE
 4444 CONTINJE
C
C   CALCULATIONS FROM THIS POINT ON COMBINE THE
C   CASES OF STALLED AND UNSTALLED AIRFOILS.
C
      DO 6500 M=1,NGP1
      RMAT(M)=AR(M)
      CMAT(M,1)=ASZ(M)
      DO 6485 N=1,NGAM
 6485 CMAT(M,N+1)=AS(M,N)
      IF( ISEP) 6486,6500,6486
 6486 DO 6499 N=1,NSIG
      NGG=N+NGP1
 6499 CMAT(M,NGG)=BS(M,N)
 6500 CONTINJE
      IF( ISEP) 6502,6501,6502
 6501 NTOT=NGP1
      GO TO 6751
 6502 DO 6750 K=1,NSIG
      KK=K+NGP1
      RMAT(KK)=ARH(K)
      CMAT(KK,1)=ASHZ(K)
      DO 6748 N=1,NGAM
 6748 CMAT(KK,N+1)=ASH(K,N)
      DO 6750 N=1,NSIG
      NGG=N+NGP1
 6750 CMAT(KK,NGG)=BSH(K,N)
      NTOT=NSIG+NGP1
 6751 CALL ALSOL(NTOT, CMAT, RMAT, NDIMC)
      DO 6800 N=1,NGP1
 6800 ACAP(N,1)=RMAT(N)
      IF( ISEP) 6805,6820,6805
 6805 DO 6810 N=1,NSIG
```

```
      NGG=N+NGP1
6810  BCAP(N,1)=RMAT(NGG)
6820  CONTINUE
      GAMAW(1)=GAM1(ACAP,DXI,PI)
      DDEL=UTU*GAMAW(1)
      DELPI=DDEL/PI
      DO 6910 M = 1, NGP1
      CMAT(M,1) = ASZ(M)
      DO 6910 N = 2, NGP1
6910  CMAT(M, N) = AS(M, N-1)
      CALL WASH2(VZ2,XGAM,NGAM,CR,THICK,NF,ACAP,GAMAW(1),ISEP,DELPI,
     1 XIW(2))
      DO 339 M=1,NGP1
339   RMAT(M)=    VZ2(M)+AR2(M)
      CALL ALSOL(NGP1,CMAT,RMAT,NDIMC)
      DO 340 M=1,NGP1
      ACAP2(M,1)=RMAT(M)
340   CONTINUE
      GAMAW2(1) = GAM1(ACAP2, DXI, PI) +DDEL
      DO 1785 M=1,MX
      SIGN=1.
      IF(M-NZ) 1780,1785,1785
1780  SIGN=-SIGN
1785  CALL QECAL(UE(M,1),XC(M),SIGN,ISEP)
2785  DO 8886 I=1,2
      US2=UE(1,1)
      DO 8886 M=1,MXM1
      US1=UE(M,I)
      UE(M,1)=(US1+US2+UE(M+1,1))/3.
8886  US2=US1
      IF(ISEPT.EQ.1) CALL TAYLOR(XSEP,UE,XC,X,NZ,MX)
      GO TO (8351,8353),IWASH
8351  DO 8352 M=1.MX
8352  SCALS(M)=0.
      GO TO 1786
8353  CALL YSET(RY1,Y(2),NY,Y)
      RY=RY1
      DO 8354 M=1,MX
8354  SCALS(M)=0.
      IF(INDV.EQ.2.AND.NTIME.LT.10) GO TO 1786
      IF(INDV.EQ.2) GO TO 8370
      IF(ISEP.EQ.0.AND.VZ1P(1).LT.ALPHS) GO TO 1786
8370  CALL STAG(MX,NY,MSTOP,MST,DXI,RY,DRY,X,Y,UE,UC,V,USAV,SCALS,ISEP,
     1 MD)
      LAMQ=1
      XSEPS=XSEP
      DXX=DXI
      IF(ISEP.EQ.1.AND.ISEPT.EQ.0.AND.NITS.EQ.1) DXX=1.E30
8367  CALL BLC(X,Y,MST,MEND,NY,RY,DRY,DXX,REB,UPRIM,FLAM,XFLAM,TEST,U,SC
     1ALE,UE,UC,V,XSEP,USEP,DISP,THETA,LOWER,LAMQ,MSEP,XC,USAV,SCALS,NIT
     1S,NTIME, MD)
      IF(XSEP=XMAX) 7736,7735,7735
7735  IF(ISEP) 1786,1786,7736
7736  DEL1=DISP
      THET1=THETA
```

```
      INDT=1-LAMQ
      IF(.NOT.(INDT.EQ.1.AND.NOTBL.EQ.2)) GO TO 378
      XSEP=XSEPS
      INDT=0
      GO TO 1786
378   WRITE(MOUT,23) XSIG(1),CPOT,XSEP
      IF(INDT) 8462,8462,8463
8462  IF(ISEP) 8562,8562,8563
8563  IF(NITS-1) 8562,8562,8662
8662  IF(ISEPT) 7742,7742,8562
8562  CALL BJBB(DEL1,THET1,REB,XSEP,USEP,XC5,DCP,DEL5,X,XC,MX,NZ,X5,U5,U
     1E,ALTC,RENEL,USTOP)
      IF(USEP.GT.USTOP) USEP=USEP+.075114*(USEP-USTOP)
      PDIFF=(USEP-U5)*(USEP+U5)
      WRITE(MOUT,22) PDIFF,DCP
      IF(DCP-PDIFF) 8263,8366,8366
8263  ISEPT=0
      GO TO 8463
8366  IF(ISEP) 8368,8368,8369
8369  IF(ISEPT) 8467,8467,8368
8467  IWASH=1
      NITS=2
      GO TO 3344
8368  GO TO (8168,1786),NOTBL
8168  CALL REATT(UC,V,X,Y,MX,NY,RY,DRY,UE,X5,DEL5,MST,REB)
      LAMQ=0
      GO TO 8367
8463  IF(ISEP) 7741,7741,7742
7741  ISEP=1
      NITS=NITS+1
      IF(INDT) 7743,7743,7643
7643  ISEPT=1
      CALL CPC(CP1,XSEP,1.,0)
      GO TO 3248
7742  CALL ELDER(BCAP,XSIG,NSIG,UBUB,ELDOT,SIGSUM,YMX)
      IF(ISEP.EQ.1.AND.ISEPT.EQ.0.AND.NITS.EQ.1) GO TO 9210
      IF(XSEP+.5) 7841,7842,7842
7841  EPS=EPSLE
      GO TO 7843
9307  IF(XSEP-XMAX) 9212,9212,7835
9212  CALL CPC(CP1,XSEP,1.,ISEP)
      GO TO 3248
7743  IF(NITS-1) 7737,7737,3248
7737  NITS=NITS+1
      ELDOT=ELD1
7842  EPS=EPSTE
7843  DXSEP=ABS(XSEP-XSEPS)
      IF(DXSEP-EPS) 7834,7834,9210
7834  IF(XSEP-XMAX) 1786,1786,7835
7835  ISEP=0
      ISEPT=0
      DO 7836 K=1,3
      DO 7836 N=1,NSIG
7836  BCAP(N,K)=0.
      GO TO 1786
```

```
9210 NITS=NITS+1
     IF(NITS.EQ.2.AND.INDT.EQ.0) XSEPS=XSEP
     IF(NITS-4) 9211,9211,1786
9211 IF(XSEP-XSEPS) 9305,9305,9306
9305 XSEP=(1.-FUP)*XSEPS+FUP*XSEP
     GO TO 9307
9306 XSMAX=XSEPS+FPAST*(1.-XSEPS)
     IF(XSEP.GT.XSMAX) XSEP=XSMAX
     GO TO 3248
1786 WRITE(MOUT,20) NTIME
     XSEP=XSIG(1)
     WRITE(MOUT,26) X1VOR
     PITC  = PITCH * 180. / PI
209  WRITE(MOUT,10) TIME,UINF,XSEP,XATT,PITC
     WRITE(MOUT,11)
     WRITE(MOUT,12) (N,XGAM(N),VZIP(N),AR(N),ACAP(N,1),XIW(N),GAMAW(N),
    1N=1,NGP1)
     WRITE(6,9005)
     DO 63 M = 1, NGP1
63   WRITE(MOUT,9006) M, XGAM(M),VZ2(M),AR2(M), ACAP2(M,1), XIW(M),
6   A   GAMAW2(M)
     IF(ISEP) 7432,7433,7432
7432 WRITE(MOUT,13)
     WRITE(MOUT,17) (N,XBSIG(N),FPRES(N),ARH(N),BCAP(N,1),N=1,NSIG)
     WRITE(MOUT,14) ELDOT
     WRITE(MOUT,18) XSIG(1),CPOT,X4,CPOT,XATT,PREC
7433 WRITE(MOUT,15)
     XPC=-1.
     DO 7102 N=1,NCP1
     CALL QECAL(QEL,XPC,-1.,ISEP)
     CALL QECAL(QEU,XPC, 1.,ISEP)
     CALL CPC(CPL,XPC,-1.,ISEP)
     CALL CPC(CPU,XPC, 1.,ISEP)
     IF(N-1) 7546,7545,7546
7545 CPL=CPJ
7546 DLIFT=CPL-CPU
     WRITE(MOUT,16) XPC,QEL,CPL,QEU,CPU,DLIFT
7102 XPC=XPC+PINT
     CMPAS=CMPA
     CALL CLCM(PITCH,AROT,ISEP,CMPA,CAMBR)
     IF(MD.EQ.1) GO TO 9999
     DO 7950 M=1,MX
     SCALE(M,2)=SCALE(M,1)
     SCALE(M,1)=SCALS(M)
     DO 7950 N=1,NY
     U(M,N,2)=U(M,N,1)
7950 U(M,N,1)=USAV(M,N)
     GO TO 9999
8989 CONTINUE
     STOP
2    FORMAT(3F10.4)
4    FORMAT(1H1///)
5    FORMAT(6F10.4)
6    FORMAT(1H1,50X,34HANALYSIS OF UNSTEADY AIRFOIL STALL///)
7    FORMAT(8X,6HUBAR =E13.5/7X,7HUFREQ =E13.5//3X,11HALPHA ONE =E13.5/
```

```
     13X,11HALPHA TWO =E13.5/8X,6HHBAR =E13.5/11X,3HA =E13.5/8X,6HFREQ =
     1E13.5//8X,6HRO/B =E13.5//9X,5HREB =E13.5///)
8     FORMAT(19X,1HN,25X,4HC(N),26X,4HT(N),25X,5HTH(N)/)
9     FORMAT(I20,3E30.5)
10    FORMAT(5X,3HT =E13.5/5X,3HU =E13.5/4X,4HXS =E13.5/4X,4HXO =E13.5/4
     1X,4HPA =E13.5//)
11    FORMAT(///4X,1HN,11X,1HX,14X,5HVZ(X),12X,5HRN(X),12X,4HA(N),21X,3H
     1X IW,14X,5HGAMMA/)
12    FORMAT(I5,4E17.5,8X,2E17.5)
13    FORMAT(1H1,8X,1HN,20X,1HX,21X,5HFP(X),22X,5HRH(N),21X,4HB(N)/)
14    FORMAT(//54X,9H L-DOT =E13.5///51X,27HPRESSURES IN SEPARATED FLCW
     1//55X,1HX,19X,2HCP/)
15    FORMAT(1H1,11X,1HX,16X,3HQEL,15X,3HCPL,15X,3HQEU,15X,3HCPU,13X,9HC
     1PL - CPU/)
16    FORMAT(6E18.5)
17    FORMAT(I10,4E25.5)
18    FORMAT(3(40X,2E20.5/))
20    FORMAT(1H1,50X,12HTIME STEP NOI3//)
22    FORMAT(///40X,26HINCREASE IN CP REQUIRED ISE13.5//40X,26HINCREASE
     1IN CP POSSIBLE ISE13.5)
23    FORMAT(///45X,23HPOTENTIAL FLOW      XS =E12.4/60X,8HCP(XS) =E12.4/
     1/45X,23HBOUNDARY LAYER      XS =E12.4)
25    FORMAT(12X,4HNV =I2,3X,3HS =E12.4,3X,3HH =E12.4,3X,3HG =E12.4,3X,4
     1HX1 =E12.4//12X,4HMI =E12.4,3X,4HWT =E12.4,3X,4HPA =E12.4///)
26    FORMAT(4X,4HX1 =E13.5)
101   FORMAT(////10X,4HCR =E13.5/7X,7HCRHAT =E13.5)
9005  FORMAT(1H1///T50,'SECOND ORDER SOLUTION'//
90051           4X,1HN,11X,1HX,14X,5HVZ(X),12X,5HRN(X),12X,4HA(N),13X,3H
90052X IW,14X,5HGAMMA/)
9006  FORMAT(I5,6E17.5)
      END
```

```
      SUBROUTINE BLC(X,Y,MST,MEND,NY,RY,DRY,DXI,REB,UPRIM,FLAM,XFLAM,TES
     1T,U,SCALE,UE,UC,V,XSEP,USEP,DISS,THETS,LOWER,LAMQ,MSEP,XC,USAV,SCA
     1LS,NITS,NTIME, MX)
C
C  PROGRAM FOR ANALYZING LAMINAR AND TURBULENT BOUNDARY LAYERS
C  BY THE METHOD OF FINITE DIFFERENCES.  IF THE INTEGER LAMQ
C  IS GREATER THAN ZERO, THE BOUNDARY LAYER IS LAMINAR.
C
      DIMENSION U(MX, NY, 2), USAV (MX, NY)
      DIMENSION SCALS(300), X(300), Y(300), UC(100,3), V(100,2),
     X     UE(300,3), XC(300)
      DIMENSION SD(100),SE(100),SF(100),VISC(100,2),GRAD(100)
      DIMENSION A(100),B(100),C(100),D(100),F(100)
      DIMENSION ALPHA(100),BETA(100),GAMMA(100),DELTA(100)
      DIMENSION SCALE(300,2),VAR1(100),VAR2(100)
      DIMENSION FLAM(10),XFLAM(10),YB1(100),YB2(100)
      DIMENSION CAPG(100),CAPH(100),CAPJ(100),CAPK(100)
      DOUBLE PRECISION AP(100),BP(100),CP(100),DP(100),FP(100),UP(100)
 10   FORMAT(1H1,41X,36H ANALYSIS OF LAMINAR BOUNDARY LAYER///51X,12HTI
     1ME STEP NOI3//51X,12HITERATION NOI3///3X,1HM,8X,1HX,13X,2HXC,12X,2
     1HUE,10X,6H-DP/DX,9X,5HDELTA,9X,5HDISPL,9X,5HTHETA,9X,5HSHEAR,4X,
     1 1HI/)
 11   FORMAT(1H1,41X,36HANALYSIS OF TURBULENT BOUNDARY LAYER///51X,12HTI
     1ME STEP NOI3//51X,12HITERATION NOI3///3X,1HM,8X,1HX,13X,2HXC,12X,2
     1HUE,10X,6H-DP/DX,9X,5HDELTA,9X,5HDISPL,9X,5HTHETA,9X,5HSHEAR,4X,
     1 1HI/)
 12   FORMAT(I4,8E14.4,I3)
 20   FORMAT(1H1,2X,3HM =I4//2X,3HX =E14.5//2X, 4HUE =E14.5,10X,17H-(1/R
     1HO)(DP/DX) =E14.5,10X,5HREB =E14.5,10X,4HU' =E14.5///)
 24   FORMAT(2X,25HPHYSICAL           DELTA =E14.5,8X,12HDELTA STAR =E14.
     15,8X,7HTHETA =E14.5//2X,25HTRANSFORMED        DELTA =E14.5,8X,12HDE
     1LTA STAR =E14.5,8X,7HTHETA =E14.5///)
 21   FORMAT(25X,1HY,19X,1HU,19X,1HV,16X,5HDU/DY,14X,6HNUE/NU/)
 22   FORMAT(10X,5E20.5)
 23   FORMAT(//30X,17HSEPARATION AT X =E13.5,6H, XC =E13.5)
 25   FORMAT(///40X,12HWALL SHEAR =E14.5//)
 30   FORMAT(//50X,17HTRANSITION AT X =E14.5)
 35   FORMAT(/20X,35HSCALE CHANGE - Y-MAX INCREASED FROME12.4,3H TOE12.4
     1/)
 810  FORMAT(10X,7HAT STEPI3,22H, THE WALL GRADIENT ISE12.4)
      DATA MOUT /6/
      BCON = 1.5/DXI
      FCON = 1./(2.*DXI)
      IF(MX.GT.1) GO TO 900
      BCON=0.
      FCON=0.
      DXI=1.E30
 900  CONTINUE
      MAXIT=0
      MTRAN=-1
      YSUB2=Y(2)
      MST2 = MST - 2
      MST1=MST-1
      GO TO (543,550),LOWER
```

```
543   IF(LAMQ) 544,544,545
544   WRITE(MOUT,11) NTIME,NITS
      GO TO 550
545   WRITE(MOUT,10) NTIME,NITS
550   CONTINUE
      YTR = SQRT(REB)
      UC(1,1) = 0.
      V(1,1) = 0.
      NV = NY - 2
      NVM1 = NV - 1
      NVP1 = NV + 1
      CALL YDIFF(NY,ALPHA,BETA,GAMMA,DELTA,SD,SE,SF,C2,C3,C4,Y)
      DO 41 N=1,NVP1
      VISC(N,1) = 1.
41    VISC(N,2) = 1.
      DO 42 M=MST2,MST1
      L = MST1-M+2
      DO 50 N=1,NV
50    GRAD(N+1) = SD(N+1)*UC(N+2,L)+SE(N+1)*UC(N+1,L)-SF(N+1)*UC(N,L)
      GRAD(1) = C2*UC(2,L)+C3*UC(3,L)+C4*UC(4,L)
      MM=M-1
      CALL PGRAD(MM,X,UE,DXI,PRESS,SA,SB,SC,SR,SS)
      DO 456 N=1,NY
456   UC(N,1)=UC(N,L)
      CALL SETUP(LAMQ,M,NV,REB,X,Y,UC,PRESS,GRAD,DELT,DISP,THETA,VISC,MT
     1RAN)
42    CONTINUE
      MEND1 = MEND - 1
      GRADS=GRAD(1)
      GRADSS=GRAD(1)
C
C  THE MAIN CALCULATION STARTS HERE.
C
      DO 99 M=MST1,MEND1
      ITER=0
      WALLG=0.
      MP1=M+1
      DELTP = DELT/YTR
      DISPT = DISP*YTR
      THETT = THETA*YTR
      SHEAR = GRAD(1)/YTR
      IF(MAXIT.EQ.10 .AND. EPWS.LE.EPW) MAXIT=9
      GO TO (561,562),LOWER
561   WRITE(MOUT,12) M,X(M),XC(M),UE(M,1),PRESS,DELTP,DISP,THETA,SHEAR,
     1 MAXIT
      GO TO 225
562   WRITE(MOUT,20) M,X(M),UE(M,1),PRESS,REB,UPRIM
      WRITE(MOUT,24) DELTP,DISP,THETA,DELT,DISPT,THETT
      WRITE(MOUT,21)
      WRITE(MOUT,22) (Y(N),UC(N,2),V(N,1),GRAD(N),VISC(N,1),N=1,NVP1)
      WRITE(MOUT,25) SHEAR
225   IF(GRADSS-GRADS-1.E-6) 229,229,408
408   XSX=X(M-2)+(X(M-1)-X(M-2))*GRADSS/(GRADSS-GRADS)
      IF(XSX-X(M)) 409,409,229
409   WFS=(XSX-X(M-1))/(X(M)-X(M-1))
```

```
      GO TO 224
  229 IF(GRAD(1)) 227,227,2000
 2000 IF(MAXIT.LT.10) GO TO 223
      IF(EPWS.GT.EPW) GO TO 223
      WFS=0.
      GO TO 224
  227 WFS=GRADS/(GRADS-GRAD(1))
  224 WFS1=1.-WFS
      XSEP=WFS1*XC(M-1)+WFS*XC(M)
      XBL=WFS1*X(M-1)+WFS*X(M)
      USEP=WFS1*UE(M-1,1)+WFS*UE(M,1)
      WFP=(XBL-X(M-2))/(X(M-1)-X(M-2))
      WFP1=1.-WFP
      DISS=DISSS*WFP1+DISS*WFP
      THETS=THETSS*WFP1+THETS*WFP
      IF(THETS.LT.0.) THETS=-THETS
      IF(M.LT.MST1+3) GO TO 223
      WRITE(MOUT,23) XBL,XSEP
      IF(LAMQ.EQ.0.AND.M.LT.MTRAN+5) LAMQ=1
      GO TO 222
  223 CONTINUE
      IF(LAMQ) 801,801,802
  802 CALL TRANS(UPRIM,PRESS,THETA,REB,UC,NY,FLAM,XFLAM,LAMQ)
      IF(LAMQ) 805,805,801
  805 WRITE(MOUT,30) X(M)
      MTRAN = M+1
  801 CONTINUE
      IF(Y(NV)-DELT) 620,641,641
  620 RY=RY+DRY
C
C  RESCALING CALCULATION STARTS HERE.
C
      DO 632 N=1,NY
      YB1(N) = Y(N)
      VAR1(N) = UC(N,2)
  632 VAR2(N) = UC(N,3)
      CALL YSET(RY,YSUB2,NY,Y)
      WRITE(MOUT,35) YB1(NY),Y(NY)
      DO 633 N=2,NVP1
      YIN = Y(N)
      CALL TERP(YIN,YB1,VAR1,NY,UPAS1)
      UC(N,2) = UPAS1
      CALL TERP(YIN,YB1,VAR2,NY,UPAS2)
  633 UC(N,3) = UPAS2
      CALL YDIFF(NY,ALPHA,BETA,GAMMA,DELTA,SD,SE,SF,C2,C3,C4,Y)
      IF(LAMQ) 700,700,701
  700 DO 635 N=2,NVP1
      VAR1(N) = VISC(N,1)
  635 VAR2(N) = VISC(N,2)
      DO 636 N=2,NVP1
      YIN = Y(N)
      CALL TERP(YIN,YB1,VAR1,NVP1,UPAS1)
      VISC(N,1) = UPAS1
      CALL TERP(YIN,YB1,VAR2,NVP1,UPAS2)
  636 VISC(N,2) = UPAS2
```

```
 701  DO 637 N=2,NVP1
      VAR1(N) = V(N,1)
 637  VAR2(N) = V(N,2)
      DO 638 N=2,NVP1
      YIN = Y(N)
      CALL TERP(YIN,YB1,VAR1,NVP1,UPAS1)
      V(N,1) = UPAS1
      CALL TERP(YIN,YB1,VAR2,NVP1,UPAS2)
 638  V(N,2) = UPAS2
 641  CONTINJE
C
C  RESCALING CALCULATION ENDS HERE.
C
      CALL PGRAD(M,X,UE,DXI,PRESS,SA,SB,SC,SR,SS)
C
C   RECURSION RELATIONS ARE SET UP HERE.
C
      IF(MX.EQ.1) GO TO 820
      IF(SCALE(M+1,1)-1.) 522,522,521
 521  IF(SCALE(M+1,2)-1.) 522,522,523
 522  LACKU=1
      FACU1=JE(M+1,2)/UE(M+1,1)
      FACU2=JE(M+1,3)/UE(M+1,1)
      GO TO 820
 523  LACKU=2
      DO 610 NN=1,NY
      VAR1(NN) = U(M+1,NN,1)
 610  VAR2(NN) = U(M+1,NN,2)
      CALL YSET(SCALE(M+1,1),YSUB2,NY,YB1)
      CALL YSET(SCALE(M+1,2),YSUB2,NY,YB2)
 820  DO 88 N=2,NV
      CALL CAPS(ITER,N,CAPG,CAPH,CAPJ,CAPK,SR,SS,SD,SE,SF,VISC,V,UC)
      A(N)=-SF(N)*CAPG(N)-DELTA(N)*CAPH(N)+SF(N)*CAPJ(N)
      B(N)=BCON+SA*CAPK(N)+SE(N)*CAPG(N)-GAMMA(N)*CAPH(N)-SE(N)*CAPJ(N)
      C(N)=SD(N)*CAPG(N)-BETA(N)*CAPH(N)-SD(N)*CAPJ(N)
      D(N) = -ALPHA(N)*CAPH(N)
      IF(MX.EQ.1) GO TO 576
      GO TO (574,575),LACKU
 574  UPAS1=FACU1*UC(N,1)
      UPAS2=FACU2*UC(N,1)
      GO TO 576
 575  YIN = Y(N)
      CALL TERP(YIN,YB1,VAR1,NY,UPAS1)
      CALL TERP(YIN,YB2,VAR2,NY,UPAS2)
 576  F(N) = PRESS+FCON*(4.*UPAS1-UPAS2)+CAPK(N)*(SR*UC(N,2)-SC*UC(N,3))
 88   CONTINJE
C
C   SOLUTION FOR VELOCITY PROFILE STARTS HERE.
C
      DO 89 N=2,NV
      AP(N) = A(N)
      BP(N) = B(N)
      CP(N) = C(N)
      DP(N) = D(N)
 89   FP(N) = F(N)
```

```
      DO 77 N=2,NVM1
      CP(N) = CP(N)/BP(N)
      DP(N) = DP(N)/BP(N)
      FP(N) = FP(N)/BP(N)
      BP(N+1) = BP(N+1) - CP(N)*AP(N+1)
      CP(N+1) = CP(N+1) - DP(N)*AP(N+1)
77    FP(N+1) = FP(N+1) - FP(N)*AP(N+1)
      UP(NY) = UE(M+1,1)
      UP(NVP1) = UP(NY)
      UP(NV) = (FP(NV)-UP(NY)*(DP(NV) + CP(NV)))/BP(NV)
      DO 66 N=3,NV
      NN=NV+2-N
66    UP(NN) = FP(NN) - DP(NN)*UP(NN+2) - CP(NN)*UP(NN+1)
      DO 65 N=2,NY
65    UC(N,1) = UP(N)
      IF(ITER) 843,841,843
841   DO 842 N=2,NVP1
      V(N,2) = V(N,1)
842   VISC(N,2)=VISC(N,1)
      DISSS=DISS
      DISS=DISP
      THETSS=THETS
      THETS=THETA
      GRADSS=GRADS
      GRADS=GRAD(1)
843   DO 55 N=2,NVP1
55    V(N,1) = V(N-1,1)-.5*(Y(N)-Y(N-1))*(SA*(UC(N,1)+UC(N-1,1))-SB*(UC(
     1N,2)+UC(N-1,2))+SC*(UC(N,3)+UC(N-1,3)))
      DO 56 N=1,NV
56    GRAD(N+1) = SD(N+1)*UC(N+2,1)+SE(N+1)*UC(N+1,1)-SF(N+1)*UC(N,1)
      GRAD(1) = C2*UC(2,1)+C3*UC(3,1)+C4*UC(4,1)
      CALL SETUP(LAMQ,MP1,NV,REB,X,Y,UC,PRESS,GRAD,DELT,DISP,THETA,VISC,
     1MTRAN)
      ITER=ITER+1
      GO TO (830,809),LOWER
809   WRITE(MOUT,810) ITER,GRAD(1)
830   IF(ITER-9) 811,811,812
811   EPWS=EPW
      EPW=ABS(GRAD(1)-WALLG)
      IF(WALLG-1.) 120,120,119
119   EPW=EPW/WALLG
120   IF(EPW-TEST) 812,814,814
814   WALLG=GRAD(1)
      GO TO 820
812   DO 44 N=1,NY
      UC(N,3) = UC(N,2)
44    UC(N,2) = UC(N,1)
      MAXIT=ITER
      IF(MX.EQ.1) GO TO 99
      SCALS(M+1)=RY
      DO 48 N=1,NY
48    USAV(M+1,N)=UC(N,1)
99    CONTINUE
      XSEP=1.1
      USEP=UE(MX,1)
```

```
222   CONTINJE
      RETURN
      END
```

```
      SUBROUTINE STAG(MX,NY,MSTOP,MST,DXI,RY,DRY,X,Y,UE,UC,V,USAV,SCALS,
     1ISEP,MD)
C  PROGRAM FOR CALCULATING THE BOUNDARY LAYER PROFILE NEAR
C  THE STAGNATION POINT
C
      DIMENSION USAV(MD, NY), SCALS(1)
      DIMENSION PHIZ(24),PHIP(24),ETAP(24)
      DIMENSION X(300),Y(100),UE(300,3),UC(100,3),V(100,2)
      DIMENSION EF(100),EFP(100)
      DATA ETAP /0.,.2,.4,.6,.8,1.,1.2,1.4,1.6,1.8,2.,2.2,2.4,2.6,2.8,3.
     1,3.2,3.4,3.6,3.8,4.,4.2,4.4,4.6/
      DATA PHIZ /0.,.0233,.0881,.1867,.3124,.4592,.622,.7967,.9798,1.168
     19,1.362,1.5578,1.7553,1.9538,2.153,2.3526,2.5523,2.7522,2.9521,3.1
     1521,3.3521,3.5521,3.7521,3.9521/
      DATA PHIP /0.,.2266,.4145,.5663,.6859,.7779,.8467,.8968,.9323,.956
     18,.9732,.9839,.9905,.9946,.997,.9984,.9992,.9996,.9998,.9999,1.,1.
     1,1.,1./
      BAG=.08
      IF(ISEP) 10,10,5
 5    BAG=.35
 10   EF(1) = 0.
      EFP(1) = 0.
      DO 20 M=1,MX
      IF(UE(M,1)) 20,20,19
 19   MSP = M
      GO TO 21
 20   CONTINUE
 21   ASTAG = (UE(MSP+2,1)-UE(MSP+1,1))/(X(MSP+2)-X(MSP+1))
      IF(ASTAG) 22,22,23
 22   ASTAG=(UE(MSP,1)-UE(MSP-1,1))/(X(MSP)-X(MSP-1))
 23   SQAS = SQRT(ASTAG)
      DELT = 2.6/SQAS
 309  IF(DELT-Y(NY-3)) 311,310,310
 310  RY=RY+DRY
      CALL YSET(RY,Y(2),NY,Y)
      GO TO 309
 311  CONTINUE
      DO 80 N=2,NY
      YET = Y(N)*SQAS
      DO 33 NN=1,24
      IF(YET-ETAP(NN)) 408,408,33
 408  MARK = NN
      GO TO 410
 33   CONTINUE
      EF(N) = YET-.6479
      EFP(N) = 1.
      GO TO 80
 410  FRACT = (YET-ETAP(MARK-1))/(ETAP(MARK)-ETAP(MARK-1))
      FRAC1 = 1.-FRACT
      EF(N) = PHIZ(MARK-1)*FRAC1+PHIZ(MARK)*FRACT
      EFP(N) = PHIP(MARK-1)*FRAC1+PHIP(MARK)*FRACT
 80   CONTINUE
      M1 = MSP-MSTOP
      M2 = MSP+MSTOP
```

```
      M=M1-1
50    M=M+1
      MST=M+1
      DO 71 N=1,NY
      UC(N,3) = UC(N,2)
      JC(N,2) = UE(M,1)*EFP(N)
      V(N,2) = V(N,1)
      V(N,1) = -SQAS*EF(N)
71    CONTINJE
      IF(MD.EQ.1) GO TO 73
      SCALS(M)=RY
      DO 72 V=1,NY
72    USAV(M,N)=UC(N,2)
73    IF(M-M2) 50,55,55
55    IF(UE(M,1)-BAG) 50,50,81
81    CONTINJE
      RETURN
      END
```

```
      SUBROUTINE THCALC(CR,THICK,CRHAT,THAT,NF)
      DIMENSION THICK(24),THAT(24),ARG(61)
      DATA NSIMP,PI /60,3.14159/
      SIGN=1.
      SUM=0.
      DO 5 N=1,NF
      THAT(N)=0.
      SIGN=-SIGN
5     SUM=SUM+SIGN*THICK(N)
      CRHAT=CR*(3.*CR+2.*SUM)
      DT=PI/FLOAT(NSIMP)
      ARG(1)=0.
      ARG(NSIMP+1)=0.
      ANGLE=0.
      DO 10 N=2,NSIMP
      ANGLE=ANGLE+DT
      SINT=SIN(ANGLE)
      COST=CJS(ANGLE)
      ST=0.
      STPR=0.
      COUNT=0.
      DO 6 I=1,NF
      COUNT=COUNT+1.
      TTI=COJNT*ANGLE
      SINIT=SIN(TTI)
      ST=ST+THICK(I)*SINIT
6     STPR=STPR+THICK(I)*(COUNT*SINT*SINIT-2.*COST*COS(TTI))
      ARG(N)=(CR*(1.-COST)+SINT*ST)*(CR*(1.-2.*COST)+STPR)
10    ARG(N)=(ARG(N)-CRHAT*(1.-COST))/SINT
      DT2=DT+DT
      ANGLE=0.
      DO 15 N=2,NSIMP,2
      NP1=N+1
      ANGLE=ANGLE+DT2
      ANGA=ANGLE-DT
      COUNT=0.
      DO 12 I=1,NF
      COUNT=COUNT+1.
12    THAT(I)=THAT(I)+4.*ARG(N)*SIN(COUNT*ANGA)+2.*ARG(NP1)*SIN(COUNT*AN
     1GLE)
15    CONTINJE
      FACT=DT2/(3.*PI)
      DO 20 N=1,NF
20    THAT(N)=FACT*THAT(N)
      RETURN
      END
```

```
      SUBROUTINE CLCM(PITCH,AROT,ISEP,CMPA,CAMBR)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XIW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      DIMENSION CAMBR(24),X(2),T(2),CPL(2),CPU(2),CP(2),TP(2),A(2),
     1 B(2),C(2)
      DATA NSIMP,PI /80,3.14159/
      CN=0.
      CT=0.
      CMZ=0.
      T1=0.
      TE=PI
      IF(ISEP.EQ.1 .AND. XATT.LT.1.) TE=ARCOS(XATT)
   99 DT=(TE-T1)/FLOAT(NSIMP)
      X(1)=COS(T1)
      CALL CPC(CPL(1),X(1),-1.,ISEP)
      CALL CPC(CPU(1),X(1), 1.,ISEP)
      CALL PRIMES(CP(1),TP(1),T1,CR,THICK,CAMBR,NF)
      SUMN=(CPL(1)-CPU(1))*SIN(T1)
      SUMT=(CP(1)-TP(1))*CPL(1)-(CP(1)+TP(1))*CPU(1)
      SUMM=SUMN*X(1)
      T(2)=T1
      DO 10 N=2,NSIMP,2
      T(1)=T(2)+DT
      T(2)=T(1)+DT
      DO 5 I=1,2
      X(I)=COS(T(I))
      CALL CPC(CPL(I),X(I),-1.,ISEP)
      CALL CPC(CPU(I),X(I), 1.,ISEP)
      CALL PRIMES(CP(I),TP(I),T(I),CR,THICK,CAMBR,NF)
      A(I)=(CPL(I)-CPU(I))*SIN(T(I))
      B(I)=(CP(I)-TP(I))*CPL(I)-(CP(I)+TP(I))*CPU(I)
    5 C(I)=A(I)*X(I)
      SUMN=SUMN+4.*A(1)+2.*A(2)
      SUMT=SUMT+4.*B(1)+2.*B(2)
   10 SUMM=SUMM+4.*C(1)+2.*C(2)
      CN=CN+DT*(SUMN-A(2))/6.
      CT=CT+DT*(SUMT-B(2))/6.
      CMZ=CMZ-DT*(SUMM-C(2))/12.
      IF(TE.EQ.PI) GO TO 20
      T1=TE
      TE=PI
      GO TO 99
   20 COSP=COS(PITCH)
      SINP=SIN(PITCH)
      CL=CN*COSP-CT*SINP
      CD=CT*COSP+CN*SINP
      CMPA=CMZ+.5*AROT*CN
      WRITE(6,2) CL,CD,CN,CT,CMZ,CMPA,AROT
    2 FORMAT(////40X,4HCL =E13.5/40X,4HCD =E13.5//40X,4HCN =E13.5/40X,
     1 4HCT =E13.5//39X,5HCMZ =E13.5/38X,6HCMPA =E13.5,5X,4H(A =E13.5,1H
     1))
      RETURN
      END
```

```
      SUBROUTINE CPC(CP,XC,SIGN,ISEP)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XIW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      DIMENSION GAMS(3),ASUM(30,3)
      NGP1 = NGAM + 1
      DO 6 I = 1, 3
      GAMS(I) = GAMAW(I) + GAMAW2(I)
      DO 6 J = 1, NGP1
    6 ASUM(J,I) = ACAP(J,I) + ACAP2(J,I)
      RECIP=1./(UINF*UINF)
      CALL QECAL(U,XC,SIGN,ISEP)
      CP =       (U/UINF)**2 - 1.
      CALL EGAMI(1,NGAM,ASUM,BCAP(1,1),XSIG(1),XSIG(NSIG+1),GAMS(1),XC,
     1VAL1)
      CALL EGAMI(2,NGAM,ASUM,BCAP(1,2),XSIGA(1),XSIGA(NSIGA+1),GAMS(2),
     1XC,VAL2)
      CALL EGAMI(3,NGAM,ASUM,BCAP(1,3),XSIGB(1),XSIGB(NSIGB+1),GAMS(3),
     1XC,VAL3)
      CP=CP+SIGN*RECIP*(1.5*VAL1-2.*VAL2+.5*VAL3)/DXI
   20 CP=-CP
      RETURN
      END
```

```
      SUBROUTINE QECAL(Q,X,SIGN,ISEP)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XTW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      DIMENSION SN(30),CS(30)
      RDT=RDBB/2.
      USIG=0.
      XP=1.+X
      XM=1.-X
      COST=X
      THETA=ARCOS(X)
      NSUM=NGAM
      IF(NF.GT.NGAM) NSUM=NF
      ANGLE=0.
      DO 5 N=1,NSUM
      ANGLE=ANGLE+THETA
      SN(N)=SIN(ANGLE)
    5 CS(N)=COS(ANGLE)
      SINT=SN(1)
      SAS=0.
      SAC=ACAP(1,1)
      DO 6 N=1,NGAM
      NP=N+1
      SAS=SAS+(ACAP(NP,1)+ACAP2(NP,2))*SN(N)
    6 SAC=SAC+ACAP(NP,1)*CS(N)
      COST2=COST+COST
      ST=0.
      ST1=0.
      ST2=0.
      ST3=0.
      EN=0.
      ENS=0.
      DO 7 N=1,NF
      EN=EN+1.
      ENS=ENS+SINT
      STICK=SN(N)*THICK(N)
      ST3=ST3+STICK*EN*EN
      ST=ST+STICK
      ST1=ST1+(THICK(N)+THAT(N))*(ENS*SN(N)-COST2*CS(N))
    7 ST2=ST2+EN*THICK(N)*CS(N)
      ST3=3.*COST*ST2-SINT*(2.*ST1+ST3)
      ST2=SINT*ST2+COST2*ST
      ALTP=SAS+.25*XP*(3.*X-1.)*(GAMAW(1)+GAMAW2(1))
      ALTM=ACAP(1,1)+ACAP2(1,1)
      IF(ISEP.EQ.0) GO TO 10
      DEADL=XATT-XSEP
      CALL SIGF(X,USIG,XSEP,XATT,BCAP,NSIG,DEADL,RDT)
      IF(XATT.GE.1.) GO TO 10
      IF(X.LT.XATT) GO TO 10
      ALTP=ALTP+((1.+3.*XATT-3.*RDT -4.*X)*SQRT(DEADL*XM/(X-XATT+RDT))/(
     11.-XATT)**1.5)*BCAP(1,1)
   10 ALTP=.5*SIGN*ALTP+USIG+UINF*((CR+CRHAT)*(XM-X)+ST1+(1.-X+X*X)*CR*
     12+(CR*XM+SINT*ST)*(3.*X*CR+ST3))
```

```
      ALTM=.5*SIGN*ALTM-UINF*CR*(1.+X+X*X)*ST
      DEN=XP+RDBB/2.
      COP=SQRT(XP/DEN)
      COM=SQRT(XM/DEN)
      QT=COP*(UT+ALTP)+COM*ALTM
      QN=UN-.5*SAC+      GAMSUM(X,GAMAW,XIW,NWAKE)-UINF*SIGN*(CR*SINT+ST2
     1)
      QNS=COP*(QN**2-(UINF*CR)**2*(2.-X-X+X*X))-2.*UINF*SIGN*X*CR*COM*QN
      Q=SIGN*(QT+.5*QNS/UINF)
      RETURN
      END
```

```
      SUBROUTINE SIGF(X,USIG,XSEP,XATT,BCAP,NSIG,DEADL,RDT)
      DIMENSION BCAP(100,3)
      IF(X.LE.XSEP .OR. X.GE.XATT) GO TO 40
      COSB=(2.*X-XATT-XSEP)/DEADL
      BETA=ARCOS(COSB)
      ANGLE=0.
      SUMC=0.
      DO 10 N=2,NSIG
      ANGLE=ANGLE+BETA
10    SUMC=SUMC+BCAP(N,1)*COS(ANGLE)
      USIG=       .5*(SUMC-BCAP(1,1))
      RETURN
40    XBAR=(2.*X-XATT-XSEP)/DEADL
      SUMAND=XBAR-XBAR*SQRT(ABS(1.-XBAR**(-2)))
      FACTOR=SUMAND
      SUM=BCAP(2,1)*SUMAND
      DO 45 N=3,NSIG
      SUMAND=SUMAND*FACTOR
45    SUM=SUM+BCAP(N,1)*SUMAND
      USIG=       SUM+.5*BCAP(1,1)*(SQRT((X-XSEP)/(X-XATT+RDT))-1.)
      RETURN
      END
```

```
      FUNCTION GAMSUM(X,GAMAW,XIW,NWAKE)
      DIMENSION GAMAW(1000),XIW(1000)
      DATA PI /3.14159/
      SUM=0.
      NWM=NWAKE-1
      DO 5 J=2,NWM
      JP=J+1
      DL=XIW(J)-X
    5 SUM=SUM+(GAMAW(J)+(GAMAW(J)-GAMAW(JP))*DL/(XIW(JP)-XIW(J)))*ALOG(
     1 (XIW(JP)-X)/DL)
      ARG=1.-X
      XM=ARG
      IF(ARG.LE.0.) ARG=1.
      XP=1.+X
      AP=XP
      IF(AP.LE.0.) AP=1.
      D2=XIW(2)-X
      GAMSUM=SUM+XM*(GAMAW(1)-GAMAW(2))*ALOG(D2/ARG)/(XIW(2)-1.)
     1 +.25*GAMAW(1)*(4.*ALOG(D2)+XP*(1.-3.*X)*ALOG(AP)+6.*X
     1 -XM*(5.+3.*X)*ALOG(ARG))
      GAMSUM=.5*GAMSUM/PI
      RETURN
      END
```

```
      SUBROUTINE WASH2(VZ,XGAM,NGAM,CR,THICK,NF,ACAP,GAM1,ISEP,DELPI,
     1 XI2)
      DIMENSION VZ(30),XGAM(30),THICK(24),ACAP(30,3),SN(30),CS(30)
      DATA NTOT /3/
      NTOP=NGAM
      IF(ISEP.EQ.1) NTOP=NTOT
      VZ(1)=DELPI*(.806853+ALOG(XI2-1.))
      NGP1=NGAM+1
      SIGN=1.
      SUM=0.
      DO 5 N=1,NF
      SIGN=-SIGN
    5 SUM=SUM+SIGN*N*THICK(N)
      FACT=2.*(CR+2.*SUM)
      SUM=0.
      SIGN=1.
      DO 6 N=1,NTOP
      SIGN=-SIGN
    6 SUM=SUM+SIGN*N*ACAP(N+1,1)
      VZ(NGP1)=2.*CR*(SUM+ACAP(1,1)) +DELPI*((XI2+1.)*ALOG((XI2+1.)*.5)/
     1(XI2-1.)-1.5)
      VZ(NGP1)=VZ(NGP1)+FACT*ACAP(1,1)
   10 NSUM=NGAM
      IF(NF.GT.NGAM) NSUM=NF
      DO 50 M=2,NGAM
      COST=XGAM(M)
      THETA=ARCOS(COST)
      DO 15 N=1,NSUM
      ANGLE=N*THETA
      SN(N)=SIN(ANGLE)
   15 CS(N)=COS(ANGLE)
      SINT=SN(1)
      COTT=COST/SINT
      ST=0.
      STPR=0.
      SA=0.
      SAPR=0.
      DO 16 N=1,NF
      ST=ST+THICK(N)*SN(N)
   16 STPR=STPR+THICK(N)*(COTT*SN(N)+N*CS(N))
      DO 17 N=1,NTOP
      NP1=N+1
      SA=SA+ACAP(NP1,1)*SN(N)
   17 SAPR=SAPR+N*ACAP(NP1,1)*CS(N)
      H=.25*GAM1*(1.+COST)*(3.*COST-1.)
      HPR=-GAM1*SINT*(1.+1.5*COST)
      FACT1=CR+STPR
      FACT2=CR*(1.-COST)+SINT*ST
      TERM1=ACAP(1,1)*(1.-COST)+SINT*(H+SA)
      TERM2=ACAP(1,1)+COTT*(H+SA)+HPR+SAPR
   40 XP=1.+COST
      XM=1.-COST
   50 VZ(M)=FACT1*TERM1+FACT2*TERM2+DELPI*(1.5*COST+.25*(1.-3.*COST)*XP*
     1ALOG(XP/XM)+(XI2-COST)*ALOG((XI2-COST)/XM)/(XI2-1.))
      RETURN
      END
```

```
      SUBROUTINE UNPOP(NGP1,AR,ALAM,AFACT,RMAT,CMAT,XGAM,AS,MX,NZ,XC,
     1 UE,NDIMC,AFAC2,AR2)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XIW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      DIMENSION              VZ(30),AR2(30)
      DIMENSION AR(30),ALAM(30),XGAM(30),AS(30,30),XC(300),UE(300,3)
      DOUBLE PRECISION RMAT(130),CMAT(NDIMC,NDIMC)
      DO 5 M=1,NGP1
      SUB=AR(M)-ALAM(M)*AFACT/3.
      RMAT(M)=SUB
      CMAT(M,1)=1.
      CMAT(M,2)=XGAM(M)
      DO 5 N=2,NGAM
5     CMAT(M,N+1)=AS(M,N)
      CALL ALSOL(NGP1, CMAT, RMAT, NDIMC)
      DO 10 N=1,NGP1
10    ACAP(N,1)=RMAT(N)
      CALL WASH2(VZ,XGAM,NGAM,CR,THICK,NF,ACAP,0.,0,0.,2.)
      DO 16 M = 1, NGP1
      CMAT(M,1) = 1.
      CMAT(M,2) = XGAM(M)
      RMAT(M)=   VZ(M)-ALAM(M)*AFAC2/3.+AR2(M)
      DO 16 N = 3, NGP1
16    CMAT(M, N) = AS(M, N-1)
      CALL ALSOL(NGP1, CMAT, RMAT, NDIMC)
      DO 17 M=1,NGP1
17    ACAP2(M,1)=RMAT(M)
      GAMAW(1)=0.
      GAMAW2(1)=0.
      DO 15 M=1,MX
      SIGN=1.
      IF(M-NZ) 12,14,14
12    SIGN=-SIGN
14    CALL QECAL(UE(M,1),XC(M),SIGN,0)
15    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRIMES(CP,TP,THETA,CR,THICK,CAMBR,NF)
      DIMENSION THICK(24),CAMBR(24)
      SST=0.
      SCT=0.
      SSR=0.
      SCR=0.
      ANGLE=0.
      COUNT=0.
      DO 5 N=1,NF
      ANGLE=ANGLE+THETA
      COUNT=COUNT+1.
      CN=COUNT*COS(ANGLE)
      SN=SIN(ANGLE)
      SST=SST+THICK(N)*SN
      SCT=SCT+THICK(N)*CN
      SSR=SSR+CAMBR(N)*SN
    5 SCR=SCR+CAMBR(N)*CN
      COST=COS(THETA)
      SINT=SIN(THETA)
      TP=CR*(COST-COST**2+SINT**2)+SINT*(2.*COST*SST+SINT*SCT)
      CP=COST*SSR+SINT*SCR
      RETURN
      END
```

```
      SUBROUTINE ATTPR(PREC,ASZ,AS,AR,CMAT,RMAT,NGP1,AR2,NDIMC,XGAM,UTU,
     1 PRECS)
      COMMON /LOADS/ CR,CRHAT,THICK(24),THAT(24),NF,ACAP(30,3),ACAP2(30,
     13),NGAM,GAMAW(1000),GAMAW2(1000),XIW(1000),NWAKE,XSEP,XATT,BCAP(10
     10,3),NSIG,RDBB,AA,BB,AKK,SS,SSLAM,ZS,UT,UN,UINF,XSIG(100),XSIGA(10
     10),XSIGB(100),NSIGA,NSIGB,DXI
      DIMENSION XGAM(30), ASZ(30),AS(30,30),AR(30),VZ(30),AR2(30)
      DOUBLE PRECISION RMAT(130), CMAT(NDIMC,NDIMC)
      DATA PI /3.14159/
      SAVE=XSIG(NSIG+1)
      XATS=XSIGA(NSIGA+1)
      IF(SAVE.LT.XATS) GO TO 20
      PREC=PRECS*(1.-SAVE)/(1.-XATS)
      RETURN
20    DO 50 M=1,NGP1
      CMAT(M,1)=ASZ(M)
      RMAT(M)=AR(M)
      DO 25 N=1,NGAM
25    CMAT(M,N+1)=AS(M,N)
50    CONTINUE
      CALL ALSOL(NGP1, CMAT, RMAT, NDIMC)
      DO 75 M=1,NGP1
75    ACAP(M,1)=RMAT(M)
      GAMAW(1)=GAM1(ACAP,DXI,PI)
      DDEL=UTU*GAMAW(1)
      DELPI=DDEL/PI
      XSIG(NSIG+1)=2.
      CALL WASH2(VZ,XGAM,NGAM,CR,THICK,NF,ACAP,GAMAW(1),0,DELPI,XIW(2))
      DO 60 M = 1, NGP1
      CMAT(M, 1) = ASZ(M)
      RMAT(M)=    VZ(M)+AR2(M)
      DO 60 N = 2, NGP1
60    CMAT(M, N) = AS(M, N-1)
      CALL ALSOL(NGP1, CMAT, RMAT, NDIMC)
      DO 65 M=1,NGP1
65    ACAP2(M,1)=RMAT(M)
      GAMAW2(1) = GAM1(ACAP2, DXI, PI)+DDEL
      CALL CPC(PREC,SAVE,1.,0)
      XSIG(NSIG+1)=SAVE
      RETURN
      END
```

```
      SUBROUTINE TAYLOR(XSEP,UE,XC,X,NZ,MX)
      DIMENSION UE(300,3),XC(300),X(300)
      DO 3 M=NZ,MX
      IF(XSEP.LT.XC(M)) GO TO 4
3     CONTINUE
4     MS=M-1
      QSEP=UE(M,1)
      XSTOP=X(MS)
      XBSEP=XSTOP+(XSEP-XC(MS))*(X(M)-XSTOP)/(XC(M)-XC(MS))
      QMAX=0.
      DO 5 M=NZ,MS
      IF(UE(M,1).LT.QMAX) GO TO 5
      QMAX=UE(M,1)
      MQMAX=M
5     CONTINUE
      SMAX=0.
      DO 6 M=MQMAX,MS
      MP=M+1
      SLOPE=(UE(M,1)-UE(MP,1))/(X(MP)-X(M))
      IF(SLOPE.LT.SMAX) GO TO 6
      SMAX=SLOPE
      MSMAX=M
6     CONTINUE
      IF(MSMAX.LT.MS) GO TO 61
      WRITE(6,85) XC(MS)
85    FORMAT(//10X,'POSITION OF MAX SLOPE DOWNSTREAM OF X =',E13.5,' SO
     1NO SMOOTHING NEEDED')
      RETURN
61    MSM=MS-1
      DO 7 M=MSMAX,MSM
      SLOPS=SLOPE
      MP=M+1
      SLOPE=(UE(M,1)-UE(MP,1))/(X(MP)-X(M))
      TLAM=(UE(M,1)-QSEP)/(XSTOP-X(M))
      IF(TLAM.GT.SLOPE) GO TO 8
7     CONTINUE
      WRITE(6,88)
88    FORMAT(//10X,'SLOPE MONOTONIC - NO SMOOTHING NEEDED')
      RETURN
8     SLOPE=SLOPS
      K=M-1
      KP=M
      XKP=X(KP)
      XIS=XBSEP-XKP
      QKP=UE(KP,1)
      QDIFF=QKP-QSEP
      B=2.*QDIFF/XIS-SLOPE
      KPP=KP+1
      DO 9 M=KPP,MS
      XI=X(M)-XKP
9     UE(M,1)=QSEP+(QDIFF+B*XI)*(1.-XI/XIS)**2
      WRITE(6,99) XC(KP),XC(MS)
99    FORMAT(//10X,'FLOW SMOOTHED BETWEEN X =',E13.5,' AND',E13.5/)
      RETURN
      END
```

```
      SUBROUTINE ELDER(BCAP,XSIG,NSIG,UINF,ELD,Y,YMAX,AZ)
      DIMENSION BCAP(100,3),XSIG(100)
          DATA   PI/ 3.1415926/
      BCAP(NSIG+1,1)=0.
      XS=XSIG(1)
      XZ=XSIG(NSIG+1)
      IF(XZ-1.) 16,16,1
1     DEADL=XZ-XS
        NSIGM= NSIG-1
          DEADL2= DEADL / 2.
      YMAX=1.E-10
        Y= PI * DEADL2 *( BCAP(1,1)+ BCAP(2,1) /2.)
         DO 30 J= 2,1001
         T= PI/1000. * (J-1)
            YX= BCAP(1,1) *( PI-T - SIN(T)     )
     1 + .5 * BCAP(2,1)  *( PI -T +.5* SIN(T*2.) )
            SUM=0.
             DO 31 I= 2, NSIGM
31          SUM = SUM + BCAP(I+1,1) * ( SIN( (I-1)* T)/(I-1)
     1 - SIN( (I+1) *T)/(I+1)           )
           YX= DEADL2 *( YX-.5*SUM)
          IF( YX .GT. YMAX) YMAX=YX
30           CONTINUE
      ELD=Y/YMAX
      IF(ABS(ELD)-UINF) 20,20,12
12    IF(ELD) 14,16,16
14    ELD=-UINF
      GO TO 20
16    ELD=UINF
20    CONTINUE
      WRITE(6,80) ELD
80    FORMAT(/10X,7HELDOT =E13.5/)
      RETURN
      END
```

```
      SUBROUTINE REATT(UC,V,X,Y,MX,NY,RY,DRY,UE,X5,DEL5,MST,REB)
      DIMENSION UC(100,3),V(100,2),Y(100)
      DIMENSION X(300),UE(300,3)
      DIMENSION TAB1(24),TAB2(24),TAB3(24),TAB4(24),XITAB(24)
      DATA TAB1 /24.98,23.29,21.04,19.33,17.61,15.29,13.46,11.54,10.36,9
     1.38,8.35,7.32,6.29,5.31,4.4,3.57,2.22,1.26,.66,.31,.14,.01,0.,0./
      DATA TAB2 /20.05,18.85,17.25,16.04,14.8,13.12,11.77,10.3,9.36,8.65
     1,7.95,7.2,6.43,5.66,4.9,4.18,2.89,1.86,1.11,.62,.32,.04,0.,0./
      DATA TAB3 /16.65,15.8,14.67,13.8,12.91,11.66,10.65,9.48,8.71,8.11,
     17.59,7.01,6.41,5.77,5.13,4.5,3.31,2.28,1.48,.9,.51,.09,.01,0./
      DATA TAB4 /10.12,10.05,9.93,9.78,9.58,9.17,8.72,8.08,7.6,7.2,6.85,
     16.53,6.18,5.79,5.36,4.91,3.98,3.05,2.21,1.5,.95,.22,.03,0./
      DATA XITAB /.0001,.0002,.0005,.001,.002,.005,.01,.02,.03,.04,.05,.
     106,.07,.08,.09,.1,.12,.14,.16,.18,.2,.25,.3,.35/
    3 FORMAT(///40X,23HAT REATTACHMENT, BETA =E13.5)
      MOUT=6
      RTR=SQRT(REB)
      UC(1,2)=0.
      UC(1,3)=0.
      V(1,1)=0.
      V(1,2)=0.
      DO 5 M=1,MX
      IF(X5-X(M)) 4,4,5
    4 MST=M+2
      GO TO 6
    5 CONTINUE
    6 XA=X(MST-2)
      XB=X(MST-1)
      UA=UE(MST-2,1)
      UB=UE(MST-1,1)
      ZA=ALOG(UA*DEL5*REB)
      PGRAD=2.*(UA-UB)/((UA+UB)*(XB-XA))
      BETM2=(.0974-SQRT(DEL5*PGRAD))/(.0249+.004565*ZA)
      IF(BETM2-1.) 8,7,7
    7 BETM2=1.
      GO TO 10
    8 IF(BETM2-.3) 9,9,10
    9 BETM2=.3
   10 BETA=1./(BETM2*BETM2)
      WRITE(MOUT,3) BETA
      AGAM=.0974*BETM2-.0249/BETA
      BGAM=.004565/BETA
      AH=1.-(5.3+3.9*BETM2)*(.0974-.0249*BETM2)
      BH=BETM2*(5.3+3.9*BETM2)*.004565
      GAMA=AGAM-BGAM*ZA
      DERIV=JA*REB*EXP(-ZA)*GAMA*GAMA*(1.+BETA*(1.+AH+BH*ZA))/(AH+BH+BH*
     1ZA)
      ZB=ZA+DERIV*(XB-XA)
      DELB=EXP(ZB)/(UB*REB)
      GAMB=AGAM-BGAM*ZB
      DELL=.35*DELB*RTR*BETM2/GAMB
   11 IF(DELL-Y(NY-3)) 14,12,12
   12 RY=RY+DRY
      CALL YSET(RY,Y(2),NY,Y)
```

```
      GO TO 11
14    IF(BETA-4.) 102,101,101
101   TERPB=1.-4./BETA
      INDEX=3
      GO TO 110
102   IF(BETA-2.) 104,103,103
103   TERPB=.5*BETA-1.
      INDEX=2
      GO TO 110
104   TERPB=BETA-1.
      INDEX=1
110   K=0
      TERP1=1.-TERPB
50    K=K+1
      GO TO (16,17,99),K
16    G=GAMA
      DELTA=DEL5
      UEDGE=JA
      L=3
      GO TO 18
17    G=GAMB
      DELTA=DELB
      UEDGE=JB
      L=2
18    XICO=G/(DELTA*RTR*BETM2)
      UCOW=RTR*(UEDGE*G)**2
      EFCO=G/BETM2
      NLAM=NY
      DO 75 N=2,NY
      XI=Y(N)*XICO
      IF(XI-.35) 20,19,19
19    UC(N,L)=UEDGE
      GO TO 75
20    CALL TERPF(XI,INDEX,TAB1,TAB2,TAB3,TAB4,XITAB,FP1)
      INDP1=INDEX+1
      CALL TERPF(XI,INDP1,TAB1,TAB2,TAB3,TAB4,XITAB,FP2)
      FP=TERP1*FP1+TERPB*FP2
      UC(N,L)=UEDGE*(1.-EFCO*FP)
      IF(N-NLAM) 21,75,75
21    ALTER=UCOW*Y(N)
      IF(ALTER-UC(N,L)) 33,33,32
32    UC(N,L)=ALTE
      GO TO 75
33    NLAM=N
75    CONTINJE
      GO TO 50
99    DO 60 K=2,3
      SAVE2=0.
      DO 60 N=3,NY
      SAVE1=JC(N-1,K)
      UC(N-1,K)=(SAVE2+SAVE1+UC(N,K))/3
60    SAVE2=SAVE1
      DUDX=0.
      COD=.5/(XB-XA)
      DO 65 N=2,NY
```

```
      DUDXP=COD*(UC(N,2)-UC(N,3))
      V(N,1)=V(N-1,1)-(Y(N)-Y(N-1))*(DUDXP+DUDX)
      V(N,2)=V(N,1)
   65 DUDX=DUDXP
      RETURN
      END
```

```
      SUBROUTINE ELPIT(ALPH1,ALPH2,EMI,TORF,THETZ,UINF,DXI,CMPA,CMPAS)
      SAVET=ALPH1
      STEP=TORF*DXI
      SINS=SIN(STEP)
      COSS=COS(STEP)
      CONST=2.*EMI*(UINF/TORF)**2
      ALPH1=THETZ+(ALPH1-THETZ)*COSS+ALPH2*SINS/TORF+CONST*(2.*CMPA-CMPA
     1S)*(1.-COSS)+CONST*(CMPAS-CMPA)*(SINS-STEP*COSS)/(TORF*DXI)
      ALPH2=ALPH2*COSS-TORF*SINS*(SAVET-THETZ)+CONST*(CMPA-CMPAS)*(1.-CO
     1SS)/DXI+CONST*CMPA*TORF*SINS
      RETURN
      END
```

```
      SUBROUTINE VWASH(BARG,H,S,NVOR,X1,UINF,VZIP,XGAM,NGP1,DXI)
      DIMENSION VZIP(30),XGAM(30)
      DO 10 N=1,NGP1
      DIFF=XGAM(N)-X1
      SUM=0.
      DO 5 K=1,NVOR
      SUM=SUM+DIFF/(DIFF*DIFF+H)
5     DIFF=DIFF-S
10    VZIP(N)=VZIP(N)+SUM*BARG
      RETURN
      END
```

```
      SUBROUTINE WASH(XGAM,NGAM,TIME,ALPH1,ALPH2,HEAVE,ARCT,FREQF,PHIH,U
     1INF,CAMBR,NF,VZIP,MOTR,INDV)
      DIMENSION XGAM(30),VZIP(30),CAMBR(24)
      NGP1 = NGAM+1
      ANGLE = FREQF*TIME
      GO TO (108,120), INDV
108   GO TO (110,120),MOTR
110   CONST =-ALPH2*COS(ANGLE)*UINF+HEAVE*COS(ANGLE+PHIH)+ALPH1*UINF
      FACT =-ALPH2*FREQF*SIN(ANGLE)*UINF
      GO TO 130
120   CONST=UINF*ALPH1+HEAVE
      FACT=-UINF*ALPH2
130   DO 10 M=1,NGP1
      X=XGAM(M)
      THETA = ARCT(X)
      SUM=0.
      COUNT=0.
      DO 20 N=1,NF
      COUNT=COUNT+1.
20    SUM=SUM+COUNT*CAMBR(N)*COS(COUNT*THETA)
      IF(M-1) 2,4,2
2     IF(NGP1-M) 3,4,3
4     SUM = SUM + SUM
      GO TO 50
3     COUNT = 0.
      COTT = X/SIN(THETA)
      DO 30 N=1,NF
      COUNT = COUNT+THETA
30    SUM=SUM+COTT*CAMBR(N)*SIN(COUNT)
50    VZIP(M) = UINF*SUM+CONST+FACT*(AROT-X)
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MIXER(FPRES,PREC,UINF,UDOT,THICK,NF,XBSIG,NSIG,INDT,DEL
     11,THET1,REB,USEP,X4,CP4,RDBB)
      DIMENSION FPRES(100),THICK(24),XBSIG(100)
      FCAP(X)=-19.556*X+107.535*X*X-336.33*X**3+508.1*X**4-295.96*X**5
      UI1(X)=-.46532*X+.68425*X*X-.45293*X**3+.6592*X**4
      UI2(X)=-.045929*X-1.91615*X*X+2.91843*X**3-5.42125*X**4
      CR=SQRT(RDBB)/4.
      DIST=.5*(XBSIG(2)-XBSIG(1))
      XSEP=XBSIG(1)-DIST
      XATT=XBSIG(NSIG)+DIST
C
C  IF INDT IS NONZERO, THE BOUNDARY LAYER IS TURBULENT
C  AT SEPARATION.
C
      CALL H4X4(INDT,XSEP,DEL1,THET1,XATT,REB,USEP,X3,H3,X4,H4)
      IF(XSEP-1.) 24,25,25
 25   CP4=0.
      GO TO 27
 24   URAT=EXP(-.08712-UI1(H4)-.24723*(.3255+UI2(H4)))
      CP4=1.-(1.-PREC)/URAT**2
      DEADL=XATT-XSEP
      IF(DEADL-2.) 5,6,6
 5    G=(.5*DEADL)**2
      GO TO 7
 6    G=1.
 7    CP4=PREC+(CP4-PREC)*(1.-G*XSEP)
 27   CONTINJE
            CP4LIN=2. * (1.-SQRT(1.-CP4))
              PRLIN=2.*(1.- SQRT(1.-PREC))
           COEF= ( PRLIN-CP4LIN)/ (XATT- X4)
      CZ=2.*JDOT/UINF
      C2 = -2.*UINF
      DO 20 M=1,NSIG
      SUM=0.
      X=XBSIG(M)
      IF(X-1.) 2,2,3
 2    THETA = ARCT(X)
      COST=COS(THETA)
      SINT=SIN(THETA)
      COUNT=0.
      DO 10 N=I,NF
      COUNT=COUNT+1.
      ANGLE=COUNT*THETA
 10   SUM=SUM+THICK(N)*(COUNT*SINT*SIN(ANGLE)-2.*COST*COS(ANGLE))
      SUM=-4.*UINF*(SUM+CR*(1.-2.*COST)),
      GO TO 35
 3    SQSS=SQRT(X*X-1.)
      XRAD=1./(X+SQSS)
      COUNT=1.
      FACT=XRAD
      DO 30 N=2,NF
      COUNT=COUNT+1.
      FACT=FACT*XRAD
 30   SUM=SUM+THICK(N)*FACT*(2.*X-COUNT*SQSS)
```

```
      SUM=SUM*2.+2.*CR*(X+X-1.-SQSS*(1.+X+X)/(1.+X))+3.*THICK(1)*XRAD**2
      SUM=SUM*UINF
35    CP=CP4LIN
      IF(X-X4) 55,50,50
50    CP=CP+(X-X4)*COEF
55    CONTINUE
      FPRES(M)=-UINF*CP+SUM
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SECT(XU,YU,XL,YL,NOFF,NF,RDBC,ST,SC)
C   PROGRAM TJ COMPUTE CJEFFICIENTS TN AND CN OF THE FOURIER SERIES
C   REPRESENTATION OF SECTION THICKNESS AND CAMBER DISTRIBUTIONS.
      DIMENSION XU(30),YU(30),XL(30),YL(30),YUC(30),YLC(30),ST(24),SC(24
     1),DUM(50),TBAR(50),CBAR(50)
 11   FORMAT(1H1,4X,6HRDBC =E13.5/4X,7HRCDBC =E13.5)
 12   FORMAT(////47X,26HINPUT AND COMPUTED OFFSETS/)
 13   FORMAT(19X,4HX1/C,12X,4HYU/C,11X,5HYUC/C,20X,4HX1/C,12X,4HYL/C,11X
     1,5HYLC/C/)
 14   FORMAT(8X,3F16.5,8X,3F16.5)
      NA=6
      RNA=6.
      RNF=FLJAT(NF)
      MOUT=6
      PI=3.14159
      DELT=PI/(2.*RNF)
      NTC=2*NF-1
      NINT=NTC+2
      NSIMP=VTC+1
      THETA=0.
      DO 89 K=1,NTC
      THETA=THETA+DELT
      X1=.5*(1.+COS(THETA))
      DO 90 LAM=2,NOFF
      IF(X1-XU(LAM)) 110,90,90
 110  YUINT=YU(LAM-1)+(X1-XU(LAM-1))*(YU(LAM)-YU(LAM-1))/(XU(LAM)-XU(LAM
     1-1))
      GO TO 111
 90   CONTINJE
 111  DO 80 LAM=2,NOFF
      IF(X1-XL(LAM)) 210,80,80
 210  YLINT=YL(LAM-1)+(X1-XL(LAM-1))*(YL(LAM)-YL(LAM-1))/(XL(LAM)-XL(LAM
     1-1))
      GO TO 112
 80   CONTINJE
 112  TBAR(K+1)=.5*(YUINT-YLINT)
 89   CBAR(K+1)=.5*(YUINT+YLINT)
      TBAR(1)=0.
      CBAR(1)=0.
      TBAR(NINT)=0.
      CBAR(NINT)=0.
      SAVT2=0.
      SAVC2=0.
      DO 39 I=2,NSIMP
      SAVT1=SAVT2
      SAVC1=SAVC2
      SAVT2=TBAR(I)
      SAVC2=CBAR(I)
      TBAR(I)=(SAVT1+SAVT2+TBAR(I+1))/3.
 39   CBAR(I)=(SAVC1+SAVC2+CBAR(I+1))/3.
      TTA=TBAR(NA)
      TTB=TBAR(NA+1)
      TTC=TBAR(NA+2)
      TAA=DELT*(RNA-1.)
```

```
      TBB=TAA+DELT
      TCC=TBB+DELT
      XA=.5*COS(TAA)
      XB=.5*COS(TBB)
      XC=.5*COS(TCC)
      SLOPE=((TTC-TTB)*(XB-XA)/(XC-XB)+(TTB-TTA)*(XC-XB)/(XB-XA))/(XC-XA
     1)
      THETA=0.
      COSB=COS(TBB)
      DO 456 I=2,NA
      THETA=THETA+DELT
      COST=COS(THETA)
      TBAR(I)=(SQRT(1.-COST)/(1.-COSB)**1.5)*(TTB*(1.+COST-2.*COSB)/(1.-
     1COSB)+.5*SLOPE*(COST-COSB))
  456 TBAR(I)=TBAR(I)*(1.-COST)
      NLE=2*NF+1-NA
      TTA=TBAR(NLE+1)
      TTB=TBAR(NLE)
      TTC=TBAR(NLE-1)
      TAA=PI-DELT*(RNA-1.)
      TBB=TAA-DELT
      TCC=TBB-DELT
      XA=.5*(1.+COS(TAA))
      XB=.5*(1.+COS(TBB))
      XC=.5*(1.+COS(TCC))
      SLOPE=((TTC-TTB)*(XB-XA)/(XC-XB)+(TTB-TTA)*(XC-XB)/(XB-XA))/(XC-XA
     1)
      CAPA=SLOPE*SQRT(2.*XB/RDBC)
      CAPB=TTB/SQRT(2.*XB*RDBC)
      AS=(2.5*CAPB-.5*CAPA-2.)/XB
      BS=(.5*CAPA-1.5*CAPB+1.)/XB**2
      THETA=PI
      COEF=SQRT(2.*RDBC)
      DO 457 I=2,NA
      IND=2*NF+2-I
      THETA=THETA-DELT
      X=.5*(1.+COS(THETA))
  457 TBAR(IND)=COEF*SQRT(X)*(1.+AS*X+BS*X*X)
      COEF=COEF/4.
      THETA=0.
      DO 458 I=2,NSIMP
      THETA=THETA+DELT
      SINT=SIN(THETA)
  458 TBAR(I)=(TBAR(I)-COEF*SINT*(1.-COS(THETA)))/SINT**2
      THETA=0.
      DO 459 I=2,NSIMP
      THETA=THETA+DELT
  459 CBAR(I)=CBAR(I)/SIN(THETA)
      RKK=0.
      DO 59 K=1,NF
      RKK=RKK+1.
      THETA=0.
      DO 777 I=1,NINT
      DUM(I)=TBAR(I)*SIN(THETA*RKK)
  777 THETA=THETA+DELT
```

```
      CALL SIMP(NSIMP,DELT,DUM,VARY)
      ST(K)=2.*VARY/PI
      THETA=0.
      DO 888 I=1,NINT
      DUM(I)=CBAR(I)*SIN(THETA*RKK)
888   THETA=THETA+DELT
      CALL SIMP(NSIMP,DELT,DUM,VARY)
59    SC(K)=2.*VARY/PI
      DO 969 I=1,NOFF
      X=XU(I)
      CALL EVAL(NF,X,SC,ST,CB,TB,COEF)
969   YUC(I)=CB+TB
      DO 869 I=1,NOFF
      X=XL(I)
      CALL EVAL(NF,X,SC,ST,CB,TB,COEF)
869   YLC(I)=CB-TB
      RSAV=RDBC
      WRITE(MOUT,11) RSAV,RDBC
      WRITE(MOUT,12)
      WRITE(MOUT,13)
      WRITE(MOUT,14) (XU(I),YU(I),YUC(I),XL(I),YL(I),YLC(I),I=1,NOFF)
      RETURN
      END
```

```
      SUBROUTINE EVAL(NNF,XX,SSC,SST,CCB,TTB,COEF)
      DIMENSION SSC(50),SST(50)
      COST=2.*XX-1.
      COSTS=COST**2
      IF(COSTS-1.E-8) 303,304,304
304   TANT=SQRT(1./COSTS-1.)
      THE=ATAN(TANT)
      GO TO 305
303   THE=1.5708
305   IF(COST) 403,404,404
403   THE=3.14159-THE
404   ARG=0.
      SUM1=0.
      SUM2=0.
      DO 551 N=1,NNF
      ARG=ARG+THE
      SUM1=SUM1+SSC(N)*SIN(ARG)
551   SUM2=SUM2+SST(N)*SIN(ARG)
      SINT=SIN(THE)
      CCB=SUM1*SINT
      TTB=SINT*(COEF*(1.-COST)+SINT*SUM2)
      RETURN
      END
```

```
      SUBROUTINE ALSOL(NT, C, R, NDIMC)
      DOUBLE PRECISION C    (NDIMC,NDIMC), R(130)
      DOUBLE PRECISION CMAX, SAVE, SUM
      NT1 = NT-1
      DO 99 J=1,NT1
      CMAX = C(NT,J)
      L=NT
      DO 10 I=J,NT1
      IF (DABS(CMAX)-DABS(C(I,J))) 5,10,10
5     CMAX = C( I,J)
      L=I
10    CONTINJE
      DO 15 JJ=J,NT
      SAVE = C(L,JJ)
      C(L,JJ) = C(J,JJ)
15    C(J,JJ) = SAVE/CMAX
      SAVE = R(L)
      R(L) = R(J)
      R(J) = SAVE/CMAX
      JP1 = J+1
      DO 25 I=JP1,NT
      DO 20 JJ=JP1,NT
20    C(I,JJ) = C(I,JJ) - C(I,J)*C(J,JJ)
25    R(I) = R(I) - R(J)*C(I,J)
99    CONTINUE
      R(NT) = R(NT)/C(NT,NT)
      DO 150 K=1,NT1
      I=NT-K
      IP1 = I+1
      SUM = 0.
      DO 125 J=IP1,NT
125   SUM = SUM + R(J)*C(I,J)
150   R(I) = R(I) - SUM
      RETURN
      END
```

```
      FUNCTION GAM1(ACAP,DXI,PI)
      DIMENSION ACAP(30,3)
      GAM1=PI*(-1.5*ACAP(1,1)-.75*ACAP(2,1)+2.*ACAP(1,2)+ACAP(2,2)-.5*AC
     1AP(1,3)-.25*ACAP(2,3))/DXI
      RETURN
      END
```

```
      SUBROUTINE EGAMI(NU,NG,A,B,XSEP,XATT,GAMMA,Y,GI)
      DIMENSION A(30,3)
      SINT=SQRT(1.-Y*Y)
      THETA=ARCT(Y)
      SUM=0.
      COUNT=1.
      DO 6 N=2,NG
      COUNT=COUNT+1.
    6 SUM=SUM+A(N+1,NU)*(SIN((COUNT+1.)*THETA)/(COUNT+1.)-SIN((COUNT-1.)
     1*THETA)/(COUNT-1.))
      GI=(3.14159-THETA+SINT)*(A(1,NU)+.5*A(2,NU))+.5*SUM-.25*GAMMA*(1.+
     1Y)*SINT*SINT
      IF(Y-XATT) 8,8,7
    7 DIFF=1.-XATT
      IF(DIFF-1.E-6) 8,8,9
    9 GI=GI+2.*B*DIFF**(-1.5)*SQRT((XATT-XSEP)*(1.-Y)*(Y-XATT))
    8 CONTINJE
      RETURN
      END
```

```
      SUBROUTINE BUBB(DEL1,THET1,REB,XC1,U1,XC5,DCP,DEL5,X,XC,MX,NZ,X5,U
     15,UE,ALTC,RENEL,USTOP)
      DIMENSION X(300),XC(300),UE(300,3)
      FCAP(X)=-19.556*X+107.535*X*X-336.33*X**3+508.1*X**4-295.96*X**5
      UI1(X)=-.46532*X+.68425*X*X-.45293*X**3+.6592*X**4
      UI2(X)=-.045929*X-1.91615*X*X+2.91843*X**3-5.42125*X**4
      FDELT(X)=EXP(2.5773-.34252*X-.4379*X*X-.076511*X**3-.0039707*X**4)
      FAICH(X)=EXP(-3.7481+.038772*X+.41967*X*X+.071046*X**3+.0032162*X*
     1*4)
      DELI(X)=-.045929*ALOG(X)-3.9242*X+.54535*X*X-1.39147*X**3-10.8425*
     1X**4
25    FORMAT(1H1,44X,31HANALYSIS OF LEADING-EDGE BUBBLE///34X,1HX,19X,1
     1HU,19X,1HH,18X,4HDISP/)
30    FORMAT(20X,4E20.5)
      MOUT=6
      H1=.25
      H5=.429
      DO 5 M=NZ,MX
      IF(XC1-XC(M)) 4,4,5
4     M1=M
      GO TO 6
5     CONTINJE
6     X1=X(M1-1)+(X(M1)-X(M1-1))*(XC1-XC(M1-1))/(XC(M1)-XC(M1-1))
      X4=X1+RENEL/(U1*REB)
      ARG=ALJG((X4-X1)/(REB*DEL1*DEL1*U1))
      H4=  .25*FAICH(ARG)
      DEL4=.58*FDELT(ARG)*DEL1
      X5=X4+10.5*DEL4*(1.-(H4/.429)**2)
      IF(U1-USTOP) 41,41,40
40    ALTL=ALTC*DEL1
      IF(X5-X1.LT.ALTL) X5=X1+ALTL
41    URAT=EXP(-.08712-UI1(H4)-.24723*(.3255+UI2(H4)))
      DCP=U1*U1*(1.-URAT**2)
      DRAT=EXP(-2.24374-FCAP(H4)+.24723*(2.0214+DELI(H4)))
      DEL5=DRAT*DEL4
      DO 7 M=NZ,MX
      IF(X5-X(M)) 16,16,7
16    M5=M
      GO TO 8
7     CONTINJE
8     FACT=(X5-X(M5-1))/(X(M5)-X(M5-1))
      FACT1=1.-FACT
      XC5=XC(M5-1)*FACT1+XC(M5)*FACT
      U5=UE(M5-1,1)*FACT1+UE(M5,1)*FACT
      WRITE(MOUT,25)
      WRITE(MOUT,30) X1,U1,H1,DEL1
      WRITE(MOUT,30) X4,U1,H4,DEL4
      WRITE(MOUT,30) X5,U5,H5,DEL5
      RETURN
      END
```

```
      SUBROUTINE YSET(R,A,NY,Y)
      DIMENSION Y(100)
      RP1=1.+R
      Y(1)=0.
      Y(2)=A
      DO 10 N=3,NY
10    Y(N)=RP1*Y(N-1)-R*Y(N-2)
      RETURN
      END
```

```
      SUBROUTINE H4X4(INDT,X1,DEL1,THET1,X5,REB,U1,X3,H3,X4,H4)
      CURLF(H)=26.703/H+305.03*ALOG(H)-2111.3*H+3327.8*H*H-2403.9*H**3
      FDELT(X)=EXP(2.5773-.34252*X-.4379*X*X-.076511*X**3-.0039707*X**4)
      FAICH(X)=EXP(-3.7481+.038772*X+.41967*X*X+.071046*X**3+.0032162*X*
     1*4)
   10 FORMAT(//20X,54HA SOLUTION FOR X4 COULD NOT BE OBTAINED IN 1000 TR
     1IALS)
      MOUT=6
C
C  IF INDT IS NONZERO, THE BOUNDARY LAYER IS TURBULENT
C  AT SEPARATION.
C
      IF(INDT) 2,5,2
    2 H3=THET1/DEL1
      X3=X1
      DEL3=DEL1
      GO TO 20
    5 X3=X1+5.E4/(U1*REB)
      ARG=ALOG((X3-X1)/(REB*DEL1*DEL1))
      H3=THET1*FAICH(ARG)/DEL1
      DEL3=.58*FDELT(ARG)*DEL1
      IF(X3-X5) 20,15,15
   15 H4=.429
      X4=X5
      GO TO 50
   20 CONTINUE
      IGO=0
      DIST=X5-X1
      UNDER=0.
      H4=H3+H3
      COEF1=DEL3*H3
      COEF2=10.5*DEL3*H3
      SUB=X3-COEF1*CURLF(H3)
   95 OVER=H4
      H4=.5*(H4+UNDER)
      X4=CURLF(H4)*COEF1+SUB
      ALTER=X5-COEF2*(1.-(H4/.429)**2)/H4
      IGO=IGO+1
      IF(X4-ALTER) 41,50,42
   41 IF(IGO-1000) 95,61,61
   42 IF(ABS(X4-ALTER)/DIST-.001) 50,50,43
   43 UNDER=H4
      H4=.5*(OVER+H4)
      X4=CURLF(H4)*COEF1+SUB
      ALTER=X5-COEF2*(1.-(H4/.429)**2)/H4
      IGO=IGO+1
      IF(X4-ALTER) 52,50,51
   51 IF(IGO-1000) 43,61,61
   52 IF(ABS(X4-ALTER)/DIST-.001) 50,50,95
   61 H4=.429
      X4=X5
      WRITE(MOUT,10)
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETSX(NSP1,XSEP,XATT,XSIG,ANGLE)
      DIMENSION XSIG(100)
      A=.5*(XSEP+XATT)
      B=.5*(XATT-XSEP)
      ARG=0.
      DO 5 N=1,NSP1
      XSIG(N)=A-B*COS(ARG)
5     ARG=ARG+ANGLE
      RETURN
      END
```

```
      FUNCTION ARCT(X)
      PI=3.14159
      IF(ABS(X)-1.E-6) 1,2,2
 1    ARCT=.5*PI
      GO TO 6
 2    IF(X+.99999) 3,4,4
 3    ARCT=PI
      GO TO 6
 4    ARCT=ATAN(SQRT(1.-X*X)/X)
      IF(ARCT) 5,6,6
 5    ARCT=ARCT+PI
 6    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SCAL(SBL,NSBL,FRZ,ARR,RDBB)
      DIMENSION SBL(300)
      DELZ=FRZ*RDBB
      EN=ARR/FRZ
      DO 5 N=1,300
      IF(EN-N) 4,4,5
4     NE=N
      GO TO 6
5     CONTINUE
6     NG=NSBL-NE
      EN=FLOAT(NG)
      NGM1=NG-1
      SBL(1)=0.
      DO 7 N=2,NE
7     SBL(N)=SBL(N-1)+DELZ
      FRACT=2.2/DELZ
      FRAC1=FRACT-1.
      R=FRACT**(1./FLOAT(NGM1))
8     SAVE=R
      R=R-(R**NG-FRACT*R+FRAC1)/(EN*R**NGM1-FRACT)
      IF(ABS(SAVE-R)-1.E-6) 9,9,8
9     RP1=R+1.
      DO 10 N=NE,NSBL
10    SBL(N+1)=RP1*SBL(N)-R*SBL(N-1)
      RETURN
      END
```

```
      SUBROUTINE TERPF(XI,J,TAB1,TAB2,TAB3,TAB4,XITAB,FP)
      DIMENSION TAB1(24),TAB2(24),TAB3(24),TAB4(24),XITAB(24)
      IF(XI-.0001) 2,2,10
2     GO TO (3,4,5,6),J
3     FP=2.53-2.439*ALOG(XI)
      GO TO 99
4     FP=3.54-1.725*ALOG(.7071*XI)
      GO TO 99
5     FP=4.58-1.2195*ALOG(.5*XI)
      GO TO 99
6     FP=10.12
      GO TO 99
10    DO 12 N=1,24
      IF(XI-XITAB(N)) 11,11,12
11    NX=N
      GO TO 13
12    CONTINJE
13    TX=(XI-XITAB(NX-1))/(XITAB(NX)-XITAB(NX-1))
      TX1=1.-TX
      GO TO (14,15,16,17),J
14    FP=TX1*TAB1(NX-1)+TX*TAB1(NX)
      GO TO 99
15    FP=TX1*TAB2(NX-1)+TX*TAB2(NX)
      GO TO 99
16    FP=TX1*TAB3(NX-1)+TX*TAB3(NX)
      GO TO 99
17    FP=TX1*TAB4(NX-1)+TX*TAB4(NX)
99    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SIMP(NS,DX,ORD,FIND)
      DIMENSION ORD(50)
C          INTEGRATION OF NS + 1 EQUALLY SPACED ORDINATE VALUES
C          BY SIMPSON'S RULE.  NS MUST BE EVEN
      SUM = 0.
      DO 88 I=2,NS,2
   88 SUM = SUM + 2.*ORD(I-1) + 4.*ORD(I)
      FIND = DX*(SUM - ORD(1) + ORD(NS+1))/3.
      RETURN
      END
```

```
      SUBROUTINE CORDX(NSBL,NZ,RDBB,SBL,X,XC)
C
C  BOUNDARY LAYER COORDINATES AND CORRESPONDING CHORDAL
C  COORDINATES ARE COMPUTED HERE.
C
      DIMENSION SBL(300),X(300),XC(300)
  336 FORMAT(//10X,31HITERATION TO COMPUTE XC FOR M =I5,32H DID NOT CONV
     1ERGE IN 1000 STEPS.)
  337 FORMAT(1H1,25X,1HM,20X,1HS,25X,1HX,24X,2HXC//)
  338 FORMAT(22X,I5,3E25.5)
      MOUT=6
      MX = NSBL + NZ - 1
      RZERO = RDBB/2.
      XC(NZ) = -1.
      DO 255 M=1,NZ
      MM = NZ + 1 - M
  255 X(M) = SBL(NZ) - SBL(MM)
      DO 256 M=NZ,MX
      MM = M + 1 - NZ
  256 X(M) = SBL(NZ) + SBL(MM)
      DO 257 M=1,MX
      IF(NZ-M) 333,257,335
  333 K = M + 1 - NZ
      GO TO 334
  335 K = NZ - M + 1
  334 XC(M) = -1. + SBL(K)
      IF(SBL(K)-RZERO) 341,341,342
  341 XC(M) = -1. + SBL(K)**2/(4.*RZERO)
  342 CONTINJE
      DO 258 L=1,1000
      SAVE = XC(M)
      CALC1 = SQRT((1.+XC(M))/RZERO)
      CALC2 = SQRT(1.+(1.+XC(M))/RZERO)
      XC(M)=XC(M)+CALC1*(SBL(K) -  RZERO*(CALC1*CALC2+ALOG(CALC1+CALC2))
     1)/CALC2
      IF(ABS(SAVE-XC(M))-1.E-6) 257,257,258
  258 CONTINJE
      WRITE(MOUT,336) M
  257 CONTINJE
      WRITE(MOUT,337)
      DO 264 M=1,MX
      IF(NZ-M) 261,261,262
  262 K=NZ-M+1
      GO TO 263
  261 K=M+1-NZ
  263 WRITE(MOUT,338) M,SBL(K),X(M),XC(M)
  264 CONTINJE
      RETURN
      END
```

```
      SUBROUTINE PGRAD(M,X,UE,DXI,PRESS,SA,SB,SC,SR,SS)
C
C  SUBROUTINE FOR CALCULATION OF PRESSURE GRADIENT AND
C  DERIVATIVE COEFFICIENTS.
C
      DIMENSION X(300),UE(300,3)
      D1Z=X(M+1)-X(M)
      D2Z=X(M+2)-X(M)
      D21=X(M+2)-X(M+1)
      D1M1=X(M+1)-X(M-1)
      DZM1=X(M)-X(M-1)
      XIM=D1Z/(D2Z*D21)
      ETAM=1./D1Z-1./D21
      ZETAM=D21/(D1Z*D2Z)
      PRESS = (3.*UE(M+1,1)-4.*UE(M+1,2)+UE(M+1,3))/(2.*DXI)+UE(M+1,1)*(
     1XIM*UE(M+2,1)+ETAM*UE(M+1,1)-ZETAM*UE(M,1))
      SA=1./D1Z+1./D1M1
      SB=D1M1/(D1Z*DZM1)
      SC=D1Z/(D1M1*DZM1)
      SR=D1M1/DZM1
      SS=D1Z/DZM1
      RETURN
      END
```

```
      SUBROUTINE TRANS(UPRIM,PRESS,THETA,REB,UC,NY,FLAM,XFLAM,LAMQ)
C
C SUBROUTINE TO TEST FOR TRANSITION IN A LAMINAR BOUNDARY LAYER.
C
      DIMENSION UC(100,3),FLAM(10),XFLAM(10)
      F(X) = .11746 - 1.0582E-3*X - 1.1023E-4*X*X
      TKAY = PRESS*REB*THETA**2/UC(NY,2)
      IF(TKAY-.077) 2,2,99
    2 IF(ABS(TKAY)-.0001) 3,3,4
    3 ARG = TKAY*72.48
      GO TO 5
    4 ARG = 0.
      DO 6 N=1,1000
      SAVE = ARG
      ARG = ARG - (ARG*F(ARG)**2-TKAY)/(F(ARG)*(.11746-ARG*3.1746E-3 - A
     1RG*ARG*5.5115E-4))
      IF(ABS(1.-SAVE/ARG)-1.E-6) 7,7,6
    6 CONTINJE
    7 IF(ARG+11.) 8,8,5
    8 EF = 1.75
      GO TO 10
    5 DO 15 N=1,10
      IF(ARG-XFLAM(N)) 24,24,15
   24 NBAR = N
      GO TO 16
   15 CONTINJE
   16 EF = FLAM(NBAR-1)+(ARG-XFLAM(NBAR-1))*(FLAM(NBAR)-FLAM(NBAR-1))/(X
     1FLAM(NBAR)-XFLAM(NBAR-1))
   10 B = .5*EF
      A = 3.36*(UPRIM/UC(NY,2))**2
      RTH = F(ARG)*(SQRT(B*B+9860.*A)-B)/A
      IF(REB*THETA-RTH) 99,50,50
   50 LAMQ = 0
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CAPS(ITER,N,CAPG,CAPH,CAPJ,CAPK,SR,SS,SD,SE,SF,VISC,V,U
     1C)
      DIMENSION CAPG(100),CAPH(100),CAPJ(100),CAPK(100)
      DIMENSION VISC(100,2),V(100,2),UC(100,3),SD(100),SE(100),SF(100)
      IF(ITER) 4,2,4
 2    CAPG(N)= SR*V(N,1) - SS*V(N,2)
      CAPH(N)=SR*VISC(N,1)-SS*VISC(N,2)
      CAPJ(N)=SR*(SD(N)*VISC(N+1,1)+SE(N)*VISC(N,1)-SF(N)*VISC(N-1,1))-S
     1S*(SD(N)*VISC(N+1,2)+SE(N)*VISC(N,2)-SF(N)*VISC(N-1,2))
      CAPK(N)= SR*UC(N,2)-SS*UC(N,3)
      GO TO 6
 4    CAPG(N)=.5*(CAPG(N)+V(N,1))
      CAPH(N)=.5*(CAPH(N)+VISC(N,1))
      CAPJ(N)=.5*(CAPJ(N)+SD(N)*VISC(N+1,1)+SE(N)*VISC(N,1)-SF(N)*VISC(N
     1-1,1))
      CAPK(N)=.5*(CAPK(N)+UC(N,1))
 6    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TERP(YIN,YBASE,VARY,NY,VALUE)
C
C  SUBROUTINE FOR DETERMINING INTERPOLATED VALUE OF THE
C    FUNCTION VARY AT Y = YIN.
C
      DIMENSION YBASE(100),VARY(100)
      IF(YIN-YBASE(NY-1)) 2,3,3
 3    VALUE = VARY(NY)
      GO TO 10
 2    DO 15 N=1,NY
      IF(YIN-YBASE(N)) 24,24,15
 24   NBAR=N
      GO TO 16
 15   CONTINUE
 16   D21=YBASE(NBAR)-YBASE(NBAR-1)
      D31=YBASE(NBAR+1)-YBASE(NBAR-1)
      D32=D31-D21
      D3A=YBASE(NBAR+1)-YIN
      D2A=YBASE(NBAR)-YIN
      DA1=YIN-YBASE(NBAR-1)
      VALUE=D3A*D2A*VARY(NBAR-1)/(D21*D31)+D3A*DA1*VARY(NBAR)/(D21*D32)-
     1D2A*DA1*VARY(NBAR+1)/(D31*D32)
 10   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE YDIFF(NY,ALPHA,BETA,GAMMA,DELTA,SD,SE,SF,C2,C3,C4,Y)
      DIMENSION ALPHA(100),BETA(100),GAMMA(100),DELTA(100)
      DIMENSION SD(100),SE(100),SF(100),Y(100)
      NV=NY-2
      NVP1=NV+1
      DO 40 N=2,NV
      ALPHA(N) = 2.*(2.*Y(N)-Y(N-1)-Y(N+1))/((Y(N+2)-Y(N-1))*(Y(N+2)-Y(N
     1+1))*(Y(N+2)-Y(N)))
      DELTA(N) = 2.*(Y(N+2)+Y(N+1)-2.*Y(N))/((Y(N+2)-Y(N-1))*(Y(N+1)-Y(N
     1-1))*(Y(N)-Y(N-1)))
      BETA(N) = (DELTA(N)*(Y(N)-Y(N-1))**3-ALPHA(N)*(Y(N+2)-Y(N))**3)/(Y
     1(N+1)-Y(N))**3
      GAMMA(N) = -ALPHA(N)-BETA(N)-DELTA(N)
   40 CONTINJE
      DO 39 N=2,NVP1
      SD(N) = (Y(N)-Y(N-1))/((Y(N+1)-Y(N-1))*(Y(N+1)-Y(N)))
      SE(N) = 1./(Y(N)-Y(N-1))-1./(Y(N+1)-Y(N))
      SF(N) = (Y(N+1)-Y(N))/((Y(N)-Y(N-1))*(Y(N+1)-Y(N-1)))
   39 CONTINJE
      C2 = Y(3)*Y(4)/(Y(2)*(Y(3)-Y(2))*(Y(4)-Y(2)))
      C3 = -Y(2)*Y(4)/(Y(3)*(Y(4)-Y(3))*(Y(3)-Y(2)))
      C4 = Y(2)*Y(3)/(Y(4)*(Y(4)-Y(3))*(Y(4)-Y(2)))
      RETURN
      END
```

```
      SUBROUTINE SETUP(LGO,M,NV,REB,X,Y,UC,PRESS,GRAD,DELT,DISP,THETA,VI
     1SC,MTRAN)
C
C  SUBROUTINE FOR CALCULATION OF BOUNDARY LAYER THICKNESS,
C  DISPLACEMENT THICKNESS, MOMENTUM THICKNESS AND EDDY VISCOSITY.
C
      DIMENSION X(300),Y(100),UC(100,3),VISC(100,2),GRAD(100)
      RTR=SQRT(REB)
      NY = NV + 2
      UEDGE = .995*UC(NY,1)
      DO 10 N=1,NV
      IF(UEDGE-UC(N+1,1)) 41,41,10
 41   NDELT = N
      GO TO 20
 10   CONTINJE
 20   DELT = Y(NDELT)+(UEDGE-UC(NDELT,1))*(Y(NDELT+1)-Y(NDELT))/(UC(NDEL
     1T+1,1)-UC(NDELT,1))
      SUM = 0.
      DO 50 N=2,NY
 50   SUM = SUM+(Y(N)-Y(N-1))*(UC(N,1)+UC(N-1,1))
      DISP = (Y(NY)-.5*SUM/UC(NY,1))/RTR
      SUM = 0.
      UEDGE = UC(NY,1)
      DO 60 N=2,NY
 60   SUM = SUM+(Y(N)-Y(N-1))*((UEDGE-UC(N,1))*UC(N,1)+(UEDGE-UC(N-1,1))
     1*UC(N-1,1))
      THETA = .5*SUM/(RTR*UEDGE**2)
      IF(LGO) 53,53,56
 53   NVP1=NV+1
      EASE = 1.
      IF(M-MTRAN) 31,32,32
 32   IF(MTRAN+5-M) 31,31,33
 33   EASE = (X(M)-X(MTRAN))/(X(MTRAN+5)-X(MTRAN))
 31   CONTINJE
      INNER=0
      FAC1 = .16*RTR*EASE
      FAC2 = .0168*UEDGE*DISP*REB*EASE
      EFAC1 = -RTR/26.
      EFAC2 = PRESS/RTR
      TAUW = GRAD(1)/RTR
      DO 160 N=2,NVP1
      ALTER = 1.+FAC2/(1.+5.5*(Y(N)/DELT)**6)
      IF(INNER) 402,401,402
 402  VISC(N,1)=ALTER
      GO TO 160
 401  CONTINJE
      TAUMY=TAUW-Y(N)*EFAC2
      IF(TAUMY) 701,701,702
 701  VISC(N,1)=1
      GO TO 703
 702  EX=Y(N)*EFAC1*SQRT(TAUMY)
      VISC(N,1) = 1.+FAC1*Y(N)*Y(N)*ABS(GRAD(N))*(1.-EXP(EX))**2
 703  IF(VISC(N,1)-ALTER) 160,160,521
 521  VISC(N,1)=ALTER
```

```
      INNER=1
160   CONTINJE
      SAVE=1.
      DO 162 N=2,NV
      RAVE=VISC(N,1)
      VISC(N,1)=(VISC(N+1,1)+RAVE+SAVE)/3.
162   SAVE=RAVE
56    CONTINUE
      RETURN
      END
```

REFERENCES

1. Ham, N. D.; and Young, M. I.: Torsional Oscillation of Helicopter Blades Due to Stall. J. Aircraft, vol. 3, no. 3, May-June 1966, pp. 218-224.

2. Liiva, J.; et al.: Two-Dimensional Tests of Airfoils Oscillating Near Stall. USAAVLABS Tech. Rept. no. 68-13A, April 1968.

3. Ericsson, L.; and Reding, J.: Dynamic Stall of Helicopter Blades. J. Am. Helicopter Soc., vol. 17, no. 1, January 1972, pp. 11-19.

4. Johnson, W.; and Ham, N. D.: On the Mechanism of Dynamic Stall. J. Am. Helicopter Soc., vol. 17, no. 4, October 1972, pp. 36-45.

5. Tarzanin, F.: Prediction of Control Loads Due to Blade Stall. J. Am. Helicopter Soc., vol. 17, no. 2, April 1972, pp. 33-46.

6. Crimi, P.; and Reeves, B. L.: A Method for Analyzing Dynamic Stall of Helicopter Rotor Blades. NASA CR 2009, May 1972.

7. Smith, A. M. O.; and Cebeci, T.: Numerical Solution of the Turbulent-Boundary-Layer Equations. McDonnell Douglas Rept. No. DAC 33735, May 1967.

8. Reeves, B. L.; and Lees, L.: Theory of Laminar Near Wake of Blunt Bodies in Hypersonic Flow. AIAA J., vol. 3, no. 11, November 1965, pp. 2061-2074.

9. Giesing, J. P.: Nonlinear Two-Dimensional Unsteady Potential Flow with Lift. J. Aircraft, vol. 5, no. 2, March-April 1968, pp. 135-143.

10. Chen, C. F.; and Wirtz, R. A.: Second-Order Theory for Flow Past Oscillating Foils. AIAA J., vol. 6, no. 8, August 1968, pp. 1556-1562.

11. Lighthill, M. J.: A New Approach to Thin Airfoil Theory. Aeron. Quart., vol. 3, November 1951, pp. 193-210.

12. Sears, W. R.; and Telionis, D. P.: Unsteady Boundary Layer Separation. Preprint for IUTAM Symposium on Unsteady Boundary Layers, Quebec, May 1971.

13. Abbott, I. H.; and Doenhoff, A. E.: Theory of Wing Sections. Dover Publications, New York, 1959.

14. McCullough, G. B.; and Gault, D. E.: Examples of Three Representative Types of Airfoil-Section Stall at Low Speed. NACA TN 2502, September 1951.

☆U.S. GOVERNMENT PRINTING OFFICE: 1974-739-157/94

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

SPECIAL FOURTH-CLASS RATE
BOOK

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION
451

U.S.MAIL

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546