ANALYSIS OF STALL FLUTTER
OF A HELICOPTER ROTOR BLADE

by Peter Crimi

Prepared by
AVCO SYSTEMS DIVISION
Wilmington, Mass. 01887
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1973
# Analysis of Stall Flutter of a Helicopter Rotor Blade

**Report No.**
NASA CR-2322

**Title and Subtitle**
ANALYSIS OF STALL FLUTTER OF A HELICOPTER ROTOR BLADE

**Author(s)**
Peter Crimi

**Performing Organization Name and Address**
AVCO Systems Division  
Wilmington, Massachusetts 01887

**Sponsoring Agency Name and Address**
National Aeronautics and Space Administration  
Washington, D.C. 20546

**Abstract**
A study of rotor blade aeroelastic stability was carried out, using an analytic model of a two-dimensional airfoil undergoing dynamic stall and an elastomechanical representation including flapping, flapwise bending and torsional degrees of freedom. Results for a hovering rotor demonstrated that the models used are capable of reproducing both classical and stall flutter. The minimum rotor speed for the occurrence of stall flutter in hover was found to be determined from coupling between torsion and flapping. Instabilities analogous to both classical and stall flutter were found to occur in forward flight. However, the large stall-related torsional oscillations which commonly limit aircraft forward speed appear to be the response to rapid changes in aerodynamic moment which accompany stall and unstall, rather than the result of an aeroelastic instability. The severity of stall-related instabilities and response was found to depend to some extent on linear stability. Increasing linear stability lessens the susceptibility to stall flutter and reduces the magnitude of the torsional response to stall and unstall.

**Key Words**
Helicopter rotor, Aeroelasticity, Dynamic stall, Torsional stability

**Distribution Statement**
Unclassified - Unlimited
A study of rotor blade aeroelastic stability was carried out, using an analytic model of a two-dimensional airfoil undergoing dynamic stall and an elastomechanical representation including flapping, flapwise bending and torsional degrees of freedom. Results for a hovering rotor demonstrated that the models used are capable of reproducing both classical and stall flutter. The minimum rotor speed for the occurrence of stall flutter in hover was found to be determined from coupling between torsion and flapping. Instabilities analogous to both classical and stall flutter were found to occur in forward flight. However, the large stall-related torsional oscillations which commonly limit aircraft forward speed appear to be the response to rapid changes in aerodynamic moment which accompany stall and unstall, rather than the result of an aeroelastic instability. The severity of stall-related instabilities and response was found to depend to some extent on linear stability. Increasing linear stability lessens the susceptibility to stall flutter and reduces the magnitude of the torsional response to stall and unstall.
INTRODUCTION

Aeroelastic stability of a helicopter rotor blade is a multifaceted problem because of the extreme variations of the aerodynamic environment within the flight envelope of the aircraft. In hovering flight, a blade can undergo classical binary flutter (Ref. 1) or stall flutter (Ref. 2). In forward flight, the linear instability experienced by systems with periodically varying parameters can occur (Ref. 3). While these types of instability are not normally encountered with blades of current design, due to the relatively low disc loading and weak coupling of translational and rotational degrees of freedom, they are certainly not precluded from new designs, particularly those intended to extend present performance capabilities. Of immediate concern, however, in both design and operation, is the occurrence of large-amplitude torsional oscillations and excessive control-linkage loads associated with blade stall on the retreating side of the rotor disc at high forward speed or gross weight, effectively limiting aircraft performance. This problem has prompted a number of recent studies of dynamic stall and the effects of stall on blade dynamics (Refs. 4-8).

While stall has been identified as a causal element of the problem, the nonlinearity of the stall process, coupled with the unsteady aerodynamic environment, has precluded an analysis to the depth required to gain a thorough understanding of the mechanisms involved. In particular, it has not been clear whether the blade undergoes a true aeroelastic instability, a simple forced response, or some hybrid phenomenon which takes on the character of one or the other extreme, depending on flight conditions and blade vibrational characteristics.

Stall flutter for axial flight is amenable to analysis by empirical methods similar to those developed for analyzing stall flutter in cascades (Ref. 9). The flutter mechanism for that case has been identified as deriving from the extraction of energy from the free stream by the periodic variation of the aerodynamic moment. Analogous methods applied to the forward-flight problem (Refs. 10 and 11) have been inconclusive, however, the primary difficulty possibly being in applying empirical methods without a clear definition of the underlying mechanism of the problem.

A method was recently developed for analyzing dynamic stall of an airfoil undergoing arbitrary pitching and plunging motions which provides an ideal tool for analyzing the stall problem in forward flight. The method, which is described in detail in Ref. 7, employs models for each of
the basic flow elements contributing to the unsteady stall of a two-dimensional airfoil. Calculations of the loading during transient and sinusoidal pitching motions are in good qualitative agreement with measured loads. Dynamic overshoot, or lift in excess of the maximum static value, as well as unstable moment variation, are in clear evidence in the computed results.

This study was directed to analyzing the aeroelastic stability of a helicopter rotor, particularly as it relates to stall, using the method of Ref. 7 to compute aerodynamic loading. The representation of the elastomechanical system includes flapping and flapwise bending degrees of freedom as well as torsion. A listing of the computer program used to perform the calculations is given in Appendix A.
### SYMBOLS

- **b**: blade semichord, m
- **$C_L$**: mean lift coefficient, ratio of time average of $1$ to $\rho \Omega^2 R^2 b$
- **$C_l$**: lift coefficient, $C_l = C_1 / (\rho U^2 b)$
- **$C_{m c/4}$**: moment coefficient referred to quarterchord, $C_{m c/4} = m_{c/4} / (2 \rho U^2 b^2)$
- **c**: blade chord, m
- **$f_\theta$**: mode shape of first uncoupled torsional mode, unit tip deflection
- **$f_\phi$**: mode shape of first uncoupled flapwise bending mode, unit tip deflection
- **$h_b$**: tip deflection due to flapping, semichords
- **$h_\phi$**: tip deflection due to bending, semichords
- **$h_i$**: translational coordinates of 2-D system $(i = 1, 2)$, semichords
- **$I_0$**: moment of inertia of 2-D system about pitch axis, kg · m
- **$I'_\theta$**: blade moment of inertia about elastic axis per unit span, kg · m
- **$k_i$**: translational spring stiffnesses of 2-D system $(i = 1, 2)$, N/m²
- **$k_\theta$**: torsional spring stiffness of 2-D system, N/rad
- **$l$**: lift per unit span at aerodynamic reference radius, N/m
- **$l_{s1}$**: offsets of springs from pitch axis of 2-D system $(i = 1, 2)$, m
- **$M_b$**: total blade mass, kg
- **m**: blade mass per unit span, kg/m
- **$m_{c/4}$**: aerodynamic moment per unit span at aerodynamic reference radius, N
\[ m_1 \] masses of 2-D system, kg/m
\[ R \] rotor radius, m
\[ r_o \] inner radius of blade lifting surface, m
\[ r_R \] aerodynamic reference radius, m
\[ U \] instantaneous free-stream speed at aerodynamic reference section, m/sec
\[ U_o \] reference speed, \( U_o = \Omega r_R \), m/sec
\[ x_m \] distance aft of elastic axis of blade section mass center, m
\[ \bar{x} \] distance aft of pitch axis of mass center of \( m_1 \), m
\[ Z_\beta \] generalized coordinate of 2-D system, equivalent to \( h_\beta \), semichords
\[ Z_\phi \] generalized coordinate of 2-D system, equivalent to \( h_\phi \), semichords
\[ a \] angle of attack, deg
\[ \delta \] flapping hinge offset, m
\[ \Theta_o \] collective pitch angle, deg or rad
\[ \Theta_1 \] blade tip torsional deflection, rad
\[ \Theta \] angle of zero restraint of 2-D system torsion spring, rad
\[ \mu \] advance ratio, ratio of forward speed to \( \Omega R \)
\[ \rho \] free-stream density, kg/m\(^3\)
\[ \tau \] dimensionless time, \( \tau = U_o t/b \)
\[ \psi \] blade azimuth angle measured from downwind direction, deg or rad
\[ \Omega \] rotor rotational speed, rad/sec
\[ \Omega^* \] dimensionless rotor speed, \( \Omega^* = \Omega R/(\omega_0 b) \)
\[ \omega_f \] flutter frequency, rad/sec
\[ \omega_\theta \] frequency of first uncoupled, nonrotating torsion mode, rad/sec

\[ \omega_\phi \] frequency of first uncoupled, nonrotating flapwise bending mode, rad/sec
PROBLEM FORMULATION

Aerodynamic Loading

In the flutter analysis, only leading-edge stall was considered, so the following relates specifically only to that type, even though the basic method can treat trailing-edge stall as well. When the airfoil is not stalled, the flow elements represented are (see Figure 1a): (1) the laminar boundary layer from the stagnation point to separation near the leading-edge, (2) the small leading-edge separation bubble; and, (3) a potential flow, including a vortex wake generated by the variation with time of the circulation about the airfoil. When the airfoil is stalled, as indicated in Figure 1b, the flow elements are: (1) the laminar boundary layer, (2) a dead-air region extending from the separation point to the pressure recovery point; and, (3) a potential flow external to the airfoil and dead-air region, again including a vortex wake. The analytic representations of these elements are described briefly below. Details are given in Ref. 7.

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.

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. 7 was made as general as possible. The method of finite differences for unsteady flows with variable step size in both streamwise and normal directions, was employed, with the error in each finite-difference approximation the order of the square of the step size. It was determined from preliminary calculations performed for this study that, at least for leading-edge stall, results are virtually unaffected by assuming quasi-steady flow in the boundary layer. That assumption was therefore employed for all flutter computations, to take advantage
Figure 1 FLOW ELEMENTS

(a) ATTACHED FLOW

1. LAMINAR BOUNDARY LAYER
2. LEADING-EDGE BUBBLE
3. AIR FOIL AND VORTEX WAKE

(b) LEADING-EDGE STALL

1. LAMINAR BOUNDARY LAYER
2. DEAD-AIR REGION
3. AIRFOIL AND VORTEX WAKE
of the resulting substantial savings in computer storage requirements and computing time.

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 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. 12, 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 these 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, using the blade motions derived by integrating the equations of motion of the elastomechanical system. 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. 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 at the free-stream speed.

Elastomechanical Representation

The equations of motion for a rotor blade with flapping, flapwise bending and torsional degrees of freedom can be written in the form (Ref. 3)

\[
\frac{d^2 h_\beta}{d \tau^2} + \frac{R}{b} \frac{M_\beta \theta}{M_\beta} \frac{d^2 \theta_1}{d \tau^2} + \frac{\bar{\omega}_\beta}{b} \frac{2}{M_\beta} h_\beta - \frac{R}{b} \Omega^2 \frac{T_\beta \theta \theta_1}{M_\beta} = \frac{Rb}{U_o^2} \frac{F_\beta}{M_\beta}
\]

\[
\frac{d^2 h_\phi}{d \tau^2} + \frac{M_\phi \theta}{b M_\phi \phi} \frac{d^2 \theta_1}{d \tau^2} + \frac{\bar{\omega}_\phi}{M_\phi \phi} \frac{2}{M_\phi \phi} h_\phi - \Omega^2 \frac{T_\phi \theta \theta_1}{M_\phi \phi} = \frac{b}{U_o^2} \frac{F_\phi}{M_\phi \phi}
\]

\[
\frac{d^2 \theta_1}{d \tau^2} + \frac{b}{R} \frac{M_\beta \theta}{M_\phi \phi} \frac{d^2 h_\beta}{d \tau^2} + \frac{b}{M_\phi \phi} \frac{d^2 h_\phi}{d \tau^2} + \frac{\bar{\omega}_\theta}{M_\phi \phi} \frac{2}{M_\phi \phi} \theta_1
\]

\[
- \frac{b}{R} \Omega^2 \frac{T_\beta \theta \theta_1}{M_\phi \phi} h_\beta - \Omega^2 \frac{b}{M_\phi \phi} \frac{T_\phi \theta \theta_1}{M_\phi \phi} h_\phi = \frac{b^2}{U_o^2} \frac{F_\phi}{M_\phi \phi}
\]
are tip displacements due to flapping and bending, respectively, in semichords, $\Theta_1$ is torsional displacement at the blade tip and the frequencies* are the following functions of rotational speed:

$$\bar{\omega}_\beta^2 = -\bar{\Omega}^2 \frac{T_{\beta\beta}}{M_{\beta\beta}}, \quad \bar{\omega}_\phi^2 = \bar{\omega}_\phi^2 - \bar{\Omega}^2 \frac{T_{\phi\phi}}{M_{\phi\phi}},$$

$$\bar{\omega}_\theta^2 = \bar{\omega}_\theta^2 - \bar{\Omega}^2 \frac{T_{\theta\theta}}{M_{\theta\theta}}$$

The inertial and centrifugal-force coefficients are given by

$$M_{\beta\beta} = \int_\delta^R (r + \delta)^2 \, m \, dr, \quad M_{\phi\phi} = \int_\delta^R m \, \phi^2 \, dr,$$

$$M_{\theta\theta} = \int_\delta^R I_\theta \, \phi^2 \, dr,$$

$$M_{\beta\theta} = -\int_\delta^R m \, x_m \, (r - \delta) \, \phi \, dr,$$

$$M_{\phi\theta} = -\int_\delta^R m \, x_m \, \phi \, \phi \, dr,$$

$$T_{\beta\beta} = -\int_\delta^R r \, (r - \delta) \, m \, dr,$$

*Barred quantities are dimensionless frequencies, $U_0/b$ being reference frequency; e.g., $\bar{\Omega} = \Omega \, b/U_0.$
The complexity of the aerodynamic representation precludes evaluation of the generalized forces $F_\beta$, $F_\phi$ and $F_\theta$ by the usual strip approximation. It was felt essential, however, to retain both translational degrees of freedom in the investigation of the forward-flight problem, so a simple two-dimensional model of the dynamics could not be used. Therefore, a two-dimensional airfoil suspended in such a way as to have three degrees of freedom was analyzed. Inertial and stiffness parameters were assigned to make the coupled natural frequencies of the two-dimensional system match those of the rotor blade.

The system analyzed is shown schematically in Figure 2. The matching of the two-dimensional system with the blade dynamics proceeds as follows. Three generalized coordinates are first defined to correspond to those of the blade. Clearly, angular displacement $\theta_1$ should correspond to blade torsional displacement at the blade tip. The counterparts of flapping and bending, $Z_\beta$ and $Z_\phi$, respectively, are defined by

$$Z_\beta = A_1 h_1 + B h_2, \quad Z_\phi = A_2 h_1 - B h_2$$

where

$$A_1 = \frac{\overline{\omega}_\beta^2 - \overline{\omega}_2^2}{\overline{\omega}_\phi^2 - \overline{\omega}_\beta^2}, \quad A_2 = \frac{\overline{\omega}_2^2 - \overline{\omega}_\phi^2}{\overline{\omega}_\phi^2 - \overline{\omega}_\beta^2}$$

$$B = \frac{(\overline{\omega}_2^2 - \overline{\omega}_\phi^2)(\overline{\omega}_2^2 - \overline{\omega}_\beta^2)}{(\overline{\omega}_\phi^2 - \overline{\omega}_\beta^2) \overline{\omega}_2^2}$$

(1)
Figure 2 TWO-DIMENSIONAL ELASTOMECHANICAL SYSTEM
and \( \bar{\omega}_1^2 = (k_1/m_1)(b/U_0)^2, \ 1 = 1, 2. \)

With the above definitions, \( Z_\beta + Z_\phi = -h_1 \), to give the correct translational correspondence. It can further be shown that the uncoupled natural frequencies of the two-dimensional system match those of the blade, provided

\[
\left( \frac{k_\theta + k_1 l_{s1}^2 + k_2 l_{s2}^2}{I_0} \right) \left( \frac{b}{U_0} \right)^2 = \bar{\omega}_\theta^2
\]

while \( \bar{\omega}_1^2 \) and \( \bar{\omega}_2^2 \) satisfy

\[
\bar{\omega}_1^2 \bar{\omega}_2^2 = \bar{\omega}_\phi^2 \bar{\omega}_\beta^2,
\]

\[
\bar{\omega}_1^2 + (1 + m_2/m_1) \bar{\omega}_2^2 = \bar{\omega}_\phi^2 + \bar{\omega}_\beta^2 \quad (2)
\]

By comparing the generalized masses of the two systems, it follows that

\[
m_1 b^2/I_0 = -A_1 M_\beta \frac{b^2}{(M_{\theta \theta} R^2)}
\]

\[
A_2/A_1 = M_\beta \frac{1}{(M_{\phi \phi} R^2)} \equiv \lambda_m
\]

The last relation, together with Eqs. (1) and (2), fixes \( m_2/m_1 \):

\[
m_2/m_1 = \frac{(1 + \lambda_m)(\bar{\omega}_\phi^4 + \lambda_m \bar{\omega}_\beta^4)}{(\lambda_m \bar{\omega}_\beta^2 + \bar{\omega}_\phi^2)^2} - 1
\]

Equating the corresponding coefficients of the characteristic equations of the two systems provides three additional relations, which can be solved for the coupling parameters \( \bar{X}, l_{s1}, l_{s2} \). That calculation is outlined in Appendix B.
To complete the matching, quasi-steady approximations to the damping terms of the flapping equations are equated with the result that

\[ m_l \frac{R}{(-A_l)} = 4 \frac{r_R}{R} \frac{M_{\beta\beta}}{R^2 \left[ 1 - \left( \frac{r_o}{R} \right)^4 \right]} \]

\[ \frac{U}{U_0} = 1 + \frac{4}{3} \left[ \frac{1 - (r_o/R)^2}{1 - (r_o/R)^4} \right] \mu \sin \psi \]

where \( \Omega r_R = U_0 \). The aerodynamic reference radius \( r_R \) was selected to be \( .75R \).

The angle of zero restraint in torsion was varied periodically to approximate the effects of cyclic pitch variation in forward flight, according to the formula

\[ \tilde{\theta} = \theta_o \left[ 1 - 2 \left( \frac{R}{r_R} \right) \mu \sin \psi \right] \]

This variation gives nominally constant lift.

The equations of motion were solved by integrating analytically, using linear extrapolations to approximate the variation of lift and aerodynamic moment over the interval of integration. This scheme was found to give satisfactory results, provided the time interval of integration is no longer than about one fifth of the period of the coupled mode having the highest natural frequency.
RESULTS OF COMPUTATIONS

Configurations Analyzed

Vibrational and aerodynamic characteristics of the blade analyzed were selected to correspond to those of the model rotor blade described in Ref. 2. That blade is untwisted, of constant chord, with offset flapping hinge. Pertinent dimensionless parameters of the model blade are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b/R$</td>
<td>0.0435</td>
</tr>
<tr>
<td>$\delta/R$</td>
<td>0.0543</td>
</tr>
<tr>
<td>$r_o/R$</td>
<td>0.174</td>
</tr>
<tr>
<td>$\omega e_0/\omega \phi_0$</td>
<td>3.69</td>
</tr>
<tr>
<td>$\rho R b^2/M_b$</td>
<td>0.00431</td>
</tr>
<tr>
<td>$x_m/b$</td>
<td>0.216</td>
</tr>
<tr>
<td>$m R/M_b$</td>
<td>1.055</td>
</tr>
<tr>
<td>$I_\theta '/M_b R$</td>
<td>3.51 x 10^{-4}</td>
</tr>
</tbody>
</table>

Two elastomechanical configurations in addition to the nominal one were analyzed. One of these had $\omega e_0/\omega \phi_0 = 2.5$, with all other parameters as listed in Table 1. The third configuration had $x_m/b = 0.108$, with the remaining parameters as listed in Table 1.

The bending mode shape, which was computed by a finite-element method, was found not to vary appreciably over the range of rotational speeds of interest. The mode shape for $\omega \phi_0/\Omega = 1.26$, which is plotted in Figure 3, was used for all computations. The torsional mode shape for the nonrotating blade, also shown in Figure 3, was used to evaluate torsional inertia parameters.
Figure 3 BENDING AND TORSION MODE SHAPES
The test blade had a NACA 23012 section. The variation of static lift and moment coefficients with angle of attack for this section were computed from a series of transient pitch calculations, and are shown in Figure 4, together with the measured section characteristics, from Ref. 13. The aerodynamic model is seen to give nearly the correct maximum lift, but at a slightly lower angle of attack, and, as indicated from the variation of $C_m c/4$, the computed center of pressure is somewhat further aft than that of the actual airfoil section below the stall angle.

**Stability in Hover**

Initial calculations were performed for hovering flight, with the nominal configuration, to allow a direct comparison with the test results of Ref. 2. First, rotor speed was varied parametrically, with the collective pitch at a value well below the stall incidence. A classical bending-torsion instability was encountered at $\Omega^* = \Omega R/(\omega \theta_0 b) = 5.3$ with $\omega_f/\omega \theta_0 = .803$. The variation of bending, flapping, and torsional displacements with azimuth angle at flutter onset are shown in Figure 5. By way of comparison, tests (Ref. 2) yielded classical flutter at about $\Omega^* = 7.1$ with $\omega_f/\omega \theta_0 = .72$.

It should be noted that since the system stability was analyzed by direct simulation, a precise point of linear instability was not computed. The values of $\Omega^*$ at onset of a linear instability, both for hover and forward flight, were obtained by successively increasing or decreasing rotor speed, in small steps, until the transient response changed from convergent to divergent, or visa versa. The maximum error in the value of flutter speed, for the results presented here, is estimated to be about three percent.

Susceptibility of the system to stall flutter was investigated next. It was found that a torsional limit cycle, at approximately the highest coupled natural frequency of the system, could be triggered for $\Omega^*$ as low as 3.4. Computed blade motions for stall flutter at $\Omega^*$ of 3.5 are shown in Figure 6.

For $\Omega^*$ below 3.4, a limit cycle could not be set up, regardless of the initial conditions or the collective pitch angle. Severe oscillations involving repeated stall and unstall could be made to occur by imposing a large initial bending deflection. However, the flapping response modulated the torsional response, and caused continuous stall and/or unstall of the blade over a significant portion of
Figure 4  AIRFOIL SECTION CHARACTERISTICS FOR NACA 23012
Figure 5 DISPLACEMENT TIME HISTORIES AT CLASSICAL FLUTTER ONSET
$\Omega^* = 5.3$, $\theta_0 = 11$ deg, $\mu = 0$
Figure 6  DISPLACEMENT TIME HISTORIES FOR STALL FLUTTER

$\Omega^* = 3.5$, $\theta_0 = 15.0$ deg, $\mu = 0$
a revolution, due to the large plunging rate generated by the flapping motion. An example of this occurrence is shown in Figure 7. Thus, while stall flutter involves only the rotational degree of freedom, the results obtained indicate that the minimum speed for its occurrence is determined by coupling with a translational degree of freedom.

Results for the hovering case are summarized in Figure 8, which compares computed and measured flutter speed and frequency, plotted against collective pitch angle. No upper limit in collective pitch angle for the occurrence of stall flutter was calculated, since that limit would depend strongly on initial conditions, and so would be arbitrary. Quantitative differences between the computed and measured stability boundaries of Figure 8 can be attributed in large part to the use of a two-dimensional aerodynamic model, which cannot precisely reproduce the aerodynamic coupling between the rotational and translational degrees of freedom.

From the basic similarity of the computed and measured stability boundaries and the character of the computed instabilities (Figures 5 and 6) it can be concluded that the aerodynamic and dynamic models formulated are capable of reproducing both classical and stall flutter as experienced by a rotor blade, and so can be employed to investigate the forward-flight problem.

Stability in Forward Flight

The nominal configuration was analyzed next for an advance ratio of \( \alpha = 0.1 \). Computations were carried out in the same sequence as for hovering. First, the rotational speed at which classical flutter occurs was determined. Then, stall-related instabilities were investigated.

A linear bending-torsion instability of the Floquet type (Ref. 14) was encountered at \( \Omega^* = 5.2 \). Blade motions as a function of azimuth angle at flutter onset are shown in Figure 9. The torsional and bending displacements are seen to display the aperiodic character typical of this type of instability. The flapping motion is the steady-state response to the cyclic pitch variation.

An instability analogous to stall flutter in hover was found to occur for \( \Omega^* \) as low as about 4.4, with collective pitch angle greater than 12 deg. Blade motions for \( \Omega^* = 4.8 \) are shown in Figure 10. The torsional displacement time history, while not strictly periodic, is nonetheless
Figure 7  BLADE RESPONSE BELOW STALL FLUTTER BOUNDARY
\( \Omega^* = 3.1, \theta_0 = 15.0 \text{ deg}, \mu = 0 \)
Figure 8 FLUTTER SPEED AND FREQUENCY VARIATION WITH COLLECTIVE PITCH ANGLE FOR A HOVERING ROTOR
Figure 9  DISPLACEMENT TIME HISTORIES AT LINEAR INSTABILITY ONSET

\[ \Omega^* = 5.2, \theta_0 = 6 \text{ deg}, \mu = 0.1 \]
Figure 10 DISPLACEMENT TIME HISTORIES FOR STALL FLUTTER
\( \Omega^* = 4.8, \theta_0 = 13 \text{ deg}, \mu = 0.1 \)
brought about by successive stall and unstall. The azimuth positions at which those events occur are marked by (S) and (U), respectively, on the $\psi$-scale.

The blade motions for the type of instability shown in Figure 10 are not of the same character as those of particular concern in the limiting of helicopter performance, in that the excessive torsional displacements shown in Figure 10 persist over a complete revolution of the blade. The control load time history, taken from flight test (Ref. 6), shown in Figure 11 illustrates the type of stall-related blade motions usually encountered at a thrust level or forward speed near the upper limit of an aircraft. Large oscillations in the control loads, presumably deriving from blade torsional oscillations, are seen from Figure 11 to persist only between about $\psi = 270$ deg and $\psi = 400$ deg, rather than throughout a complete revolution of the blade.

A torsional displacement time history closely resembling the variation of control loads in Figure 11 was obtained for $\Omega^* < 4.4$, for collective pitch angles between 12 and 13 deg. Results for two typical cases are shown in Figures 12 and 13. The occurrences of stall and unstall are indicated on the abscissas. The large oscillations in torsion are clearly related to stall, but their persistence is not the result of successive stalling and unstalling, as would be the case for true stall flutter. The blade appears to be responding to the sudden changes in aerodynamic moment at stall onset and unstall, as can be seen by comparing the variation of moment coefficient shown in Figures 12 and 13 with that of torsional displacement, and noting the azimuth positions at which stall and unstall occur. There is some cyclic stall-unstall within the stall zone evident in the results, particularly at the higher rotor speed ($\Omega^* = 4.15$, Figure 13). However, the major contributors to the oscillations appear to be the initial and final pulses associated with stall and unstall upon entering and leaving that zone. There are, in general, two cycles of torsional oscillation of excessive amplitude after the blade unstalls the last time on a given revolution. The response can be regarded as transient, on a localized time scale, or forced, when viewed on a scale of several rotor revolutions. The severity of the response is apparently due in part to the suddenness of load changes at stall and unstall, and partly to the relative lack of aerodynamic damping in pitch, particularly when the blade is not stalled.

If the collective pitch angle is increased, the blade does undergo stall flutter, as seen from the time history plotted in Figure 14. These results are for the same rotor
Figure 11 VARIATION OF PITCH LINK LOAD IN FLIGHT TEST OF CH-47 AT 123 KNOTS
(from Ref. 6)
Figure 12 DISPLACEMENT AND MOMENT TIME HISTORIES FOR EXCESSIVE TORSIONAL RESPONSE

\( \Omega^* = 3.89, \theta_0 = 12 \text{ deg, } \mu = 0.1 \)
Figure 13 DISPLACEMENT AND MOMENT TIME HISTORIES FOR EXCESSIVE TORTIONAL RESPONSE

\[ \Omega^* = 4.15, \theta_0 = 12 \text{ deg}, \mu = 0.1 \]
Figure 14  DISPLACEMENT TIME HISTORIES FOR STALL FLUTTER AT LOW ROTOR SPEED

\[ \Omega^* = 3.89, \theta_0 = 14.3 \text{ deg}, \mu = 0.1 \]
speed as those of Figure 12, but with $\Theta_0$ increased from 12 deg to 14.3 deg. Successive stall and unstall persists over the whole revolution of the blade for this case.

It could be argued that the blade torsional oscillations of Figures 12 and 13 are still a manifestation of stall flutter, even though successive stall and unstall is not taking place, since the aerodynamic moment can undergo unstable variations when the blade remains stalled throughout a cycle (Ref. 4). It may, in fact, be the case that the large deflections do result partly from that effect, so choosing to term them as simply a response may be somewhat misleading. On the other hand, the solutions are distinctly different from what is definitely stall flutter obtained both in hover (Figure 6) and in forward flight (Figures 10 and 14) so that label would seem to be even less appropriate. Further, the persistence of the oscillations after exit from the stall zone is clearly symptomatic of a response, so, for lack of a more precise term, solutions of the type shown in Figures 12 and 13 are identified in what follows as excessive response.

**Linear Stability Boundaries**

The value of $\Omega^*$ at the onset of linear instability was determined for the three configurations considered, for advance ratios of 0, .1, .2, and .3. The effects of advance ratio and torsion-bending frequency ratio on linear stability are shown in Figure 15, where $\Omega^*$ is plotted against $\mu$ for two different frequency ratios. Increasing advance ratio is seen to cause some decrease in flutter rotational speed, with most of the decrease occurring between advance ratios of .1 and .2. The substantial decrease in frequency ratio, from 3.69 to 2.5, caused only about a 4 percent reduction in flutter speed over the range of advance ratios considered. The insensitivity to frequency ratio can be attributed to the large chordwise mass imbalance, which produces the same effect in classical binary flutter of a wing (Ref. 15).

The effect of chordwise mass imbalance on linear stability is shown in Figure 16, where $\Omega^*$ at flutter onset is plotted against $\mu$ for values of $x_m$ of .216 and .108 semichords. As one would expect, the reduction in $x_m$, and hence in the coupling between bending and torsion, causes a substantial increase in the flutter rotational speed.
Figure 15  EFFECT OF ADVANCE RATIO AND TORSION-BONDING FREQUENCY RATIO ON LINEAR STABILITY - $X_{mb} = 0.216$
Figure 16 EFFECT OF Xm ON LINEAR STABILITY
$\omega_0 / \omega_0' = 3.69$
Stall Flutter and Response Boundaries

The effect of forward speed on stall-related instabilities for the three configurations was investigated by systematically varying the collective pitch angle and advance ratio, with $\Omega^*$ equal to 3.89. In order to relate the results to rotor performance, a mean lift coefficient $\overline{C_L}$ is defined, according to

$$\overline{C_L} = \frac{\overline{\ell}}{\rho \Omega^2 R^2 b}$$

where $\overline{\ell}$ is the time-averaged lift per unit span at the aerodynamic reference radius. This coefficient is, to a good approximation, directly proportional to the thrust coefficient (see Ref. 16). The two-dimensional aerodynamic model does not provide a good measure of $\overline{C_L}$ when the rotor is partially stalled, so $\overline{C_L}$ was computed assuming it varies linearly with the collective pitch angle, using the formula

$$\overline{C_L} = a(\mu)(\theta_0 + .0217)$$

The slope $a$ and zero-lift collective pitch angle of -.0217 rad were obtained from calculations of $\overline{C_L}$ for the nominal configuration with stall precluded. The variation of $a$ with $\mu$ is shown in Figure 17.

The results obtained for the nominal configuration are summarized in Figure 18 as a plot of $\overline{C_L}$ vs $\mu$. As thrust is increased at a given $\mu$, the rotor is seen to first encounter a region of excessive response, of the type discussed previously, and then, for $\mu$ of .2 or less, a region where stall flutter occurs. Increasing advance ratio has the effect of suppressing the tendency for stall flutter. At $\mu = .2$, stall flutter occurs at $\overline{C_L} = .85$, but a further increase in $\overline{C_L}$ results in excessive response again. At $\mu = .3$ a limit-cycle type of oscillation could not be triggered at all. As a result, stall flutter is confined to a region somewhat as indicated by the shaded area in Figure 18.

The suppression of stall flutter at high advance ratio is apparently caused by an effect similar to the one encountered at low rotor speed in hover, whereby the flapping motion prevented a limit cycle from occurring. This can be seen from the blade motions obtained for $\mu = .3$ and
Figure 17 VARIATION OF $a = d\bar{C}_L/d\theta_o$ WITH ADVANCE RATIO
Figure 18 STALL STABILITY BOUNDARIES FOR $\Omega^* = 3.89$, $\omega\theta_0 / \omega_0 = 3.69$ AND $Xm/b = 0.216$
$\bar{C}_L = .78$, plotted in Figure 19. On the first revolution, as the blade enters the stall zone on the retreating side, it appears that a limit cycle is being set up, with repeated stall and unstall occurring. However, at about $\psi = 420$ deg, the flapping motion has built up in response to the large cyclic pitch changes, producing a negative plunging rate sufficient to keep the blade unstalled over the remainder of its passage on the advancing side. Then, when the blade again enters the stall zone, the large positive flap-induced plunging rate precludes unstall until exit from the stall zone at about $\psi = 670$ deg. As a result, the blade subsequently undergoes excessive torsional response, rather than stall flutter.

The effect of torsion-bending frequency ratio on stall-related instabilities can be seen from Figure 20, where $\bar{C}_L$ is plotted against $\mu$ for $\omega_\theta / \omega_\phi = 2.5$. No instance of excessive torsional response occurred with this configuration for an advance ratio of .2 or less. Instead, limit-cycle type oscillations were set up, with almost no evidence of suppression by the flapping motion, even at relatively high values of $\bar{C}_L$ with $\mu = .2$. At $\mu = .3$, however, only excessive response was obtained, similar to the results for $\omega_\theta / \omega_\phi = 3.69$.

The marked deterioration in stability at the lower frequency ratio is apparently associated with the lessened linear stability of the system. The configuration with $x_m/b = .108$, which is more stable, in the linear sense, than the nominal one, exhibited a trend opposite to the one resulting from a decrease in frequency ratio. The results for the smaller mass center offset, shown in Figure 21, are similar to those of the nominal configuration, Figure 18, but the region in which stall flutter occurs is somewhat reduced, there being no occurrence of stall flutter at an advance ratio of .2. Also, the amplitude of the torsional oscillations in the region of excessive response is considerably reduced, as evidenced by comparing the blade motions plotted in Figure 22, which are for $\mu = .1$, $\bar{C}_L = .95$ and $x_m/b = .108$, with those of the nominal configuration plotted in Figure 12.
Figure 19 DISPLACEMENT TIME HISTORIES AT HIGH ADVANCE RATIO -
$\Omega^* = 3.89$, $C_L = 0.78$, $\mu = 0.3$
Figure 20 STALL STABILITY BOUNDARIES FOR $\Omega^* = 3.89$, $\omega_{\theta_o}/\omega_{\phi_o} = 2.5$
AND $X_m/b = 0.216$
Figure 21 STALL STABILITY BOUNDARIES FOR $\Omega^* = 3.89$, $\omega_{\theta_o}/\omega_{\phi_o} = 3.69$ AND $X_m/b = 0.108$
Figure 22 DISPLACEMENT TIME HISTORIES FOR EXCESSIVE TORSIONAL RESPONSE. 
\( \Omega^* = 3.89, \bar{C}_L = 0.95, \mu = 0.1, \text{ AND } X_m/b = 0.108 \)
CONCLUSIONS

An analysis has been performed of the aeroelastic stability of a helicopter rotor blade in hovering and forward flight. An analytical model of an airfoil undergoing unsteady stall and an elastomechanical representation including flapping, flapwise bending and torsional degrees of freedom were employed in the study. The following conclusions can be drawn from the results obtained.

1. Analysis of aeroelastic stability for a hovering rotor demonstrated that the aerodynamic and dynamic representations developed are capable of reproducing classical and stall flutter.

2. While stall flutter is an instability involving a single rotational degree of freedom, the minimum rotational speed for its occurrence, in hover, is determined from coupling with a translational degree of freedom.

3. In forward flight, the rotor can undergo a linear instability analogous to classical flutter and a stall-induced flutter which, while not manifested by a strictly periodic limit cycle, has the same basic mechanism for its occurrence as stall flutter of a hovering rotor.

4. The large stall-related torsional oscillations which limit forward speed and thrust are primarily the response to the rapid changes in aerodynamic moment which accompany stall and un stall, rather than the result of an aeroelastic instability.

5. Linear stability is relatively insensitive to advance ratio for advance ratios as large as .3.

6. While excessive response due to stall occurs at high advance ratio, stall flutter is precluded by the large flap-induced plunging rates.
7. The severity of stall-related instabilities and response depends to some extent on linear stability. Increasing linear stability lessens the susceptibility to stall flutter and reduces the magnitude of the torsional response to stall and unstall.
APPENDIX A

PROGRAM LISTING
APPENDIX A

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.
PROGRAM TO ANALYZE UNSTEADY AIRFOIL STALL

COMMON /RL1/, NTIME, NDIYC, ISTD
COMMON /CLCMAT / CLVB, CMVR, CMPAVR

COMMON /INPTVB/, FTVB(64), FPVB(64), FPPRVR(64), DIDRVR(64), SETUPS17
A X4VB(64), DELVB, XMUVB, FOVB, XMUAVB, SETUPS18
B ATOVB, ATCVB, ATSVB, ROVB, RVB(64), SETUPS19
C MVVB(64), NVR

COMMON /INPUTS/, NSBL, NZ, NOFF, NGAM, NSIG, SETUPS20
A NCOL, NCORD, LOWER, MSTOP, MAXT, MTTR, SETUPS21
B NOTBL, INOV, ELSIG, DXI, REB, ROAB, SETUPS22
C FRZ, ARR, AMPLU, FREQU, ALPH1, ALPH2, SETUPS23
D HEAVE, AROT, FREQF, PHIM, NY, RV1, SETUPS24
E DRY, Y(100), TEST, UPRIM, XU(30), YU(30), SETUPS25
F XL(30), YL(30), ER1, ER2, ER3, BD3R, SETUPS26
G RRRBR

H, CMPA, CMPAS, BARG, EM1, HVOR, MVOR, SSPA, SVOR, TORF, X1VOR
I, PLOTOPI, PSTLOW, PSIUP
J, MOUT

COMMON /ZZZ/ Z(3)

DIMENSION USAV(300,100), SCALS(300)
DIMENSION USAVL(1), SCALS(300)
DIMENSION CAMBR(24), THICK(24)
DIMENSION XGAM(301), XSIG(100), XSIGA(100), XSIGB(100), XC(300), X(300)
DIMENSION LSBL(300), XBSIG(100)
DIMENSION ACAP(30,3), BCAP(100,3), AS2(30), AS(30,30), BS(30, 30), ASHMAIN
DIMENSION ALAM(30), VZPI(30), FPRES(100), GAMAM(1000), XW(1000)
DIMENSION BLAM(30), FLAM(10), XFLAM(10)
DIMENSION SCALE(100,2), U(1,1,1), UCI(100,3), V(100,2)

DOUBLE PRECISION CMAT(60,60), RMAT(130)

DATA IN, MOUT, NF/ 5,6, 24/
DATA PI, TIME, UINF, RENEL, USTOP/3.14159,0.,1.,4.75E4,2.8/
DATA FLAM /1.75,1.75,1.724,1.527,1.354,1.,663.,452.,25MA /
DATA XFLAM /-100.,-11.26,-7.01,-3.48,-1.766,0.,1.888,4.,MA /
DATA DEGREG /1.74 53292 51994 3300-2/

EQUIVALENCE (CMAT(1),USAV(1)), (ASH(1), SCALS(1))

IF ISTO = 1 TIME DERIVATIVES NOT USED
ISTD = 1
RAD = 180. / PI
IL = 8888
NDMC = 60
CALL SFUPS
IF (ISTD .EQ. 1) GO TO 40
DO 100 J = 1, 1300
SCALJS(J) = 0.
DO 100 I = 1, 100
100 USAV (J, I) = 0
CONTINUE
C
CALL READIN (IL, & 60)
C
NOTE - OFFSETS ARE PUT IN AS LISTED IN THEORY OF WING SECTIONS, I.E. MAIN 66
C
AS A FRACTION OF TOTAL CHORD, X1 BEING MEASURED FROM THE
C
LEADING EDGE. MAKE SURE NF IS AN EVEN NUMBER.
C
TIME=0.
NTIME=0.
NMAKE = 999
ISEP = 0
ISEPY = 0
IWASH = 2
UNF = 1.

L=0
INDV = INDV+1
WRITE (MOUT, 6)
PITCH = ALPH1
IF (INDV .GT. 0) PITCH = PITCH - ALPH2
X
AMPLU = 1.33333 * XMUAV8 * (1. - ROVB**3) / (1. - ROVB**4)
IF (INDV .LE. 2) FREQU = BDBR/RDBR
IF (INDV .GT. 2) GO TO 343
WRITE (MOUT, 25) NVOR, SVOR, HVOR, BARG, XIVOR, EM1, TORF, SSPA
RY = RY1
HVOR = HVOR**2
BARG = BARG/6.2832
343 CALL SECT(XU, YU, X, Y, NOFF, NF, RDBB, THDBB, CMDBB, THICK, CAMBR1)
DO 7875 N = 1, NF
CAMBR(N) = CAMBR(N) + CMDBB
7875 THICK(N) = THICK(N) + TMDDB
WRITE (MOUT, 41)
WRITE (MOUT, 7) AMPLU, FREQU, ALPH1, ALPH2, HEAVE, AROT, FREQ, RDBB, REB
WRITE (MOUT, 8)
WRITE (MOUT, 9) (M, CAMBR(N), THICK(N), N = 1, NF)
MX = NSLB + NZ - 1
CALL SCAL(NBL, NSBL, FRZ, ARR, RDBB)
CALL CORDX(NSBL, NZ, RDBB, SBL, X, CI)
DO 2420 N = 1, MX
IF (XC(N)) 1, 7 2420, 2419, 2419
2419 MEND = M - 1
GO TO 2421
2420 CONTINUE
2421 MX = MEND

49
MXM1=MX-1
LE(MX+1,1)=1.
EPSLF=2.*(X(NZ)-X(NZ-1))
FPSTF=X(MX)-X(MX-1)
ALTG=8.36E4/SQRT(REB)

IF( ISTD.EQ. 1) GO TO 50

DO 2422 M=1,MX
SCALE(M,1)=0.
SCALE(M,2)=0.
DO 2422 N=1,MY
U(M,N,1)=0.

2422 U(M,N,2)=0.
50 CONTINUE

NSIGA=NSIG
NSIGA=NSIG
NSIGA=NSIG
MOTR=MOTR+1
NOTBL=NOTBL+1
XMAX=XMAX-ELSIG
CCNA=.375*PI/IXI
ANGS=PI/2/ATN(NSIG)
CALL_SETSX(NSIG1,1.1.2.,XSIG,ANGS)

XSEP=1.1
DO 2430 N=1,NSIG
XSIG1(N)=XSIG(N)

2430 XSIG1(N)=XSIG(N)

DO 2431 N=1,NSIG
DO 2431 N=1,3

2431 BCAPIN,NUI=0.
PINT=2./FLOAT(NCRO)
NC1=NCRO+1
THX=1.5/OXI
NGP=NGAM+1
NW=1.NWAKE-1
COUNT=0.

DO 8456 N=1,NWAKE
GAMAM(N)=0.
XW(N)=1.4+COUN

8456 COUNT=COUNT+DXI
ANGLE=PI/2/ATN(NGAM)
COUNT=0.

DO 1002 M=1,NGP1
PHM=COUNT*ANGLE
XGAM(M)=ATN(PHM)
COUNT=2.

DO 1001 N=2,NGAM
AX(M,N)=COS(COUNT*PHM)
1001 COUNT=COUNT+1.

1002 COUNT=COUNT+1.

CALL_WASH1(XGAM,NGAM,TM,AMPH,ALPH,AVE,AROT,FREQ,PHM,UINF,CAI)

1M01,NP,VZIP,,I1
DO 8458 M=1,NGP1

9M11=1.

CALL_M1=TEMP
CMAT(M, 2) = XGAM(M)
DO 8457 N = 3, NGAP1
8457 CMAT(M, N) = AS(M, N - 1)
8458 CONTINUE
CALL ALSOLI(NGAP1, CMAT, RMAT)
DO 8459 N = 1, NGAP1
ACAP(N, 1) = RMAT(N)
ACAP(N, 3) = RMAT(N)
8459 ACAP(N, 2) = ACAP(N, 1)
DO 2784 M = 1, MX
SIGN = 1.
IF (M - N7) 2774, 2775, 2775
2774 SIGN = -SIGN
2775 CALL QCAL(ISEP, NGAM, NSIG, NF, XSIG, ACAP, BCAP, THICK, RCRR, GAMAV1L), UFM, MAIN 162
1NF, XCM(M), UF(M, 1), SIGN
2784 UF(M, 2) = UE(M, 1)
DO 1004 M2 = 2, NGAM1
1004 BLAM(M) = (1.125 * XGAM(M) + 1.975 * (1. + XGAM(M)) * (1. - 3. * XGAM(M)) * ALOG(1.) / DM(M)
4 + XGAM(M)) / (1. - XGAM(M)) / DXI
BLAM(NGAP1) = 1.125 / DXI
CALL CLCMNCDT, ISEP, NGAM, NSIG, XSIG, BCAP, ACAP, RCRR, GAMAV1L, UFM, MAIN 504
1AP, THICK, RCRR, GAMAV1L, UINF, UD Ot, DXI, AROT, CMPA)
IF (INDV .EQ. 2)
CALL SUPPl
C INDEXING IN TIME IS CARRIED OUT AT THIS POINT.
C 9559 CONTINUE
CALL ACUCPU(IACU)
IF (IACU .LT. 35000) GO TO 99
C
C NOTE -- FOR READ-IN CF FCIL MOTIONS, MAKE ALPH1 = ALPHA,
C
C ALPH2 = ALPHA-DOT, AND HEAVE = H-DOT.
C
C IF (ICHR .EQ. 2)
XREAD(IN, 2, END=8989) ALPH1, ALPH2, HEAVE
158 NITS = 1
TIME = TIME + DXI
NTIME = NTIME + 1
NWAKE = NTIME + 2
IF (NWAKE = 998) 202, 201, 201
201 NWAKE = 998
202 IF (MAXI - NTIME) 8989, 8800, 8800
8800 SAVFU = UINF
L = L + 1
PILL) = BCBR / RRDRBR * TIME * RAD
PSI360 = AMMT1 PIL, IT, 360, 1
UINF = 1.0 * AMPUL * SIN(FREQ * TIME)
IF (INDV .EQ. 2)
XCALL SUPPl(UINF)
PITCH = ALPH1
IF (INDV + MOTR .LE. 2) PITCH = PITCH - ALPH2 * COS(FREQ * TIME)
UDDOT = FREQ * AMPUL * COS(FREQ * TIME)
STEPX = 5 * DXI * (UINF + SAVEU)
DO 1003 J = 2, NWAKE

MAIN 150
MAIN 151
MAIN 152
MAIN 153
MAIN 154
MAIN 155
MAIN 156
MAIN 157
MAIN 158
MAIN 159
MAIN 160
MAIN 161
MAIN 162
MAIN 163
MAIN 164
MAIN 165
MAIN 166
MAIN 167
MAIN 168
MAIN 504
MAIN 505
MAIN 1569
MAIN 170
MAIN 171
MAIN 172
MAIN 173
MAIN 174
MAIN 175
MAIN 176
MAIN 177
MAIN 178
MAIN 179
MAIN 180
MAIN 181
MAIN 182
MAIN 183
MAIN 184
MAIN 185
MAIN 186
MAIN 187
MAIN 188
MAIN 189
MAIN 190
MAIN 475
MAIN 191
MAIN 192
MAIN 193
JC=NWAF+J+2
GAMAW(JC)=GAMAW(JC-1)
1CC3 XIW(JC)=XIW(JC-1)+STFPX
IF (ISEP) 2009,2009,2007
2CC7 DC 2008 N=1,NSIG
RCAP(N,3)=RCAP(N,2)
2008 RCAP(N,2)=RCAP(N,1)
DO 4433 N=1,NSIG1
XSIGR(N)=XSIGA(N)
4433 XSIGA(N)=XSIG(N)
GO TO 2010
2009 DEADL=0.
ELDNT=UNF
2010 DO 1014 M=1,MX
UE(M,3)=UE(M,2)
1014 UE(M,2)=UE(M,1)
DEADL=DEADL
ELDNT=ELDNT
ALAM(1)=(1.125*+.75*ALOG(STEPX*5))/DXI
DO 1005 M=2,NGP1
1005 ALAM(M)=BLAMM)+.75*(1.+1.-XGAM(M))/STEPX*ALOG(1.+STEPX-XGAM(M))
DXI=1./1.-XGAM(M))/DXI
DC 2006 M=1,NGP1
ACAP(M,3)=ACAP(M,2)
2006 ACAP(M,2)=ACAP(M,1)
ACAP(M)=ACAP(M,1)+APE(M,2)*ACAP(M,2)+APE(M,1)*ACAP(M,1)
ALPHS=VZIP(1)
CALL WASH(XGAM,NGAM,TIME,ALPH1,ALPH2,HEAVY,AROT,FREQF,PHI,HINF,CA)
IMBR,NF,VZIP,MOTR,INDV)
DO 1006 M=1,NGP1
ASZ(M)=1.+2.*ALAM(M)
ASZ(M)=XGAM(M)+ALAM(M)
SUM=0.
DO 4343 J=2,NWML
4343 SUM=SUM+(GAMAW(J+1)-GAMAW(J-1))*XGAM(J)/XI(W(J)+1)
1-XI(W(J))*ALOG(XI(W(J)-XGAM(M))/XI(W(J)-XGAM(M))
ELX=1.-XGAM(M)
IF (M=1) 1006,2130,1006
2130 ELX=1.
1006 AR(M)=2.*VZIP(M)+VZIP(M)*ALAM(M)*ACAP(3)/3.+(SUM-GAMAW(2)*(1.-XGAM(M))*ALOG(MAIN 235
111.+STEPX-XGAMW(1)/ELX)/STEPX/PI
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 (3344,3345,3349,1WASH
3344 XSEP=XSEP+DXI
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 ISEP=0
3015 RCAP(N,K)=0.
GO TO 4444
3345 IF(N leaks) 3344, 3348, 3248
3344 IF(NITS=1) 3248, 3349, 3248
3344 IF(N(ND)=2) GO TO 6349
IF(VNP(11)-ALPHS) 6349, 6348, 6348
6348 NITS=2
GO TO 3248
6349 CALL UNP(N,TGAM,AR, ALAM,AFACT, RMAT, CMAT, XGAM, A, ACAP, M, N, I, XSIG) MAIN 259
1, RCAP, THICK, RDBB, UINF, XC, UE)
GO TO 2785
3248 XATT=XSEP+DEADL+.5*(ELD1+ELDOT)*DI
DEADL=XATT-XSEP
DIFF=1.*XATT
XTEST = XSEP + 3. * EPSLE
CALL SETS(XNSIG, XSEP, XATT, XSIG, ANG)
DO 4434 N=1, NSIG
4434 XSIG(N)=.5*(XSIG(N)+XSIG(N+1))
DO 3086 M=1, NGP
DO 3086 N=1, NSIG
3086 X(SM(N)=0.
DO 3087 M=1, NGP
IF(XGAM(M)-XSEP) 3088, 3089, 3091
3088 IF(XATT-XGAM(M)) 3107, 3087, 3091
3091 DO 3092 I=1, NSIG
IF(XGAM(M)-XSIG(1)) 3093, 3092, 3092
3093 MARK=I
GO TO 3094
3092 CONTINUE
3C54 WIDE S = XSIG(MARK)-XSIG(MARK-1)
BS(M,MARK-1)=(XSG(MARK)-XGAM(M))/WIDES
BS(M,MARK)=(XGAM(M)-XSIG(MARK-1))/WIDES
BS(M,1)=SQRT((XGAM(M)-XSEP)/(XATT-XGAM(M)))
3088 IF(DIFF=1.E-6) 3087, 3098, 3098
3087 X(SM(N)=3. * DIFF/(2.*DI)(SQRT([1.-XGAM(M)]MAINE 284
1)/XATT-XGAM(M))-1.)*((3.*XGAM(M)-1.3.*XATT))
GO TO 3087
3167 BS(N,1)=DIFF*1.5*SQRT(DDEADL)*(3.*XGAM(M))
3087 CONTINUE
C
C SET-UP OF THE SECOND SET OF EQUATIONS STARTS HERE.
C
DO 4350 K=1, NSIG
IF(XRIG(K)-1.1) 4348, 4349, 4349
4348 CSSK=XSIG(K)
SINK=SQRT(1.-COSK*COSK)
THEK=ARCT(COSK)
TAN=SIGN(5.*THEK)/COS(5.*THEK)
ASHZ(K)=TAN*CONA(1.,+COSEQ1.-3.*COSK)+/UINF+THEK*PI+2*SIGN+UINF
1CONAD(1.+COSK)*SINK**2)/UINF
ASHZ(K,1)=5.(ASHZ(K)-TAN)*SINK
COUNT=1.
DO 4355 N=2, NGAM
COUNT=COUNT+1.
4355 ASHT(K,N)=SIGN(COUNT+THEK)+75.*SIGN(COUNT+1.)*THEK/(COUNT+1.)-S
MAIN 304
IN((COUNT-1.)*(THETK)/(COUNT-1.))/(DXI*UINF)
GO TO 4350

4349 ASH2(K)=0.
DO 4359 N=1,NGAM

4355 ASH(K,N)=0.
4350 CONTINUE
IF((DIFF-1.E-6)>5005,5006,5006)

5005 PREC=0.
GO TO 5007

5006 CALL ATTPR(PREC,XSIG,NSIG,ASZ,AS,AR,CMAT,MRAT,NGAM,NF,ACAP,THICK,MAIN)
IDBB,GAMAW, UINF, JDOT,DXI,RCAP)
5007 CALL MIXERFPPRES,PREC,UINF,JDOT,THICK, NF, XBSIG,NSIG,IND,DEL1,THETMAIN
11,REF,USEF,X4,CP1)
CPCT=CP1
DO 4800 K=1,NSIG
CORD=XBSIG(K)

BREP=-1.*THXI*BINT(XSEP,XATT,CORD)/UINF
DO 4808 N=2,NSIG

4806 BSH(K,N)=F(BSIG(N-1),XSIG(N),XSIG(N+1),CORD)+THXI*GB(XSIG(N-1),XSMAIN)
1IG(N),XSIG(N+1),CORD)/UINF
CALL ESIG(T,NSIG,XSIG,BCAP,CORD,VAL1)
CALL ESIG2(N,NSIG,XSIG,BCAP,CORD,VAL2)

ARH(K)=FPREST(K)*(2.*VAL-5*VAL2)/(DXI*UINF)
IF(CORD-1.1 5008,4800,4800)

5008 CALL EGAM2(2,NGAM,ACAP,RCAP,1,2,XSIGA1,NSIGA1,GAMAW2,MAIN)
1CORD,VAL1)
CALL EGAM2(3,NGAM,ACAP,RCAP,1,3,XSIGA1,NSIGA1,GAMAW3,MAIN)

4860 CONTINUE
4444 CONTINUE

C CASES OF STALLED AND UNSTALLED AIRFOILS.

C DO 6500 K=1,NGP1
CMAT(K,L)=ASZ(K)
DO 6485 N=1,NGAM

6465 CMAT(K,N+1)=ASZ(K,N)
IF(1SEP) 6486,6500,6486

6466 DO 6499 N=1,NSIG
NGG=N+NGP1

6459 CMAT(N,NGS)=BS(N,N)
6500 CONTINUE

TF(TESEP) 6502,6501,6502

6501 NTO=NGP1
GO TO 6751

6502 DO 6750 K=1,NSIG
KK=K+NGP1
RMAT(KK)=ARH(K)

CMAT(KK,1)=ASH2(K)
DO 6748 N=1,NGAM

6748 CMAT(KK,N+1)=ASH2(K,N)
CALL RURB(DELL,THET1,RFB,XSEP,USEP,XCS,DCP,DELS,X,MAC,MZ,X5,US5,UMAIN 416
IF(ALT,RFN,USSTOP) USEP=USEP*0.02046#USEP**3 PDIFF=USEP-U*1 #USEP-U5*) WRITF('MOUT,22') PDIFF,DCP IF(DCP-PDIFF) 8263,8366,8366
8263 ISEP=0 GO TO 8463
8366 IF(ISEP) 8368,8369,8369
8369 IF(ISEP) 8467,8467,8368
8467 I=WASH=1 NITS=2 GO TO 3344
8368 GO TO (8168,1786),NOTRL
8168 CALL RFATTUC,V,X,Y,MN,RY,DY,UE,X5,DELS,MST,REB) LAMQ=0 GO TO 8367
8463 IF(ISEP) 7741,7741,7742
7741 ISEP=1 NITS=NITS+1 IF(NINT) 7743,7743,7643
7643 ISEP=1

DXSEP=1.*-XSEP
XSEP=6.*XSEP+.4 CALL CPC(ISEP,NGAM,NSIG,NSIG,NSIG,NSIG,NSIG,NSIG,ACAP,BCAP,MAIN 440
LTHICK,DRAR,GAMAM,UNDF,UDOT,1.,XSEP,DXI,CPL)
GO TO 3248
7742 CALL FLDERT(BCAP,XSIG,NSIG,UNIF,ELDCT,SIGSUM,YMXI)
IF(ISEP,EQ.1,.AND.ISEP,EQ.0,.AND.NITS,EQ.1) GO TO 9210
IF(ISEP,.5) 7841,7842,7842
7841 EPS=EPSLE GO TO 7843
7842 EPS=EPSLE GO TO 7843
7843 DXSEP=ABS(XSEP-XSEP)
IF(DXSEP-EP) 7836,7834,9210
7834 IF(ISEP-XMAX) 1786,1786,7835
7635 ISEP=0

ISEP=0
GO 7836 K=1,3
GO 7836 N=1,NSIG
7836 BCAP(IN,K)=0.
GO TO 1786
9210 NITS=NITS+1 IF(NITS,EQ.2,.AND.NINT,EQ.0) XSEPS=XSEP
IF(NITS,.4) 9211,9211,1786
9211 IF(ISEP-XSEPS) 9335,9305,9306
9305 XSEP=.6*XSEPS+.4*XSEP GO TO 9307
9307 XSEP=.6*XSEPS+.4*XSEPS
9366 XSEP=.6*XSEPS+.4*XSEPS
9307 IF(ISEP-XMAX) 9212,9212,7835
9212 CALL CPC(ISEP,NGAM,NSIG,NSIG,NSIG,NSIG,NSIG,ACAP,BCAP,MAIN 466
LTHICK,DRAR,GAMAM,UNDF,UDOT,1.,XSEP,DXI,CPL)
IF(NOTRL,.EQ.2,.AND.XSEP.GT.0.1) XSEP=-.98
GO TO 3248
7743 IF(NITS-1) 7737,7737,3248
7737 NITS=NITS+1
ELDOT=ELD1
GO TO 3248

1786 WRITE(MOUT,20) NTIME
WRITE(MOUT,26) XIVOR
PITC = PITCH * LRO / PI

2C5 WRITE(MOUT,10) TIME,UNINF,XSEP,XATT,PITC
ALDFG= ALPH1/DEGRFS
WRITE(G,9001) Z,ALDEG,ALPH1, ALPH2, HEAVE
IF(PS3130 .GE. PSILOW .AND. PS3130 .LE. PSIUP) GO TO 101
IF( NOUT .GE. EQ. 0) WRITE(MOUT,10)
1 WRITE(MOUT,11)
IF( NOUT .GE. EQ. 0) WRITE(MOUT,12) (N,XGAM(N),VZIP(N),ARIN(N),ACAP(N),XI(N),GAM(N)),MAIN 480
2N=1,NGPI)
IF(ISEP) 7432,7433,7432

7432 WRITE(MOUT,13)
IF( NOUT .GE. EQ. 0) WRITE(MOUT,14) ELOOT
WRITE(MOUT,15) XSIG(1),CPOT,X4,CPOT,XATT,PREC

7433 WRITE(MOUT,15)
XPC=-1.
DO 7102 N=1,NGPI)
CALL OECAL(ISEP,NGAM,NSIG,NXIG,ACAP,BCAP,THICK,RBB,GAMA(N),UNINF)
1 INF,XPC,QUE(1)
CALL OECAL(ISEP,NGAM,NSIG,NXIG,ACAP,BCAP,THICK,RBB,GAMA(N),UNINF)
1 INF,XPC,QUE(1)
CALL CPC(ISEP,NGAM,NSIG,NXIG,NSIGA,NSIGB,ACAP,BCAP,THICK,RCAP,PREC)
1 THICK,RBB,GAM(N),UNINF,JDOT,1,0,XPC,DXI,CPU)
CALL CPC(ISEP,NGAM,NSIG,NXIG,NSIGA,NSIGB,ACAP,BCAP,THICK,RCAP,PREC)
1 THICK,RBB,GAM(N),UNINF,JDOT,1,XPC,DXI,CPU)
IF(IN-1) 7546,7545,7546

7548 CPL=CPU
7546 DLIFT=CPL-CPU
WRITE(MOUT,16) XPC,QUE,CPL,CPL,QUE,CPU,DLLIFT
7102 XPC=XPC-PINT
101 CONTINUE
CMPA=CMFA
CALL CLCM(NCOL,ISEP,NGAM,NXIG,NSIGA,NSIGB,ACAP,NC,MAIN 504
1AP,THICK,RBB,GAM(N),UNINF,JDOT,1,AROT,CMPA)
P(L,2) = PITC
P(L,3) = Z(3)
P(L,4) = Z(1)
P(L,5) = Z(2)
P(L,6) = CMRA
P(L,7) = CMFA
IF(L.LT.200) GO TO 98
CALL PLOTSB(PLOTNP,P,L)
L=0
98 CONTINUE
IF(ISYM .GE. 1) GO TO 9999
DO 7950 M=1,MX
SCALE(M,2)=SCALE(M,1)

57
SCALE(N,1) = SCALS(M)
DC 7950 N=1,NY
U(M,N+1) = U(M,N,1)
7550 U(M,N,1) = USAV(M,N)
GO TO 9999
8589 CONTINUE
99 CONTINUE
CALL PLOTSB( PLOTOP, P, L )
CALL ACUCPU( IACU )
IF( IACU .LT. 35000 ) GO TO 60
GO TO 40
60 CONTINUE
IF( PLOTOP.EQ. 0. ) CALL EXIT
CALL PLTND
CALL EXIT
RETURN

C

1 FORMAT(13F10.4) MAIN 23
2 FORMAT(3F10.4) MAIN 24
3 FORMAT(2F10.4) MAIN 25
4 FORMAT(1H1/) MAIN 26
5 FORMAT(6F10.4) MAIN 27
6 FORMAT(1H1,50X,34HANALYSIS OF UNSTEADY AIRFOIL STALL///) MAIN 28
7 FORMAT(1X,6FHUBAR =E13.5/7X,7HFREQ =E13.5//3X,11HALPHA ONE =E13.5/MAIN 29
1F13.5/7X,6HRO/9 =E13.5//9X,5HRER =E13.5///) MAIN 31
8 FORMAT(29X,1HN*25X~4HCo 926X9H(N)/ MAIN 32
9 FORMAT(130,2E30.5) MAIN 33
10 FORMAT(5X,3HT =E13.5/5X,3HU =E13.5/4X,4HX =E13.5/4X,4HXO =E13.5/MAIN 34
1X,4HPA =E13.5////) MAIN 35
11 FORMAT(///4X,1HN,11X,1HX,14X,5HVZ(X),12X,5HRN(X),12X,4HAI(N),21X,3HMAIN 36
12 FORMAT(11X,14X,5HGMMA/) MAIN 37
13 FORMAT(15,4E17.5,8X,2E17.5) MAIN 38
14 FORMAT(11H1,8X,1HN,20X,1HX,21X,5HFPI(X),22X,5HRHN(N),21X,4HBN(/1 MAIN 39
15 FORMAT(///45X,9H,1-0CT =E13.5///5X,27HPressures IN Separated Flow\ MAIN 40
1F55X,1HX,19X,2HCP/) MAIN 41
16 FORMA \x12HTIME STEP NOI3///) MAIN 47
17 FORMAT(11H1,50X,12HTIME STEP NOI3///) MAIN 48
18 FORMAT(///45X,26HINCREASE IN CP REQUIRED IS=13.5/40X,26HINCERSE MAIN 49
1X/45X,2HBOUNDARY LAYER XS =E12.4/40X,8HCP(XS) =E12.4/MAIN 50
1/45X,23BOUNDARY LAYER XS =E12.4/ MAIN 51
19 FORMAT(///45X,23H POTENTIAL FLOW XS =E12.4/60X,8HCP(XS) =E12.4/ MAIN 52
1X/45X,24BOUNDARY LAYER XS =E12.4/ MAIN 53
20 FORMAT(///45X,24BOUNDARY LAYER XS =E12.4/) MAIN 54
22 FORMAT(///45X,24BOUNDARY LAYER XS =E12.4//7X,4HNP) =E12.4/3X,4HMP =E12.4/3X,4HMP =E12.4/ MAIN 56
900I FORMATT0, 750, "EQUIVALENT ROTOR BLADE RESPONSE" SUPPL380

9CC1A // T 5, 'FLAP DISP =', G14.5
9CC1P // T47, 'BENDING DISP =', G14.5
9CC1C // T39, 'TORSIONAL DISP =', G14.5
9CC1D // T38, 'SECTION PITCH ANGLE =', F9.3, 'DEGREES OR',
9CC1E // F9.4, 'RADIANS'
9CC1F // T21, 'SECTION PITCH RATE =', G14.5
9CC1G // T71, 'SECTION PLUNGING RATE =', G14.5 //
SUBROUTINE SUPPL
IMPLICIT REAL*8 (A-H,P-F,L-Z)
REAL*4 FR1S, FR2S, FR3S, ANSK, OMS
REAL** CLVB, CMVB, CMPAVB
1, DUMMY, PLOTOP
REAL FTVB, FPVB, FPVRVB, DIDRVB, XMVB, DELVB, XMUVB,
A FOVB, XMUAVR, ATOVBR, ATCVB, ATSVB, ROVB, RVB, MV.
C WDI, PSI, UINF
REAL ELSIG, DIX, REB, RDBR, FRZ, ARR, AMPLU, FREQU,
A ALPHA, ALPH2, HEAVE, AROF, FREOF, PHIH, RY1, DRY,
B X, TEST, UPRIM, XU, YU, XLI, YLI, ERI, ER2, ER3, RDDR,
C RDDR
REAL SUM(8), YCLUD(8), YNEW(8), DEL(3,3), CMPA(3), CL(3), S(3)
A Z, ZPR(3), SMALLG(3), Y(3,3), YPR(3,3), GCAP(3,3)
COMMON /BL1/ NTME, NDIMC
COMMON /CLM/ CLVB, CMVB, CMPAVB
COMMON /ZI/ Z(3)
COMMON /NPFTV/ FTIVB(64), FPVB(64), FPVRVB(64), DIDRVB(64),
A XMVB(64), DELVB, XMUVB, FOVB, XMUAVB,
B ATOVB, ATCVB, ATSVB, ROVB, RVB(64), MV,
C MV(64), NVB
COMMON /NPUTS/ NSBR, NZ, NOFF, NGAM, NSIG, SUPPL 15
A NCOI, NCORD, LOWER, MSTOP, MAXT, MJTR, SUPPL 16
B NOTBL, INDV, ELSIG, DIX, REB, RDBR, SUPPL 17
C FRZ, ARR, AMPLU, FREQU, ALPH1, ALPH2, SUPPL 18
D HEAVE, AROF, FREOF, PHIH, NY, RY1, SUPPL 19
E DRY, X(100), TEST, UPRIM, XU(30), YU(30), SUPPL 20
F XL(30), YL(30), ERI, ER2, ER3, BODR, SUPPL 21
G RDDR
H, DUMMY(10), PLOTOP
DIMENSION DELTA(3,3)
DIMENSION ALPHA(3,3), BETA(3,3), GAMMA(3,3), OMS(3), OMEGA(3), CM(3)
DIMENSION AA(10), AB(10), ANB(20), ANT(20), AAX(10), ANS(20), SORT(3)
1, TOT(2)
CF4(X) = F4 - B4 + (B4 * C6 - C4) ** X
Z(1) = B(1) * (CF4(X) / GB) ** 2 + (CF4(X) * FRIS + (1 - C6 * X * X) * B2 - F2) ** X
Z(2) = (F2 / FRIS + FRIS * CF4(X) - R2 + (1 - C6 * X * X) * (R2 - RZ - FRIS)) ** X
S1(X) = (2 * H * CF4(X) / GB) ** 2 + (FRIS - FR2S) ** X
S2(X) = (FRIS - FR2S) ** GA ** X
FUNT(X) = (R1 * Z(X1) - R2 * Z(X1) ** 2 + R1 * S2(X1) - R2 * S1(X1) ** 2 + S1(X1) * S1(X1) - Z1(X1) ** 2) / (X1 ** 2)
DATA BBS, REL, NPOL /1.E-7, 1.E-6, 3/
C MASSES AND H'S ARE NONDIMENSIONAL, WITH BLADE MASS AND RADIUS
C AS REFERENCES. NONROTATING NATURAL FREQUENCIES ARE
C DIMENSIONLESS, USING ROTOR SPEED AS REFERENCE. DISTANCES XBAB, SILB, SUPPL 41
C AND ZL ARE FRACTIONS OF SEMICHORD. XBAR, SIL, AND ZL ARE
C FRACTIONS OF ROTOR RADIUS.
C DO 63 K = 1, 8
DO 63 K = 1, 8
C NDFMC = 3
C DO 63 K = 1, 8
C SUM(K) = 0.
C do 63 K = 1, 8
\[ C_6 = (E_{1111}A_1 + 2 + E_{22}A_2 + 2) / E_{33} \]
\[ F_4 = E_{33} \]
\[ C_4 = (10R_{2}E_{11} + A_1 + 2 + 10R_{2}E_{22}A_2 + 2) / E_{33} \]
\[ G_A = 2 \cdot E_{1111}A_1 / E_{33} \]
\[ G_B = 2 \cdot E_{22}A_2 / E_{33} \]
\[ F_2 = \text{ADD2} \]
\[ H_A = E_{1111} / E_{33} \]
\[ H_B = E_{22} / E_{33} \]
\[ F_2 = \text{ADD2} \]

\[ R_1 = -HA - H_B \cdot (G_A / G_B) \cdot 2 \]
\[ R_2 = H_A \cdot (R_{22} / R_{11}) \]
\[ Z_L = F_4 - B_4 \]
\[ T_W = (R_{22} - C_4) \]
\[ F_2 H = B_2 - F_2 \cdot R_{22} + R_{11} \cdot (R_{22} / R_{11}) \]
\[ F_4 H = C_6 + R_{22} \cdot T_W + H_B \cdot (T_W / G_B) \cdot 2 \]
\[ G_2 H = B_2 - F_2 \cdot R_{22} + R_{11} \cdot (R_{22} / R_{11}) \]
\[ G_4 H = C_6 + R_{22} \cdot H_B \cdot (R_{22} / R_{11}) \]
\[ S_I = F_4 - R_{22} \cdot H_B \cdot (G_A / G_B) \cdot 2 \]
\[ G_2 A = F_1 - F_1 \cdot R_{22} \cdot (G_A / G_B) \cdot 2 \]
\[ U = R_2 \cdot F_1 H \]
\[ U_1 = R_1 \cdot G_2 H - R_2 \cdot F_1 H \]
\[ U_2 = R_1 \cdot G_4 H - R_2 \cdot F_4 H \]
\[ U_3 = R_2 \cdot S_I \]
\[ U_4 = R_1 \cdot G_2 H - R_2 \cdot S_I \]
\[ U_5 = S_I \cdot G_2 H - R_2 \cdot G_2 H \]
\[ U_6 = S_I \cdot G_4 H - R_2 \cdot S_I \]
\[ U_7 = S_I \cdot G_4 H - R_2 \cdot F_4 H \]
\[ A_1X(1) = U_2 \]
\[ A_1X(2) = U_2 \cdot U_1 \]
\[ A_1X(3) = U_2 \cdot U_3 \]
\[ A_1X(4) = U_2 \cdot U_4 \]
\[ A_1X(5) = U_2 \cdot U_4 \]

\[ \text{CALL POLLY}(4, R_{BS}, R_{EL}, \text{ANSX}, AAX) \]
\[ \text{XBAR} = 1.25 \]

\[ \text{DO 86 } I = 1, 4 \]
\[ \text{IP} = 2 \cdot I \]
\[ \text{IM} = \text{IP} - 1 \]

IF (RAB(1) > 0.10) GO TO 86
IF (RAB(1) < 0.90) GO TO 86
\[ \text{XFART} = \text{DSQRT}(\text{RAB}(1)) \]

\[ \text{CONTINUE} \]

IF (XFART = 1.25) GO TO 88

\[ \text{WRITE}(16, 87) \]

87 FORMAT(1H1, 10X, *NO SOLUTION FOR XBAR*)
STOP

88 CONTINUE

15 ALOW = (R_1 Z_2(XBAR) - R_2 Z_1(XBAR)) / (R_1 S_2(XBAR) - R_2 S_1(XBAR))
ALOW = ALOW / XBAR
BLOW = (C_6(XBAR) - G_A * ALOW * XBAR) / (XBAR / GB)
X_1 = -ALOW + BLOW
ETA = (BLOW / A_1 + ALOW / A_2) / (A_1 - A_2)
S_2L = F_4 / (B * HTGHS)
71    I1=2
    I2=3
    GO TO 74
72    I1=1
    I2=3
    GO TO 74
73    I1=1
    I2=2
74    IF OMS(1) .GT. OMS(2) GO TO 75
    MINI=1
    MIDI=2
    GO TO 76
75    MINI=1
    MIDI=1
76    SORT(1)=OMS(MINI)
    SORT(2)=OMS(MIDI)
    SORT(3)=OMS(MAXI)
    DO 77  I=1,3
77    OMEGA(I)=DSQRT(OMS(I))
    DO 302 I=1,3
302   ALPHAL(I,1,1)=1.
        DENB=RETA(2,1)*BETA(3,2)-BETA(3,1)*BETA(2,2)-OMS(1)
        ALPHA(1,2)=(BETA(1,2)*BETA(3,1)-BETA(3,2)*BETA(1,1)-OMS(1))/DENB
        ALPHA(1,3)=((BETA(2,2)-OMS(1))*BETA(1,1)-OMS(1))*RETA(1,2)*BETA(2,2)
        L(1,1))/DENB
        CHK(1)=BETA(1,3)*ALPHA(1,1)*BETA(2,3)*ALPHA(1,2)*(BETA(3,3)-OMS(1))
        11*ALPHA(1,1)
        DENB=OMEGA(2)(I)*BETA(3,2)-BETA(3,1)*BETA(1,2)
        ALPHA(2,1)=(BETA(1,1)*BETA(2,2)-BETA(2,1)*BETA(1,3))/DENB
        ALPHA(2,3)=(BETA(2,1)*BETA(1,3)-BETA(1,1)*BETA(2,2)-OMS(2))/DENB
        L(1,2))/DENB
        CHK(2)=RETA(1,3)*ALPHA(2,1)*BETA(2,3)*ALPHA(2,2)*(BETA(3,3)-OMS(2))
        11*ALPHA(2,1)
        DENB=RETA(2,3)*BETA(3,1)-BETA(3,3)*BETA(1,2)
        ALPHA(3,1)=(BETA(1,2)*BETA(3,3)-BETA(1,3)*BETA(3,1))/DENB
        ALPHA(3,3)=((BETA(2,2)-OMS(3))*BETA(1,1)-OMS(3))*RETA(1,3)*BETA(2,2)
        L(1,3))/DENB
        CHK(3)=BETA(1,2)*ALPHA(3,1)+(BETA(2,2)-OMS(3))*ALPHA(3,2)*BETA(3,3)
        L(1,3)
        WRITE(6,488) I, OMEGA(I), BETA(I,1), BETA(I,2), BETA(I,3), ALPHAL(I,1)
        LALPHAL(1,2), ALPHAL(1,3), CHK(I), I=1,3
        SORT(1)=0.
        SORT(2)=0.
    DO 382 J=1,3
        SORT(1)=0.
432   J=1,3
        GO TO 381
382   SORT(2)=0.
    GO TO 381
383   SORT(3)=0.
    GO TO 381
384   SORT(3)=1.

64
CALL ALTSOL(1, DELTA, SRTET, 1)

431 GAMMA(1, J) = SORT(II)
432 CONTINUE
WRITE(6, 11)
WRITE(6, 12) (I, GAMMA(I, 1), GAMMA(I, 2), GAMMA(I, 3), I = 1, 3)
AMPL = XMUVB * (1. - ROVB**3) / (1. - ROVB**4) * 1.33333333330
SA = SMALS * SILA + RM * S2LB + HIGHS
SB = SMALS * SILA**2 + RM * S2LR**2 + S2LB + HIGHS
DEL(1, 1) = XMUVB * (1. - ROVB**4) / (4. * (1. - SLAMZ * XBAB**2))
A = RROBR * EDMU
DEL(1, 2) = 2. * SLAMZ * XBAB * DEL(1, 1)
DEL(1, 3) = A1 * (SLAMZ * XBAB + SB - SA) / (1. - SLAMZ * XBAB**2)
A = B * HIGHS * S2LB
DEL(2, 1) = A2 / A1 * DEL(1, 1)
DEL(2, 2) = A2 / A1 * DEL(1, 2)
DEL(2, 3) = A2 * (SLAMZ * XBAB + SB - SA) / (1. - SLAMZ * XBAB**2)
A = R * SMALS * S2LB
DEL(3, 1) = - SLAMZ * XBAB * DEL(1, 1) / A1
DEL(3, 2) = -2. * SLAMZ * DEL(1, 1) / A1
DEL(3, 3) = (BDRR / RROBR)**2 + SLAMZ * (XBAB + SA - SP) / (1. - SLAMZ * XBAB**2)
A = (1. - SLAMZ * XBAB**2)
CMPA(2) = SMPAVB
CLV(2) = CLVB
NDIMC = 60
COSPSI = 1.
SINPSI = 0.
TO = ATOVB + ATCVB * COS PSI + ATSVB * SIN PSI
TOT(I) = TO - ATOVB
DN 50 I = 1, 3
SMALLG(I) = DEL(I, 1) * CLVB + DEL(I, 2) * CMPAVB
DN 51 I = 1, 3
GCAP(I, I) = 0.
DN 52 J = 1, 3
YPR(I, J) = 0.
DN 52 J = 1, 3
YPR(I, J) = GCAP(I, I) + ALPHA(I, J) * SMALLG(J)
Y(I, I) = GCAP(I, I) / CMS(I)
IF ( PLOT == 1 ) THEN
9CC0 FORMAT(77) TO = 'IPE13.6', ' Z=', ' IPE13.6', TOPR='1', IPE13.6
1 = ' ZPR=', ' IPE13.6', YPR='1', IPE13.6
RETURN

ENTRY SUPPI (UINF)

CMPA(3) = CMPA(2)
CMPA(2) = CMPAVB
CL(1) = 2. * CL(2) - CL(3)
CL(2) = CLVB
CL(1) = 2. * CL(2) - CL(3)
PSI = (BDAR / RDRBR) * NTIME * DXI
SIN PSI = SIN(PSI)
COS PSI = COS(PSI)
TOT(2) = TOT(1)
TO = ATOVB + ATCVB + COS PSI + ATSVB * SIN PSI
TOT(1) = TO - ATOVB
TO PR = (BDAR / RDRBR) * (ATSVB * COS PSI - ATCVB * SIN PSI)
DO 60 K = 1, 2
DO 64 I = 1, 3
64 SMALL G(I) = UINF **2 * (DEL(I,1)) * CL(K) + DEL(I,2) * CMPA(K)
A + DEL(I,3) * TOT(K)
DO 65 I = 1, 3
G CAP(I, K) = 0.
DO 65 J = 1, 3
65 G CAP(I, K) = GCAP(I, K) + ALPHA(I,J) + SMALLG(J)
CONTINUE
DO 62 I = 1, 3
Y(I,2) = Y(I,1)
YPR(I,2) = YPR(I,1)
WDXI = OMEGA(I) * DXI
SWDXI = SIN(WDXI)
CWDXI = COS(WDXI)
Y(I+1) = Y(I,2) * CWDXI + YPR(I,2) * SWDXI / OMEGA(I)
A + (GCAP(I,2) - GCAP(I,1)) * (SWDXI - WDXI) / CWDXI
B + GCAP(I,1) * (1. - CWDXI) / OMEGA(I)**2
62 YPR(I,1) = YPR(I,2) * CWDXI - OMEGA(I) * Y(I,2) + SWDXI
A + (GCAP(I,2) - GCAP(I,1)) * (WDXI * SWDXI + CWDXI - 1.)
B / WDXI + GCAP(I,1) * SWDXI / OMEGA(I)
DO 61 I = 1, 3
Z(I) = 0.
ZPR(I) = 0.
DO 61 J = 1, 3
61 I = Z(I) + GAMMA(I,J) * Y(J,1)
ZPR(I) = ZPR(I) + GAMMA(I,J) * YPR(J,1)
ALPH1 = TO + Z(3)
ALPH2 = TO PR + ZPR(3)
HEAVE = ZPR(1) - ZPR(2)
IF ( PLOTOP .LT., 0.)
WRITE (6,9000) TO, Z, TOPR, ZPR, Y, YPR, DEL, SMALLG
RETURN
1 FORMAT(5F10.4)
2 FORMAT(5F10.4)
3 FORMAT(1H1,10X,*ITERATION FOR XBAR DIVerged*)
4 FORMAT(1H1,5X,4HF1 =E13.5,5X,4HF2 =E13.5,5X,4HF3 =E13.5//5X,4HRM =E13.5///)
7 FORMAT(20X,6HX87R =E13.5,10X,6HX87B =E13.5//20X,6HL1/R =E13.5,10X,6SUPPL361

1/9X, 74K2/M2 =E13.51

41 FORMAT(/10X,5HR/R =E13.5,20X,6HRR/R =E13.5//)

44 FORMAT(1HI,20X,'POLYNOMIAL COEFFICIENTS'//7X,'5HPower,12X,5+BLADE',SUPPL 365)

126X,3HR2-0/7)

46 FORMAT(10,2030.9)

47 FORMAT(1HI,20X,'ROOTS OF POLYNOMIALS'//30X,'BLADE',60X,'2-1',20X)

14HREAL,21X,4HIMAG,31X,4HREAL,21X,4HIMAG/)

49 FORMAT(2D25.9,10X,2D25.9)

11 FORMAT(/10X,1HI,15X,10Hgamma(I,1),15X,10Hgamma(I,2),15X,10HgammaSUPPL 371)

1A(1,3)/7)

12 FORMAT((10,3E25.5)

488 FORMAT(1HI,8X,1HI,7X,5Homega,4X,9HBeta(I,1),4X,9Hbeta(I,2),4X,9HBeta SUPPL 374)

11A(I,3),3X,10HALPHA(I,1),3X,10HALPHA(I,2),3X,10HALPHA(I,3),9X,3HchSUPPL 375

1K//)

489 FORMAT(110,8E13.5)

721 FORMAT(7/10X,5HFR1 =E13.5,10X,5HFR2 =E13.5,10X,5HFR3 =E13.5//10X, SUPPL 378)

14HSA =E13.5,10X,4HSA =E13.5//)

END
SUPROUTINE SETUPS
C
IMPLICIT REAL*8 (A-H,O-Z)
C
REAL FTVB, FPVB, FPPVB, DIDRVB, XMVB, DELVB, XMUVB,
A FOVB, XMUVB, ATOVB, ATCVB, ATSVB, ROVB, RVB, NVH
REAL ELSIG, DXI, REB, ROBB, FRZ, ARR, AMPLU,_FREQ,
A NI, ALPH2, HEAVE, AROT, FREQF, PHIH, RYI, DRY,
I, TEST, UPRIM, XU, YU, XL, YL, ER1, ER2, ER3, RDBR,
C RROBR
H, CMPA, CMRAS, BARG, EML, HVOR, SSPA, SVOR, TORF, XIVOR
I, PLOTOP, PSLLOW, PSIUP
C
INTEGER TABLE(7, 80) /560 */
C
COMMON /BLI/ NTIME
C
COMMON /INTVBA/ FTVB(64), FPVB(64), FPPVB(64), DIDRVB(54),
A XMVB(64), DELVB, XMUVB, FOVB, XMUVB,
B ATOVB, ATCVB, ATSVB, ROVB, RVB
C NVB(64), NVB
C
COMMON /INPUTS/ NSBL, NSC, NOFF, NGAM, NSIG,
A NC01, NC02, NOFF, MSTOP, MTV, MTR,
B NOTAB, INDV, ELSIG, DXI, REL, ROBB,
C FZ, ARR, AMPLU, FREQ, AMPLH, ALPH2,
D HEAVE, AROT, FREQF, PHIH, RYI, RYI,
E DRY, YIOO, TEST, UPRIM, XU(5), YU(5),
F XL(30), YL(30), ER1, ER2, ER3, RDBR,
G RROBR
H, CMPA, CMRAS, BARG, EML, HVOR, NVOR, SSPA, SVOR, TORF, XIVOR
I, PLOTOP, PSLLOW, PSIUP
J, NCUT
C
CALL WHERE(TABLE)
CALL ZERION
C
CALL SETUP('ALPHI', 'ALPHI')
CALL SETUP('ALPHAI', 'ALPHI')
CALL SETUP('ALPH2', 'ALPH2')
CALL SETUP('ALPHA2', 'ALPH2')
CALL SETUP('AMPLU', 'AMPLU')
CALL SETUP('ARR', 'ARR')
CALL SETUP('AROT', 'AROT')
CALL SETUP('ATOVB', 'ATOVB')
CALL SETUP('ATCVB', 'ATCVB')
CALL SETUP('ATSVB', 'ATSVB')
CALL SETUP('BARG', 'BARG')
CALL SETUP('BDBR', 'BDBR')
CALL SETUP('CMPA', 'CMPA')

SETUPS 1
SETUPS 2
SETUPS 3
SETUPS 4
SETUPS 5
SETUPS 6
SETUPS 7
SETUPS 8
SETUPS 9
SETUPS 10
SETUPS 11
SETUPS 12
SETUPS 13
SETUPS 14
SETUPS 15
SETUPS 16
SETUPS 17
SETUPS 18
SETUPS 19
SETUPS 20
SETUPS 21
SETUPS 22
SETUPS 23
SETUPS 24
SETUPS 25
SETUPS 26
SETUPS 27
SETUPS 28
SETUPS 29
SETUPS 30
SETUPS 31
SETUPS 32
SETUPS 33
SETUPS 34
SETUPS 35
SETUPS 36
SETUPS 37
SETUPS 38
SETUPS 39
SETUPS 40
SETUPS 41
SETUPS 42
SETUPS 43
SETUPS 44
SETUPS 45
SETUPS 46
SETUPS 47
SETUPS 48
CALL SETUP(*CMPAS, 4, CMPAS)
CALL SETUP(*DELVB, 4, DELVB)
CALL SETUP(*IDORV, 4, IDORV, 64)
CALL SETUP(*DRY, 4, DRY)
CALL SETUP(*DTI, 4, DTI)
CALL SETUP(*ELSIG, 4, ELSIG)
CALL SETUP(*EMI, 4, EMI)
CALL SETUP(*ER1, 4, ER1)
CALL SETUP(*ER2, 4, ER2)
CALL SETUP(*ER3, 4, ER3)
CALL SETUP(*FPVB, 4, FPVB, 64)
CALL SETUP(*FPFRV, 4, FPFRV, 64)
CALL SETUP(*FRZ, 4, FRZ)
CALL SETUP(*FREQ, 4, FREQ)
CALL SETUP(*FREQF, 4, FREQF)
CALL SETUP(*FTVB, 4, FTVB, 64)
CALL SETUP(*FPOVB, 4, FPOVB)
CALL SETUP(*HEAVE, 4, HEAVE)
CALL SETUP(*NVOR, 4, NVOR)
CALL SETUP(*INDV, 4, INDV)
CALL SETUP(*LOWER, 4, LOWER)
CALL SETUP(*MXT, 4, MXT)
CALL SETUP(*MRTR, 4, MRTR)
CALL SETUP(*MSTOP, 4, MSTOP)
CALL SETUP(*MVBA, 4, MVBA, 64)
CALL SETUP(*NCOI, 4, NCOI)
CALL SETUP(*NCORD, 4, NCORD)
CALL SETUP(*NAGAM, 4, NAGAM)
CALL SETUP(*NOFF, 4, NOFF)
CALL SETUP(*NOTBL, 4, NOTBL)
CALL SETUP(*NOVT, 4, NOVT)
CALL SETUP(*NSBL, 4, NSBL)
CALL SETUP(*NSIG, 4, NSIG)
CALL SETUP(*NVB, 4, NVB)
CALL SETUP(*NY, 4, NY)
CALL SETUP(*NZ, 4, NZ)
CALL SETUP(*PHIM, 4, PHIM)
CALL SETUP(*PLOTOP, 4, PLOTOP)
CALL SETUP(*PSTOLW, 4, PSTOLW)
CALL SETUP(*PSIUP, 4, PSIUP)
CALL SETUP(*RVB, 4, RVB, 64)
CALL SETUP(*RRAB, 4, RRAB)
CALL SETUP(*REFB, 4, REB)
CALL SETUP(*RRDRBR, 4, RRDRBR)
CALL SETUP(*ROVB, 4, ROVB)
CALL SETUP(*RYI, 4, RYI)
CALL SETUP(*SSPA, 4, SSPA)
CALL SETUP(*SVOR, 4, SVOR)
CALL SETUP(*TEST, 4, TEST)
CALL SETUP(*TORF, 4, TORF)
CALL SETUP(*UPRIM, 4, UPRIM)
CALL SETUP(*XIVOR, 4, XIVOR)
CALL SETUP(*XL, 4, XL, 30)
CALL SETUP(*XMVB, 4, XMVB, 64)
CALL SETUP(*XMUVB, X, XMUVB)
CALL SETUP(*XMUAVB, X, XMUAVB)
CALL SETUP(*XU, X, XU, 30)
CALL SETUP(*XU, X, XU, 100)
CALL SETUP(*YL, Y, YL, 30)
CALL SETUP(*YU, Y, YU, 30)

PSILOW = 1.E10
PSIUP = -1.E10
PLOTNP = 1.
NOUT = 0

RETURN

END
SUBROUTINE BLCTK, Y, MST, MENO, NY, RX, DRY, DXI, RFR, JPRE, FLAM, XFLAM, TESRLC
1T, U, SCALE, UJ, UC, V, XEP, USEP, DISS, THE, LAMQ, MSEP, XC, USAV, SCALC
1LS, NTS, NTME, NCTBL, XTEST, NZ, NOUT)

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
COMMON /RAL/, NODUMY, NDINC, ISTO
DIMENSION USAV(300,100), SCALS(300)
DIMENSION X(300), Y(100), UJ(300,3), UC(100,3), V(100,2), XC(300)
DIMENSION D(100), S(100), SF(100), VISC(100,2), GRADV(100)
DIMENSION A(100), R(100), C(100), D(100), F(100)
DIMENSION ALPHA(100), ETA(100), GAMMA(100), DELTA(100)
DIMENSION SCALE(300,2), VAR1(100), VAR2(100)
DIMENSION FLAM(10), XFLAM(10), YR1(10), YR2(10)
DIMENSION U(300,10,2)
DIMENSION CAPG(100), CAPX(100), CAPY(100)
DOUBLE PRECISION AP(100), BP(100), CP(100), DP(100), FP(100), UP(100)

10 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

20 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

20 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

3 1/1)
12 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

20 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

3 1/1)
12 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

20 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

3 1/1)
12 FORMAT(14X, 36H ANALYSIS OF LAMINAR BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)
11 FORMAT(14X, 36H ANALYSIS OF TURBULENT BOUNDARY LAYER// 51X, 12HT1RLC
1ME STEP NO17// 51X, 12HT1RLC
1UE, 10X, 6H/DP/DX, 9X, 5HDELTA, 9X, 5HTHE, 9X, 5HSHA//)

3 1/1)
GO TO (543, 550), \textit{LOWER}

IF(LAMQ) 544, 544, 545

WRITE(MOUT, 11) 	extit{NTIME}, 	extit{NITS}
GO TO 550

WRITE(MOUT, 10) 	extit{NTIME}, 	extit{NITS}

CONTINUE

YTR = SQRT(RER)

UC(1, 1) = 0.

V(1, 1) = 0.

NV = NY - 2

NV1 = NY - 1

CALL YDIFF(NY, ALPHA, BETA, GAMMA, DELTA, SD, SE, SF, C2, C3, C4, Y)

DO 41 N = 1, NV1

VISC(N, 1) = 1.

DO 41 N = 1, NV1

1

VISC(N, 2) = 1.

DO 41 M = MST2, MST1

L = MST1 - M + 2

DO 50 N = 1, NV

40

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, M, UE, DXI, PRESS, SA, SB, SC, SR, SS)

DO 456 N = 1, NY

CALL SETIT(LAMQ, M, NV, REB, X, Y, UC, PRESS, GRAD, DELT, DISP, THETA, VISC, MTBL)

IF (MOD(M, NOUT1), NE. MST1, MD) GO TO 225

42

CONTINUE

MEND1 = MEND - 1

GRADS = GRAD(1)

GRADSS = GRAD(1)

C

THE MAIN CALCULATION STARTS HERE.

C

DO 99 M = MST1, MEND1

ITER = 0.

WALLG = 0.

MPI = M + 1

DELT = DELT / YTR

DISP = DISP * YTR

THET = YTHET / YTR

SHEAR = GRAD(1) / YTR

IF (MOD(M, NOUT1), NE. MST1, MD) GO TO 225

GO TO (561, 562), \textit{LOWER}

561 WRITE(MOUT, 12) M, X(M), XC(M), UE(M, 1), PRESS, DELT, DISP, THETA, SHEAR

1, \textit{MAXIT}

GO TO 225

562 WRITE(MOUT, 20) M, X(M), UE(M, 1), PRESS, REB, UPRIM

WRITE(MOUT, 24) DELT, DISP, THETA, DELT, DISP, THET

WRITE(MOUT, 21)

WRITE(MOUT, 224) (YIN), UC(N, Z), VIN, 11, GRAD(N), VISC(N, 1), NV = 1, NV1

WRITE(MOUT, 25) SHEAR

225 IF (GRADSS - GRADS - 1.1 E - 6) 229, 229, 408

4C8 XSX = X(M - 2) + (X(M - 1) - X(M - 2)) * GRADSS / (GRADSS - GRADS)

IF (XSX = X(M)) 409, 409, 229

72
WFS=\((X_{S}-X_{M-1}) / (X_{M}) - (X_{M-1})\)

GO TO 224

229 IF (GRAD(1)) 227, 227, 273

273 IF (NISP .GT. 0 .AND. THETA .LT. 0.) GO TO 223

283 CONTINUE

XSEP = XC(M-1)
USEP = UE(M-1,1)
XR = X(M-1)
WRITE (MOUT,23) XBL, XSEP
RETURN

227 WFS = GRADS2 / (GRADS2 - GRAD(1))

224 WFS1 = 1., -WFS
XSEP = WFS1 * XC(M-1) + WFS * XC(M)
XR = WFS1 * X(M-1) + WFS * X(M)
USEP = WFS1 * UE(M-1,1) + WFS * UE(M,1)
WFP = (XR - X(M-2)) / (X(M-1) - X(M-2))
WFP1 = 1. - WFP
DISS = DISS * WFP1 + DISS * WFP
THETSS = THETSS * WFP1 + THETSS * WFP
WRITE (MOUT,23) XBL, XSEP
IF (LAMQ.EQ.0.AND..M.LT..MTRAN+5) LAMQ = 1
GO TO 222

223 CONTINUE

IF (N0TL .EQ. 2 .AND. NITS GT. 1 .AND. M .GT. NZ .AND.
1 XC(M) .GT. XTEST) GO TO 283

IF (LAMQ) 801, 801, 902

802 CALL TRANS (UPRIM, PRESS, THETA, REB, UC, NY, FLAM, XFLAM, LAMQ)

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)

VAR2(N) = UC(N,3)

CALL YSET (RY, YSUB2, NY, Y)

WRITE (MOUT,351) YB1(NY), Y(NY)

DO 633 N = 2, NVPI

YIN = Y(N)

CALL TERP (YIN, YB1, VAR1, NY, UPAS1)

UC(N,21) = UPAS1

CALL TERP (YIN, YB1, VAR1, NY, UPAS2)

DO 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, NVPI

VAR1(N) = VISC(N,1)

VAR2(N) = VISC(N,2)

DO 636 N = 2, NVPI
YIN = Y(N)
CALL TERP(YIN, YB1, VAR1, NVPL, UPAS1)
VISC(N+1) = UPAS1
CALL TERP(YIN, YB1, VAR2, NVPL, UPAS2)

VISC(N+2) = UPAS2
DO 637 N = 2, NVPL
VAR1(N) = V(N+1)
DO 637 N = 2, NVPL
YIN = Y(N)
CALL TERP(YIN, YB1, VAR1, NVPL, UPAS1)
V(N+1) = UPAS1
CALL TERP(YIN, YB1, VAR2, NVPL, UPAS2)

638 V(N+2) = UPAS2
641 CONTINUE

C RESCALING CALCULATION ENDS HERE.
C CALL PGRAD(MX, UE, DXI, PRESS, SA, SB, SC, SR, SS)
C RECURSION RELATIONS ARE SET UP HERE.

C IF (ISTD, EQ. 1) GO TO 820
IF (SCALE(M+1,1) .LT. 1) 522, 522, 521
522 LACKU = 1
FACU1 = UE(M+1,2)/UE(M+1,1)
FACU2 = UE(M+1,1)/UE(M+1,1)
GO TO 820
523 LACKU = 2
DO 610 N = 1, NV
VAR1(N) = U(M+1, N+1)
610 VAR2(N) = U(M+1, N+2)
CALL YSET(SCALE(M+1,1), YSUB2, NV, YB1)
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) * CAPH(N) + DELTA(N) * CAPH(N) + SF(N) * CAPJ(N)
B(N) = BCONSA + CAPK(N) * SF(N) * CAPH(N) - GAMMA(N) * CAPH(N) - SE(N) * CAPJ(N)
C(N) = SD(N) * CAPG(N) - BETA(N) * CAPH(N) - SD(N) * CAPJ(N)
 DT(N) = -ALPHA(N) * CAPH(N)
IF (ISTD, EQ. 1) GO TO 576
88 CONTINUE
GO TO 874, 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, NV, UPAS1)
CALL TERP(YIN, YB2, VAR2, NV, UPAS2)
576 F(N) = PRESS + FCON*4.*UPAS1 - UPAS2 + CAPK(N)*SB*UC(N+2) - SC*UC(N+3)
88 CONTINUE

C SOLUTION FOR VELOCITY PROFILE STARTS HERE.

C DD 89 N = 2, NV
AP(N) = A(N)
RP(N) = B(N)
CP(N) = C(N)
DP(N) = D(N)

89 FP(N) = F(N)
DO 77 N = 2, NVML
CP(N) = CP(N) / RP(N)
DP(N) = DP(N) / RP(N)
FP(N) = FP(N) / RP(N)
RP(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(N+1, 1)
UP(NV) = (FP(NV) - IP(NY) * (DP(NV) + CP(NV))) / RP(NV)
DO 56 N = 3, NV
NN = NV + 2
66 UP(N) = FP(NN) - DP(NN) * UP(NN+2) + CP(NN) * UP(NN+1)
DO 65 N = 2, NV
65 UC(N, 1) = UP(N)
IF (ITER) 843, 841, 843
841 DO 842 N = 2, NVPL
V(N, 2) = V(N, 1)
842 VISC(N, 2) = VISC(N, 1)
DISS = DISS
DISS = DISS
THETS = THETS
THETS = THETS
GRAD = GRAD
GRAD = GRAD(1)
843 DO 55 N = 2, NVPL
55 V(N, 1) = V(N-1, 1) - 5*(Y(N) - Y(N-1)) * (SA*(UC(N, 1) + UC(N-1, 1)) - 3*BLC(N, 1) + UC(N, 1)) - 5*(UC(N, 1) + UC(N-1, 1))
IN(N, 2) + UC(N-1, 2) + SC*(UC(N, 1) + UC(N-1, 1))
DO 56 N = 1, NV
56 GRAD(N+1) = SD(N+1) * UC(N+2, 1) + SC(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 SFIT(T) LAMQ, NPI, N, REB, X, Y, UC, PRESS, GRAD, DELT, DISP, THETS, VISC, BLC
1IMTRAN
BLC 234
ITER = ITER + 1
GO TO (830, 809), LOWER
809 WRITE (MOUT, 810) ITER, GRAD(1)
830 IF (ITER = 9) 811, 811, 812
811 FPW = ABST GRAD(1) - WALLC
IF (WALLC = 1) 120, 120, 119
119 EPW = EPW WALLC
120 IF (EPW = TEST) 812, 814, 814
814 WALLC = GRAD(1)
GO TO 820
812 DO 44 N = 1, NV
UC(N, 3) = UC(N, 2)
UC(N, 2) = UC(N, 1)
44 CONTINUE
MAX = ITER
IF (ISTD .EQ. 1) GO TO 99
DO 48 N = 1, NV

48 USAV(M+1,N) = UC(N,1)
SCALS(M+1) = RY
99 CONTINUE
XSEP = 1.1
USEP = UE(MX,1)
222 CONTINUE
RETURN
END
SUBROUTINE PLOTS( PLOTOP , P , L )
REAL * 9 ORD(6)
DIMENSION P(200,7), TITL(56)
1, NFP(6)
DATA NL, N2, NO, N42 /1, 2, 0, 42 /
DATA ORD/ THETA-P , TORS , FLAP-H , RENG-H , /
1, CM-A / IF(PLOTOP .EQ. 0.) RETURN
IF( L .LT. 2) RETURN
IF ( PLOTOP .EQ. 2.) GO TO 2
PLOTOP = 2,
CONTINUE
CALL LDORMV ('CRIMI-PETE ', '30', '5100', )
3 NL = 1
DO 1 J = 1, 6
CALL EZPLTO(9, N1, NI, P, P(I,J+1), L, -N1, N2
1, N42, 1, 12, PSI-DEGREES', 8, ORD(J)
2, N1, XI, XU, NI, YL, YU, NI, NO, NL)
1 CONTINUE
NFP(1) = -1
NFP(2) = 66
NFP(3) = 50
NFP(4) = 50
NFP(5) = 680
CALL EZPLTO(9, N1, NI, P, P(I,2), L, -N1, N2
1, N42, 1, 12, PSI-DEGREES', 8, ORD(1)
2, NFP, N1, XI, XU, NI, YL, YU, NI, NO, NI)
NFP(1) = -2
NFP(2) = 66
NFP(4) = 350
NFP(5) = 380
CALL EZPLTO(9, N1, NI, P, P(I,6), L, -N1, N2
1, N42, 1, 12, 8, ORD(5)
2, NFP, NI, XI, XU, NI, YL, YU, NI, NO, NI)
NFP(2) = 50
NFP(4) = 690
NFP(5) = 40
CALL EZPLTO(9, N1, NI, P, P(I,7), L, -N1, N2
1, N42, 1, 12, 8, ORD(6)
2, NFP, NI, XI, XU, NI, YL, YU, NI, NO, NI)
NFP(1) = -1
NFP(2) = 50
NFP(3) = 50
NFP(4) = 690
CALL EZPLTO(9, N1, NI, P, P(I,3), L, -N1, N2
1, N42, 1, 12, PSI-DEGREES', 8, ORD(2)
2, NFP, NI, XI, XJ, NI, YL, YU, NI, NO, NI)
NFP(1) = -2
NFP(2) = 66
NFP(4) = 350
NFP(5) = 380
CALL EZPLTO(9, N1, NI, P, P(I,4), L, -N1, N2
1, N42, 1, 12, 8, ORD( 3)
CALL EZPLO T (\(N_2\), N1, N1, P, P(1,5), L, -N1, N2
1, N2, 1, 12, 8, ORD(4)
2, NFP, N1, XL, XU, N1, YL, YU, N1, N0, N1)
RETURN
END
SUBROUTINE STAG(X,NY,MSSTOP,MST,DXI,RY,DRY,X,Y,UE,UC,V,USAV,SCALS,STAG)

11SF)

C PROGRAM FOR CALCULATING THE BOUNDARY LAYER PROFILE NEAR

C THE STAGNATION POINT

C

COMMON /HL1/ NTIME, NDIMC, ISTD
DIMENSION USAV(300,100), SCALS(300)
DIMENSION PHI7(24), PHIP(24), FTAP(24)
DIMENSION X(300), Y(100), UE(300,3), UC(100,3), V(100,2)
DIMENSION FE(100), EFP(100)
DATA FTAP/9,2,4,6,8,1,1,2,1,4,1,6,1,8,2,2,2,2,2,4,2,6,2,8,3/
DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24
data phi12/0.0233,0.0981,0.1867,0.3124,0.4592,0.622,0.7967,0.9793,1.164/
data 1.352,1.5578,1.7553,1.9538,2.153,2.3526,2.5523,2.7522,2.9521,3.15/
data 1521,3.3521,3.5521,3.7521,3.9521/
data DATA PHIPE/0.0226,0.145,0.566,0.8859,0.7759,0.5867,0.952,0.23,0.5956/
data 18,0.9732,0.9839,0.9935,0.9946,0.9977,0.9984,0.9992,0.9996,0.9998,0.9999,1,1,1/
data 1,1,1,1/
data 

RAG=0.08
IF(ISFP) 10,10,5
5
RAG=.5
10 EF1=0.
EFP1=0.
DO 20 M=1,MX
20 IF(UF(M,1)) 20,20,19
15 MSP=M
GO TO 21
20 CONTINUE
21 ASAG=(UF(MSP+2,1)-UF(MSP+1,1))/(X(MSP+2)-X(MSP+1))
IF(ASTAG) 22,22,23
22 ASAG=(UE(MSP,1)-UE(MSP-1,1))/(X(MSP)-X(MSP-1))
30 SQAS=SQRT(ASTAG)
23 DELT=2.67*SQAS
30 IF(DIFL+YNY)-311,310,310
310 RY=RY+DRY
CALL YSFTR(RY,Y(2),NY,Y)
GO TO 309
311 CONTINUE
DO 30 N=2,NY
32 YET=Y(N)*SQAS
30 IF(YET-ETAP(NN)) 408,408,33
408 MARK=NN
GO TO 410
33 CONTINUE
EF(N)=YET-.6479
EF(N)=1.
GO TO 80
410 FRAC=(YET-ETAP(MARK-1))/(ETAP(MARK)-ETAP(MARK-1))
SFAC1=1.-FRAC
EF(N)=PHITZ(MARK-1)*SFAC1+PHIZ(MARK)*SFAC1
EF(N)=PHI(MARK-1)*SFAC1+PHI(MARK)*SFAC1
8C CONTINUE
M1=MSPI-MSSTOP
M2=MSPI+MSSTOP

79
M=M+1
M=M+1
MST=M+1
SCALS(M)=RY

DO 71 N=1,NV
   UC(N+2) = UC(N+2)
   UC(N+2) = UE(M,1)*EFP(N)
   V(N+2) = V(N,1)
   V(N+1) = -SQA5*EF(N)
   IF(ISTD .EQ. 1) GO TO 71
   USAV(M,N)=UC(N+2)
71 CONTINUE

IF(M-M2) 50,55,55
55 IF(UF(M,1)-BAG) 50,50,81
81 CONTINUE
RETURN
END
SUBROUTINE HTPR(PREC, XSIG, NSIG, ASZ, AS, AR, CMAT, RMAT, NGAM, NF, ACAP, TATPR)

NICK, NDBR, GAMMAW, UINF, UDOT, DXI, BCAP)

DIMENSION XSIG(100), ASZ(30), AS(30,30), AR(30), ACAP(1000)

DO NATURE PRECISION CMAT(50, 50, 1), RMAT(100)

PI = 3.14159

NGP1 = NGAM

DO 50 M = 1, NGP1

CMAT(M,1) = ASZ(M)

RMAT(M) = AR(M)

DC 25 N = 1, NGAM

CMAT(M, N+1) = AS(M, N)

50 CONTINUE

CALL ALSOL(NGP1, CMAT, RMAT)

DO 75 M = 1, NGP1

ACAP(M,1) = RMAT(M)

GAMMAW(L) = GAMMAACAP(DXI, PI)

SAVE = XSIG(NSIG+1)

SIG(NSIG+1) = 2.

CALL CPC(0, NGAM, NF, XSIG, NSIG, XSIG, NSIG, XSIG, ASC, ACAP, BCAP, THICK, TATPR)

1000, GAMMAW, UINF, UDOT, 1., SAVE, DXI, PREC)

SIG(NSIG+1) = SAVE

RETURN

END
SUBROUTINE UNPOP(NGAM, AR, ALAM, AFAC, RMAT, CMAT, XGAM, AS, ACAP, YX, NZ, NUNPOP)
IF, XSIG, ACAP, THICK, RDBC, UINF, XC, UF)
DIMENSION AR(30), ALAM(30), XGAM(30), AS(30,30), ACAP(30, 3), XSIG(130), UNPOP
IBCAP(100, 3), THICK(24), XC(300), JF(300, 3)
DOUBLE PRECISION RMAT(130), CMAT(60, 60)
NGP1=NGAM+1
DO 5 M=1, NGP1
SUB=AR(M)-ALAM(M)*AFAC/3.
RMAT(M)=SUB
CMAT(M, 1)=1.
CMAT(M, 2)=XGAM(M)
DO 5 N=2, NGAM
CMAT(M, N+1)=AS(M, N)
CALL ALSCL(NGP1, CMAT, RMAT)
DO 10 N=1, NGP1
ACAP(N, 1)=RMAT(N)
DO 15 M=1, MX
SIGN=1.
IF (M-N2) 12, 14, 14
12 SIGN=-SIGN
14 CALL QECAL(0, NGAM, NGAM, NF, XSIG, ACAP, BCAP, THICK, RDBC, 0., UINF, XC(M), UNPOP
1UE(M, 1), SIGN)
15 CONTINUE
RETURN
END
SURROUTINE ALSOL(NT, C, R)
DOUBLE PRECISION C, NDIMC, NOIMC, R(130)
DOUBLE PRECISION CMAX, SAVE, SUM,
COMM /R1/ NTIME, NDIMC
NT1 = NT-1
DC 99 J=1, NT1
CMAX = C(NT, J)
L=NT
DC 10 I=J, NT1
IF (DARS(CMAX) - DARS(C(I, J))) 5, 10, 10
5 CMAX = C(I, J)
L=1
10 CONTINUE
DC 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
JPI = J+1
DO 25 I=JPI, NT
DO 20 JJ=JPI, 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
IPI = I+1
SUM = 0.
DO 125 J=IPI, NT
125 SUM = SUM + R(J)*C(I, J)
150 R(I) = R(I) - SUM
RETURN
END
SURFPUTNF CPC (ISEP, NGAM, NF, XSIG, NSIG, XSIGA, NSIGA, XSIGB, NSIGB3, ACAP, CPC)
1, BCAP, THICK, RDBB, GAMAW, UNIF, UD, SIGN, XC, DXX, CP)
DIMENSION XSIG(100), XSIGA(100), XSIGB(100), ACAP(30, 3), BCAP(100, 3)
DIMENSION GAMAW(1000), THICK(24)
THETA=ARCT(XC)
RECIP=1. / (UNIF * UNIF)
SUM=0.
ANGLE=0.
DC 5 N=1, NF
ANGLE=ANGLE+THETA
5 SUM=SUM+THICK(N)*COS(ANGLE)
CP=UD*RECIP*(THICK(1)+2.*NCF-XC)*SUM
CALL OECAL (ISEP, NGAM, NSIG, NF, XSIG, ACAP, BCAP, THICK, RDBB, GAMAW(I), UI, CPC
INF, XC, U, SIGN)
CP=CP+2.*SIGN/UNIF-1.
CALL EGAMI (ISEP, NGAM, ACAP, RCAP(1, 1), XSIG(1), XSIG-NSIGA+1), GAMAW(I), XC, CPC
IVAL(I)
CALL EGAMI (ISEP, NGAM, ACAP, RCAP(1, 2), XSIGA(I), XSIGA(1), NSIGA+1), GAMAW(2), CPC
1X, VAL(I)
CALL EGAMI (ISEP, NGAM, ACAP, BCAP(1, 3), XSIGB(I), XSIGB(1), NSIGB+1), GAMAW(3), CPC
1X, VAL(3)
CP=CP+SIGN*RECIP*(1.5*VAL-2.5*VAL+2.5*VAL)/DXI
IF (ISEP) 20, 20, 10
CALL FSIGI (ISEP, XSIG, XSIGA, BCAP, XC, VAL(I)
CALL FSIGI (ISEP, XSIGA, XSIGA, RCAP, XC, VAL(I)
CALL FSIGI (ISEP, XSIGB, XSIGB, BCAP, XC, VAL(3)
CP=CP+RECIP*(1.5*VAL-2.5*VAL+2.5*VAL)/DXI
20 CP=CP
RETURN
END
SUBROUTINE CLC(INCO, ISEP, NGAM, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ALCM)
ICAP, BCAP, THICK, RDBR, GAMW, UINF, UDOT, DXI, AROT, CMPI
COMMON /CLCMAL/ CLVR, CMVR, CMPIVR

DIMENSION ARG(21), ARG4(21)

DIMENSION GAMAW(1000), THICK(24)

DIMENSION XSIG(100), XSIGA(100), XSIGR(100), ACAP(30, 31), RCAP(100, 3)

FORMAT(//40X, 4H1C = F13.5/40X, 4HCM = F13.5, 17H (ABOUT MINDOR))/40X, CLC

14CM = F13.5.24H (ABOUT PITCH AXIS - A = F7.4, 1H)

MCMB = 6
SAVE = THICK(1)
THICK(1) = 0
NT = 3.1 + 159/FLOAT(NCM)
CL = 0
CM = 0
XI = 1
ANGLE = 0
FLI = 0
FM1 = 0

IF(ISEP) 5, 7

7
XATT = XSIG(NSIG + 1)

IF(XATT < -95) 8, 5, 5

8
XAO = XATT + 5. F-4

XAP = XAO + 0.25

C1 = .5*(1. + XATT)

C2 = C1 + XATT

C1P = .5*(1. - XAP)

C2P = C1 + C2P

DC 10 I = 1, NCM

ANGLE = ANGLE + DT

XI1 = CL* C1* COS(ANGLE) + C2

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, BCA

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

FLI = CPL - CPU

FMPl = XIPI - FLI

CL = CL + (XIPI - XI) * (FLI + FLI)

CM = CM + (XIPI - XI) * (FMPl + FMPl)

XI = XIPI

FLI = FLI

10

FM1 = FMPl

XI = 1

FLI = 0

FM1 = 0

ANGLE = 0

DC 15 I = 1, NCM

ANGLE = ANGLE + DT

XI1 = CL* C1* COS(ANGLE) + C2

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, BCA

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

CALL CPCITSEP, NGAM, I, XSIG, NSIG, XSIGA, NSIGA, XSIGR, NSIGR, ACAP, RCAP

FLI = CPL - CPU

FMPl = XIPI - FLI

CL = CL - (XIPI - XI) * (FLI + FLI)

CM = CM - (XIPI - XI) * (FMPl + FMPl)

XI = XIPI

FLI = FLI

10
CM=CM-(XIPl-XI)*FMIPl+FMI
X1=XIPI
FPL=FPI

15 FMI=FMIPI
DO 16 I=1,21
CALL CPC(ISEP,NGAM,1,XSIG,NSIG,NSIGA,NSIGIA,NSIGB,NSIGB,ACAP,BCAP,TCLCM)
IHICK,RDBR,GAMAW,UNF,UDOT,1,0,XIPI,DI1,CPL)
CALL CPC(ISEP,NGAM,1,XSIG,NSIG,NSIGA,NSIGIA,NSIGB,NSIGB,ACAP,BCAP,TCLCM)
IHICK,RDBR,GAMAW,UNF,UDOT,1,0,XIPI,DI1,CPL)
ARGL(I)=CPL-CPU
ARGM(I)=XIPI*ARGL(I)

16 XIPI=XIPI+.00125
SUML=0.
SUMM=0.
DO 17 I=1,19,2
SUML=SUML+2.*ARGL(I)+4.*ARGL(I+1)
SUMM=SUMM+2.*ARMG(I)+4.*ARMG(I+1)
CL=CL+0.833333F-3*(SUML+ARGL(Z1)-ARGL(Z1))
CM=CM+0.833333F-3*(SUML+ARGL(Z1)-ARGL(Z1))
RCON=16.*RCAP(1,1)*SORT(5*E-4*(XATT-XSIG(1)))/UNF
CL=CL+RCON

17 CL=CL+RCON
GO TO 130

5 DO 99 I=1,NCDE
ANGLE=ANGLE+DT
X1=X1+DT

99 ANGLE=ANGLE+DT
X1=X1+DT

99 ANGLE=ANGLE+DT
X1=X1+DT

CL=CL+RCON
GO TO 130

END
SUBROUTINE QFCAL(ISEP, NGAM, NSIG, NF, XSIG, ACAP, RCAP, THICK, RDBA, GAMMA)
1, UNIF, XC, U, SIGN)
DIMENSION ACAP(30, 31), RCAP(100, 3), XSIG(100)
DIMENSION THICK(24)
EPS = 1.0E-6
CORR = 7.07107/1.0 - 43662*SQRT(RDBA) + 25*RCABA
SINT = SQRT(1.0 - XC*XC)
THETA = ARCT(XC)
COUNT = 0.
SUM = 0.
SINT2 = SIN(5.0*THETA)
COST2 = COS(5.0*THETA)
IF (SINT - EPS) 4, 5, 6
4 FACT = THETA**5.
GO TO 8
6 FACT = (1.0 - XC)/SINT
8 DO 10 N = 1, NF
COUNT = COUNT + 1.
ANGLE = THETA*COUNT
SUM = SUM + THICK(N)*COUNT*FACT*SIN(ANGLE) - COS(ANGLE)
10 CONTINUE
U = 2.0*SIGN*UNIF*COST2*SUM + ACAP(1, 1)*SINT2 + 25*COST2*(1.0 + XC)*(3.0 + XC)
11.0*GAMMA
SUM = 0.
ANGLE = 0.
DO 12 N = 1, NGAM
ANGLE = ANGLE + THETA
12 SUM = SUM + ACAP(N, 1, 1)*SIN(ANGLE)
U = U + COST2*SUM
IF (ISEP) 25, 99, 25
25 SUM = 0.
XSEP = XSIG(1)
XATT = XSIG(NSIG + 1)
DO 40 N = 2, NSIG
SUM = SUM + RCAP(N, 1)*EBT(XSIG(N - 1), XSIG(N), XSIG(N + 1), XC)
IF (XC - XATT - EPS) 45, 44, 46
46 FACT = (1.0 - XATT)**(-1.5)*SQRT((XATT - XSEP)*(1.0 - XC)/(XC - XATT))*(1.0 + 3.0)
IF (XSEP - XC) 49, 49, 48
48 FACT = SIGN*(1.0 - SQRT((XSEP - XC)/(XATT - XC)))
GO TO 55
49 FACT = SIGN
55 U = U + COST2*(BCAP(1, 1)*FACT + SIGN*SMJ)
99 U = SIGN*UNIF*SQRT((1.0 + XC) + CORR*U)*SQRTT(I + XC + 5.0*RCABA)
RETURN
END
SUBROUTINE YVR(Y, I)
REAL Y(I)
REAL YVR
COMMON /INPTVR/ FTVB(64), FPVR(64), FPPRVR(64), DIDRVB(64), YVR
A XMVR(64), DELVB, XMUVB, FDVR, XMUAVR, YVB
B ATSR, ATCVR, ATSBR, ROVB, RVB(64), YVB
C MVB(64), MVB
Y(1) = (RVB(1) - DELVB)**2 * MVB
Y(2) = FPVB(1)**2 * MVB
Y(3) = FTVB(1)**2 * DIDRVB
Y(4) = (DELVB - FVA(1)) * FTVA(I) * XMVB(I) * MVB(I)
Y(5) = FPVB(1) * FTVB(1) * XMVB(I) * MVB(I)
Y(6) = RVB(I) * (DELVB - RVB(I)) * MVB(I)
Y(8) = (RVB(I) - DELVB) * FPPRVR(I) * FTVA(I) * XMVB(I) * MVB(I)
IPI = I+1
IF(IPI .GE. NVB) GO TO 12
SUM = 0.
DO 10 J = IPI, NVB
  SUM = SUM -(RVB(4+1) - RVB(4)) * (RVB(4+1) * MVB(J+1)
  A + RVB(J) * MVB(J))
10 RETURN
12 Y(7) = FPPRVR(1) ** 2 * SUM / 2.
RETURN
END
SURACUTING POLLY(4,ABS,REL,AN,AA)

IMPLIED REAL*8 (A-H,Q-Z)

CCMPLXX ROOTS OF A POLYNOMIAL BAIRSTOWS METHOD

DIMENSION A(30), An(60), C(26), ABAR(26), B(30), AA(30)

III = 1

7 NPL = N+1
   NPP = N+2
   DO 66 CI = 1, NPL
66 LLI = NPP + I

601 A(I) = AA(LLI)

13 DO 14 K = 1, NPL
14 ABAR(K) = A(K)
   ABSSQ = ABS*ABS
   RELSQ = REL*REL
   NBAR = N
   B(1) = A(1)
   C(1) = A(1)

15 IF(NBAR-2).GT.200,210,17
17 P(I) = 2
   Q(I) = 1
18 IF(R = 0)
19 P(I) = P(I)*5.
   Q(I) = Q(I)*1O.
33 P = P(I)
34 L = 1
   LAST = NBAR
   OPEST = 9.999996
   DO 50 K = 3, NPL
C BAIRSTOW ITERATION
37 B(2) = ABAR(2) - P*B(1)
   DO 40 K = 3, NPL
40 B(K) = ABAR(K) - P*B(K-1) - Q*B(K-2)
45 C(2) = B(2) - P*C(1)
   DO 50 K = 3, LAST
50 C(K) = B(K) - P*C(K-1) - Q*C(K-2)
   D(C(LAST)) = C(LAST) - C(LAST-1)*C(LAST-2)
   D = D(C(LAST))/D
   D = D(C(LAST-1))/D

C TEST FOR CONVERGENCE
   Relp = Delp/P
   RELQ = DELQ/Q
   RELPS = Relp*RELPS
   RELQS = RELQ*RELQ
   DELSQ = RELPS + RELQS
   P = P + Delp
   Q = Q + DELQ
   IF(Relp = RELSQ).GT.79.765
65 IF(Delp*DELP = ABSSQ).GT.70.70.80
70 IF(DELQ*DELQ = ABSSQ).GT.120.120.80
80 GO TO 99
ITFR=ITER+1
IF (250-ITER) LE 0,0,0
100 IF (DTEST-DELSQ) LE 0
110 DTEST=DELSQ
R(2)=A(2)-P*B(1)
DO 115 K=3,NPI
115 R(K)=A(K)-P*K*(K-1)-Q*R(K-2)
GO TO 45
C ITERATION HAS CONVERGED
120 GO TO (130,143),L
130 L=2
LAST=N
GO TO 110
C FACTOR OUT QUADRATIC
140 NBAR=NBAR-2
NBPI=NBAR+1
AR(2)=AR(2)-P*AR(1)
DO 150 K=3,NBP1
150 AR(K)=AR(K)-P*AR(K-1)-Q*AR(K-2)
GO TO 250
C SOLVE LINEAR EQUATION
200 NBAR=NBAR-1
R1=AR(2)/AR(1)
R2=0.
GO TO 262
C NORMALIZE QUADRATIC
210 P=AR(2)/AR(1)
Q=AR(3)/AR(1)
NBAR=NBAR-2
C SOLVE NORMALIZED QUADRATIC
250 R1=-P/2.
C1=R1*R1-O
IF (C1*270-280,260)
260 C1=DSORT(C1)
R2=R1-C1
R1=R1+C1
GO TO 290
262 C1=0.
GO TO 290
270 C1=C1
C1=DSORT(C1)
290 R2=R1
290 C2=-C1
A(I+1)=C1
A(I+1)=R1
A(I+2)=C2
A(I+3)=R2
I=I+4
IF (NBAR-11 LE 200,15)
C SPECIAL CONDITIONS
310 WRITE ((6,600)
600 FORMAT(1X,50HNO CONVERGENCE IN 250 ITERATIONS,POLLY HAS SPoken)
4 CONTINUE
RETURN
END
SUBROUTINE SETTLG0,M,NV,REG,X,Y,UC,PRESS,GRAD,DELT,DISP,THET,VIS
1SC,MTRAN)

SUBROUTINE FOR CALCULATION OF BOUNDARY LAYER THICKNESS.

DISPLACEMENT THICKNESS, MOMENTUM THICKNESS AND FLOU VISCOSITY.

DIMENSION X(300),Y(100),UC(100,3),VIS(100,2),GRAD(100)

RTR=SQRT(REB)

NY = NV + 2

UDGE = .995*UC(NY,1)

DO 10 N=1,NV

10 CONTINUE

41 NOELT = N

GO TO 20

10 CONTINUE

20 DELT = Y(INDEL)*((UDGE-UC(NOELT,1))*(Y(INDEL+1)-Y(INDEL))/UC(NDEL))

[T(1,1)-UC(TNDEL,1))]

DO 50 N=1,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.

UDGE = UC(NY,1)

DO 60 N=2,NV

60 SUM = SUM+(Y(N)-Y(N-1))*((UDGE-UC(N,1))*UC(N,1)+UDGE-UC(N-1,1))

THETA = .5*SUM/GL/NR*UD(EDGE**2)

IF(LGO) 53,53,56

53 NVPI=NV+1

FAC = 1.

IF(N-MTRAN) 31,32,32

32 IF(MTRAN+5-M) 31,31,33

33 EASE = (X(M)-X(MTRAN))/(X(MTRAN+5)-X(MTRAN))

31 CONTINUE

INNER=0

FAC1 = .16*RTR*EASE

FAC2 = .016*UDGE*DISP*REG*EASE

FAC1 = -RTR/26.

FAC2 = -PRESS/RTR

TAUW = GRAD(1)/RTR

DO 160 N=2,NVPI

160 ALTER = 1.+FAC2/(1.+5.5*(Y(N)/DELT)**6)

IF(INNER) 402,401,402

402 VISC(N,1)=ALTER

GO TO 160

401 CONTINUE

TAUW=TAUW-Y(N)*FAC2

IF(TAUW) 701,701,702

7C1 VISC(N,1)=1.

GO TO 703

702 PX=Y(N)*FAC2*SQRT(TAUNY)

VISC(N,1) = 1.*FAC1*Y(N)*Y(N)*ABS(GRAV(1))*(1.-EXP(EX1)**2

7C3 IF(VISC(N,1)=ALTER) 150,160,521

521 VISC(N,1)=ALTER

INNER=1

91
16C CONTINUE
SAVE=1.
DO 162 N=2,NV
RAVE=VISC(N,1)
VISC(N,1)=(VISC(N+1,1)+RAVE+SAVE)/3.
162 SAVF=RAVE
56 CONTINUE
RETURN
END
MISSING TEXT
35 CP=CP4
  IF(X-X4) 55,50,50
50 CP=CP+(X-X4)*COEF
55 CONTINUE
  FPRES(M)=/UINF*CP+SUM
20 CONTINUE
  RETURN
END

MIXER 56
MIXER 57
MIXER 58
MIXER 59
MIXER 60
MIXER 61
MIXER 62
MIXER 63
SUBROUTINE RARK (DFLL, THET1, RE9, XCI, U1, XC5, DCP, DEL5, X, XC, M5, N, X5, UBR4)

N, U, ALT, REFL, USTOP)

DIMENSION X(300), U(300, 3)

FCAP(X) = -19.556*X + 107.555*XX - 336.33*X**3 + 508.81*X**4 - 295.76*X**5

UI(X) = -4.6523*X + 4.8425*XX - 4.5293*X**3 + 4.5293*X**4

U12(X) = -0.04529*X - 1.9161*X**2 + 9.1843*X**3 - 4.4225*X**4

FDELT(X) = EXP(-13.5773 - 13.252*X - 14.379*X**2 - 0.7561**3 + 0.3707**3)**4

FAICHI(X) = EXP(-3.7461 - 0.38777*X - 4.1967*X**2 + 0.71046*X**3 + 0.032162*X**4)

DELI(X) = -0.04532*LAG(X) - 3.9242*X + 5.4535*X**2 - 1.39147*X**3 - 1.38425*X**4

I**4

25 FORMAT(1MH1, 14X, 3X, 3HANALYSIS OF LEADING-EDGE RARKLE/////34X, 1MH1, 1X, 1MH1)

IHU, 19X, IHU, 18X, 9D15P)

30 FORMAT(20X, 4F20.5)

MOUT = 6

H1 = 25

M5 = 4.29

I(xc1 - xc1(m1)) = 1, 4, 5

4 M1 = M

GOTO 6

5 CONTINUE

6 X1 = (M1 - 1) + (X1(m1) - X1(m1) + 1) + (X1 - X1(m1)(1) - X1(m1))(X1 - X1(m1)(1))

X4 = X1*REEL/1U1**2

ARG = ALG5(X4 - X1)/TREP*DEL5*DEL5*U11

H4 = .25*FAICHI(ARG)

DFEL4 = .53*FDELT(ARG) * DEL5

X5 = X4 + 10.5 * DEL4 * (1 - (.49/4.29)**2)

IF ((U1 - USTOP) = GT11) 41, 41, 10

43 FORMAT(1MH1, 1X, 1MH1)

41 FORMAT(1MH1, 2H20X, 1MH1)

40 FORMAT(1MH1, 14X, 1MH1)

ALTL = ALTCC*DEL5

IF (X5 = X1.LT. ALTL) X5 = X1 + ALTL

35 FORMAT(1MH1, 1X, 1MH1)

DCP = DCP + 1**2

33 FORMAT(1MH1, 1X, 1MH1)

DO 7 N = 1, 1, N

32 FORMAT(1MH1, 1X, 1MH1)

IF (X5 = X(M)) 16, 16, 7

16 M5 = M

GOTO 8

7 CONTINUE

8 FACT1 = (X5 - X(M5 - 1177) (X5(M5) - X(M5 - 1))

FACT1 = 1.0 - FACT

XC5 = XC1(M5 - 1) * FACT1 + XC1(M5) * FACT

U5 = UF (M5 - 1, 1) * FACT1 + UF (X5, M5, 1) * FACT

WRITE (*MOUT, 251)

WRITE (*MOUT, 10) X1, U1, H1, DEL1

WRITE (*MOUT, 10) X5, U5, H5, DEL5

RETURN

END

95
SUBROUTINE YSET(R,A,NY,Y)       
DIMENSION Y(100)                
RPI=1.+R                        
Y(1)=0.                         
Y(2)=A                          
DO 10 N=3,NY                    
  Y(N)=RPI*Y(N-1)-A*Y(N-2)      
10  RETURN                      
END
SURROUNTE  H4 X4 (IND1, X1, DEL1, THET1, X5, REA, U1, X3, H3, X4, H4)  H4 X4 1
CURLF (H) = 26.703/H+305.33*ALOG(H)-211.3*H+3327, 1.68*H-24.03, 9.34*H  H4 X4 2
FDEL (ARG) = EXP(-2.5773-3.252*ARG-0.3799*ARG-0.076511*ARG-0.0039737*ARG)  H4 X4 3
FATCH (X1) = EXP(-3.7481+0.038772*X1+0.1967*X1+0.071046*X1+0.0032162*X1)*H  H4 X4 4
1*4)  H4 X4 5
10 FORMAT (/20X, 54HA SOLUTION FOR X4 COULD NOT BE OBTAINED IN 1000 TRY/ H4 X4 6
11 (ALS)  H4 X4 7
MOUT =6  H4 X4 8
C  H4 X4 9
C IF IND1 IS NONZERO, THE BOUNDARY LAYER IS TURBULENT  H4 X4 10
C AT SEPARATION.  H4 X4 11
C  H4 X4 12
1 IF (IND1) 2, 5, 12
2 H3 = THET1/DEL1  H4 X4 14
X3 = X1  H4 X4 15
DEL3 = DEL1  H4 X4 16
GO TO 20  H4 X4 17
5 X3 = X1+5,F4/(UL1*REA)  H4 X4 18
ARG = ALOG((X3-X1)/(REA*DEL1*DEL1))  H4 X4 19
H3 = THET1*FATCH(ARG)/DEL1  H4 X4 20
DEL3 = 5.8*FDEL(ARG)*DEL1  H4 X4 21
IF (X3-X5) 20, 15, 15  H4 X4 22
15 H4 = 0.429  H4 X4 23
X4 = X5  H4 X4 24
GO TO 50  H4 X4 25
20 CCONTINUE  H4 X4 26
IGN = 0  H4 X4 27
DIST = X5 - X1  H4 X4 28
UNDER = 0  H4 X4 29
H4 = H3*H3  H4 X4 30
COEF1 = DEL3*H3  H4 X4 31
COEF2 = 10.5*DEL3*H3  H4 X4 32
SUB = X3-COEF1*CURLF(H3)  H4 X4 33
95 OVER = H4  H4 X4 34
H4 = 5*(H4+UNDER)  H4 X4 35
X4 = CURLF(H4)*COEF1+SUB  H4 X4 36
ALTER = X5-COEF2*(1.-(H4/0.429)**21)*H4  H4 X4 37
IGN = IGN+1  H4 X4 38
IF (X4-ALTER) 41, 50, 42  H4 X4 39
41 IF (IGN-1000) 95, 61, 61  H4 X4 40
42 IF (ABS(X4-ALTER)/DIST-.001) 50, 50, 43  H4 X4 41
UNDER = H4  H4 X4 42
H4 = 5*(OVER+H4)  H4 X4 43
X4 = CURLF(H4)*COEF1+SUB  H4 X4 44
ALTER = X5-COEF2*(1.-(H4/.429)**21)*H4  H4 X4 45
IGN = IGN+1  H4 X4 46
IF (X4-ALTER) 52, 50, 51  H4 X4 47
51 IF (IGN-1000) 43, 61, 61  H4 X4 48
52 IF (ABS(X4-ALTER)/DIST-.001) 50, 50, 95  H4 X4 49
61 H4 = 0.429  H4 X4 50
X4 = X5  H4 X4 51
WRITE (MOUT, 10)  H4 X4 52
50 CONTINUE  H4 X4 53
RETURN  H4 X4 54
END  H4 X4 55
SUBROUTINE SFTSX(NSEP1,XSEP,XATT,XSIG,ANGLE)
DIMENSION XSIG(100)
A=.5*(XSEP+XATT)
B=.5*(XATT-XSEP)
ARG=0.
DO 5 N=1,NSEP1
  XSIG(N)=A-B*COS(ARG)
  ARG=ARG+ANGLE
RETURN
END

SFTSX 1
SFTSX 2
SFTSX 3
SFTSX 4
SFTSX 5
SFTSX 6
SFTSX 7
SFTSX 8
SFTSX 9
SFTSX 10
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
FUNCTION GAM1(ACAP, DXI, PI)

DIMENSION ACAP(30,3)

GAM1 = PI * (-1.5 * ACAP(1,1) - 0.75 * ACAP(2,1) + 2.0 * ACAP(1,2) + ACAP(2,2) - 0.5 * ACAP(1,3) - 2.5 * ACAP(2,3)) / DXI

RETURN

END
FUNCTION FB(X1, X2, X3, Y)
D1=1./(X2-X1)
D2=1./(X3-X2)
T1=ABS(Y-X1)
T2=ABS(Y-X2)
T3=ABS(Y-X3)
EPS=1.E-6
IF(T1-EPS) 2,3,3
  F1=0.
  F2=ALOG(T2)
  F3=ALOG(T3)
  GO TO 10
3  F1=ALOG(T1)
  IF(T2-EPS) 4,5,5
  F2=0.
  F3=ALOG(T3)
  GO TO 10
4  F2=ALOG(T2)
  IF(T3-EPS) 6,7,7
  F3=0.
  GO TO 10
5  F3=ALOG(T3)
  GO TO 10
6  FB=(Y-X1)*F1+D1+(D1+D2)*(X2-Y)*F2+(Y-X3)*F3+D2)/3.14159
RETURN
END
SUBROUTINE EGAM\(I(N,NG,A,B,XSEP,XATT,GAMMA,Y,GI)\)
DIMENSION A(30,3)
SINT=SQR\(T(1.-Y*Y)\)
THE\(T=AR\)CT\(Y)\)
SUM=0.
COUNT=1.
DO 6 N=2,NG
COUNT=COUNT+1.
6 SUM=SUM+A(N+1,NU)*(SINT(COUNT+1.)*THE\(T)A)/(COUNT+1.)-SINT(COUNT-1.)*THE\(T)A)/(COUNT-1.)
GI=(3.14159-THE\(T)A+SINT)*(A(1,NU)+.5*A(2,NU))+.5*SUM-.25*GAMMA*(1.+GAMMA)
IF(Y-XATT) .LT. 8,8,7
7 DIFF=1.-XATT
IF(DIFF-1.E-6) 8,8,9
8 CONTINUE
RETURN
END
SUBROUTINE ESIGI(NU, X, XS, S, Y, SI)
DIMENSION XS(100), Y(0, 3)
SUM = 0.
DO 10 I = 2, NX
  SUM = SUM + S(I, NU) * GA(XS(I-1), XS(I), XS(I+1), Y)
  SI = SIGI(1, NU) * SINT(XS(1), XS(NX+1), Y) + SUM
RETURN
END

ESIGI 1
ESIGI 2
ESIGI 3
ESIGI 4
ESIGI 5
ESIGI 6
ESIGI 7
ESIGI 8
<table>
<thead>
<tr>
<th>F U N C T I O N</th>
<th>G B (X1, X2, X3, X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GB = ABINT(X1, X2, X) - ABINT(X3, X2, X)</td>
<td></td>
</tr>
<tr>
<td>GB = GB / 3.14159</td>
<td></td>
</tr>
<tr>
<td>R E T U R N</td>
<td></td>
</tr>
<tr>
<td>F N D</td>
<td></td>
</tr>
</tbody>
</table>

| GB | 1 |
| GB | 2 |
| GR | 3 |
| GA | 4 |
| GB | 5 |
FUNCTION ABINT(A,B,X)
ARGA=ABS(X-A)
ARGR=ABS(X-B)
COEF=2.*(B-A)
API=A+1.
BPI=R+1.
IF(ARGA-1.E-6) 2,3,3
2
CA=0.
GO TO 5
3
CA=ALOG(ARGA)
IF(ARGA-1.E-6) 4,5,5
4
CR=0.
GO TO 6
5
CR=ALOG(ARGA)
6
ABINT=(CA-.5)*ARGA**2-(CB-.5)*ARGB**2=(ALOG(API)-.5)*API**2+(ALOG(API-1))
1BPI-.5)*BPI**2-COEFF((X-C)*(B-1.)*BPI*ALOG(BPI-1.))
ABINT=ABINT/COEF
RETURN
FND
FUNCTION BINT(XS, XZ, X)
RTS = SORT(1 + XS)
RTZ = SORT(1 + XZ)
BINT = -1 - X * RTS * RTZ
IF (XZ - X) 2, 3, 3
  RTSX = SQR((X - XS))
  RTZX = SQR((X - XZ))
  BINT = BINT + (XZ - XS) * ALOG((RTSX + RTZX) / (RTS + RTZ)) + RTSX * RTZX
  GO TO 50
3  IF (X - XS) 5, 5, 4
  BINT = BINT + (XZ - XS) * ALOG((SQR((XZ - XS)) / (RTS + RTZ))
  GO TO 50
5  RTSX = SQR((XS - X))
  RTZX = SQR((XZ - X))
  BINT = BINT + (XZ - XS) * ALOG((RTSX + RTZX) / (RTS + RTZ)) - RTSX * RTZX
50  CONTINUE
RETURN
END
SUBROUTINE SCALE(SAL, NSAL, FRZ, ARR, ROBAR)

DIMENSION SAL(300)
DELZ = FRZ * ROBAR
EN = ARR / FRZ
DO 5 N = 1, 300
IF (EN - N) 4, 4, 5

4 NE = N
GO TO 6

5 CONTINUE

6 NE = NSAL - NE
EN = FLOAT(NG)
NGM1 = NG - 1
SBL(1) = 0.
DO 7 N = 2, NE

7 SAL(N) = SAL(N-1) + CELZ
FRACT = 2.2 / DELZ
FRAC1 = FRAC1 - 1.
R = FRACT * (1. / FLOAT(NGM1))
SAVE = R
R = R - (R ** NG - FRAC1 * R + FRAC1) / (EN ** NGM1 - FRAC1)
IF (ABS(SAVE - R)**1.6 - 6) 9, 9, 8

9 RPL = R + 1.
DO 10 N = NE, NSBL

10 SAL(N+1) = RPL * SAL(N) - R * SAL(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 - 0.001) 2, 2, 10
2 GC TC (1, 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 CONTINUE
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)
   CONTINUE
99 RETURN
SUBROUTINE EVAL(NNF, XX, SSC, SST, CCA, TTB, CCM, TTM)
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.)
THF = 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)
CC = SUM1*SIN(THE)*CCM
TTB = (1. - COS(THE))*SUM2*TTM
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.
DC 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

110
SUBROUTINE SEC((Xu,Yu,XL,YL,NOFF,NF,RDCR,TMAX,CMAX,ST,SC) SECT 1  
C PROGRAM TO COMPUTE COEFFICIENTS TN AND ON OF THE FOURIER SERIES SECT 2  
C REPRESENTATION OF SECTION THICKNESS AND CAMBER DISTRIBUTIONS SECT 3  
DIMENSION Xu(30),Yu(30),XL(30),YL(30),YUC(30),YLC(30),ST(24),SC(24) SECT 4  
1),NUM(50),TBAR(50),CPar(50) SECT 5  
12 FORMAT(/'4X,26HINPUT AND COMPUTED OFFSETS/) SECT 6  
13 FORMAT(19X,4HX1/C,12X,4HYU/C,11X,5HYUC/C,23X,4HYL/C,11X) SECT 7  
1,5HYLC/C/) SECT 8  
14 FORMAT(19X,3F16.5,3X,3F16.5) SECT 9  
NA=6 SECT 10  
RNA=6. SECT 11  
NK=FLOAT(NF) SECT 12  
MCUT=6 SECT 13  
PI = 3.14159 SECT 14  
DEL T = PI/(2.*NF) SECT 15  
NTC = 2*NF - 1 SECT 16  
NINT = NTC + 2 SECT 17  
NSIMP = NTC + 1 SECT 18  
RDCR=5*RDCBC SECT 19  
VARY = 0 SECT 20  
CR = 0. SECT 21  
TB = 0. SECT 22  
THETA = 0. SECT 23  
DO 89 K=1,NTC SECT 24  
89 THETA = THETA + DEL T SECT 25  
XI = .5*(1. + COS(THTFA)) SECT 26  
DO 90 LAM=2,NOFF SECT 27  
90 IF(X1-XU(LAM)) 110,90,90 SECT 28  
110 YUINT = YU(LAM-1) + XI - XU(LAM-1))*YU(LAM) - YU(LAM-1))/(XU(LAM) SECT 29  
1) - XU(LAM-1)) SECT 30  
GC TO 111 SECT 31  
9C CONTINUE SECT 32  
111 DO 80 LAM=2,NOFF SECT 33  
80 IF(X1-XL(LAM)) 210,80,80 SECT 34  
210 YLINT = YL(LAM-1) + (XI - XL(LAM-1))*YL(LAM) - YL(LAM-1))/(XL(LAM) SECT 35  
1) - XL(LAM-1)) SECT 36  
GC TO 112 SECT 37  
112 CONTINUE SECT 38  
113 TBAR(K+1) = .5*(YU(INT) - YL(INT) SECT 39  
9 IF(CR(K+1)) SECT 40  
9 CR(K+1) = .5*(YU(INT) + YLINT) SECT 41  
9 TMAX = 0. SECT 42  
CMAX = 0. SECT 43  
DO 79 K = 2,NSIMP SECT 44  
79 IF(TBAR(K)-TMAX) 801,802,802 SECT 45  
801 TMAX = TBAR(K) SECT 46  
802 CONTINUE SECT 47  
702 CMAX = CR(K) SECT 48  
75 CONTINUE SECT 49  
75 IF(CMAX-L.E.-5) 1201,1202,1202 SECT 50  
1201 CMAX=1. SECT 51  
1202 CONTINUE SECT 52  
1203 IF(TMAX-L.E.-5) 1140,1141,1141 SECT 53  
1140 TMAX=1. SECT 54  
1141 DO 69 K=2,NSIMP SECT 55  
69 TBAR(K) = TBAR(K)/TMAX SECT 56
CBAR(K) = CRAR(K)/CMAX
TBAR(1) = 0.
CRAR(1) = 0.
TBAR(NINT) = 0.
CRAR(NINT) = 0.
TTA = TBAR(NA)
TTB = TBAR(NA+1)
TTC = TBAR(NA+2)
TAA = DELT*(RNA-1.)
TBA = TAA + DELT
TCC = TBA + DELT
KA = .5*COS(TAA)
XB = .5*COS(TBB)
XC = .5*COS(TCC)
SLOPE = ((TTT-XTT)*(XB-XA)/(XC-XB) + (TTT-XTT)*(XC-XR)/(XB-XA))/XSEC
IC-XA =
THETA = 0.
COSB = COS(TBB)
DO 456 I=2,NA
THETA = THETA + DELT
COST = COS(THETA)

456 TBAR(I) = (SQR((1.-COST)/(1.-COSB))**1.5)*(TTB(1.+COST-2.*COSB))/XSEC

1.-COSB) + .5*SLOPE*(COST-COSB))
NLE = 2*NFR + 1 - NA
COSR = 1. + COS(PI-RNA*DELT)
THETA = PI
SINAS = SIN(RNA*DELT)**2
COSAS = COS(RNA*DELT)
ANG=0.
DO 457 I=2,NA
IND = 2*NFR + 2 - I
THETA = THETA - DELT
COST = 1. + COS(THETA)
ANG = ANG + DELT
COEF = (SINAS - SIN(ANG)**2)/(COSR1*(COS(ANG)+COSAS))

457 TBAR(IND) = (SQR((RDBC*COST1)*COEF/TMAX*TBAR(NLE)**(COST1/COSR1)**I)

1.5)/(2.-COST1)
THETA = TAA
DO 458 I = NAPI,NLE
THETA = THETA + DELT

458 TBAR(I) = TRAR(I)/(1.-COS(THETA))
THETA = 0.
DO 459 I = 2,NSIMP
THETA = THETA + DELT

459 CBAR(I) = CRAR(I)/SIN(THETA)

KK = 0.
DO 59 K=1,NF
KK = KK + 1.
THETA = 0.
DO 777 I=1,NINT
DUM(I) = TBAR(I)*SIN(THETA*KK)

777 THETA = THETA + DELT
CALL SIMP(NSIMP,DELV,DUM,VARY)

STK(I) = 2.*VARY/PI

SEC 56
SEC 57
SEC 58
SEC 59
SEC 60
SEC 61
SEC 62
SEC 63
SEC 64
SEC 65
SEC 66
SEC 67
SEC 68
SEC 69
SEC 70
SEC 71
SEC 72
SEC 73
SEC 74
SEC 75
SEC 76
SEC 77
SEC 78
SEC 79
SEC 80
SEC 81
SEC 82
SEC 83
SEC 84
SEC 85
SEC 86
SEC 87
SEC 88
SEC 89
SEC 90
SEC 91
SEC 92
SEC 93
SEC 94
SEC 95
SEC 96
SEC 97
SEC 98
SEC 99
SEC 100
SEC 101
SEC 102
SEC 103
SEC 104
SEC 105
SEC 106
SEC 107
SEC 108
SEC 109
SEC 110
THETA = 0.

DO 988 I = 1, NINT
   DO M = 1, 8
      DO J = 1, 8
         DUM(I) = CARR(I)*SIN(THETA*RKK)
      END
   END
   EP8 THETA = THETA + DELT
   CALL SIMP(NSIMP,DELT,DUM,VARY)
988 CONTINUE

59 SC(X) = 2.*VARY/PI
DO 969 I = 1, NOFF
   X = XU(I)
   CALL FVAL(NF,X,SC,ST,CR,TR,CMAK,TMAX)
969 CONTINUE

569 YUC(I) = CB + TB
   NO 869 I = 1, NOFF
   X = XL(I)
   CALL FVAL(NF,X,SC,ST,CR,TR,CMAK,TMAX)

E69 YLC(I) = CB - TR
   SUM1 = 0.
   COUNT = 0.
   DO 659 I = 1, NF
      COUNT = COUNT + 1.
559 SUM1 = SUM1 - ST(I)*COUNT*(-1.)*I**2
   RCDBC = R*(TMAX*SUM1)**2
   RCDAC = 2.*RCDBC
   TMAX = 2.*TMAX
   CMAK = 2.*CMAK
   WRITE (MOUT,12)
   WRITE (MOUT,13)
   WRITE (MOUT,14) (XI(I),YU(I),YUC(I),XL(I),YLC(I),YLC(I),I = 1, NFF)
RF TURN
END
SUBROUTINE CORDX(NSRL, NZ, RDBB, SRL, X, XC)
C
C BOUNDARY LAYER COORDINATES AND CORRESPONDING CHORDAL
C COORDINATES ARE COMPUTED HERE.
C
DIMENSION SRL(3), X(300), XC(300)

336 FORMAT(10X, 3H1 ITERATION TO COMPUTE XC FOR M = 15, 32H DID NOT CONVERGE)
1000 STEPS.]

337 FORMAT(1H1, 25X, 1HM, 20X, 1HS, 25X, 1HX, 2X, 2HXC//)

338 FORMAT(22X, 15, 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) - SRL(MM)
DO 256 M=NZ, MX
MM = M + 1 - NZ
256 X(M) = SRL(NZ) + SRL(MM)
DO 257 M=1, MX
IF(NZ-M) 333, 257, 335
257 CONTINUE
K = M + 1 - NZ
GO TO 334
333 K = M + 1 - NZ
GO TO 334
335 K = NZ - M + 1
GO TO 334
336 XC(M) = -1. + SRL(K)
IF(SBL(K)-RZERO) 341, 341, 342
341 XC(M) = -1. + SRL(K)**2/(4. * RZERO)
342 CONTINUE
DO 258 M=1, 1000
SAVE = XC(M)
CALC1 = SQRT((1. + XC(M)/RZERO)
CALC2 = SQRT((1. + XC(M)/RZERO)
XC(M) = XC(M) + (CALC1 * CALC2 + ALOG(CALC1 * CALC2)) / CALC2
IF(ABS(SAVE-XC(M)) - 1.E-6) 257, 257, 258
258 CONTINUE
WRITE((MOUT, 336) M
257 CONTINUE
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 CONTINUE
RETURN
END

114
SUBROUTINE PGRAD(*, X, UE, DXI, PRESS, SA, SR, SC, SR, SS)

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)*1

IXIM*UE(M+2,1)+ETAM*UE(M+1,1)-ZETAM*UE(M,1))

SA=1./D1Z+1./D1M1
SR=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, LAM *)

C
C SUBROUTINE TO TEST FOR TRANSITION IN A LAMINAR BOUNCERY LAYER.
C
DIMENSION UC(100,3), FLAM(10), XFLAM(10)
F(X) = 1.11746 - 1.05826 - 3*X - 1.1023*4*X*X
TKAY = PRESS * REP * THETA ** 2 / UC(NY, 2)
IF (TKAY > 0.077) 2, 2, 99
2 IF (ARS(TKAY) > 0.01) 3, 3, 4
3 ARG = TKAY ** 0.49
 GO TO 5
5 ARG = 0.
2 DO 6 N = 1, 1000
 SAVE = ARG
 ARG = ARG - (ARG ** 2 - TKAY) / (F(ARG) - 1.11746 - ARG ** 3.1746 ** 3 - ARG ** 5 ** 5.115E-4)
 IF (F(ARG) - SAVE/ARG) < 1.0E-6) 7, 7, 6
6 CONTINUE
7 IF (ARG < 0.1) 8, 8, 5
8 EF = 1.75
5 GO TO 10
10 DO 15 N = 1, 10
 IF (ARG - XFLAM (N)) < 24.24, 15
24 NBAR = N
5 GO TO 16
15 CONTINUE
16 EF = FLAM (NBAR) / (ARG - XFLAM (NBAR - 1)) / (FLAM (NBAR) - FLAM (NBAR - 1)) / (XTRANS)
 IF (FLAM (NBAR) - XFLAM (NBAR - 1)) > 0
25 R = 0.5 * EF
 A = 3.36 * UPRIM / UC(NY, 2) ** 2
 RTH = F(ARG) / SQRT (A ** 2 + 9860. ** A) - B1/A
 IF (REB * THETA - RTH) > 99.50, 50
50 LAMQ = 0
95 CONTINUE
 END
RETURN
TRANS 1
TRANS 2
TRANS 3
TRANS 4
TRANS 5
TRANS 6
TRANS 7
TRANS 8
TRANS 9
TRANS 10
TRANS 11
TRANS 12
TRANS 13
TRANS 14
TRANS 15
TRANS 16
TRANS 17
TRANS 18
TRANS 19
TRANS 20
TRANS 21
TRANS 22
TRANS 23
TRANS 24
TRANS 25
TRANS 26
TRANS 27
TRANS 28
TRANS 29
TRANS 30
TRANS 31
TRANS 32
TRANS 33
TRANS 34
TRANS 35
TRANS 36
SUBROUTINE CAPS(ITER, N, CAPG, CAPH, CAPJ, CAPK, SR, SS, SD, SE, SF, VISC, V, UCAPS)
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)) - SCAPS
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))
CAPK(N) = 5 * (CAPK(N) + UC(N, 1))
CONTINUE
RETURN
END
SUBROUTINE TERPI(YIN,YBASE,VARY,NY,VALUE)

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   NRAR=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)
10  CONTINUE
   RETURN
   END
SURTOUTINE YDIFF(NY,ALPHA,RETA,GAMMA,DELTA,SD,SE,SY,C2,C3,C4,Y) YDIFF 1
DIMENSION ALPHA(100),RETA(100),GAMMA(100),DELTA(100) YDIFF 2
DIMENSION SD(100),SE(100),SY(100),Y(100) YDIFF 3
NV=NY-2 YDIFF 4
NVPI=NY+1 YDIFF 5
DC 40 N=2,NV YDIFF 6
ALPHA(N) = 2.*((2.*Y(N)-Y(N-1)-Y(N+1))/((Y(N+2)-Y(N-1))*(Y(N+2)-Y(N+1)))) YDIFF 7
1+1)*(Y(N+2)-Y(N)) YDIFF 8
DELTA(N) = 2.*((Y(N+2)+Y(N+1)-2.*Y(N))/((Y(N+2)-Y(N-1))*(Y(N+1)-Y(N))) YDIFF 9
1-1)*(Y(N)-Y(N-1)) YDIFF 10
RETA(N) = (DELTA(N)*(Y(N)-Y(N-1))**3-ALPHA(N)*(Y(N+2)-Y(N))**3)/YYDIFF 11
1(N+1)-Y(N))**3 YDIFF 12
GAMMA(N) = -ALPHA(N)-RETA(N)-DELTA(N) YDIFF 13
CONTINUE YDIFF 14
DC 39 N=2,NVPI YDIFF 15
SD(N) = (Y(N)-Y(N-1))/((Y(N+1)-Y(N-1))*(Y(N+1)-Y(N))) YDIFF 16
SE(N) = 1./((Y(N)-Y(N-1))-1.)/(Y(N+1)-Y(N)) YDIFF 17
SF(N) = (Y(N+1)-Y(N))/((Y(N)-Y(N-1))*(Y(N+1)-Y(N))) YDIFF 18
CONTINUE YDIFF 19
C2 = Y(3)*Y(4)/(Y(2)*Y(3)*Y(4)-Y(2)) YDIFF 20
C3 = -Y(2)*Y(4)/(Y(3)*Y(4)-Y(3)) YDIFF 21
C4 = Y(2)*Y(3)/(Y(4)*Y(4)-Y(3)) YDIFF 22
RETURN YDIFF 23
END YDIFF 24
SUBROUTINE ELDER(BCAP, XSIG, NSIG, UINF, ELD, Y, YMAX)
DIMENSION RCAP(100,3), XSIG(100)
BCAP(NSIG+1,1)=0.
XS=XSIG(1)
XZ=XSIG(NSIG+1)
IF(XZ-1,16,16,1
1 DEADL=XZ-XS
YMAX=1.E-10
SUM=5*(XSIG(2)-XS)*RCAP(2,1)
DO 10 N=2,NSIG
X=XSIG(N+1)
SUM=SUM+5*(X-XSIG(N))*RCAP(N+1,1)+BCAP(N,1))
10 IF(IN-NSIG) 4,2,4
2 ANGLE=1.5708
GO TO 6
4 ANGLE=ATAN(SQRT((X-XS)/(XZ-X)))
6 Y=SUM+BCAP(1,1)*(DEADL*ANGLE-SQRT((X-XS)*(XZ-X)))
IF(Y-YMAX) 10,10,8
8 YMAX=Y
10 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
GO TO 20
20 CONTINUE
RETURN
END

ELDER 1
ELDER 2
ELDER 3
ELDER 4
ELDER 5
ELDER 6
ELDER 7
ELDER 8
ELDER 9
ELDER 10
ELDER 11
ELDER 12
ELDER 13
ELDER 14
ELDER 15
ELDER 16
ELDER 17
ELDER 18
ELDER 19
ELDER 20
ELDER 21
ELDER 22
ELDER 23
ELDER 24
ELDER 25
ELDER 26
ELDER 27
ELDER 28
ELDER 29
SUBROUTINE REATT(UC, VX, X, VX, Y, DRY, UE, X5, DEL5, MST, REB)  
DIMENSION UC(100,3), V(100,2), Y(100)  
DIMENSION X(300), IF(300,3)  
DIMENSION TAR1(24), TAR2(24), TAB3(24), TAB4(24), XITAB(24)  
DATA TAB1(24), TAB2(24), TAB3(24), TAB4(24), XITAB(24) /  
1.38, 8.35, 7.32, 6.22, 5.31, 4.4, 3.57, 2.22, 1.26, 0.66, 31.14, 0.01, 1.0, 0. /  
DATA TAB2(24), TAB3(24), TAB4(24), XITAB(24) /  
1.47, 95.7, 2.64, 4.56, 4.9, 4.18, 2.99, 1.86, 1.11, 0.62, 0.32, 0.04, 0.0, 0. /  
DATA TAB3(16,65), 15.8, 1.67, 13.8, 12.91, 11.66, 13.65, 5.43, 8.71, 1.11, 0. /  
DATA TAB4(10,12,10,05,9.93,9.78,9.58,9.17,8.72,8.08,7.6,7.2,6.85,11. /  
DATA XITAB(/3001,0002,0005,0011,0002,0005,0011,02,03,04,05,06,07,08,09,11,12,14,16,18,22,27,29,33,35, /  
FORMAT(///40X,23HAT REATTACHMENT, BETA =E13.5)  
MOUT=6  
RTR=SCRAT(REB)  
UC(1,2)=0.  
UC(1,3)=0.  
V(1,1)=0.  
V(1,2)=0.  
DO 5 M=1, MX  
IF(X5=XMT(1), 4, 6)  
5  
CONTINUE  
6  
XA=X(MST-2)  
XB=X(MST-1)  
UA=UE(MST-2,1)  
UB=UE(MST-1,1)  
ZA=LOG10(UA*DEL5*REB)  
PGRAD2=(*UA-UB)/*(UA+UB)*X(B-XA))  
BETM2=(.0974-SORTDEL5*PGRAD)/(1.0249+.004565*ZA)  
IF(BETM2-1.0, 8, 7)  
7  
BETM2=1.  
GO TO 10  
8  
IF(RETM2-3, 9, 9, 10)  
9  
BETM2=3  
10  
BETM2=1.7/(BETM2*BETM2)  
WRITE(10,MOUT,3) BETA  
AGAM=.097*BETM2-.0249*BETA  
9GAM=.004565*BETA  
AH=1.0-(5.343.9*BETM2)*.0974-.0249*BETM2  
BH=ETM2*5.333.061.004565  
GAMI=AGAM*8GAM*ZA  
DERV=UA*FRF*EXP(-ZA)*GAMA*GAMA*(1.+BETA*(1.+AH+ZH*ZA))/(AH+BH+AH*REATT  
11  
IZA=  
ZA=ZA+DERIV*(X-B-XA)  
DELX=EXP(IZA)/(UA*FRF)  
GAMA=AGAM-BGAM*ZA  
DELZ=.35*DELX*RTR*BETM2/GAMB  
12  
IF(DELZ-Y(NY-3)) 14, 12, 12  
RY=RY+DRY  
CALL YSET(RY, Y(2), NY, Y)  
GO TO 11
14 IF( BETA-41 = 102, 101, 101
101 TERPA=1.-4./BETA
INDEX=3
GO TO 110
102 IF( BETA-21 = 104, 103, 103
103 TERPA=5*BETA-1.
INDEX=2
GO TO 110
104 TFRP= BETA-1.
INDEX=1
110 K=0
TFRP1=1.-TERPA
50 K=K+1
GO TO (16, 17, 99), K
16 G=GAMA
DELTA=DELS
UEDGE=UA
L=3
GO TO 18
17 G=GAMB
DELTA=DELS
UEDGE=UA
L=2
18 XICN=G/DTA*RTR*BETM2
UCM=RTR*(UEDGE*G)**2
EFCO=G/BETM2
NLAM=NY
DO 75 N=2, NY
XI=N(Y)*XICN
19 UC(N,L)=UEDGE
GO TO 75
20 CALL TERPF(XI, INDEX, TAB1, TAB2, TAB3, TAB4, XI TAB, FP1)
INDEX=1
CALL TERPF(XI, INDEX, TAB1, TAB2, TAB3, TAB4, XI TAB, FP2)
FP=TERP1*FP1+TERPA*FP2
UC(N,L)=UEDGE*(1.-EFCO*FP)
IF(N-NLAM) 21, 75, 75
21 ALTER=UCM*Y(N)
IF(ALTER=-UC(N,L)) 33, 33, 32
32 UC(N,L)=ALTFR
GO TO 75
33 NLAM=N
CONTINUE
GO TO 50
21 UC(N,L)=ALTFR
GO TO 75
33 NLAM=N
CONTINUE
GO TO 50
11 99 DO 60 K=2, 3
99 SAVE2=0.
DO 60 N=3, NY
SAVE1=UC(N-1, K)
UC(N-1, K)=SAVE2+SAVE1+UC(N, K1)/3.
60 SAVE2=SAVE1
DO 65 N=2, NY
DUXP=COO(C(N, 2)-UC(N, 3)
\[ V(n+1) = V(n-1) + (Y(n) - Y(n-1)) \times (dx_2 + dx_1) \]

RETURN
END
SUBROUTINE ELPI T(ALPH1, ALPH2, EMI, TORF, THETZ, UINF, DXI, CMPA, CYPAS) ELPI T 1
SAVET=ALPH1 ELPI T 2
STEP=TORF*DXI ELPI T 3
SINS=SIN(STEP) ELPI T 4
CSS=COS(STEP) ELPI T 5
CONST=2.*EMI*(UINF/Torf)**2 ELPI T 6
ALPH1=THETZ+(ALPH1-THETZ)*CSS+ALPH2*SINS/TORF+CONST*(2.*CMPA-CMPA) ELPI T 7
1S)*(1.-CSS)+CONST*(CYPAS-CMPA)*(SINS-STEP*CSS)/TORF*DXI ELPI T 8
ALPH2=ALPH2*CSS-TORF*SINS*(SAVET-THETZ)+CONST*(CMPA-CMPA)*(1.-CSS) ELPI T 9
1S)/DXI+CONST*CMPA*TORF*SINS RETURN ELPI T 10
END ELPI T 12
SUBROUTINE VWASH(BARG, H, S, NVOR, XL, UINF, VZIP, XGAM, NGPL, DXI)

DIMENSION VZIP(30), XGAM(3)

DO 10 N=1, NGPL
DIFF = XGAM(N) - XL
SUM = 0.
DO 5 K=1, NVOR
SUM = SUM + DIFF/(DIFF**2 + H)
5 DIFF = DIFF - S
10 VZIP(N) = VZIP(N) + SUM * BARG
RETURN
END
SUBROUTINE WASI(XGAM, NGAM, TIME, ALPH1, ALPH2, HEAVE, AROT, FREQ, PHIH, UWASH)
DIMENSION XGAM(30), VZIP(30), CAMR(24)

NGP1 = NGAM+1

ANGLE = FREQ*TIME
GO TO (108,120), INDV

GO TO (110,120), MTR

100 CONST = -ALPH2*COS(ANGLE)*UINF+HEAVE*COS(ANGLE+PHIH)*ALPH1*UINF
FACT = -ALPH2*FREQ*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.
DO 20 N=1, NF
COUNT = COUNT+1.
20 SUM = SUM + COUNT*CAMR(N)*CCS(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*CAMR(N)*SIN(COUNT)
50 VZIP(M) = UINF*SUM+CONST+FACT*(AROT-X)
10 CONTINUE
RETURN
END
APPENDIX B

DETERMINATION OF COUPLING PARAMETERS
APPENDIX B

DETERMINATION OF COUPLING PARAMETERS

The characteristic equation for the rotor blade is

\[ \sum_{k=0}^{3} B_{2k} \lambda^{2k} = 0 \]

where

\[ B_0 = f_0 - \frac{\bar{\omega}^2 \beta \Theta}{M_{\beta\beta} M_{ee}} - \frac{\bar{\omega}^2 \phi \Theta}{M_{\phi\phi} M_{ee}} \]

\[ B_2 = f_2 + 2 \frac{\bar{\omega}^2 \beta \Theta}{M_{\beta\beta} M_{ee}} T_\beta \Theta + 2 \frac{\bar{\omega}^2 \phi \Theta}{M_{\phi\phi} M_{ee}} T_\phi \Theta \]

\[ - \frac{T_\beta \Theta}{M_{\beta\beta} M_{ee}} - \frac{T_\phi \Theta}{M_{\phi\phi} M_{ee}} \]

\[ B_4 = f_4 - \frac{\bar{\omega}^2 \beta \Theta}{M_{\beta\beta} M_{ee}} - \frac{\bar{\omega}^2 \phi \Theta}{M_{\phi\phi} M_{ee}} \]

\[ + 2 \frac{M_{\beta \Theta} T_\beta \Theta}{M_{\beta\beta} M_{ee}} + 2 \frac{M_{\phi \Theta} T_\phi \Theta}{M_{\phi\phi} M_{ee}} \]

\[ B_6 = 1 - \frac{M_{\beta \Theta}^2}{M_{\beta\beta} M_{ee}} - \frac{M_{\phi \Theta}^2}{M_{\phi\phi} M_{ee}} \]
in which

\[ f_0 = \bar{\omega}_\beta^2 \bar{\omega}_\phi^2 \bar{\omega}_\theta^2 \]

\[ f_2 = \bar{\omega}_\beta^2 \bar{\omega}_\phi^2 + \bar{\omega}_\phi^2 \bar{\omega}_\theta^2 + \bar{\omega}_\phi^2 \bar{\omega}_\theta^2 \]

\[ f_4 = \bar{\omega}_\beta^2 + \bar{\omega}_\phi^2 + \bar{\omega}_\theta^2 \]

The characteristic equation for the two-dimensional system is found to be

\[ \sum_{k=0}^{3} D_{2k} \lambda^{2k} = 0 \]

where

\[ D_0 = f_0 - \bar{\omega}_\phi^2 h_a a_1^2 - \bar{\omega}_\beta^2 h_b b_1^2 \]

\[ D_2 = f_2 - \bar{\omega}_\phi^2 g_a \bar{x} a_1 - \bar{\omega}_\beta^2 g_b \bar{x} b_1 - h_a a_1^2 - h_b b_2 \]

\[ D_4 = f_4 - C_4 \bar{x}^2 - g_a \bar{x} a_1 - g_b \bar{x} b_1 \]

\[ D_6 = 1 - C_6 \bar{x}^2 \]

in which

\[ h_a = \frac{M_{\beta\beta}}{R^2 M_{\theta\theta}} \quad h_b = \frac{M_{\phi\phi}}{M_{\theta\theta}} \]
\[ g_a = 2h_a A_1 \quad g_b = 2h_b A_2 \]

\[ c_4 = \bar{\omega}_\beta^2 h_a A_1^2 + \bar{\omega}_\beta^2 h_b A_2^2 \]

\[ c_6 = h_a A_1^2 + h_b A_2^2 \]

\[ a_1 = A_1 (\bar{\omega}_\beta^2 l_s + r_m \bar{\omega}_\phi^2 l_{s_2}) - B \bar{\omega}_\phi^2 l_{s_2} \]

\[ b_1 = A_2 (\bar{\omega}_\beta^2 l_s + r_m \bar{\omega}_\phi^2 l_{s_2}) + B \bar{\omega}_\phi^2 l_{s_2} \]

Equating \( D_0/D_6 \) to \( B_0/B_6 \), \( D_2/D_6 \) to \( B_2/B_6 \) and \( D_4/D_6 \) to \( B_4/B_6 \) provides three relations in the three unknowns \( x, l_s, \) and \( l_{s_2} \). If \( a_1 \) and \( b_1 \) are eliminated, the following equation for \( x \) is obtained:

\[(r_1 t_2 - r_2 t_1)^2 + (r_1 s_2 - r_2 s_1)(t_2 s_1 - t_1 s_2) = 0\]

where

\[ r_1 = -\left[ h_a + \frac{h_b g_a^2}{g_b^2} \right] \quad r_2 = \left[ \frac{\bar{\omega}_\phi^2}{\bar{\omega}_\beta^2} - 1 \right] h_a \]

\[ s_2 = (\bar{\omega}_\beta^2 - \bar{\omega}_\phi^2) g_a x, \quad s_1 = s_2 + \frac{2 h_b g_a}{g_b^2} \frac{F}{x} \]

\[ t_1 = (1 - c_6 x^2) B_2/B_6 - f_2 + \bar{\omega}_\beta^2 F + \frac{h_b}{g_b^2} \frac{F^2}{x} \]
\[ t_2 = (1 - c_6 \overline{x}^2)(b_2 - b_0 \overline{\omega}_\beta^2)/b_6 - f_2 + \overline{\omega}_\beta^2 F + f_0/\overline{\omega}_\beta^2 \]

in which
\[ F = f_4 - b_4/b_6 + (b_4 c_6/b_6 - c_4) \overline{x}^2 \]

With some algebraic manipulation, a polynomial of fourth degree in \( \overline{x} \) can be extracted from that equation. The value of \( \overline{x} \) is taken to be the square root of the smallest positive root of that polynomial. The original equations are then used to solve for \( a_1 \) and \( b_1 \), from which \( l_{s1} \) and \( l_{s2} \) are readily obtained.
REFERENCES


15. Theodorsen, T.; and Garrick, I. E.: Mechanism of Flutter--A Theoretical and Experimental Investigation of the Flutter Problem. NACA TR 685, 1940.